project-w2024-1

April 1, 2024

##NLP Project - 1199011 Namratha Jagadeesh

0.1 Project steps

Following steps has been performed:

1. Viewing the video files
2. Transcribing the videos
3. Normalizing the text
4. Extracting key phrases and topics
5. Creating the dashboard

0.2 Useful information

The following cell contains some information that might be useful as you complete this project.

[32]:
```
bucket = "c56161a939430l3396553t1w744137092661-labbucket-rn642jaq01e9"
job_data_access_role = 'arn:aws:iam::744137092661:role/service-role/
  ↪c56161a939430l3396553t1w7-ComprehendDataAccessRole-1P24MSS91ADHP'
```

0.3 1. Viewing the video files

(Go to top)

The source video files are located in the following shared Amazon Simple Storage Service (Amazon S3) bucket.

[1]:
```
!aws s3 ls s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/ | awk -F' ' '/:/
  ↪{print $4}'
```

```
Mod01_Course
Mod02_Intro.mp4
Mod02_Sect01.mp4
Mod02_Sect02.mp4
Mod02_Sect03.mp4
Mod02_Sect04.mp4
Mod02_Sect05.mp4
Mod02_WrapUp.mp4
Mod03_Intro.mp4
Mod03_Sect01.mp4
```

```
Mod03_Sect02_part1.mp4
Mod03_Sect02_part2.mp4
Mod03_Sect02_part3.mp4
Mod03_Sect03_part1.mp4
Mod03_Sect03_part2.mp4
Mod03_Sect03_part3.mp4
Mod03_Sect04_part1.mp4
Mod03_Sect04_part2.mp4
Mod03_Sect04_part3.mp4
Mod03_Sect05.mp4
Mod03_Sect06.mp4
Mod03_Sect07_part1.mp4
Mod03_Sect07_part2.mp4
Mod03_Sect07_part3.mp4
Mod03_Sect08.mp4
Mod03_WrapUp.mp4
Mod04_Intro.mp4
Mod04_Sect01.mp4
Mod04_Sect02_part1.mp4
Mod04_Sect02_part2.mp4
Mod04_Sect02_part3.mp4
Mod04_WrapUp.mp4
Mod05_Intro.mp4
Mod05_Sect01_ver2.mp4
Mod05_Sect02_part1_ver2.mp4
Mod05_Sect02_part2.mp4
Mod05_Sect03_part1.mp4
Mod05_Sect03_part2.mp4
Mod05_Sect03_part3.mp4
Mod05_Sect03_part4_ver2.mp4
Mod05_WrapUp_ver2.mp4
Mod06_Intro.mp4
Mod06_Sect01.mp4
Mod06_Sect02.mp4
Mod06_WrapUp.mp4
Mod07_Sect01.mp4
```

[2]:

```
import subprocess
files = subprocess.check_output("aws s3 ls s3://aws-tc-largeobjects/
 ↪CUR-TF-200-ACMNLP-1/video/ | awk -F' ' '/:/{print $4}' ", shell=True)
```

[3]:

```
files=files.decode('utf-8').strip()
files=files.split('\n')
files
```

[3]:

```
['Mod01_Course',
 'Mod02_Intro.mp4',
 'Mod02_Sect01.mp4',
 'Mod02_Sect02.mp4',
 'Mod02_Sect03.mp4',
 'Mod02_Sect04.mp4',
 'Mod02_Sect05.mp4',
 'Mod02_WrapUp.mp4',
 'Mod03_Intro.mp4',
 'Mod03_Sect01.mp4',
 'Mod03_Sect02_part1.mp4',
 'Mod03_Sect02_part2.mp4',
 'Mod03_Sect02_part3.mp4',
 'Mod03_Sect03_part1.mp4',
 'Mod03_Sect03_part2.mp4',
 'Mod03_Sect03_part3.mp4',
 'Mod03_Sect04_part1.mp4',
 'Mod03_Sect04_part2.mp4',
 'Mod03_Sect04_part3.mp4',
 'Mod03_Sect05.mp4',
 'Mod03_Sect06.mp4',
 'Mod03_Sect07_part1.mp4',
 'Mod03_Sect07_part2.mp4',
 'Mod03_Sect07_part3.mp4',
 'Mod03_Sect08.mp4',
 'Mod03_WrapUp.mp4',
 'Mod04_Intro.mp4',
 'Mod04_Sect01.mp4',
 'Mod04_Sect02_part1.mp4',
 'Mod04_Sect02_part2.mp4',
 'Mod04_Sect02_part3.mp4',
 'Mod04_WrapUp.mp4',
 'Mod05_Intro.mp4',
 'Mod05_Sect01_ver2.mp4',
 'Mod05_Sect02_part1_ver2.mp4',
 'Mod05_Sect02_part2.mp4',
 'Mod05_Sect03_part1.mp4',
 'Mod05_Sect03_part2.mp4',
 'Mod05_Sect03_part3.mp4',
 'Mod05_Sect03_part4_ver2.mp4',
 'Mod05_WrapUp_ver2.mp4',
 'Mod06_Intro.mp4',
 'Mod06_Sect01.mp4',
 'Mod06_Sect02.mp4',
 'Mod06_WrapUp.mp4',
 'Mod07_Sect01.mp4']
```

0.4 2. Transcribing the videos

(Go to top)

[4]:
```
# Downloading the videos from the bucket
!aws s3 cp s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/ ./videos␣
 ↪--recursive
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod02_Intro.mp4 to
videos/Mod02_Intro.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod02_Sect03.mp4 to
videos/Mod02_Sect03.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod02_Sect04.mp4 to
videos/Mod02_Sect04.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod01_Course
Overview.mp4 to videos/Mod01_Course Overview.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod02_Sect02.mp4 to
videos/Mod02_Sect02.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod02_WrapUp.mp4 to
videos/Mod02_WrapUp.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod02_Sect01.mp4 to
videos/Mod02_Sect01.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect02_part1.mp4 to videos/Mod03_Sect02_part1.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod03_Sect01.mp4 to
videos/Mod03_Sect01.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod03_Intro.mp4 to
videos/Mod03_Intro.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect02_part2.mp4 to videos/Mod03_Sect02_part2.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect02_part3.mp4 to videos/Mod03_Sect02_part3.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect04_part1.mp4 to videos/Mod03_Sect04_part1.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod02_Sect05.mp4 to
videos/Mod02_Sect05.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect04_part2.mp4 to videos/Mod03_Sect04_part2.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect03_part3.mp4 to videos/Mod03_Sect03_part3.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod03_Sect06.mp4 to
videos/Mod03_Sect06.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect04_part3.mp4 to videos/Mod03_Sect04_part3.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod03_Sect05.mp4 to
videos/Mod03_Sect05.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect03_part1.mp4 to videos/Mod03_Sect03_part1.mp4
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod03_Sect08.mp4 to
videos/Mod03_Sect08.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect07_part1.mp4 to videos/Mod03_Sect07_part1.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod04_Intro.mp4 to
videos/Mod04_Intro.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod03_WrapUp.mp4 to
videos/Mod03_WrapUp.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect07_part2.mp4 to videos/Mod03_Sect07_part2.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod04_Sect01.mp4 to
videos/Mod04_Sect01.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect03_part2.mp4 to videos/Mod03_Sect03_part2.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod04_WrapUp.mp4 to
videos/Mod04_WrapUp.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod04_Sect02_part2.mp4 to videos/Mod04_Sect02_part2.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod04_Sect02_part1.mp4 to videos/Mod04_Sect02_part1.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod05_Intro.mp4 to
videos/Mod05_Intro.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod03_Sect07_part3.mp4 to videos/Mod03_Sect07_part3.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod05_Sect01_ver2.mp4 to videos/Mod05_Sect01_ver2.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod05_Sect02_part1_ver2.mp4 to
videos/Mod05_Sect02_part1_ver2.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod04_Sect02_part3.mp4 to videos/Mod04_Sect02_part3.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod05_Sect02_part2.mp4 to videos/Mod05_Sect02_part2.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod05_WrapUp_ver2.mp4 to videos/Mod05_WrapUp_ver2.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod05_Sect03_part1.mp4 to videos/Mod05_Sect03_part1.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod06_Intro.mp4 to
videos/Mod06_Intro.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod05_Sect03_part3.mp4 to videos/Mod05_Sect03_part3.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod06_WrapUp.mp4 to
videos/Mod06_WrapUp.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod05_Sect03_part2.mp4 to videos/Mod05_Sect03_part2.mp4
download: s3://aws-tc-largeobjects/CUR-
TF-200-ACMNLP-1/video/Mod05_Sect03_part4_ver2.mp4 to
videos/Mod05_Sect03_part4_ver2.mp4
```

```
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod07_Sect01.mp4 to
videos/Mod07_Sect01.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod06_Sect02.mp4 to
videos/Mod06_Sect02.mp4
download: s3://aws-tc-largeobjects/CUR-TF-200-ACMNLP-1/video/Mod06_Sect01.mp4 to
videos/Mod06_Sect01.mp4
```

[5]:

```python
# Import libraries

!pip install SpeechRecognition
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
!pip install -U textblob
from textblob import TextBlob
!pip install wordcloud
from wordcloud import WordCloud
!pip install moviepy
import moviepy.editor as mp
from IPython.display import Video
import sys
```

```
Collecting SpeechRecognition
  Downloading SpeechRecognition-3.10.3-py2.py3-none-any.whl.metadata (29 kB)
Requirement already satisfied: requests>=2.26.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
SpeechRecognition) (2.31.0)
Requirement already satisfied: typing-extensions in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
SpeechRecognition) (4.9.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests>=2.26.0->SpeechRecognition) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests>=2.26.0->SpeechRecognition) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests>=2.26.0->SpeechRecognition) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests>=2.26.0->SpeechRecognition) (2024.2.2)
Downloading SpeechRecognition-3.10.3-py2.py3-none-any.whl (32.8 MB)
                         32.8/32.8 MB
16.4 MB/s eta 0:00:00:00:0100:01
Installing collected packages: SpeechRecognition
Successfully installed SpeechRecognition-3.10.3
```

```
Collecting textblob
  Downloading textblob-0.18.0.post0-py3-none-any.whl.metadata (4.5 kB)
Requirement already satisfied: nltk>=3.8 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
textblob) (3.8.1)
Requirement already satisfied: click in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
nltk>=3.8->textblob) (8.1.7)
Requirement already satisfied: joblib in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
nltk>=3.8->textblob) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
nltk>=3.8->textblob) (2023.12.25)
Requirement already satisfied: tqdm in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
nltk>=3.8->textblob) (4.66.2)
Downloading textblob-0.18.0.post0-py3-none-any.whl (626 kB)
                         626.3/626.3 kB
27.6 MB/s eta 0:00:00
Installing collected packages: textblob
Successfully installed textblob-0.18.0.post0
Collecting wordcloud
  Downloading wordcloud-1.9.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x8
6_64.whl.metadata (3.4 kB)
Requirement already satisfied: numpy>=1.6.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
wordcloud) (1.22.4)
Requirement already satisfied: pillow in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
wordcloud) (10.2.0)
Requirement already satisfied: matplotlib in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
wordcloud) (3.8.3)
Requirement already satisfied: contourpy>=1.0.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
matplotlib->wordcloud) (1.2.0)
Requirement already satisfied: cycler>=0.10 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
matplotlib->wordcloud) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
matplotlib->wordcloud) (4.49.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
matplotlib->wordcloud) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
matplotlib->wordcloud) (21.3)
Requirement already satisfied: pyparsing>=2.3.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
matplotlib->wordcloud) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six>=1.5 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from python-
dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Downloading
wordcloud-1.9.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (511
kB)
                         511.1/511.1 kB
9.1 MB/s eta 0:00:00:00:01
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.3
Collecting moviepy
  Downloading moviepy-1.0.3.tar.gz (388 kB)
                           388.3/388.3

kB 5.8 MB/s eta 0:00:0000:01
  Preparing metadata (setup.py) … done
Collecting decorator<5.0,>=4.0.2 (from moviepy)
  Downloading decorator-4.4.2-py2.py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: tqdm<5.0,>=4.11.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
moviepy) (4.66.2)
Requirement already satisfied: requests<3.0,>=2.8.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
moviepy) (2.31.0)
Collecting proglog<=1.0.0 (from moviepy)
  Downloading proglog-0.1.10-py3-none-any.whl.metadata (639 bytes)
Requirement already satisfied: numpy>=1.17.3 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
moviepy) (1.22.4)
Requirement already satisfied: imageio<3.0,>=2.5 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
moviepy) (2.34.0)
Collecting imageio_ffmpeg>=0.2.0 (from moviepy)
  Downloading imageio_ffmpeg-0.4.9-py3-none-manylinux2010_x86_64.whl.metadata
(1.7 kB)
Requirement already satisfied: pillow>=8.3.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
imageio<3.0,>=2.5->moviepy) (10.2.0)
Requirement already satisfied: setuptools in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
imageio_ffmpeg>=0.2.0->moviepy) (69.1.0)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0,>=2.8.1->moviepy) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0,>=2.8.1->moviepy) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0,>=2.8.1->moviepy) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0,>=2.8.1->moviepy) (2024.2.2)
Downloading decorator-4.4.2-py2.py3-none-any.whl (9.2 kB)
Downloading imageio_ffmpeg-0.4.9-py3-none-manylinux2010_x86_64.whl (26.9 MB)
                         26.9/26.9 MB
20.1 MB/s eta 0:00:00:00:0100:01
Downloading proglog-0.1.10-py3-none-any.whl (6.1 kB)
Building wheels for collected packages: moviepy
  Building wheel for moviepy (setup.py) … done
  Created wheel for moviepy: filename=moviepy-1.0.3-py3-none-any.whl
size=110720
sha256=acea2eb36ab5b500fb3109615c38b143458508045f5c3f716ff02b68047b0403
  Stored in directory: /home/ec2-user/.cache/pip/wheels/96/32/2d/e10123bd88fbfc0
2fed53cc18c80a171d3c87479ed845fa7c1
Successfully built moviepy
Installing collected packages: proglog, imageio_ffmpeg, decorator, moviepy
  Attempting uninstall: decorator
    Found existing installation: decorator 5.1.1
    Uninstalling decorator-5.1.1:
      Successfully uninstalled decorator-5.1.1
ERROR: pip's dependency resolver does not currently take into account all

the packages that are installed. This behaviour is the source of the following

dependency conflicts.

sparkmagic 0.21.0 requires pandas<2.0.0,>=0.17.1, but you have pandas 2.2.0

which is incompatible.

Successfully installed decorator-4.4.2 imageio_ffmpeg-0.4.9 moviepy-1.0.3
proglog-0.1.10

Matplotlib is building the font cache; this may take a moment.
```

[6]:

```
#Renaming the first file since it has a space in its name
files[0]='Mod01_Course_Overview.mp4'
files
```

```
[6]: ['Mod01_Course_Overview.mp4',
      'Mod02_Intro.mp4',
      'Mod02_Sect01.mp4',
      'Mod02_Sect02.mp4',
      'Mod02_Sect03.mp4',
      'Mod02_Sect04.mp4',
      'Mod02_Sect05.mp4',
      'Mod02_WrapUp.mp4',
      'Mod03_Intro.mp4',
      'Mod03_Sect01.mp4',
      'Mod03_Sect02_part1.mp4',
      'Mod03_Sect02_part2.mp4',
      'Mod03_Sect02_part3.mp4',
      'Mod03_Sect03_part1.mp4',
      'Mod03_Sect03_part2.mp4',
      'Mod03_Sect03_part3.mp4',
      'Mod03_Sect04_part1.mp4',
      'Mod03_Sect04_part2.mp4',
      'Mod03_Sect04_part3.mp4',
      'Mod03_Sect05.mp4',
      'Mod03_Sect06.mp4',
      'Mod03_Sect07_part1.mp4',
      'Mod03_Sect07_part2.mp4',
      'Mod03_Sect07_part3.mp4',
      'Mod03_Sect08.mp4',
      'Mod03_WrapUp.mp4',
      'Mod04_Intro.mp4',
      'Mod04_Sect01.mp4',
      'Mod04_Sect02_part1.mp4',
      'Mod04_Sect02_part2.mp4',
      'Mod04_Sect02_part3.mp4',
      'Mod04_WrapUp.mp4',
      'Mod05_Intro.mp4',
      'Mod05_Sect01_ver2.mp4',
      'Mod05_Sect02_part1_ver2.mp4',
      'Mod05_Sect02_part2.mp4',
      'Mod05_Sect03_part1.mp4',
      'Mod05_Sect03_part2.mp4',
      'Mod05_Sect03_part3.mp4',
      'Mod05_Sect03_part4_ver2.mp4',
      'Mod05_WrapUp_ver2.mp4',
      'Mod06_Intro.mp4',
      'Mod06_Sect01.mp4',
      'Mod06_Sect02.mp4',
      'Mod06_WrapUp.mp4',
      'Mod07_Sect01.mp4']
```

[7]:

```
# Transcribing the videos
from moviepy.editor import *
import os

# Create the output directory if it does not exist
output_dir = "/home/ec2-user/SageMaker/audio/"
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

input_dir = "/home/ec2-user/SageMaker/videos/"
for filename in os.listdir(input_dir):
    if not filename.endswith(".mp4"):
        continue

    input_file = os.path.join(input_dir, filename)
    output_file = os.path.join(output_dir, os.path.splitext(filename)[0] + ".
 ↪wav")

    # Load the video file and extract the audio
    clip = VideoFileClip(input_file)
    audio = clip.audio

    # Write the audio to a WAV file using the PCM 16-bit codec
    audio.write_audiofile(output_file, codec="pcm_s16le")

    # Release the resources used by the clip
    clip.close()
```

```
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod02_Sect05.wav



MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod05_Intro.wav



MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod04_WrapUp.wav



MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod05_Sect01_ver2.wav



MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect04_part2.wav
MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod05_Sect03_part1.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod05_Sect03_part3.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod05_Sect03_part2.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod04_Sect02_part3.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod02_Sect01.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod02_Sect02.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect02_part1.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod05_Sect02_part2.wav


MoviePy - Done.
MoviePy - Writing audio in
/home/ec2-user/SageMaker/audio/Mod05_Sect02_part1_ver2.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect02_part3.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect06.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod05_WrapUp_ver2.wav
MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_WrapUp.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect02_part2.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod06_Sect02.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod04_Sect02_part2.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod04_Sect02_part1.wav


MoviePy - Done.
MoviePy - Writing audio in
/home/ec2-user/SageMaker/audio/Mod05_Sect03_part4_ver2.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod06_WrapUp.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect07_part1.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect08.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod02_Sect04.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect03_part2.wav
MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect04_part1.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect03_part1.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect07_part3.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod02_Sect03.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod06_Intro.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod04_Intro.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod01_Course
Overview.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod06_Sect01.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect03_part3.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod02_WrapUp.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect04_part3.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect07_part2.wav
MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod07_Sect01.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect05.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Intro.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod04_Sect01.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod03_Sect01.wav


MoviePy - Done.
MoviePy - Writing audio in /home/ec2-user/SageMaker/audio/Mod02_Intro.wav


MoviePy - Done.
```

[8]:

```
!pip install pydub
```

```
Collecting pydub
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub
Successfully installed pydub-0.25.1
```

[10]:

```python
import pydub
from pydub import AudioSegment
from pydub.silence import split_on_silence
def get_audio_transcription(path):
    """
    Splitting the large audio file into chunks
    and apply speech recognition on each of these chunks
    """
    # open the audio file using pydub
    sound = AudioSegment.from_wav(path)
    # split audio sound where silence is 700 miliseconds or more and get chunks
    chunks = split_on_silence(sound,
        min_silence_len = 500,
        silence_thresh = sound.dBFS-14,
        keep_silence=500,
    )
    folder_name = "audio-chunks"

    # creating a directory to store the audio chunks
    if not os.path.isdir(folder_name):
        os.mkdir(folder_name)
    whole_text = ""

    # process each chunk
    for i, audio_chunk in enumerate(chunks, start=1):
        # export audio chunk and save it in the created directory.
        chunk_filename = os.path.join(folder_name, f"chunk{i}.wav")
        audio_chunk.export(chunk_filename, format="wav")
        # recognize the chunk
        with sr.AudioFile(chunk_filename) as source:
            audio_listened = r.record(source)
            # try converting it to text
            try:
                text = r.recognize_google(audio_listened)
                print(chunk_filename, ":", text)
            except sr.UnknownValueError as e:
                print("Error:", str(e))
            else:
                text = f"{text.capitalize()}. "
                print(chunk_filename, ":", text)
                whole_text += text
    return whole_text
```

[11]:

```
import speech_recognition as sr
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re

r = sr.Recognizer()
audios_path = '/home/ec2-user/SageMaker/audio/'
audio_files = os.listdir(audios_path)
transcribed_texts = []

for audio_file in audio_files:
    # Load the audio file into an AudioFile object
    text=get_audio_transcription(audios_path+ audio_file)
    transcribed_texts.append(text)
```

```
[nltk_data] Downloading package punkt to /home/ec2-user/nltk_data…
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

```
audio-chunks/chunk1.wav : hi and welcome back
audio-chunks/chunk1.wav : Hi and welcome back.
audio-chunks/chunk2.wav : this is Section 3 and we're going to cover how to
evaluate your data
audio-chunks/chunk2.wav : This is section 3 and we're going to cover how to
evaluate your data.
audio-chunks/chunk3.wav : in this section we'll look at different data formats
and types
audio-chunks/chunk3.wav : In this section we'll look at different data formats
and types.
audio-chunks/chunk4.wav : we'll also look at how you can visualize and analyze
the data before feature engineering
audio-chunks/chunk4.wav : We'll also look at how you can visualize and analyze
the data before feature engineering.
audio-chunks/chunk5.wav : before you can start running statistics on your data
to better understand what you're working with
audio-chunks/chunk5.wav : Before you can start running statistics on your data
to better understand what you're working with.
audio-chunks/chunk6.wav : you need to ensure it's in the right format for
analysis
audio-chunks/chunk6.wav : You need to ensure it's in the right format for
analysis.
audio-chunks/chunk7.wav : for Amazon sagemaker algorithm support training with
data in CSV format
audio-chunks/chunk7.wav : For amazon sagemaker algorithm support training with
data in csv format.
audio-chunks/chunk8.wav : many of the tools you'll use to explore visualize and
analyze the data can also read it in CSV format
audio-chunks/chunk8.wav : Many of the tools you'll use to explore visualize and
analyze the data can also read it in csv format.
audio-chunks/chunk9.wav : generally speaking you'll need to have at least some
domain knowledge for the problem you're trying to solve with machine learning
audio-chunks/chunk9.wav : Generally speaking you'll need to have at least some
domain knowledge for the problem you're trying to solve with machine learning.
audio-chunks/chunk10.wav : for example if you're developing a model to predict
if a set of symptoms indicates a disease
audio-chunks/chunk10.wav : For example if you're developing a model to predict
if a set of symptoms indicates a disease.
audio-chunks/chunk11.wav : you need to know the relationship between the
symptoms and the disease
audio-chunks/chunk11.wav : You need to know the relationship between the
symptoms and the disease.
audio-chunks/chunk12.wav : data typically needs to be a numeric form so machine
```

```
learning algorithms can use the data to make predictions
audio-chunks/chunk12.wav : Data typically needs to be a numeric form so machine
learning algorithms can use the data to make predictions.
audio-chunks/chunk13.wav : we'll look at ways you can convert Text data in the
next section
audio-chunks/chunk13.wav : We'll look at ways you can convert text data in the
next section.
audio-chunks/chunk14.wav : for now we'll just explore the data and try to gain
some insights into the overall data set
audio-chunks/chunk14.wav : For now we'll just explore the data and try to gain
some insights into the overall data set.
audio-chunks/chunk15.wav : one popular open source python library is pandas
audio-chunks/chunk15.wav : One popular open source python library is pandas.
audio-chunks/chunk16.wav : it can take data in various formats
audio-chunks/chunk16.wav : It can take data in various formats.
audio-chunks/chunk17.wav : reformat it and load it into a tabular representation
of your data
audio-chunks/chunk17.wav : Reformat it and load it into a tabular representation
of your data.
audio-chunks/chunk18.wav : presenting it in rows and columns
audio-chunks/chunk18.wav : Presenting it in rows and columns.
audio-chunks/chunk19.wav : some of the formats that pandas can reformat and load
include CSV
audio-chunks/chunk19.wav : Some of the formats that pandas can reformat and load
include csv.
audio-chunks/chunk20.wav : Excel
audio-chunks/chunk20.wav : Excel.
audio-chunks/chunk21.wav : pickle and JavaScript object notation or Json
audio-chunks/chunk21.wav : Pickle and javascript object notation or json.
audio-chunks/chunk22.wav : pandas also has data analysis and manipulation
features
audio-chunks/chunk22.wav : Pandas also has data analysis and manipulation
features.
audio-chunks/chunk23.wav : and we'll use them throughout this module
audio-chunks/chunk23.wav : And we'll use them throughout this module.
audio-chunks/chunk24.wav : loading data can be as simple as the example which
pulls in the CSV file from the specified URL
audio-chunks/chunk24.wav : Loading data can be as simple as the example which
pulls in the csv file from the specified url.
audio-chunks/chunk25.wav : when you load data into pandas
audio-chunks/chunk25.wav : When you load data into pandas.
audio-chunks/chunk26.wav : it's stored as a pandas dataframe
audio-chunks/chunk26.wav : It's stored as a pandas dataframe.
audio-chunks/chunk27.wav : in the pandas documentation a data frame is described
as a general 2D labeled size mutable tabular structure with potentially
heterogeneously typed column
audio-chunks/chunk27.wav : In the pandas documentation a data frame is described
as a general 2d labeled size mutable tabular structure with potentially
```

```
heterogeneously typed column.
audio-chunks/chunk28.wav : a more helpful way to think of a data frame is to
think of it as a spreadsheet or a SQL table
audio-chunks/chunk28.wav : A more helpful way to think of a data frame is to
think of it as a spreadsheet or a sql table.
audio-chunks/chunk29.wav : like a table or spread
audio-chunks/chunk29.wav : Like a table or spread.
audio-chunks/chunk30.wav : a data frame will have rows which are also known as
instances
audio-chunks/chunk30.wav : A data frame will have rows which are also known as
instances.
audio-chunks/chunk31.wav : and we'll have columns which are also known as
attributes
audio-chunks/chunk31.wav : And we'll have columns which are also known as
attributes.
audio-chunks/chunk32.wav : the shape property of a data frame describes the
number of rows and columns it has
audio-chunks/chunk32.wav : The shape property of a data frame describes the
number of rows and columns it has.
audio-chunks/chunk33.wav : each column in a data frame is a series
audio-chunks/chunk33.wav : Each column in a data frame is a series.
audio-chunks/chunk34.wav : a series is a one-dimensional labeled array
audio-chunks/chunk34.wav : A series is a one-dimensional labeled array.
audio-chunks/chunk35.wav : a series can store data of any type
audio-chunks/chunk35.wav : A series can store data of any type.
audio-chunks/chunk36.wav : to learn more about data structures in pandas see the
pandas documentation
audio-chunks/chunk36.wav : To learn more about data structures in pandas see the
pandas documentation.
audio-chunks/chunk37.wav : along with data you can load a data frame with row
labels and column labels
audio-chunks/chunk37.wav : Along with data you can load a data frame with row
labels and column labels.
audio-chunks/chunk38.wav : the row labels are known as an index
audio-chunks/chunk38.wav : The row labels are known as an index.
audio-chunks/chunk39.wav : and the column labels are known as columns
audio-chunks/chunk39.wav : And the column labels are known as columns.
audio-chunks/chunk40.wav : if you loaded your data from a CSV file with a hetero
audio-chunks/chunk40.wav : If you loaded your data from a csv file with a
hetero.
audio-chunks/chunk41.wav : The Columns will be created from the first line of
the file
audio-chunks/chunk41.wav : The columns will be created from the first line of
the file.
audio-chunks/chunk42.wav : you can change that behavior however
audio-chunks/chunk42.wav : You can change that behavior however.
audio-chunks/chunk43.wav : if you don't have column names in the source file you
can pass them as a parameter
```

```
audio-chunks/chunk43.wav : If you don't have column names in the source file you
can pass them as a parameter.
audio-chunks/chunk44.wav : when performing data analysis it's important to make
sure you're using the correct data
audio-chunks/chunk44.wav : When performing data analysis it's important to make
sure you're using the correct data.
audio-chunks/chunk45.wav : in many cases pandas will correctly infer the correct
data types when it loads data and you can move on
audio-chunks/chunk45.wav : In many cases pandas will correctly infer the correct
data types when it loads data and you can move on.
audio-chunks/chunk46.wav : if you have domain knowledge or access to a domain
expert they can often identify data type issues
audio-chunks/chunk46.wav : If you have domain knowledge or access to a domain
expert they can often identify data type issues.
audio-chunks/chunk47.wav : you can use either d-types or the info function to
obtain information on the column types as shown in the example
audio-chunks/chunk47.wav : You can use either d-types or the info function to
obtain information on the column types as shown in the example.
audio-chunks/chunk48.wav : if you don't have the correct data types
audio-chunks/chunk48.wav : If you don't have the correct data types.
audio-chunks/chunk49.wav : you need to figure out why this is the case
audio-chunks/chunk49.wav : You need to figure out why this is the case.
audio-chunks/chunk50.wav : often a numeric column could have been missing data
or it could be a single text value
audio-chunks/chunk50.wav : Often a numeric column could have been missing data
or it could be a single text value.
audio-chunks/chunk51.wav : for example in the car data set
audio-chunks/chunk51.wav : For example in the car data set.
audio-chunks/chunk52.wav : the number of doors can be 23
audio-chunks/chunk52.wav : The number of doors can be 23.
audio-chunks/chunk53.wav : or five more
audio-chunks/chunk53.wav : Or five more.
audio-chunks/chunk54.wav : after you've analyzed the data you can convert a
column to the correct data type using pandas
audio-chunks/chunk54.wav : After you've analyzed the data you can convert a
column to the correct data type using pandas.
audio-chunks/chunk55.wav : that's it for part one of this section
audio-chunks/chunk55.wav : That's it for part one of this section.
audio-chunks/chunk56.wav : we'll see you again for part 2 where we will review
how to describe your data
audio-chunks/chunk56.wav : We'll see you again for part 2 where we will review
how to describe your data.
audio-chunks/chunk1.wav : hi and welcome to module 2 of AWS Academy machine
learning
audio-chunks/chunk1.wav : Hi and welcome to module 2 of aws academy machine
learning.
audio-chunks/chunk2.wav : in this module we're going to introduce machine
learning
```

```
audio-chunks/chunk2.wav : In this module we're going to introduce machine
learning.
audio-chunks/chunk3.wav : will first look at the business problems that can be
solved by Machine learning
audio-chunks/chunk3.wav : Will first look at the business problems that can be
solved by machine learning.
audio-chunks/chunk4.wav : well then talk about terminology
audio-chunks/chunk4.wav : Well then talk about terminology.
audio-chunks/chunk5.wav : process
audio-chunks/chunk5.wav : Process.
audio-chunks/chunk6.wav : tools and some of the challenges you'll face
audio-chunks/chunk6.wav : Tools and some of the challenges you'll face.
audio-chunks/chunk7.wav : after completing this module
audio-chunks/chunk7.wav : After completing this module.
audio-chunks/chunk8.wav : you should be able to recognize how machine learning
and deep learning are part of artificial intelligence
audio-chunks/chunk8.wav : You should be able to recognize how machine learning
and deep learning are part of artificial intelligence.
audio-chunks/chunk9.wav : describe artificial intelligence and machine learning
terminology
audio-chunks/chunk9.wav : Describe artificial intelligence and machine learning
terminology.
audio-chunks/chunk10.wav : identify how machine learning can be used to solve a
business problem
audio-chunks/chunk10.wav : Identify how machine learning can be used to solve a
business problem.
audio-chunks/chunk11.wav : describe the machine learning process
audio-chunks/chunk11.wav : Describe the machine learning process.
audio-chunks/chunk12.wav : list the tools available to data scientists
audio-chunks/chunk12.wav : List the tools available to data scientists.
audio-chunks/chunk13.wav : and identify when to use machine learning instead of
traditional software development methods
audio-chunks/chunk13.wav : And identify when to use machine learning instead of
traditional software development methods.
audio-chunks/chunk14.wav : you're now ready to get started with Section 1
audio-chunks/chunk14.wav : You're now ready to get started with section 1.
audio-chunks/chunk15.wav : see you in the next video
audio-chunks/chunk15.wav : See you in the next video.
audio-chunks/chunk1.wav : welcome back
audio-chunks/chunk1.wav : Welcome back.
audio-chunks/chunk2.wav : in this section will explore image analysis in more
detail
audio-chunks/chunk2.wav : In this section will explore image analysis in more
detail.
audio-chunks/chunk3.wav : and in part 2 we'll take a closer look into video
analysis
audio-chunks/chunk3.wav : And in part 2 we'll take a closer look into video
analysis.
```

```
audio-chunks/chunk4.wav : to
audio-chunks/chunk4.wav : To.
audio-chunks/chunk5.wav : will introduce the main Amazon service will be using
audio-chunks/chunk5.wav : Will introduce the main amazon service will be using.
audio-chunks/chunk6.wav : Amazon recognition
audio-chunks/chunk6.wav : Amazon recognition.
audio-chunks/chunk7.wav : Amazon recognition is a computer vision service that's
based on deep learning
audio-chunks/chunk7.wav : Amazon recognition is a computer vision service that's
based on deep learning.
audio-chunks/chunk8.wav : you can use it to add image and video analysis to your
applications
audio-chunks/chunk8.wav : You can use it to add image and video analysis to your
applications.
audio-chunks/chunk9.wav : there are many uses for Amazon recognition
audio-chunks/chunk9.wav : There are many uses for amazon recognition.
audio-chunks/chunk10.wav : including
audio-chunks/chunk10.wav : Including.
audio-chunks/chunk11.wav : creating searchable image and video libraries
audio-chunks/chunk11.wav : Creating searchable image and video libraries.
audio-chunks/chunk12.wav : Amazon recognition makes both images and stored
videos searchable
audio-chunks/chunk12.wav : Amazon recognition makes both images and stored
videos searchable.
audio-chunks/chunk13.wav : so you can discover the objects and scenes that
appear in them
audio-chunks/chunk13.wav : So you can discover the objects and scenes that
appear in them.
audio-chunks/chunk14.wav : you can use Amazon recognition to build a face based
user verification system
audio-chunks/chunk14.wav : You can use amazon recognition to build a face based
user verification system.
audio-chunks/chunk15.wav : so your applications can confirm user identities by
comparing their live image with a reference image
audio-chunks/chunk15.wav : So your applications can confirm user identities by
comparing their live image with a reference image.
audio-chunks/chunk16.wav : Amazon recognition interprets emotional Expressions
such as happy sad or surprise
audio-chunks/chunk16.wav : Amazon recognition interprets emotional expressions
such as happy sad or surprise.
audio-chunks/chunk17.wav : it can also interpret demographic information from
facial images
audio-chunks/chunk17.wav : It can also interpret demographic information from
facial images.
audio-chunks/chunk18.wav : such as gender
audio-chunks/chunk18.wav : Such as gender.
audio-chunks/chunk19.wav : Amazon recognition can also detect inappropriate
content in both images and stored videos
```

```
audio-chunks/chunk19.wav : Amazon recognition can also detect inappropriate
content in both images and stored videos.
audio-chunks/chunk20.wav : and finally Amazon recognition can recognize and
extract text content from images
audio-chunks/chunk20.wav : And finally amazon recognition can recognize and
extract text content from images.
audio-chunks/chunk21.wav : before we go further here's a quick note on security
audio-chunks/chunk21.wav : Before we go further here's a quick note on security.
audio-chunks/chunk22.wav : you need to check if the applications you build using
Amazon recognition
audio-chunks/chunk22.wav : You need to check if the applications you build using
amazon recognition.
audio-chunks/chunk23.wav : fall under any regulatory restrictions as defined in
your field or country
audio-chunks/chunk23.wav : Fall under any regulatory restrictions as defined in
your field or country.
audio-chunks/chunk24.wav : security and compliance for Amazon recognition is a
shared responsibility between AWS and the customer
audio-chunks/chunk24.wav : Security and compliance for amazon recognition is a
shared responsibility between aws and the customer.
audio-chunks/chunk25.wav : for more information about this topic see the AWS
compliance page
audio-chunks/chunk25.wav : For more information about this topic see the aws
compliance page.
audio-chunks/chunk26.wav : Amazon recognition is an AWS managed service
audio-chunks/chunk26.wav : Amazon recognition is an aws managed service.
audio-chunks/chunk27.wav : with a managed service
audio-chunks/chunk27.wav : With a managed service.
audio-chunks/chunk28.wav : Amazon hosts the machine learning models
audio-chunks/chunk28.wav : Amazon hosts the machine learning models.
audio-chunks/chunk29.wav : maintains an API and scales out to meet demand for
you
audio-chunks/chunk29.wav : Maintains an api and scales out to meet demand for
you.
audio-chunks/chunk30.wav : you can benefit from a set of models that constantly
learn and improve
audio-chunks/chunk30.wav : You can benefit from a set of models that constantly
learn and improve.
audio-chunks/chunk31.wav : also you can focus on building applications that use
the API and optionally training the service to understand your unique business
needs
audio-chunks/chunk31.wav : Also you can focus on building applications that use
the api and optionally training the service to understand your unique business
needs.
audio-chunks/chunk32.wav : there are various resources you can use to access and
interact with Amazon recognition
audio-chunks/chunk32.wav : There are various resources you can use to access and
interact with amazon recognition.
```

```
audio-chunks/chunk33.wav : such as apis sdks and commands for the AWS command
line interface which is also known as the AWS CLI
audio-chunks/chunk33.wav : Such as apis sdks and commands for the aws command
line interface which is also known as the aws cli.
audio-chunks/chunk34.wav : the language is supported by the SDK
audio-chunks/chunk34.wav : The language is supported by the sdk.
audio-chunks/chunk35.wav : include JavaScript
audio-chunks/chunk35.wav : Include javascript.
audio-chunks/chunk36.wav : python
audio-chunks/chunk36.wav : Python.
audio-chunks/chunk37.wav : PHP
audio-chunks/chunk37.wav : Php.
audio-chunks/chunk38.wav : dotnet
audio-chunks/chunk38.wav : Dotnet.
Error:
audio-chunks/chunk40.wav : jav
audio-chunks/chunk40.wav : Jav.
Error:
audio-chunks/chunk42.wav : no j
audio-chunks/chunk42.wav : No j.
audio-chunks/chunk43.wav : AMC plus plus
audio-chunks/chunk43.wav : Amc plus plus.
audio-chunks/chunk44.wav : finally
audio-chunks/chunk44.wav : Finally.
audio-chunks/chunk45.wav : Amazon recognition integrates with other AWS services
audio-chunks/chunk45.wav : Amazon recognition integrates with other aws
services.
audio-chunks/chunk46.wav : for example if you need storage
audio-chunks/chunk46.wav : For example if you need storage.
audio-chunks/chunk47.wav : you can use Amazon simple storage service or S3
audio-chunks/chunk47.wav : You can use amazon simple storage service or s3.
audio-chunks/chunk48.wav : for authentication and authorization
audio-chunks/chunk48.wav : For authentication and authorization.
audio-chunks/chunk49.wav : you can use AWS identity and access management
audio-chunks/chunk49.wav : You can use aws identity and access management.
Error:
audio-chunks/chunk51.wav : this diagram illustrates an image search feature
where users can take pictures and get information about the real estate
properties of viewing
audio-chunks/chunk51.wav : This diagram illustrates an image search feature
where users can take pictures and get information about the real estate
properties of viewing.
Error:
audio-chunks/chunk53.wav : the user takes a picture with their mobile device
audio-chunks/chunk53.wav : The user takes a picture with their mobile device.
audio-chunks/chunk54.wav : the user then initiates a search
audio-chunks/chunk54.wav : The user then initiates a search.
audio-chunks/chunk55.wav : which causes the application to upload to Amazon S3
```

```
audio-chunks/chunk55.wav : Which causes the application to upload to amazon s3.
audio-chunks/chunk56.wav : S3 is configured to call other services when a right
of an occurs
audio-chunks/chunk56.wav : S3 is configured to call other services when a right
of an occurs.
audio-chunks/chunk57.wav : in this case the bucket passes the S3 path of the new
object to AWS Lambda
audio-chunks/chunk57.wav : In this case the bucket passes the s3 path of the new
object to aws lambda.
audio-chunks/chunk58.wav : when the Lambda function is called
audio-chunks/chunk58.wav : When the lambda function is called.
audio-chunks/chunk59.wav : it uses the Amazon recognition SDK to call the
service
audio-chunks/chunk59.wav : It uses the amazon recognition sdk to call the
service.
audio-chunks/chunk60.wav : Amazon recognition
audio-chunks/chunk60.wav : Amazon recognition.
audio-chunks/chunk61.wav : analyzes the image
audio-chunks/chunk61.wav : Analyzes the image.
audio-chunks/chunk62.wav : detect aspects of the property
audio-chunks/chunk62.wav : Detect aspects of the property.
audio-chunks/chunk63.wav : creates labels
audio-chunks/chunk63.wav : Creates labels.
audio-chunks/chunk64.wav : and passes the information back to Lambda as an
object formatted in JavaScript object notation or Json
audio-chunks/chunk64.wav : And passes the information back to lambda as an
object formatted in javascript object notation or json.
audio-chunks/chunk65.wav : Lambda then stores the labels and confidence score in
Amazon elastic service which is also known as Amazon ES
audio-chunks/chunk65.wav : Lambda then stores the labels and confidence score in
amazon elastic service which is also known as amazon es.
audio-chunks/chunk66.wav : application users can now identify aspects of a
property using the objects that were detected in the image
audio-chunks/chunk66.wav : Application users can now identify aspects of a
property using the objects that were detected in the image.
audio-chunks/chunk67.wav : in this example architecture
audio-chunks/chunk67.wav : In this example architecture.
audio-chunks/chunk68.wav : the system checks uploaded images for inappropriate
content
audio-chunks/chunk68.wav : The system checks uploaded images for inappropriate
content.
audio-chunks/chunk69.wav : like the previous example processing begins when the
user uploads content
audio-chunks/chunk69.wav : Like the previous example processing begins when the
user uploads content.
Error:
audio-chunks/chunk71.wav : the user uploads an image to Amazon S3
audio-chunks/chunk71.wav : The user uploads an image to amazon s3.
```

```
audio-chunks/chunk72.wav : second
audio-chunks/chunk72.wav : Second.
audio-chunks/chunk73.wav : the S3 bucket is configured to call a Lambda function
when an object is written to the bucket
audio-chunks/chunk73.wav : The s3 bucket is configured to call a lambda function
when an object is written to the bucket.
Error:
audio-chunks/chunk75.wav : Lambda calls Amazon recognition via the SDK
audio-chunks/chunk75.wav : Lambda calls amazon recognition via the sdk.
audio-chunks/chunk76.wav : Amazon recognition then analyzes the images for
inappropriate content and sends the response back to Lambda
audio-chunks/chunk76.wav : Amazon recognition then analyzes the images for
inappropriate content and sends the response back to lambda.
Error:
audio-chunks/chunk78.wav : if the content is appropriate the content is approved
audio-chunks/chunk78.wav : If the content is appropriate the content is
approved.
Error:
audio-chunks/chunk80.wav : if the content isn't appropriate the content can be
sent for manual inspection
audio-chunks/chunk80.wav : If the content isn't appropriate the content can be
sent for manual inspection.
audio-chunks/chunk81.wav : and finally if the content isn't approved a
notification is sent to the user
audio-chunks/chunk81.wav : And finally if the content isn't approved a
notification is sent to the user.
audio-chunks/chunk82.wav : in this final use case
audio-chunks/chunk82.wav : In this final use case.
audio-chunks/chunk83.wav : the system analyzes a video feed for sentiment
analysis
audio-chunks/chunk83.wav : The system analyzes a video feed for sentiment
analysis.
Error:
audio-chunks/chunk85.wav : an in-store camera captures video that's been sent to
a back office or a cloud-based application
audio-chunks/chunk85.wav : An in-store camera captures video that's been sent to
a back office or a cloud-based application.
audio-chunks/chunk86.wav : typically an application like this uses Amazon
Kinesis to stream the video
audio-chunks/chunk86.wav : Typically an application like this uses amazon
kinesis to stream the video.
audio-chunks/chunk87.wav : second the application uses the SDK to send the video
to Amazon recognition for further analysis
audio-chunks/chunk87.wav : Second the application uses the sdk to send the video
to amazon recognition for further analysis.
audio-chunks/chunk88.wav : visual sentiment is extracted along with other
attributes such as age
audio-chunks/chunk88.wav : Visual sentiment is extracted along with other
```

```
attributes such as age.
Error:
audio-chunks/chunk90.wav : the discovered attributes are sent to Amazon kinesis
audio-chunks/chunk90.wav : The discovered attributes are sent to amazon kinesis.
Error:
audio-chunks/chunk92.wav : a Lambda function extracts the data from the Stream
audio-chunks/chunk92.wav : A lambda function extracts the data from the stream.
Error:
audio-chunks/chunk94.wav : the data is then written to S3
audio-chunks/chunk94.wav : The data is then written to s3.
audio-chunks/chunk95.wav : next
audio-chunks/chunk95.wav : Next.
audio-chunks/chunk96.wav : the data is loaded into Amazon redshift on a regular
basis
audio-chunks/chunk96.wav : The data is loaded into amazon redshift on a regular
basis.
audio-chunks/chunk97.wav : and finally
audio-chunks/chunk97.wav : And finally.
audio-chunks/chunk98.wav : tools like Amazon quicksight can be used to generate
reports from the data
audio-chunks/chunk98.wav : Tools like amazon quicksight can be used to generate
reports from the data.
audio-chunks/chunk99.wav : Amazon recognition is designed to integrate into your
applications through the API and sdks
audio-chunks/chunk99.wav : Amazon recognition is designed to integrate into your
applications through the api and sdks.
audio-chunks/chunk100.wav : API operations are provided for detecting labels
audio-chunks/chunk100.wav : Api operations are provided for detecting labels.
audio-chunks/chunk101.wav : faces
audio-chunks/chunk101.wav : Faces.
audio-chunks/chunk102.wav : recognizing celebrities and detecting unsafe images
audio-chunks/chunk102.wav : Recognizing celebrities and detecting unsafe images.
audio-chunks/chunk103.wav : to perform a prediction
audio-chunks/chunk103.wav : To perform a prediction.
audio-chunks/chunk104.wav : provide the service with an image object in Amazon
S3 or upload a bite stream of an image
audio-chunks/chunk104.wav : Provide the service with an image object in amazon
s3 or upload a bite stream of an image.
audio-chunks/chunk105.wav : images can be in jpeg or PNG formats
audio-chunks/chunk105.wav : Images can be in jpeg or png formats.
audio-chunks/chunk106.wav : Amazon recognition processes the image
audio-chunks/chunk106.wav : Amazon recognition processes the image.
audio-chunks/chunk107.wav : performs the prediction and returns a Json object
with the results
audio-chunks/chunk107.wav : Performs the prediction and returns a json object
with the results.
audio-chunks/chunk108.wav : when Amazon recognition performs predictions
audio-chunks/chunk108.wav : When amazon recognition performs predictions.
```

```
audio-chunks/chunk109.wav : it often returns multiple labels
audio-chunks/chunk109.wav : It often returns multiple labels.
audio-chunks/chunk110.wav : each label has a confidence level
audio-chunks/chunk110.wav : Each label has a confidence level.
audio-chunks/chunk111.wav : this confidence level indicates How likely the label
was found in the image
audio-chunks/chunk111.wav : This confidence level indicates how likely the label
was found in the image.
audio-chunks/chunk112.wav : like this example shows
audio-chunks/chunk112.wav : Like this example shows.
audio-chunks/chunk113.wav : labels can also have hierarchies
audio-chunks/chunk113.wav : Labels can also have hierarchies.
audio-chunks/chunk114.wav : when you find instances of objects
audio-chunks/chunk114.wav : When you find instances of objects.
audio-chunks/chunk115.wav : you need to understand where the detected object is
in the image
audio-chunks/chunk115.wav : You need to understand where the detected object is
in the image.
audio-chunks/chunk116.wav : for each instance the results from Amazon
recognition include a bounding box that contains the starting coordinate of top
audio-chunks/chunk116.wav : For each instance the results from amazon
recognition include a bounding box that contains the starting coordinate of top.
audio-chunks/chunk117.wav : left
audio-chunks/chunk117.wav : Left.
audio-chunks/chunk118.wav : and box dimensions of
audio-chunks/chunk118.wav : And box dimensions of.
audio-chunks/chunk119.wav : height
audio-chunks/chunk119.wav : Height.
audio-chunks/chunk120.wav : like the example you can use this information to
determine the location of the detected object in the image
audio-chunks/chunk120.wav : Like the example you can use this information to
determine the location of the detected object in the image.
audio-chunks/chunk121.wav : it's important to note that all findings contain a
confidence score
audio-chunks/chunk121.wav : It's important to note that all findings contain a
confidence score.
audio-chunks/chunk122.wav : you can use the confidence score in your
applications
audio-chunks/chunk122.wav : You can use the confidence score in your
applications.
audio-chunks/chunk123.wav : to tune your response to predictions
audio-chunks/chunk123.wav : To tune your response to predictions.
audio-chunks/chunk124.wav : with a higher score it's more likely that the object
was correctly labeled
audio-chunks/chunk124.wav : With a higher score it's more likely that the object
was correctly labeled.
audio-chunks/chunk125.wav : that's it for part 1 of this section we'll see you
again for part 2 where we'll explore facial detection
```

```
audio-chunks/chunk125.wav : That's it for part 1 of this section we'll see you
again for part 2 where we'll explore facial detection.
audio-chunks/chunk1.wav : welcome back
audio-chunks/chunk1.wav : Welcome back.
audio-chunks/chunk2.wav : it's now time to review the module and wrap it up
audio-chunks/chunk2.wav : It's now time to review the module and wrap it up.
audio-chunks/chunk3.wav : in summary in this module you learn how to describe
the NLP use cases
audio-chunks/chunk3.wav : In summary in this module you learn how to describe
the nlp use cases.
audio-chunks/chunk4.wav : better solved by using managed Amazon ml services
audio-chunks/chunk4.wav : Better solved by using managed amazon ml services.
audio-chunks/chunk5.wav : and describe the managed ml services available for NLP
audio-chunks/chunk5.wav : And describe the managed ml services available for
nlp.
audio-chunks/chunk6.wav : good job
audio-chunks/chunk6.wav : Good job.
audio-chunks/chunk7.wav : thanks for watching we'll see you in the next module
audio-chunks/chunk7.wav : Thanks for watching we'll see you in the next module.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring data collection by reviewing
how to secure your data
audio-chunks/chunk2.wav : Will continue exploring data collection by reviewing
how to secure your data.
audio-chunks/chunk3.wav : it's important to consider the security of your data
audio-chunks/chunk3.wav : It's important to consider the security of your data.
audio-chunks/chunk4.wav : go to data sets used in this course are all public
audio-chunks/chunk4.wav : Go to data sets used in this course are all public.
audio-chunks/chunk5.wav : real data about customer transactions or health
records need to be kept secure
audio-chunks/chunk5.wav : Real data about customer transactions or health
records need to be kept secure.
audio-chunks/chunk6.wav : you can use AWS identity and access management which
is also known as iam
audio-chunks/chunk6.wav : You can use aws identity and access management which
is also known as iam.
audio-chunks/chunk7.wav : it's a service that controls access to resources
audio-chunks/chunk7.wav : It's a service that controls access to resources.
audio-chunks/chunk8.wav : make sure you're securing your data within AWS
correctly so you can avoid data breaches
audio-chunks/chunk8.wav : Make sure you're securing your data within aws
correctly so you can avoid data breaches.
audio-chunks/chunk9.wav : the diagram shows a simple IAM policy that allows only
read access to a specific S3 bucket for the listed role
audio-chunks/chunk9.wav : The diagram shows a simple iam policy that allows only
read access to a specific s3 bucket for the listed role.
audio-chunks/chunk10.wav : in addition to controlling access to data you need to
```

```
make sure your data is secure
audio-chunks/chunk10.wav : In addition to controlling access to data you need to
make sure your data is secure.
audio-chunks/chunk11.wav : it's a good practice and it might also be legally
required for certain data types
audio-chunks/chunk11.wav : It's a good practice and it might also be legally
required for certain data types.
audio-chunks/chunk12.wav : such as financial data or Healthcare records
audio-chunks/chunk12.wav : Such as financial data or healthcare records.
audio-chunks/chunk13.wav : AWS provides encryption features for storage services
audio-chunks/chunk13.wav : Aws provides encryption features for storage
services.
audio-chunks/chunk14.wav : typically for data that's at rest or in transit
audio-chunks/chunk14.wav : Typically for data that's at rest or in transit.
audio-chunks/chunk15.wav : you can often meet these encryption requirements by
enabling encryption on the object or service you want to protect
audio-chunks/chunk15.wav : You can often meet these encryption requirements by
enabling encryption on the object or service you want to protect.
audio-chunks/chunk16.wav : for data in transit you must use secure transports
like secure socket Slayer transport layer security or SSL TLS
audio-chunks/chunk16.wav : For data in transit you must use secure transports
like secure socket slayer transport layer security or ssl tls.
audio-chunks/chunk17.wav : another aspect to consider is compliance audits
audio-chunks/chunk17.wav : Another aspect to consider is compliance audits.
audio-chunks/chunk18.wav : when dealing with data from regulated Industries
audio-chunks/chunk18.wav : When dealing with data from regulated industries.
audio-chunks/chunk19.wav : you'll often need to audit access to the data
audio-chunks/chunk19.wav : You'll often need to audit access to the data.
audio-chunks/chunk20.wav : AWS cloud trail is a service that enables governance
audio-chunks/chunk20.wav : Aws cloud trail is a service that enables governance.
audio-chunks/chunk21.wav : compliance
audio-chunks/chunk21.wav : Compliance.
audio-chunks/chunk22.wav : operational auditing and risk auditing of your AWS
account
audio-chunks/chunk22.wav : Operational auditing and risk auditing of your aws
account.
audio-chunks/chunk23.wav : with cloudtrail you can log
audio-chunks/chunk23.wav : With cloudtrail you can log.
audio-chunks/chunk24.wav : continuously Monitor and retain account activity
related to actions across your entire AWS infrastructure
audio-chunks/chunk24.wav : Continuously monitor and retain account activity
related to actions across your entire aws infrastructure.
audio-chunks/chunk25.wav : cloudtrail provides an event history of your AWS
account activity
audio-chunks/chunk25.wav : Cloudtrail provides an event history of your aws
account activity.
audio-chunks/chunk26.wav : including actions taken through the AWS Management
console
```

```
audio-chunks/chunk26.wav : Including actions taken through the aws management
console.
audio-chunks/chunk27.wav : AWS SDK
audio-chunks/chunk27.wav : Aws sdk.
audio-chunks/chunk28.wav : command line tools and other AWS services
audio-chunks/chunk28.wav : Command line tools and other aws services.
audio-chunks/chunk29.wav : this event history simplify security analysis
audio-chunks/chunk29.wav : This event history simplify security analysis.
audio-chunks/chunk30.wav : resource change tracking and troubleshooting
audio-chunks/chunk30.wav : Resource change tracking and troubleshooting.
audio-chunks/chunk31.wav : you can also use cloudtrail to detect unusual
activity in your AWS accounts
audio-chunks/chunk31.wav : You can also use cloudtrail to detect unusual
activity in your aws accounts.
audio-chunks/chunk32.wav : all these features can help you simplify operational
analysis and troubleshoot
audio-chunks/chunk32.wav : All these features can help you simplify operational
analysis and troubleshoot.
audio-chunks/chunk33.wav : here are the key takeaways for this section
audio-chunks/chunk33.wav : Here are the key takeaways for this section.
audio-chunks/chunk34.wav : we looked at the first step in solving machine
learning problems
audio-chunks/chunk34.wav : We looked at the first step in solving machine
learning problems.
audio-chunks/chunk35.wav : obtaining the data required to train your machine
learning model
audio-chunks/chunk35.wav : Obtaining the data required to train your machine
learning model.
audio-chunks/chunk36.wav : we also reviewed how ETL can be used to obtain data
from multiple sources
audio-chunks/chunk36.wav : We also reviewed how etl can be used to obtain data
from multiple sources.
audio-chunks/chunk37.wav : services like AWS glue can make it easy to obtain
data from multiple data stores
audio-chunks/chunk37.wav : Services like aws glue can make it easy to obtain
data from multiple data stores.
audio-chunks/chunk38.wav : finally make sure you understand your security
requirements
audio-chunks/chunk38.wav : Finally make sure you understand your security
requirements.
audio-chunks/chunk39.wav : these are based on both business need and any
regulatory requirements
audio-chunks/chunk39.wav : These are based on both business need and any
regulatory requirements.
audio-chunks/chunk40.wav : also make sure your data is secure
audio-chunks/chunk40.wav : Also make sure your data is secure.
audio-chunks/chunk41.wav : only authorized users should be able to access your
data and it should be encrypted where possible
```

```
audio-chunks/chunk41.wav : Only authorized users should be able to access your
data and it should be encrypted where possible.
audio-chunks/chunk42.wav : that's it for section 2
audio-chunks/chunk42.wav : That's it for section 2.
audio-chunks/chunk43.wav : we'll see you in the next video
audio-chunks/chunk43.wav : We'll see you in the next video.
audio-chunks/chunk1.wav : welcome back to AWS Academy machine learning
audio-chunks/chunk1.wav : Welcome back to aws academy machine learning.
audio-chunks/chunk2.wav : this is module 5 and we have a great topic for you
today computer vision
audio-chunks/chunk2.wav : This is module 5 and we have a great topic for you
today computer vision.
audio-chunks/chunk3.wav : in this module we'll start with an overview of the
computer vision space
audio-chunks/chunk3.wav : In this module we'll start with an overview of the
computer vision space.
audio-chunks/chunk4.wav : and you'll learn about some of the use cases and
terminology
audio-chunks/chunk4.wav : And you'll learn about some of the use cases and
terminology.
audio-chunks/chunk5.wav : next
audio-chunks/chunk5.wav : Next.
audio-chunks/chunk6.wav : will explore details about analyzing image and video
with managed services from Amazon web services or AWS
audio-chunks/chunk6.wav : Will explore details about analyzing image and video
with managed services from amazon web services or aws.
audio-chunks/chunk7.wav : finally
audio-chunks/chunk7.wav : Finally.
audio-chunks/chunk8.wav : we'll look at how you can use your own customized data
sets for performing object detection
audio-chunks/chunk8.wav : We'll look at how you can use your own customized data
sets for performing object detection.
audio-chunks/chunk9.wav : at the end of this module
audio-chunks/chunk9.wav : At the end of this module.
audio-chunks/chunk10.wav : you'll be able to describe the use cases for computer
vision
audio-chunks/chunk10.wav : You'll be able to describe the use cases for computer
vision.
audio-chunks/chunk11.wav : describe the Amazon management learning services
available for image and video analysis
audio-chunks/chunk11.wav : Describe the amazon management learning services
available for image and video analysis.
audio-chunks/chunk12.wav : list the steps required to prepare a custom data set
for object detection
audio-chunks/chunk12.wav : List the steps required to prepare a custom data set
for object detection.
audio-chunks/chunk13.wav : describe how Amazon sagemaker ground truth
audio-chunks/chunk13.wav : Describe how amazon sagemaker ground truth.
```

```
audio-chunks/chunk14.wav : can be used to prepare a custom data set
audio-chunks/chunk14.wav : Can be used to prepare a custom data set.
audio-chunks/chunk15.wav : and finally
audio-chunks/chunk15.wav : And finally.
audio-chunks/chunk16.wav : use Amazon recognition to perform facial detection
audio-chunks/chunk16.wav : Use amazon recognition to perform facial detection.
audio-chunks/chunk17.wav : thanks for watching we'll see you in the next video
audio-chunks/chunk17.wav : Thanks for watching we'll see you in the next video.
audio-chunks/chunk1.wav : in this section we'll look at preparing custom data
sets for computer vision
audio-chunks/chunk1.wav : In this section we'll look at preparing custom data
sets for computer vision.
audio-chunks/chunk2.wav : so you can detect custom objects
audio-chunks/chunk2.wav : So you can detect custom objects.
audio-chunks/chunk3.wav : one challenge of using a pre-built model
audio-chunks/chunk3.wav : One challenge of using a pre-built model.
audio-chunks/chunk4.wav : is that it will only find images it was trained to
find
audio-chunks/chunk4.wav : Is that it will only find images it was trained to
find.
audio-chunks/chunk5.wav : though Amazon recognition was trained with tens of
millions of images it can't detect objects that it wasn't trained on
audio-chunks/chunk5.wav : Though amazon recognition was trained with tens of
millions of images it can't detect objects that it wasn't trained on.
audio-chunks/chunk6.wav : for example
audio-chunks/chunk6.wav : For example.
audio-chunks/chunk7.wav : consider the aid of hearts playing card
audio-chunks/chunk7.wav : Consider the aid of hearts playing card.
audio-chunks/chunk8.wav : if you run this car do Amazon recognition
audio-chunks/chunk8.wav : If you run this car do amazon recognition.
audio-chunks/chunk9.wav : the results show various attributes
audio-chunks/chunk9.wav : The results show various attributes.
audio-chunks/chunk10.wav : however none of the labels are playing card or eight
of hearts
audio-chunks/chunk10.wav : However none of the labels are playing card or eight
of hearts.
audio-chunks/chunk11.wav : if you want Amazon recognition to detect images in
your problem domain
audio-chunks/chunk11.wav : If you want amazon recognition to detect images in
your problem domain.
audio-chunks/chunk12.wav : you must train the model with your images
audio-chunks/chunk12.wav : You must train the model with your images.
audio-chunks/chunk13.wav : so in this section
audio-chunks/chunk13.wav : So in this section.
audio-chunks/chunk14.wav : you'll learn how to train Amazon recognition with
images from your problem domain
audio-chunks/chunk14.wav : You'll learn how to train amazon recognition with
images from your problem domain.
```

```
audio-chunks/chunk15.wav : do you focus only on using Amazon recognition here
audio-chunks/chunk15.wav : Do you focus only on using amazon recognition here.
audio-chunks/chunk16.wav : you'll encounter a similar process if you use other
pre-trained models
audio-chunks/chunk16.wav : You'll encounter a similar process if you use other
pre-trained models.
audio-chunks/chunk17.wav : training a computer vision algorithm to recognize
images
audio-chunks/chunk17.wav : Training a computer vision algorithm to recognize
images.
audio-chunks/chunk18.wav : requires a large input data set which isn't practical
for most organizations
audio-chunks/chunk18.wav : Requires a large input data set which isn't practical
for most organizations.
audio-chunks/chunk19.wav : many machine learning problems today
audio-chunks/chunk19.wav : Many machine learning problems today.
audio-chunks/chunk20.wav : can be solved by training existing models
audio-chunks/chunk20.wav : Can be solved by training existing models.
audio-chunks/chunk21.wav : or you can use a managed service like Amazon
recognition custom labels
audio-chunks/chunk21.wav : Or you can use a managed service like amazon
recognition custom labels.
audio-chunks/chunk22.wav : like other machine learning processes
audio-chunks/chunk22.wav : Like other machine learning processes.
audio-chunks/chunk23.wav : you need to train Amazon recognition so it recognizes
scenes and objects that are in a specific domain
audio-chunks/chunk23.wav : You need to train amazon recognition so it recognizes
scenes and objects that are in a specific domain.
audio-chunks/chunk24.wav : you'll need both a training data set and a test data
set that contain labeled images
audio-chunks/chunk24.wav : You'll need both a training data set and a test data
set that contain labeled images.
audio-chunks/chunk25.wav : if you have images that need labels
audio-chunks/chunk25.wav : If you have images that need labels.
audio-chunks/chunk26.wav : you can use Amazon recognition custom labels
audio-chunks/chunk26.wav : You can use amazon recognition custom labels.
audio-chunks/chunk27.wav : to simplify your labeling tasks
audio-chunks/chunk27.wav : To simplify your labeling tasks.
audio-chunks/chunk28.wav : for example
audio-chunks/chunk28.wav : For example.
audio-chunks/chunk29.wav : it provides a UI for labeling images
audio-chunks/chunk29.wav : It provides a ui for labeling images.
audio-chunks/chunk30.wav : which includes a feature you can use to draw bounding
boxes around images
audio-chunks/chunk30.wav : Which includes a feature you can use to draw bounding
boxes around images.
audio-chunks/chunk31.wav : it can also help find objects and scenes that are
unique to your business needs
```

```
audio-chunks/chunk31.wav : It can also help find objects and scenes that are
unique to your business needs.
audio-chunks/chunk32.wav : you can use it to classify images
audio-chunks/chunk32.wav : You can use it to classify images.
audio-chunks/chunk33.wav : or detect objects within an image
audio-chunks/chunk33.wav : Or detect objects within an image.
audio-chunks/chunk34.wav : say you want to identify specific machine parts in
images
audio-chunks/chunk34.wav : Say you want to identify specific machine parts in
images.
audio-chunks/chunk35.wav : such as turbochargers or torque converters
audio-chunks/chunk35.wav : Such as turbochargers or torque converters.
audio-chunks/chunk36.wav : you can collect pictures of each kind of machine
audio-chunks/chunk36.wav : You can collect pictures of each kind of machine.
audio-chunks/chunk37.wav : and use them to train your model
audio-chunks/chunk37.wav : And use them to train your model.
audio-chunks/chunk38.wav : Amazon recognition custom labels
audio-chunks/chunk38.wav : Amazon recognition custom labels.
audio-chunks/chunk39.wav : also includes Automated machine learning capabilities
that handle the machine learning process for you
audio-chunks/chunk39.wav : Also includes automated machine learning capabilities
that handle the machine learning process for you.
audio-chunks/chunk40.wav : when you provide training images
audio-chunks/chunk40.wav : When you provide training images.
audio-chunks/chunk41.wav : the service can automatically load and inspect the
data
audio-chunks/chunk41.wav : The service can automatically load and inspect the
data.
audio-chunks/chunk42.wav : select the correct machine learning algorithms
audio-chunks/chunk42.wav : Select the correct machine learning algorithms.
audio-chunks/chunk43.wav : train a model
audio-chunks/chunk43.wav : Train a model.
audio-chunks/chunk44.wav : and provide model performance metrics
audio-chunks/chunk44.wav : And provide model performance metrics.
audio-chunks/chunk45.wav : when you finish training your model
audio-chunks/chunk45.wav : When you finish training your model.
audio-chunks/chunk46.wav : you can then evaluate your custom models performance
on your test set
audio-chunks/chunk46.wav : You can then evaluate your custom models performance
on your test set.
audio-chunks/chunk47.wav : each image in the test set has a side-by-side
comparison of the models prediction versus the label it assigned
audio-chunks/chunk47.wav : Each image in the test set has a side-by-side
comparison of the models prediction versus the label it assigned.
audio-chunks/chunk48.wav : there are also detailed performance metrics for you
to review
audio-chunks/chunk48.wav : There are also detailed performance metrics for you
to review.
```

```
audio-chunks/chunk49.wav : you can start using your model immediately for image
analysis
audio-chunks/chunk49.wav : You can start using your model immediately for image
analysis.
audio-chunks/chunk50.wav : or you can iterate and retrain new versions with more
images to refine the model
audio-chunks/chunk50.wav : Or you can iterate and retrain new versions with more
images to refine the model.
audio-chunks/chunk51.wav : after you start using your model you can track your
predictions
audio-chunks/chunk51.wav : After you start using your model you can track your
predictions.
audio-chunks/chunk52.wav : correct any mistakes
audio-chunks/chunk52.wav : Correct any mistakes.
audio-chunks/chunk53.wav : and use the feedback data to retrain new model
versions and improve their performance
audio-chunks/chunk53.wav : And use the feedback data to retrain new model
versions and improve their performance.
audio-chunks/chunk54.wav : so how do you label images
audio-chunks/chunk54.wav : So how do you label images.
audio-chunks/chunk55.wav : the diagram shows a typical process for training a
computer vision model
audio-chunks/chunk55.wav : The diagram shows a typical process for training a
computer vision model.
audio-chunks/chunk56.wav : which includes the Amazon recognition custom labels
feature
audio-chunks/chunk56.wav : Which includes the amazon recognition custom labels
feature.
audio-chunks/chunk57.wav : will step through this in some detail
audio-chunks/chunk57.wav : Will step through this in some detail.
audio-chunks/chunk58.wav : the process of developing a custom model to analyze
images requires time expertise and resources
audio-chunks/chunk58.wav : The process of developing a custom model to analyze
images requires time expertise and resources.
audio-chunks/chunk59.wav : it often takes months to complete
audio-chunks/chunk59.wav : It often takes months to complete.
audio-chunks/chunk60.wav : it can also require thousands or tens of thousands of
hand-labeled images
audio-chunks/chunk60.wav : It can also require thousands or tens of thousands of
hand-labeled images.
audio-chunks/chunk61.wav : so the model has enough data to make accurate
decisions
audio-chunks/chunk61.wav : So the model has enough data to make accurate
decisions.
audio-chunks/chunk62.wav : it can take months to generate and gather this data
audio-chunks/chunk62.wav : It can take months to generate and gather this data.
audio-chunks/chunk63.wav : and it can require large teams of laborers to prepare
it for use in machine learning
```

```
audio-chunks/chunk63.wav : And it can require large teams of laborers to prepare
it for use in machine learning.
audio-chunks/chunk64.wav : Amazon recognition custom labels Builds on the
existing capabilities of Amazon recognition
audio-chunks/chunk64.wav : Amazon recognition custom labels builds on the
existing capabilities of amazon recognition.
audio-chunks/chunk65.wav : which is already trained on tens of millions of
images across many categories
audio-chunks/chunk65.wav : Which is already trained on tens of millions of
images across many categories.
audio-chunks/chunk66.wav : instead of thousands of images
audio-chunks/chunk66.wav : Instead of thousands of images.
audio-chunks/chunk67.wav : you can upload a small set of training images that
are specific to your use case
audio-chunks/chunk67.wav : You can upload a small set of training images that
are specific to your use case.
audio-chunks/chunk68.wav : typically you'd use a few hundred images for this
audio-chunks/chunk68.wav : Typically you'd use a few hundred images for this.
audio-chunks/chunk69.wav : you can use the AWS Management console to upload
training images
audio-chunks/chunk69.wav : You can use the aws management console to upload
training images.
audio-chunks/chunk70.wav : if your images are already labeled Amazon recognition
custom labels can begin training your model
audio-chunks/chunk70.wav : If your images are already labeled amazon recognition
custom labels can begin training your model.
audio-chunks/chunk71.wav : if they're not
audio-chunks/chunk71.wav : If they're not.
audio-chunks/chunk72.wav : you can label the images
audio-chunks/chunk72.wav : You can label the images.
audio-chunks/chunk73.wav : directly in the labeling interface
audio-chunks/chunk73.wav : Directly in the labeling interface.
audio-chunks/chunk74.wav : or you can use Amazon sagemaker ground truth to label
them for you
audio-chunks/chunk74.wav : Or you can use amazon sagemaker ground truth to label
them for you.
audio-chunks/chunk75.wav : tell me more on that shortly
audio-chunks/chunk75.wav : Tell me more on that shortly.
audio-chunks/chunk76.wav : Amazon recognition custom labels works best when you
use different models for different domains
audio-chunks/chunk76.wav : Amazon recognition custom labels works best when you
use different models for different domains.
audio-chunks/chunk77.wav : for example if you need to detect both machine parts
and plant health
audio-chunks/chunk77.wav : For example if you need to detect both machine parts
and plant health.
audio-chunks/chunk78.wav : you just two different models
audio-chunks/chunk78.wav : You just two different models.
```

```
audio-chunks/chunk79.wav : images you select for training should be similar to
the images that will be used for inference
audio-chunks/chunk79.wav : Images you select for training should be similar to
the images that will be used for inference.
audio-chunks/chunk80.wav : use images that use various lighting conditions
audio-chunks/chunk80.wav : Use images that use various lighting conditions.
audio-chunks/chunk81.wav : backgrounds and resolutions
audio-chunks/chunk81.wav : Backgrounds and resolutions.
audio-chunks/chunk82.wav : ideally your training images will mirror images you'd
want to perform detection on
audio-chunks/chunk82.wav : Ideally your training images will mirror images you'd
want to perform detection on.
audio-chunks/chunk83.wav : if you can use the same Source like you'd use in
production that works best
audio-chunks/chunk83.wav : If you can use the same source like you'd use in
production that works best.
audio-chunks/chunk84.wav : the documentation includes additional guidelines on
image type
audio-chunks/chunk84.wav : The documentation includes additional guidelines on
image type.
audio-chunks/chunk85.wav : what's the weather there jpegs or pngs and other
properties like image size and resolution
audio-chunks/chunk85.wav : What's the weather there jpegs or pngs and other
properties like image size and resolution.
audio-chunks/chunk86.wav : that's it for part 1 of this section
audio-chunks/chunk86.wav : That's it for part 1 of this section.
audio-chunks/chunk87.wav : we'll see you again for part 2 where will review how
to create the training data set
audio-chunks/chunk87.wav : We'll see you again for part 2 where will review how
to create the training data set.
audio-chunks/chunk1.wav : hi and welcome back
audio-chunks/chunk1.wav : Hi and welcome back.
audio-chunks/chunk2.wav : in this section we're going to look at the types of
business problems machine learning can help you solve
audio-chunks/chunk2.wav : In this section we're going to look at the types of
business problems machine learning can help you solve.
audio-chunks/chunk3.wav : machine learning is used all across your digital lives
audio-chunks/chunk3.wav : Machine learning is used all across your digital
lives.
audio-chunks/chunk4.wav : your email spam filter is the result of a machine
learning program that was trained with examples of spam and regular email
messages
audio-chunks/chunk4.wav : Your email spam filter is the result of a machine
learning program that was trained with examples of spam and regular email
messages.
audio-chunks/chunk5.wav : based on books you're reading or products you bought
audio-chunks/chunk5.wav : Based on books you're reading or products you bought.
audio-chunks/chunk6.wav : machine learning programs can predict other books or
```

```
products you likely to be interested in
audio-chunks/chunk6.wav : Machine learning programs can predict other books or
products you likely to be interested in.
audio-chunks/chunk7.wav : again the machine learning program was trained with
data from other readers habits and purchases
audio-chunks/chunk7.wav : Again the machine learning program was trained with
data from other readers habits and purchases.
audio-chunks/chunk8.wav : when detecting credit card fra
audio-chunks/chunk8.wav : When detecting credit card fra.
audio-chunks/chunk9.wav : the machine learning program was trained on examples
of transactions that turned out to be fraud along with normal transactions
audio-chunks/chunk9.wav : The machine learning program was trained on examples
of transactions that turned out to be fraud along with normal transactions.
audio-chunks/chunk10.wav : you can probably think of many more examples from
social media applications using facial detection to group your photos
audio-chunks/chunk10.wav : You can probably think of many more examples from
social media applications using facial detection to group your photos.
audio-chunks/chunk11.wav : to detecting brain tumors in brain scans or finding
anomalies in x-rays
audio-chunks/chunk11.wav : To detecting brain tumors in brain scans or finding
anomalies in x-rays.
audio-chunks/chunk12.wav : there are three main types of machine learning
audio-chunks/chunk12.wav : There are three main types of machine learning.
audio-chunks/chunk13.wav : they're supervised learning
audio-chunks/chunk13.wav : They're supervised learning.
audio-chunks/chunk14.wav : we're a model uses known inputs and outputs to
generalize future outputs
audio-chunks/chunk14.wav : We're a model uses known inputs and outputs to
generalize future outputs.
audio-chunks/chunk15.wav : there's unsupervised learning
audio-chunks/chunk15.wav : There's unsupervised learning.
audio-chunks/chunk16.wav : were the model doesn't know inputs or outputs
audio-chunks/chunk16.wav : Were the model doesn't know inputs or outputs.
audio-chunks/chunk17.wav : so it finds patterns in the data without help
audio-chunks/chunk17.wav : So it finds patterns in the data without help.
audio-chunks/chunk18.wav : and there's reinforcement learning
audio-chunks/chunk18.wav : And there's reinforcement learning.
audio-chunks/chunk19.wav : were the model interacts with its environment and
learns to take actions that will maximize Rewards
audio-chunks/chunk19.wav : Were the model interacts with its environment and
learns to take actions that will maximize rewards.
audio-chunks/chunk20.wav : it's important to know the different types of ml
because the type will guide you toward selecting algorithms that make sense for
solving your business problem
audio-chunks/chunk20.wav : It's important to know the different types of ml
because the type will guide you toward selecting algorithms that make sense for
solving your business problem.
audio-chunks/chunk21.wav : let's look more into each of these types
```

```
audio-chunks/chunk21.wav : Let's look more into each of these types.
audio-chunks/chunk22.wav : supervised learning is a popular type of ml because
it's widely applicable
audio-chunks/chunk22.wav : Supervised learning is a popular type of ml because
it's widely applicable.
audio-chunks/chunk23.wav : it's called supervised learning because there needs
to be a supervisor
audio-chunks/chunk23.wav : It's called supervised learning because there needs
to be a supervisor.
audio-chunks/chunk24.wav : a teacher who can show the right answers so to speak
audio-chunks/chunk24.wav : A teacher who can show the right answers so to speak.
audio-chunks/chunk25.wav : like any student a supervised algorithm needs to
learn by example
audio-chunks/chunk25.wav : Like any student a supervised algorithm needs to
learn by example.
audio-chunks/chunk26.wav : essentially it needs a teacher who uses training data
to help it determine the patterns and relationships between the inputs and
outputs
audio-chunks/chunk26.wav : Essentially it needs a teacher who uses training data
to help it determine the patterns and relationships between the inputs and
outputs.
audio-chunks/chunk27.wav : if you want to build an application to detect credit
card fraud
audio-chunks/chunk27.wav : If you want to build an application to detect credit
card fraud.
audio-chunks/chunk28.wav : you need training data that includes examples of
Fraud and examples of normal transactions
audio-chunks/chunk28.wav : You need training data that includes examples of
fraud and examples of normal transactions.
audio-chunks/chunk29.wav : within supervised learning there are different types
of problems
audio-chunks/chunk29.wav : Within supervised learning there are different types
of problems.
audio-chunks/chunk30.wav : classification
audio-chunks/chunk30.wav : Classification.
audio-chunks/chunk31.wav : and regression
audio-chunks/chunk31.wav : And regression.
audio-chunks/chunk32.wav : there are two subtypes of classification problems
audio-chunks/chunk32.wav : There are two subtypes of classification problems.
audio-chunks/chunk33.wav : the first is binary classification
audio-chunks/chunk33.wav : The first is binary classification.
audio-chunks/chunk34.wav : think back to the example with identifying fraudulent
transactions
audio-chunks/chunk34.wav : Think back to the example with identifying fraudulent
transactions.
audio-chunks/chunk35.wav : the target variable in this example is limited to two
options
audio-chunks/chunk35.wav : The target variable in this example is limited to two
```

```
options.
audio-chunks/chunk36.wav : fraudulent or not fraudulent
audio-chunks/chunk36.wav : Fraudulent or not fraudulent.
audio-chunks/chunk37.wav : this is a binary classification problem
audio-chunks/chunk37.wav : This is a binary classification problem.
audio-chunks/chunk38.wav : there are also multi-class classification problems
audio-chunks/chunk38.wav : There are also multi-class classification problems.
audio-chunks/chunk39.wav : these ml problems classify an observation into one of
three or more categories
audio-chunks/chunk39.wav : These ml problems classify an observation into one of
three or more categories.
audio-chunks/chunk40.wav : say that you have an ml model that predicts why a
customer is calling your store so you can reduce the number of transfers needed
before the customer gets to the correct customer support department
audio-chunks/chunk40.wav : Say that you have an ml model that predicts why a
customer is calling your store so you can reduce the number of transfers needed
before the customer gets to the correct customer support department.
audio-chunks/chunk41.wav : in this case
audio-chunks/chunk41.wav : In this case.
audio-chunks/chunk42.wav : the different customer support departments represent
the variety of potential Target variables
audio-chunks/chunk42.wav : The different customer support departments represent
the variety of potential target variables.
audio-chunks/chunk43.wav : which could be many different departments much more
than just two
audio-chunks/chunk43.wav : Which could be many different departments much more
than just two.
audio-chunks/chunk44.wav : there are also regression problems
audio-chunks/chunk44.wav : There are also regression problems.
audio-chunks/chunk45.wav : in a regression problem
audio-chunks/chunk45.wav : In a regression problem.
audio-chunks/chunk46.wav : you're no longer mapping an input to a defined number
of categories
audio-chunks/chunk46.wav : You're no longer mapping an input to a defined number
of categories.
audio-chunks/chunk47.wav : instead you're mapping an input to a continuous value
like an integer
audio-chunks/chunk47.wav : Instead you're mapping an input to a continuous value
like an integer.
audio-chunks/chunk48.wav : one example of an ml regression problem is predicting
the price of a company stock
audio-chunks/chunk48.wav : One example of an ml regression problem is predicting
the price of a company stock.
audio-chunks/chunk49.wav : computer vision is a good example of supervised
learning
audio-chunks/chunk49.wav : Computer vision is a good example of supervised
learning.
audio-chunks/chunk50.wav : is this a
```

```
audio-chunks/chunk50.wav : Is this a.
audio-chunks/chunk51.wav : or a
audio-chunks/chunk51.wav : Or a.
audio-chunks/chunk52.wav : is there a tumor in this x-ray
audio-chunks/chunk52.wav : Is there a tumor in this x-ray.
audio-chunks/chunk53.wav : computer vision is often built with deep learning
models
audio-chunks/chunk53.wav : Computer vision is often built with deep learning
models.
audio-chunks/chunk54.wav : it automates the extraction analysis classification
and understanding of useful information from a single image or a sequence of
images
audio-chunks/chunk54.wav : It automates the extraction analysis classification
and understanding of useful information from a single image or a sequence of
images.
audio-chunks/chunk55.wav : computer vision enables machines to identify people
audio-chunks/chunk55.wav : Computer vision enables machines to identify people.
audio-chunks/chunk56.wav : places and things in images with accuracy at or above
human levels
audio-chunks/chunk56.wav : Places and things in images with accuracy at or above
human levels.
audio-chunks/chunk57.wav : and with greater speed and efficiency
audio-chunks/chunk57.wav : And with greater speed and efficiency.
audio-chunks/chunk58.wav : the image data can take many forms
audio-chunks/chunk58.wav : The image data can take many forms.
audio-chunks/chunk59.wav : search a single images
audio-chunks/chunk59.wav : Search a single images.
audio-chunks/chunk60.wav : video sequences
audio-chunks/chunk60.wav : Video sequences.
audio-chunks/chunk61.wav : views from multiple cameras
audio-chunks/chunk61.wav : Views from multiple cameras.
audio-chunks/chunk62.wav : or three-dimensional data
audio-chunks/chunk62.wav : Or three-dimensional data.
audio-chunks/chunk63.wav : you'll learn more about computer vision later in this
audio-chunks/chunk63.wav : You'll learn more about computer vision later in
this.
audio-chunks/chunk64.wav : will now discuss
audio-chunks/chunk64.wav : Will now discuss.
audio-chunks/chunk65.wav : unsupervised machine learning
audio-chunks/chunk65.wav : Unsupervised machine learning.
audio-chunks/chunk66.wav : sometimes all you have is the data
audio-chunks/chunk66.wav : Sometimes all you have is the data.
audio-chunks/chunk67.wav : there's no supervisor in the room
audio-chunks/chunk67.wav : There's no supervisor in the room.
audio-chunks/chunk68.wav : in unsupervised learning labels aren't provided like
they are with supervised learning
audio-chunks/chunk68.wav : In unsupervised learning labels aren't provided like
they are with supervised learning.
```

```
audio-chunks/chunk69.wav : you don't know all the variables and patterns
audio-chunks/chunk69.wav : You don't know all the variables and patterns.
audio-chunks/chunk70.wav : in these instances the machine has to uncover and
create the labels itself
audio-chunks/chunk70.wav : In these instances the machine has to uncover and
create the labels itself.
audio-chunks/chunk71.wav : these models use the date of the presented with to
detect emerging properties of the entire data set
audio-chunks/chunk71.wav : These models use the date of the presented with to
detect emerging properties of the entire data set.
audio-chunks/chunk72.wav : then they construct patterns from those properties
audio-chunks/chunk72.wav : Then they construct patterns from those properties.
audio-chunks/chunk73.wav : clustering is a common subcategory of unsupervised
learning
audio-chunks/chunk73.wav : Clustering is a common subcategory of unsupervised
learning.
audio-chunks/chunk74.wav : this kind of algorithm groups data into different
clusters based on similar features
audio-chunks/chunk74.wav : This kind of algorithm groups data into different
clusters based on similar features.
audio-chunks/chunk75.wav : it does this to better understand the attributes of a
specific cluster
audio-chunks/chunk75.wav : It does this to better understand the attributes of a
specific cluster.
audio-chunks/chunk76.wav : for example by analyzing customer purchasing habits
audio-chunks/chunk76.wav : For example by analyzing customer purchasing habits.
audio-chunks/chunk77.wav : unsupervised algorithms can identify groups of
customers that are associated with the size tear of a company
audio-chunks/chunk77.wav : Unsupervised algorithms can identify groups of
customers that are associated with the size tear of a company.
audio-chunks/chunk78.wav : the advantage of unsupervised algorithms
audio-chunks/chunk78.wav : The advantage of unsupervised algorithms.
audio-chunks/chunk79.wav : is that they enable you to see patterns in the data
that you weren't aware of before
audio-chunks/chunk79.wav : Is that they enable you to see patterns in the data
that you weren't aware of before.
audio-chunks/chunk80.wav : natural language processing is also known as NLP
audio-chunks/chunk80.wav : Natural language processing is also known as nlp.
audio-chunks/chunk81.wav : this is another area of machine learning that's
experiencing growth
audio-chunks/chunk81.wav : This is another area of machine learning that's
experiencing growth.
audio-chunks/chunk82.wav : if you've ever used Alexa or any other voice
audio-chunks/chunk82.wav : If you've ever used alexa or any other voice.
audio-chunks/chunk83.wav : values NLP to try and answer your question
audio-chunks/chunk83.wav : Values nlp to try and answer your question.
audio-chunks/chunk84.wav : NLP isn't just about speech
audio-chunks/chunk84.wav : Nlp isn't just about speech.
```

```
Error:
audio-chunks/chunk86.wav : NLP shows up in many applications
audio-chunks/chunk86.wav : Nlp shows up in many applications.
audio-chunks/chunk87.wav : for example NLP is used with chat or call center Bots
audio-chunks/chunk87.wav : For example nlp is used with chat or call center
bots.
audio-chunks/chunk88.wav : what are automated systems that help you get your
bank balance or order food from a restaurant
audio-chunks/chunk88.wav : What are automated systems that help you get your
bank balance or order food from a restaurant.
audio-chunks/chunk89.wav : you can use NLP in Translation tools which convert
text between languages
audio-chunks/chunk89.wav : You can use nlp in translation tools which convert
text between languages.
audio-chunks/chunk90.wav : for example you might use applications that translate
menus in real time
audio-chunks/chunk90.wav : For example you might use applications that translate
menus in real time.
audio-chunks/chunk91.wav : NLP is also used in voice to text translations
audio-chunks/chunk91.wav : Nlp is also used in voice to text translations.
audio-chunks/chunk92.wav : which converts spoken words into text
audio-chunks/chunk92.wav : Which converts spoken words into text.
audio-chunks/chunk93.wav : and finally
audio-chunks/chunk93.wav : And finally.
audio-chunks/chunk94.wav : NLP can be used in sentiment analysis
audio-chunks/chunk94.wav : Nlp can be used in sentiment analysis.
audio-chunks/chunk95.wav : which you can use to analyze the sentiment of
comments and reviews of products
audio-chunks/chunk95.wav : Which you can use to analyze the sentiment of
comments and reviews of products.
audio-chunks/chunk96.wav : music and movies
audio-chunks/chunk96.wav : Music and movies.
audio-chunks/chunk97.wav : these sentiments could be used to give the movie an
audience rating
audio-chunks/chunk97.wav : These sentiments could be used to give the movie an
audience rating.
audio-chunks/chunk98.wav : you learn more about NLP later in this
audio-chunks/chunk98.wav : You learn more about nlp later in this.
audio-chunks/chunk99.wav : another kind of machine learning that's been gaining
popularity recently is reinforcement learning
audio-chunks/chunk99.wav : Another kind of machine learning that's been gaining
popularity recently is reinforcement learning.
audio-chunks/chunk100.wav : unlike other machine learning reinforcement learning
continuously improves its model by mining feedback from previous iterations
audio-chunks/chunk100.wav : Unlike other machine learning reinforcement learning
continuously improves its model by mining feedback from previous iterations.
audio-chunks/chunk101.wav : in reinforcement learning an agent continuously
learns through trial and error as it interacts in an environment
```

```
audio-chunks/chunk101.wav : In reinforcement learning an agent continuously
learns through trial and error as it interacts in an environment.
audio-chunks/chunk102.wav : reinforcement learning is broadly useful when the
reward of a desired outcome is known but the path to achieving it isn't
audio-chunks/chunk102.wav : Reinforcement learning is broadly useful when the
reward of a desired outcome is known but the path to achieving it isn't.
audio-chunks/chunk103.wav : Annette path requires a lot of trial and error to
discover
audio-chunks/chunk103.wav : Annette path requires a lot of trial and error to
discover.
audio-chunks/chunk104.wav : take the example of AWS deepracer
audio-chunks/chunk104.wav : Take the example of aws deepracer.
audio-chunks/chunk105.wav : in the AWS deep racer Simulator the agent is the
virtual car
audio-chunks/chunk105.wav : In the aws deep racer simulator the agent is the
virtual car.
audio-chunks/chunk106.wav : the environment is a virtual race track
audio-chunks/chunk106.wav : The environment is a virtual race track.
audio-chunks/chunk107.wav : the actions are throttle and steering inputs to the
car
audio-chunks/chunk107.wav : The actions are throttle and steering inputs to the
car.
audio-chunks/chunk108.wav : and the goal is completing the racetrack as quickly
as possible without deviating from the track
audio-chunks/chunk108.wav : And the goal is completing the racetrack as quickly
as possible without deviating from the track.
audio-chunks/chunk109.wav : the car needs to learn the desired driving behavior
to reach the goal of completing the track
audio-chunks/chunk109.wav : The car needs to learn the desired driving behavior
to reach the goal of completing the track.
audio-chunks/chunk110.wav : for the car to learn this
audio-chunks/chunk110.wav : For the car to learn this.
audio-chunks/chunk111.wav : AWS deepracer teams use rewards to incentivize their
model to learn the desired driving behavior
audio-chunks/chunk111.wav : Aws deepracer teams use rewards to incentivize their
model to learn the desired driving behavior.
audio-chunks/chunk112.wav : in reinforcement learning
audio-chunks/chunk112.wav : In reinforcement learning.
audio-chunks/chunk113.wav : the thing driving the learning is called the agent
audio-chunks/chunk113.wav : The thing driving the learning is called the agent.
audio-chunks/chunk114.wav : in this case it's the AWS deep racer car
audio-chunks/chunk114.wav : In this case it's the aws deep racer car.
audio-chunks/chunk115.wav : the environment is the place where the agent learns
audio-chunks/chunk115.wav : The environment is the place where the agent learns.
audio-chunks/chunk116.wav : which in this example would be the marked race track
audio-chunks/chunk116.wav : Which in this example would be the marked race
track.
audio-chunks/chunk117.wav : when the agent does something in the environment
```

```
that provokes a response such as crossing a boundary it shouldn't cross
audio-chunks/chunk117.wav : When the agent does something in the environment
that provokes a response such as crossing a boundary it shouldn't cross.
audio-chunks/chunk118.wav : that's called an action
audio-chunks/chunk118.wav : That's called an action.
audio-chunks/chunk119.wav : that response is called a reward or penalty
depending on whether the agent did something to be reinforced or discouraged in
the model
audio-chunks/chunk119.wav : That response is called a reward or penalty
depending on whether the agent did something to be reinforced or discouraged in
the model.
audio-chunks/chunk120.wav : as the agent moves within the
audio-chunks/chunk120.wav : As the agent moves within the.
audio-chunks/chunk121.wav : it's action should start receiving more rewards and
fewer penalties until it meets the desired business outcome
audio-chunks/chunk121.wav : It's action should start receiving more rewards and
fewer penalties until it meets the desired business outcome.
audio-chunks/chunk122.wav : self-driving Vehicles bring together many machine
and deep learning algorithms and models to solve the problem of driving from
point A to point B
audio-chunks/chunk122.wav : Self-driving vehicles bring together many machine
and deep learning algorithms and models to solve the problem of driving from
point a to point b.
audio-chunks/chunk123.wav : two of its main tasks are the continuous detection
of the environment and forecasting changes
audio-chunks/chunk123.wav : Two of its main tasks are the continuous detection
of the environment and forecasting changes.
audio-chunks/chunk124.wav : these involve detecting objects
audio-chunks/chunk124.wav : These involve detecting objects.
audio-chunks/chunk125.wav : and localizing and predicting the movement of the
detected objects
audio-chunks/chunk125.wav : And localizing and predicting the movement of the
detected objects.
audio-chunks/chunk126.wav : the outputs of these findings act as inputs to other
systems that make decisions on what they should do with the vehicles various
controls
audio-chunks/chunk126.wav : The outputs of these findings act as inputs to other
systems that make decisions on what they should do with the vehicles various
controls.
audio-chunks/chunk127.wav : there are use cases in self-driving vehicles that
require real-time responses to the environment
audio-chunks/chunk127.wav : There are use cases in self-driving vehicles that
require real-time responses to the environment.
audio-chunks/chunk128.wav : for example if a previously hidden pedestrian walks
out from behind an obstacle the vehicle breaks need to be applied immediately
audio-chunks/chunk128.wav : For example if a previously hidden pedestrian walks
out from behind an obstacle the vehicle breaks need to be applied immediately.
audio-chunks/chunk129.wav : there can be no latency or room for error with these
```

```
actions
audio-chunks/chunk129.wav : There can be no latency or room for error with these
actions.
audio-chunks/chunk130.wav : not every problem should be solved with machine
learning
audio-chunks/chunk130.wav : Not every problem should be solved with machine
learning.
audio-chunks/chunk131.wav : sometimes regular programming will work well for
your needs
audio-chunks/chunk131.wav : Sometimes regular programming will work well for
your needs.
audio-chunks/chunk132.wav : if you're interested in exploring a potential
machine learning solution
audio-chunks/chunk132.wav : If you're interested in exploring a potential
machine learning solution.
audio-chunks/chunk133.wav : look for the existence of large data sets and a
large number of variables
audio-chunks/chunk133.wav : Look for the existence of large data sets and a
large number of variables.
audio-chunks/chunk134.wav : machine learning is often the best choice if you're
uncertain of the business logic or procedures needed to obtain an answer or
accomplish a task
audio-chunks/chunk134.wav : Machine learning is often the best choice if you're
uncertain of the business logic or procedures needed to obtain an answer or
accomplish a task.
audio-chunks/chunk135.wav : machine Learning Systems can be complex
audio-chunks/chunk135.wav : Machine learning systems can be complex.
audio-chunks/chunk136.wav : the supporting infrastructure management support and
Technical expertise need to be in place to help ensure the project success
audio-chunks/chunk136.wav : The supporting infrastructure management support and
technical expertise need to be in place to help ensure the project success.
audio-chunks/chunk137.wav : here are the key takeaways for this section
audio-chunks/chunk137.wav : Here are the key takeaways for this section.
audio-chunks/chunk138.wav : where we explored some machine learning applications
that are already part of everyday life
audio-chunks/chunk138.wav : Where we explored some machine learning applications
that are already part of everyday life.
Error:
audio-chunks/chunk140.wav : machine learning problems can be grouped into three
categories
audio-chunks/chunk140.wav : Machine learning problems can be grouped into three
categories.
audio-chunks/chunk141.wav : supervised learning is where you have training data
where you already know the answer
audio-chunks/chunk141.wav : Supervised learning is where you have training data
where you already know the answer.
audio-chunks/chunk142.wav : unsupervised learning is where you have data but are
looking for insights within the data
```

```
audio-chunks/chunk142.wav : Unsupervised learning is where you have data but are
looking for insights within the data.
audio-chunks/chunk143.wav : reinforcement learning is where the model learns
based on experience and feedback
audio-chunks/chunk143.wav : Reinforcement learning is where the model learns
based on experience and feedback.
audio-chunks/chunk144.wav : most business problems are supervised learning
problems
audio-chunks/chunk144.wav : Most business problems are supervised learning
problems.
audio-chunks/chunk145.wav : that's it for this section we'll see you in the next
video
audio-chunks/chunk145.wav : That's it for this section we'll see you in the next
video.
audio-chunks/chunk1.wav : welcome to module 7 course wrap up
audio-chunks/chunk1.wav : Welcome to module 7 course wrap up.
audio-chunks/chunk2.wav : congratulations on completing the AWS Academy machine
learning course
audio-chunks/chunk2.wav : Congratulations on completing the aws academy machine
learning course.
audio-chunks/chunk3.wav : we'll take a few minutes to review what you've learned
and where you can go from here
audio-chunks/chunk3.wav : We'll take a few minutes to review what you've learned
and where you can go from here.
audio-chunks/chunk4.wav : we're going to start with a review of what you've
learned in this course
audio-chunks/chunk4.wav : We're going to start with a review of what you've
learned in this course.
audio-chunks/chunk5.wav : you learned how to describe machine learning
audio-chunks/chunk5.wav : You learned how to describe machine learning.
audio-chunks/chunk6.wav : Implement a machine learning Pipeline and use Amazon
machine learning services for forecasting computer vision and natural language
processing
audio-chunks/chunk6.wav : Implement a machine learning pipeline and use amazon
machine learning services for forecasting computer vision and natural language
processing.
audio-chunks/chunk7.wav : well done
audio-chunks/chunk7.wav : Well done.
audio-chunks/chunk8.wav : although this course isn't designed to prepare you to
become certified for the AWS certified machine learning specialty
audio-chunks/chunk8.wav : Although this course isn't designed to prepare you to
become certified for the aws certified machine learning specialty.
audio-chunks/chunk9.wav : will review how you can continue to work towards that
certification
audio-chunks/chunk9.wav : Will review how you can continue to work towards that
certification.
audio-chunks/chunk10.wav : AWS certification helps you build credibility and
confidence by validating your Cloud expertise
```

```
audio-chunks/chunk10.wav : Aws certification helps you build credibility and
confidence by validating your cloud expertise.
audio-chunks/chunk11.wav : with an industry recognized credential
audio-chunks/chunk11.wav : With an industry recognized credential.
audio-chunks/chunk12.wav : it also helps organizations identify skilled
professionals who can lead Cloud initiatives by using AWS
audio-chunks/chunk12.wav : It also helps organizations identify skilled
professionals who can lead cloud initiatives by using aws.
audio-chunks/chunk13.wav : you must earn a passing score by taking a proctored
exam to earn an AWS certification
audio-chunks/chunk13.wav : You must earn a passing score by taking a proctored
exam to earn an aws certification.
audio-chunks/chunk14.wav : after receiving a passing score you'll receive your
certification credentials
audio-chunks/chunk14.wav : After receiving a passing score you'll receive your
certification credentials.
audio-chunks/chunk15.wav : AWS certification doesn't publish a list of all
services or features that are covered in a certification exam
audio-chunks/chunk15.wav : Aws certification doesn't publish a list of all
services or features that are covered in a certification exam.
audio-chunks/chunk16.wav : however there's an exam guide for each exam
audio-chunks/chunk16.wav : However there's an exam guide for each exam.
audio-chunks/chunk17.wav : and it lists the current topic areas and objectives
covered in the exam
audio-chunks/chunk17.wav : And it lists the current topic areas and objectives
covered in the exam.
audio-chunks/chunk18.wav : exam guides can be found on the prepare for your AWS
certification exam webpage
audio-chunks/chunk18.wav : Exam guides can be found on the prepare for your aws
certification exam webpage.
audio-chunks/chunk19.wav : you'll be required to update your certification or
recertify every 3 years
audio-chunks/chunk19.wav : You'll be required to update your certification or
recertify every 3 years.
audio-chunks/chunk20.wav : view the AWS certification recertification page for
more details
audio-chunks/chunk20.wav : View the aws certification recertification page for
more details.
audio-chunks/chunk21.wav : the information on this slide is current as of June
2020
audio-chunks/chunk21.wav : The information on this slide is current as of june
2020.
audio-chunks/chunk22.wav : however exams are frequently updated
audio-chunks/chunk22.wav : However exams are frequently updated.
audio-chunks/chunk23.wav : also the details regarding which exams are available
and what topics are tested by each exam are subject to change
audio-chunks/chunk23.wav : Also the details regarding which exams are available
and what topics are tested by each exam are subject to change.
```

```
audio-chunks/chunk24.wav : the AWS certified machine learning specialty means
you can select and justify the appropriate machine learning approach for a given
business problem
audio-chunks/chunk24.wav : The aws certified machine learning specialty means
you can select and justify the appropriate machine learning approach for a given
business problem.
audio-chunks/chunk25.wav : you can also identify appropriate AWS services to
implement machine learning Solutions
audio-chunks/chunk25.wav : You can also identify appropriate aws services to
implement machine learning solutions.
audio-chunks/chunk26.wav : and finally you can design and Implement scalable
cost optimized reliable and secure machine learning Solutions
audio-chunks/chunk26.wav : And finally you can design and implement scalable
cost optimized reliable and secure machine learning solutions.
audio-chunks/chunk27.wav : before sitting for the AWS certified machine learning
specialty exam
audio-chunks/chunk27.wav : Before sitting for the aws certified machine learning
specialty exam.
audio-chunks/chunk28.wav : we recommend that you have the following knowledge
and experience
audio-chunks/chunk28.wav : We recommend that you have the following knowledge
and experience.
Error:
audio-chunks/chunk30.wav : you should have one to two years of experience
developing
audio-chunks/chunk30.wav : You should have one to two years of experience
developing.
audio-chunks/chunk31.wav : architecting or running ml or deep learning workloads
on the AWS cloud
audio-chunks/chunk31.wav : Architecting or running ml or deep learning workloads
on the aws cloud.
audio-chunks/chunk32.wav : your experience should include performing basic
hyperparameter optimization and working with machine learning and deep learning
Frameworks
audio-chunks/chunk32.wav : Your experience should include performing basic
hyperparameter optimization and working with machine learning and deep learning
frameworks.
audio-chunks/chunk33.wav : you should also be able to express the intuition
behind basic ml algorithms
audio-chunks/chunk33.wav : You should also be able to express the intuition
behind basic ml algorithms.
audio-chunks/chunk34.wav : finally
audio-chunks/chunk34.wav : Finally.
audio-chunks/chunk35.wav : you should be able to follow best practices for model
training
audio-chunks/chunk35.wav : You should be able to follow best practices for model
training.
audio-chunks/chunk36.wav : in addition to best practices for deployment and
```

```
operations
audio-chunks/chunk36.wav : In addition to best practices for deployment and
operations.
audio-chunks/chunk37.wav : thanks for watching and congratulations on completing
the AWS Academy machine learning course
audio-chunks/chunk37.wav : Thanks for watching and congratulations on completing
the aws academy machine learning course.
audio-chunks/chunk1.wav : welcome back
audio-chunks/chunk1.wav : Welcome back.
audio-chunks/chunk2.wav : in this section we'll look at some of the tools you'll
be using throughout the rest of this course
audio-chunks/chunk2.wav : In this section we'll look at some of the tools you'll
be using throughout the rest of this course.
audio-chunks/chunk3.wav : before we start this list isn't an exhaustive list of
all the tools available today
audio-chunks/chunk3.wav : Before we start this list isn't an exhaustive list of
all the tools available today.
audio-chunks/chunk4.wav : we're only going to cover them at a high level but
it's a good place to get started
audio-chunks/chunk4.wav : We're only going to cover them at a high level but
it's a good place to get started.
Error:
audio-chunks/chunk6.wav : there's the jupyter notebook
audio-chunks/chunk6.wav : There's the jupyter notebook.
audio-chunks/chunk7.wav : did jupyter notebook is an open source web application
you can use to create and share documents that contain Live code equations
visualizations and narrative text
audio-chunks/chunk7.wav : Did jupyter notebook is an open source web application
you can use to create and share documents that contain live code equations
visualizations and narrative text.
audio-chunks/chunk8.wav : uses include
audio-chunks/chunk8.wav : Uses include.
audio-chunks/chunk9.wav : data cleaning and transformation
audio-chunks/chunk9.wav : Data cleaning and transformation.
audio-chunks/chunk10.wav : numerical simulation
audio-chunks/chunk10.wav : Numerical simulation.
audio-chunks/chunk11.wav : statistical modeling
audio-chunks/chunk11.wav : Statistical modeling.
audio-chunks/chunk12.wav : data visualization
audio-chunks/chunk12.wav : Data visualization.
audio-chunks/chunk13.wav : machine learning and much more
audio-chunks/chunk13.wav : Machine learning and much more.
audio-chunks/chunk14.wav : Jupiter lab is a web-based interactive development
environment for jupyter notebooks
audio-chunks/chunk14.wav : Jupiter lab is a web-based interactive development
environment for jupyter notebooks.
audio-chunks/chunk15.wav : code and data
audio-chunks/chunk15.wav : Code and data.
```

```
audio-chunks/chunk16.wav : Jupiter Labs flexible
audio-chunks/chunk16.wav : Jupiter labs flexible.
audio-chunks/chunk17.wav : you can use it to configure and arrange the user
interface to support a wide range of workflows in data science
audio-chunks/chunk17.wav : You can use it to configure and arrange the user
interface to support a wide range of workflows in data science.
audio-chunks/chunk18.wav : scientific Computing and machine learning
audio-chunks/chunk18.wav : Scientific computing and machine learning.
audio-chunks/chunk19.wav : Jupiter lab is extensible and modular
audio-chunks/chunk19.wav : Jupiter lab is extensible and modular.
audio-chunks/chunk20.wav : you can write plugins that add new components and
integrate with existing ones
audio-chunks/chunk20.wav : You can write plugins that add new components and
integrate with existing ones.
audio-chunks/chunk21.wav : later in this course you'll use Amazon sagemaker
which hosts both Jupiter notebooks and Jupiter lab
audio-chunks/chunk21.wav : Later in this course you'll use amazon sagemaker
which hosts both jupiter notebooks and jupiter lab.
audio-chunks/chunk22.wav : pandas is an open source python Library
audio-chunks/chunk22.wav : Pandas is an open source python library.
audio-chunks/chunk23.wav : it's used for data handling and Analysis
audio-chunks/chunk23.wav : It's used for data handling and analysis.
audio-chunks/chunk24.wav : pandas represents data in a table similar to a
spreadsheet
audio-chunks/chunk24.wav : Pandas represents data in a table similar to a
spreadsheet.
audio-chunks/chunk25.wav : this table is known as a pandas dataframe
audio-chunks/chunk25.wav : This table is known as a pandas dataframe.
audio-chunks/chunk26.wav : matplotlib is a python library for creating
scientific static animated and interactive visualizations in Python
audio-chunks/chunk26.wav : Matplotlib is a python library for creating
scientific static animated and interactive visualizations in python.
audio-chunks/chunk27.wav : you use it to generate plots of your data later in
this course
audio-chunks/chunk27.wav : You use it to generate plots of your data later in
this course.
audio-chunks/chunk28.wav : seaborne is another data visualization library for
python that's built on matplotlib
audio-chunks/chunk28.wav : Seaborne is another data visualization library for
python that's built on matplotlib.
audio-chunks/chunk29.wav : it provides a high-level interface for drawing
attractive and informative statistical graphs
audio-chunks/chunk29.wav : It provides a high-level interface for drawing
attractive and informative statistical graphs.
audio-chunks/chunk30.wav : numpy is one of the fundamental scientific Computing
packages in Python
audio-chunks/chunk30.wav : Numpy is one of the fundamental scientific computing
packages in python.
```

```
audio-chunks/chunk31.wav : it contains functions for n-dimensional Ray objects
audio-chunks/chunk31.wav : It contains functions for n-dimensional ray objects.
audio-chunks/chunk32.wav : it also has useful math function
audio-chunks/chunk32.wav : It also has useful math function.
audio-chunks/chunk33.wav : such as linear algebra
audio-chunks/chunk33.wav : Such as linear algebra.
audio-chunks/chunk34.wav : Fourier transform and random number capabilities
audio-chunks/chunk34.wav : Fourier transform and random number capabilities.
audio-chunks/chunk35.wav : psychic learn is an open source machine learning
library that supports supervised and unsupervised learning
audio-chunks/chunk35.wav : Psychic learn is an open source machine learning
library that supports supervised and unsupervised learning.
audio-chunks/chunk36.wav : it also provides various tools for model fitting
audio-chunks/chunk36.wav : It also provides various tools for model fitting.
audio-chunks/chunk37.wav : data preprocessing model selection and evaluation and
many other utilities
audio-chunks/chunk37.wav : Data preprocessing model selection and evaluation and
many other utilities.
Error:
audio-chunks/chunk39.wav : sci-fi and matplotlib
audio-chunks/chunk39.wav : Sci-fi and matplotlib.
audio-chunks/chunk40.wav : it's a good tool for exploring machine learning
audio-chunks/chunk40.wav : It's a good tool for exploring machine learning.
audio-chunks/chunk41.wav : although you'll only use it to borrow a few functions
in this course you might want to consider exploring it after you complete
audio-chunks/chunk41.wav : Although you'll only use it to borrow a few functions
in this course you might want to consider exploring it after you complete.
audio-chunks/chunk42.wav : moving up from Individual libraries and packages
there are also tools that contain production-ready Frameworks
audio-chunks/chunk42.wav : Moving up from individual libraries and packages
there are also tools that contain production-ready frameworks.
audio-chunks/chunk43.wav : we already mentioned sidekick learn which is a good
library for machine learning
audio-chunks/chunk43.wav : We already mentioned sidekick learn which is a good
library for machine learning.
audio-chunks/chunk44.wav : the framework supported on AWS such as tensorflow and
Keras also include libraries you can use for machine learning
audio-chunks/chunk44.wav : The framework supported on aws such as tensorflow and
keras also include libraries you can use for machine learning.
audio-chunks/chunk45.wav : all the Frameworks listed here are supported on AWS
and can be used from Amazon sagemaker
audio-chunks/chunk45.wav : All the frameworks listed here are supported on aws
and can be used from amazon sagemaker.
audio-chunks/chunk46.wav : AWS also provides compute instances that are tuned
for machine learning in both the cloud and the edge
audio-chunks/chunk46.wav : Aws also provides compute instances that are tuned
for machine learning in both the cloud and the edge.
audio-chunks/chunk47.wav : compute instances can be optimized for Learning and
```

```
inference
audio-chunks/chunk47.wav : Compute instances can be optimized for learning and
inference.
audio-chunks/chunk48.wav : another AWS resource you can use are certain Amazon
machine images or amiss
audio-chunks/chunk48.wav : Another aws resource you can use are certain amazon
machine images or amiss.
audio-chunks/chunk49.wav : we offer pre-packaged Amis that contain many of the
popular Frameworks
audio-chunks/chunk49.wav : We offer pre-packaged amis that contain many of the
popular frameworks.
audio-chunks/chunk50.wav : finally there's Amazon sagemaker which is an AWS
service with many capabilities
audio-chunks/chunk50.wav : Finally there's amazon sagemaker which is an aws
service with many capabilities.
Error:
audio-chunks/chunk52.wav : sagemaker can deploy machine learning instances
running Jupiter notebooks and Jupiter lab
audio-chunks/chunk52.wav : Sagemaker can deploy machine learning instances
running jupiter notebooks and jupiter lab.
audio-chunks/chunk53.wav : it manages the deployment of these compute resources
audio-chunks/chunk53.wav : It manages the deployment of these compute resources.
audio-chunks/chunk54.wav : so you only need to connect to the Jupiter
environment
audio-chunks/chunk54.wav : So you only need to connect to the jupiter
environment.
audio-chunks/chunk55.wav : sagemaker also provides tools for labeling data
audio-chunks/chunk55.wav : Sagemaker also provides tools for labeling data.
audio-chunks/chunk56.wav : training models and hosting train models
audio-chunks/chunk56.wav : Training models and hosting train models.
audio-chunks/chunk57.wav : AWS Marketplace also provides a selection of ready-
to-use model packages and algorithms from third-party machine learning
Developers
audio-chunks/chunk57.wav : Aws marketplace also provides a selection of ready-
to-use model packages and algorithms from third-party machine learning
developers.
audio-chunks/chunk58.wav : AWS also provides a set of management machine
learning services
audio-chunks/chunk58.wav : Aws also provides a set of management machine
learning services.
audio-chunks/chunk59.wav : and you can integrate them into your applications
even if you don't have substantial machine learning experience
audio-chunks/chunk59.wav : And you can integrate them into your applications
even if you don't have substantial machine learning experience.
audio-chunks/chunk60.wav : for computer vision Amazon recognition provides
object and facial recognition for both image and video
audio-chunks/chunk60.wav : For computer vision amazon recognition provides
object and facial recognition for both image and video.
```

```
audio-chunks/chunk61.wav : also Amazon textract can extract text from images
audio-chunks/chunk61.wav : Also amazon textract can extract text from images.
audio-chunks/chunk62.wav : speech services include Amazon Polly which can speak
text
audio-chunks/chunk62.wav : Speech services include amazon polly which can speak
text.
audio-chunks/chunk63.wav : another speech service is Amazon transcribe which
converts spoken audio to
audio-chunks/chunk63.wav : Another speech service is amazon transcribe which
converts spoken audio to.
audio-chunks/chunk64.wav : for language
audio-chunks/chunk64.wav : For language.
audio-chunks/chunk65.wav : Amazon comprehend uses NLP to find insights and
relationships in text
audio-chunks/chunk65.wav : Amazon comprehend uses nlp to find insights and
relationships in text.
audio-chunks/chunk66.wav : also Amazon translate can translate text into
different languages
audio-chunks/chunk66.wav : Also amazon translate can translate text into
different languages.
audio-chunks/chunk67.wav : if you want to work with chat
audio-chunks/chunk67.wav : If you want to work with chat.
audio-chunks/chunk68.wav : Amazon Alexa helps you build interactive
conversational applications that use voice or text
audio-chunks/chunk68.wav : Amazon alexa helps you build interactive
conversational applications that use voice or text.
audio-chunks/chunk69.wav : for forecasting
audio-chunks/chunk69.wav : For forecasting.
audio-chunks/chunk70.wav : Amazon forecast uses machine learning to combine time
series data with additional variables so you can build forecast
audio-chunks/chunk70.wav : Amazon forecast uses machine learning to combine time
series data with additional variables so you can build forecast.
audio-chunks/chunk71.wav : and finally if you'd like to work with
recommendations
audio-chunks/chunk71.wav : And finally if you'd like to work with
recommendations.
audio-chunks/chunk72.wav : Amazon personalized can help you create individual
personalized recommendations for customers
audio-chunks/chunk72.wav : Amazon personalized can help you create individual
personalized recommendations for customers.
audio-chunks/chunk73.wav : these managed Services have already been trained in
many aspects of the problem domain
audio-chunks/chunk73.wav : These managed services have already been trained in
many aspects of the problem domain.
audio-chunks/chunk74.wav : you only need to provide your specific data to get
started
audio-chunks/chunk74.wav : You only need to provide your specific data to get
started.
```

```
audio-chunks/chunk75.wav : we're going to look at many of these managed services
in the second half of this course
audio-chunks/chunk75.wav : We're going to look at many of these managed services
in the second half of this course.
audio-chunks/chunk76.wav : after you learn how to do things on your own
audio-chunks/chunk76.wav : After you learn how to do things on your own.
audio-chunks/chunk77.wav : the key takeaways for the section include these
points
audio-chunks/chunk77.wav : The key takeaways for the section include these
points.
Error:
audio-chunks/chunk79.wav : python is the most popular language for performing
machine learning tasks
audio-chunks/chunk79.wav : Python is the most popular language for performing
machine learning tasks.
audio-chunks/chunk80.wav : Jupiter notebooks provide you with a web-based hosted
development environment for machine learning
audio-chunks/chunk80.wav : Jupiter notebooks provide you with a web-based hosted
development environment for machine learning.
audio-chunks/chunk81.wav : you'll use Jupiter notebooks frequently in machine
learning
audio-chunks/chunk81.wav : You'll use jupiter notebooks frequently in machine
learning.
audio-chunks/chunk82.wav : there are a large number of Open Source tools such as
Pand
audio-chunks/chunk82.wav : There are a large number of open source tools such as
pand.
audio-chunks/chunk83.wav : you'll use often as a machine learning practitioner
audio-chunks/chunk83.wav : You'll use often as a machine learning practitioner.
audio-chunks/chunk84.wav : finally depending upon your requirements you might
start with low-level Frameworks to create your own solution
audio-chunks/chunk84.wav : Finally depending upon your requirements you might
start with low-level frameworks to create your own solution.
audio-chunks/chunk85.wav : you might also use tools such as Amazon sagemaker to
help with some of the heavy lifting
audio-chunks/chunk85.wav : You might also use tools such as amazon sagemaker to
help with some of the heavy lifting.
audio-chunks/chunk86.wav : or you can simply use and adapt one of the managed
Amazon ml services for your specific problem domain
audio-chunks/chunk86.wav : Or you can simply use and adapt one of the managed
amazon ml services for your specific problem domain.
audio-chunks/chunk87.wav : that's it for this video we'll see you in the next
one
audio-chunks/chunk87.wav : That's it for this video we'll see you in the next
one.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring how to evaluate your model
```

```
audio-chunks/chunk2.wav : Will continue exploring how to evaluate your model.
audio-chunks/chunk3.wav : the diagram shows the confusion Matrix of how two
different models performed on the same data
audio-chunks/chunk3.wav : The diagram shows the confusion matrix of how two
different models performed on the same data.
audio-chunks/chunk4.wav : can you tell which one's better
audio-chunks/chunk4.wav : Can you tell which one's better.
audio-chunks/chunk5.wav : which is better isn't a good question to ask
audio-chunks/chunk5.wav : Which is better isn't a good question to ask.
audio-chunks/chunk6.wav : what do you mean by better
audio-chunks/chunk6.wav : What do you mean by better.
audio-chunks/chunk7.wav : does better mean making sure you find all the cats
audio-chunks/chunk7.wav : Does better mean making sure you find all the cats.
audio-chunks/chunk8.wav : even if it means you'll get many false positives
audio-chunks/chunk8.wav : Even if it means you'll get many false positives.
audio-chunks/chunk9.wav : what does better mean making sure the model is the
most accurate
audio-chunks/chunk9.wav : What does better mean making sure the model is the
most accurate.
audio-chunks/chunk10.wav : it's difficult to see just by looking at the two
charts
audio-chunks/chunk10.wav : It's difficult to see just by looking at the two
charts.
audio-chunks/chunk11.wav : what if you're trying several model
audio-chunks/chunk11.wav : What if you're trying several model.
audio-chunks/chunk12.wav : using multiple folds and have hundreds of data points
to compare
audio-chunks/chunk12.wav : Using multiple folds and have hundreds of data points
to compare.
audio-chunks/chunk13.wav : to do that you'll need to calculate more metrics
audio-chunks/chunk13.wav : To do that you'll need to calculate more metrics.
audio-chunks/chunk14.wav : the first metric is sensitivity
audio-chunks/chunk14.wav : The first metric is sensitivity.
audio-chunks/chunk15.wav : this is sometimes referred to as recall
audio-chunks/chunk15.wav : This is sometimes referred to as recall.
audio-chunks/chunk16.wav : hit rate or true positive rate
audio-chunks/chunk16.wav : Hit rate or true positive rate.
audio-chunks/chunk17.wav : sensitivity is the percentage of positive
identification
audio-chunks/chunk17.wav : Sensitivity is the percentage of positive
identification.
audio-chunks/chunk18.wav : in the cat example it represents what percentage of
cats were correctly identified
audio-chunks/chunk18.wav : In the cat example it represents what percentage of
cats were correctly identified.
audio-chunks/chunk19.wav : to calculate sensitivity
audio-chunks/chunk19.wav : To calculate sensitivity.
audio-chunks/chunk20.wav : take the number of true positives for the number of
```

```
positive identification of cats
audio-chunks/chunk20.wav : Take the number of true positives for the number of
positive identification of cats.
audio-chunks/chunk21.wav : and divide that by the total number of actual cats
audio-chunks/chunk21.wav : And divide that by the total number of actual cats.
audio-chunks/chunk22.wav : in this example
audio-chunks/chunk22.wav : In this example.
audio-chunks/chunk23.wav : 60% of cats that were cats were correctly identified
as cats
audio-chunks/chunk23.wav : 60% of cats that were cats were correctly identified
as cats.
audio-chunks/chunk24.wav : specificity is sometimes referred to as selectivity
or true negative rate
audio-chunks/chunk24.wav : Specificity is sometimes referred to as selectivity
or true negative rate.
audio-chunks/chunk25.wav : specificity is the percentage of negatives correctly
identified
audio-chunks/chunk25.wav : Specificity is the percentage of negatives correctly
identified.
audio-chunks/chunk26.wav : in the cat example this is the number of images that
were not cats that were correctly identified as not cats
audio-chunks/chunk26.wav : In the cat example this is the number of images that
were not cats that were correctly identified as not cats.
audio-chunks/chunk27.wav : to calculate specificity
audio-chunks/chunk27.wav : To calculate specificity.
audio-chunks/chunk28.wav : take the number of true negatives
audio-chunks/chunk28.wav : Take the number of true negatives.
audio-chunks/chunk29.wav : and divide that by the total number of actual
negatives
audio-chunks/chunk29.wav : And divide that by the total number of actual
negatives.
audio-chunks/chunk30.wav : so for the example
audio-chunks/chunk30.wav : So for the example.
audio-chunks/chunk31.wav : that's the number of not cats that were correctly
identified
audio-chunks/chunk31.wav : That's the number of not cats that were correctly
identified.
audio-chunks/chunk32.wav : divided by the total number of actual not cats
audio-chunks/chunk32.wav : Divided by the total number of actual not cats.
audio-chunks/chunk33.wav : this means that in the example 64% of not cats were
identified as not cats
audio-chunks/chunk33.wav : This means that in the example 64% of not cats were
identified as not cats.
audio-chunks/chunk34.wav : now that you have these metrics for each model
knowing what your business goal is makes it easier to decide which model to use
audio-chunks/chunk34.wav : Now that you have these metrics for each model
knowing what your business goal is makes it easier to decide which model to use.
audio-chunks/chunk35.wav : which model would you choose if you wanted to make
```

```
sure you'll identify as many cats as possible
audio-chunks/chunk35.wav : Which model would you choose if you wanted to make
sure you'll identify as many cats as possible.
audio-chunks/chunk36.wav : model B would be a good answer if you're not
concerned about having many false positives that is
audio-chunks/chunk36.wav : Model b would be a good answer if you're not
concerned about having many false positives that is.
audio-chunks/chunk37.wav : if you're not concerned about having incorrectly
identified not cats
audio-chunks/chunk37.wav : If you're not concerned about having incorrectly
identified not cats.
audio-chunks/chunk38.wav : which model would you choose if you wanted to make
sure you identified animals that were not cats
audio-chunks/chunk38.wav : Which model would you choose if you wanted to make
sure you identified animals that were not cats.
audio-chunks/chunk39.wav : Model A might work for this scenario
audio-chunks/chunk39.wav : Model a might work for this scenario.
audio-chunks/chunk40.wav : again it would depend on how many false negatives you
can tolerate
audio-chunks/chunk40.wav : Again it would depend on how many false negatives you
can tolerate.
audio-chunks/chunk41.wav : if this was a classification of patients who had
heart disease or not
audio-chunks/chunk41.wav : If this was a classification of patients who had
heart disease or not.
audio-chunks/chunk42.wav : which model would be best
audio-chunks/chunk42.wav : Which model would be best.
audio-chunks/chunk43.wav : this is where it gets interesting
audio-chunks/chunk43.wav : This is where it gets interesting.
audio-chunks/chunk44.wav : a fun website might get a bad reputation if it can't
identify cats correctly
audio-chunks/chunk44.wav : A fun website might get a bad reputation if it can't
identify cats correctly.
audio-chunks/chunk45.wav : but if you're trying to diagnose patience
audio-chunks/chunk45.wav : But if you're trying to diagnose patience.
audio-chunks/chunk46.wav : your focus will probably be very different
audio-chunks/chunk46.wav : Your focus will probably be very different.
audio-chunks/chunk47.wav : it's important to understand the trade-offs you're
making when you decide which model to use
audio-chunks/chunk47.wav : It's important to understand the trade-offs you're
making when you decide which model to use.
audio-chunks/chunk48.wav : there are also other metrics that can help you make
your decisions
audio-chunks/chunk48.wav : There are also other metrics that can help you make
your decisions.
audio-chunks/chunk49.wav : that's it for part 2 of this section we'll see you
again for part 3 where we'll start looking at thresholds
audio-chunks/chunk49.wav : That's it for part 2 of this section we'll see you
```

```
again for part 3 where we'll start looking at thresholds.
Error:
audio-chunks/chunk1.wav : will get started by reviewing what natural language
processing means
audio-chunks/chunk1.wav : Will get started by reviewing what natural language
processing means.
audio-chunks/chunk2.wav : natural language processing is also known as NLP
audio-chunks/chunk2.wav : Natural language processing is also known as nlp.
audio-chunks/chunk3.wav : before we explain what NLP is will consider an example
of NLP
audio-chunks/chunk3.wav : Before we explain what nlp is will consider an example
of nlp.
audio-chunks/chunk4.wav : Amazon Alexa
audio-chunks/chunk4.wav : Amazon alexa.
audio-chunks/chunk5.wav : Alexa works by having a device such as an Amazon Echo
record your words
audio-chunks/chunk5.wav : Alexa works by having a device such as an amazon echo
record your words.
audio-chunks/chunk6.wav : the recording of your speech is sent to Amazon servers
to be analyzed more efficiently
audio-chunks/chunk6.wav : The recording of your speech is sent to amazon servers
to be analyzed more efficiently.
audio-chunks/chunk7.wav : Amazon breaks down your phrase into individual sounds
audio-chunks/chunk7.wav : Amazon breaks down your phrase into individual sounds.
audio-chunks/chunk8.wav : then it connects to a database containing the
pronunciation of various words
audio-chunks/chunk8.wav : Then it connects to a database containing the
pronunciation of various words.
audio-chunks/chunk9.wav : Define which words most closely correspond to the
combination of individual sounds
audio-chunks/chunk9.wav : Define which words most closely correspond to the
combination of individual sounds.
audio-chunks/chunk10.wav : Amazon identifies important words to make sense of
the task
audio-chunks/chunk10.wav : Amazon identifies important words to make sense of
the task.
audio-chunks/chunk11.wav : and Carry Out corresponding functions
audio-chunks/chunk11.wav : And carry out corresponding functions.
audio-chunks/chunk12.wav : for instance
audio-chunks/chunk12.wav : For instance.
audio-chunks/chunk13.wav : if Alexa notices words like outside or temperature
audio-chunks/chunk13.wav : If alexa notices words like outside or temperature.
audio-chunks/chunk14.wav : it will open the Weather Alexa skill
audio-chunks/chunk14.wav : It will open the weather alexa skill.
audio-chunks/chunk15.wav : Amazon servers then send the information back to your
device and Alexa speaks
audio-chunks/chunk15.wav : Amazon servers then send the information back to your
device and alexa speaks.
```

```
audio-chunks/chunk16.wav : MLP is a broad term for a general set of business or
computational problems you can solve with machine learning or ml
audio-chunks/chunk16.wav : Mlp is a broad term for a general set of business or
computational problems you can solve with machine learning or ml.
Error:
audio-chunks/chunk18.wav : NLP systems predate machine learning
audio-chunks/chunk18.wav : Nlp systems predate machine learning.
audio-chunks/chunk19.wav : for example
audio-chunks/chunk19.wav : For example.
audio-chunks/chunk20.wav : speech to text on older pretty smartphone cell phones
used NLP and so did screen readers
audio-chunks/chunk20.wav : Speech to text on older pretty smartphone cell phones
used nlp and so did screen readers.
audio-chunks/chunk21.wav : many NLP systems now use some form of machine
learning
audio-chunks/chunk21.wav : Many nlp systems now use some form of machine
learning.
audio-chunks/chunk22.wav : MLP considers the hierarchical structure of language
audio-chunks/chunk22.wav : Mlp considers the hierarchical structure of language.
audio-chunks/chunk23.wav : words are at the lowest layer in a hierarchy
audio-chunks/chunk23.wav : Words are at the lowest layer in a hierarchy.
audio-chunks/chunk24.wav : a group of words make a phrase
audio-chunks/chunk24.wav : A group of words make a phrase.
audio-chunks/chunk25.wav : in the next level up phrases make a sentence
audio-chunks/chunk25.wav : In the next level up phrases make a sentence.
audio-chunks/chunk26.wav : and ultimately sentences convey ideas
audio-chunks/chunk26.wav : And ultimately sentences convey ideas.
audio-chunks/chunk27.wav : NLP systems face several significant challenges
audio-chunks/chunk27.wav : Nlp systems face several significant challenges.
audio-chunks/chunk28.wav : we'll look at its challenges next
audio-chunks/chunk28.wav : We'll look at its challenges next.
audio-chunks/chunk29.wav : language isn't precise
audio-chunks/chunk29.wav : Language isn't precise.
audio-chunks/chunk30.wav : words can have different meanings based on the other
words that surround them
audio-chunks/chunk30.wav : Words can have different meanings based on the other
words that surround them.
audio-chunks/chunk31.wav : this is known as context
audio-chunks/chunk31.wav : This is known as context.
audio-chunks/chunk32.wav : often the same words or phrases can have multiple
meanings
audio-chunks/chunk32.wav : Often the same words or phrases can have multiple
meanings.
audio-chunks/chunk33.wav : for example consider the term weather
audio-chunks/chunk33.wav : For example consider the term weather.
audio-chunks/chunk34.wav : you could be under the weather
audio-chunks/chunk34.wav : You could be under the weather.
audio-chunks/chunk35.wav : which has a colloquial meaning in English that you're
```

```
sick
audio-chunks/chunk35.wav : Which has a colloquial meaning in english that you're
sick.
audio-chunks/chunk36.wav : or you could say there's wonderful weather outside
audio-chunks/chunk36.wav : Or you could say there's wonderful weather outside.
audio-chunks/chunk37.wav : which means the weather conditions outside are good
audio-chunks/chunk37.wav : Which means the weather conditions outside are good.
audio-chunks/chunk38.wav : the phrase oh really
audio-chunks/chunk38.wav : The phrase oh really.
audio-chunks/chunk39.wav : could convey surprise disagreement and many other
things
audio-chunks/chunk39.wav : Could convey surprise disagreement and many other
things.
audio-chunks/chunk40.wav : it depends on the context and inflection
audio-chunks/chunk40.wav : It depends on the context and inflection.
audio-chunks/chunk41.wav : here are some of the main challenges for NLP
audio-chunks/chunk41.wav : Here are some of the main challenges for nlp.
audio-chunks/chunk42.wav : One Challenge is discovering the structure of the
text
audio-chunks/chunk42.wav : One challenge is discovering the structure of the
text.
audio-chunks/chunk43.wav : one of the first tasks of any NLP application is to
break down the text into meaningful units such as words phrases and sentences
audio-chunks/chunk43.wav : One of the first tasks of any nlp application is to
break down the text into meaningful units such as words phrases and sentences.
audio-chunks/chunk44.wav : another challenge is labeling data
audio-chunks/chunk44.wav : Another challenge is labeling data.
audio-chunks/chunk45.wav : after the system converts the text to data it must
apply labels representing the various parts of speech
audio-chunks/chunk45.wav : After the system converts the text to data it must
apply labels representing the various parts of speech.
audio-chunks/chunk46.wav : every language will require a different labeling
scheme to match the languages grammar
audio-chunks/chunk46.wav : Every language will require a different labeling
scheme to match the languages grammar.
audio-chunks/chunk47.wav : NLP also faces a challenge in representing context
audio-chunks/chunk47.wav : Nlp also faces a challenge in representing context.
audio-chunks/chunk48.wav : because word meaning can depend heavily on context
audio-chunks/chunk48.wav : Because word meaning can depend heavily on context.
audio-chunks/chunk49.wav : any NLP system needs a way to represent it
audio-chunks/chunk49.wav : Any nlp system needs a way to represent it.
audio-chunks/chunk50.wav : this is a large challenge because there are many
contact
audio-chunks/chunk50.wav : This is a large challenge because there are many
contact.
audio-chunks/chunk51.wav : and it's difficult to convert contacts into a form
computers can understand
audio-chunks/chunk51.wav : And it's difficult to convert contacts into a form
```

```
computers can understand.
audio-chunks/chunk52.wav : finally
audio-chunks/chunk52.wav : Finally.
audio-chunks/chunk53.wav : although grammar defines a structure for language
audio-chunks/chunk53.wav : Although grammar defines a structure for language.
audio-chunks/chunk54.wav : the application of grammar is indescribably large in
scope
audio-chunks/chunk54.wav : The application of grammar is indescribably large in
scope.
audio-chunks/chunk55.wav : handling the variation in how language is used by
humans is a major challenge for MLP systems
audio-chunks/chunk55.wav : Handling the variation in how language is used by
humans is a major challenge for mlp systems.
audio-chunks/chunk56.wav : that's where machine learning can have a large impact
audio-chunks/chunk56.wav : That's where machine learning can have a large
impact.
audio-chunks/chunk57.wav : you can apply NLP to a range of problems
audio-chunks/chunk57.wav : You can apply nlp to a range of problems.
audio-chunks/chunk58.wav : some of the more common applications include
audio-chunks/chunk58.wav : Some of the more common applications include.
audio-chunks/chunk59.wav : search applications such as Google or Bing
audio-chunks/chunk59.wav : Search applications such as google or bing.
audio-chunks/chunk60.wav : human machine interactions like Alexa
audio-chunks/chunk60.wav : Human machine interactions like alexa.
audio-chunks/chunk61.wav : sentiment analysis for marketing or political
campaigns
audio-chunks/chunk61.wav : Sentiment analysis for marketing or political
campaigns.
audio-chunks/chunk62.wav : social research based on media analysis
audio-chunks/chunk62.wav : Social research based on media analysis.
audio-chunks/chunk63.wav : and chatbots to mimic human speech in applications
audio-chunks/chunk63.wav : And chatbots to mimic human speech in applications.
audio-chunks/chunk64.wav : you can apply the machine learning development
pipeline you've seen throughout this course
audio-chunks/chunk64.wav : You can apply the machine learning development
pipeline you've seen throughout this course.
audio-chunks/chunk65.wav : When developing an MLP solution
audio-chunks/chunk65.wav : When developing an mlp solution.
audio-chunks/chunk66.wav : the first task is to formulate a problem
audio-chunks/chunk66.wav : The first task is to formulate a problem.
audio-chunks/chunk67.wav : then collect and label data
audio-chunks/chunk67.wav : Then collect and label data.
audio-chunks/chunk68.wav : for NLP
audio-chunks/chunk68.wav : For nlp.
audio-chunks/chunk69.wav : collecting data consists of breaking down the text
into meaningful subsets and labeling the sets
audio-chunks/chunk69.wav : Collecting data consists of breaking down the text
into meaningful subsets and labeling the sets.
```

```
audio-chunks/chunk70.wav : feature engineering is a major part of MLP
applications
audio-chunks/chunk70.wav : Feature engineering is a major part of mlp
applications.
audio-chunks/chunk71.wav : this process gets complicated when you're dealing
with highly irregular or unstructured text
audio-chunks/chunk71.wav : This process gets complicated when you're dealing
with highly irregular or unstructured text.
audio-chunks/chunk72.wav : for example
audio-chunks/chunk72.wav : For example.
audio-chunks/chunk73.wav : say you're building an application to classify
documents
audio-chunks/chunk73.wav : Say you're building an application to classify
documents.
audio-chunks/chunk74.wav : you need to be able to distinguish between the words
with common terms but different meanings
audio-chunks/chunk74.wav : You need to be able to distinguish between the words
with common terms but different meanings.
audio-chunks/chunk75.wav : labeling data in NLP domain is sometimes also called
tagging
audio-chunks/chunk75.wav : Labeling data in nlp domain is sometimes also called
tagging.
audio-chunks/chunk76.wav : in the labeling process
audio-chunks/chunk76.wav : In the labeling process.
audio-chunks/chunk77.wav : you assign individual text strings to different parts
of speech
audio-chunks/chunk77.wav : You assign individual text strings to different parts
of speech.
audio-chunks/chunk78.wav : there are specialized tools you can use for NLP
labeling
audio-chunks/chunk78.wav : There are specialized tools you can use for nlp
labeling.
audio-chunks/chunk79.wav : the first task for an MLP application is to convert
the text to data so it can be analyzed
audio-chunks/chunk79.wav : The first task for an mlp application is to convert
the text to data so it can be analyzed.
audio-chunks/chunk80.wav : you convert
audio-chunks/chunk80.wav : You convert.
audio-chunks/chunk81.wav : by removing words that aren't needed for analysis
from the input text
audio-chunks/chunk81.wav : By removing words that aren't needed for analysis
from the input text.
audio-chunks/chunk82.wav : in the example the words this
audio-chunks/chunk82.wav : In the example the words this.
audio-chunks/chunk83.wav : and is are removed to leave the phrase
audio-chunks/chunk83.wav : And is are removed to leave the phrase.
audio-chunks/chunk84.wav : sample text
audio-chunks/chunk84.wav : Sample text.
```

```
audio-chunks/chunk85.wav : after removing stop words you can normalize text by
converting similar words into a common form
audio-chunks/chunk85.wav : After removing stop words you can normalize text by
converting similar words into a common form.
audio-chunks/chunk86.wav : for example the words run
audio-chunks/chunk86.wav : For example the words run.
Error:
audio-chunks/chunk88.wav : ran and running are all different forms of the word
run
audio-chunks/chunk88.wav : Ran and running are all different forms of the word
run.
audio-chunks/chunk89.wav : you can normalize all instances of these words in a
block of text using the stemming and limitation processes
audio-chunks/chunk89.wav : You can normalize all instances of these words in a
block of text using the stemming and limitation processes.
audio-chunks/chunk90.wav : limitation groups different forms of a word into a
single term
audio-chunks/chunk90.wav : Limitation groups different forms of a word into a
single term.
audio-chunks/chunk91.wav : limitation of the versions of the word run would
group all instances of those forms into a single term
audio-chunks/chunk91.wav : Limitation of the versions of the word run would
group all instances of those forms into a single term.
Error:
audio-chunks/chunk93.wav : stemming on the other hand removes characters that
the stemming algorithm considers unnecessary
audio-chunks/chunk93.wav : Stemming on the other hand removes characters that
the stemming algorithm considers unnecessary.
audio-chunks/chunk94.wav : standing might not work with the Run example as the
form ran might not be recognized as a form of the word run
audio-chunks/chunk94.wav : Standing might not work with the run example as the
form ran might not be recognized as a form of the word run.
audio-chunks/chunk95.wav : after you've normalised the
audio-chunks/chunk95.wav : After you've normalised the.
audio-chunks/chunk96.wav : you can standardize it by removing words that aren't
in the dictionary you're using for analysis
audio-chunks/chunk96.wav : You can standardize it by removing words that aren't
in the dictionary you're using for analysis.
audio-chunks/chunk97.wav : for example you could remove acronym
audio-chunks/chunk97.wav : For example you could remove acronym.
audio-chunks/chunk98.wav : slang in special characters
audio-chunks/chunk98.wav : Slang in special characters.
audio-chunks/chunk99.wav : the natural language toolkit is also known as nltk
audio-chunks/chunk99.wav : The natural language toolkit is also known as nltk.
audio-chunks/chunk100.wav : their python Library provides functions for removing
stop words and normalizing text
audio-chunks/chunk100.wav : Their python library provides functions for removing
stop words and normalizing text.
```

```
audio-chunks/chunk101.wav : another first step in creating an MLP system is to
convert the text into a data collection such as a data frame
audio-chunks/chunk101.wav : Another first step in creating an mlp system is to
convert the text into a data collection such as a data frame.
audio-chunks/chunk102.wav : all MLP libraries provide functions to assist with
this process
audio-chunks/chunk102.wav : All mlp libraries provide functions to assist with
this process.
audio-chunks/chunk103.wav : the example shows using the word tokenize function
audio-chunks/chunk103.wav : The example shows using the word tokenize function.
audio-chunks/chunk104.wav : from the nltk library
audio-chunks/chunk104.wav : From the nltk library.
audio-chunks/chunk105.wav : after you've cleaned up your text and loaded it into
a data frame
audio-chunks/chunk105.wav : After you've cleaned up your text and loaded it into
a data frame.
audio-chunks/chunk106.wav : you can apply one of the NLP models to create
features
audio-chunks/chunk106.wav : You can apply one of the nlp models to create
features.
audio-chunks/chunk107.wav : here are a couple of common models
audio-chunks/chunk107.wav : Here are a couple of common models.
audio-chunks/chunk108.wav : the first model is known as bag of words
audio-chunks/chunk108.wav : The first model is known as bag of words.
audio-chunks/chunk109.wav : this is a simple model for capturing the frequency
of words in a document
audio-chunks/chunk109.wav : This is a simple model for capturing the frequency
of words in a document.
audio-chunks/chunk110.wav : the model creates a key for each word
audio-chunks/chunk110.wav : The model creates a key for each word.
audio-chunks/chunk111.wav : the value of the key is the number of times that
word occurs in the document
audio-chunks/chunk111.wav : The value of the key is the number of times that
word occurs in the document.
audio-chunks/chunk112.wav : the second model is term frequency and inverse
document frequency
audio-chunks/chunk112.wav : The second model is term frequency and inverse
document frequency.
audio-chunks/chunk113.wav : which is also known as tf-idf
audio-chunks/chunk113.wav : Which is also known as tf-idf.
audio-chunks/chunk114.wav : term frequency is a count of how many times a word
appears in a document
audio-chunks/chunk114.wav : Term frequency is a count of how many times a word
appears in a document.
audio-chunks/chunk115.wav : inverse document frequency is the number of times a
word occurs in a group of documents
audio-chunks/chunk115.wav : Inverse document frequency is the number of times a
word occurs in a group of documents.
```

```
audio-chunks/chunk116.wav : these two values are used together to calculate a
weight for the words
audio-chunks/chunk116.wav : These two values are used together to calculate a
weight for the words.
audio-chunks/chunk117.wav : words that frequently appear in many documents have
a lower weight
audio-chunks/chunk117.wav : Words that frequently appear in many documents have
a lower weight.
audio-chunks/chunk118.wav : there are many established models in the NLP field
audio-chunks/chunk118.wav : There are many established models in the nlp field.
audio-chunks/chunk119.wav : the example shows a bag of words model
audio-chunks/chunk119.wav : The example shows a bag of words model.
audio-chunks/chunk120.wav : bag of words is a vector model
audio-chunks/chunk120.wav : Bag of words is a vector model.
audio-chunks/chunk121.wav : Vector models convert each sentence or phrase into a
vector which is a mathematical object that records both directionality and
magnitude
audio-chunks/chunk121.wav : Vector models convert each sentence or phrase into a
vector which is a mathematical object that records both directionality and
magnitude.
audio-chunks/chunk122.wav : in the example a simple sentence is converted into a
vector
audio-chunks/chunk122.wav : In the example a simple sentence is converted into a
vector.
audio-chunks/chunk123.wav : where the frequency of each word is recorded
audio-chunks/chunk123.wav : Where the frequency of each word is recorded.
audio-chunks/chunk124.wav : the word is has a value of 2 because it appears
twice in the sentence
audio-chunks/chunk124.wav : The word is has a value of 2 because it appears
twice in the sentence.
audio-chunks/chunk125.wav : bag of words is often used to classify documents
into different categories
audio-chunks/chunk125.wav : Bag of words is often used to classify documents
into different categories.
audio-chunks/chunk126.wav : it's also used to derive attributes that feed into
NLP applications such as in sentiment analysis
audio-chunks/chunk126.wav : It's also used to derive attributes that feed into
nlp applications such as in sentiment analysis.
audio-chunks/chunk127.wav : there are three broad categories of text analysis
audio-chunks/chunk127.wav : There are three broad categories of text analysis.
Error:
audio-chunks/chunk129.wav : the classification of text is similar to other
classification systems you've seen in this course
audio-chunks/chunk129.wav : The classification of text is similar to other
classification systems you've seen in this course.
audio-chunks/chunk130.wav : text provides the input to a process that extracts
features
audio-chunks/chunk130.wav : Text provides the input to a process that extracts
```

```
features.
audio-chunks/chunk131.wav : then you send the features through a machine
learning algorithm that interacts with a classifier model and infers the
classification
audio-chunks/chunk131.wav : Then you send the features through a machine
learning algorithm that interacts with a classifier model and infers the
classification.
audio-chunks/chunk132.wav : there are many applications for text matching
audio-chunks/chunk132.wav : There are many applications for text matching.
audio-chunks/chunk133.wav : for example autocorrect spelling and grammar
checking are based on text matching
audio-chunks/chunk133.wav : For example autocorrect spelling and grammar
checking are based on text matching.
audio-chunks/chunk134.wav : the algorithm for edit
audio-chunks/chunk134.wav : The algorithm for edit.
audio-chunks/chunk135.wav : also known as the levinstein
audio-chunks/chunk135.wav : Also known as the levinstein.
audio-chunks/chunk136.wav : is frequently used
audio-chunks/chunk136.wav : Is frequently used.
audio-chunks/chunk137.wav : you can drive relationships between different words
or phrases in the text
audio-chunks/chunk137.wav : You can drive relationships between different words
or phrases in the text.
audio-chunks/chunk138.wav : using a process called core reference resolution
audio-chunks/chunk138.wav : Using a process called core reference resolution.
audio-chunks/chunk139.wav : several NLP systems provide python libraries for
deriving relationships
audio-chunks/chunk139.wav : Several nlp systems provide python libraries for
deriving relationships.
audio-chunks/chunk140.wav : one of the biggest challenges for NLP is how to
describe the context for the text
audio-chunks/chunk140.wav : One of the biggest challenges for nlp is how to
describe the context for the text.
audio-chunks/chunk141.wav : consider this example where a user is searching for
the term tablet
audio-chunks/chunk141.wav : Consider this example where a user is searching for
the term tablet.
audio-chunks/chunk142.wav : because the word tablet has at least two distinct
meanings the search engine needs to know which meaning the user has in mind
audio-chunks/chunk142.wav : Because the word tablet has at least two distinct
meanings the search engine needs to know which meaning the user has in mind.
audio-chunks/chunk143.wav : most search engines rely on the most commonly used
context
audio-chunks/chunk143.wav : Most search engines rely on the most commonly used
context.
audio-chunks/chunk144.wav : if the term isn't qualified further
audio-chunks/chunk144.wav : If the term isn't qualified further.
audio-chunks/chunk145.wav : for example
```

```
audio-chunks/chunk145.wav : For example.
audio-chunks/chunk146.wav : by adding another term like medicine
audio-chunks/chunk146.wav : By adding another term like medicine.
audio-chunks/chunk147.wav : or Computing to the search
audio-chunks/chunk147.wav : Or computing to the search.
audio-chunks/chunk148.wav : the process of extracting entities is known as named
entity recognition or any are
audio-chunks/chunk148.wav : The process of extracting entities is known as named
entity recognition or any are.
audio-chunks/chunk149.wav : in any armor has the following functions
audio-chunks/chunk149.wav : In any armor has the following functions.
Error:
audio-chunks/chunk151.wav : it can identify noun phrases using dependency charts
and part of speech tagging
audio-chunks/chunk151.wav : It can identify noun phrases using dependency charts
and part of speech tagging.
audio-chunks/chunk152.wav : it can classify phrases using a classification
algorithm such as word to VEC
audio-chunks/chunk152.wav : It can classify phrases using a classification
algorithm such as word to vec.
audio-chunks/chunk153.wav : finally it can disambiguate entities using a
knowledge graph
audio-chunks/chunk153.wav : Finally it can disambiguate entities using a
knowledge graph.
audio-chunks/chunk154.wav : here's an example of using ner to extract the
entities Titanic
audio-chunks/chunk154.wav : Here's an example of using ner to extract the
entities titanic.
audio-chunks/chunk155.wav : and North Atlantic from the text
audio-chunks/chunk155.wav : And north atlantic from the text.
audio-chunks/chunk156.wav : after the named entities are extracted you can use a
Knowledge Graph to extract meaning
audio-chunks/chunk156.wav : After the named entities are extracted you can use a
knowledge graph to extract meaning.
audio-chunks/chunk157.wav : a Knowledge Graph combines subject matter expertise
with machine learning to drive meaning
audio-chunks/chunk157.wav : A knowledge graph combines subject matter expertise
with machine learning to drive meaning.
audio-chunks/chunk158.wav : the Amazon recommendations engine is an example of a
knowledge graph
audio-chunks/chunk158.wav : The amazon recommendations engine is an example of a
knowledge graph.
audio-chunks/chunk159.wav : here are the main points to remember from the
section
audio-chunks/chunk159.wav : Here are the main points to remember from the
section.
Error:
audio-chunks/chunk161.wav : NLP predates machine learning
```

```
audio-chunks/chunk161.wav : Nlp predates machine learning.
audio-chunks/chunk162.wav : you can use the same ml workflow that you've seen in
other modules for NLP
audio-chunks/chunk162.wav : You can use the same ml workflow that you've seen in
other modules for nlp.
audio-chunks/chunk163.wav : some of the main use cases for NLP are search query
analysis
audio-chunks/chunk163.wav : Some of the main use cases for nlp are search query
analysis.
audio-chunks/chunk164.wav : human machine interaction and marketing and social
research
audio-chunks/chunk164.wav : Human machine interaction and marketing and social
research.
audio-chunks/chunk165.wav : NLP is complicated because human language lacks
precision
audio-chunks/chunk165.wav : Nlp is complicated because human language lacks
precision.
audio-chunks/chunk166.wav : thanks for watching we'll see you in the next video
audio-chunks/chunk166.wav : Thanks for watching we'll see you in the next video.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring image analysis with a closer
look at facial detection
audio-chunks/chunk2.wav : Will continue exploring image analysis with a closer
look at facial detection.
audio-chunks/chunk3.wav : facial detection uses a model that was tuned to
perform predictions specifically for detecting faces and facial features
audio-chunks/chunk3.wav : Facial detection uses a model that was tuned to
perform predictions specifically for detecting faces and facial features.
audio-chunks/chunk4.wav : facial detection has many of the same features as
standard object detection
audio-chunks/chunk4.wav : Facial detection has many of the same features as
standard object detection.
audio-chunks/chunk5.wav : such
audio-chunks/chunk5.wav : Such.
audio-chunks/chunk6.wav : a bounding box
audio-chunks/chunk6.wav : A bounding box.
audio-chunks/chunk7.wav : were the coordinates of the Box surrounding the face
that was detected
audio-chunks/chunk7.wav : Were the coordinates of the box surrounding the face
that was detected.
audio-chunks/chunk8.wav : this will include a value representing the confidence
that the bounding box contains a face
audio-chunks/chunk8.wav : This will include a value representing the confidence
that the bounding box contains a face.
audio-chunks/chunk9.wav : there will be a list of attributes if found
audio-chunks/chunk9.wav : There will be a list of attributes if found.
audio-chunks/chunk10.wav : such as if the face has a beard or if it appears to
```

```
be male or female
audio-chunks/chunk10.wav : Such as if the face has a beard or if it appears to
be male or female.
audio-chunks/chunk11.wav : there will also be a confidence score for these
attributes
audio-chunks/chunk11.wav : There will also be a confidence score for these
attributes.
audio-chunks/chunk12.wav : it can also detect physical emotions like whether the
person is smiling or frowning
audio-chunks/chunk12.wav : It can also detect physical emotions like whether the
person is smiling or frowning.
audio-chunks/chunk13.wav : it's important to understand this classification is
based only on visual Clues
audio-chunks/chunk13.wav : It's important to understand this classification is
based only on visual clues.
audio-chunks/chunk14.wav : and so it might not represent the actual emotion of
the person
audio-chunks/chunk14.wav : And so it might not represent the actual emotion of
the person.
audio-chunks/chunk15.wav : facial landmarks are components of the face such as
eyes and mouth
audio-chunks/chunk15.wav : Facial landmarks are components of the face such as
eyes and mouth.
audio-chunks/chunk16.wav : typical landmarks also include X and Y coordinates
audio-chunks/chunk16.wav : Typical landmarks also include x and y coordinates.
audio-chunks/chunk17.wav : quality describes the brightness and the sharpness of
the face
audio-chunks/chunk17.wav : Quality describes the brightness and the sharpness of
the face.
audio-chunks/chunk18.wav : and pose describes the rotation of the face inside
the image
audio-chunks/chunk18.wav : And pose describes the rotation of the face inside
the image.
audio-chunks/chunk19.wav : again confidence is a feature here
audio-chunks/chunk19.wav : Again confidence is a feature here.
audio-chunks/chunk20.wav : and it's provided for each detected feature
audio-chunks/chunk20.wav : And it's provided for each detected feature.
audio-chunks/chunk21.wav : and remember the feature prediction is based only on
visual observations
audio-chunks/chunk21.wav : And remember the feature prediction is based only on
visual observations.
audio-chunks/chunk22.wav : with Amazon recognition
audio-chunks/chunk22.wav : With amazon recognition.
audio-chunks/chunk23.wav : you can compare two images to determine if they
contain the same person
audio-chunks/chunk23.wav : You can compare two images to determine if they
contain the same person.
audio-chunks/chunk24.wav : comparisons require both a source and a Target image
```

```
audio-chunks/chunk24.wav : Comparisons require both a source and a target image.
audio-chunks/chunk25.wav : the results will include all the faces that were
found
audio-chunks/chunk25.wav : The results will include all the faces that were
found.
audio-chunks/chunk26.wav : and they include information about matching and non-
matching faces
audio-chunks/chunk26.wav : And they include information about matching and non-
matching faces.
Error:
audio-chunks/chunk28.wav : confidence scores indicate How likely each prediction
is
audio-chunks/chunk28.wav : Confidence scores indicate how likely each prediction
is.
audio-chunks/chunk29.wav : Amazon recognition can also search for known faces
audio-chunks/chunk29.wav : Amazon recognition can also search for known faces.
audio-chunks/chunk30.wav : to use this feature you need to train the model by
providing a collection of images to use
audio-chunks/chunk30.wav : To use this feature you need to train the model by
providing a collection of images to use.
audio-chunks/chunk31.wav : after you train the model you can then detect those
people and images you provide
audio-chunks/chunk31.wav : After you train the model you can then detect those
people and images you provide.
audio-chunks/chunk32.wav : to find known faces
audio-chunks/chunk32.wav : To find known faces.
audio-chunks/chunk33.wav : first create a collection
audio-chunks/chunk33.wav : First create a collection.
audio-chunks/chunk34.wav : add faces to the collection
audio-chunks/chunk34.wav : Add faces to the collection.
audio-chunks/chunk35.wav : Amazon recognition will perform facial recognition on
the images you provide
audio-chunks/chunk35.wav : Amazon recognition will perform facial recognition on
the images you provide.
audio-chunks/chunk36.wav : it will then return typical information like the
bounding box coordinates or the confidence score
audio-chunks/chunk36.wav : It will then return typical information like the
bounding box coordinates or the confidence score.
audio-chunks/chunk37.wav : to associate faces with an image
audio-chunks/chunk37.wav : To associate faces with an image.
audio-chunks/chunk38.wav : specify an image ID in the external image ID request
parameter
audio-chunks/chunk38.wav : Specify an image id in the external image id request
parameter.
audio-chunks/chunk39.wav : this could be the file name of the image or another
ID that you create
audio-chunks/chunk39.wav : This could be the file name of the image or another
id that you create.
```

```
audio-chunks/chunk40.wav : after you create your collection
audio-chunks/chunk40.wav : After you create your collection.
audio-chunks/chunk41.wav : you can then use the search Faces by image operation
audio-chunks/chunk41.wav : You can then use the search faces by image operation.
audio-chunks/chunk42.wav : to search for faces from the collection
audio-chunks/chunk42.wav : To search for faces from the collection.
audio-chunks/chunk43.wav : The Returned data contains an array of all faces that
matched
audio-chunks/chunk43.wav : The returned data contains an array of all faces that
matched.
audio-chunks/chunk44.wav : the information includes bounding boxes
audio-chunks/chunk44.wav : The information includes bounding boxes.
audio-chunks/chunk45.wav : confidence scores and the external image ID value
audio-chunks/chunk45.wav : Confidence scores and the external image id value.
audio-chunks/chunk46.wav : you can then use this ID value to link back to the
source image
audio-chunks/chunk46.wav : You can then use this id value to link back to the
source image.
audio-chunks/chunk47.wav : now that you've learned about the facial detection
features of Amazon recognition here's a summary of the guidelines we've
discussed so far
audio-chunks/chunk47.wav : Now that you've learned about the facial detection
features of amazon recognition here's a summary of the guidelines we've
discussed so far.
audio-chunks/chunk48.wav : when Amazon recognition detects a human face
audio-chunks/chunk48.wav : When amazon recognition detects a human face.
audio-chunks/chunk49.wav : it captures a bounding box that shows where the face
was found in the video
audio-chunks/chunk49.wav : It captures a bounding box that shows where the face
was found in the video.
audio-chunks/chunk50.wav : it can also detect attributes such as the position of
the eyes nose and mouth
audio-chunks/chunk50.wav : It can also detect attributes such as the position of
the eyes nose and mouth.
audio-chunks/chunk51.wav : it can detect emot
audio-chunks/chunk51.wav : It can detect emot.
audio-chunks/chunk52.wav : the quality of the detection
audio-chunks/chunk52.wav : The quality of the detection.
audio-chunks/chunk53.wav : and any landmarks that might appear
audio-chunks/chunk53.wav : And any landmarks that might appear.
audio-chunks/chunk54.wav : all these items will have an Associated confidence
score
audio-chunks/chunk54.wav : All these items will have an associated confidence
score.
audio-chunks/chunk55.wav : a higher score means that the model has greater
confidence about the detection
audio-chunks/chunk55.wav : A higher score means that the model has greater
confidence about the detection.
```

```
audio-chunks/chunk56.wav : gender is inferred from the image
audio-chunks/chunk56.wav : Gender is inferred from the image.
audio-chunks/chunk57.wav : not inferred from Identity
audio-chunks/chunk57.wav : Not inferred from identity.
audio-chunks/chunk58.wav : similarly emotion is also determined from the image
audio-chunks/chunk58.wav : Similarly emotion is also determined from the image.
audio-chunks/chunk59.wav : and it might not reflect the subject's actual
emotional state
audio-chunks/chunk59.wav : And it might not reflect the subject's actual
emotional state.
audio-chunks/chunk60.wav : how should I apply facial recognition responsibly
audio-chunks/chunk60.wav : How should i apply facial recognition responsibly.
audio-chunks/chunk61.wav : facial recognition should never be used in a way that
violates an individual's rights
audio-chunks/chunk61.wav : Facial recognition should never be used in a way that
violates an individual's rights.
audio-chunks/chunk62.wav : including the right to privacy
audio-chunks/chunk62.wav : Including the right to privacy.
audio-chunks/chunk63.wav : it should also never be used to make autonomous
decisions for scenarios that require a human being to analyze them
audio-chunks/chunk63.wav : It should also never be used to make autonomous
decisions for scenarios that require a human being to analyze them.
audio-chunks/chunk64.wav : for example
audio-chunks/chunk64.wav : For example.
audio-chunks/chunk65.wav : suppose that a bank uses tools like Amazon
recognition
audio-chunks/chunk65.wav : Suppose that a bank uses tools like amazon
recognition.
audio-chunks/chunk66.wav : in a financial application to verify their customers
identities
audio-chunks/chunk66.wav : In a financial application to verify their customers
identities.
audio-chunks/chunk67.wav : the bank should always clearly disclosed the use of
the technology
audio-chunks/chunk67.wav : The bank should always clearly disclosed the use of
the technology.
audio-chunks/chunk68.wav : and ask the customer to approve their terms and
conditions
audio-chunks/chunk68.wav : And ask the customer to approve their terms and
conditions.
audio-chunks/chunk69.wav : for more information about this topic
audio-chunks/chunk69.wav : For more information about this topic.
audio-chunks/chunk70.wav : see the AWS web page about the facts on facial
recognition with artificial intelligence
audio-chunks/chunk70.wav : See the aws web page about the facts on facial
recognition with artificial intelligence.
audio-chunks/chunk71.wav : will now explain how you can use Amazon recognition
to process videos
```

```
audio-chunks/chunk71.wav : Will now explain how you can use amazon recognition
to process videos.
audio-chunks/chunk72.wav : you can perform video processing on both stored
videos and video streams
audio-chunks/chunk72.wav : You can perform video processing on both stored
videos and video streams.
audio-chunks/chunk73.wav : stored videos should be uploaded and stored in an S3
bucket
audio-chunks/chunk73.wav : Stored videos should be uploaded and stored in an s3
bucket.
audio-chunks/chunk74.wav : each type of detection has its own start operation
audio-chunks/chunk74.wav : Each type of detection has its own start operation.
audio-chunks/chunk75.wav : you can search for people
audio-chunks/chunk75.wav : You can search for people.
audio-chunks/chunk76.wav : faces labels
audio-chunks/chunk76.wav : Faces labels.
audio-chunks/chunk77.wav : celebrities
audio-chunks/chunk77.wav : Celebrities.
audio-chunks/chunk78.wav : text and inappropriate content
audio-chunks/chunk78.wav : Text and inappropriate content.
audio-chunks/chunk79.wav : Amazon recognition publishes a completion status to a
topic in Amazon simple notification service
audio-chunks/chunk79.wav : Amazon recognition publishes a completion status to a
topic in amazon simple notification service.
audio-chunks/chunk80.wav : which is also known as Amazon SNS
audio-chunks/chunk80.wav : Which is also known as amazon sns.
audio-chunks/chunk81.wav : then S&S can route these messages to subscribers
audio-chunks/chunk81.wav : Then s&s can route these messages to subscribers.
audio-chunks/chunk82.wav : for durability it's a best practice to Route messages
to a message queue in Amazon simple queue service or Amazon sqs
audio-chunks/chunk82.wav : For durability it's a best practice to route messages
to a message queue in amazon simple queue service or amazon sqs.
audio-chunks/chunk83.wav : your application should monitor the sqsq for
completion
audio-chunks/chunk83.wav : Your application should monitor the sqsq for
completion.
audio-chunks/chunk84.wav : each start operation has a corresponding get
operation for retrieving the results
audio-chunks/chunk84.wav : Each start operation has a corresponding get
operation for retrieving the results.
audio-chunks/chunk85.wav : if you call get detection results
audio-chunks/chunk85.wav : If you call get detection results.
audio-chunks/chunk86.wav : it returns an array of labels
audio-chunks/chunk86.wav : It returns an array of labels.
audio-chunks/chunk87.wav : the contain information about any labels found in the
video
audio-chunks/chunk87.wav : The contain information about any labels found in the
video.
```

```
audio-chunks/chunk88.wav : the label information includes the same labels as
image detection
audio-chunks/chunk88.wav : The label information includes the same labels as
image detection.
audio-chunks/chunk89.wav : but it also includes a timestamp of where the label
was detected in milliseconds
audio-chunks/chunk89.wav : But it also includes a timestamp of where the label
was detected in milliseconds.
audio-chunks/chunk90.wav : from the start of the video
audio-chunks/chunk90.wav : From the start of the video.
audio-chunks/chunk91.wav : in addition to stored videos you can also use Amazon
recognition video
audio-chunks/chunk91.wav : In addition to stored videos you can also use amazon
recognition video.
audio-chunks/chunk92.wav : to detect and recognize faces in streaming video
audio-chunks/chunk92.wav : To detect and recognize faces in streaming video.
audio-chunks/chunk93.wav : a typical use case for this is detecting a known face
in a video stream
audio-chunks/chunk93.wav : A typical use case for this is detecting a known face
in a video stream.
audio-chunks/chunk94.wav : Amazon recognition video uses Amazon Kinesis video
streams
audio-chunks/chunk94.wav : Amazon recognition video uses amazon kinesis video
streams.
audio-chunks/chunk95.wav : to receive and process a video stream
audio-chunks/chunk95.wav : To receive and process a video stream.
audio-chunks/chunk96.wav : the analysis results are output from Amazon
recognition video to a kinesis datastream
audio-chunks/chunk96.wav : The analysis results are output from amazon
recognition video to a kinesis datastream.
audio-chunks/chunk97.wav : they are then read by your client application
audio-chunks/chunk97.wav : They are then read by your client application.
audio-chunks/chunk98.wav : Amazon recognition video provides a stream processor
that's called create stream processor
audio-chunks/chunk98.wav : Amazon recognition video provides a stream processor
that's called create stream processor.
audio-chunks/chunk99.wav : and you can use it to start and manage the analysis
of the streaming video
audio-chunks/chunk99.wav : And you can use it to start and manage the analysis
of the streaming video.
audio-chunks/chunk100.wav : to use Amazon recognition video with your streaming
video
audio-chunks/chunk100.wav : To use amazon recognition video with your streaming
video.
audio-chunks/chunk101.wav : your application must Implement these resources
audio-chunks/chunk101.wav : Your application must implement these resources.
Error:
audio-chunks/chunk103.wav : you need a Kinesis video stream to send streaming
```

```
video to Amazon recognition video
audio-chunks/chunk103.wav : You need a kinesis video stream to send streaming
video to amazon recognition video.
Error:
audio-chunks/chunk105.wav : you need an Amazon recognition video stream
processor to manage the streaming video analysis
audio-chunks/chunk105.wav : You need an amazon recognition video stream
processor to manage the streaming video analysis.
audio-chunks/chunk106.wav : and finally you need a Kinesis datastream consumer
audio-chunks/chunk106.wav : And finally you need a kinesis datastream consumer.
audio-chunks/chunk107.wav : to read the analysis results that Amazon recognition
video sends to the datastream
audio-chunks/chunk107.wav : To read the analysis results that amazon recognition
video sends to the datastream.
audio-chunks/chunk108.wav : if you want to find a face in a video you need to
create a collection
audio-chunks/chunk108.wav : If you want to find a face in a video you need to
create a collection.
audio-chunks/chunk109.wav : this process is the same as creating a collection
for still images
audio-chunks/chunk109.wav : This process is the same as creating a collection
for still images.
audio-chunks/chunk110.wav : Amazon recognition video places a Json frame record
for each analyzed frame into the Kinesis output stream
audio-chunks/chunk110.wav : Amazon recognition video places a json frame record
for each analyzed frame into the kinesis output stream.
audio-chunks/chunk111.wav : Amazon recognition video doesn't analyze every frame
that's passed to it through the Kinesis video stream
audio-chunks/chunk111.wav : Amazon recognition video doesn't analyze every frame
that's passed to it through the kinesis video stream.
audio-chunks/chunk112.wav : a frame record that sent to a Kinesis datastream
contains information about which video stream fragment the frame is in
audio-chunks/chunk112.wav : A frame record that sent to a kinesis datastream
contains information about which video stream fragment the frame is in.
audio-chunks/chunk113.wav : where are the frame is in the fragment
audio-chunks/chunk113.wav : Where are the frame is in the fragment.
audio-chunks/chunk114.wav : and faces that are recognized in the frame
audio-chunks/chunk114.wav : And faces that are recognized in the frame.
audio-chunks/chunk115.wav : it also includes status information for the stream
processor
audio-chunks/chunk115.wav : It also includes status information for the stream
processor.
audio-chunks/chunk116.wav : before we wrap
audio-chunks/chunk116.wav : Before we wrap.
audio-chunks/chunk117.wav : here's a quick summary
audio-chunks/chunk117.wav : Here's a quick summary.
audio-chunks/chunk118.wav : Amazon recognition is a computer vision service
that's based on deep learning
```

```
audio-chunks/chunk118.wav : Amazon recognition is a computer vision service
that's based on deep learning.
audio-chunks/chunk119.wav : you can easily add image and video analysis to your
applications
audio-chunks/chunk119.wav : You can easily add image and video analysis to your
applications.
audio-chunks/chunk120.wav : Amazon recognition can detect faces
audio-chunks/chunk120.wav : Amazon recognition can detect faces.
audio-chunks/chunk121.wav : sentiment
audio-chunks/chunk121.wav : Sentiment.
audio-chunks/chunk122.wav : text
audio-chunks/chunk122.wav : Text.
audio-chunks/chunk123.wav : unsafe content and Library search in both images and
video
audio-chunks/chunk123.wav : Unsafe content and library search in both images and
video.
audio-chunks/chunk124.wav : Amazon recognition is integrated with other AWS
services
audio-chunks/chunk124.wav : Amazon recognition is integrated with other aws
services.
audio-chunks/chunk125.wav : thanks for watching we'll see you in the next video
audio-chunks/chunk125.wav : Thanks for watching we'll see you in the next video.
audio-chunks/chunk1.wav : hi and welcome to section 4
audio-chunks/chunk1.wav : Hi and welcome to section 4.
audio-chunks/chunk2.wav : in this section we're going to look at feature
engineering
audio-chunks/chunk2.wav : In this section we're going to look at feature
engineering.
audio-chunks/chunk3.wav : feature engineering is one of the most impactful
things you can do to improve your machine learning model
audio-chunks/chunk3.wav : Feature engineering is one of the most impactful
things you can do to improve your machine learning model.
audio-chunks/chunk4.wav : will now look at what it is
audio-chunks/chunk4.wav : Will now look at what it is.
audio-chunks/chunk5.wav : there are two things that can help make your models
more successful
audio-chunks/chunk5.wav : There are two things that can help make your models
more successful.
audio-chunks/chunk6.wav : the first
audio-chunks/chunk6.wav : The first.
audio-chunks/chunk7.wav : is feature selection
audio-chunks/chunk7.wav : Is feature selection.
audio-chunks/chunk8.wav : and the second is feature extraction
audio-chunks/chunk8.wav : And the second is feature extraction.
audio-chunks/chunk9.wav : or the process of creating features
audio-chunks/chunk9.wav : Or the process of creating features.
audio-chunks/chunk10.wav : in feature selection
audio-chunks/chunk10.wav : In feature selection.
```

```
audio-chunks/chunk11.wav : you select the most relevant features and discard the
rest
audio-chunks/chunk11.wav : You select the most relevant features and discard the
rest.
audio-chunks/chunk12.wav : you can apply feature selection to prevent redundancy
or irrelevance in the existing features
audio-chunks/chunk12.wav : You can apply feature selection to prevent redundancy
or irrelevance in the existing features.
audio-chunks/chunk13.wav : you can also use it to limit the number of features
to help prevent overfitting
audio-chunks/chunk13.wav : You can also use it to limit the number of features
to help prevent overfitting.
audio-chunks/chunk14.wav : feature extraction
audio-chunks/chunk14.wav : Feature extraction.
audio-chunks/chunk15.wav : builds valuable information from raw data by
reformatting
audio-chunks/chunk15.wav : Builds valuable information from raw data by
reformatting.
audio-chunks/chunk16.wav : combining and transforming primary features into new
ones
audio-chunks/chunk16.wav : Combining and transforming primary features into new
ones.
audio-chunks/chunk17.wav : this process continues until it yields a new data set
that can be consumed by the model to achieve your goals
audio-chunks/chunk17.wav : This process continues until it yields a new data set
that can be consumed by the model to achieve your goals.
audio-chunks/chunk18.wav : as the diagram shows
audio-chunks/chunk18.wav : As the diagram shows.
audio-chunks/chunk19.wav : feature extraction covers a range of activities
audio-chunks/chunk19.wav : Feature extraction covers a range of activities.
audio-chunks/chunk20.wav : from dealing with missing data to converting text
Data into numerical data
audio-chunks/chunk20.wav : From dealing with missing data to converting text
data into numerical data.
audio-chunks/chunk21.wav : although the list isn't exhaustive it should give you
some idea of the data handling that's needed to get data into a useful state
audio-chunks/chunk21.wav : Although the list isn't exhaustive it should give you
some idea of the data handling that's needed to get data into a useful state.
audio-chunks/chunk22.wav : many of the tasks are no different than any other job
working with data
audio-chunks/chunk22.wav : Many of the tasks are no different than any other job
working with data.
audio-chunks/chunk23.wav : you'll want to make sure data is in the correct
format
audio-chunks/chunk23.wav : You'll want to make sure data is in the correct
format.
audio-chunks/chunk24.wav : that it's consistently represented
audio-chunks/chunk24.wav : That it's consistently represented.
```

```
audio-chunks/chunk25.wav : correctly spelled among other
audio-chunks/chunk25.wav : Correctly spelled among other.
audio-chunks/chunk26.wav : for example you might combine data or extract data
into multiple columns
audio-chunks/chunk26.wav : For example you might combine data or extract data
into multiple columns.
audio-chunks/chunk27.wav : or you could also remove columns altogether
audio-chunks/chunk27.wav : Or you could also remove columns altogether.
audio-chunks/chunk28.wav : specific to machine learn
audio-chunks/chunk28.wav : Specific to machine learn.
audio-chunks/chunk29.wav : you'll need to convert text columns to numerical
values
audio-chunks/chunk29.wav : You'll need to convert text columns to numerical
values.
audio-chunks/chunk30.wav : you'll also need to decide how to handle outliers
audio-chunks/chunk30.wav : You'll also need to decide how to handle outliers.
audio-chunks/chunk31.wav : and potentially rescale your data
audio-chunks/chunk31.wav : And potentially rescale your data.
audio-chunks/chunk32.wav : next
audio-chunks/chunk32.wav : Next.
audio-chunks/chunk33.wav : we'll look at some of the more common tasks in this
section
audio-chunks/chunk33.wav : We'll look at some of the more common tasks in this
section.
audio-chunks/chunk34.wav : most machine learning algorithms work best with
numerical data
audio-chunks/chunk34.wav : Most machine learning algorithms work best with
numerical data.
audio-chunks/chunk35.wav : you'll need to make sure that all columns in your
data set contain numeric data by converting or encoding it
audio-chunks/chunk35.wav : You'll need to make sure that all columns in your
data set contain numeric data by converting or encoding it.
audio-chunks/chunk36.wav : you might need to make several passes through the
datasheet before you can encode it
audio-chunks/chunk36.wav : You might need to make several passes through the
datasheet before you can encode it.
audio-chunks/chunk37.wav : for example
audio-chunks/chunk37.wav : For example.
audio-chunks/chunk38.wav : you might have variability in the text values
audio-chunks/chunk38.wav : You might have variability in the text values.
audio-chunks/chunk39.wav : such as rose that contain both medium and Med is
values
audio-chunks/chunk39.wav : Such as rose that contain both medium and med is
values.
audio-chunks/chunk40.wav : if the categorical data has ordered to it
audio-chunks/chunk40.wav : If the categorical data has ordered to it.
audio-chunks/chunk41.wav : you want to encode the text into numerical values
that capture this ordinal relationship
```

```
audio-chunks/chunk41.wav : You want to encode the text into numerical values
that capture this ordinal relationship.
audio-chunks/chunk42.wav : say you have data showing maintenance costs
audio-chunks/chunk42.wav : Say you have data showing maintenance costs.
audio-chunks/chunk43.wav : you might encode low to one
audio-chunks/chunk43.wav : You might encode low to one.
audio-chunks/chunk44.wav : medium to
audio-chunks/chunk44.wav : Medium to.
audio-chunks/chunk45.wav : hi to three and very high to four
audio-chunks/chunk45.wav : Hi to three and very high to four.
audio-chunks/chunk46.wav : after you've made sure your categorical data is All
Uniform
audio-chunks/chunk46.wav : After you've made sure your categorical data is all
uniform.
audio-chunks/chunk47.wav : you can use tools like skykit learn and pandas to
encode your data
audio-chunks/chunk47.wav : You can use tools like skykit learn and pandas to
encode your data.
audio-chunks/chunk48.wav : if the categorical data doesn't have any order to it
audio-chunks/chunk48.wav : If the categorical data doesn't have any order to it.
audio-chunks/chunk49.wav : then you'll need to break the data into multiple
columns
audio-chunks/chunk49.wav : Then you'll need to break the data into multiple
columns.
audio-chunks/chunk50.wav : this will help make sure you don't introduce an
ordinal relationship to the data that isn't there
audio-chunks/chunk50.wav : This will help make sure you don't introduce an
ordinal relationship to the data that isn't there.
audio-chunks/chunk51.wav : for example suppose you assigned the value of 1 to
the first color such as red
audio-chunks/chunk51.wav : For example suppose you assigned the value of 1 to
the first color such as red.
audio-chunks/chunk52.wav : and you then assigned to the next value say blue
audio-chunks/chunk52.wav : And you then assigned to the next value say blue.
audio-chunks/chunk53.wav : the model could interpret blue as being more
important than red because blue has a higher numeric value
audio-chunks/chunk53.wav : The model could interpret blue as being more
important than red because blue has a higher numeric value.
audio-chunks/chunk54.wav : encoding non ordinal data into multiple columns or
features is a better way
audio-chunks/chunk54.wav : Encoding non ordinal data into multiple columns or
features is a better way.
audio-chunks/chunk55.wav : think of the new features like a checkbox
audio-chunks/chunk55.wav : Think of the new features like a checkbox.
audio-chunks/chunk56.wav : consider the example
audio-chunks/chunk56.wav : Consider the example.
audio-chunks/chunk57.wav : there are three features that were generated
audio-chunks/chunk57.wav : There are three features that were generated.
```

```
audio-chunks/chunk58.wav : the volume one indicates that the instance has that
feature like its color
audio-chunks/chunk58.wav : The volume one indicates that the instance has that
feature like its color.
audio-chunks/chunk59.wav : that's it for this section we'll see you again in the
next video
audio-chunks/chunk59.wav : That's it for this section we'll see you again in the
next video.
audio-chunks/chunk1.wav : it's now time to review the module
audio-chunks/chunk1.wav : It's now time to review the module.
audio-chunks/chunk2.wav : here are the main takeaways for this module
audio-chunks/chunk2.wav : Here are the main takeaways for this module.
Error:
audio-chunks/chunk4.wav : we looked at defining machine learning and how it fits
into the broader AI landscape
audio-chunks/chunk4.wav : We looked at defining machine learning and how it fits
into the broader ai landscape.
audio-chunks/chunk5.wav : we also looked at the types of problem machine
learning can help us solve
audio-chunks/chunk5.wav : We also looked at the types of problem machine
learning can help us solve.
audio-chunks/chunk6.wav : and how machine learning applies learning algorithms
to develop models from large data sets
audio-chunks/chunk6.wav : And how machine learning applies learning algorithms
to develop models from large data sets.
audio-chunks/chunk7.wav : we then looked at the machine learning Pipeline and
the different stages for developing a machine learning application
audio-chunks/chunk7.wav : We then looked at the machine learning pipeline and
the different stages for developing a machine learning application.
audio-chunks/chunk8.wav : finally we introduce some of the tools and services
you can use
audio-chunks/chunk8.wav : Finally we introduce some of the tools and services
you can use.
audio-chunks/chunk9.wav : before discussing some of the challenges of the
machine learning
audio-chunks/chunk9.wav : Before discussing some of the challenges of the
machine learning.
audio-chunks/chunk10.wav : in summary in this module you learned how to
audio-chunks/chunk10.wav : In summary in this module you learned how to.
audio-chunks/chunk11.wav : recognize how machine learning and deep learning are
part of artificial intelligence
audio-chunks/chunk11.wav : Recognize how machine learning and deep learning are
part of artificial intelligence.
audio-chunks/chunk12.wav : describe artificial intelligence and machine learning
terminology
audio-chunks/chunk12.wav : Describe artificial intelligence and machine learning
terminology.
audio-chunks/chunk13.wav : identify how machine learning can be used to solve a
```

```
business problem
audio-chunks/chunk13.wav : Identify how machine learning can be used to solve a
business problem.
audio-chunks/chunk14.wav : describe the machine learning process
audio-chunks/chunk14.wav : Describe the machine learning process.
audio-chunks/chunk15.wav : list the tools available to data scientists
audio-chunks/chunk15.wav : List the tools available to data scientists.
audio-chunks/chunk16.wav : identify when to use machine learning instead of
traditional software development methods
audio-chunks/chunk16.wav : Identify when to use machine learning instead of
traditional software development methods.
audio-chunks/chunk17.wav : thanks for watching we'll see you in the next video
audio-chunks/chunk17.wav : Thanks for watching we'll see you in the next video.
audio-chunks/chunk1.wav : hi and welcome back to module 3
audio-chunks/chunk1.wav : Hi and welcome back to module 3.
audio-chunks/chunk2.wav : this is Section 8
audio-chunks/chunk2.wav : This is section 8.
audio-chunks/chunk3.wav : in this section we're going to take a look at how you
can tune the models hyperparameters to improve model performance
audio-chunks/chunk3.wav : In this section we're going to take a look at how you
can tune the models hyperparameters to improve model performance.
audio-chunks/chunk4.wav : recall from an earlier module that hyperparameters can
be thought of as the knobs that tune the machine learning algorithm to improve
its performance
audio-chunks/chunk4.wav : Recall from an earlier module that hyperparameters can
be thought of as the knobs that tune the machine learning algorithm to improve
its performance.
audio-chunks/chunk5.wav : now that we're looking more explicitly at tuning
models
audio-chunks/chunk5.wav : Now that we're looking more explicitly at tuning
models.
audio-chunks/chunk6.wav : it's time to look more specifically
audio-chunks/chunk6.wav : It's time to look more specifically.
audio-chunks/chunk7.wav : at the different types of hyperparameters
audio-chunks/chunk7.wav : At the different types of hyperparameters.
audio-chunks/chunk8.wav : and how to perform hyperparameter optimization
audio-chunks/chunk8.wav : And how to perform hyperparameter optimization.
audio-chunks/chunk9.wav : there are a couple of different categories of
hyperparameters
audio-chunks/chunk9.wav : There are a couple of different categories of
hyperparameters.
audio-chunks/chunk10.wav : the first kind are model hyperparameters
audio-chunks/chunk10.wav : The first kind are model hyperparameters.
audio-chunks/chunk11.wav : they helped Define the model itself
audio-chunks/chunk11.wav : They helped define the model itself.
audio-chunks/chunk12.wav : as an example
audio-chunks/chunk12.wav : As an example.
audio-chunks/chunk13.wav : consider a neural network for a computer vision
```

```
problem
audio-chunks/chunk13.wav : Consider a neural network for a computer vision
problem.
audio-chunks/chunk14.wav : for this case
audio-chunks/chunk14.wav : For this case.
audio-chunks/chunk15.wav : additional attributes of the architecture need to be
defined
audio-chunks/chunk15.wav : Additional attributes of the architecture need to be
defined.
audio-chunks/chunk16.wav : like filter size
audio-chunks/chunk16.wav : Like filter size.
audio-chunks/chunk17.wav : Cooling
audio-chunks/chunk17.wav : Cooling.
audio-chunks/chunk18.wav : and the stride or padding
audio-chunks/chunk18.wav : And the stride or padding.
audio-chunks/chunk19.wav : the second kind are optimizer hyperparameters
audio-chunks/chunk19.wav : The second kind are optimizer hyperparameters.
audio-chunks/chunk20.wav : they relate to how the model learns patterns Based on
data
audio-chunks/chunk20.wav : They relate to how the model learns patterns based on
data.
audio-chunks/chunk21.wav : and they're used for a neural network model
audio-chunks/chunk21.wav : And they're used for a neural network model.
audio-chunks/chunk22.wav : these types of hyperparameters include optimizers
like
audio-chunks/chunk22.wav : These types of hyperparameters include optimizers
like.
audio-chunks/chunk23.wav : gradient Desu
audio-chunks/chunk23.wav : Gradient desu.
audio-chunks/chunk24.wav : and stochastic gradient descent
audio-chunks/chunk24.wav : And stochastic gradient descent.
audio-chunks/chunk25.wav : they can also include optimizers that use momentum
like atom
audio-chunks/chunk25.wav : They can also include optimizers that use momentum
like atom.
audio-chunks/chunk26.wav : or that initialize the parameter weights with methods
like Xavier initialization or he initialization
audio-chunks/chunk26.wav : Or that initialize the parameter weights with methods
like xavier initialization or he initialization.
audio-chunks/chunk27.wav : the Third Kind are data hyperparameters
audio-chunks/chunk27.wav : The third kind are data hyperparameters.
audio-chunks/chunk28.wav : they relate to the attributes of the data itself
audio-chunks/chunk28.wav : They relate to the attributes of the data itself.
audio-chunks/chunk29.wav : these include attributes that Define different data
augmentation techniques
audio-chunks/chunk29.wav : These include attributes that define different data
augmentation techniques.
audio-chunks/chunk30.wav : like cropping or resizing for image related problems
```

```
audio-chunks/chunk30.wav : Like cropping or resizing for image related problems.
audio-chunks/chunk31.wav : they're often used when you don't have enough data or
enough variation in your data
audio-chunks/chunk31.wav : They're often used when you don't have enough data or
enough variation in your data.
audio-chunks/chunk32.wav : tuning hyperparameters can be very labor-intensive
audio-chunks/chunk32.wav : Tuning hyperparameters can be very labor-intensive.
audio-chunks/chunk33.wav : traditionally this was done manually by someone who
had domain experience related to the hyperparameter and the use case
audio-chunks/chunk33.wav : Traditionally this was done manually by someone who
had domain experience related to the hyperparameter and the use case.
audio-chunks/chunk34.wav : this person would manually select the hyperparameters
audio-chunks/chunk34.wav : This person would manually select the
hyperparameters.
audio-chunks/chunk35.wav : based on their intuition and experience
audio-chunks/chunk35.wav : Based on their intuition and experience.
audio-chunks/chunk36.wav : then they would train the model and score it on the
validation data
audio-chunks/chunk36.wav : Then they would train the model and score it on the
validation data.
audio-chunks/chunk37.wav : this process would be repeated over and over again
audio-chunks/chunk37.wav : This process would be repeated over and over again.
audio-chunks/chunk38.wav : until they achieved satisfactory results
audio-chunks/chunk38.wav : Until they achieved satisfactory results.
audio-chunks/chunk39.wav : this manual process isn't always the most thorough
and efficient way of tuning hyperparameters
audio-chunks/chunk39.wav : This manual process isn't always the most thorough
and efficient way of tuning hyperparameters.
audio-chunks/chunk40.wav : with sagemaker you can perform automated
hyperparameter tuning
audio-chunks/chunk40.wav : With sagemaker you can perform automated
hyperparameter tuning.
audio-chunks/chunk41.wav : with Amazon sagemaker automatic model tuning
audio-chunks/chunk41.wav : With amazon sagemaker automatic model tuning.
audio-chunks/chunk42.wav : it finds the best version of a model by running
multiple training jobs on your data set
audio-chunks/chunk42.wav : It finds the best version of a model by running
multiple training jobs on your data set.
audio-chunks/chunk43.wav : by using the algorithm and hyperparameter ranges you
specify
audio-chunks/chunk43.wav : By using the algorithm and hyperparameter ranges you
specify.
audio-chunks/chunk44.wav : it then chooses the hyperparameter values that
results in a model that performs the best
audio-chunks/chunk44.wav : It then chooses the hyperparameter values that
results in a model that performs the best.
audio-chunks/chunk45.wav : as measured by a metric you choose
audio-chunks/chunk45.wav : As measured by a metric you choose.
```

```
audio-chunks/chunk46.wav : it uses gaussian process regression to predict which
hyperparameter values
audio-chunks/chunk46.wav : It uses gaussian process regression to predict which
hyperparameter values.
audio-chunks/chunk47.wav : might be most effective at improving fit
audio-chunks/chunk47.wav : Might be most effective at improving fit.
audio-chunks/chunk48.wav : it also uses Bayesian optimization to balance
exploring the hyperparameter space and exploiting specific hyperparameter values
when appropriate
audio-chunks/chunk48.wav : It also uses bayesian optimization to balance
exploring the hyperparameter space and exploiting specific hyperparameter values
when appropriate.
audio-chunks/chunk49.wav : an importantly
audio-chunks/chunk49.wav : An importantly.
audio-chunks/chunk50.wav : automatic model tuning can be used with built-in
algorithms from sagemaker
audio-chunks/chunk50.wav : Automatic model tuning can be used with built-in
algorithms from sagemaker.
audio-chunks/chunk51.wav : pre-built deep learning Frameworks
audio-chunks/chunk51.wav : Pre-built deep learning frameworks.
audio-chunks/chunk52.wav : and bring your own algorithm containers
audio-chunks/chunk52.wav : And bring your own algorithm containers.
audio-chunks/chunk53.wav : suppose that you want to solve a binary
classification problem on a fraud data set
audio-chunks/chunk53.wav : Suppose that you want to solve a binary
classification problem on a fraud data set.
audio-chunks/chunk54.wav : your goal is to maximize the area under the AUC Curve
Metric of the algorithm by training a linear learner algorithm model
audio-chunks/chunk54.wav : Your goal is to maximize the area under the auc curve
metric of the algorithm by training a linear learner algorithm model.
audio-chunks/chunk55.wav : you don't know which values of the learning rate
audio-chunks/chunk55.wav : You don't know which values of the learning rate.
audio-chunks/chunk56.wav : beta one
audio-chunks/chunk56.wav : Beta one.
audio-chunks/chunk57.wav : beta 2 and epoxy should use to train the best model
audio-chunks/chunk57.wav : Beta 2 and epoxy should use to train the best model.
audio-chunks/chunk58.wav : to find the best values for these hyperparameters
audio-chunks/chunk58.wav : To find the best values for these hyperparameters.
audio-chunks/chunk59.wav : you can specify ranges of values that sagemaker
hyperparameter tuning will then search
audio-chunks/chunk59.wav : You can specify ranges of values that sagemaker
hyperparameter tuning will then search.
audio-chunks/chunk60.wav : it will find the combination of values that results
in the training job that performs the best
audio-chunks/chunk60.wav : It will find the combination of values that results
in the training job that performs the best.
audio-chunks/chunk61.wav : as measure by the objective metric that you chose
audio-chunks/chunk61.wav : As measure by the objective metric that you chose.
```

```
audio-chunks/chunk62.wav : in the example
audio-chunks/chunk62.wav : In the example.
audio-chunks/chunk63.wav : sagemaker hyperparameter tuning launches training
jobs that use hyperparameter values in the ranges you specified
audio-chunks/chunk63.wav : Sagemaker hyperparameter tuning launches training
jobs that use hyperparameter values in the ranges you specified.
audio-chunks/chunk64.wav : and then return to the training job with the highest
AUC
audio-chunks/chunk64.wav : And then return to the training job with the highest
auc.
audio-chunks/chunk65.wav : hyperparameter tuning might not necessarily improve
your model
audio-chunks/chunk65.wav : Hyperparameter tuning might not necessarily improve
your model.
audio-chunks/chunk66.wav : it's an advanced tool for building machine Solutions
audio-chunks/chunk66.wav : It's an advanced tool for building machine solutions.
audio-chunks/chunk67.wav : as
audio-chunks/chunk67.wav : As.
audio-chunks/chunk68.wav : it should be considered part of the scientific method
process
audio-chunks/chunk68.wav : It should be considered part of the scientific method
process.
audio-chunks/chunk69.wav : when you build complex machine learning systems like
deep learning neural networks
audio-chunks/chunk69.wav : When you build complex machine learning systems like
deep learning neural networks.
audio-chunks/chunk70.wav : exploring all possible combinations is impractical
audio-chunks/chunk70.wav : Exploring all possible combinations is impractical.
audio-chunks/chunk71.wav : to improve optimization
audio-chunks/chunk71.wav : To improve optimization.
audio-chunks/chunk72.wav : use the following guidelines when you create
hyperparameters
audio-chunks/chunk72.wav : Use the following guidelines when you create
hyperparameters.
Error:
audio-chunks/chunk74.wav : instead of using all hyperparameters
audio-chunks/chunk74.wav : Instead of using all hyperparameters.
audio-chunks/chunk75.wav : limit the number of hyperparameters to the ones you
think would give you good results
audio-chunks/chunk75.wav : Limit the number of hyperparameters to the ones you
think would give you good results.
audio-chunks/chunk76.wav : the range of values for the hyperparameters you
choose to search
audio-chunks/chunk76.wav : The range of values for the hyperparameters you
choose to search.
audio-chunks/chunk77.wav : can significantly affect the success of
hyperparameter optimization
audio-chunks/chunk77.wav : Can significantly affect the success of
```

```
hyperparameter optimization.
audio-chunks/chunk78.wav : although you might want to specify a large range that
covers every possible value for a hyperparameter
audio-chunks/chunk78.wav : Although you might want to specify a large range that
covers every possible value for a hyperparameter.
audio-chunks/chunk79.wav : you'll get better results by limiting your search to
a small range of values
audio-chunks/chunk79.wav : You'll get better results by limiting your search to
a small range of values.
audio-chunks/chunk80.wav : if you get the best metric values within a part of a
range
audio-chunks/chunk80.wav : If you get the best metric values within a part of a
range.
audio-chunks/chunk81.wav : consider limiting the range to only that part
audio-chunks/chunk81.wav : Consider limiting the range to only that part.
audio-chunks/chunk82.wav : during hyperparameter tuning
audio-chunks/chunk82.wav : During hyperparameter tuning.
audio-chunks/chunk83.wav : sagemaker attempts to figure out if your
hyperparameters are log scaled or linear scaled
audio-chunks/chunk83.wav : Sagemaker attempts to figure out if your
hyperparameters are log scaled or linear scaled.
audio-chunks/chunk84.wav : initially it assumes the hyperparameters are linear
scaled
audio-chunks/chunk84.wav : Initially it assumes the hyperparameters are linear
scaled.
audio-chunks/chunk85.wav : if they should be log scaled it might take time for
sagemaker to discover that on its own
audio-chunks/chunk85.wav : If they should be log scaled it might take time for
sagemaker to discover that on its own.
audio-chunks/chunk86.wav : if you know that a hyperparameter should be long
scaled and you can convert it yourself
audio-chunks/chunk86.wav : If you know that a hyperparameter should be long
scaled and you can convert it yourself.
audio-chunks/chunk87.wav : doing so can improve hyperparameter optimization
audio-chunks/chunk87.wav : Doing so can improve hyperparameter optimization.
audio-chunks/chunk88.wav : running more hyperparameter tuning jobs concurrently
gets more work done quickly
audio-chunks/chunk88.wav : Running more hyperparameter tuning jobs concurrently
gets more work done quickly.
audio-chunks/chunk89.wav : but a tuning job improves only through successive
rounds of experiments
audio-chunks/chunk89.wav : But a tuning job improves only through successive
rounds of experiments.
audio-chunks/chunk90.wav : typically running one training job at a time achieves
the best results with the least amount of compute time
audio-chunks/chunk90.wav : Typically running one training job at a time achieves
the best results with the least amount of compute time.
audio-chunks/chunk91.wav : say that you have a distributed training job that
```

```
runs on multiple instances
audio-chunks/chunk91.wav : Say that you have a distributed training job that
runs on multiple instances.
audio-chunks/chunk92.wav : in this case
audio-chunks/chunk92.wav : In this case.
audio-chunks/chunk93.wav : hyperparameter tuning uses the last reported
objective metric from all instances of that training job as the value of the
objective metric for that training job
audio-chunks/chunk93.wav : Hyperparameter tuning uses the last reported
objective metric from all instances of that training job as the value of the
objective metric for that training job.
audio-chunks/chunk94.wav : design distributed training jobs
audio-chunks/chunk94.wav : Design distributed training jobs.
audio-chunks/chunk95.wav : so that they report the objective metric you want
audio-chunks/chunk95.wav : So that they report the objective metric you want.
audio-chunks/chunk96.wav : now that you've gone through the end-to-end process
of training and tuning a machine learning model
audio-chunks/chunk96.wav : Now that you've gone through the end-to-end process
of training and tuning a machine learning model.
audio-chunks/chunk97.wav : it's worth talking about Amazon sagemaker autopilot
audio-chunks/chunk97.wav : It's worth talking about amazon sagemaker autopilot.
audio-chunks/chunk98.wav : this service can help you find a good model with
little effort or input on your part
audio-chunks/chunk98.wav : This service can help you find a good model with
little effort or input on your part.
audio-chunks/chunk99.wav : with autopilot you create a job that supplies the
test
audio-chunks/chunk99.wav : With autopilot you create a job that supplies the
test.
audio-chunks/chunk100.wav : training and Target
audio-chunks/chunk100.wav : Training and target.
audio-chunks/chunk101.wav : autopilot will analyze the data
audio-chunks/chunk101.wav : Autopilot will analyze the data.
audio-chunks/chunk102.wav : select appropriate features
audio-chunks/chunk102.wav : Select appropriate features.
audio-chunks/chunk103.wav : and then train and tune the models
audio-chunks/chunk103.wav : And then train and tune the models.
audio-chunks/chunk104.wav : it will document the metric
audio-chunks/chunk104.wav : It will document the metric.
audio-chunks/chunk105.wav : and find the best model
audio-chunks/chunk105.wav : And find the best model.
audio-chunks/chunk106.wav : based on the provided data
audio-chunks/chunk106.wav : Based on the provided data.
audio-chunks/chunk107.wav : the results include the winning model and Metric
audio-chunks/chunk107.wav : The results include the winning model and metric.
audio-chunks/chunk108.wav : and a jupyter notebook you can use to investigate
the results
audio-chunks/chunk108.wav : And a jupyter notebook you can use to investigate
```

```
the results.
audio-chunks/chunk109.wav : although using autopilot doesn't remove your need to
pre-process the data
audio-chunks/chunk109.wav : Although using autopilot doesn't remove your need to
pre-process the data.
audio-chunks/chunk110.wav : it can save you time during feature selection and
model tuning
audio-chunks/chunk110.wav : It can save you time during feature selection and
model tuning.
audio-chunks/chunk111.wav : some key takeaways from the section of the module
include these points
audio-chunks/chunk111.wav : Some key takeaways from the section of the module
include these points.
Error:
audio-chunks/chunk113.wav : model tuning is important for finding the best
solution to your business problem
audio-chunks/chunk113.wav : Model tuning is important for finding the best
solution to your business problem.
audio-chunks/chunk114.wav : hyperparameters can be tuned for the model Optimizer
and data
audio-chunks/chunk114.wav : Hyperparameters can be tuned for the model optimizer
and data.
audio-chunks/chunk115.wav : sagemaker can perform automatic hyperparameter
tuning
audio-chunks/chunk115.wav : Sagemaker can perform automatic hyperparameter
tuning.
audio-chunks/chunk116.wav : and finally
audio-chunks/chunk116.wav : And finally.
audio-chunks/chunk117.wav : overall model development can be accelerated by
using autopilot
audio-chunks/chunk117.wav : Overall model development can be accelerated by
using autopilot.
audio-chunks/chunk118.wav : that's it for this video see you in the next one
audio-chunks/chunk118.wav : That's it for this video see you in the next one.
Error:
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring how to describe your data
audio-chunks/chunk2.wav : Will continue exploring how to describe your data.
audio-chunks/chunk3.wav : now that your data is in a readable format you can
perform descriptive statistics on the data to better understand it
audio-chunks/chunk3.wav : Now that your data is in a readable format you can
perform descriptive statistics on the data to better understand it.
audio-chunks/chunk4.wav : descriptive statistics help you gain valuable insights
into your data so that you can effectively pre-process the data and prepare it
for your ml model
audio-chunks/chunk4.wav : Descriptive statistics help you gain valuable insights
into your data so that you can effectively pre-process the data and prepare it
```

```
for your ml model.
audio-chunks/chunk5.wav : we'll look at how you can do that and discuss why it's
so important
audio-chunks/chunk5.wav : We'll look at how you can do that and discuss why it's
so important.
Error:
audio-chunks/chunk7.wav : descriptive statistics can be organized into a few
different categories
audio-chunks/chunk7.wav : Descriptive statistics can be organized into a few
different categories.
audio-chunks/chunk8.wav : overall statistics include the number of rows and the
number of columns in your data set
audio-chunks/chunk8.wav : Overall statistics include the number of rows and the
number of columns in your data set.
audio-chunks/chunk9.wav : this information which relates to the dimensions of
your data is very important
audio-chunks/chunk9.wav : This information which relates to the dimensions of
your data is very important.
audio-chunks/chunk10.wav : for example it can indicate that you have too many
features
audio-chunks/chunk10.wav : For example it can indicate that you have too many
features.
audio-chunks/chunk11.wav : which can lead to high dimensionality and poor model
performance
audio-chunks/chunk11.wav : Which can lead to high dimensionality and poor model
performance.
audio-chunks/chunk12.wav : attribute statistics are another type of descriptive
statist
audio-chunks/chunk12.wav : Attribute statistics are another type of descriptive
statist.
audio-chunks/chunk13.wav : specifically for numeric attributes
audio-chunks/chunk13.wav : Specifically for numeric attributes.
audio-chunks/chunk14.wav : they're used to get a better sense of the shape of
your attributes
audio-chunks/chunk14.wav : They're used to get a better sense of the shape of
your attributes.
audio-chunks/chunk15.wav : this includes properties like the mean
audio-chunks/chunk15.wav : This includes properties like the mean.
audio-chunks/chunk16.wav : standard deviation VAR
audio-chunks/chunk16.wav : Standard deviation var.
audio-chunks/chunk17.wav : and minimum and maximum values
audio-chunks/chunk17.wav : And minimum and maximum values.
audio-chunks/chunk18.wav : if you need to look at relationships between more
than one variable
audio-chunks/chunk18.wav : If you need to look at relationships between more
than one variable.
audio-chunks/chunk19.wav : you can consider multivariate statistics
audio-chunks/chunk19.wav : You can consider multivariate statistics.
```

```
audio-chunks/chunk20.wav : they mostly relate to the correlations and
relationships between your attributes
audio-chunks/chunk20.wav : They mostly relate to the correlations and
relationships between your attributes.
audio-chunks/chunk21.wav : for cases when you have multiple variables or
features
audio-chunks/chunk21.wav : For cases when you have multiple variables or
features.
audio-chunks/chunk22.wav : you might want to look at the correlations between
them
audio-chunks/chunk22.wav : You might want to look at the correlations between
them.
audio-chunks/chunk23.wav : it's important to identify correlations between
attributes
audio-chunks/chunk23.wav : It's important to identify correlations between
attributes.
audio-chunks/chunk24.wav : because a high correlation between two attributes
audio-chunks/chunk24.wav : Because a high correlation between two attributes.
audio-chunks/chunk25.wav : can sometimes lead to poor model performance
audio-chunks/chunk25.wav : Can sometimes lead to poor model performance.
audio-chunks/chunk26.wav : when features are closely correlated and they're all
used in the same model to predict the response variable there could be problems
audio-chunks/chunk26.wav : When features are closely correlated and they're all
used in the same model to predict the response variable there could be problems.
audio-chunks/chunk27.wav : for example the model loss might not converge to a
minimum state
audio-chunks/chunk27.wav : For example the model loss might not converge to a
minimum state.
audio-chunks/chunk28.wav : so be aware of highly correlated features in your
data set
audio-chunks/chunk28.wav : So be aware of highly correlated features in your
data set.
audio-chunks/chunk29.wav : mean and median are two different measures describing
the extent that your data is clustered around some value or position
audio-chunks/chunk29.wav : Mean and median are two different measures describing
the extent that your data is clustered around some value or position.
audio-chunks/chunk30.wav : mean can be a useful method for understanding your
data when the data is symmetrical
audio-chunks/chunk30.wav : Mean can be a useful method for understanding your
data when the data is symmetrical.
audio-chunks/chunk31.wav : however if your data is skewed or contains outliers
audio-chunks/chunk31.wav : However if your data is skewed or contains outliers.
audio-chunks/chunk32.wav : the median tends to provide the better metric for
understanding your data as it relates to central tendency
audio-chunks/chunk32.wav : The median tends to provide the better metric for
understanding your data as it relates to central tendency.
audio-chunks/chunk33.wav : for instance if you have outliers with large values
audio-chunks/chunk33.wav : For instance if you have outliers with large values.
```

```
audio-chunks/chunk34.wav : the mean can be skewed one way
audio-chunks/chunk34.wav : The mean can be skewed one way.
audio-chunks/chunk35.wav : and it wouldn't serve as an accurate representation
of where your values are truly centered
audio-chunks/chunk35.wav : And it wouldn't serve as an accurate representation
of where your values are truly centered.
audio-chunks/chunk36.wav : median isn't affected by outliers in the same way
audio-chunks/chunk36.wav : Median isn't affected by outliers in the same way.
audio-chunks/chunk37.wav : we'll talk more about outliers soon
audio-chunks/chunk37.wav : We'll talk more about outliers soon.
audio-chunks/chunk38.wav : statistics are available and they can be viewed on
numerical data by using methods such as described
audio-chunks/chunk38.wav : Statistics are available and they can be viewed on
numerical data by using methods such as described.
audio-chunks/chunk39.wav : there are also other methods to calculate the mean
median and others
audio-chunks/chunk39.wav : There are also other methods to calculate the mean
median and others.
audio-chunks/chunk40.wav : you can also view statistics on single or multiple
columns
audio-chunks/chunk40.wav : You can also view statistics on single or multiple
columns.
audio-chunks/chunk41.wav : you can even group data by specific values
audio-chunks/chunk41.wav : You can even group data by specific values.
audio-chunks/chunk42.wav : for categorical attributes
audio-chunks/chunk42.wav : For categorical attributes.
audio-chunks/chunk43.wav : you can look at the frequency of attribute values in
your data set
audio-chunks/chunk43.wav : You can look at the frequency of attribute values in
your data set.
audio-chunks/chunk44.wav : that information will give you some idea about what
is inside that categorical variable
audio-chunks/chunk44.wav : That information will give you some idea about what
is inside that categorical variable.
audio-chunks/chunk45.wav : the diagram here shows the car data set
audio-chunks/chunk45.wav : The diagram here shows the car data set.
audio-chunks/chunk46.wav : which is made up of several categorical values
audio-chunks/chunk46.wav : Which is made up of several categorical values.
Error:
Error:
audio-chunks/chunk49.wav : lug Bo
audio-chunks/chunk49.wav : Lug bo.
audio-chunks/chunk50.wav : safety and class
audio-chunks/chunk50.wav : Safety and class.
audio-chunks/chunk51.wav : safety can be either low
audio-chunks/chunk51.wav : Safety can be either low.
audio-chunks/chunk52.wav : medium or high
audio-chunks/chunk52.wav : Medium or high.
```

```
audio-chunks/chunk53.wav : from the describe function
audio-chunks/chunk53.wav : From the describe function.
audio-chunks/chunk54.wav : you can see that there are three unique values with
low being the most frequent
audio-chunks/chunk54.wav : You can see that there are three unique values with
low being the most frequent.
audio-chunks/chunk55.wav : looking at the class column
audio-chunks/chunk55.wav : Looking at the class column.
audio-chunks/chunk56.wav : it appears that the top value of the four is unacc
audio-chunks/chunk56.wav : It appears that the top value of the four is unacc.
audio-chunks/chunk57.wav : which stands for unaccounted
audio-chunks/chunk57.wav : Which stands for unaccounted.
audio-chunks/chunk58.wav : this accounts for 1,210 of the 1,728 values or 70%
audio-chunks/chunk58.wav : This accounts for 1,210 of the 1,728 values or 70%.
audio-chunks/chunk59.wav : this might suggest an imbalance
audio-chunks/chunk59.wav : This might suggest an imbalance.
audio-chunks/chunk60.wav : for a Target variable that's also of a categorical
type
audio-chunks/chunk60.wav : For a target variable that's also of a categorical
type.
audio-chunks/chunk61.wav : you can look at the class distribution to see whether
there's a class imbalance in your data set
audio-chunks/chunk61.wav : You can look at the class distribution to see whether
there's a class imbalance in your data set.
audio-chunks/chunk62.wav : imbalance data can Mark a disproportionate ratio for
your classes
audio-chunks/chunk62.wav : Imbalance data can mark a disproportionate ratio for
your classes.
audio-chunks/chunk63.wav : for instance
audio-chunks/chunk63.wav : For instance.
audio-chunks/chunk64.wav : your data set is made up of credit card transactions
audio-chunks/chunk64.wav : Your data set is made up of credit card transactions.
audio-chunks/chunk65.wav : but only a tenth of a percent is labeled as fraud
audio-chunks/chunk65.wav : But only a tenth of a percent is labeled as fraud.
audio-chunks/chunk66.wav : in this case
audio-chunks/chunk66.wav : In this case.
audio-chunks/chunk67.wav : your algorithm might not learn well enough to predict
examples of credit card fraud
audio-chunks/chunk67.wav : Your algorithm might not learn well enough to predict
examples of credit card fraud.
audio-chunks/chunk68.wav : visualization could help you gain insights into your
data that you might not be aware of otherwise
audio-chunks/chunk68.wav : Visualization could help you gain insights into your
data that you might not be aware of otherwise.
audio-chunks/chunk69.wav : a histogram is often a good visualization technique
for seeing the overall behavior of a particular feature
audio-chunks/chunk69.wav : A histogram is often a good visualization technique
for seeing the overall behavior of a particular feature.
```

```
audio-chunks/chunk70.wav : with a histogram you can answer questions like
audio-chunks/chunk70.wav : With a histogram you can answer questions like.
audio-chunks/chunk71.wav : is the feature data normally distributed
audio-chunks/chunk71.wav : Is the feature data normally distributed.
audio-chunks/chunk72.wav : how many Peaks are there in the data
audio-chunks/chunk72.wav : How many peaks are there in the data.
audio-chunks/chunk73.wav : is there any skewness for that particular feature
audio-chunks/chunk73.wav : Is there any skewness for that particular feature.
audio-chunks/chunk74.wav : when using histograms for your data visualization
values are Bend
audio-chunks/chunk74.wav : When using histograms for your data visualization
values are bend.
audio-chunks/chunk75.wav : the taller peaks of the histogram indicate the most
common values
audio-chunks/chunk75.wav : The taller peaks of the histogram indicate the most
common values.
audio-chunks/chunk76.wav : for numerical features you can use density plots and
box plots
audio-chunks/chunk76.wav : For numerical features you can use density plots and
box plots.
audio-chunks/chunk77.wav : in addition to histogram
audio-chunks/chunk77.wav : In addition to histogram.
audio-chunks/chunk78.wav : to get an idea of what's inside that particular
feature
audio-chunks/chunk78.wav : To get an idea of what's inside that particular
feature.
audio-chunks/chunk79.wav : like a histogram these visualizations will help you
answer questions like
audio-chunks/chunk79.wav : Like a histogram these visualizations will help you
answer questions like.
audio-chunks/chunk80.wav : what's the range of the data
audio-chunks/chunk80.wav : What's the range of the data.
audio-chunks/chunk81.wav : the peak of the data
audio-chunks/chunk81.wav : The peak of the data.
Error:
audio-chunks/chunk83.wav : are there any special features
audio-chunks/chunk83.wav : Are there any special features.
audio-chunks/chunk84.wav : answering these questions
audio-chunks/chunk84.wav : Answering these questions.
Error:
audio-chunks/chunk86.wav : a box plot is a method for graphically depicting
groups of numerical data through their quartiles
audio-chunks/chunk86.wav : A box plot is a method for graphically depicting
groups of numerical data through their quartiles.
audio-chunks/chunk87.wav : when you have more than two numerical variables in a
feature data set
audio-chunks/chunk87.wav : When you have more than two numerical variables in a
feature data set.
```

```
audio-chunks/chunk88.wav : you might want to look at their relationship
audio-chunks/chunk88.wav : You might want to look at their relationship.
audio-chunks/chunk89.wav : a scatter plot is a good way to identify any special
relationships among those variables
audio-chunks/chunk89.wav : A scatter plot is a good way to identify any special
relationships among those variables.
audio-chunks/chunk90.wav : in this case
audio-chunks/chunk90.wav : In this case.
audio-chunks/chunk91.wav : the left diagram has sulfates and alcohol
audio-chunks/chunk91.wav : The left diagram has sulfates and alcohol.
audio-chunks/chunk92.wav : there are two numerical variables
audio-chunks/chunk92.wav : There are two numerical variables.
audio-chunks/chunk93.wav : suppose you want to show the relationship between
these variables
audio-chunks/chunk93.wav : Suppose you want to show the relationship between
these variables.
audio-chunks/chunk94.wav : you can use a scatter plot to help you visualize that
audio-chunks/chunk94.wav : You can use a scatter plot to help you visualize
that.
audio-chunks/chunk95.wav : there are plot scattered around and the correlation
among them might not be that high because the data is scattered
audio-chunks/chunk95.wav : There are plot scattered around and the correlation
among them might not be that high because the data is scattered.
audio-chunks/chunk96.wav : however you might find some relatively positive
relationships between the two variables
audio-chunks/chunk96.wav : However you might find some relatively positive
relationships between the two variables.
audio-chunks/chunk97.wav : scatter plot matrices help you look at the
relationship between multiple different features
audio-chunks/chunk97.wav : Scatter plot matrices help you look at the
relationship between multiple different features.
audio-chunks/chunk98.wav : in pandas you can easily create scatter plot matrices
based on the columns you want to look at
audio-chunks/chunk98.wav : In pandas you can easily create scatter plot matrices
based on the columns you want to look at.
audio-chunks/chunk99.wav : this example has three columns
audio-chunks/chunk99.wav : This example has three columns.
audio-chunks/chunk100.wav : and it will give the pairwise scatter plot for any
two columns
audio-chunks/chunk100.wav : And it will give the pairwise scatter plot for any
two columns.
audio-chunks/chunk101.wav : with a scatter plot you might want to identify
special regions that a particular subset of data can fit into
audio-chunks/chunk101.wav : With a scatter plot you might want to identify
special regions that a particular subset of data can fit into.
audio-chunks/chunk102.wav : in the example is there a relationship between
alcohol sulfates and quality
audio-chunks/chunk102.wav : In the example is there a relationship between
```

```
alcohol sulfates and quality.
audio-chunks/chunk103.wav : you can plot those values against good and poor
quality wines like the example
audio-chunks/chunk103.wav : You can plot those values against good and poor
quality wines like the example.
audio-chunks/chunk104.wav : plotting gives you an idea of how useful particular
variables can be if you're using them for a classification problem
audio-chunks/chunk104.wav : Plotting gives you an idea of how useful particular
variables can be if you're using them for a classification problem.
audio-chunks/chunk105.wav : that's it for part two of this section
audio-chunks/chunk105.wav : That's it for part two of this section.
audio-chunks/chunk106.wav : we'll see you again for part 3 where will review
correlations and the takeaways for this section
audio-chunks/chunk106.wav : We'll see you again for part 3 where will review
correlations and the takeaways for this section.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring video analysis by reviewing
how to evaluate and improve your model
audio-chunks/chunk2.wav : Will continue exploring video analysis by reviewing
how to evaluate and improve your model.
audio-chunks/chunk3.wav : in general you can improve the quality of your model
with larger quantities of better quality data
audio-chunks/chunk3.wav : In general you can improve the quality of your model
with larger quantities of better quality data.
audio-chunks/chunk4.wav : use training images that clearly show the object or
seen
audio-chunks/chunk4.wav : Use training images that clearly show the object or
seen.
audio-chunks/chunk5.wav : and don't include many things that you're not
interested in
audio-chunks/chunk5.wav : And don't include many things that you're not
interested in.
audio-chunks/chunk6.wav : for bounding boxes around objects
audio-chunks/chunk6.wav : For bounding boxes around objects.
audio-chunks/chunk7.wav : use training images that show the object as fully
visible and not hidden by other objects
audio-chunks/chunk7.wav : Use training images that show the object as fully
visible and not hidden by other objects.
audio-chunks/chunk8.wav : make sure that your training and test data set
audio-chunks/chunk8.wav : Make sure that your training and test data set.
audio-chunks/chunk9.wav : match the type of images that you'll eventually run in
France on
audio-chunks/chunk9.wav : Match the type of images that you'll eventually run in
france on.
audio-chunks/chunk10.wav : for objects where you have just a few training
examples like logos
audio-chunks/chunk10.wav : For objects where you have just a few training
```

```
examples like logos.
audio-chunks/chunk11.wav : you should provide bounding boxes around the logo on
your test images
audio-chunks/chunk11.wav : You should provide bounding boxes around the logo on
your test images.
audio-chunks/chunk12.wav : these images represent the scenarios you want to
localize the object in
audio-chunks/chunk12.wav : These images represent the scenarios you want to
localize the object in.
audio-chunks/chunk13.wav : reducing false positives often results in better
position
audio-chunks/chunk13.wav : Reducing false positives often results in better
position.
audio-chunks/chunk14.wav : to reduce false positives
audio-chunks/chunk14.wav : To reduce false positives.
Error:
audio-chunks/chunk16.wav : check if increasing the confidence threshold enables
you to keep the correct predictions while eliminating false positives
audio-chunks/chunk16.wav : Check if increasing the confidence threshold enables
you to keep the correct predictions while eliminating false positives.
audio-chunks/chunk17.wav : increase the confidence threshold eventually results
in diminishing gains
audio-chunks/chunk17.wav : Increase the confidence threshold eventually results
in diminishing gains.
audio-chunks/chunk18.wav : because of the trade-off between precision and recall
for a given model
audio-chunks/chunk18.wav : Because of the trade-off between precision and recall
for a given model.
Error:
audio-chunks/chunk20.wav : check to see if you need to add additional classes
for training
audio-chunks/chunk20.wav : Check to see if you need to add additional classes
for training.
audio-chunks/chunk21.wav : for example if you are detecting cat
audio-chunks/chunk21.wav : For example if you are detecting cat.
audio-chunks/chunk22.wav : but often dogs are being flagged as cats
audio-chunks/chunk22.wav : But often dogs are being flagged as cats.
audio-chunks/chunk23.wav : add dog as a label to your training data set along
with the images of dogs that you got the false positive on
audio-chunks/chunk23.wav : Add dog as a label to your training data set along
with the images of dogs that you got the false positive on.
audio-chunks/chunk24.wav : effectively you're helping the model learn to predict
dog and not cat
audio-chunks/chunk24.wav : Effectively you're helping the model learn to predict
dog and not cat.
audio-chunks/chunk25.wav : do the new training images
audio-chunks/chunk25.wav : Do the new training images.
audio-chunks/chunk26.wav : you might find that the model is confused between two
```

```
of your custom labels
audio-chunks/chunk26.wav : You might find that the model is confused between two
of your custom labels.
audio-chunks/chunk27.wav : cat and dog
audio-chunks/chunk27.wav : Cat and dog.
audio-chunks/chunk28.wav : the test image with label cat is predicted as having
labeled dog and vice versa
audio-chunks/chunk28.wav : The test image with label cat is predicted as having
labeled dog and vice versa.
audio-chunks/chunk29.wav : in this case
audio-chunks/chunk29.wav : In this case.
audio-chunks/chunk30.wav : first check for mislabeled images in your training
and test sets
audio-chunks/chunk30.wav : First check for mislabeled images in your training
and test sets.
audio-chunks/chunk31.wav : also
audio-chunks/chunk31.wav : Also.
audio-chunks/chunk32.wav : adding more training images that reflect this
confusion will help a retrain model learn to better discriminate between cat and
dog
audio-chunks/chunk32.wav : Adding more training images that reflect this
confusion will help a retrain model learn to better discriminate between cat and
dog.
audio-chunks/chunk33.wav : reducing false negatives
audio-chunks/chunk33.wav : Reducing false negatives.
audio-chunks/chunk34.wav : often results in better recall
audio-chunks/chunk34.wav : Often results in better recall.
audio-chunks/chunk35.wav : to reduce false negatives
audio-chunks/chunk35.wav : To reduce false negatives.
audio-chunks/chunk36.wav : first lower the confidence threshold
audio-chunks/chunk36.wav : First lower the confidence threshold.
audio-chunks/chunk37.wav : this should improve recall
audio-chunks/chunk37.wav : This should improve recall.
audio-chunks/chunk38.wav : also use better examples to model the variety of both
the object and the images they appear in
audio-chunks/chunk38.wav : Also use better examples to model the variety of both
the object and the images they appear in.
audio-chunks/chunk39.wav : finally
audio-chunks/chunk39.wav : Finally.
audio-chunks/chunk40.wav : split your label into two classes that are easier to
learn
audio-chunks/chunk40.wav : Split your label into two classes that are easier to
learn.
audio-chunks/chunk41.wav : for example instead of good cookies and bad cookies
audio-chunks/chunk41.wav : For example instead of good cookies and bad cookies.
audio-chunks/chunk42.wav : you might want good cookies burnt cookies and broken
audio-chunks/chunk42.wav : You might want good cookies burnt cookies and broken.
audio-chunks/chunk43.wav : to help the model learn each unique concept better
```

```
audio-chunks/chunk43.wav : To help the model learn each unique concept better.
audio-chunks/chunk44.wav : if you're satisfied with the performance of your
model
audio-chunks/chunk44.wav : If you're satisfied with the performance of your
model.
audio-chunks/chunk45.wav : you can make it available for use by starting it from
the console or by using code
audio-chunks/chunk45.wav : You can make it available for use by starting it from
the console or by using code.
audio-chunks/chunk46.wav : after the model is running
audio-chunks/chunk46.wav : After the model is running.
audio-chunks/chunk47.wav : you can perform an inference with the AWS CLI or the
SDK
audio-chunks/chunk47.wav : You can perform an inference with the aws cli or the
sdk.
audio-chunks/chunk48.wav : when you call the API you specify
audio-chunks/chunk48.wav : When you call the api you specify.
audio-chunks/chunk49.wav : the Amazon resource name of the Amazon recognition
custom labels model that you want to use
audio-chunks/chunk49.wav : The amazon resource name of the amazon recognition
custom labels model that you want to use.
audio-chunks/chunk50.wav : the Amazon resource name is also known as an Arn
audio-chunks/chunk50.wav : The amazon resource name is also known as an arn.
audio-chunks/chunk51.wav : you'll also specify the image you want the model to
make a prediction with
audio-chunks/chunk51.wav : You'll also specify the image you want the model to
make a prediction with.
audio-chunks/chunk52.wav : you can provide an input image as an image byte array
of Base 64 encoded image bites
audio-chunks/chunk52.wav : You can provide an input image as an image byte array
of base 64 encoded image bites.
audio-chunks/chunk53.wav : what is an S3 object
audio-chunks/chunk53.wav : What is an s3 object.
audio-chunks/chunk54.wav : custom labels are returned in an array of custom
label objects
audio-chunks/chunk54.wav : Custom labels are returned in an array of custom
label objects.
audio-chunks/chunk55.wav : each custom label represents a single object seen or
concept that's found in the image
audio-chunks/chunk55.wav : Each custom label represents a single object seen or
concept that's found in the image.
audio-chunks/chunk56.wav : a custom label includes
audio-chunks/chunk56.wav : A custom label includes.
audio-chunks/chunk57.wav : a label for the object
audio-chunks/chunk57.wav : A label for the object.
audio-chunks/chunk58.wav : seen or concept that was found in the image
audio-chunks/chunk58.wav : Seen or concept that was found in the image.
audio-chunks/chunk59.wav : it also includes a bounding box for objects that were
```

```
found in the image
audio-chunks/chunk59.wav : It also includes a bounding box for objects that were
found in the image.
audio-chunks/chunk60.wav : the bounding box coordinates show where the object is
located on the source image
audio-chunks/chunk60.wav : The bounding box coordinates show where the object is
located on the source image.
audio-chunks/chunk61.wav : the coordinate values are a ratio of the overall
image size
audio-chunks/chunk61.wav : The coordinate values are a ratio of the overall
image size.
audio-chunks/chunk62.wav : finally the custom label includes the confidence
score
audio-chunks/chunk62.wav : Finally the custom label includes the confidence
score.
audio-chunks/chunk63.wav : this represents how confident Amazon recognition
custom labels is in the accuracy of the label and bounding box
audio-chunks/chunk63.wav : This represents how confident amazon recognition
custom labels is in the accuracy of the label and bounding box.
audio-chunks/chunk64.wav : during training
audio-chunks/chunk64.wav : During training.
audio-chunks/chunk65.wav : a model calculates a threshold value that determines
if a prediction for a label is true
audio-chunks/chunk65.wav : A model calculates a threshold value that determines
if a prediction for a label is true.
audio-chunks/chunk66.wav : by default the detect custom labels operation doesn't
return labels with a confidence value that's less than the models calculated
threshold value
audio-chunks/chunk66.wav : By default the detect custom labels operation doesn't
return labels with a confidence value that's less than the models calculated
threshold value.
audio-chunks/chunk67.wav : to filter the return labels
audio-chunks/chunk67.wav : To filter the return labels.
audio-chunks/chunk68.wav : specify a value for men confidence
audio-chunks/chunk68.wav : Specify a value for men confidence.
audio-chunks/chunk69.wav : that's greater than the models calculated threshold
audio-chunks/chunk69.wav : That's greater than the models calculated threshold.
audio-chunks/chunk70.wav : you can get the models calculated threshold from the
models training results in the Amazon recognition custom labels console
audio-chunks/chunk70.wav : You can get the models calculated threshold from the
models training results in the amazon recognition custom labels console.
audio-chunks/chunk71.wav : to get all the labels regardless of confidence
audio-chunks/chunk71.wav : To get all the labels regardless of confidence.
audio-chunks/chunk72.wav : specify A Min confidence value of 0
audio-chunks/chunk72.wav : Specify a min confidence value of 0.
audio-chunks/chunk73.wav : if you find that the confidence values returned by
the detect custom labels operation are too low
audio-chunks/chunk73.wav : If you find that the confidence values returned by
```

```
the detect custom labels operation are too low.
audio-chunks/chunk74.wav : consider retraining the model
audio-chunks/chunk74.wav : Consider retraining the model.
audio-chunks/chunk75.wav : you can restrict the number of custom labels that are
returned from the detect custom labels operation
audio-chunks/chunk75.wav : You can restrict the number of custom labels that are
returned from the detect custom labels operation.
audio-chunks/chunk76.wav : by specifying the Max results input parameter
audio-chunks/chunk76.wav : By specifying the max results input parameter.
audio-chunks/chunk77.wav : the return results are sorted from the highest
confidence to the lowest confidence
audio-chunks/chunk77.wav : The return results are sorted from the highest
confidence to the lowest confidence.
audio-chunks/chunk78.wav : here are some key takeaways from this section of the
module
audio-chunks/chunk78.wav : Here are some key takeaways from this section of the
module.
audio-chunks/chunk79.wav : models must be trained for the specific domain that
you want to analyze
audio-chunks/chunk79.wav : Models must be trained for the specific domain that
you want to analyze.
audio-chunks/chunk80.wav : if you're looking for turbochargers
audio-chunks/chunk80.wav : If you're looking for turbochargers.
audio-chunks/chunk81.wav : you'll need many pictures of turbochargers to train
your model
audio-chunks/chunk81.wav : You'll need many pictures of turbochargers to train
your model.
audio-chunks/chunk82.wav : you can set custom labeling for the specific business
case
audio-chunks/chunk82.wav : You can set custom labeling for the specific business
case.
audio-chunks/chunk83.wav : we looked at the custom labeling process and some of
the tools you can use
audio-chunks/chunk83.wav : We looked at the custom labeling process and some of
the tools you can use.
audio-chunks/chunk84.wav : if you want objects to be detected
audio-chunks/chunk84.wav : If you want objects to be detected.
audio-chunks/chunk85.wav : you need to label images and create bounding boxes
for these objects
audio-chunks/chunk85.wav : You need to label images and create bounding boxes
for these objects.
audio-chunks/chunk86.wav : you can use Amazon sagemaker ground truth
audio-chunks/chunk86.wav : You can use amazon sagemaker ground truth.
audio-chunks/chunk87.wav : to build training data sets for your model
audio-chunks/chunk87.wav : To build training data sets for your model.
audio-chunks/chunk88.wav : which can also use machine learning to label your
images
audio-chunks/chunk88.wav : Which can also use machine learning to label your
```

```
images.
audio-chunks/chunk89.wav : thanks for watching and we'll see you in the next
video
audio-chunks/chunk89.wav : Thanks for watching and we'll see you in the next
video.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring how to evaluate your model
audio-chunks/chunk2.wav : Will continue exploring how to evaluate your model.
audio-chunks/chunk3.wav : classification models are going to return a
probability for the Target
audio-chunks/chunk3.wav : Classification models are going to return a
probability for the target.
audio-chunks/chunk4.wav : this is a value of the input belonging to the target
class
audio-chunks/chunk4.wav : This is a value of the input belonging to the target
class.
audio-chunks/chunk5.wav : and it will be between 0 and 1
audio-chunks/chunk5.wav : And it will be between 0 and 1.
audio-chunks/chunk6.wav : to convert the value to a class you need to determine
the threshold to use
audio-chunks/chunk6.wav : To convert the value to a class you need to determine
the threshold to use.
audio-chunks/chunk7.wav : you might think it's 50%
audio-chunks/chunk7.wav : You might think it's 50%.
audio-chunks/chunk8.wav : but you could change it to be lower or higher to
improve your results
audio-chunks/chunk8.wav : But you could change it to be lower or higher to
improve your results.
audio-chunks/chunk9.wav : as you've seen with sensitivity and specificity
there's a trade-off between correctly and incorrectly identifying classes
audio-chunks/chunk9.wav : As you've seen with sensitivity and specificity
there's a trade-off between correctly and incorrectly identifying classes.
audio-chunks/chunk10.wav : changing the threshold can impact that outcome
audio-chunks/chunk10.wav : Changing the threshold can impact that outcome.
audio-chunks/chunk11.wav : we're going to take a look at how you can visualize
this
audio-chunks/chunk11.wav : We're going to take a look at how you can visualize
this.
audio-chunks/chunk12.wav : a receiver operating characteristic graph is also
known as an Roc graph
audio-chunks/chunk12.wav : A receiver operating characteristic graph is also
known as an roc graph.
audio-chunks/chunk13.wav : it's summarizes all the confusion matrices that each
threshold produced
audio-chunks/chunk13.wav : It's summarizes all the confusion matrices that each
threshold produced.
audio-chunks/chunk14.wav : to build one
```

```
audio-chunks/chunk14.wav : To build one.
audio-chunks/chunk15.wav : you calculate and plot the sensitivity or true
positive rate against the false positive rate
audio-chunks/chunk15.wav : You calculate and plot the sensitivity or true
positive rate against the false positive rate.
audio-chunks/chunk16.wav : on a graph for each threshold value
audio-chunks/chunk16.wav : On a graph for each threshold value.
audio-chunks/chunk17.wav : you can calculate the false positive rate by
subtracting the specificity from one
audio-chunks/chunk17.wav : You can calculate the false positive rate by
subtracting the specificity from one.
audio-chunks/chunk18.wav : after you plot those points you can draw a line
between them
audio-chunks/chunk18.wav : After you plot those points you can draw a line
between them.
audio-chunks/chunk19.wav : the dotted black line from 00 to 11 mean that the
sensitivity or true positive rate is equal to the false positive rate
audio-chunks/chunk19.wav : The dotted black line from 00 to 11 mean that the
sensitivity or true positive rate is equal to the false positive rate.
audio-chunks/chunk20.wav : the point at 11 means that you've correctly
identified all the cats
audio-chunks/chunk20.wav : The point at 11 means that you've correctly
identified all the cats.
audio-chunks/chunk21.wav : but you've also incorrectly identified all the not
cat
audio-chunks/chunk21.wav : But you've also incorrectly identified all the not
cat.
audio-chunks/chunk22.wav : this is bad
audio-chunks/chunk22.wav : This is bad.
audio-chunks/chunk23.wav : any point on this line means that the proportion of
correctly classified samples is the same as the proportion of incorrectly
classified samples
audio-chunks/chunk23.wav : Any point on this line means that the proportion of
correctly classified samples is the same as the proportion of incorrectly
classified samples.
audio-chunks/chunk24.wav : the point at 00
audio-chunks/chunk24.wav : The point at 00.
audio-chunks/chunk25.wav : true positives and 0 fives
audio-chunks/chunk25.wav : True positives and 0 fives.
audio-chunks/chunk26.wav : a model that has high sensitivity and low false
positive rate is usually the goal
audio-chunks/chunk26.wav : A model that has high sensitivity and low false
positive rate is usually the goal.
audio-chunks/chunk27.wav : so it's considered to be better when the line between
the threshold recordings is closer towards the top left corner
audio-chunks/chunk27.wav : So it's considered to be better when the line between
the threshold recordings is closer towards the top left corner.
audio-chunks/chunk28.wav : if you had the data from two models
```

```
audio-chunks/chunk28.wav : If you had the data from two models.
audio-chunks/chunk29.wav : you could plot out the ROC curve for each model and
compare them
audio-chunks/chunk29.wav : You could plot out the roc curve for each model and
compare them.
audio-chunks/chunk30.wav : however that can be tedious
audio-chunks/chunk30.wav : However that can be tedious.
audio-chunks/chunk31.wav : there's another graph you can use for this which will
look at next
audio-chunks/chunk31.wav : There's another graph you can use for this which will
look at next.
audio-chunks/chunk32.wav : another evaluation metric you can use is the area
under the curve receiver operator Cur
audio-chunks/chunk32.wav : Another evaluation metric you can use is the area
under the curve receiver operator cur.
audio-chunks/chunk33.wav : which is also known as an AUC Roc
audio-chunks/chunk33.wav : Which is also known as an auc roc.
audio-chunks/chunk34.wav : the AUC part is the area under the plotted line
audio-chunks/chunk34.wav : The auc part is the area under the plotted line.
audio-chunks/chunk35.wav : when the AUC is higher it means the model will be
better at predicting cats as cats and not cats as not cats
audio-chunks/chunk35.wav : When the auc is higher it means the model will be
better at predicting cats as cats and not cats as not cats.
audio-chunks/chunk36.wav : you can use the AUC to quickly compare models with
each other
audio-chunks/chunk36.wav : You can use the auc to quickly compare models with
each other.
audio-chunks/chunk37.wav : with the four numbers from our confusion Matrix
audio-chunks/chunk37.wav : With the four numbers from our confusion matrix.
audio-chunks/chunk38.wav : you can calculate the models accuracy
audio-chunks/chunk38.wav : You can calculate the models accuracy.
audio-chunks/chunk39.wav : this is also known as its score
audio-chunks/chunk39.wav : This is also known as its score.
audio-chunks/chunk40.wav : you can do this by adding up the correct predictions
audio-chunks/chunk40.wav : You can do this by adding up the correct predictions.
audio-chunks/chunk41.wav : and then dividing that number by the total number of
predictions
audio-chunks/chunk41.wav : And then dividing that number by the total number of
predictions.
audio-chunks/chunk42.wav : the accuracy is widely used metric for classification
problems it has limitations
audio-chunks/chunk42.wav : The accuracy is widely used metric for classification
problems it has limitations.
audio-chunks/chunk43.wav : this metric isn't as effective when there are a lot
of true negative cases in your data set
audio-chunks/chunk43.wav : This metric isn't as effective when there are a lot
of true negative cases in your data set.
audio-chunks/chunk44.wav : think about the cat not cat example
```

```
audio-chunks/chunk44.wav : Think about the cat not cat example.
audio-chunks/chunk45.wav : if most of your accuracy is based on true negatives
audio-chunks/chunk45.wav : If most of your accuracy is based on true negatives.
audio-chunks/chunk46.wav : it says that your model is good at predicting what
isn't a cat
audio-chunks/chunk46.wav : It says that your model is good at predicting what
isn't a cat.
audio-chunks/chunk47.wav : in this case
audio-chunks/chunk47.wav : In this case.
audio-chunks/chunk48.wav : you might not feel confident in your model's ability
to predict cats after you roll it out into production
audio-chunks/chunk48.wav : You might not feel confident in your model's ability
to predict cats after you roll it out into production.
audio-chunks/chunk49.wav : this leads to an example of why it's important to
make sure that the metric you choose for model evaluation aligns to your
business goal
audio-chunks/chunk49.wav : This leads to an example of why it's important to
make sure that the metric you choose for model evaluation aligns to your
business goal.
audio-chunks/chunk50.wav : think about the credit card fraud example
audio-chunks/chunk50.wav : Think about the credit card fraud example.
audio-chunks/chunk51.wav : in this case using accuracy is your main metric
probably isn't a good idea
audio-chunks/chunk51.wav : In this case using accuracy is your main metric
probably isn't a good idea.
audio-chunks/chunk52.wav : because you have a lot of true negatives
audio-chunks/chunk52.wav : Because you have a lot of true negatives.
audio-chunks/chunk53.wav : your high true negative number might hide the fact
that your models ability to identify cases of fraud that is to identify true
positives isn't ideal
audio-chunks/chunk53.wav : Your high true negative number might hide the fact
that your models ability to identify cases of fraud that is to identify true
positives isn't ideal.
audio-chunks/chunk54.wav : as a credit card company it's probably unacceptable
to have less than almost perfect performance identifying fraud cases
audio-chunks/chunk54.wav : As a credit card company it's probably unacceptable
to have less than almost perfect performance identifying fraud cases.
audio-chunks/chunk55.wav : that would drive customers away which would be the
opposite of what you'd want to achieve from a business standpoint
audio-chunks/chunk55.wav : That would drive customers away which would be the
opposite of what you'd want to achieve from a business standpoint.
audio-chunks/chunk56.wav : this is why two other metrics are often used in these
situations
audio-chunks/chunk56.wav : This is why two other metrics are often used in these
situations.
audio-chunks/chunk57.wav : the first one is precision
audio-chunks/chunk57.wav : The first one is precision.
audio-chunks/chunk58.wav : which essentially removes the negative predictions
```

```
audio-chunks/chunk58.wav : Which essentially removes the negative predictions.
audio-chunks/chunk59.wav : Precision is the proportion of positive predictions
that are actually correct
audio-chunks/chunk59.wav : Precision is the proportion of positive predictions
that are actually correct.
audio-chunks/chunk60.wav : you can calculate it by taking the true positive
audio-chunks/chunk60.wav : You can calculate it by taking the true positive.
audio-chunks/chunk61.wav : and dividing it by true positive plus false positive
audio-chunks/chunk61.wav : And dividing it by true positive plus false positive.
audio-chunks/chunk62.wav : when the cost of false positives is high in your
particular business situation
audio-chunks/chunk62.wav : When the cost of false positives is high in your
particular business situation.
audio-chunks/chunk63.wav : Precision might be a good metric
audio-chunks/chunk63.wav : Precision might be a good metric.
audio-chunks/chunk64.wav : think about a classification model that identifies
email messages as spam or not
audio-chunks/chunk64.wav : Think about a classification model that identifies
email messages as spam or not.
audio-chunks/chunk65.wav : in this case you don't want your model to label an
email messages spam and thus prevent your users from seeing that message when
it's actually legitimate
audio-chunks/chunk65.wav : In this case you don't want your model to label an
email messages spam and thus prevent your users from seeing that message when
it's actually legitimate.
audio-chunks/chunk66.wav : or consider an example of a model that needs to
predict whether a patient has a terminal illness
audio-chunks/chunk66.wav : Or consider an example of a model that needs to
predict whether a patient has a terminal illness.
audio-chunks/chunk67.wav : in this case
audio-chunks/chunk67.wav : In this case.
audio-chunks/chunk68.wav : using precision as your evaluation metric doesn't
account for false negatives in your model
audio-chunks/chunk68.wav : Using precision as your evaluation metric doesn't
account for false negatives in your model.
audio-chunks/chunk69.wav : here for the models to be successful
audio-chunks/chunk69.wav : Here for the models to be successful.
audio-chunks/chunk70.wav : it's crucial that it doesn't fall asleep predict the
absence of illness in a patient who actually has that illness
audio-chunks/chunk70.wav : It's crucial that it doesn't fall asleep predict the
absence of illness in a patient who actually has that illness.
audio-chunks/chunk71.wav : sensitivity would be a better metric to use for this
situation
audio-chunks/chunk71.wav : Sensitivity would be a better metric to use for this
situation.
audio-chunks/chunk72.wav : but it doesn't always need to be one or the other
audio-chunks/chunk72.wav : But it doesn't always need to be one or the other.
audio-chunks/chunk73.wav : the F1 score combines precision and sensitivity
```

```
together
audio-chunks/chunk73.wav : The f1 score combines precision and sensitivity
together.
audio-chunks/chunk74.wav : it gives you one number the quantifies the overall
performance of a particular ml algorithm
audio-chunks/chunk74.wav : It gives you one number the quantifies the overall
performance of a particular ml algorithm.
audio-chunks/chunk75.wav : you should consider using an F1 score when you have a
class imbalance but want to preserve the equality between precision and
sensitivity
audio-chunks/chunk75.wav : You should consider using an f1 score when you have a
class imbalance but want to preserve the equality between precision and
sensitivity.
audio-chunks/chunk76.wav : what do you do if you're dealing with a regression
problem
audio-chunks/chunk76.wav : What do you do if you're dealing with a regression
problem.
audio-chunks/chunk77.wav : in that case there are other common metrics you can
use to evaluate your model including the mean squared error
audio-chunks/chunk77.wav : In that case there are other common metrics you can
use to evaluate your model including the mean squared error.
audio-chunks/chunk78.wav : the mean squared error is frequently used
audio-chunks/chunk78.wav : The mean squared error is frequently used.
audio-chunks/chunk79.wav : its general purpose is the same as what you saw with
classification metrics
audio-chunks/chunk79.wav : Its general purpose is the same as what you saw with
classification metrics.
audio-chunks/chunk80.wav : you determine the prediction from the model
audio-chunks/chunk80.wav : You determine the prediction from the model.
audio-chunks/chunk81.wav : and you compare the difference between the prediction
audio-chunks/chunk81.wav : And you compare the difference between the
prediction.
audio-chunks/chunk82.wav : in the actual outcome
audio-chunks/chunk82.wav : In the actual outcome.
audio-chunks/chunk83.wav : more specifically
audio-chunks/chunk83.wav : More specifically.
audio-chunks/chunk84.wav : you take the difference between the prediction and
actual value
audio-chunks/chunk84.wav : You take the difference between the prediction and
actual value.
audio-chunks/chunk85.wav : square that difference
audio-chunks/chunk85.wav : Square that difference.
audio-chunks/chunk86.wav : and then sum up all the square differences for all
the observations
audio-chunks/chunk86.wav : And then sum up all the square differences for all
the observations.
audio-chunks/chunk87.wav : in skykit learn you can use the mean squared error
function directly from the metrics Library
```

```
audio-chunks/chunk87.wav : In skykit learn you can use the mean squared error
function directly from the metrics library.
audio-chunks/chunk88.wav : there are other metrics you can use for linear models
such as r-squared
audio-chunks/chunk88.wav : There are other metrics you can use for linear models
such as r-squared.
audio-chunks/chunk89.wav : so you've trained your model
audio-chunks/chunk89.wav : So you've trained your model.
audio-chunks/chunk90.wav : performed a batch transformation on your test data
and calculated your metric
audio-chunks/chunk90.wav : Performed a batch transformation on your test data
and calculated your metric.
audio-chunks/chunk91.wav : now what will you do
audio-chunks/chunk91.wav : Now what will you do.
audio-chunks/chunk92.wav : you'll use these metrics to help you tune the model
audio-chunks/chunk92.wav : You'll use these metrics to help you tune the model.
audio-chunks/chunk93.wav : you could select a different set of features and
train the model again
audio-chunks/chunk93.wav : You could select a different set of features and
train the model again.
audio-chunks/chunk94.wav : after you retrain the model ask yourself
audio-chunks/chunk94.wav : After you retrain the model ask yourself.
audio-chunks/chunk95.wav : which was the better model
audio-chunks/chunk95.wav : Which was the better model.
audio-chunks/chunk96.wav : the metrics will help inform you
audio-chunks/chunk96.wav : The metrics will help inform you.
audio-chunks/chunk97.wav : you could also use different data and retrain the
model with the same features
audio-chunks/chunk97.wav : You could also use different data and retrain the
model with the same features.
audio-chunks/chunk98.wav : remember k-fold cross-validation from earlier in this
module
audio-chunks/chunk98.wav : Remember k-fold cross-validation from earlier in this
module.
audio-chunks/chunk99.wav : finally
audio-chunks/chunk99.wav : Finally.
audio-chunks/chunk100.wav : you could tune the parameters of the model itself
audio-chunks/chunk100.wav : You could tune the parameters of the model itself.
audio-chunks/chunk101.wav : which is the subject of the next section
audio-chunks/chunk101.wav : Which is the subject of the next section.
audio-chunks/chunk102.wav : here are key takeaways from this section of the
module
audio-chunks/chunk102.wav : Here are key takeaways from this section of the
module.
audio-chunks/chunk103.wav : to evaluate the model you need to have data that the
model hasn't seen
audio-chunks/chunk103.wav : To evaluate the model you need to have data that the
model hasn't seen.
```

```
audio-chunks/chunk104.wav : this could be either a holdout set
audio-chunks/chunk104.wav : This could be either a holdout set.
audio-chunks/chunk105.wav : or you could use k-fold cross-validation
audio-chunks/chunk105.wav : Or you could use k-fold cross-validation.
audio-chunks/chunk106.wav : different machine learning models use different
metrics
audio-chunks/chunk106.wav : Different machine learning models use different
metrics.
audio-chunks/chunk107.wav : classification can use the confusion Matrix and the
AUC Roc that you can generate from it
audio-chunks/chunk107.wav : Classification can use the confusion matrix and the
auc roc that you can generate from it.
audio-chunks/chunk108.wav : regression can use mean squared
audio-chunks/chunk108.wav : Regression can use mean squared.
audio-chunks/chunk109.wav : that's it for Section 7
audio-chunks/chunk109.wav : That's it for section 7.
audio-chunks/chunk110.wav : see you in the next video
audio-chunks/chunk110.wav : See you in the next video.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring data collection by reviewing
how to extract
audio-chunks/chunk2.wav : Will continue exploring data collection by reviewing
how to extract.
audio-chunks/chunk3.wav : transform and load data
audio-chunks/chunk3.wav : Transform and load data.
audio-chunks/chunk4.wav : data is typically spread across many different systems
and data providers
audio-chunks/chunk4.wav : Data is typically spread across many different systems
and data providers.
audio-chunks/chunk5.wav : this presents a challenge
audio-chunks/chunk5.wav : This presents a challenge.
audio-chunks/chunk6.wav : you'll need to bring all these data sources together
into something that can be consumed by a machine learning model
audio-chunks/chunk6.wav : You'll need to bring all these data sources together
into something that can be consumed by a machine learning model.
audio-chunks/chunk7.wav : you can do this through extract
audio-chunks/chunk7.wav : You can do this through extract.
audio-chunks/chunk8.wav : transform and load which is also known as ETL
audio-chunks/chunk8.wav : Transform and load which is also known as etl.
audio-chunks/chunk9.wav : the steps in ETL are defined this way
audio-chunks/chunk9.wav : The steps in etl are defined this way.
audio-chunks/chunk10.wav : in the extract
audio-chunks/chunk10.wav : In the extract.
audio-chunks/chunk11.wav : you pull the data from the sources to a single
location
audio-chunks/chunk11.wav : You pull the data from the sources to a single
location.
```

```
audio-chunks/chunk12.wav : during extraction you might need to modify the data
combine matching records or do other tasks that transform the data
audio-chunks/chunk12.wav : During extraction you might need to modify the data
combine matching records or do other tasks that transform the data.
audio-chunks/chunk13.wav : finally in the load
audio-chunks/chunk13.wav : Finally in the load.
audio-chunks/chunk14.wav : the data is loaded into a repository such as Amazon
S3
audio-chunks/chunk14.wav : The data is loaded into a repository such as amazon
s3.
audio-chunks/chunk15.wav : a typical ETL framework has several components
audio-chunks/chunk15.wav : A typical etl framework has several components.
audio-chunks/chunk16.wav : as an example consider the diagram
audio-chunks/chunk16.wav : As an example consider the diagram.
audio-chunks/chunk17.wav : first the crawler a program connects to a datastore
which can be a source or a Target
audio-chunks/chunk17.wav : First the crawler a program connects to a datastore
which can be a source or a target.
audio-chunks/chunk18.wav : it progresses through a ranked list of classifiers to
determine the schema for your data
audio-chunks/chunk18.wav : It progresses through a ranked list of classifiers to
determine the schema for your data.
audio-chunks/chunk19.wav : then it creates metadata tables in the AWS glue data
catalog
audio-chunks/chunk19.wav : Then it creates metadata tables in the aws glue data
catalog.
audio-chunks/chunk20.wav : a job defines the business logic that's needed to
perform ETL work
audio-chunks/chunk20.wav : A job defines the business logic that's needed to
perform etl work.
audio-chunks/chunk21.wav : to run the job you'll need to use a schedule or event
audio-chunks/chunk21.wav : To run the job you'll need to use a schedule or
event.
audio-chunks/chunk22.wav : as a final note the services we just discussed exist
in the transformed partition of the ETL process
audio-chunks/chunk22.wav : As a final note the services we just discussed exist
in the transformed partition of the etl process.
audio-chunks/chunk23.wav : AWS glue is a fully managed ETL service that makes it
simple and cost-effective to categorize your data
audio-chunks/chunk23.wav : Aws glue is a fully managed etl service that makes it
simple and cost-effective to categorize your data.
audio-chunks/chunk24.wav : clean
audio-chunks/chunk24.wav : Clean.
audio-chunks/chunk25.wav : enrich
audio-chunks/chunk25.wav : Enrich.
audio-chunks/chunk26.wav : and move it reliably between various data stores
audio-chunks/chunk26.wav : And move it reliably between various data stores.
audio-chunks/chunk27.wav : AWS glue consists of a central metadata repository
```

```
known as the AWS glue data catalog
audio-chunks/chunk27.wav : Aws glue consists of a central metadata repository
known as the aws glue data catalog.
audio-chunks/chunk28.wav : this is an ETL engine that automatically generates
python or Scala code
audio-chunks/chunk28.wav : This is an etl engine that automatically generates
python or scala code.
audio-chunks/chunk29.wav : it also provides a flexible scheduler that handles
dependency resolution
audio-chunks/chunk29.wav : It also provides a flexible scheduler that handles
dependency resolution.
audio-chunks/chunk30.wav : job monitoring and retries
audio-chunks/chunk30.wav : Job monitoring and retries.
audio-chunks/chunk31.wav : AWS glue is server
audio-chunks/chunk31.wav : Aws glue is server.
audio-chunks/chunk32.wav : so you don't need to set up or manage any
infrastructure
audio-chunks/chunk32.wav : So you don't need to set up or manage any
infrastructure.
audio-chunks/chunk33.wav : you can use the AWS glue console to discover data
audio-chunks/chunk33.wav : You can use the aws glue console to discover data.
audio-chunks/chunk34.wav : transform
audio-chunks/chunk34.wav : Transform.
audio-chunks/chunk35.wav : and make it available for search inquiries
audio-chunks/chunk35.wav : And make it available for search inquiries.
audio-chunks/chunk36.wav : the console calls the underlying services to
orchestrate the work needed to transform your data
audio-chunks/chunk36.wav : The console calls the underlying services to
orchestrate the work needed to transform your data.
audio-chunks/chunk37.wav : you can also use the AWS glue API operations to
interface with the AWS glue services
audio-chunks/chunk37.wav : You can also use the aws glue api operations to
interface with the aws glue services.
audio-chunks/chunk38.wav : this way you can edit
audio-chunks/chunk38.wav : This way you can edit.
audio-chunks/chunk39.wav : debug and test your python or Scala Apache spark ETL
code using a familiar development environment
audio-chunks/chunk39.wav : Debug and test your python or scala apache spark etl
code using a familiar development environment.
audio-chunks/chunk40.wav : AWS glue is well-suited to machine learning because
it can receive label data that can be used for training
audio-chunks/chunk40.wav : Aws glue is well-suited to machine learning because
it can receive label data that can be used for training.
audio-chunks/chunk41.wav : here's an example
audio-chunks/chunk41.wav : Here's an example.
audio-chunks/chunk42.wav : say that you provide AWS glue with training data that
teaches the model what duplicate records in the data source look like
audio-chunks/chunk42.wav : Say that you provide aws glue with training data that
```

```
teaches the model what duplicate records in the data source look like.
audio-chunks/chunk43.wav : then AWS glue can identify the duplicates and present
them for further analysis by a data engineer
audio-chunks/chunk43.wav : Then aws glue can identify the duplicates and present
them for further analysis by a data engineer.
audio-chunks/chunk44.wav : AWS glue enables the orchestration of complex ETL
jobs
audio-chunks/chunk44.wav : Aws glue enables the orchestration of complex etl
jobs.
audio-chunks/chunk45.wav : in the example AWS glue crawls the data sources and
presents the information to clients as a data catalog
audio-chunks/chunk45.wav : In the example aws glue crawls the data sources and
presents the information to clients as a data catalog.
audio-chunks/chunk46.wav : AWS glue can run your ETL jobs based on an event such
as getting a new data set
audio-chunks/chunk46.wav : Aws glue can run your etl jobs based on an event such
as getting a new data set.
audio-chunks/chunk47.wav : for example you can use an AWS Lambda function to
trigger your ETL jobs to run as soon as new data becomes available in Amazon S3
audio-chunks/chunk47.wav : For example you can use an aws lambda function to
trigger your etl jobs to run as soon as new data becomes available in amazon s3.
audio-chunks/chunk48.wav : you can also register this new data set in the AWS
glue data catalog as part of your ETL jobs
audio-chunks/chunk48.wav : You can also register this new data set in the aws
glue data catalog as part of your etl jobs.
audio-chunks/chunk49.wav : although manage tools are available in AWS to
manipulate data a data scientist will also write scripts in their jupyter
notebook to handle it data
audio-chunks/chunk49.wav : Although manage tools are available in aws to
manipulate data a data scientist will also write scripts in their jupyter
notebook to handle it data.
audio-chunks/chunk50.wav : a very simple extract and load script is shown here
audio-chunks/chunk50.wav : A very simple extract and load script is shown here.
audio-chunks/chunk51.wav : the Imports and variable section Imports the library
that are used
audio-chunks/chunk51.wav : The imports and variable section imports the library
that are used.
audio-chunks/chunk52.wav : note that Moto 3 is the library for AWS
audio-chunks/chunk52.wav : Note that moto 3 is the library for aws.
audio-chunks/chunk53.wav : variables are also set here for the zip files web
location and a local folder for extraction
audio-chunks/chunk53.wav : Variables are also set here for the zip files web
location and a local folder for extraction.
audio-chunks/chunk54.wav : the download and extract section makes a web request
audio-chunks/chunk54.wav : The download and extract section makes a web request.
audio-chunks/chunk55.wav : saving the bites from the URL is a stream
audio-chunks/chunk55.wav : Saving the bites from the url is a stream.
audio-chunks/chunk56.wav : this stream is passed to the zip file function which
```

```
is then used to extract the data
audio-chunks/chunk56.wav : This stream is passed to the zip file function which
is then used to extract the data.
audio-chunks/chunk57.wav : with the extracted files in a folder
audio-chunks/chunk57.wav : With the extracted files in a folder.
audio-chunks/chunk58.wav : the upload to S3 section enumerates the folders files
audio-chunks/chunk58.wav : The upload to s3 section enumerates the folders
files.
audio-chunks/chunk59.wav : Amazon S3
audio-chunks/chunk59.wav : Amazon s3.
audio-chunks/chunk60.wav : if you discover that the script is used often it
should be migrated to a stand-alone function that can be imported by other
python applications
audio-chunks/chunk60.wav : If you discover that the script is used often it
should be migrated to a stand-alone function that can be imported by other
python applications.
audio-chunks/chunk61.wav : that's it for part 2 of this section we'll see you
again for part 3 where we will review how to secure your data
audio-chunks/chunk61.wav : That's it for part 2 of this section we'll see you
again for part 3 where we will review how to secure your data.
audio-chunks/chunk1.wav : hi welcome back to module 3
audio-chunks/chunk1.wav : Hi welcome back to module 3.
audio-chunks/chunk2.wav : this is Section 5 on training
audio-chunks/chunk2.wav : This is section 5 on training.
audio-chunks/chunk3.wav : in this section we're going to look at how to select a
model and train it with the data we have pre-processed
audio-chunks/chunk3.wav : In this section we're going to look at how to select a
model and train it with the data we have pre-processed.
audio-chunks/chunk4.wav : at this point you've done a lot to clean and prepare
your data
audio-chunks/chunk4.wav : At this point you've done a lot to clean and prepare
your data.
audio-chunks/chunk5.wav : but that doesn't mean your data is completely ready to
train the algorithm
audio-chunks/chunk5.wav : But that doesn't mean your data is completely ready to
train the algorithm.
audio-chunks/chunk6.wav : some algorithms may not be able to work with training
data in a data frame format
audio-chunks/chunk6.wav : Some algorithms may not be able to work with training
data in a data frame format.
audio-chunks/chunk7.wav : some file formats like CSV are commonly used by
various algorithms
audio-chunks/chunk7.wav : Some file formats like csv are commonly used by
various algorithms.
audio-chunks/chunk8.wav : but they do not make use of that optimizations that
some of the file formats like record IO protobuf can use
audio-chunks/chunk8.wav : But they do not make use of that optimizations that
some of the file formats like record io protobuf can use.
```

```
audio-chunks/chunk9.wav : many Amazon sagemaker algorithms support training with
data in a CSV format
audio-chunks/chunk9.wav : Many amazon sagemaker algorithms support training with
data in a csv format.
audio-chunks/chunk10.wav : Amazon sagemaker requires that a CSV file doesn't
have a header record
audio-chunks/chunk10.wav : Amazon sagemaker requires that a csv file doesn't
have a header record.
audio-chunks/chunk11.wav : and that the target variable is in the first column
audio-chunks/chunk11.wav : And that the target variable is in the first column.
audio-chunks/chunk12.wav : most Amazon sagemaker algorithms work best when you
use the optimized protobuf record IO format for the training data
audio-chunks/chunk12.wav : Most amazon sagemaker algorithms work best when you
use the optimized protobuf record io format for the training data.
audio-chunks/chunk13.wav : using this format allows you to take advantage of
pipe mode when training the algorithms that support it
audio-chunks/chunk13.wav : Using this format allows you to take advantage of
pipe mode when training the algorithms that support it.
audio-chunks/chunk14.wav : in pipe mode
audio-chunks/chunk14.wav : In pipe mode.
audio-chunks/chunk15.wav : your training job streams data directly from Amazon
S3
audio-chunks/chunk15.wav : Your training job streams data directly from amazon
s3.
audio-chunks/chunk16.wav : when using the CSV format
audio-chunks/chunk16.wav : When using the csv format.
audio-chunks/chunk17.wav : the target variable in your training data set should
be the first column on the left
audio-chunks/chunk17.wav : The target variable in your training data set should
be the first column on the left.
audio-chunks/chunk18.wav : and your features should be to the right of the
target variable column
audio-chunks/chunk18.wav : And your features should be to the right of the
target variable column.
audio-chunks/chunk19.wav : evaluating a model with the same data that it trained
on will lead to overfitting
audio-chunks/chunk19.wav : Evaluating a model with the same data that it trained
on will lead to overfitting.
audio-chunks/chunk20.wav : recall overfitting is where your model learns the
particulars of a data set too well
audio-chunks/chunk20.wav : Recall overfitting is where your model learns the
particulars of a data set too well.
audio-chunks/chunk21.wav : it's essentially memorizing the training data
audio-chunks/chunk21.wav : It's essentially memorizing the training data.
audio-chunks/chunk22.wav : rather than learning the relationships between
features and labels
audio-chunks/chunk22.wav : Rather than learning the relationships between
features and labels.
```

```
audio-chunks/chunk23.wav : this means the model isn't learning from those
relationships and patterns to apply them to new data in the future
audio-chunks/chunk23.wav : This means the model isn't learning from those
relationships and patterns to apply them to new data in the future.
audio-chunks/chunk24.wav : hold out is when you split your data into multiple
sets
audio-chunks/chunk24.wav : Hold out is when you split your data into multiple
sets.
audio-chunks/chunk25.wav : commonly sets for training data
audio-chunks/chunk25.wav : Commonly sets for training data.
audio-chunks/chunk26.wav : validation data and testing data
audio-chunks/chunk26.wav : Validation data and testing data.
audio-chunks/chunk27.wav : training data which includes both features and labels
audio-chunks/chunk27.wav : Training data which includes both features and
labels.
audio-chunks/chunk28.wav : feeds into the algorithm you selected to produce your
model
audio-chunks/chunk28.wav : Feeds into the algorithm you selected to produce your
model.
audio-chunks/chunk29.wav : you then use the model to make predictions over the
validation data set which is where you'll likely notice things you'll want to
tweak and tune and change
audio-chunks/chunk29.wav : You then use the model to make predictions over the
validation data set which is where you'll likely notice things you'll want to
tweak and tune and change.
audio-chunks/chunk30.wav : then when you're ready you run the test data set
which only includes features
audio-chunks/chunk30.wav : Then when you're ready you run the test data set
which only includes features.
audio-chunks/chunk31.wav : since you want the labels to actually be predicted
audio-chunks/chunk31.wav : Since you want the labels to actually be predicted.
audio-chunks/chunk32.wav : the performance you get here with the test data set
is what you can reasonably expect to see in production
audio-chunks/chunk32.wav : The performance you get here with the test data set
is what you can reasonably expect to see in production.
audio-chunks/chunk33.wav : a common split when using the holdout method is using
80% of the data for a training set
audio-chunks/chunk33.wav : A common split when using the holdout method is using
80% of the data for a training set.
audio-chunks/chunk34.wav : 10% for validation
audio-chunks/chunk34.wav : 10% for validation.
audio-chunks/chunk35.wav : and 10% for test
audio-chunks/chunk35.wav : And 10% for test.
audio-chunks/chunk36.wav : or if you have a lot of data you can split it into
70% training
audio-chunks/chunk36.wav : Or if you have a lot of data you can split it into
70% training.
audio-chunks/chunk37.wav : 15% validation and 15%
```

```
audio-chunks/chunk37.wav : 15% validation and 15%.
audio-chunks/chunk38.wav : so for a small data set
audio-chunks/chunk38.wav : So for a small data set.
audio-chunks/chunk39.wav : we can use k-fold cross-validation to utilize as much
of the data as possible
audio-chunks/chunk39.wav : We can use k-fold cross-validation to utilize as much
of the data as possible.
audio-chunks/chunk40.wav : while still having relatively good metrics
audio-chunks/chunk40.wav : While still having relatively good metrics.
audio-chunks/chunk41.wav : in order to choose which model is better
audio-chunks/chunk41.wav : In order to choose which model is better.
audio-chunks/chunk42.wav : k-fold cross-validation randomly partitions the data
into K different segments
audio-chunks/chunk42.wav : K-fold cross-validation randomly partitions the data
into k different segments.
audio-chunks/chunk43.wav : for each segment we'll use the rest of the data
outside of it for training in order to do a validation on that particular
segment
audio-chunks/chunk43.wav : For each segment we'll use the rest of the data
outside of it for training in order to do a validation on that particular
segment.
audio-chunks/chunk44.wav : let's look at an example
audio-chunks/chunk44.wav : Let's look at an example.
audio-chunks/chunk45.wav : here we have a five-fold cross validation
audio-chunks/chunk45.wav : Here we have a five-fold cross validation.
audio-chunks/chunk46.wav : the available training data is separated into five
different chunks
audio-chunks/chunk46.wav : The available training data is separated into five
different chunks.
audio-chunks/chunk47.wav : for the training of the first model we're using all
those chunks as the training data
audio-chunks/chunk47.wav : For the training of the first model we're using all
those chunks as the training data.
audio-chunks/chunk48.wav : and then we're going to calculate the metrics on this
test
audio-chunks/chunk48.wav : And then we're going to calculate the metrics on this
test.
audio-chunks/chunk49.wav : for the second model we're going to use these pieces
as training
audio-chunks/chunk49.wav : For the second model we're going to use these pieces
as training.
audio-chunks/chunk50.wav : after the model is trained
audio-chunks/chunk50.wav : After the model is trained.
audio-chunks/chunk51.wav : you apply it to this test
audio-chunks/chunk51.wav : You apply it to this test.
audio-chunks/chunk52.wav : we do the same thing 5 times
audio-chunks/chunk52.wav : We do the same thing 5 times.
audio-chunks/chunk53.wav : we use all the training data and we tested on five
```

```
different models on different chunks of the test data eventually testing it on
all data points
audio-chunks/chunk53.wav : We use all the training data and we tested on five
different models on different chunks of the test data eventually testing it on
all data points.
audio-chunks/chunk54.wav : one other thing to note about splitting your data
audio-chunks/chunk54.wav : One other thing to note about splitting your data.
audio-chunks/chunk55.wav : data in a specific order can lead to biases on your
model
audio-chunks/chunk55.wav : Data in a specific order can lead to biases on your
model.
audio-chunks/chunk56.wav : this is especially true if you're working with
structured data
audio-chunks/chunk56.wav : This is especially true if you're working with
structured data.
audio-chunks/chunk57.wav : for example the wine data is ordered by the quality
column
audio-chunks/chunk57.wav : For example the wine data is ordered by the quality
column.
audio-chunks/chunk58.wav : when you run your model against your test data this
ordered pattern will be applied biasing the model
audio-chunks/chunk58.wav : When you run your model against your test data this
ordered pattern will be applied biasing the model.
audio-chunks/chunk59.wav : it might also mean that some targets are missing from
the training data
audio-chunks/chunk59.wav : It might also mean that some targets are missing from
the training data.
audio-chunks/chunk60.wav : typically randomizing your data set prior to
splitting is sufficient
audio-chunks/chunk60.wav : Typically randomizing your data set prior to
splitting is sufficient.
audio-chunks/chunk61.wav : and many libraries will provide functions for this
audio-chunks/chunk61.wav : And many libraries will provide functions for this.
audio-chunks/chunk62.wav : with smaller set
audio-chunks/chunk62.wav : With smaller set.
audio-chunks/chunk63.wav : it is sometimes useful to use stratified sampling
audio-chunks/chunk63.wav : It is sometimes useful to use stratified sampling.
audio-chunks/chunk64.wav : stratified sampling ensures that the training and
test sets have approximately the same percentage of samples of each target class
is the complete set
audio-chunks/chunk64.wav : Stratified sampling ensures that the training and
test sets have approximately the same percentage of samples of each target class
is the complete set.
audio-chunks/chunk65.wav : an internet search will give you many ways to shuffle
and split the data
audio-chunks/chunk65.wav : An internet search will give you many ways to shuffle
and split the data.
audio-chunks/chunk66.wav : one of the easiest is to use the train
```

```
audio-chunks/chunk66.wav : One of the easiest is to use the train.
audio-chunks/chunk67.wav : split function from sklearn
audio-chunks/chunk67.wav : Split function from sklearn.
audio-chunks/chunk68.wav : Amazon sagemaker provides four different ways you can
train models
audio-chunks/chunk68.wav : Amazon sagemaker provides four different ways you can
train models.
audio-chunks/chunk69.wav : the built-in algorithms available can be easily
deployed from the AWS console
audio-chunks/chunk69.wav : The built-in algorithms available can be easily
deployed from the aws console.
audio-chunks/chunk70.wav : CLI or a jupyter notebook
audio-chunks/chunk70.wav : Cli or a jupyter notebook.
audio-chunks/chunk71.wav : containers are used behind the scenes when you use
one of the Amazon sagemaker built in algorithms
audio-chunks/chunk71.wav : Containers are used behind the scenes when you use
one of the amazon sagemaker built in algorithms.
audio-chunks/chunk72.wav : but you do not have to deal with them directly
audio-chunks/chunk72.wav : But you do not have to deal with them directly.
audio-chunks/chunk73.wav : Amazon sagemaker supported Frameworks
audio-chunks/chunk73.wav : Amazon sagemaker supported frameworks.
audio-chunks/chunk74.wav : provide pre-built containers to support deep learning
Frameworks such as Apache MXN
audio-chunks/chunk74.wav : Provide pre-built containers to support deep learning
frameworks such as apache mxn.
audio-chunks/chunk75.wav : tensorflow
audio-chunks/chunk75.wav : Tensorflow.
audio-chunks/chunk76.wav : pie
audio-chunks/chunk76.wav : Pie.
audio-chunks/chunk77.wav : enchanter
audio-chunks/chunk77.wav : Enchanter.
audio-chunks/chunk78.wav : it also supports machine learning libraries such as
skykit learn and Spark ml by providing pre-built Docker images
audio-chunks/chunk78.wav : It also supports machine learning libraries such as
skykit learn and spark ml by providing pre-built docker images.
audio-chunks/chunk79.wav : if you use the Amazon sagemaker python SDK
audio-chunks/chunk79.wav : If you use the amazon sagemaker python sdk.
audio-chunks/chunk80.wav : they're deployed using their respective Amazon
sagemaker SDK estimator class
audio-chunks/chunk80.wav : They're deployed using their respective amazon
sagemaker sdk estimator class.
audio-chunks/chunk81.wav : if there is no pre-built Amazon sagemaker container
image that you can use or modify for an advanced scenario
audio-chunks/chunk81.wav : If there is no pre-built amazon sagemaker container
image that you can use or modify for an advanced scenario.
audio-chunks/chunk82.wav : you can package your own scripter algorithm to use
with Amazon sagemaker
audio-chunks/chunk82.wav : You can package your own scripter algorithm to use
```

```
with amazon sagemaker.
audio-chunks/chunk83.wav : you can use any programming language or framework to
develop your container
audio-chunks/chunk83.wav : You can use any programming language or framework to
develop your container.
audio-chunks/chunk84.wav : for an example
audio-chunks/chunk84.wav : For an example.
audio-chunks/chunk85.wav : if your team works and builds ml models in are you
can build your own containers to train and host an algorithm in are as well
audio-chunks/chunk85.wav : If your team works and builds ml models in are you
can build your own containers to train and host an algorithm in are as well.
audio-chunks/chunk86.wav : someone else may have already developed in tune to
model it is worth looking in the AWS Marketplace to find available models
audio-chunks/chunk86.wav : Someone else may have already developed in tune to
model it is worth looking in the aws marketplace to find available models.
audio-chunks/chunk87.wav : Amazon sagemaker provides high-performance
audio-chunks/chunk87.wav : Amazon sagemaker provides high-performance.
audio-chunks/chunk88.wav : scalable machine learning algorithms optimized for
Speed
audio-chunks/chunk88.wav : Scalable machine learning algorithms optimized for
speed.
audio-chunks/chunk89.wav : scale and accuracy
audio-chunks/chunk89.wav : Scale and accuracy.
audio-chunks/chunk90.wav : for supervised learning Amazon sagemaker includes
xgboost and linear learner algorithms for classification and quantitative or
regression problems
audio-chunks/chunk90.wav : For supervised learning amazon sagemaker includes
xgboost and linear learner algorithms for classification and quantitative or
regression problems.
audio-chunks/chunk91.wav : there is also a factorization machine to address
recommendation and time series prediction problems
audio-chunks/chunk91.wav : There is also a factorization machine to address
recommendation and time series prediction problems.
audio-chunks/chunk92.wav : Amazon sagemaker includes support for unsupervised
learning such as with K means clustering
audio-chunks/chunk92.wav : Amazon sagemaker includes support for unsupervised
learning such as with k means clustering.
audio-chunks/chunk93.wav : and principal component analysis PCA
audio-chunks/chunk93.wav : And principal component analysis pca.
audio-chunks/chunk94.wav : to solve problems like identifying customer groupings
based on purchasing Behavior
audio-chunks/chunk94.wav : To solve problems like identifying customer groupings
based on purchasing behavior.
audio-chunks/chunk95.wav : finally
audio-chunks/chunk95.wav : Finally.
audio-chunks/chunk96.wav : there are a selection of specialized algorithms for
processing images and other deep learning tasks
audio-chunks/chunk96.wav : There are a selection of specialized algorithms for
```

```
processing images and other deep learning tasks.
audio-chunks/chunk97.wav : let's look a little closer at three of the most
commonly used built-in algorithms and their use cases
audio-chunks/chunk97.wav : Let's look a little closer at three of the most
commonly used built-in algorithms and their use cases.
audio-chunks/chunk98.wav : xgboost or extreme gradient boosting is a popular and
efficient open source implementation of the gradient boosted trees algorithm
audio-chunks/chunk98.wav : Xgboost or extreme gradient boosting is a popular and
efficient open source implementation of the gradient boosted trees algorithm.
audio-chunks/chunk99.wav : gradient boosting
audio-chunks/chunk99.wav : Gradient boosting.
audio-chunks/chunk100.wav : is a supervised learning algorithm that attempts to
accurately predict a Target variable
audio-chunks/chunk100.wav : Is a supervised learning algorithm that attempts to
accurately predict a target variable.
audio-chunks/chunk101.wav : by combining an ensemble of estimates from a set of
simpler weaker models
audio-chunks/chunk101.wav : By combining an ensemble of estimates from a set of
simpler weaker models.
audio-chunks/chunk102.wav : xgboost has done remarkably well in machine learning
competitions because it robustly handles a variety of data types relationships
and distributions
audio-chunks/chunk102.wav : Xgboost has done remarkably well in machine learning
competitions because it robustly handles a variety of data types relationships
and distributions.
audio-chunks/chunk103.wav : the large number of hyperparameters can be tweaked
and tuned for improved fit
audio-chunks/chunk103.wav : The large number of hyperparameters can be tweaked
and tuned for improved fit.
audio-chunks/chunk104.wav : this flexibility makes xgboost a solid choice for
problems in regression
audio-chunks/chunk104.wav : This flexibility makes xgboost a solid choice for
problems in regression.
audio-chunks/chunk105.wav : classification binary and multaq
audio-chunks/chunk105.wav : Classification binary and multaq.
audio-chunks/chunk106.wav : and ranking
audio-chunks/chunk106.wav : And ranking.
audio-chunks/chunk107.wav : the Amazon sagemaker linear learner algorithm
provides a solution for both classification and regression problems
audio-chunks/chunk107.wav : The amazon sagemaker linear learner algorithm
provides a solution for both classification and regression problems.
audio-chunks/chunk108.wav : with the Amazon sagemaker algorithm you can
simultaneously explore different training objectives and choose the best
solution from your validation set
audio-chunks/chunk108.wav : With the amazon sagemaker algorithm you can
simultaneously explore different training objectives and choose the best
solution from your validation set.
audio-chunks/chunk109.wav : you can also explore a large number of models and
```

```
choose the best one for your needs
audio-chunks/chunk109.wav : You can also explore a large number of models and
choose the best one for your needs.
audio-chunks/chunk110.wav : compare with methods that provide a solution for
only continuous objectives
audio-chunks/chunk110.wav : Compare with methods that provide a solution for
only continuous objectives.
audio-chunks/chunk111.wav : the Amazon sagemaker linear learner algorithm
provides a significant increase in speed over naive hyperparameter optimization
techniques
audio-chunks/chunk111.wav : The amazon sagemaker linear learner algorithm
provides a significant increase in speed over naive hyperparameter optimization
techniques.
audio-chunks/chunk112.wav : K means is an unsupervised learning algorithm
audio-chunks/chunk112.wav : K means is an unsupervised learning algorithm.
audio-chunks/chunk113.wav : it attempts to find discreet groupings within data
where members of a group are as similar as possible to one another
audio-chunks/chunk113.wav : It attempts to find discreet groupings within data
where members of a group are as similar as possible to one another.
audio-chunks/chunk114.wav : and as different as possible from members of other
groups
audio-chunks/chunk114.wav : And as different as possible from members of other
groups.
audio-chunks/chunk115.wav : you to find the attributes that you want the
algorithm to use to determine similarity
audio-chunks/chunk115.wav : You to find the attributes that you want the
algorithm to use to determine similarity.
audio-chunks/chunk116.wav : to train a model in Amazon sagemaker you create a
training job
audio-chunks/chunk116.wav : To train a model in amazon sagemaker you create a
training job.
audio-chunks/chunk117.wav : the training job includes
audio-chunks/chunk117.wav : The training job includes.
audio-chunks/chunk118.wav : the URL of the Amazon S3 bucket where you stored the
training data
audio-chunks/chunk118.wav : The url of the amazon s3 bucket where you stored the
training data.
audio-chunks/chunk119.wav : the URL of the S3 bucket where you want to store the
output of the job
audio-chunks/chunk119.wav : The url of the s3 bucket where you want to store the
output of the job.
audio-chunks/chunk120.wav : the Amazon elastic container registry path where the
training code is stored
audio-chunks/chunk120.wav : The amazon elastic container registry path where the
training code is stored.
audio-chunks/chunk121.wav : the compute resources that you want Amazon sagemaker
to use for model training
audio-chunks/chunk121.wav : The compute resources that you want amazon sagemaker
```

```
to use for model training.
audio-chunks/chunk122.wav : compute resources are ml compute instances managed
by Amazon sagemaker
audio-chunks/chunk122.wav : Compute resources are ml compute instances managed
by amazon sagemaker.
audio-chunks/chunk123.wav : Amazon sagemaker provides a selection of instance
types optimized to fit different machine learning use cases
audio-chunks/chunk123.wav : Amazon sagemaker provides a selection of instance
types optimized to fit different machine learning use cases.
audio-chunks/chunk124.wav : instance types comprise varying combinations of CPU
GPU memory and networking capacity
audio-chunks/chunk124.wav : Instance types comprise varying combinations of cpu
gpu memory and networking capacity.
audio-chunks/chunk125.wav : and give you the flexibility to choose the
appropriate mix of resources for building
audio-chunks/chunk125.wav : And give you the flexibility to choose the
appropriate mix of resources for building.
audio-chunks/chunk126.wav : training and deploying your ml models
audio-chunks/chunk126.wav : Training and deploying your ml models.
audio-chunks/chunk127.wav : each instance type includes one or more instant
sizes allowing you to scale your resources to the requirements of your target
workload
audio-chunks/chunk127.wav : Each instance type includes one or more instant
sizes allowing you to scale your resources to the requirements of your target
workload.
audio-chunks/chunk128.wav : some key takeaways from this section of the module
include
audio-chunks/chunk128.wav : Some key takeaways from this section of the module
include.
audio-chunks/chunk129.wav : split data into training and testing sets helps you
validate the models accuracy
audio-chunks/chunk129.wav : Split data into training and testing sets helps you
validate the models accuracy.
audio-chunks/chunk130.wav : K fold cross validation can help with smaller data
sets
audio-chunks/chunk130.wav : K fold cross validation can help with smaller data
sets.
audio-chunks/chunk131.wav : two key algorithms for supervised learning are
xgboost
audio-chunks/chunk131.wav : Two key algorithms for supervised learning are
xgboost.
audio-chunks/chunk132.wav : and linear learner
audio-chunks/chunk132.wav : And linear learner.
audio-chunks/chunk133.wav : use K means for unsupervised learning
audio-chunks/chunk133.wav : Use k means for unsupervised learning.
audio-chunks/chunk134.wav : and use Amazon sagemaker to train models
audio-chunks/chunk134.wav : And use amazon sagemaker to train models.
audio-chunks/chunk135.wav : that's it for Section 5 I hope to see you in the
```

```
next video
audio-chunks/chunk135.wav : That's it for section 5 i hope to see you in the
next video.
audio-chunks/chunk1.wav : it's now time to review the module and wrap up with a
knowledge check
audio-chunks/chunk1.wav : It's now time to review the module and wrap up with a
knowledge check.
audio-chunks/chunk2.wav : in this module you learned how to
audio-chunks/chunk2.wav : In this module you learned how to.
audio-chunks/chunk3.wav : formulate a problem from a business request
audio-chunks/chunk3.wav : Formulate a problem from a business request.
audio-chunks/chunk4.wav : obtain and secure data for machine learning
audio-chunks/chunk4.wav : Obtain and secure data for machine learning.
audio-chunks/chunk5.wav : build a jupyter notebook by using Amazon sagemaker
audio-chunks/chunk5.wav : Build a jupyter notebook by using amazon sagemaker.
audio-chunks/chunk6.wav : outline the process for evaluating data
audio-chunks/chunk6.wav : Outline the process for evaluating data.
audio-chunks/chunk7.wav : explain why data needs to be pre-processed
audio-chunks/chunk7.wav : Explain why data needs to be pre-processed.
audio-chunks/chunk8.wav : use open source tools to examine and pre-process data
audio-chunks/chunk8.wav : Use open source tools to examine and pre-process data.
audio-chunks/chunk9.wav : use Amazon sagemaker to train and host a machine
learning model
audio-chunks/chunk9.wav : Use amazon sagemaker to train and host a machine
learning model.
audio-chunks/chunk10.wav : use cross-validation to test the performance of an ml
model
audio-chunks/chunk10.wav : Use cross-validation to test the performance of an ml
model.
audio-chunks/chunk11.wav : use a hosted model for inference
audio-chunks/chunk11.wav : Use a hosted model for inference.
audio-chunks/chunk12.wav : and create an Amazon sagemaker hyperparameter tuning
job to optimize a model's effectiveness
audio-chunks/chunk12.wav : And create an amazon sagemaker hyperparameter tuning
job to optimize a model's effectiveness.
audio-chunks/chunk13.wav : that concludes this module thanks for watching we'll
see you again in the next video
audio-chunks/chunk13.wav : That concludes this module thanks for watching we'll
see you again in the next video.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring feature engineering by
describing how to work with outliers
audio-chunks/chunk2.wav : Will continue exploring feature engineering by
describing how to work with outliers.
audio-chunks/chunk3.wav : you might also need to clean your data based on any
outliers that exist
audio-chunks/chunk3.wav : You might also need to clean your data based on any
```

```
outliers that exist.
audio-chunks/chunk4.wav : outliers are points in your data set that lie at an
abnormal distance from other values
audio-chunks/chunk4.wav : Outliers are points in your data set that lie at an
abnormal distance from other values.
audio-chunks/chunk5.wav : they're not always something you want to clean up
because they can add richness to your data set
audio-chunks/chunk5.wav : They're not always something you want to clean up
because they can add richness to your data set.
audio-chunks/chunk6.wav : but they can also make it harder to make accurate
predictions
audio-chunks/chunk6.wav : But they can also make it harder to make accurate
predictions.
audio-chunks/chunk7.wav : because they skew values away from the other more
normal values related to that feature
audio-chunks/chunk7.wav : Because they skew values away from the other more
normal values related to that feature.
audio-chunks/chunk8.wav : an outlier might also indicate that the data point
actually belongs to another column
audio-chunks/chunk8.wav : An outlier might also indicate that the data point
actually belongs to another column.
audio-chunks/chunk9.wav : you can think of outliers as falling into two broad
categories
audio-chunks/chunk9.wav : You can think of outliers as falling into two broad
categories.
audio-chunks/chunk10.wav : the first is a single variation for just a single
variable or a univariate outlier
audio-chunks/chunk10.wav : The first is a single variation for just a single
variable or a univariate outlier.
audio-chunks/chunk11.wav : the second is a variation of two or more variables or
a multivariate outlier
audio-chunks/chunk11.wav : The second is a variation of two or more variables or
a multivariate outlier.
audio-chunks/chunk12.wav : one of the more common ways to find univariate
outliers is with a box plot
audio-chunks/chunk12.wav : One of the more common ways to find univariate
outliers is with a box plot.
audio-chunks/chunk13.wav : a box plot shows how far a data point is to the mean
for that variable
audio-chunks/chunk13.wav : A box plot shows how far a data point is to the mean
for that variable.
audio-chunks/chunk14.wav : the box in the plot shows the data values within two
quartiles of the mean
audio-chunks/chunk14.wav : The box in the plot shows the data values within two
quartiles of the mean.
audio-chunks/chunk15.wav : values outside that range are represented by the
lines extending from the box
audio-chunks/chunk15.wav : Values outside that range are represented by the
```

```
lines extending from the box.
audio-chunks/chunk16.wav : which are sometimes called whiskers
audio-chunks/chunk16.wav : Which are sometimes called whiskers.
audio-chunks/chunk17.wav : a scatter plot can be an effective way to see
multivariate outliers
audio-chunks/chunk17.wav : A scatter plot can be an effective way to see
multivariate outliers.
audio-chunks/chunk18.wav : for example this diagram shows the amount of sulfates
and alcohol in a collection of wines
audio-chunks/chunk18.wav : For example this diagram shows the amount of sulfates
and alcohol in a collection of wines.
audio-chunks/chunk19.wav : with the scatter plot you can quickly visualize
whether there are multivariate outliers for the two variables
audio-chunks/chunk19.wav : With the scatter plot you can quickly visualize
whether there are multivariate outliers for the two variables.
audio-chunks/chunk20.wav : the origin of your outlier will most likely inform
how you deal with it during this pre-processing phase of the pipeline or
possibly later during feature engineering
audio-chunks/chunk20.wav : The origin of your outlier will most likely inform
how you deal with it during this pre-processing phase of the pipeline or
possibly later during feature engineering.
audio-chunks/chunk21.wav : there are several different approaches to dealing
with outliers
audio-chunks/chunk21.wav : There are several different approaches to dealing
with outliers.
audio-chunks/chunk22.wav : you could delete the outlier
audio-chunks/chunk22.wav : You could delete the outlier.
audio-chunks/chunk23.wav : if you're outlier is based on an artificial error
audio-chunks/chunk23.wav : If you're outlier is based on an artificial error.
audio-chunks/chunk24.wav : this means the outlier isn't natural and was
introduced because of some failure like incorrectly enter data
audio-chunks/chunk24.wav : This means the outlier isn't natural and was
introduced because of some failure like incorrectly enter data.
audio-chunks/chunk25.wav : you can also transform the outlier
audio-chunks/chunk25.wav : You can also transform the outlier.
audio-chunks/chunk26.wav : by taking the natural log of a value
audio-chunks/chunk26.wav : By taking the natural log of a value.
audio-chunks/chunk27.wav : this in turn reduces the variation caused by the
extreme outlier value
audio-chunks/chunk27.wav : This in turn reduces the variation caused by the
extreme outlier value.
audio-chunks/chunk28.wav : which would then reduce the outliers influence on the
overall data set
audio-chunks/chunk28.wav : Which would then reduce the outliers influence on the
overall data set.
audio-chunks/chunk29.wav : finally you could use the mean of the feature and
impute that value to replace the outlier value
audio-chunks/chunk29.wav : Finally you could use the mean of the feature and
```

```
impute that value to replace the outlier value.
audio-chunks/chunk30.wav : again this would be a good approach of the outlier
was caused by artificial error
audio-chunks/chunk30.wav : Again this would be a good approach of the outlier
was caused by artificial error.
audio-chunks/chunk31.wav : this isn't an exhaustive list but it describes the
most common options
audio-chunks/chunk31.wav : This isn't an exhaustive list but it describes the
most common options.
audio-chunks/chunk32.wav : after you've extracted features
audio-chunks/chunk32.wav : After you've extracted features.
audio-chunks/chunk33.wav : you'll need to select the most appropriate features
for training your model
audio-chunks/chunk33.wav : You'll need to select the most appropriate features
for training your model.
audio-chunks/chunk34.wav : there are three main feature selection methods
audio-chunks/chunk34.wav : There are three main feature selection methods.
audio-chunks/chunk35.wav : filter methods used statistical methods to measure
the relevance of features by their correlation with the target variable
audio-chunks/chunk35.wav : Filter methods used statistical methods to measure
the relevance of features by their correlation with the target variable.
audio-chunks/chunk36.wav : rapper methods
audio-chunks/chunk36.wav : Rapper methods.
audio-chunks/chunk37.wav : measure how useful a subset of a feature is
audio-chunks/chunk37.wav : Measure how useful a subset of a feature is.
audio-chunks/chunk38.wav : they do this by training a model on the feature
audio-chunks/chunk38.wav : They do this by training a model on the feature.
audio-chunks/chunk39.wav : and then measuring how successful the model is
audio-chunks/chunk39.wav : And then measuring how successful the model is.
audio-chunks/chunk40.wav : filters are faster and cheaper than rapper methods
audio-chunks/chunk40.wav : Filters are faster and cheaper than rapper methods.
audio-chunks/chunk41.wav : because they don't involve training the models
repeatedly
audio-chunks/chunk41.wav : Because they don't involve training the models
repeatedly.
audio-chunks/chunk42.wav : rappers typically find the best subset of features
audio-chunks/chunk42.wav : Rappers typically find the best subset of features.
audio-chunks/chunk43.wav : but there's a risk of overfitting compared to using
subsets of features from Filter methods
audio-chunks/chunk43.wav : But there's a risk of overfitting compared to using
subsets of features from filter methods.
audio-chunks/chunk44.wav : embedded methods are algorithms specific
audio-chunks/chunk44.wav : Embedded methods are algorithms specific.
audio-chunks/chunk45.wav : and they might use a combination of both filters and
wrappers
audio-chunks/chunk45.wav : And they might use a combination of both filters and
wrappers.
audio-chunks/chunk46.wav : filter methods use a proxy measure instead of the
```

```
actual models performance
audio-chunks/chunk46.wav : Filter methods use a proxy measure instead of the
actual models performance.
audio-chunks/chunk47.wav : they're fast to compute
audio-chunks/chunk47.wav : They're fast to compute.
audio-chunks/chunk48.wav : but they can still capture how useful the feature set
is
audio-chunks/chunk48.wav : But they can still capture how useful the feature set
is.
audio-chunks/chunk49.wav : here are some common measures
audio-chunks/chunk49.wav : Here are some common measures.
audio-chunks/chunk50.wav : the first is Pearson's correlation coefficient
audio-chunks/chunk50.wav : The first is pearson's correlation coefficient.
audio-chunks/chunk51.wav : which measures the statistical relationship or
association between two continuous variables
audio-chunks/chunk51.wav : Which measures the statistical relationship or
association between two continuous variables.
audio-chunks/chunk52.wav : the second is linear discriminant analysis or LDA
audio-chunks/chunk52.wav : The second is linear discriminant analysis or lda.
audio-chunks/chunk53.wav : this is used to find a linear combination of features
that separates two or more classes
audio-chunks/chunk53.wav : This is used to find a linear combination of features
that separates two or more classes.
audio-chunks/chunk54.wav : the third is analysis of variance or Anova
audio-chunks/chunk54.wav : The third is analysis of variance or anova.
audio-chunks/chunk55.wav : this is used to analyze the differences among group
means in a sample
audio-chunks/chunk55.wav : This is used to analyze the differences among group
means in a sample.
audio-chunks/chunk56.wav : and finally chi-square is a single number that tells
you how much difference exists between your observed count
audio-chunks/chunk56.wav : And finally chi-square is a single number that tells
you how much difference exists between your observed count.
audio-chunks/chunk57.wav : and the counts you'd expect if there were absolutely
no relationships in the population
audio-chunks/chunk57.wav : And the counts you'd expect if there were absolutely
no relationships in the population.
audio-chunks/chunk58.wav : filters are usually less computationally intensive
than rappers
audio-chunks/chunk58.wav : Filters are usually less computationally intensive
than rappers.
audio-chunks/chunk59.wav : but they produce a feature set that isn't tuned to a
specific type of predictive model
audio-chunks/chunk59.wav : But they produce a feature set that isn't tuned to a
specific type of predictive model.
audio-chunks/chunk60.wav : this lack of tuning means a feature set from a filter
is more General than one from a rapper
audio-chunks/chunk60.wav : This lack of tuning means a feature set from a filter
```

```
is more general than one from a rapper.
audio-chunks/chunk61.wav : the filter also usually has a lower prediction
performance than a rapper
audio-chunks/chunk61.wav : The filter also usually has a lower prediction
performance than a rapper.
audio-chunks/chunk62.wav : however the filters feature set doesn't contain the
assumptions of a prediction model
audio-chunks/chunk62.wav : However the filters feature set doesn't contain the
assumptions of a prediction model.
audio-chunks/chunk63.wav : so it's more useful for exposing relationships
between features
audio-chunks/chunk63.wav : So it's more useful for exposing relationships
between features.
audio-chunks/chunk64.wav : many filters provide feature ranking instead of an
explicit best feature subset
audio-chunks/chunk64.wav : Many filters provide feature ranking instead of an
explicit best feature subset.
audio-chunks/chunk65.wav : and the cutoff point in the ranking is chosen through
cross-validation
audio-chunks/chunk65.wav : And the cutoff point in the ranking is chosen through
cross-validation.
audio-chunks/chunk66.wav : filters have also been used as a preprocessing step
for wrappers
audio-chunks/chunk66.wav : Filters have also been used as a preprocessing step
for wrappers.
audio-chunks/chunk67.wav : which enables a wrapper to be used on larger problems
audio-chunks/chunk67.wav : Which enables a wrapper to be used on larger
problems.
audio-chunks/chunk68.wav : rapper methods use a predictive model to score
feature subsets
audio-chunks/chunk68.wav : Rapper methods use a predictive model to score
feature subsets.
audio-chunks/chunk69.wav : each new subset is used to train a
audio-chunks/chunk69.wav : Each new subset is used to train a.
audio-chunks/chunk70.wav : which is then tested on a holdout set
audio-chunks/chunk70.wav : Which is then tested on a holdout set.
audio-chunks/chunk71.wav : the score for that subset is calculated by counting
the number of mistakes made on that holdout set
audio-chunks/chunk71.wav : The score for that subset is calculated by counting
the number of mistakes made on that holdout set.
audio-chunks/chunk72.wav : what are the error rate of the model
audio-chunks/chunk72.wav : What are the error rate of the model.
audio-chunks/chunk73.wav : because rappers train a new model for each subset
their computationally intensive
audio-chunks/chunk73.wav : Because rappers train a new model for each subset
their computationally intensive.
audio-chunks/chunk74.wav : however they usually provide the best performing
feature set for that particular type of model or problem
```

```
audio-chunks/chunk74.wav : However they usually provide the best performing
feature set for that particular type of model or problem.
audio-chunks/chunk75.wav : forward selection starts with no features and adds
them until the best model is found
audio-chunks/chunk75.wav : Forward selection starts with no features and adds
them until the best model is found.
audio-chunks/chunk76.wav : backward selection starts with all features
audio-chunks/chunk76.wav : Backward selection starts with all features.
audio-chunks/chunk77.wav : drop some one at a time and then select the best
model
audio-chunks/chunk77.wav : Drop some one at a time and then select the best
model.
audio-chunks/chunk78.wav : embedded methods combine the qualities of filter and
wrapper methods
audio-chunks/chunk78.wav : Embedded methods combine the qualities of filter and
wrapper methods.
audio-chunks/chunk79.wav : they're implemented by algorithms that have their own
built-in feature selection methods
audio-chunks/chunk79.wav : They're implemented by algorithms that have their own
built-in feature selection methods.
audio-chunks/chunk80.wav : some of the most popular examples of these methods
are lasso and ridge regression
audio-chunks/chunk80.wav : Some of the most popular examples of these methods
are lasso and ridge regression.
audio-chunks/chunk81.wav : they have built-in penalization functions to reduce
overfitting
audio-chunks/chunk81.wav : They have built-in penalization functions to reduce
overfitting.
audio-chunks/chunk82.wav : here are some key takeaways from this section of the
module
audio-chunks/chunk82.wav : Here are some key takeaways from this section of the
module.
Error:
audio-chunks/chunk84.wav : feature engineering involves selecting the best
features for machine learning
audio-chunks/chunk84.wav : Feature engineering involves selecting the best
features for machine learning.
audio-chunks/chunk85.wav : preprocessing gives you better data to work with and
better data typically provides better results
audio-chunks/chunk85.wav : Preprocessing gives you better data to work with and
better data typically provides better results.
audio-chunks/chunk86.wav : two categories for pre-processing are converting data
to numerical values
audio-chunks/chunk86.wav : Two categories for pre-processing are converting data
to numerical values.
audio-chunks/chunk87.wav : and cleaning up dirty data by removing missing data
and cleaning outliers
audio-chunks/chunk87.wav : And cleaning up dirty data by removing missing data
```

```
and cleaning outliers.
audio-chunks/chunk88.wav : finally
audio-chunks/chunk88.wav : Finally.
audio-chunks/chunk89.wav : how you handle dirty data impacts your model
audio-chunks/chunk89.wav : How you handle dirty data impacts your model.
audio-chunks/chunk90.wav : that's it for section 4 we'll see you in the next
video
audio-chunks/chunk90.wav : That's it for section 4 we'll see you in the next
video.
audio-chunks/chunk1.wav : welcome back to AWS Academy machine learning
audio-chunks/chunk1.wav : Welcome back to aws academy machine learning.
audio-chunks/chunk2.wav : this is module 3 and we're going to work through the
entire machine learning pipeline by using Amazon sagemaker
audio-chunks/chunk2.wav : This is module 3 and we're going to work through the
entire machine learning pipeline by using amazon sagemaker.
audio-chunks/chunk3.wav : this module discuss a typical process for handling a
machine learning problem
audio-chunks/chunk3.wav : This module discuss a typical process for handling a
machine learning problem.
audio-chunks/chunk4.wav : the machine learning pipeline can be applied to many
machine learning problems
audio-chunks/chunk4.wav : The machine learning pipeline can be applied to many
machine learning problems.
audio-chunks/chunk5.wav : the focus is on supervised learning
audio-chunks/chunk5.wav : The focus is on supervised learning.
audio-chunks/chunk6.wav : but the process you learn in this module can be
adapted to other types of machine learning as well
audio-chunks/chunk6.wav : But the process you learn in this module can be
adapted to other types of machine learning as well.
audio-chunks/chunk7.wav : this is a large mod
audio-chunks/chunk7.wav : This is a large mod.
audio-chunks/chunk8.wav : and we'll be covering a lot of material
audio-chunks/chunk8.wav : And we'll be covering a lot of material.
audio-chunks/chunk9.wav : at the end of this module you'll be able to
audio-chunks/chunk9.wav : At the end of this module you'll be able to.
audio-chunks/chunk10.wav : formulate a problem from a business request
audio-chunks/chunk10.wav : Formulate a problem from a business request.
audio-chunks/chunk11.wav : obtain and secure data for machine learning
audio-chunks/chunk11.wav : Obtain and secure data for machine learning.
audio-chunks/chunk12.wav : build a jupyter notebook by using Amazon sagemaker
audio-chunks/chunk12.wav : Build a jupyter notebook by using amazon sagemaker.
audio-chunks/chunk13.wav : outline the process for evaluating data
audio-chunks/chunk13.wav : Outline the process for evaluating data.
audio-chunks/chunk14.wav : explain why data needs to be pre-processed
audio-chunks/chunk14.wav : Explain why data needs to be pre-processed.
audio-chunks/chunk15.wav : use open source tools to examine and pre-process data
audio-chunks/chunk15.wav : Use open source tools to examine and pre-process
data.
```

```
audio-chunks/chunk16.wav : use Amazon sagemaker to train and host a machine
learning model
audio-chunks/chunk16.wav : Use amazon sagemaker to train and host a machine
learning model.
audio-chunks/chunk17.wav : use cross validation to test the performance of a
machine learning model
audio-chunks/chunk17.wav : Use cross validation to test the performance of a
machine learning model.
audio-chunks/chunk18.wav : use a hosted model for inference
audio-chunks/chunk18.wav : Use a hosted model for inference.
audio-chunks/chunk19.wav : and finally create an Amazon sagemaker hyperparameter
tuning job to optimize a model's effectiveness
audio-chunks/chunk19.wav : And finally create an amazon sagemaker hyperparameter
tuning job to optimize a model's effectiveness.
audio-chunks/chunk20.wav : we're ready to get started
audio-chunks/chunk20.wav : We're ready to get started.
audio-chunks/chunk21.wav : see you in the next video
audio-chunks/chunk21.wav : See you in the next video.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : this is section one and we're going to introduce
computer vision
audio-chunks/chunk2.wav : This is section one and we're going to introduce
computer vision.
audio-chunks/chunk3.wav : computer vision is an exciting space in machine
learning
audio-chunks/chunk3.wav : Computer vision is an exciting space in machine
learning.
audio-chunks/chunk4.wav : you can think of computer vision as the automated
extraction of information from digital images
audio-chunks/chunk4.wav : You can think of computer vision as the automated
extraction of information from digital images.
audio-chunks/chunk5.wav : using computer vision machines can identify people
places and things in images with an accuracy that's at or above human levels
audio-chunks/chunk5.wav : Using computer vision machines can identify people
places and things in images with an accuracy that's at or above human levels.
audio-chunks/chunk6.wav : and with greater speed and efficiency
audio-chunks/chunk6.wav : And with greater speed and efficiency.
audio-chunks/chunk7.wav : computer vision is often built with deep learning
models
audio-chunks/chunk7.wav : Computer vision is often built with deep learning
models.
audio-chunks/chunk8.wav : it automates the extraction
audio-chunks/chunk8.wav : It automates the extraction.
audio-chunks/chunk9.wav : analysis
audio-chunks/chunk9.wav : Analysis.
audio-chunks/chunk10.wav : classification and understanding of useful
information from a single image or a sequence of images
```

```
audio-chunks/chunk10.wav : Classification and understanding of useful
information from a single image or a sequence of images.
audio-chunks/chunk11.wav : the image data can take many forms
audio-chunks/chunk11.wav : The image data can take many forms.
audio-chunks/chunk12.wav : such as single images
audio-chunks/chunk12.wav : Such as single images.
audio-chunks/chunk13.wav : video sequences
audio-chunks/chunk13.wav : Video sequences.
audio-chunks/chunk14.wav : views from multiple cameras
audio-chunks/chunk14.wav : Views from multiple cameras.
audio-chunks/chunk15.wav : or three-dimensional data
audio-chunks/chunk15.wav : Or three-dimensional data.
audio-chunks/chunk16.wav : computing power and algorithms have advanced over the
last 10 years
audio-chunks/chunk16.wav : Computing power and algorithms have advanced over the
last 10 years.
audio-chunks/chunk17.wav : this has led to an increase in capabilities and
easier access to computer vision Technologies
audio-chunks/chunk17.wav : This has led to an increase in capabilities and
easier access to computer vision technologies.
audio-chunks/chunk18.wav : so how is computer vision being used
audio-chunks/chunk18.wav : So how is computer vision being used.
audio-chunks/chunk19.wav : here are some of the primary use cases for computer
vision
audio-chunks/chunk19.wav : Here are some of the primary use cases for computer
vision.
audio-chunks/chunk20.wav : you can use image and facial recognition to improve
Public Safety and Home Security
audio-chunks/chunk20.wav : You can use image and facial recognition to improve
public safety and home security.
audio-chunks/chunk21.wav : or as a way to authenticate access to personal
devices
audio-chunks/chunk21.wav : Or as a way to authenticate access to personal
devices.
audio-chunks/chunk22.wav : you can also use it to automatically classify images
for Content management and Analysis
audio-chunks/chunk22.wav : You can also use it to automatically classify images
for content management and analysis.
audio-chunks/chunk23.wav : autonomous driving is partly enabled by computer
vision Technologies
audio-chunks/chunk23.wav : Autonomous driving is partly enabled by computer
vision technologies.
audio-chunks/chunk24.wav : and so are safety features of cars
audio-chunks/chunk24.wav : And so are safety features of cars.
audio-chunks/chunk25.wav : such as Lane detection or collision avoidance
audio-chunks/chunk25.wav : Such as lane detection or collision avoidance.
audio-chunks/chunk26.wav : medical image analysis with computer vision can
improve the accuracy and speed of a patient's medical diagnosis
```

```
audio-chunks/chunk26.wav : Medical image analysis with computer vision can
improve the accuracy and speed of a patient's medical diagnosis.
audio-chunks/chunk27.wav : this can result in better treatment outcomes and life
expectancy for the patient
audio-chunks/chunk27.wav : This can result in better treatment outcomes and life
expectancy for the patient.
audio-chunks/chunk28.wav : and finally in manufacturing
audio-chunks/chunk28.wav : And finally in manufacturing.
audio-chunks/chunk29.wav : well-trained computer vision is incorporated into
robotics
audio-chunks/chunk29.wav : Well-trained computer vision is incorporated into
robotics.
audio-chunks/chunk30.wav : this can improve quality assurance and operational
efficiencies
audio-chunks/chunk30.wav : This can improve quality assurance and operational
efficiencies.
audio-chunks/chunk31.wav : these are just a few examples and you can probably
think of more
audio-chunks/chunk31.wav : These are just a few examples and you can probably
think of more.
audio-chunks/chunk32.wav : computer vision problems can be broken down into a
few areas
audio-chunks/chunk32.wav : Computer vision problems can be broken down into a
few areas.
audio-chunks/chunk33.wav : content recognition is about identifying things in
images
audio-chunks/chunk33.wav : Content recognition is about identifying things in
images.
audio-chunks/chunk34.wav : it's a classification problem
audio-chunks/chunk34.wav : It's a classification problem.
audio-chunks/chunk35.wav : what is a complex one with several layers
audio-chunks/chunk35.wav : What is a complex one with several layers.
audio-chunks/chunk36.wav : in the picture here what's represented
audio-chunks/chunk36.wav : In the picture here what's represented.
audio-chunks/chunk37.wav : is it breakfast lunch or dinner
audio-chunks/chunk37.wav : Is it breakfast lunch or dinner.
audio-chunks/chunk38.wav : will the classification only be food
audio-chunks/chunk38.wav : Will the classification only be food.
audio-chunks/chunk39.wav : the answer depends on what model you use to perform
the classification
audio-chunks/chunk39.wav : The answer depends on what model you use to perform
the classification.
audio-chunks/chunk40.wav : models must be trained
audio-chunks/chunk40.wav : Models must be trained.
audio-chunks/chunk41.wav : and the training data provides the algorithm with
data for it to learn from
audio-chunks/chunk41.wav : And the training data provides the algorithm with
data for it to learn from.
```

```
audio-chunks/chunk42.wav : say that you have a model that was trained with
pictures of different types of food
audio-chunks/chunk42.wav : Say that you have a model that was trained with
pictures of different types of food.
audio-chunks/chunk43.wav : you might expect the image to Output categories such
as milk
audio-chunks/chunk43.wav : You might expect the image to output categories such
as milk.
audio-chunks/chunk44.wav : peaches mashed potato chicken nuggets and salad
audio-chunks/chunk44.wav : Peaches mashed potato chicken nuggets and salad.
audio-chunks/chunk45.wav : if you trained the model with different images
audio-chunks/chunk45.wav : If you trained the model with different images.
audio-chunks/chunk46.wav : it could classify objects as Trey
audio-chunks/chunk46.wav : It could classify objects as trey.
audio-chunks/chunk47.wav : Cutlery and napkin instead
audio-chunks/chunk47.wav : Cutlery and napkin instead.
audio-chunks/chunk48.wav : when you work with images
audio-chunks/chunk48.wav : When you work with images.
audio-chunks/chunk49.wav : you might want to know what kinds of objects are in
the image
audio-chunks/chunk49.wav : You might want to know what kinds of objects are in
the image.
audio-chunks/chunk50.wav : and the location of those objects
audio-chunks/chunk50.wav : And the location of those objects.
audio-chunks/chunk51.wav : object detection provides the image categories and
where the objects are located in the image
audio-chunks/chunk51.wav : Object detection provides the image categories and
where the objects are located in the image.
audio-chunks/chunk52.wav : there's a set of coordinates defining the location of
a box surrounding the image
audio-chunks/chunk52.wav : There's a set of coordinates defining the location of
a box surrounding the image.
audio-chunks/chunk53.wav : which is known as the bounding box
audio-chunks/chunk53.wav : Which is known as the bounding box.
audio-chunks/chunk54.wav : bounding boxes for detection
audio-chunks/chunk54.wav : Bounding boxes for detection.
audio-chunks/chunk55.wav : are typically
audio-chunks/chunk55.wav : Are typically.
audio-chunks/chunk56.wav : left
audio-chunks/chunk56.wav : Left.
Error:
audio-chunks/chunk58.wav : and height coordinates surrounding the images
audio-chunks/chunk58.wav : And height coordinates surrounding the images.
audio-chunks/chunk59.wav : you can use these coordinates in your applications
audio-chunks/chunk59.wav : You can use these coordinates in your applications.
audio-chunks/chunk60.wav : when objects are detected in an image
audio-chunks/chunk60.wav : When objects are detected in an image.
audio-chunks/chunk61.wav : there's a confidence number usually associated with
```

```
that object
audio-chunks/chunk61.wav : There's a confidence number usually associated with
that object.
audio-chunks/chunk62.wav : this percentage indicates the probability that the
object belongs to a specific class
audio-chunks/chunk62.wav : This percentage indicates the probability that the
object belongs to a specific class.
audio-chunks/chunk63.wav : this confidence level is important when you want to
determine an action that's based on object detection
audio-chunks/chunk63.wav : This confidence level is important when you want to
determine an action that's based on object detection.
audio-chunks/chunk64.wav : especially in facial detection applications
audio-chunks/chunk64.wav : Especially in facial detection applications.
audio-chunks/chunk65.wav : or cases where the action has significance
audio-chunks/chunk65.wav : Or cases where the action has significance.
audio-chunks/chunk66.wav : object segmentation is also known as semantic
segmentation
audio-chunks/chunk66.wav : Object segmentation is also known as semantic
segmentation.
Error:
audio-chunks/chunk68.wav : but you go into more detail to get fine boundaries
for each detected object
audio-chunks/chunk68.wav : But you go into more detail to get fine boundaries
for each detected object.
audio-chunks/chunk69.wav : basically it's a fine-grained inference for
predicting each pixel in the image
audio-chunks/chunk69.wav : Basically it's a fine-grained inference for
predicting each pixel in the image.
audio-chunks/chunk70.wav : some applications that require object segmentation
include autonomous vehicles and Advanced Computer human interactions
audio-chunks/chunk70.wav : Some applications that require object segmentation
include autonomous vehicles and advanced computer human interactions.
audio-chunks/chunk71.wav : the object segmentation is a key problem in the field
of computer vision we won't be covering it in this course
audio-chunks/chunk71.wav : The object segmentation is a key problem in the field
of computer vision we won't be covering it in this course.
audio-chunks/chunk72.wav : video add another dimension to computer vision
audio-chunks/chunk72.wav : Video add another dimension to computer vision.
audio-chunks/chunk73.wav : with video you get more data to work with
audio-chunks/chunk73.wav : With video you get more data to work with.
audio-chunks/chunk74.wav : so you can capture the movement of people or objects
which are referred to as instances
audio-chunks/chunk74.wav : So you can capture the movement of people or objects
which are referred to as instances.
audio-chunks/chunk75.wav : for example you can detect people who enter and leave
frames
audio-chunks/chunk75.wav : For example you can detect people who enter and leave
frames.
```

```
audio-chunks/chunk76.wav : and also deal with moving cameras
audio-chunks/chunk76.wav : And also deal with moving cameras.
audio-chunks/chunk77.wav : here's a use case for computer vision
audio-chunks/chunk77.wav : Here's a use case for computer vision.
audio-chunks/chunk78.wav : building on detection and tracking you can analyze
Shopper behavior in your retail store by studying the path each person follows
audio-chunks/chunk78.wav : Building on detection and tracking you can analyze
shopper behavior in your retail store by studying the path each person follows.
audio-chunks/chunk79.wav : if you use face analysis you can understand other
details about shoppers
audio-chunks/chunk79.wav : If you use face analysis you can understand other
details about shoppers.
audio-chunks/chunk80.wav : such
audio-chunks/chunk80.wav : Such.
audio-chunks/chunk81.wav : average age ranges gender distribution and expressed
emotions without identifying them
audio-chunks/chunk81.wav : Average age ranges gender distribution and expressed
emotions without identifying them.
audio-chunks/chunk82.wav : here's another computer vision use case
audio-chunks/chunk82.wav : Here's another computer vision use case.
audio-chunks/chunk83.wav : you can also analyze images to identify actions using
the motion in the video
audio-chunks/chunk83.wav : You can also analyze images to identify actions using
the motion in the video.
audio-chunks/chunk84.wav : for example activities such as delivering a package
or dancing
audio-chunks/chunk84.wav : For example activities such as delivering a package
or dancing.
audio-chunks/chunk85.wav : looking at this image of a baseball player
audio-chunks/chunk85.wav : Looking at this image of a baseball player.
audio-chunks/chunk86.wav : some examples from the image could include capturing
the Batters accuracy
audio-chunks/chunk86.wav : Some examples from the image could include capturing
the batters accuracy.
audio-chunks/chunk87.wav : the pitchers pitching style
audio-chunks/chunk87.wav : The pitchers pitching style.
audio-chunks/chunk88.wav : the type of pitch
audio-chunks/chunk88.wav : The type of pitch.
audio-chunks/chunk89.wav : slow ball slider and others
audio-chunks/chunk89.wav : Slow ball slider and others.
audio-chunks/chunk90.wav : the Inn
audio-chunks/chunk90.wav : The inn.
audio-chunks/chunk91.wav : and the Batters performance versus the specific
pitcher
audio-chunks/chunk91.wav : And the batters performance versus the specific
pitcher.
audio-chunks/chunk92.wav : managers could use all that data to coach players on
how to improve their performance
```

```
audio-chunks/chunk92.wav : Managers could use all that data to coach players on
how to improve their performance.
audio-chunks/chunk93.wav : and they do
audio-chunks/chunk93.wav : And they do.
audio-chunks/chunk94.wav : coaches can also use the data during the game to make
game time decisions
audio-chunks/chunk94.wav : Coaches can also use the data during the game to make
game time decisions.
audio-chunks/chunk95.wav : say you want to initiate various actions based on the
speed of the baseball leaving the
audio-chunks/chunk95.wav : Say you want to initiate various actions based on the
speed of the baseball leaving the.
audio-chunks/chunk96.wav : and its trajectory
audio-chunks/chunk96.wav : And its trajectory.
audio-chunks/chunk97.wav : a hit that's calculated by an ml model could lead to
an audio or visual warning about a possible foul ball into the crowd
audio-chunks/chunk97.wav : A hit that's calculated by an ml model could lead to
an audio or visual warning about a possible foul ball into the crowd.
audio-chunks/chunk98.wav : or it could result in a preemptive alarm that a hit
has a high probability of being a home run
audio-chunks/chunk98.wav : Or it could result in a preemptive alarm that a hit
has a high probability of being a home run.
audio-chunks/chunk99.wav : this means that events following a home run could be
both well-timed and automated
audio-chunks/chunk99.wav : This means that events following a home run could be
both well-timed and automated.
audio-chunks/chunk100.wav : such as playing music or setting off fireworks when
the Home Run is hit by the home team
audio-chunks/chunk100.wav : Such as playing music or setting off fireworks when
the home run is hit by the home team.
audio-chunks/chunk101.wav : to wrap up this section
audio-chunks/chunk101.wav : To wrap up this section.
audio-chunks/chunk102.wav : here are some key takeaways for this section
audio-chunks/chunk102.wav : Here are some key takeaways for this section.
Error:
audio-chunks/chunk104.wav : we covered how computer vision is the automated
extraction of information from images
audio-chunks/chunk104.wav : We covered how computer vision is the automated
extraction of information from images.
audio-chunks/chunk105.wav : you can divide computer vision into two distinct
areas
audio-chunks/chunk105.wav : You can divide computer vision into two distinct
areas.
audio-chunks/chunk106.wav : image analysis and video analysis
audio-chunks/chunk106.wav : Image analysis and video analysis.
audio-chunks/chunk107.wav : image analysis includes object classification
detection and segmentation
audio-chunks/chunk107.wav : Image analysis includes object classification
```

```
detection and segmentation.
audio-chunks/chunk108.wav : video analysis includes instance tracking
audio-chunks/chunk108.wav : Video analysis includes instance tracking.
audio-chunks/chunk109.wav : action recognition and motion estimation
audio-chunks/chunk109.wav : Action recognition and motion estimation.
audio-chunks/chunk110.wav : thanks for watching
audio-chunks/chunk110.wav : Thanks for watching.
audio-chunks/chunk111.wav : we'll see you in the next video
audio-chunks/chunk111.wav : We'll see you in the next video.
audio-chunks/chunk1.wav : hi and welcome back to module 3
audio-chunks/chunk1.wav : Hi and welcome back to module 3.
audio-chunks/chunk2.wav : this is section one and we're going to take a look at
some of the data sets will use in this module
audio-chunks/chunk2.wav : This is section one and we're going to take a look at
some of the data sets will use in this module.
audio-chunks/chunk3.wav : we'll also look at guidance for how to formulate a
business problem
audio-chunks/chunk3.wav : We'll also look at guidance for how to formulate a
business problem.
audio-chunks/chunk4.wav : before we get started here's a reminder of the machine
learning pipeline we looked at in the previous module
audio-chunks/chunk4.wav : Before we get started here's a reminder of the machine
learning pipeline we looked at in the previous module.
audio-chunks/chunk5.wav : and how that maps to the sections in this module
audio-chunks/chunk5.wav : And how that maps to the sections in this module.
audio-chunks/chunk6.wav : this section Section 1
audio-chunks/chunk6.wav : This section section 1.
audio-chunks/chunk7.wav : will cover how to formulate a problem
audio-chunks/chunk7.wav : Will cover how to formulate a problem.
audio-chunks/chunk8.wav : it will also cover the data sets we'll use throughout
this module
audio-chunks/chunk8.wav : It will also cover the data sets we'll use throughout
this module.
audio-chunks/chunk9.wav : section 2 we'll discuss how to obtain and secure data
for your machine learning activities
audio-chunks/chunk9.wav : Section 2 we'll discuss how to obtain and secure data
for your machine learning activities.
audio-chunks/chunk10.wav : in Section 3 will show you tools and techniques for
gaining and understanding of your data
audio-chunks/chunk10.wav : In section 3 will show you tools and techniques for
gaining and understanding of your data.
audio-chunks/chunk11.wav : then in section 4 we'll look at pre-processing your
data so it's ready to train a model
audio-chunks/chunk11.wav : Then in section 4 we'll look at pre-processing your
data so it's ready to train a model.
audio-chunks/chunk12.wav : Section 5 will cover selecting and training an
appropriate machine learning model
audio-chunks/chunk12.wav : Section 5 will cover selecting and training an
```

```
appropriate machine learning model.
audio-chunks/chunk13.wav : section
audio-chunks/chunk13.wav : Section.
audio-chunks/chunk14.wav : will show you how to deploy a model so you can make a
prediction
audio-chunks/chunk14.wav : Will show you how to deploy a model so you can make a
prediction.
audio-chunks/chunk15.wav : section 7 will examine the process of evaluating the
performance of a machine learning model
audio-chunks/chunk15.wav : Section 7 will examine the process of evaluating the
performance of a machine learning model.
audio-chunks/chunk16.wav : and finally in Section 8 will look at tuning the
model
audio-chunks/chunk16.wav : And finally in section 8 will look at tuning the
model.
audio-chunks/chunk17.wav : the machine learning pipeline is an iterative process
audio-chunks/chunk17.wav : The machine learning pipeline is an iterative
process.
audio-chunks/chunk18.wav : when you work on a real-world problem
audio-chunks/chunk18.wav : When you work on a real-world problem.
audio-chunks/chunk19.wav : you might find yourself iterating many times before
you arrive at a solution that meets your businesses needs
audio-chunks/chunk19.wav : You might find yourself iterating many times before
you arrive at a solution that meets your businesses needs.
audio-chunks/chunk20.wav : in this first section
audio-chunks/chunk20.wav : In this first section.
audio-chunks/chunk21.wav : will examine how to think about turning a business
requirement into a machine learning problem
audio-chunks/chunk21.wav : Will examine how to think about turning a business
requirement into a machine learning problem.
audio-chunks/chunk22.wav : the first step in this phase
audio-chunks/chunk22.wav : The first step in this phase.
audio-chunks/chunk23.wav : is to Simply Define the problem you want to solve
audio-chunks/chunk23.wav : Is to simply define the problem you want to solve.
audio-chunks/chunk24.wav : and the goal you want to reach
audio-chunks/chunk24.wav : And the goal you want to reach.
audio-chunks/chunk25.wav : understanding the business goal is key because you'll
use it to measure the performance of your solution
audio-chunks/chunk25.wav : Understanding the business goal is key because you'll
use it to measure the performance of your solution.
audio-chunks/chunk26.wav : It's Not Unusual to solidify the business problem
before you can begin targeting a solution
audio-chunks/chunk26.wav : It's not unusual to solidify the business problem
before you can begin targeting a solution.
audio-chunks/chunk27.wav : there are a lot of other questions you can ask to
develop a good understanding of the problem
audio-chunks/chunk27.wav : There are a lot of other questions you can ask to
develop a good understanding of the problem.
```

```
audio-chunks/chunk28.wav : with more information about the problem
audio-chunks/chunk28.wav : With more information about the problem.
audio-chunks/chunk29.wav : you can begin Framing and approach
audio-chunks/chunk29.wav : You can begin framing and approach.
Error:
audio-chunks/chunk31.wav : can the problem even be solved by Machine learning
audio-chunks/chunk31.wav : Can the problem even be solved by machine learning.
audio-chunks/chunk32.wav : or would a traditional approach make more sense
audio-chunks/chunk32.wav : Or would a traditional approach make more sense.
audio-chunks/chunk33.wav : is this a supervised or unsupervised machine learning
problem
audio-chunks/chunk33.wav : Is this a supervised or unsupervised machine learning
problem.
audio-chunks/chunk34.wav : do you have labeled data to train a supervised model
audio-chunks/chunk34.wav : Do you have labeled data to train a supervised model.
audio-chunks/chunk35.wav : there are many questions you could ask yourself and
The Business
audio-chunks/chunk35.wav : There are many questions you could ask yourself and
the business.
audio-chunks/chunk36.wav : ultimately you should try to validate the use of
machine learning
audio-chunks/chunk36.wav : Ultimately you should try to validate the use of
machine learning.
audio-chunks/chunk37.wav : and you should make sure you have access to the right
people and data
audio-chunks/chunk37.wav : And you should make sure you have access to the right
people and data.
audio-chunks/chunk38.wav : you should also try to come up with the simplest
solution to the problem
audio-chunks/chunk38.wav : You should also try to come up with the simplest
solution to the problem.
audio-chunks/chunk39.wav : here's an example
audio-chunks/chunk39.wav : Here's an example.
audio-chunks/chunk40.wav : you want to identify fraudulent credit card
transactions so you can stop the transaction before it processes
audio-chunks/chunk40.wav : You want to identify fraudulent credit card
transactions so you can stop the transaction before it processes.
audio-chunks/chunk41.wav : that's your problem
audio-chunks/chunk41.wav : That's your problem.
audio-chunks/chunk42.wav : now what's the business goal or outcome driving this
problem statement
audio-chunks/chunk42.wav : Now what's the business goal or outcome driving this
problem statement.
audio-chunks/chunk43.wav : in this case
audio-chunks/chunk43.wav : In this case.
audio-chunks/chunk44.wav : say the intended outcome is a reduction in the number
of customers who end their membership to the credit card as a result of a
fraudulent transaction
```

```
audio-chunks/chunk44.wav : Say the intended outcome is a reduction in the number
of customers who end their membership to the credit card as a result of a
fraudulent transaction.
audio-chunks/chunk45.wav : from a business perspective
audio-chunks/chunk45.wav : From a business perspective.
audio-chunks/chunk46.wav : how do you define success given this problem in the
desired outcome
audio-chunks/chunk46.wav : How do you define success given this problem in the
desired outcome.
audio-chunks/chunk47.wav : this is the stage where you need to move from
qualitative statements
audio-chunks/chunk47.wav : This is the stage where you need to move from
qualitative statements.
audio-chunks/chunk48.wav : to quantitative statements that can be easily
measured
audio-chunks/chunk48.wav : To quantitative statements that can be easily
measured.
audio-chunks/chunk49.wav : continue with the example
audio-chunks/chunk49.wav : Continue with the example.
audio-chunks/chunk50.wav : a metric you could use to define success with this
problem might be
audio-chunks/chunk50.wav : A metric you could use to define success with this
problem might be.
audio-chunks/chunk51.wav : a 10% reduction in the number of customers who file
claims for fraudulent transactions within a 6-month period
audio-chunks/chunk51.wav : A 10% reduction in the number of customers who file
claims for fraudulent transactions within a 6-month period.
audio-chunks/chunk52.wav : so now you define the business side of your problem
audio-chunks/chunk52.wav : So now you define the business side of your problem.
audio-chunks/chunk53.wav : it's time to start thinking about this in terms of
your machine learning model itself
audio-chunks/chunk53.wav : It's time to start thinking about this in terms of
your machine learning model itself.
audio-chunks/chunk54.wav : what's the actual output you want to see from your
model
audio-chunks/chunk54.wav : What's the actual output you want to see from your
model.
audio-chunks/chunk55.wav : you want to be specific here
audio-chunks/chunk55.wav : You want to be specific here.
audio-chunks/chunk56.wav : it should be a statement that reflects what an ml
model could actually output
audio-chunks/chunk56.wav : It should be a statement that reflects what an ml
model could actually output.
audio-chunks/chunk57.wav : an example might be
audio-chunks/chunk57.wav : An example might be.
audio-chunks/chunk58.wav : the model will output whether or not a credit card
transaction is fraudulent
audio-chunks/chunk58.wav : The model will output whether or not a credit card
```

```
transaction is fraudulent.
audio-chunks/chunk59.wav : or not fraudulent
audio-chunks/chunk59.wav : Or not fraudulent.
audio-chunks/chunk60.wav : now that you know what you want your ml model to
actually achieve
audio-chunks/chunk60.wav : Now that you know what you want your ml model to
actually achieve.
audio-chunks/chunk61.wav : you can use this information to determine the type of
ml you're working with
audio-chunks/chunk61.wav : You can use this information to determine the type of
ml you're working with.
audio-chunks/chunk62.wav : if you have historical data where customers filed
reports for fraud transactions
audio-chunks/chunk62.wav : If you have historical data where customers filed
reports for fraud transactions.
audio-chunks/chunk63.wav : you can use this data for your machine learning
purpose
audio-chunks/chunk63.wav : You can use this data for your machine learning
purpose.
audio-chunks/chunk64.wav : this historical data falls under the supervised
learning approach
audio-chunks/chunk64.wav : This historical data falls under the supervised
learning approach.
audio-chunks/chunk65.wav : were the labels are already defined
audio-chunks/chunk65.wav : Were the labels are already defined.
audio-chunks/chunk66.wav : recall from earlier in this course that supervised ml
types are categorized into two groups
audio-chunks/chunk66.wav : Recall from earlier in this course that supervised ml
types are categorized into two groups.
audio-chunks/chunk67.wav : classification and regression
audio-chunks/chunk67.wav : Classification and regression.
audio-chunks/chunk68.wav : in the credit card example
audio-chunks/chunk68.wav : In the credit card example.
audio-chunks/chunk69.wav : the desired output of categorizing a transaction is
fraud or not fraud
audio-chunks/chunk69.wav : The desired output of categorizing a transaction is
fraud or not fraud.
audio-chunks/chunk70.wav : so you can see that you're dealing with a binary
classification problem
audio-chunks/chunk70.wav : So you can see that you're dealing with a binary
classification problem.
audio-chunks/chunk71.wav : throughout this module you'll see several data sets
being used
audio-chunks/chunk71.wav : Throughout this module you'll see several data sets
being used.
audio-chunks/chunk72.wav : you can access these data sets and many more from the
UC Irvine machine learning Repository
audio-chunks/chunk72.wav : You can access these data sets and many more from the
```

```
uc irvine machine learning repository.
audio-chunks/chunk73.wav : the first data set contains numerical information
about the composition of wine along with the quality of the wine
audio-chunks/chunk73.wav : The first data set contains numerical information
about the composition of wine along with the quality of the wine.
audio-chunks/chunk74.wav : the question you might want to ask on this data set
is
audio-chunks/chunk74.wav : The question you might want to ask on this data set
is.
audio-chunks/chunk75.wav : based on the composition of the wine
audio-chunks/chunk75.wav : Based on the composition of the wine.
audio-chunks/chunk76.wav : could we predict the quality and therefore the price
audio-chunks/chunk76.wav : Could we predict the quality and therefore the price.
audio-chunks/chunk77.wav : in addition to that question will also use this data
set to view statistics
audio-chunks/chunk77.wav : In addition to that question will also use this data
set to view statistics.
audio-chunks/chunk78.wav : deal with outliers and scale numerical data
audio-chunks/chunk78.wav : Deal with outliers and scale numerical data.
audio-chunks/chunk79.wav : the second data set is a car evaluation database
audio-chunks/chunk79.wav : The second data set is a car evaluation database.
audio-chunks/chunk80.wav : this data set is heavily text-based
audio-chunks/chunk80.wav : This data set is heavily text-based.
audio-chunks/chunk81.wav : this enables you to explore the encoding categorical
data which converts the text values into numbers that can be processed by
Machine learning
audio-chunks/chunk81.wav : This enables you to explore the encoding categorical
data which converts the text values into numbers that can be processed by
machine learning.
audio-chunks/chunk82.wav : the third data set is a biomedical data set
audio-chunks/chunk82.wav : The third data set is a biomedical data set.
audio-chunks/chunk83.wav : would you also use in the labs
audio-chunks/chunk83.wav : Would you also use in the labs.
audio-chunks/chunk84.wav : the question to answer for this data set is
audio-chunks/chunk84.wav : The question to answer for this data set is.
audio-chunks/chunk85.wav : based on the biomechanical features
audio-chunks/chunk85.wav : Based on the biomechanical features.
audio-chunks/chunk86.wav : can you predict if a patient has an abnormality
audio-chunks/chunk86.wav : Can you predict if a patient has an abnormality.
audio-chunks/chunk87.wav : this data set will take you through the entire end-
to-end process
audio-chunks/chunk87.wav : This data set will take you through the entire end-
to-end process.
audio-chunks/chunk88.wav : you'll end with a trained model that's been tuned and
that you can use to make a prediction
audio-chunks/chunk88.wav : You'll end with a trained model that's been tuned and
that you can use to make a prediction.
audio-chunks/chunk89.wav : in this section we looked at how business problems
```

```
need to be converted into an ml problem
audio-chunks/chunk89.wav : In this section we looked at how business problems
need to be converted into an ml problem.
audio-chunks/chunk90.wav : we also looked at some of the key questions to ask
audio-chunks/chunk90.wav : We also looked at some of the key questions to ask.
audio-chunks/chunk91.wav : which are what is defining success
audio-chunks/chunk91.wav : Which are what is defining success.
audio-chunks/chunk92.wav : can you measure the outcome or impact if your
solution is implemented
audio-chunks/chunk92.wav : Can you measure the outcome or impact if your
solution is implemented.
audio-chunks/chunk93.wav : most business problems fall into one of two
categories
audio-chunks/chunk93.wav : Most business problems fall into one of two
categories.
audio-chunks/chunk94.wav : the first category is classification
audio-chunks/chunk94.wav : The first category is classification.
audio-chunks/chunk95.wav : which can be binary or multiclass
audio-chunks/chunk95.wav : Which can be binary or multiclass.
audio-chunks/chunk96.wav : ask yourself
audio-chunks/chunk96.wav : Ask yourself.
audio-chunks/chunk97.wav : does the Target belong to a class
audio-chunks/chunk97.wav : Does the target belong to a class.
audio-chunks/chunk98.wav : and the second categories regression
audio-chunks/chunk98.wav : And the second categories regression.
audio-chunks/chunk99.wav : for this ask yourself
audio-chunks/chunk99.wav : For this ask yourself.
audio-chunks/chunk100.wav : can you predict a numerical value
audio-chunks/chunk100.wav : Can you predict a numerical value.
audio-chunks/chunk101.wav : that's it for Section 1 we'll see you in the next
video
audio-chunks/chunk101.wav : That's it for section 1 we'll see you in the next
video.
audio-chunks/chunk1.wav : welcome back
audio-chunks/chunk1.wav : Welcome back.
audio-chunks/chunk2.wav : in this section we'll review five manage machine
learning Services you can use for various use cases
audio-chunks/chunk2.wav : In this section we'll review five manage machine
learning services you can use for various use cases.
audio-chunks/chunk3.wav : these Services simplify the process of creating a
machine learning application
audio-chunks/chunk3.wav : These services simplify the process of creating a
machine learning application.
audio-chunks/chunk4.wav : will start
audio-chunks/chunk4.wav : Will start.
audio-chunks/chunk5.wav : by looking at Amazon transcribe
audio-chunks/chunk5.wav : By looking at amazon transcribe.
audio-chunks/chunk6.wav : you can use Amazon transcribe
```

```
audio-chunks/chunk6.wav : You can use amazon transcribe.
audio-chunks/chunk7.wav : to recognize speech in audio files and produce a
transcription
audio-chunks/chunk7.wav : To recognize speech in audio files and produce a
transcription.
audio-chunks/chunk8.wav : it can recognize specific voices in an audio file
audio-chunks/chunk8.wav : It can recognize specific voices in an audio file.
audio-chunks/chunk9.wav : and you can create a customized vocabulary for terms
that are specialized for a particular domain
audio-chunks/chunk9.wav : And you can create a customized vocabulary for terms
that are specialized for a particular domain.
audio-chunks/chunk10.wav : you can also add a transcription service to your
applications by integrating with webs
audio-chunks/chunk10.wav : You can also add a transcription service to your
applications by integrating with webs.
audio-chunks/chunk11.wav : an Internet Protocol you can use for two-way
communication between an application and Amazon transcript
audio-chunks/chunk11.wav : An internet protocol you can use for two-way
communication between an application and amazon transcript.
audio-chunks/chunk12.wav : here are some of the more common use cases for Amazon
transcribe
audio-chunks/chunk12.wav : Here are some of the more common use cases for amazon
transcribe.
Error:
audio-chunks/chunk14.wav : medical professionals can record their notes
audio-chunks/chunk14.wav : Medical professionals can record their notes.
audio-chunks/chunk15.wav : an Amazon transcribe can capture their spoken notes
as text
audio-chunks/chunk15.wav : An amazon transcribe can capture their spoken notes
as text.
audio-chunks/chunk16.wav : also video production organizations can generate
subtitles automatically from video
audio-chunks/chunk16.wav : Also video production organizations can generate
subtitles automatically from video.
audio-chunks/chunk17.wav : this could also be done in real time for a live feed
to add closed captioning
audio-chunks/chunk17.wav : This could also be done in real time for a live feed
to add closed captioning.
audio-chunks/chunk18.wav : media companies can use Amazon transcribe to capture
and label content
audio-chunks/chunk18.wav : Media companies can use amazon transcribe to capture
and label content.
audio-chunks/chunk19.wav : they can then feed the content into Amazon comprehend
for further analysis
audio-chunks/chunk19.wav : They can then feed the content into amazon comprehend
for further analysis.
audio-chunks/chunk20.wav : last
audio-chunks/chunk20.wav : Last.
```

```
audio-chunks/chunk21.wav : companies can record customer service or sales calls
and transcribe them
audio-chunks/chunk21.wav : Companies can record customer service or sales calls
and transcribe them.
audio-chunks/chunk22.wav : they can analyze the results for training or for
strategic opportunities
audio-chunks/chunk22.wav : They can analyze the results for training or for
strategic opportunities.
audio-chunks/chunk23.wav : Amazon Polly can convert text into lifelike speech
audio-chunks/chunk23.wav : Amazon polly can convert text into lifelike speech.
audio-chunks/chunk24.wav : you can input either plain text files
audio-chunks/chunk24.wav : You can input either plain text files.
audio-chunks/chunk25.wav : or a file that's formatted in speech synthesis markup
language or ssml
audio-chunks/chunk25.wav : Or a file that's formatted in speech synthesis markup
language or ssml.
audio-chunks/chunk26.wav : ssml is a markup language used to provide special
instructions for how speech to sound
audio-chunks/chunk26.wav : Ssml is a markup language used to provide special
instructions for how speech to sound.
audio-chunks/chunk27.wav : for example
audio-chunks/chunk27.wav : For example.
audio-chunks/chunk28.wav : if you want to introduce a pause in the flow of
speech
audio-chunks/chunk28.wav : If you want to introduce a pause in the flow of
speech.
audio-chunks/chunk29.wav : you can add an ssml tag that instructs Amazon Polly
to pause between two words
audio-chunks/chunk29.wav : You can add an ssml tag that instructs amazon polly
to pause between two words.
audio-chunks/chunk30.wav : you can also output speech from Amazon Polly to MP3
vorbis and PCM audio stream formats
audio-chunks/chunk30.wav : You can also output speech from amazon polly to mp3
vorbis and pcm audio stream formats.
audio-chunks/chunk31.wav : Amazon Polly is eligible for use with certain
regulated workloads
audio-chunks/chunk31.wav : Amazon polly is eligible for use with certain
regulated workloads.
audio-chunks/chunk32.wav : for example
audio-chunks/chunk32.wav : For example.
audio-chunks/chunk33.wav : it's eligible for use with the US Health Insurance
portability and accountability Act of 1996 or HIPAA
audio-chunks/chunk33.wav : It's eligible for use with the us health insurance
portability and accountability act of 1996 or hipaa.
audio-chunks/chunk34.wav : Amazon Polly is also eligible for use with payment
card industry data security standard or PCI DSS
audio-chunks/chunk34.wav : Amazon polly is also eligible for use with payment
card industry data security standard or pci dss.
```

```
audio-chunks/chunk35.wav : here are some of the more common use cases for Amazon
Polly
audio-chunks/chunk35.wav : Here are some of the more common use cases for amazon
polly.
audio-chunks/chunk36.wav : as a first example
audio-chunks/chunk36.wav : As a first example.
audio-chunks/chunk37.wav : major news companies are using Amazon Polly to
generate vocal content directly from the written stories
audio-chunks/chunk37.wav : Major news companies are using amazon polly to
generate vocal content directly from the written stories.
audio-chunks/chunk38.wav : it's also been embedded in mapping apis
audio-chunks/chunk38.wav : It's also been embedded in mapping apis.
audio-chunks/chunk39.wav : so developers can add voice to their Geo based
applications
audio-chunks/chunk39.wav : So developers can add voice to their geo based
applications.
audio-chunks/chunk40.wav : language training companies have used Amazon Polly to
create systems for learning a new language
audio-chunks/chunk40.wav : Language training companies have used amazon polly to
create systems for learning a new language.
audio-chunks/chunk41.wav : finally
audio-chunks/chunk41.wav : Finally.
audio-chunks/chunk42.wav : animators have used it to add voices to their
characters
audio-chunks/chunk42.wav : Animators have used it to add voices to their
characters.
audio-chunks/chunk43.wav : with Amazon translate you can create multi-language
experiences in your applications
audio-chunks/chunk43.wav : With amazon translate you can create multi-language
experiences in your applications.
audio-chunks/chunk44.wav : you can create systems that read documents in one
language
audio-chunks/chunk44.wav : You can create systems that read documents in one
language.
audio-chunks/chunk45.wav : and then render or store them in another language
audio-chunks/chunk45.wav : And then render or store them in another language.
audio-chunks/chunk46.wav : you can also use it as part of a document analysis
system
audio-chunks/chunk46.wav : You can also use it as part of a document analysis
system.
audio-chunks/chunk47.wav : Amazon translate is fully integrated with other
machine learning services
audio-chunks/chunk47.wav : Amazon translate is fully integrated with other
machine learning services.
audio-chunks/chunk48.wav : such as Amazon comprehend Amazon transcribe and
Amazon Polly
audio-chunks/chunk48.wav : Such as amazon comprehend amazon transcribe and
amazon polly.
```

```
audio-chunks/chunk49.wav : with this integration you can extract named entities
audio-chunks/chunk49.wav : With this integration you can extract named entities.
audio-chunks/chunk50.wav : sentiment and key phrases by integrating with Amazon
comprehend
audio-chunks/chunk50.wav : Sentiment and key phrases by integrating with amazon
comprehend.
audio-chunks/chunk51.wav : create multilingual subtitles with Amazon transcribe
audio-chunks/chunk51.wav : Create multilingual subtitles with amazon transcribe.
audio-chunks/chunk52.wav : and speech translated content with Amazon Polly
audio-chunks/chunk52.wav : And speech translated content with amazon polly.
audio-chunks/chunk53.wav : here are some of the more common use cases for Amazon
Translate
audio-chunks/chunk53.wav : Here are some of the more common use cases for amazon
translate.
audio-chunks/chunk54.wav : the first use case is building International websites
audio-chunks/chunk54.wav : The first use case is building international
websites.
audio-chunks/chunk55.wav : you can use Amazon translate to quickly globalize
your websites
audio-chunks/chunk55.wav : You can use amazon translate to quickly globalize
your websites.
audio-chunks/chunk56.wav : Amazon translate can also be used to develop
multilingual chatbots
audio-chunks/chunk56.wav : Amazon translate can also be used to develop
multilingual chatbots.
audio-chunks/chunk57.wav : chatbots are used to create a more human-like
interface to Applications
audio-chunks/chunk57.wav : Chatbots are used to create a more human-like
interface to applications.
audio-chunks/chunk58.wav : with Amazon Translate
audio-chunks/chunk58.wav : With amazon translate.
audio-chunks/chunk59.wav : you can create a chatbot that speaks multiple
languages
audio-chunks/chunk59.wav : You can create a chatbot that speaks multiple
languages.
audio-chunks/chunk60.wav : another use case is software localization
audio-chunks/chunk60.wav : Another use case is software localization.
audio-chunks/chunk61.wav : localization is a major cost for all software aimed
at a global audience
audio-chunks/chunk61.wav : Localization is a major cost for all software aimed
at a global audience.
audio-chunks/chunk62.wav : Amazon translate can decrease software development
time and significantly reduce costs for localizing software
audio-chunks/chunk62.wav : Amazon translate can decrease software development
time and significantly reduce costs for localizing software.
audio-chunks/chunk63.wav : the final example use case is international media
management
audio-chunks/chunk63.wav : The final example use case is international media
```

```
management.
audio-chunks/chunk64.wav : companies that manage media for Global audience have
used Amazon translate to reduce their costs for localization
audio-chunks/chunk64.wav : Companies that manage media for global audience have
used amazon translate to reduce their costs for localization.
audio-chunks/chunk65.wav : Amazon comprehend implements many of the NLP
techniques that we reviewed earlier in this module
audio-chunks/chunk65.wav : Amazon comprehend implements many of the nlp
techniques that we reviewed earlier in this module.
audio-chunks/chunk66.wav : you can extract key ENT
audio-chunks/chunk66.wav : You can extract key ent.
audio-chunks/chunk67.wav : perform sentiment analysis and tag words with parts
of speech
audio-chunks/chunk67.wav : Perform sentiment analysis and tag words with parts
of speech.
audio-chunks/chunk68.wav : here are some of the more common use cases for Amazon
comprehend
audio-chunks/chunk68.wav : Here are some of the more common use cases for amazon
comprehend.
audio-chunks/chunk69.wav : the first example is analyzing legal and medical
documents
audio-chunks/chunk69.wav : The first example is analyzing legal and medical
documents.
audio-chunks/chunk70.wav : legal insurance and medical organizations have used
Amazon comprehend to perform many of the NLP functions we reviewed in this
module
audio-chunks/chunk70.wav : Legal insurance and medical organizations have used
amazon comprehend to perform many of the nlp functions we reviewed in this
module.
audio-chunks/chunk71.wav : another
audio-chunks/chunk71.wav : Another.
audio-chunks/chunk72.wav : is for large-scale mobile app analysis
audio-chunks/chunk72.wav : Is for large-scale mobile app analysis.
audio-chunks/chunk73.wav : mobile app developers use Amazon comprehend to look
for patterns of usage with their apps so they can design improvements
audio-chunks/chunk73.wav : Mobile app developers use amazon comprehend to look
for patterns of usage with their apps so they can design improvements.
audio-chunks/chunk74.wav : financial fraud detection is another use case for
Amazon comprehend
audio-chunks/chunk74.wav : Financial fraud detection is another use case for
amazon comprehend.
audio-chunks/chunk75.wav : banking financial and other institutions have used it
to examine very large data sets of financial transactions
audio-chunks/chunk75.wav : Banking financial and other institutions have used it
to examine very large data sets of financial transactions.
audio-chunks/chunk76.wav : do you uncover fraud and look for patterns of illegal
transactions
audio-chunks/chunk76.wav : Do you uncover fraud and look for patterns of illegal
```

```
transactions.
audio-chunks/chunk77.wav : finally it can be used for Content management
audio-chunks/chunk77.wav : Finally it can be used for content management.
audio-chunks/chunk78.wav : media and other content companies can use Amazon
comprehend to tag content for analysis and management
audio-chunks/chunk78.wav : Media and other content companies can use amazon
comprehend to tag content for analysis and management.
audio-chunks/chunk79.wav : with Amazon Lex
audio-chunks/chunk79.wav : With amazon lex.
audio-chunks/chunk80.wav : you can add a human language front end to your
applications
audio-chunks/chunk80.wav : You can add a human language front end to your
applications.
audio-chunks/chunk81.wav : Amazon Lex let you use the same conversational engine
that powers Amazon Alexa
audio-chunks/chunk81.wav : Amazon lex let you use the same conversational engine
that powers amazon alexa.
audio-chunks/chunk82.wav : you can automatically increase capacity for your
Amazon Lex Solution by creating AWS Lambda functions that scale on demand
audio-chunks/chunk82.wav : You can automatically increase capacity for your
amazon lex solution by creating aws lambda functions that scale on demand.
audio-chunks/chunk83.wav : you can also store log files of the conversations for
further analysis
audio-chunks/chunk83.wav : You can also store log files of the conversations for
further analysis.
audio-chunks/chunk84.wav : here are some of the more common use cases for Amazon
Lex
audio-chunks/chunk84.wav : Here are some of the more common use cases for amazon
lex.
audio-chunks/chunk85.wav : the first use case
audio-chunks/chunk85.wav : The first use case.
audio-chunks/chunk86.wav : is building front-end interfaces for inventory
management and sales
audio-chunks/chunk86.wav : Is building front-end interfaces for inventory
management and sales.
audio-chunks/chunk87.wav : voice interfaces are becoming more common
audio-chunks/chunk87.wav : Voice interfaces are becoming more common.
audio-chunks/chunk88.wav : companies have used Amazon LAX to add chatbots to
their inventory and sales applications
audio-chunks/chunk88.wav : Companies have used amazon lax to add chatbots to
their inventory and sales applications.
audio-chunks/chunk89.wav : another use for Amazon Lex is creating customer
service interfaces
audio-chunks/chunk89.wav : Another use for amazon lex is creating customer
service interfaces.
audio-chunks/chunk90.wav : human-like voice applications are quickly becoming
the standard for many customer service applications
audio-chunks/chunk90.wav : Human-like voice applications are quickly becoming
```

```
the standard for many customer service applications.
audio-chunks/chunk91.wav : Amazon Alexa can reduce the time it takes to develop
these chatbots and increase their quality
audio-chunks/chunk91.wav : Amazon alexa can reduce the time it takes to develop
these chatbots and increase their quality.
audio-chunks/chunk92.wav : Amazon Alexa can also be used to develop interactive
assist
audio-chunks/chunk92.wav : Amazon alexa can also be used to develop interactive
assist.
audio-chunks/chunk93.wav : by combining Amazon Lex with other ml services
audio-chunks/chunk93.wav : By combining amazon lex with other ml services.
audio-chunks/chunk94.wav : customers are creating more sophisticated assistance
for many different Industries
audio-chunks/chunk94.wav : Customers are creating more sophisticated assistance
for many different industries.
audio-chunks/chunk95.wav : the final example use case
audio-chunks/chunk95.wav : The final example use case.
audio-chunks/chunk96.wav : is querying databases with a human-like language
audio-chunks/chunk96.wav : Is querying databases with a human-like language.
audio-chunks/chunk97.wav : Amazon Lex has been combined with other AWS database
services
audio-chunks/chunk97.wav : Amazon lex has been combined with other aws database
services.
audio-chunks/chunk98.wav : to create sophisticated data analysis applications
audio-chunks/chunk98.wav : To create sophisticated data analysis applications.
audio-chunks/chunk99.wav : with a human like language interface
audio-chunks/chunk99.wav : With a human like language interface.
audio-chunks/chunk100.wav : here are some of the main points you should take
away from this module
audio-chunks/chunk100.wav : Here are some of the main points you should take
away from this module.
Error:
audio-chunks/chunk102.wav : Amazon transcribe can automatically convert spoken
language to text
audio-chunks/chunk102.wav : Amazon transcribe can automatically convert spoken
language to text.
audio-chunks/chunk103.wav : Amazon Polly can convert written text to spoken
language
audio-chunks/chunk103.wav : Amazon polly can convert written text to spoken
language.
audio-chunks/chunk104.wav : Amazon translate can create real-time translation
between languages
audio-chunks/chunk104.wav : Amazon translate can create real-time translation
between languages.
audio-chunks/chunk105.wav : Amazon comprehend
audio-chunks/chunk105.wav : Amazon comprehend.
audio-chunks/chunk106.wav : automate many of the NLP use cases reviewed in this
module
```

```
audio-chunks/chunk106.wav : Automate many of the nlp use cases reviewed in this
module.
audio-chunks/chunk107.wav : and finally
audio-chunks/chunk107.wav : And finally.
audio-chunks/chunk108.wav : Amazon Lex can create a human-like interface to your
applications
audio-chunks/chunk108.wav : Amazon lex can create a human-like interface to your
applications.
audio-chunks/chunk109.wav : thanks for watching we'll see you in the next video
audio-chunks/chunk109.wav : Thanks for watching we'll see you in the next video.
audio-chunks/chunk1.wav : it's now time to summarize some of the main points in
this module
audio-chunks/chunk1.wav : It's now time to summarize some of the main points in
this module.
audio-chunks/chunk2.wav : in this module you learned how to describe the use
cases for computer vision
audio-chunks/chunk2.wav : In this module you learned how to describe the use
cases for computer vision.
audio-chunks/chunk3.wav : describe the Amazon management learning services
available for image and video analysis
audio-chunks/chunk3.wav : Describe the amazon management learning services
available for image and video analysis.
audio-chunks/chunk4.wav : list the steps required to prepare a custom data set
for object detection
audio-chunks/chunk4.wav : List the steps required to prepare a custom data set
for object detection.
audio-chunks/chunk5.wav : describe how Amazon sagemaker ground TR
audio-chunks/chunk5.wav : Describe how amazon sagemaker ground tr.
audio-chunks/chunk6.wav : can be used to prepare a custom data set
audio-chunks/chunk6.wav : Can be used to prepare a custom data set.
audio-chunks/chunk7.wav : and use Amazon recognition to perform facial detection
audio-chunks/chunk7.wav : And use amazon recognition to perform facial
detection.
audio-chunks/chunk8.wav : that concludes this introduction to computer vision
audio-chunks/chunk8.wav : That concludes this introduction to computer vision.
audio-chunks/chunk9.wav : thanks for watching we'll see you again in the next
video
audio-chunks/chunk9.wav : Thanks for watching we'll see you again in the next
video.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : now we'll review how to find correlations in your data
set
audio-chunks/chunk2.wav : Now we'll review how to find correlations in your data
set.
audio-chunks/chunk3.wav : how can you quantify the linear relationship among the
variables you're seeing in a scatter plot
audio-chunks/chunk3.wav : How can you quantify the linear relationship among the
```

```
variables you're seeing in a scatter plot.
audio-chunks/chunk4.wav : a correlation Matrix is a good tool in the situation
audio-chunks/chunk4.wav : A correlation matrix is a good tool in the situation.
audio-chunks/chunk5.wav : it conveys both the strong and weak linear
relationships among numerical variables
audio-chunks/chunk5.wav : It conveys both the strong and weak linear
relationships among numerical variables.
audio-chunks/chunk6.wav : correlation can go as high as one
audio-chunks/chunk6.wav : Correlation can go as high as one.
audio-chunks/chunk7.wav : or as low as -1
audio-chunks/chunk7.wav : Or as low as -1.
audio-chunks/chunk8.wav : when the correlation is one this means those two
numerical features are perfectly correlated with each other
audio-chunks/chunk8.wav : When the correlation is one this means those two
numerical features are perfectly correlated with each other.
audio-chunks/chunk9.wav : it's like saying why is proportional to X
audio-chunks/chunk9.wav : It's like saying why is proportional to x.
audio-chunks/chunk10.wav : when the correlation of those two variables is -1
audio-chunks/chunk10.wav : When the correlation of those two variables is -1.
audio-chunks/chunk11.wav : it's like saying why is proportional to - x
audio-chunks/chunk11.wav : It's like saying why is proportional to - x.
audio-chunks/chunk12.wav : any linear relationship in between can be Quantified
by the correlation
audio-chunks/chunk12.wav : Any linear relationship in between can be quantified
by the correlation.
audio-chunks/chunk13.wav : so if the correlation is 0
audio-chunks/chunk13.wav : So if the correlation is 0.
audio-chunks/chunk14.wav : this means there's no linear relationship
audio-chunks/chunk14.wav : This means there's no linear relationship.
audio-chunks/chunk15.wav : but it doesn't mean that there's no relationship
audio-chunks/chunk15.wav : But it doesn't mean that there's no relationship.
audio-chunks/chunk16.wav : it's just an indication that there's no linear
relationship between those two variables
audio-chunks/chunk16.wav : It's just an indication that there's no linear
relationship between those two variables.
audio-chunks/chunk17.wav : however looking at a number isn't always
straightforward
audio-chunks/chunk17.wav : However looking at a number isn't always
straightforward.
audio-chunks/chunk18.wav : often it's easier to view the numbers when the
represented by colors
audio-chunks/chunk18.wav : Often it's easier to view the numbers when the
represented by colors.
Error:
audio-chunks/chunk20.wav : the highest number one in dark green
audio-chunks/chunk20.wav : The highest number one in dark green.
audio-chunks/chunk21.wav : and minus one is in dark brown
audio-chunks/chunk21.wav : And minus one is in dark brown.
```

```
audio-chunks/chunk22.wav : the color gives you both the positive and negative
directions
audio-chunks/chunk22.wav : The color gives you both the positive and negative
directions.
audio-chunks/chunk23.wav : and it also shows how strong the correlations are
audio-chunks/chunk23.wav : And it also shows how strong the correlations are.
audio-chunks/chunk24.wav : we can use the Seaborn heat map function to show the
correlation Matrix
audio-chunks/chunk24.wav : We can use the seaborn heat map function to show the
correlation matrix.
audio-chunks/chunk25.wav : looking at the chart there's some correlation between
citric acid and fixed acidity
audio-chunks/chunk25.wav : Looking at the chart there's some correlation between
citric acid and fixed acidity.
audio-chunks/chunk26.wav : that would be expected in wine because citric acid
contributes to the acidity of the wine
audio-chunks/chunk26.wav : That would be expected in wine because citric acid
contributes to the acidity of the wine.
audio-chunks/chunk27.wav : however there isn't much correlation between fixed
acidity and pH
audio-chunks/chunk27.wav : However there isn't much correlation between fixed
acidity and ph.
audio-chunks/chunk28.wav : pH is a measurement of the strength of those acids
present
audio-chunks/chunk28.wav : Ph is a measurement of the strength of those acids
present.
audio-chunks/chunk29.wav : but fixed acidity is a measure of the quantity
audio-chunks/chunk29.wav : But fixed acidity is a measure of the quantity.
audio-chunks/chunk30.wav : in this particular data set there doesn't appear to
be a correlation here
audio-chunks/chunk30.wav : In this particular data set there doesn't appear to
be a correlation here.
audio-chunks/chunk31.wav : some key takeaways from this section of the module
include these points
audio-chunks/chunk31.wav : Some key takeaways from this section of the module
include these points.
audio-chunks/chunk32.wav : the first step is to get your data into a format that
can be used easily
audio-chunks/chunk32.wav : The first step is to get your data into a format that
can be used easily.
audio-chunks/chunk33.wav : pandas is a popular python library for working with
data
audio-chunks/chunk33.wav : Pandas is a popular python library for working with
data.
audio-chunks/chunk34.wav : descriptive statistics will help you gain insights
into the data
audio-chunks/chunk34.wav : Descriptive statistics will help you gain insights
into the data.
```

```
audio-chunks/chunk35.wav : you can use visualizations to examine the data set in
more detail
audio-chunks/chunk35.wav : You can use visualizations to examine the data set in
more detail.
audio-chunks/chunk36.wav : that's it for this section we'll see you again in the
next video
audio-chunks/chunk36.wav : That's it for this section we'll see you again in the
next video.
audio-chunks/chunk1.wav : hi and welcome to section 1
audio-chunks/chunk1.wav : Hi and welcome to section 1.
audio-chunks/chunk2.wav : in this section we're going to talk about what machine
learning is
audio-chunks/chunk2.wav : In this section we're going to talk about what machine
learning is.
audio-chunks/chunk3.wav : this course is an introduction to machine learning
audio-chunks/chunk3.wav : This course is an introduction to machine learning.
audio-chunks/chunk4.wav : which is also known as ml
audio-chunks/chunk4.wav : Which is also known as ml.
audio-chunks/chunk5.wav : but first we'll discuss where machine learning fits
into the larger picture
audio-chunks/chunk5.wav : But first we'll discuss where machine learning fits
into the larger picture.
audio-chunks/chunk6.wav : machine learning is a subset of artificial
intelligence or AI
audio-chunks/chunk6.wav : Machine learning is a subset of artificial
intelligence or ai.
audio-chunks/chunk7.wav : this is a broad branch of computer science that's
focused on building machines that can do human tasks
audio-chunks/chunk7.wav : This is a broad branch of computer science that's
focused on building machines that can do human tasks.
audio-chunks/chunk8.wav : deep learning is a subdomain of machine learning
audio-chunks/chunk8.wav : Deep learning is a subdomain of machine learning.
audio-chunks/chunk9.wav : do you understand where these all fit together we'll
discuss each one
audio-chunks/chunk9.wav : Do you understand where these all fit together we'll
discuss each one.
audio-chunks/chunk10.wav : as we just mentioned machine learning is a subset of
a broader computer science field known as artificial intelligence
audio-chunks/chunk10.wav : As we just mentioned machine learning is a subset of
a broader computer science field known as artificial intelligence.
audio-chunks/chunk11.wav : AI focuses on building machines that can perform
tasks a human would typically perform
audio-chunks/chunk11.wav : Ai focuses on building machines that can perform
tasks a human would typically perform.
audio-chunks/chunk12.wav : in contemporary popular culture you've probably seen
AIS in movies television or works of fiction
audio-chunks/chunk12.wav : In contemporary popular culture you've probably seen
ais in movies television or works of fiction.
```

```
audio-chunks/chunk13.wav : for example you might have seen a eyes that control
the world around them
audio-chunks/chunk13.wav : For example you might have seen a eyes that control
the world around them.
audio-chunks/chunk14.wav : or that start acting on their own initiative
audio-chunks/chunk14.wav : Or that start acting on their own initiative.
audio-chunks/chunk15.wav : these AI started as computer agents that perceived
their actions to achieve a specific goal
audio-chunks/chunk15.wav : These ai started as computer agents that perceived
their actions to achieve a specific goal.
audio-chunks/chunk16.wav : so maybe not the outcome that created originally
wished for
audio-chunks/chunk16.wav : So maybe not the outcome that created originally
wished for.
audio-chunks/chunk17.wav : other fictional AIS interact extensively with humans
as helpers or workers
audio-chunks/chunk17.wav : Other fictional ais interact extensively with humans
as helpers or workers.
audio-chunks/chunk18.wav : and they generally do a better job working with
Humanity
audio-chunks/chunk18.wav : And they generally do a better job working with
humanity.
audio-chunks/chunk19.wav : but they're more General in purpose
audio-chunks/chunk19.wav : But they're more general in purpose.
audio-chunks/chunk20.wav : these kinds of AIS are examples of artificial general
intelligence or AGI
audio-chunks/chunk20.wav : These kinds of ais are examples of artificial general
intelligence or agi.
audio-chunks/chunk21.wav : they have the capacity to learn or understand any
task that a human can
audio-chunks/chunk21.wav : They have the capacity to learn or understand any
task that a human can.
audio-chunks/chunk22.wav : AI problems typically span many fields of research
audio-chunks/chunk22.wav : Ai problems typically span many fields of research.
audio-chunks/chunk23.wav : such as natural language processing
audio-chunks/chunk23.wav : Such as natural language processing.
audio-chunks/chunk24.wav : reason
audio-chunks/chunk24.wav : Reason.
audio-chunks/chunk25.wav : knowledge representation
audio-chunks/chunk25.wav : Knowledge representation.
audio-chunks/chunk26.wav : learning
audio-chunks/chunk26.wav : Learning.
audio-chunks/chunk27.wav : perception and physical environment interaction
audio-chunks/chunk27.wav : Perception and physical environment interaction.
audio-chunks/chunk28.wav : Eli isn't yet a reality for us
audio-chunks/chunk28.wav : Eli isn't yet a reality for us.
audio-chunks/chunk29.wav : living in a simulation
audio-chunks/chunk29.wav : Living in a simulation.
```

```
audio-chunks/chunk30.wav : but every year we move closer to it in each of those
areas
audio-chunks/chunk30.wav : But every year we move closer to it in each of those
areas.
audio-chunks/chunk31.wav : you might have also read or seen commentary on the
ethics of creating AI
audio-chunks/chunk31.wav : You might have also read or seen commentary on the
ethics of creating ai.
audio-chunks/chunk32.wav : not all views are positive
audio-chunks/chunk32.wav : Not all views are positive.
audio-chunks/chunk33.wav : malicious fictional AI is that want to destroy
Humanity or use them as power sources
audio-chunks/chunk33.wav : Malicious fictional ai is that want to destroy
humanity or use them as power sources.
audio-chunks/chunk34.wav : or perhaps they're concerned about the risk of mass
unemployment because an intelligent machine could work 24/7 and not need any
breaks
audio-chunks/chunk34.wav : Or perhaps they're concerned about the risk of mass
unemployment because an intelligent machine could work 24/7 and not need any
breaks.
audio-chunks/chunk35.wav : don't worry though we're not going to build the next
Rogue AI in this course
audio-chunks/chunk35.wav : Don't worry though we're not going to build the next
rogue ai in this course.
audio-chunks/chunk36.wav : maybe in the next one
audio-chunks/chunk36.wav : Maybe in the next one.
audio-chunks/chunk37.wav : if you do a search
audio-chunks/chunk37.wav : If you do a search.
audio-chunks/chunk38.wav : many definitions of machine learning
audio-chunks/chunk38.wav : Many definitions of machine learning.
audio-chunks/chunk39.wav : there isn't a universally agreed upon definition
audio-chunks/chunk39.wav : There isn't a universally agreed upon definition.
audio-chunks/chunk40.wav : so we'll start by looking at a couple of definition
audio-chunks/chunk40.wav : So we'll start by looking at a couple of definition.
audio-chunks/chunk41.wav : for example we could say machine learning is the
scientific study of algorithms and statistical models
audio-chunks/chunk41.wav : For example we could say machine learning is the
scientific study of algorithms and statistical models.
audio-chunks/chunk42.wav : to perform a task by using inference instead of
instructions
audio-chunks/chunk42.wav : To perform a task by using inference instead of
instructions.
audio-chunks/chunk43.wav : this isn't a bad starting point
audio-chunks/chunk43.wav : This isn't a bad starting point.
audio-chunks/chunk44.wav : the key Point here is using algorithms and
statistical models instead of instructions
audio-chunks/chunk44.wav : The key point here is using algorithms and
statistical models instead of instructions.
```

```
audio-chunks/chunk45.wav : to help you better understand this
audio-chunks/chunk45.wav : To help you better understand this.
audio-chunks/chunk46.wav : will apply this idea to a concrete example
audio-chunks/chunk46.wav : Will apply this idea to a concrete example.
audio-chunks/chunk47.wav : suppose you need to write an application that
determines if an email message is Spam or not
audio-chunks/chunk47.wav : Suppose you need to write an application that
determines if an email message is spam or not.
audio-chunks/chunk48.wav : without machine learning you'd need to write a
complex series of decision statements using IF and else statements
audio-chunks/chunk48.wav : Without machine learning you'd need to write a
complex series of decision statements using if and else statements.
audio-chunks/chunk49.wav : you'd also need to use words in the subject or body
the number of links and the length of the message to determine if an email
message is spam
audio-chunks/chunk49.wav : You'd also need to use words in the subject or body
the number of links and the length of the message to determine if an email
message is spam.
audio-chunks/chunk50.wav : it would be hard and labor-intensive to build a large
set of rules covering every possibility
audio-chunks/chunk50.wav : It would be hard and labor-intensive to build a large
set of rules covering every possibility.
audio-chunks/chunk51.wav : with machine learning however
audio-chunks/chunk51.wav : With machine learning however.
audio-chunks/chunk52.wav : you could use a list of email messages that were
marked as spam or not spam
audio-chunks/chunk52.wav : You could use a list of email messages that were
marked as spam or not spam.
audio-chunks/chunk53.wav : and train a machine learning model
audio-chunks/chunk53.wav : And train a machine learning model.
audio-chunks/chunk54.wav : the model would then learn which patterns of words
length and other attributes are good indicators of spam messages
audio-chunks/chunk54.wav : The model would then learn which patterns of words
length and other attributes are good indicators of spam messages.
audio-chunks/chunk55.wav : if you presented the model with an email message it
hadn't seen before
audio-chunks/chunk55.wav : If you presented the model with an email message it
hadn't seen before.
audio-chunks/chunk56.wav : the model would perform a prediction to say whether
the message was spam or not spam
audio-chunks/chunk56.wav : The model would perform a prediction to say whether
the message was spam or not spam.
audio-chunks/chunk57.wav : deep learning represents a significant Leap Forward
in the capabilities for artificial intelligence and machine learning
audio-chunks/chunk57.wav : Deep learning represents a significant leap forward
in the capabilities for artificial intelligence and machine learning.
audio-chunks/chunk58.wav : the theory behind deep learning was created from how
the human brain works
```

```
audio-chunks/chunk58.wav : The theory behind deep learning was created from how
the human brain works.
audio-chunks/chunk59.wav : an artificial neural network or an
audio-chunks/chunk59.wav : An artificial neural network or an.
audio-chunks/chunk60.wav : is inspired by the biological neurons found in the
brain
audio-chunks/chunk60.wav : Is inspired by the biological neurons found in the
brain.
audio-chunks/chunk61.wav : although the implementation has become very different
audio-chunks/chunk61.wav : Although the implementation has become very
different.
audio-chunks/chunk62.wav : artificial neurons have one or more inputs and a
single output
audio-chunks/chunk62.wav : Artificial neurons have one or more inputs and a
single output.
audio-chunks/chunk63.wav : these neurons fire or activate their outputs based on
a transformation of the inputs
audio-chunks/chunk63.wav : These neurons fire or activate their outputs based on
a transformation of the inputs.
audio-chunks/chunk64.wav : a neural network is composed of layers of these
artificial neurons
audio-chunks/chunk64.wav : A neural network is composed of layers of these
artificial neurons.
audio-chunks/chunk65.wav : with connections between the layers
audio-chunks/chunk65.wav : With connections between the layers.
audio-chunks/chunk66.wav : there are typically input
audio-chunks/chunk66.wav : There are typically input.
audio-chunks/chunk67.wav : output and hidden layers in the network
audio-chunks/chunk67.wav : Output and hidden layers in the network.
audio-chunks/chunk68.wav : the output of a single neuron connects to the inputs
of all the neurons in the next layer
audio-chunks/chunk68.wav : The output of a single neuron connects to the inputs
of all the neurons in the next layer.
audio-chunks/chunk69.wav : the network is then asked to solve a problem
audio-chunks/chunk69.wav : The network is then asked to solve a problem.
audio-chunks/chunk70.wav : the input layer is populated from the training data
audio-chunks/chunk70.wav : The input layer is populated from the training data.
audio-chunks/chunk71.wav : and then neurons activate throughout the layers until
an answer is presented in the output layer
audio-chunks/chunk71.wav : And then neurons activate throughout the layers until
an answer is presented in the output layer.
audio-chunks/chunk72.wav : the accuracy of the output is then measured
audio-chunks/chunk72.wav : The accuracy of the output is then measured.
audio-chunks/chunk73.wav : if the output doesn't meet your threshold the
training is repeated but with slight changes to the weights of the connections
between neurons
audio-chunks/chunk73.wav : If the output doesn't meet your threshold the
training is repeated but with slight changes to the weights of the connections
```

```
between neurons.
audio-chunks/chunk74.wav : the neural network will do this repeatedly
audio-chunks/chunk74.wav : The neural network will do this repeatedly.
audio-chunks/chunk75.wav : each time it strengthens the connections that lead to
success
audio-chunks/chunk75.wav : Each time it strengthens the connections that lead to
success.
audio-chunks/chunk76.wav : and diminishes the connections that lead to failure
audio-chunks/chunk76.wav : And diminishes the connections that lead to failure.
audio-chunks/chunk77.wav : as you'll see in this course
audio-chunks/chunk77.wav : As you'll see in this course.
audio-chunks/chunk78.wav : machine learning practitioners spend a lot of time
optimizing the ml models
audio-chunks/chunk78.wav : Machine learning practitioners spend a lot of time
optimizing the ml models.
audio-chunks/chunk79.wav : selecting the best data features to train with
audio-chunks/chunk79.wav : Selecting the best data features to train with.
Error:
audio-chunks/chunk81.wav : in contrast
audio-chunks/chunk81.wav : In contrast.
audio-chunks/chunk82.wav : deep learning practitioners spend almost no time on
those tasks
audio-chunks/chunk82.wav : Deep learning practitioners spend almost no time on
those tasks.
audio-chunks/chunk83.wav : instead they spend their time modeling data with
different a&n architectures
audio-chunks/chunk83.wav : Instead they spend their time modeling data with
different a&n architectures.
audio-chunks/chunk84.wav : do the theory for deep learning goes back decades
audio-chunks/chunk84.wav : Do the theory for deep learning goes back decades.
audio-chunks/chunk85.wav : the hardware needed to run deep learning problems
wasn't generally accessible until recently
audio-chunks/chunk85.wav : The hardware needed to run deep learning problems
wasn't generally accessible until recently.
audio-chunks/chunk86.wav : but now that it's available
audio-chunks/chunk86.wav : But now that it's available.
audio-chunks/chunk87.wav : you can use deep learning to address problems that
are more complex than the problems you could have worked on before
audio-chunks/chunk87.wav : You can use deep learning to address problems that
are more complex than the problems you could have worked on before.
audio-chunks/chunk88.wav : mainstream machine learning is a recent occurrence
audio-chunks/chunk88.wav : Mainstream machine learning is a recent occurrence.
audio-chunks/chunk89.wav : rapid advancements in machine and deep learning only
started around the mid-2000s
audio-chunks/chunk89.wav : Rapid advancements in machine and deep learning only
started around the mid-2000s.
audio-chunks/chunk90.wav : this is partly because Moore's law and the rise of
cloud computing resulted in easier access to larger faster and cheaper compute
```

```
and storage capabilities
audio-chunks/chunk90.wav : This is partly because moore's law and the rise of
cloud computing resulted in easier access to larger faster and cheaper compute
and storage capabilities.
audio-chunks/chunk91.wav : you can now rent computing power for a few hours for
a pennies
audio-chunks/chunk91.wav : You can now rent computing power for a few hours for
a pennies.
audio-chunks/chunk92.wav : before this you needed substantial Investments to buy
and operate large-scale compute clusters on your own
audio-chunks/chunk92.wav : Before this you needed substantial investments to buy
and operate large-scale compute clusters on your own.
audio-chunks/chunk93.wav : in 2012 neural network started to be used in the
image net large-scale visual recognition
audio-chunks/chunk93.wav : In 2012 neural network started to be used in the
image net large-scale visual recognition.
audio-chunks/chunk94.wav : a machine learning competition for image recognition
audio-chunks/chunk94.wav : A machine learning competition for image recognition.
audio-chunks/chunk95.wav : the accuracy rate jumped up to about 82% and has been
steadily climbing ever since
audio-chunks/chunk95.wav : The accuracy rate jumped up to about 82% and has been
steadily climbing ever since.
audio-chunks/chunk96.wav : in fact it exceeded human performance in 2015
audio-chunks/chunk96.wav : In fact it exceeded human performance in 2015.
audio-chunks/chunk97.wav : here are some of the key takeaways for this section
audio-chunks/chunk97.wav : Here are some of the key takeaways for this section.
Error:
Error:
audio-chunks/chunk100.wav : also machine learning is a subset of AI it focuses
on using data to train machine learning models so they can make predictions
audio-chunks/chunk100.wav : Also machine learning is a subset of ai it focuses
on using data to train machine learning models so they can make predictions.
audio-chunks/chunk101.wav : deep learning is a technique inspired from human
biology
audio-chunks/chunk101.wav : Deep learning is a technique inspired from human
biology.
audio-chunks/chunk102.wav : it uses layers of artificial neurons to build
networks that solve problems
audio-chunks/chunk102.wav : It uses layers of artificial neurons to build
networks that solve problems.
Error:
audio-chunks/chunk104.wav : advancements in technology
audio-chunks/chunk104.wav : Advancements in technology.
audio-chunks/chunk105.wav : cloud computing and algorithm development have led
to a corresponding advance in machine learning capabilities and applications
audio-chunks/chunk105.wav : Cloud computing and algorithm development have led
to a corresponding advance in machine learning capabilities and applications.
audio-chunks/chunk106.wav : that's it for this section we'll see you in the next
```

```
video
audio-chunks/chunk106.wav : That's it for this section we'll see you in the next
video.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring video analysis by reviewing
how to create the test data set
audio-chunks/chunk2.wav : Will continue exploring video analysis by reviewing
how to create the test data set.
audio-chunks/chunk3.wav : the final step before you train your model is to
identify a test data set
audio-chunks/chunk3.wav : The final step before you train your model is to
identify a test data set.
audio-chunks/chunk4.wav : you will use this test data set to validate and
evaluate the models performance
audio-chunks/chunk4.wav : You will use this test data set to validate and
evaluate the models performance.
audio-chunks/chunk5.wav : you'll do this by performing an inference on the
images in the test data set
audio-chunks/chunk5.wav : You'll do this by performing an inference on the
images in the test data set.
audio-chunks/chunk6.wav : you'll then compare the results
audio-chunks/chunk6.wav : You'll then compare the results.
audio-chunks/chunk7.wav : with the labeling information that's in the training
data set
audio-chunks/chunk7.wav : With the labeling information that's in the training
data set.
audio-chunks/chunk8.wav : you can create your own test data set
audio-chunks/chunk8.wav : You can create your own test data set.
audio-chunks/chunk9.wav : alternatively you can use Amazon recognition custom
labels to split your training data set into two data set
audio-chunks/chunk9.wav : Alternatively you can use amazon recognition custom
labels to split your training data set into two data set.
audio-chunks/chunk10.wav : by using an 80/20 split
audio-chunks/chunk10.wav : By using an 80/20 split.
audio-chunks/chunk11.wav : this split means that 80% of the data is used for
training and 20% is used for testing
audio-chunks/chunk11.wav : This split means that 80% of the data is used for
training and 20% is used for testing.
audio-chunks/chunk12.wav : after you define the training and test data sets
audio-chunks/chunk12.wav : After you define the training and test data sets.
audio-chunks/chunk13.wav : Amazon recognition custom labels
audio-chunks/chunk13.wav : Amazon recognition custom labels.
audio-chunks/chunk14.wav : can automatically train the model for you
audio-chunks/chunk14.wav : Can automatically train the model for you.
audio-chunks/chunk15.wav : the service automatically loads and inspects the data
audio-chunks/chunk15.wav : The service automatically loads and inspects the
data.
```

```
audio-chunks/chunk16.wav : select the correct machine learning algorithms
audio-chunks/chunk16.wav : Select the correct machine learning algorithms.
audio-chunks/chunk17.wav : trains a model
audio-chunks/chunk17.wav : Trains a model.
audio-chunks/chunk18.wav : and provides model performance metrics
audio-chunks/chunk18.wav : And provides model performance metrics.
audio-chunks/chunk19.wav : you're charged for the amount of time a model takes
to train
audio-chunks/chunk19.wav : You're charged for the amount of time a model takes
to train.
audio-chunks/chunk20.wav : a data set that contains more images and labels will
take longer to train
audio-chunks/chunk20.wav : A data set that contains more images and labels will
take longer to train.
audio-chunks/chunk21.wav : when training is complete
audio-chunks/chunk21.wav : When training is complete.
audio-chunks/chunk22.wav : you evaluate the performance of the model
audio-chunks/chunk22.wav : You evaluate the performance of the model.
audio-chunks/chunk23.wav : during testing
audio-chunks/chunk23.wav : During testing.
audio-chunks/chunk24.wav : Amazon recognition custom labels
audio-chunks/chunk24.wav : Amazon recognition custom labels.
audio-chunks/chunk25.wav : predicts if a test image contains a custom label
audio-chunks/chunk25.wav : Predicts if a test image contains a custom label.
audio-chunks/chunk26.wav : the confidence score is a value that quantifies the
certainty of the models prediction
audio-chunks/chunk26.wav : The confidence score is a value that quantifies the
certainty of the models prediction.
audio-chunks/chunk27.wav : because this is a classification problem
audio-chunks/chunk27.wav : Because this is a classification problem.
audio-chunks/chunk28.wav : the results can be mapped to a confusion Matrix
audio-chunks/chunk28.wav : The results can be mapped to a confusion matrix.
audio-chunks/chunk29.wav : with a true positive
audio-chunks/chunk29.wav : With a true positive.
audio-chunks/chunk30.wav : the model correctly predicts the presence of the
custom label in the test image
audio-chunks/chunk30.wav : The model correctly predicts the presence of the
custom label in the test image.
audio-chunks/chunk31.wav : that is the predicted label is also a ground truth
label for that image
audio-chunks/chunk31.wav : That is the predicted label is also a ground truth
label for that image.
audio-chunks/chunk32.wav : for example Amazon recognition custom labels
audio-chunks/chunk32.wav : For example amazon recognition custom labels.
audio-chunks/chunk33.wav : correctly returns a cat label when a cat is present
in an image
audio-chunks/chunk33.wav : Correctly returns a cat label when a cat is present
in an image.
```

```
audio-chunks/chunk34.wav : for a false positive
audio-chunks/chunk34.wav : For a false positive.
audio-chunks/chunk35.wav : the model incorrectly predicts the presence of a
custom label in a test image
audio-chunks/chunk35.wav : The model incorrectly predicts the presence of a
custom label in a test image.
audio-chunks/chunk36.wav : that is the predicted label isn't a ground truth
label for the image
audio-chunks/chunk36.wav : That is the predicted label isn't a ground truth
label for the image.
audio-chunks/chunk37.wav : for example Amazon recognition custom labels
audio-chunks/chunk37.wav : For example amazon recognition custom labels.
audio-chunks/chunk38.wav : returns a cat label
audio-chunks/chunk38.wav : Returns a cat label.
audio-chunks/chunk39.wav : but there's no cat label in the ground Truth for that
image
audio-chunks/chunk39.wav : But there's no cat label in the ground truth for that
image.
audio-chunks/chunk40.wav : play false negative
audio-chunks/chunk40.wav : Play false negative.
audio-chunks/chunk41.wav : the model doesn't predict that a custom label is
present in the image
audio-chunks/chunk41.wav : The model doesn't predict that a custom label is
present in the image.
audio-chunks/chunk42.wav : but the ground Truth for that image includes this
label
audio-chunks/chunk42.wav : But the ground truth for that image includes this
label.
audio-chunks/chunk43.wav : for example
audio-chunks/chunk43.wav : For example.
audio-chunks/chunk44.wav : Amazon recognition custom labels doesn't return a cat
custom label for an image that contains a cat
audio-chunks/chunk44.wav : Amazon recognition custom labels doesn't return a cat
custom label for an image that contains a cat.
audio-chunks/chunk45.wav : with a true negative the model correctly predicts
that a custom label isn't present in the test image
audio-chunks/chunk45.wav : With a true negative the model correctly predicts
that a custom label isn't present in the test image.
audio-chunks/chunk46.wav : for example Amazon recognition custom labels doesn't
return a cat label for an image that doesn't contain a cat
audio-chunks/chunk46.wav : For example amazon recognition custom labels doesn't
return a cat label for an image that doesn't contain a cat.
audio-chunks/chunk47.wav : the console provides access to True positive
audio-chunks/chunk47.wav : The console provides access to true positive.
audio-chunks/chunk48.wav : false positive and false negative values for each
image in your test data set
audio-chunks/chunk48.wav : False positive and false negative values for each
image in your test data set.
```

```
audio-chunks/chunk49.wav : these prediction results are used to calculate the
various metrics for each label
audio-chunks/chunk49.wav : These prediction results are used to calculate the
various metrics for each label.
audio-chunks/chunk50.wav : and an aggregate of metrics for your entire test set
audio-chunks/chunk50.wav : And an aggregate of metrics for your entire test set.
audio-chunks/chunk51.wav : the same definitions apply to predictions that the
model makes at the bounding Box level
audio-chunks/chunk51.wav : The same definitions apply to predictions that the
model makes at the bounding box level.
audio-chunks/chunk52.wav : with bounding boxes
audio-chunks/chunk52.wav : With bounding boxes.
audio-chunks/chunk53.wav : all metrics are calculated over each bounding box in
each test image
audio-chunks/chunk53.wav : All metrics are calculated over each bounding box in
each test image.
audio-chunks/chunk54.wav : regardless of whether the boxes are prediction or
ground truth
audio-chunks/chunk54.wav : Regardless of whether the boxes are prediction or
ground truth.
audio-chunks/chunk55.wav : to help you
audio-chunks/chunk55.wav : To help you.
audio-chunks/chunk56.wav : Amazon recognition custom labels
audio-chunks/chunk56.wav : Amazon recognition custom labels.
audio-chunks/chunk57.wav : provides various metrics
audio-chunks/chunk57.wav : Provides various metrics.
audio-chunks/chunk58.wav : for example you can view summary metrics and
evaluation metrics for each label
audio-chunks/chunk58.wav : For example you can view summary metrics and
evaluation metrics for each label.
audio-chunks/chunk59.wav : it also provides Precision metrics for each label
audio-chunks/chunk59.wav : It also provides precision metrics for each label.
audio-chunks/chunk60.wav : an average Precision metric for the entire test data
set
audio-chunks/chunk60.wav : An average precision metric for the entire test data
set.
audio-chunks/chunk61.wav : Precision is the proportion of positive results that
were correctly classified
audio-chunks/chunk61.wav : Precision is the proportion of positive results that
were correctly classified.
audio-chunks/chunk62.wav : Amazon recognition custom labels
audio-chunks/chunk62.wav : Amazon recognition custom labels.
audio-chunks/chunk63.wav : provides average recall metrics for each label
audio-chunks/chunk63.wav : Provides average recall metrics for each label.
audio-chunks/chunk64.wav : in an average recall metric for the entire test data
set
audio-chunks/chunk64.wav : In an average recall metric for the entire test data
set.
```

```
audio-chunks/chunk65.wav : recall is the fraction of your test set labels that
were correctly classified
audio-chunks/chunk65.wav : Recall is the fraction of your test set labels that
were correctly classified.
audio-chunks/chunk66.wav : using the previous example of cats
audio-chunks/chunk66.wav : Using the previous example of cats.
audio-chunks/chunk67.wav : that would be how many cats were correctly classified
audio-chunks/chunk67.wav : That would be how many cats were correctly
classified.
audio-chunks/chunk68.wav : the service also provides an average model
performance score for each label
audio-chunks/chunk68.wav : The service also provides an average model
performance score for each label.
audio-chunks/chunk69.wav : an average model performance score for the entire
test data set
audio-chunks/chunk69.wav : An average model performance score for the entire
test data set.
audio-chunks/chunk70.wav : the F1 score combines precision and recall together
audio-chunks/chunk70.wav : The f1 score combines precision and recall together.
audio-chunks/chunk71.wav : to give you just one number that quantifies the
overall performance of a particular machine learning algorithm
audio-chunks/chunk71.wav : To give you just one number that quantifies the
overall performance of a particular machine learning algorithm.
audio-chunks/chunk72.wav : you might use the F1 score when you have a class
imbalance
audio-chunks/chunk72.wav : You might use the f1 score when you have a class
imbalance.
audio-chunks/chunk73.wav : but you also want to preserve the equality between
precision and sensitivity
audio-chunks/chunk73.wav : But you also want to preserve the equality between
precision and sensitivity.
audio-chunks/chunk74.wav : a higher value means better model performance for
both recall and precision
audio-chunks/chunk74.wav : A higher value means better model performance for
both recall and precision.
audio-chunks/chunk75.wav : if you're satisfied with the accuracy of your model
you can start using it
audio-chunks/chunk75.wav : If you're satisfied with the accuracy of your model
you can start using it.
audio-chunks/chunk76.wav : that's it for part 3 of this section will see you
again for part 4 we will review how to evaluate and improve your model
audio-chunks/chunk76.wav : That's it for part 3 of this section will see you
again for part 4 we will review how to evaluate and improve your model.
audio-chunks/chunk1.wav : hi welcome back to module 3
audio-chunks/chunk1.wav : Hi welcome back to module 3.
audio-chunks/chunk2.wav : in this section we'll look at how you can evaluate
your model success in predicting results
audio-chunks/chunk2.wav : In this section we'll look at how you can evaluate
```

```
your model success in predicting results.
audio-chunks/chunk3.wav : at this point you've trained your model
audio-chunks/chunk3.wav : At this point you've trained your model.
audio-chunks/chunk4.wav : it's now time to evaluate that model to determine if
it will do a good job predicting the Target on new and future data
audio-chunks/chunk4.wav : It's now time to evaluate that model to determine if
it will do a good job predicting the target on new and future data.
audio-chunks/chunk5.wav : because future instances have unknown Target values
you need to assess how the model will perform on data where you already know the
target answer
audio-chunks/chunk5.wav : Because future instances have unknown target values
you need to assess how the model will perform on data where you already know the
target answer.
audio-chunks/chunk6.wav : you'll then use this assessment as a proxy for
performance on future data
audio-chunks/chunk6.wav : You'll then use this assessment as a proxy for
performance on future data.
audio-chunks/chunk7.wav : this is the reason why you hold out a sample of your
data for evaluating your testing
audio-chunks/chunk7.wav : This is the reason why you hold out a sample of your
data for evaluating your testing.
audio-chunks/chunk8.wav : an important part of this phase involves choosing the
most appropriate metric for your business situation
audio-chunks/chunk8.wav : An important part of this phase involves choosing the
most appropriate metric for your business situation.
audio-chunks/chunk9.wav : think back to the earlier section on problem
formulation
audio-chunks/chunk9.wav : Think back to the earlier section on problem
formulation.
audio-chunks/chunk10.wav : during that phase you define your business problem
and outcome
audio-chunks/chunk10.wav : During that phase you define your business problem
and outcome.
audio-chunks/chunk11.wav : and then you craft a business metric to evaluate
success
audio-chunks/chunk11.wav : And then you craft a business metric to evaluate
success.
audio-chunks/chunk12.wav : the model metric you choose at this phase should be
linked to that business metric as much as possible
audio-chunks/chunk12.wav : The model metric you choose at this phase should be
linked to that business metric as much as possible.
audio-chunks/chunk13.wav : there's often a high correlation between the two
metrics
audio-chunks/chunk13.wav : There's often a high correlation between the two
metrics.
audio-chunks/chunk14.wav : in addition to considering your business problem and
success metric
audio-chunks/chunk14.wav : In addition to considering your business problem and
```

```
success metric.
audio-chunks/chunk15.wav : the type of ml problem you're working with Will
influence the model metric you choose
audio-chunks/chunk15.wav : The type of ml problem you're working with will
influence the model metric you choose.
audio-chunks/chunk16.wav : throughout the rest of this module we'll look at
examples of common metrics used in classification problems
audio-chunks/chunk16.wav : Throughout the rest of this module we'll look at
examples of common metrics used in classification problems.
audio-chunks/chunk17.wav : will also look at common metrics used in regression
problem
audio-chunks/chunk17.wav : Will also look at common metrics used in regression
problem.
audio-chunks/chunk18.wav : we're going to start by considering a simple binary
classification problem
audio-chunks/chunk18.wav : We're going to start by considering a simple binary
classification problem.
audio-chunks/chunk19.wav : here's a specific example
audio-chunks/chunk19.wav : Here's a specific example.
audio-chunks/chunk20.wav : imagine that you have a simple image recognition
model that's labeling data as either cat
audio-chunks/chunk20.wav : Imagine that you have a simple image recognition
model that's labeling data as either cat.
audio-chunks/chunk21.wav : or not cat
audio-chunks/chunk21.wav : Or not cat.
audio-chunks/chunk22.wav : after the model's been trained
audio-chunks/chunk22.wav : After the model's been trained.
audio-chunks/chunk23.wav : you can use the test data set you held back to
perform predictions
audio-chunks/chunk23.wav : You can use the test data set you held back to
perform predictions.
audio-chunks/chunk24.wav : to help examine the performance of the model you can
compare the predicted values with the actual values
audio-chunks/chunk24.wav : To help examine the performance of the model you can
compare the predicted values with the actual values.
audio-chunks/chunk25.wav : if you plot the values into a table like the example
audio-chunks/chunk25.wav : If you plot the values into a table like the example.
audio-chunks/chunk26.wav : you can start getting some insights into how well the
model performed
audio-chunks/chunk26.wav : You can start getting some insights into how well the
model performed.
audio-chunks/chunk27.wav : in a confusion Matrix
audio-chunks/chunk27.wav : In a confusion matrix.
audio-chunks/chunk28.wav : you can get a high-level comparison of how the
predicted classes matched up against the actual classes
audio-chunks/chunk28.wav : You can get a high-level comparison of how the
predicted classes matched up against the actual classes.
audio-chunks/chunk29.wav : if the actual label or classes cat which is
```

```
identified as P for positive
audio-chunks/chunk29.wav : If the actual label or classes cat which is
identified as p for positive.
audio-chunks/chunk30.wav : and the predicted label or class is also cat
audio-chunks/chunk30.wav : And the predicted label or class is also cat.
audio-chunks/chunk31.wav : then you have a true positive
audio-chunks/chunk31.wav : Then you have a true positive.
audio-chunks/chunk32.wav : this is a good outcome for your model
audio-chunks/chunk32.wav : This is a good outcome for your model.
audio-chunks/chunk33.wav : similarly if you have an actual label of not cat
which is identified as n for negative
audio-chunks/chunk33.wav : Similarly if you have an actual label of not cat
which is identified as n for negative.
audio-chunks/chunk34.wav : and the predicted label or classes also not cat
audio-chunks/chunk34.wav : And the predicted label or classes also not cat.
audio-chunks/chunk35.wav : then you have a true negative
audio-chunks/chunk35.wav : Then you have a true negative.
audio-chunks/chunk36.wav : this is also a good outcome for your model
audio-chunks/chunk36.wav : This is also a good outcome for your model.
audio-chunks/chunk37.wav : in both these cases your model predicted the correct
outcome when it use the testing data
audio-chunks/chunk37.wav : In both these cases your model predicted the correct
outcome when it use the testing data.
audio-chunks/chunk38.wav : there are two other possible outcomes and both aren't
considered good outcomes
audio-chunks/chunk38.wav : There are two other possible outcomes and both aren't
considered good outcomes.
audio-chunks/chunk39.wav : the first one is when the actual class is neg
audio-chunks/chunk39.wav : The first one is when the actual class is neg.
Error:
audio-chunks/chunk41.wav : but the predicted class is positive or cat
audio-chunks/chunk41.wav : But the predicted class is positive or cat.
audio-chunks/chunk42.wav : this is called a false positive because the
prediction is positive but incorrect
audio-chunks/chunk42.wav : This is called a false positive because the
prediction is positive but incorrect.
audio-chunks/chunk43.wav : finally there are false negatives
audio-chunks/chunk43.wav : Finally there are false negatives.
audio-chunks/chunk44.wav : these happen when the actual class is positive so you
got cat
audio-chunks/chunk44.wav : These happen when the actual class is positive so you
got cat.
audio-chunks/chunk45.wav : but the predicted class is negative or not cat
audio-chunks/chunk45.wav : But the predicted class is negative or not cat.
audio-chunks/chunk46.wav : that's it for part one of the section
audio-chunks/chunk46.wav : That's it for part one of the section.
audio-chunks/chunk47.wav : we'll see you again for part 2 where we'll review
calculating classification metrics
```

```
audio-chunks/chunk47.wav : We'll see you again for part 2 where we'll review
calculating classification metrics.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : this is Section 5 and we're going to discuss
challenges with machine learning
audio-chunks/chunk2.wav : This is section 5 and we're going to discuss
challenges with machine learning.
audio-chunks/chunk3.wav : you'll come across many challenges in machine learning
audio-chunks/chunk3.wav : You'll come across many challenges in machine
learning.
audio-chunks/chunk4.wav : there are a lot of poor quality and inconsistent data
available
audio-chunks/chunk4.wav : There are a lot of poor quality and inconsistent data
available.
audio-chunks/chunk5.wav : a significant portion of your job will be getting
access to or generating enough good data that's representative of the problem
you want to solve
audio-chunks/chunk5.wav : A significant portion of your job will be getting
access to or generating enough good data that's representative of the problem
you want to solve.
audio-chunks/chunk6.wav : a key issue to watch out for is under or overfitting
the model
audio-chunks/chunk6.wav : A key issue to watch out for is under or overfitting
the model.
audio-chunks/chunk7.wav : it's not all about the data
audio-chunks/chunk7.wav : It's not all about the data.
audio-chunks/chunk8.wav : although it mostly is
audio-chunks/chunk8.wav : Although it mostly is.
audio-chunks/chunk9.wav : do you have data science experience
audio-chunks/chunk9.wav : Do you have data science experience.
audio-chunks/chunk10.wav : is Staffing a team of data scientists cost-effective
audio-chunks/chunk10.wav : Is staffing a team of data scientists cost-effective.
audio-chunks/chunk11.wav : does management support using machine learning
audio-chunks/chunk11.wav : Does management support using machine learning.
audio-chunks/chunk12.wav : what does the business landscape look like
audio-chunks/chunk12.wav : What does the business landscape look like.
audio-chunks/chunk13.wav : are the problems too complex to formulate into a
machine learning problem
audio-chunks/chunk13.wav : Are the problems too complex to formulate into a
machine learning problem.
audio-chunks/chunk14.wav : can the resulting model be explained to the business
audio-chunks/chunk14.wav : Can the resulting model be explained to the business.
audio-chunks/chunk15.wav : if it can't be explained it might not get adopted
audio-chunks/chunk15.wav : If it can't be explained it might not get adopted.
audio-chunks/chunk16.wav : what's the cost of building updating and operating a
machine learning solution
audio-chunks/chunk16.wav : What's the cost of building updating and operating a
```

```
machine learning solution.
audio-chunks/chunk17.wav : finally how does the technology map
audio-chunks/chunk17.wav : Finally how does the technology map.
audio-chunks/chunk18.wav : does the business unit have access to the data that's
needed
audio-chunks/chunk18.wav : Does the business unit have access to the data that's
needed.
audio-chunks/chunk19.wav : can the data be secured to meet any regulatory
requirements
audio-chunks/chunk19.wav : Can the data be secured to meet any regulatory
requirements.
audio-chunks/chunk20.wav : what tools and Frameworks will be used
audio-chunks/chunk20.wav : What tools and frameworks will be used.
audio-chunks/chunk21.wav : how will the solution integrate with other systems
audio-chunks/chunk21.wav : How will the solution integrate with other systems.
audio-chunks/chunk22.wav : these are important questions
audio-chunks/chunk22.wav : These are important questions.
audio-chunks/chunk23.wav : to be successful you'll need to be able to answer and
address them
audio-chunks/chunk23.wav : To be successful you'll need to be able to answer and
address them.
audio-chunks/chunk24.wav : many machine learning problems can be solved today by
using existing models and without substantial machine learning knowledge
audio-chunks/chunk24.wav : Many machine learning problems can be solved today by
using existing models and without substantial machine learning knowledge.
audio-chunks/chunk25.wav : we've already talked about the AWS managed services
for machine learning
audio-chunks/chunk25.wav : We've already talked about the aws managed services
for machine learning.
audio-chunks/chunk26.wav : you can add sophisticated machine learning
capabilities to your applications with only some basic developer skills for
calling apis
audio-chunks/chunk26.wav : You can add sophisticated machine learning
capabilities to your applications with only some basic developer skills for
calling apis.
audio-chunks/chunk27.wav : there are other pre-built models you can use or adapt
audio-chunks/chunk27.wav : There are other pre-built models you can use or
adapt.
audio-chunks/chunk28.wav : one example is Yolo which means you only look once
audio-chunks/chunk28.wav : One example is yolo which means you only look once.
audio-chunks/chunk29.wav : YOLO is a popular computer vision model
audio-chunks/chunk29.wav : Yolo is a popular computer vision model.
audio-chunks/chunk30.wav : in addition to these scenarios
audio-chunks/chunk30.wav : In addition to these scenarios.
audio-chunks/chunk31.wav : you can use the AWS Marketplace if you'd prefer to
buy models and services from independent software vendors instead of developing
your own
audio-chunks/chunk31.wav : You can use the aws marketplace if you'd prefer to
```

```
buy models and services from independent software vendors instead of developing
your own.
audio-chunks/chunk32.wav : here are the key takeaways for this section
audio-chunks/chunk32.wav : Here are the key takeaways for this section.
Error:
audio-chunks/chunk34.wav : you'll face many machine learning challenges
audio-chunks/chunk34.wav : You'll face many machine learning challenges.
audio-chunks/chunk35.wav : the biggest ones that you can directly influence are
related to data
audio-chunks/chunk35.wav : The biggest ones that you can directly influence are
related to data.
audio-chunks/chunk36.wav : you should consider managed services to solve machine
learning problems within the domains they support
audio-chunks/chunk36.wav : You should consider managed services to solve machine
learning problems within the domains they support.
audio-chunks/chunk37.wav : such as using Amazon recognition for computer vision
problems
audio-chunks/chunk37.wav : Such as using amazon recognition for computer vision
problems.
audio-chunks/chunk38.wav : that's it for this section
audio-chunks/chunk38.wav : That's it for this section.
audio-chunks/chunk39.wav : we'll see you in the next video
audio-chunks/chunk39.wav : We'll see you in the next video.
audio-chunks/chunk1.wav : hi and welcome to Amazon Academy machine learning
foundations
audio-chunks/chunk1.wav : Hi and welcome to amazon academy machine learning
foundations.
audio-chunks/chunk2.wav : in this module you'll learn about the course
objectives
audio-chunks/chunk2.wav : In this module you'll learn about the course
objectives.
audio-chunks/chunk3.wav : various job roles in the machine learning domain
audio-chunks/chunk3.wav : Various job roles in the machine learning domain.
audio-chunks/chunk4.wav : and where you can go to learn more about machine
learning
audio-chunks/chunk4.wav : And where you can go to learn more about machine
learning.
audio-chunks/chunk5.wav : after completing this module you should be able to
identify course prerequisites and objectives
audio-chunks/chunk5.wav : After completing this module you should be able to
identify course prerequisites and objectives.
audio-chunks/chunk6.wav : indicate the role of the data scientist in business
audio-chunks/chunk6.wav : Indicate the role of the data scientist in business.
audio-chunks/chunk7.wav : and identify resources for further learning
audio-chunks/chunk7.wav : And identify resources for further learning.
audio-chunks/chunk8.wav : we're now going to look at the prerequisites for
taking this course
audio-chunks/chunk8.wav : We're now going to look at the prerequisites for
```

```
taking this course.
audio-chunks/chunk9.wav : before you take this course
audio-chunks/chunk9.wav : Before you take this course.
audio-chunks/chunk10.wav : we recommend that you first complete AWS Academy
Cloud foundations
audio-chunks/chunk10.wav : We recommend that you first complete aws academy
cloud foundations.
audio-chunks/chunk11.wav : you should also have some general technical knowledge
of it
audio-chunks/chunk11.wav : You should also have some general technical knowledge
of it.
audio-chunks/chunk12.wav : including foundational computer literacy skills like
audio-chunks/chunk12.wav : Including foundational computer literacy skills like.
audio-chunks/chunk13.wav : basic Computer Concepts
audio-chunks/chunk13.wav : Basic computer concepts.
audio-chunks/chunk14.wav : email
audio-chunks/chunk14.wav : Email.
audio-chunks/chunk15.wav : file management
audio-chunks/chunk15.wav : File management.
audio-chunks/chunk16.wav : and a good understanding of the internet
audio-chunks/chunk16.wav : And a good understanding of the internet.
audio-chunks/chunk17.wav : we also recommend that you have intermediate skills
with Python programming
audio-chunks/chunk17.wav : We also recommend that you have intermediate skills
with python programming.
audio-chunks/chunk18.wav : and a general knowledge of Applied statistics
audio-chunks/chunk18.wav : And a general knowledge of applied statistics.
audio-chunks/chunk19.wav : finally General business knowledge is important for
this course
audio-chunks/chunk19.wav : Finally general business knowledge is important for
this course.
audio-chunks/chunk20.wav : this includes insight into how information technology
is used in business
audio-chunks/chunk20.wav : This includes insight into how information technology
is used in business.
audio-chunks/chunk21.wav : it's also important to have business-related skill
set
audio-chunks/chunk21.wav : It's also important to have business-related skill
set.
audio-chunks/chunk22.wav : such as communication skills
audio-chunks/chunk22.wav : Such as communication skills.
audio-chunks/chunk23.wav : leadership skills
audio-chunks/chunk23.wav : Leadership skills.
audio-chunks/chunk24.wav : an orientation towards customer service
audio-chunks/chunk24.wav : An orientation towards customer service.
audio-chunks/chunk25.wav : in this course you'll be introduced to the key
concepts of machine learning its tools and its uses
audio-chunks/chunk25.wav : In this course you'll be introduced to the key
```

```
concepts of machine learning its tools and its uses.
audio-chunks/chunk26.wav : you'll also be introduced to and work with some of
the AWS services for machine learning
audio-chunks/chunk26.wav : You'll also be introduced to and work with some of
the aws services for machine learning.
audio-chunks/chunk27.wav : you learn how to recognize how machine learning and
deep learning are part of artificial intelligence
audio-chunks/chunk27.wav : You learn how to recognize how machine learning and
deep learning are part of artificial intelligence.
audio-chunks/chunk28.wav : describe artificial intelligence and machine learning
terminology
audio-chunks/chunk28.wav : Describe artificial intelligence and machine learning
terminology.
audio-chunks/chunk29.wav : identify how machine learning can be used to solve a
business problem
audio-chunks/chunk29.wav : Identify how machine learning can be used to solve a
business problem.
audio-chunks/chunk30.wav : describe the machine learning process
audio-chunks/chunk30.wav : Describe the machine learning process.
audio-chunks/chunk31.wav : list the tools available to data scientists
audio-chunks/chunk31.wav : List the tools available to data scientists.
audio-chunks/chunk32.wav : and identify when to use machine learning instead of
traditional software development methods
audio-chunks/chunk32.wav : And identify when to use machine learning instead of
traditional software development methods.
audio-chunks/chunk33.wav : as part of this course
audio-chunks/chunk33.wav : As part of this course.
audio-chunks/chunk34.wav : you'll also learn how to implement a machine learning
pipeline
audio-chunks/chunk34.wav : You'll also learn how to implement a machine learning
pipeline.
audio-chunks/chunk35.wav : this includes how to formulate a problem from a
business request
audio-chunks/chunk35.wav : This includes how to formulate a problem from a
business request.
audio-chunks/chunk36.wav : obtain and secure data for machine learning
audio-chunks/chunk36.wav : Obtain and secure data for machine learning.
audio-chunks/chunk37.wav : build a jupyter notebook by using Amazon sagemaker
audio-chunks/chunk37.wav : Build a jupyter notebook by using amazon sagemaker.
audio-chunks/chunk38.wav : outline the process for evaluating data
audio-chunks/chunk38.wav : Outline the process for evaluating data.
audio-chunks/chunk39.wav : explain why data needs to be pre-processed
audio-chunks/chunk39.wav : Explain why data needs to be pre-processed.
audio-chunks/chunk40.wav : and use open source tools to examine and pre-process
data
audio-chunks/chunk40.wav : And use open source tools to examine and pre-process
data.
audio-chunks/chunk41.wav : you will also use Amazon sagemaker to train and host
```

```
a machine learning model
audio-chunks/chunk41.wav : You will also use amazon sagemaker to train and host
a machine learning model.
audio-chunks/chunk42.wav : use cross validation to test the performance of a
machine learning model
audio-chunks/chunk42.wav : Use cross validation to test the performance of a
machine learning model.
audio-chunks/chunk43.wav : use a hosted model for inference
audio-chunks/chunk43.wav : Use a hosted model for inference.
audio-chunks/chunk44.wav : create an Amazon sagemaker hyperparameter tuning job
to optimize a model's effectiveness
audio-chunks/chunk44.wav : Create an amazon sagemaker hyperparameter tuning job
to optimize a model's effectiveness.
audio-chunks/chunk45.wav : and finally how to use managed Amazon machine
learning services
audio-chunks/chunk45.wav : And finally how to use managed amazon machine
learning services.
audio-chunks/chunk46.wav : to solve specific machine learning problems in
forecasting computer vision and natural language processing
audio-chunks/chunk46.wav : To solve specific machine learning problems in
forecasting computer vision and natural language processing.
audio-chunks/chunk47.wav : will now review the course outline
audio-chunks/chunk47.wav : Will now review the course outline.
audio-chunks/chunk48.wav : to achieve the course objectives
audio-chunks/chunk48.wav : To achieve the course objectives.
audio-chunks/chunk49.wav : you complete the following modules
audio-chunks/chunk49.wav : You complete the following modules.
audio-chunks/chunk50.wav : to start in module 2 you'll get an introduction to
machine learning
audio-chunks/chunk50.wav : To start in module 2 you'll get an introduction to
machine learning.
audio-chunks/chunk51.wav : in module 3 you'll learn how to implement a machine
learning Pipeline with Amazon sagemaker
audio-chunks/chunk51.wav : In module 3 you'll learn how to implement a machine
learning pipeline with amazon sagemaker.
audio-chunks/chunk52.wav : modules for 5 and 6
audio-chunks/chunk52.wav : Modules for 5 and 6.
audio-chunks/chunk53.wav : describe how to apply managed Amazon machine learning
services
audio-chunks/chunk53.wav : Describe how to apply managed amazon machine learning
services.
audio-chunks/chunk54.wav : computer vision and natural language processing
audio-chunks/chunk54.wav : Computer vision and natural language processing.
audio-chunks/chunk55.wav : finally module 7 is a summary of the course
audio-chunks/chunk55.wav : Finally module 7 is a summary of the course.
audio-chunks/chunk56.wav : it also includes an overview of steps you can take to
work towards the AWS certified machine learning specialty
audio-chunks/chunk56.wav : It also includes an overview of steps you can take to
```

```
work towards the aws certified machine learning specialty.
audio-chunks/chunk57.wav : the next five slides provide more detail about the
subtopics covered in each module
audio-chunks/chunk57.wav : The next five slides provide more detail about the
subtopics covered in each module.
audio-chunks/chunk58.wav : the purpose of module 2
audio-chunks/chunk58.wav : The purpose of module 2.
audio-chunks/chunk59.wav : is to introduce you to Major concepts for
understanding machine learning
audio-chunks/chunk59.wav : Is to introduce you to major concepts for
understanding machine learning.
audio-chunks/chunk60.wav : Section 1 describes the overall field of machine
learning
audio-chunks/chunk60.wav : Section 1 describes the overall field of machine
learning.
audio-chunks/chunk61.wav : and how machine learning relates to artificial
intelligence
audio-chunks/chunk61.wav : And how machine learning relates to artificial
intelligence.
audio-chunks/chunk62.wav : and deep learning
audio-chunks/chunk62.wav : And deep learning.
audio-chunks/chunk63.wav : in section 2 you'll learn about some of the most
common business problems you can solve with machine learning
audio-chunks/chunk63.wav : In section 2 you'll learn about some of the most
common business problems you can solve with machine learning.
audio-chunks/chunk64.wav : Section 3 describes the general workflow for solving
machine learning problems
audio-chunks/chunk64.wav : Section 3 describes the general workflow for solving
machine learning problems.
audio-chunks/chunk65.wav : you'll also learn some of the more common machine
learning terms
audio-chunks/chunk65.wav : You'll also learn some of the more common machine
learning terms.
audio-chunks/chunk66.wav : in section 4 you'll review some of the commonly used
tools by Machine learning professionals
audio-chunks/chunk66.wav : In section 4 you'll review some of the commonly used
tools by machine learning professionals.
audio-chunks/chunk67.wav : and lastly in Section 5 you'll get an overview of
some of the common challenges you'll face when working with machine learning
problems
audio-chunks/chunk67.wav : And lastly in section 5 you'll get an overview of
some of the common challenges you'll face when working with machine learning
problems.
audio-chunks/chunk68.wav : in module 3 you'll get an introduction to Amazon
sagemaker and how you can use it to implement a machine learning pipeline
audio-chunks/chunk68.wav : In module 3 you'll get an introduction to amazon
sagemaker and how you can use it to implement a machine learning pipeline.
audio-chunks/chunk69.wav : the module focuses on the application of machine
```

```
learning to solve problems with several public domain data sets as examples of
the machine learning pipeline
audio-chunks/chunk69.wav : The module focuses on the application of machine
learning to solve problems with several public domain data sets as examples of
the machine learning pipeline.
audio-chunks/chunk70.wav : Section 1 introduces you to defining business
problems and the data sets will use during this module
audio-chunks/chunk70.wav : Section 1 introduces you to defining business
problems and the data sets will use during this module.
audio-chunks/chunk71.wav : section 2 through 8 describe the phases of the
machine learning pipeline by using computer vision as an example application
audio-chunks/chunk71.wav : Section 2 through 8 describe the phases of the
machine learning pipeline by using computer vision as an example application.
audio-chunks/chunk72.wav : in section 2
audio-chunks/chunk72.wav : In section 2.
audio-chunks/chunk73.wav : you'll learn how to collect and secure data
audio-chunks/chunk73.wav : You'll learn how to collect and secure data.
audio-chunks/chunk74.wav : Section 3 describes different techniques for
evaluating data
audio-chunks/chunk74.wav : Section 3 describes different techniques for
evaluating data.
audio-chunks/chunk75.wav : in section 4 you'll learn about the process of
feature engineering
audio-chunks/chunk75.wav : In section 4 you'll learn about the process of
feature engineering.
audio-chunks/chunk76.wav : Section 5 describe the steps you'll take to train a
model with sagemaker
audio-chunks/chunk76.wav : Section 5 describe the steps you'll take to train a
model with sagemaker.
audio-chunks/chunk77.wav : in section 6
audio-chunks/chunk77.wav : In section 6.
audio-chunks/chunk78.wav : you'll get an overview of the options in sagemaker
for hosting and using a model
audio-chunks/chunk78.wav : You'll get an overview of the options in sagemaker
for hosting and using a model.
audio-chunks/chunk79.wav : finally section 7 and 8 cover how to evaluate and
tune your model with sagemaker
audio-chunks/chunk79.wav : Finally section 7 and 8 cover how to evaluate and
tune your model with sagemaker.
audio-chunks/chunk80.wav : in this module you'll be introduced to using machine
learning to create forecasts based on a Time series data
audio-chunks/chunk80.wav : In this module you'll be introduced to using machine
learning to create forecasts based on a time series data.
audio-chunks/chunk81.wav : in Section 1 you'll be introduced to forecasting and
some of its common applications
audio-chunks/chunk81.wav : In section 1 you'll be introduced to forecasting and
some of its common applications.
audio-chunks/chunk82.wav : section 2 outline some of the pitfalls of using time
```

```
series data to make forecasts
audio-chunks/chunk82.wav : Section 2 outline some of the pitfalls of using time
series data to make forecasts.
audio-chunks/chunk83.wav : finally in Section 3 you'll get an overview of how to
use Amazon forecast
audio-chunks/chunk83.wav : Finally in section 3 you'll get an overview of how to
use amazon forecast.
audio-chunks/chunk84.wav : in this module you'll learn about using machine
learning for computer vision
audio-chunks/chunk84.wav : In this module you'll learn about using machine
learning for computer vision.
audio-chunks/chunk85.wav : Section 1 describes the general problems you can
solve with computer vision
audio-chunks/chunk85.wav : Section 1 describes the general problems you can
solve with computer vision.
audio-chunks/chunk86.wav : in section 2 you'll learn about the process for
analyzing images and videos
audio-chunks/chunk86.wav : In section 2 you'll learn about the process for
analyzing images and videos.
audio-chunks/chunk87.wav : an in Section 3 you'll learn the steps you'll need to
take to prepare data sets for computer vision
audio-chunks/chunk87.wav : An in section 3 you'll learn the steps you'll need to
take to prepare data sets for computer vision.
audio-chunks/chunk88.wav : in this module you'll be introduced to natural
language processing with machine learning
audio-chunks/chunk88.wav : In this module you'll be introduced to natural
language processing with machine learning.
audio-chunks/chunk89.wav : in Section 1 you'll learn about the general set of
problems you can solve with natural language processing
audio-chunks/chunk89.wav : In section 1 you'll learn about the general set of
problems you can solve with natural language processing.
audio-chunks/chunk90.wav : section 2 review some of the managed Amazon machine
learning Services you can use to address natural language processing problems
audio-chunks/chunk90.wav : Section 2 review some of the managed amazon machine
learning services you can use to address natural language processing problems.
audio-chunks/chunk91.wav : these Services include Amazon transcribe
audio-chunks/chunk91.wav : These services include amazon transcribe.
audio-chunks/chunk92.wav : Amazon Translate
audio-chunks/chunk92.wav : Amazon translate.
audio-chunks/chunk93.wav : Amazon Lex
audio-chunks/chunk93.wav : Amazon lex.
audio-chunks/chunk94.wav : Amazon comprehend
audio-chunks/chunk94.wav : Amazon comprehend.
audio-chunks/chunk95.wav : and Amazon Polly
audio-chunks/chunk95.wav : And amazon polly.
audio-chunks/chunk96.wav : module 7 is the final module of the course
audio-chunks/chunk96.wav : Module 7 is the final module of the course.
audio-chunks/chunk97.wav : in this module you'll review what you've learned
```

```
throughout this course
audio-chunks/chunk97.wav : In this module you'll review what you've learned
throughout this course.
audio-chunks/chunk98.wav : you'll also be introduced to the next steps you
should take if you want to achieve the AWS certified machine learning specialty
audio-chunks/chunk98.wav : You'll also be introduced to the next steps you
should take if you want to achieve the aws certified machine learning specialty.
audio-chunks/chunk99.wav : Section 1 of this module summarizes the topics you've
covered in this course
audio-chunks/chunk99.wav : Section 1 of this module summarizes the topics you've
covered in this course.
audio-chunks/chunk100.wav : in section 2 you'll learn more about the AWS
documentation
audio-chunks/chunk100.wav : In section 2 you'll learn more about the aws
documentation.
audio-chunks/chunk101.wav : you'll also review two common Frameworks for
applying AWS services
audio-chunks/chunk101.wav : You'll also review two common frameworks for
applying aws services.
audio-chunks/chunk102.wav : and finally Section 3 describes the steps you should
take if you want to continue working towards the AWS certified machine learning
special
audio-chunks/chunk102.wav : And finally section 3 describes the steps you should
take if you want to continue working towards the aws certified machine learning
special.
audio-chunks/chunk103.wav : in this section you'll learn about some of the more
common job roles for machine learning professionals
audio-chunks/chunk103.wav : In this section you'll learn about some of the more
common job roles for machine learning professionals.
audio-chunks/chunk104.wav : if you're interested in a data scientist role
audio-chunks/chunk104.wav : If you're interested in a data scientist role.
audio-chunks/chunk105.wav : focus on developing analytical statistical and
programming skills
audio-chunks/chunk105.wav : Focus on developing analytical statistical and
programming skills.
audio-chunks/chunk106.wav : as a data scientist you'll use those skills to
collect
audio-chunks/chunk106.wav : As a data scientist you'll use those skills to
collect.
audio-chunks/chunk107.wav : analyze and interpret large data sets
audio-chunks/chunk107.wav : Analyze and interpret large data sets.
audio-chunks/chunk108.wav : some universities now offered degrees in data
science
audio-chunks/chunk108.wav : Some universities now offered degrees in data
science.
audio-chunks/chunk109.wav : but data scientists often have degrees in related
fields like
audio-chunks/chunk109.wav : But data scientists often have degrees in related
```

```
fields like.
audio-chunks/chunk110.wav : statistics
audio-chunks/chunk110.wav : Statistics.
Error:
audio-chunks/chunk112.wav : computer science
audio-chunks/chunk112.wav : Computer science.
audio-chunks/chunk113.wav : or economics
audio-chunks/chunk113.wav : Or economics.
audio-chunks/chunk114.wav : as a data scientist you'll need technical
competencies in statistics
audio-chunks/chunk114.wav : As a data scientist you'll need technical
competencies in statistics.
audio-chunks/chunk115.wav : machine learning
audio-chunks/chunk115.wav : Machine learning.
audio-chunks/chunk116.wav : programming languages
audio-chunks/chunk116.wav : Programming languages.
audio-chunks/chunk117.wav : add data analytics
audio-chunks/chunk117.wav : Add data analytics.
audio-chunks/chunk118.wav : if you'd like to have a career as a machine learning
engineer
audio-chunks/chunk118.wav : If you'd like to have a career as a machine learning
engineer.
audio-chunks/chunk119.wav : the skills you need will be similar to a data
scientist skill set
audio-chunks/chunk119.wav : The skills you need will be similar to a data
scientist skill set.
audio-chunks/chunk120.wav : like data scientists machine learning Engineers also
require technical competencies and statistics and machine learning
audio-chunks/chunk120.wav : Like data scientists machine learning engineers also
require technical competencies and statistics and machine learning.
audio-chunks/chunk121.wav : however you'll focus more on programming skills and
software architecture than analysis and interpretation
audio-chunks/chunk121.wav : However you'll focus more on programming skills and
software architecture than analysis and interpretation.
audio-chunks/chunk122.wav : as a machine learning engineer you apply those
programming and architecture skills to design and develop machine learning
systems
audio-chunks/chunk122.wav : As a machine learning engineer you apply those
programming and architecture skills to design and develop machine learning
systems.
audio-chunks/chunk123.wav : machine learning Engineers often have previous
experience with software development and they rely more heavily on programming
and software engineering than other machine learning roles
audio-chunks/chunk123.wav : Machine learning engineers often have previous
experience with software development and they rely more heavily on programming
and software engineering than other machine learning roles.
audio-chunks/chunk124.wav : you might also be interested in a career in science
where you can apply machine learning technology to your field
```

```
audio-chunks/chunk124.wav : You might also be interested in a career in science
where you can apply machine learning technology to your field.
audio-chunks/chunk125.wav : machine learning is having an impact in everything
from astronomy to zoology
audio-chunks/chunk125.wav : Machine learning is having an impact in everything
from astronomy to zoology.
audio-chunks/chunk126.wav : so there are many different paths open to you
audio-chunks/chunk126.wav : So there are many different paths open to you.
audio-chunks/chunk127.wav : as an applied science researcher your primary focus
will be on the type of science you're working on
audio-chunks/chunk127.wav : As an applied science researcher your primary focus
will be on the type of science you're working on.
audio-chunks/chunk128.wav : you'll need some of the same skills as a data
scientist
audio-chunks/chunk128.wav : You'll need some of the same skills as a data
scientist.
audio-chunks/chunk129.wav : but you'll also need to know how to apply those
skills to your chosen domain
audio-chunks/chunk129.wav : But you'll also need to know how to apply those
skills to your chosen domain.
audio-chunks/chunk130.wav : does applied science Rules also require technical
competencies and statistics and machine learning
audio-chunks/chunk130.wav : Does applied science rules also require technical
competencies and statistics and machine learning.
audio-chunks/chunk131.wav : many software developers are now integrating machine
learning into their applications
audio-chunks/chunk131.wav : Many software developers are now integrating machine
learning into their applications.
audio-chunks/chunk132.wav : if you're interested in a career as a software
developer
audio-chunks/chunk132.wav : If you're interested in a career as a software
developer.
audio-chunks/chunk133.wav : you should also include machine learning technology
in your studies
audio-chunks/chunk133.wav : You should also include machine learning technology
in your studies.
audio-chunks/chunk134.wav : as a machine learning developer
audio-chunks/chunk134.wav : As a machine learning developer.
audio-chunks/chunk135.wav : your primary focus will be software development
skills
audio-chunks/chunk135.wav : Your primary focus will be software development
skills.
audio-chunks/chunk136.wav : but you'll also need some of the same skills as a
data scientist
audio-chunks/chunk136.wav : But you'll also need some of the same skills as a
data scientist.
audio-chunks/chunk137.wav : so make sure you take coursework in statistics and
Applied Mathematics
```

```
audio-chunks/chunk137.wav : So make sure you take coursework in statistics and
applied mathematics.
audio-chunks/chunk138.wav : and here's a final note for this module
audio-chunks/chunk138.wav : And here's a final note for this module.
audio-chunks/chunk139.wav : we recommend reviewing your student guide
audio-chunks/chunk139.wav : We recommend reviewing your student guide.
audio-chunks/chunk140.wav : you'll find links to documentation and other
resources you'll use throughout the course
audio-chunks/chunk140.wav : You'll find links to documentation and other
resources you'll use throughout the course.
audio-chunks/chunk141.wav : that's it for this introduction thanks for watching
audio-chunks/chunk141.wav : That's it for this introduction thanks for watching.
audio-chunks/chunk142.wav : we'll see you in the next video
audio-chunks/chunk142.wav : We'll see you in the next video.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : we're now going to look at a few ways you can collect
and secure data
audio-chunks/chunk2.wav : We're now going to look at a few ways you can collect
and secure data.
audio-chunks/chunk3.wav : in this section we'll explore some of the techniques
and challenges associated with collecting and securing the data that's needed
for machine learning
audio-chunks/chunk3.wav : In this section we'll explore some of the techniques
and challenges associated with collecting and securing the data that's needed
for machine learning.
audio-chunks/chunk4.wav : consider again the original example about predicting
credit card fraud
audio-chunks/chunk4.wav : Consider again the original example about predicting
credit card fraud.
audio-chunks/chunk5.wav : you further formulated the problem
audio-chunks/chunk5.wav : You further formulated the problem.
audio-chunks/chunk6.wav : but what data do you need to actually train your model
so you can get the desired output and subsequently achieve your intended
business outcome
audio-chunks/chunk6.wav : But what data do you need to actually train your model
so you can get the desired output and subsequently achieve your intended
business outcome.
audio-chunks/chunk7.wav : do you have access to the data
audio-chunks/chunk7.wav : Do you have access to the data.
audio-chunks/chunk8.wav : if so how much data do you have and where is it
audio-chunks/chunk8.wav : If so how much data do you have and where is it.
audio-chunks/chunk9.wav : what solution can you use to bring all this data into
one centralized Repository
audio-chunks/chunk9.wav : What solution can you use to bring all this data into
one centralized repository.
audio-chunks/chunk10.wav : the answers to these questions are essential at this
stage
```

```
audio-chunks/chunk10.wav : The answers to these questions are essential at this
stage.
audio-chunks/chunk11.wav : the good news for a budding data scientist is that
there are many places where you can obtain data
audio-chunks/chunk11.wav : The good news for a budding data scientist is that
there are many places where you can obtain data.
audio-chunks/chunk12.wav : private data from you or your existing customer
already exists
audio-chunks/chunk12.wav : Private data from you or your existing customer
already exists.
audio-chunks/chunk13.wav : including everything from log files to customary
invoice databases
audio-chunks/chunk13.wav : Including everything from log files to customary
invoice databases.
audio-chunks/chunk14.wav : private data can be useful depending on the problem
you're trying to solve
audio-chunks/chunk14.wav : Private data can be useful depending on the problem
you're trying to solve.
audio-chunks/chunk15.wav : in many cases
audio-chunks/chunk15.wav : In many cases.
audio-chunks/chunk16.wav : private data is found in many different systems
audio-chunks/chunk16.wav : Private data is found in many different systems.
audio-chunks/chunk17.wav : we'll look at how to bring the sources together
shortly
audio-chunks/chunk17.wav : We'll look at how to bring the sources together
shortly.
audio-chunks/chunk18.wav : sometimes you want to use data that was collected and
made available by a commercial organization
audio-chunks/chunk18.wav : Sometimes you want to use data that was collected and
made available by a commercial organization.
audio-chunks/chunk19.wav : companies such as Reuters change Healthcare Dun &
Bradstreet and Foursquare maintain databases you can subscribe to
audio-chunks/chunk19.wav : Companies such as reuters change healthcare dun &
bradstreet and foursquare maintain databases you can subscribe to.
audio-chunks/chunk20.wav : they include curated news
audio-chunks/chunk20.wav : They include curated news.
audio-chunks/chunk21.wav : anonymized Healthcare transactions
audio-chunks/chunk21.wav : Anonymized healthcare transactions.
audio-chunks/chunk22.wav : Global business record
audio-chunks/chunk22.wav : Global business record.
audio-chunks/chunk23.wav : what's and location data
audio-chunks/chunk23.wav : What's and location data.
audio-chunks/chunk24.wav : if you supplement your own data with commercial data
audio-chunks/chunk24.wav : If you supplement your own data with commercial data.
audio-chunks/chunk25.wav : you can get useful insights you wouldn't have gotten
otherwise
audio-chunks/chunk25.wav : You can get useful insights you wouldn't have gotten
otherwise.
```

```
audio-chunks/chunk26.wav : there are also many open source data sets ranging
from wine quality to movie reviews
audio-chunks/chunk26.wav : There are also many open source data sets ranging
from wine quality to movie reviews.
audio-chunks/chunk27.wav : these data sets are made available for use in
research or for teaching purposes
audio-chunks/chunk27.wav : These data sets are made available for use in
research or for teaching purposes.
audio-chunks/chunk28.wav : AWS
audio-chunks/chunk28.wav : Aws.
audio-chunks/chunk29.wav : kaggle and the UCI machine learning repositories are
good places to find open source data sets
audio-chunks/chunk29.wav : Kaggle and the uci machine learning repositories are
good places to find open source data sets.
audio-chunks/chunk30.wav : government and health organizations are other sources
of data that could be useful
audio-chunks/chunk30.wav : Government and health organizations are other sources
of data that could be useful.
audio-chunks/chunk31.wav : supervised machine learning problems need a lot of
data
audio-chunks/chunk31.wav : Supervised machine learning problems need a lot of
data.
audio-chunks/chunk32.wav : these are also called observations
audio-chunks/chunk32.wav : These are also called observations.
audio-chunks/chunk33.wav : and you already need to know the target answer or
prediction for that data
audio-chunks/chunk33.wav : And you already need to know the target answer or
prediction for that data.
audio-chunks/chunk34.wav : this kind of data were you already know the target
answer or prediction is called labeled data
audio-chunks/chunk34.wav : This kind of data were you already know the target
answer or prediction is called labeled data.
audio-chunks/chunk35.wav : each observation in your data is made up of two
element
audio-chunks/chunk35.wav : Each observation in your data is made up of two
element.
audio-chunks/chunk36.wav : the target
audio-chunks/chunk36.wav : The target.
audio-chunks/chunk37.wav : end the features
audio-chunks/chunk37.wav : End the features.
audio-chunks/chunk38.wav : the target is the answer you want to predict
audio-chunks/chunk38.wav : The target is the answer you want to predict.
audio-chunks/chunk39.wav : so in the credit card transaction example the target
of Any Given observation is either fraud or not fraud
audio-chunks/chunk39.wav : So in the credit card transaction example the target
of any given observation is either fraud or not fraud.
audio-chunks/chunk40.wav : a feature is an attribute of the example that you can
use to identify patterns for predicting the target answer
```

```
audio-chunks/chunk40.wav : A feature is an attribute of the example that you can
use to identify patterns for predicting the target answer.
audio-chunks/chunk41.wav : a feature in the credit card example could be the
date of the transaction the vendor or the amount in dollars of the transaction
audio-chunks/chunk41.wav : A feature in the credit card example could be the
date of the transaction the vendor or the amount in dollars of the transaction.
audio-chunks/chunk42.wav : you might wonder if the source of the target is fraud
or not fraud
audio-chunks/chunk42.wav : You might wonder if the source of the target is fraud
or not fraud.
audio-chunks/chunk43.wav : typically this information is discovered only after
the transaction is complete
audio-chunks/chunk43.wav : Typically this information is discovered only after
the transaction is complete.
audio-chunks/chunk44.wav : and the actual card owner notices a fraudulent
transaction on their statement
audio-chunks/chunk44.wav : And the actual card owner notices a fraudulent
transaction on their statement.
audio-chunks/chunk45.wav : this information would be recorded with the
transaction for exactly the purpose of using it to train a future model
audio-chunks/chunk45.wav : This information would be recorded with the
transaction for exactly the purpose of using it to train a future model.
audio-chunks/chunk46.wav : so given what you know about the elements of an ml
data set will return to one of the original questions
audio-chunks/chunk46.wav : So given what you know about the elements of an ml
data set will return to one of the original questions.
audio-chunks/chunk47.wav : what data do you need to actually train your model to
reach the desired output and subsequently your intended business outcome
audio-chunks/chunk47.wav : What data do you need to actually train your model to
reach the desired output and subsequently your intended business outcome.
audio-chunks/chunk48.wav : this is an example of a stage in the ml pipeline when
audio-chunks/chunk48.wav : This is an example of a stage in the ml pipeline
when.
audio-chunks/chunk49.wav : crucial to get domain expertise to help you answer
this question
audio-chunks/chunk49.wav : Crucial to get domain expertise to help you answer
this question.
audio-chunks/chunk50.wav : with domain knowledge you can start determining the
features and Target data your model will need to make accurate predictions
audio-chunks/chunk50.wav : With domain knowledge you can start determining the
features and target data your model will need to make accurate predictions.
audio-chunks/chunk51.wav : your data should be representative of the data you'll
have when you're using the model to make a prediction
audio-chunks/chunk51.wav : Your data should be representative of the data you'll
have when you're using the model to make a prediction.
audio-chunks/chunk52.wav : for example if you want to predict credit card fraud
audio-chunks/chunk52.wav : For example if you want to predict credit card fraud.
audio-chunks/chunk53.wav : you need to collect data for positive or fraudulent
```

```
transactions
audio-chunks/chunk53.wav : You need to collect data for positive or fraudulent
transactions.
audio-chunks/chunk54.wav : you also need to collect data for negative or non
fraudulent transactions
audio-chunks/chunk54.wav : You also need to collect data for negative or non
fraudulent transactions.
audio-chunks/chunk55.wav : you need both types of data so the machine learning
algorithm can find patterns that will distinguish between the two types
audio-chunks/chunk55.wav : You need both types of data so the machine learning
algorithm can find patterns that will distinguish between the two types.
audio-chunks/chunk56.wav : suppose your average amount of fraudulent
transactions is actually 3%
audio-chunks/chunk56.wav : Suppose your average amount of fraudulent
transactions is actually 3%.
audio-chunks/chunk57.wav : put your training data set only includes a very small
fraction of fraudulent observations say 0.4%
audio-chunks/chunk57.wav : Put your training data set only includes a very small
fraction of fraudulent observations say 0.4%.
audio-chunks/chunk58.wav : in this case
audio-chunks/chunk58.wav : In this case.
audio-chunks/chunk59.wav : it'll be difficult for your model to truly learn
patterns related to fraudulent transactions that it might encounter in
production
audio-chunks/chunk59.wav : It'll be difficult for your model to truly learn
patterns related to fraudulent transactions that it might encounter in
production.
audio-chunks/chunk60.wav : there are many different services in AWS where you
can find or store your data
audio-chunks/chunk60.wav : There are many different services in aws where you
can find or store your data.
audio-chunks/chunk61.wav : here are some key Services you might use
audio-chunks/chunk61.wav : Here are some key services you might use.
audio-chunks/chunk62.wav : Amazon simple storage service is also known as Amazon
S3
audio-chunks/chunk62.wav : Amazon simple storage service is also known as amazon
s3.
audio-chunks/chunk63.wav : it provides object level storage
audio-chunks/chunk63.wav : It provides object level storage.
audio-chunks/chunk64.wav : with S3 you can store as much data as you want in the
form of objects which you can think of as files
audio-chunks/chunk64.wav : With s3 you can store as much data as you want in the
form of objects which you can think of as files.
audio-chunks/chunk65.wav : they could be CSV files or files of other formats you
need
audio-chunks/chunk65.wav : They could be csv files or files of other formats you
need.
audio-chunks/chunk66.wav : S3 can be accessed through the web-based AWS
```

```
Management console
audio-chunks/chunk66.wav : S3 can be accessed through the web-based aws
management console.
audio-chunks/chunk67.wav : you can also access S3 programmatically through the
API and SDK
audio-chunks/chunk67.wav : You can also access s3 programmatically through the
api and sdk.
audio-chunks/chunk68.wav : or with third-party Solutions which also use the API
and sdks
audio-chunks/chunk68.wav : Or with third-party solutions which also use the api
and sdks.
audio-chunks/chunk69.wav : if your training data is already in S3 and you're
planning to run training jobs several times with different algorithms and
parameters
audio-chunks/chunk69.wav : If your training data is already in s3 and you're
planning to run training jobs several times with different algorithms and
parameters.
audio-chunks/chunk70.wav : you could use Amazon FSX for Lustre
audio-chunks/chunk70.wav : You could use amazon fsx for lustre.
audio-chunks/chunk71.wav : it's a file system service that speeds up your
training jobs by serving your S3 data to Amazon sagemaker at high speeds
audio-chunks/chunk71.wav : It's a file system service that speeds up your
training jobs by serving your s3 data to amazon sagemaker at high speeds.
audio-chunks/chunk72.wav : the first time you run a training job FSX for Lustre
automatically copies data from S3
audio-chunks/chunk72.wav : The first time you run a training job fsx for lustre
automatically copies data from s3.
audio-chunks/chunk73.wav : and makes it available to sagemaker
audio-chunks/chunk73.wav : And makes it available to sagemaker.
audio-chunks/chunk74.wav : you can use the same Amazon FSX file system for
subsequent iterations of training jobs
audio-chunks/chunk74.wav : You can use the same amazon fsx file system for
subsequent iterations of training jobs.
audio-chunks/chunk75.wav : which prevents repeated downloads of common S3
objects
audio-chunks/chunk75.wav : Which prevents repeated downloads of common s3
objects.
audio-chunks/chunk76.wav : alternatively your training data might already be in
Amazon elastic file system or Amazon EFS
audio-chunks/chunk76.wav : Alternatively your training data might already be in
amazon elastic file system or amazon efs.
audio-chunks/chunk77.wav : if so we recommend using EFS as your data source for
training data
audio-chunks/chunk77.wav : If so we recommend using efs as your data source for
training data.
audio-chunks/chunk78.wav : it can launch your training jobs directly from the
service without needing data movement which results in Faster training start
times
```

```
audio-chunks/chunk78.wav : It can launch your training jobs directly from the
service without needing data movement which results in faster training start
times.
audio-chunks/chunk79.wav : this is often the case an environments where data
scientists have home directories in Amazon EFS
audio-chunks/chunk79.wav : This is often the case an environments where data
scientists have home directories in amazon efs.
audio-chunks/chunk80.wav : they can quickly iterate on their models by bringing
in new data
audio-chunks/chunk80.wav : They can quickly iterate on their models by bringing
in new data.
audio-chunks/chunk81.wav : sharing data with colleagues
audio-chunks/chunk81.wav : Sharing data with colleagues.
audio-chunks/chunk82.wav : an experimenting with different fields or labels in
their data set
audio-chunks/chunk82.wav : An experimenting with different fields or labels in
their data set.
audio-chunks/chunk83.wav : for example a data scientist can use a jupyter
notebook to do an initial cleansing on a training set and launched a training
job from Amazon sagemaker
audio-chunks/chunk83.wav : For example a data scientist can use a jupyter
notebook to do an initial cleansing on a training set and launched a training
job from amazon sagemaker.
audio-chunks/chunk84.wav : they could then use their jupyter notebook to drop a
column and relaunch the training job
audio-chunks/chunk84.wav : They could then use their jupyter notebook to drop a
column and relaunch the training job.
audio-chunks/chunk85.wav : and finally compare the resulting models to see which
one works better
audio-chunks/chunk85.wav : And finally compare the resulting models to see which
one works better.
audio-chunks/chunk86.wav : there are many other AWS services and resources where
you might find data
audio-chunks/chunk86.wav : There are many other aws services and resources where
you might find data.
audio-chunks/chunk87.wav : for example you could use Amazon relational database
service or Amazon RDS a managed relational database service
audio-chunks/chunk87.wav : For example you could use amazon relational database
service or amazon rds a managed relational database service.
audio-chunks/chunk88.wav : you can also use Amazon redshift which is a manage
data warehouse service
audio-chunks/chunk88.wav : You can also use amazon redshift which is a manage
data warehouse service.
audio-chunks/chunk89.wav : another option is Amazon timestream a manage time
series database designed specifically to handle large amounts of data from the
internet of things or iot
audio-chunks/chunk89.wav : Another option is amazon timestream a manage time
series database designed specifically to handle large amounts of data from the
```

```
internet of things or iot.
audio-chunks/chunk90.wav : you could even spin up your own instances on Amazon
elastic compute Cloud which is also known as Amazon ec2
audio-chunks/chunk90.wav : You could even spin up your own instances on amazon
elastic compute cloud which is also known as amazon ec2.
audio-chunks/chunk91.wav : and host your own database on these instances
audio-chunks/chunk91.wav : And host your own database on these instances.
audio-chunks/chunk92.wav : when you have data sources
audio-chunks/chunk92.wav : When you have data sources.
audio-chunks/chunk93.wav : you'll need to extract useful data from these sources
when assembling your data for machine learning
audio-chunks/chunk93.wav : You'll need to extract useful data from these sources
when assembling your data for machine learning.
audio-chunks/chunk94.wav : we'll look at this next
audio-chunks/chunk94.wav : We'll look at this next.
audio-chunks/chunk95.wav : that's it for part 1 of this section will see you
again for part 2 where we'll review how to extract
audio-chunks/chunk95.wav : That's it for part 1 of this section will see you
again for part 2 where we'll review how to extract.
audio-chunks/chunk96.wav : transform and load data
audio-chunks/chunk96.wav : Transform and load data.
audio-chunks/chunk1.wav : hi and welcome to section 1
audio-chunks/chunk1.wav : Hi and welcome to section 1.
audio-chunks/chunk2.wav : we'll get started by reviewing what forecasting is and
some use cases for it
audio-chunks/chunk2.wav : We'll get started by reviewing what forecasting is and
some use cases for it.
audio-chunks/chunk3.wav : forecasting is an important area of machine learning
audio-chunks/chunk3.wav : Forecasting is an important area of machine learning.
audio-chunks/chunk4.wav : it's important because there are so many opportunities
for predicting future outcomes based on historical data
audio-chunks/chunk4.wav : It's important because there are so many opportunities
for predicting future outcomes based on historical data.
audio-chunks/chunk5.wav : many of these opportunities involve a Time component
audio-chunks/chunk5.wav : Many of these opportunities involve a time component.
audio-chunks/chunk6.wav : however while the time component adds additional
information
audio-chunks/chunk6.wav : However while the time component adds additional
information.
audio-chunks/chunk7.wav : it also makes time series problems more difficult to
handle
audio-chunks/chunk7.wav : It also makes time series problems more difficult to
handle.
audio-chunks/chunk8.wav : compared to other types of predictions
audio-chunks/chunk8.wav : Compared to other types of predictions.
audio-chunks/chunk9.wav : you can think of Time series data as falling into two
broad categories
audio-chunks/chunk9.wav : You can think of time series data as falling into two
```

```
broad categories.
audio-chunks/chunk10.wav : the first
audio-chunks/chunk10.wav : The first.
audio-chunks/chunk11.wav : is univariate data
audio-chunks/chunk11.wav : Is univariate data.
audio-chunks/chunk12.wav : which means there's just one variable
audio-chunks/chunk12.wav : Which means there's just one variable.
audio-chunks/chunk13.wav : the second one is multivariate data
audio-chunks/chunk13.wav : The second one is multivariate data.
audio-chunks/chunk14.wav : which means there's more than one variable
audio-chunks/chunk14.wav : Which means there's more than one variable.
audio-chunks/chunk15.wav : there are several common patterns in Time series data
audio-chunks/chunk15.wav : There are several common patterns in time series
data.
audio-chunks/chunk16.wav : the first pattern is a trend
audio-chunks/chunk16.wav : The first pattern is a trend.
audio-chunks/chunk17.wav : with a trend you get a pattern with the values
increasing decreasing or staying the same over time
audio-chunks/chunk17.wav : With a trend you get a pattern with the values
increasing decreasing or staying the same over time.
audio-chunks/chunk18.wav : there are seasonal patterns
audio-chunks/chunk18.wav : There are seasonal patterns.
audio-chunks/chunk19.wav : these reflect times of the year month
audio-chunks/chunk19.wav : These reflect times of the year month.
audio-chunks/chunk20.wav : day or other patterns
audio-chunks/chunk20.wav : Day or other patterns.
audio-chunks/chunk21.wav : cyclical patterns are similar to seasonal patterns
audio-chunks/chunk21.wav : Cyclical patterns are similar to seasonal patterns.
audio-chunks/chunk22.wav : these are patterns that repeat
audio-chunks/chunk22.wav : These are patterns that repeat.
audio-chunks/chunk23.wav : like a large retail sale event that happens the same
time each year
audio-chunks/chunk23.wav : Like a large retail sale event that happens the same
time each year.
audio-chunks/chunk24.wav : finally there are changes in the data over time that
appear to be random
audio-chunks/chunk24.wav : Finally there are changes in the data over time that
appear to be random.
audio-chunks/chunk25.wav : or that have no discernible pattern
audio-chunks/chunk25.wav : Or that have no discernible pattern.
audio-chunks/chunk26.wav : there are many uses for forecasting
audio-chunks/chunk26.wav : There are many uses for forecasting.
audio-chunks/chunk27.wav : you can use forecasting in marketing applications
audio-chunks/chunk27.wav : You can use forecasting in marketing applications.
audio-chunks/chunk28.wav : searches for sales forecasting or demand projections
audio-chunks/chunk28.wav : Searches for sales forecasting or demand projections.
audio-chunks/chunk29.wav : it could also be used in Inventory management systems
that anticipate required inventory levels
```

```
audio-chunks/chunk29.wav : It could also be used in inventory management systems
that anticipate required inventory levels.
audio-chunks/chunk30.wav : forecasting energy consumption can help predict when
and where energy is needed
audio-chunks/chunk30.wav : Forecasting energy consumption can help predict when
and where energy is needed.
audio-chunks/chunk31.wav : and weather forecasting systems
audio-chunks/chunk31.wav : And weather forecasting systems.
audio-chunks/chunk32.wav : can be used for governments in commercial
applications such as agriculture
audio-chunks/chunk32.wav : Can be used for governments in commercial
applications such as agriculture.
audio-chunks/chunk33.wav : that's it for this section
audio-chunks/chunk33.wav : That's it for this section.
audio-chunks/chunk34.wav : see you in the next video
audio-chunks/chunk34.wav : See you in the next video.
audio-chunks/chunk1.wav : hi and welcome to module 4 of AWS Academy machine
learning
audio-chunks/chunk1.wav : Hi and welcome to module 4 of aws academy machine
learning.
audio-chunks/chunk2.wav : in this module we're going to look at forecasting
audio-chunks/chunk2.wav : In this module we're going to look at forecasting.
audio-chunks/chunk3.wav : we'll start with an introduction to forecast
audio-chunks/chunk3.wav : We'll start with an introduction to forecast.
audio-chunks/chunk4.wav : and look at how time series data is different from
other kinds of data
audio-chunks/chunk4.wav : And look at how time series data is different from
other kinds of data.
audio-chunks/chunk5.wav : then we're going to look at Amazon forecast
audio-chunks/chunk5.wav : Then we're going to look at amazon forecast.
audio-chunks/chunk6.wav : a service that helps you simplify building forecasts
audio-chunks/chunk6.wav : A service that helps you simplify building forecasts.
audio-chunks/chunk7.wav : at the end of this module you'll be able to describe
the business problems solved with Amazon forecast
audio-chunks/chunk7.wav : At the end of this module you'll be able to describe
the business problems solved with amazon forecast.
audio-chunks/chunk8.wav : describe the challenges of working with time series
data
audio-chunks/chunk8.wav : Describe the challenges of working with time series
data.
audio-chunks/chunk9.wav : list the steps required to create a forecast by using
Amazon forecast
audio-chunks/chunk9.wav : List the steps required to create a forecast by using
amazon forecast.
audio-chunks/chunk10.wav : and use Amazon forecast to make a prediction
audio-chunks/chunk10.wav : And use amazon forecast to make a prediction.
audio-chunks/chunk11.wav : see you in the next video
audio-chunks/chunk11.wav : See you in the next video.
```

```
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : it's now time to review the module and wrap it up
audio-chunks/chunk2.wav : It's now time to review the module and wrap it up.
audio-chunks/chunk3.wav : in this module you learned how to
audio-chunks/chunk3.wav : In this module you learned how to.
audio-chunks/chunk4.wav : describe the business problem solved by Amazon
forecast
audio-chunks/chunk4.wav : Describe the business problem solved by amazon
forecast.
audio-chunks/chunk5.wav : describe the challenges of working with time series
data
audio-chunks/chunk5.wav : Describe the challenges of working with time series
data.
audio-chunks/chunk6.wav : list the steps required to create forecast by using
Amazon forecast
audio-chunks/chunk6.wav : List the steps required to create forecast by using
amazon forecast.
audio-chunks/chunk7.wav : and use Amazon forecast to make a prediction
audio-chunks/chunk7.wav : And use amazon forecast to make a prediction.
audio-chunks/chunk8.wav : thanks for participating
audio-chunks/chunk8.wav : Thanks for participating.
audio-chunks/chunk9.wav : see you in the next mod
audio-chunks/chunk9.wav : See you in the next mod.
Error:
audio-chunks/chunk1.wav : hi and welcome back
audio-chunks/chunk1.wav : Hi and welcome back.
audio-chunks/chunk2.wav : this is section 2 and we're going to focus on
processing time series data
audio-chunks/chunk2.wav : This is section 2 and we're going to focus on
processing time series data.
audio-chunks/chunk3.wav : because it can be different from other types of data
you've been using so far
audio-chunks/chunk3.wav : Because it can be different from other types of data
you've been using so far.
audio-chunks/chunk4.wav : time series data is data that is captured in
chronological sequence over a defined period of time
audio-chunks/chunk4.wav : Time series data is data that is captured in
chronological sequence over a defined period of time.
audio-chunks/chunk5.wav : introducing time into a machine learning model has a
positive impact
audio-chunks/chunk5.wav : Introducing time into a machine learning model has a
positive impact.
audio-chunks/chunk6.wav : because the model can derive meaning from changes in
the data points over time
audio-chunks/chunk6.wav : Because the model can derive meaning from changes in
the data points over time.
audio-chunks/chunk7.wav : time series data tends to be correlated
```

```
audio-chunks/chunk7.wav : Time series data tends to be correlated.
audio-chunks/chunk8.wav : this means that there's a dependency between data
point
audio-chunks/chunk8.wav : This means that there's a dependency between data
point.
audio-chunks/chunk9.wav : this has mixed results for forecasting
audio-chunks/chunk9.wav : This has mixed results for forecasting.
audio-chunks/chunk10.wav : this is because you're dealing with a regression
problem
audio-chunks/chunk10.wav : This is because you're dealing with a regression
problem.
audio-chunks/chunk11.wav : and regression assumes that data points are
independent
audio-chunks/chunk11.wav : And regression assumes that data points are
independent.
audio-chunks/chunk12.wav : you need to develop a method for dealing with data
dependence
audio-chunks/chunk12.wav : You need to develop a method for dealing with data
dependence.
audio-chunks/chunk13.wav : so you can increase the validity of the predictions
audio-chunks/chunk13.wav : So you can increase the validity of the predictions.
audio-chunks/chunk14.wav : in addition to the time series data
audio-chunks/chunk14.wav : In addition to the time series data.
audio-chunks/chunk15.wav : you can add related data to augmenta forecasting
model
audio-chunks/chunk15.wav : You can add related data to augmenta forecasting
model.
audio-chunks/chunk16.wav : for example suppose you want to make a prediction
about retail sales
audio-chunks/chunk16.wav : For example suppose you want to make a prediction
about retail sales.
audio-chunks/chunk17.wav : you could include information about the product being
sold such as item identification
audio-chunks/chunk17.wav : You could include information about the product being
sold such as item identification.
audio-chunks/chunk18.wav : are sales
audio-chunks/chunk18.wav : Are sales.
audio-chunks/chunk19.wav : along with the number of units sold per time period
audio-chunks/chunk19.wav : Along with the number of units sold per time period.
audio-chunks/chunk20.wav : the third type of data is metadata about the data set
audio-chunks/chunk20.wav : The third type of data is metadata about the data
set.
audio-chunks/chunk21.wav : for instance
audio-chunks/chunk21.wav : For instance.
audio-chunks/chunk22.wav : say that you have a retail data set
audio-chunks/chunk22.wav : Say that you have a retail data set.
audio-chunks/chunk23.wav : you might want to include metadata like a brand name
or a genre for music or videos so you can group results
```

```
audio-chunks/chunk23.wav : You might want to include metadata like a brand name
or a genre for music or videos so you can group results.
audio-chunks/chunk24.wav : it's better to have more data
audio-chunks/chunk24.wav : It's better to have more data.
audio-chunks/chunk25.wav : when you work with multiple data sources
audio-chunks/chunk25.wav : When you work with multiple data sources.
audio-chunks/chunk26.wav : you'll face the challenge of handling the timestamp
of the data
audio-chunks/chunk26.wav : You'll face the challenge of handling the timestamp
of the data.
audio-chunks/chunk27.wav : you'll observe differences in the timestamp format
and other challenges such as incomplete data
audio-chunks/chunk27.wav : You'll observe differences in the timestamp format
and other challenges such as incomplete data.
audio-chunks/chunk28.wav : however you might be able to infer missing data in
some cases
audio-chunks/chunk28.wav : However you might be able to infer missing data in
some cases.
audio-chunks/chunk29.wav : for example
audio-chunks/chunk29.wav : For example.
audio-chunks/chunk30.wav : say you have some data that contains both the month
and the day but no year
audio-chunks/chunk30.wav : Say you have some data that contains both the month
and the day but no year.
audio-chunks/chunk31.wav : observe whether the data seems to sequence through
the month numbers in the database
audio-chunks/chunk31.wav : Observe whether the data seems to sequence through
the month numbers in the database.
audio-chunks/chunk32.wav : repeating after 12:00
audio-chunks/chunk32.wav : Repeating after 12:00.
audio-chunks/chunk33.wav : if it does so
audio-chunks/chunk33.wav : If it does so.
audio-chunks/chunk34.wav : you could add the year if you knew when the data
started
audio-chunks/chunk34.wav : You could add the year if you knew when the data
started.
audio-chunks/chunk35.wav : you can then infer future years based on the order of
the data
audio-chunks/chunk35.wav : You can then infer future years based on the order of
the data.
audio-chunks/chunk36.wav : much time stamp data is stored in UTC form
audio-chunks/chunk36.wav : Much time stamp data is stored in utc form.
audio-chunks/chunk37.wav : but not all data is
audio-chunks/chunk37.wav : But not all data is.
audio-chunks/chunk38.wav : you should check if the timestamp is in local or
universal time
audio-chunks/chunk38.wav : You should check if the timestamp is in local or
universal time.
```

```
audio-chunks/chunk39.wav : sometimes the timestamp doesn't represent the time
you think it does
audio-chunks/chunk39.wav : Sometimes the timestamp doesn't represent the time
you think it does.
audio-chunks/chunk40.wav : for example
audio-chunks/chunk40.wav : For example.
audio-chunks/chunk41.wav : suppose you have a database of cars that were
serviced at a garage
audio-chunks/chunk41.wav : Suppose you have a database of cars that were
serviced at a garage.
audio-chunks/chunk42.wav : does the timestamp indicate the time the car arrived
was completed or picked up
audio-chunks/chunk42.wav : Does the timestamp indicate the time the car arrived
was completed or picked up.
audio-chunks/chunk43.wav : or does it indicate when the final entry was entered
into the system
audio-chunks/chunk43.wav : Or does it indicate when the final entry was entered
into the system.
audio-chunks/chunk44.wav : say you're trying to model the hourly caloric intake
of patients
audio-chunks/chunk44.wav : Say you're trying to model the hourly caloric intake
of patients.
audio-chunks/chunk45.wav : however you only have daily data
audio-chunks/chunk45.wav : However you only have daily data.
audio-chunks/chunk46.wav : then you'll need to adjust your target time scale
audio-chunks/chunk46.wav : Then you'll need to adjust your target time scale.
audio-chunks/chunk47.wav : also
audio-chunks/chunk47.wav : Also.
audio-chunks/chunk48.wav : your data might not have any time stamps
audio-chunks/chunk48.wav : Your data might not have any time stamps.
audio-chunks/chunk49.wav : there could be other ways to extrapolate a Time
series depending on the data and domain
audio-chunks/chunk49.wav : There could be other ways to extrapolate a time
series depending on the data and domain.
audio-chunks/chunk50.wav : for example you might have wavelength measurements or
vectors within an image
audio-chunks/chunk50.wav : For example you might have wavelength measurements or
vectors within an image.
audio-chunks/chunk51.wav : as a final note
audio-chunks/chunk51.wav : As a final note.
audio-chunks/chunk52.wav : remember that daylight savings is different around
the world
audio-chunks/chunk52.wav : Remember that daylight savings is different around
the world.
audio-chunks/chunk53.wav : also because of daylight savings time might even
occur twice a year in their time zones
audio-chunks/chunk53.wav : Also because of daylight savings time might even
occur twice a year in their time zones.
```

```
audio-chunks/chunk54.wav : a common occurrence in real-world forecasting
problems is missing values in the Raw data
audio-chunks/chunk54.wav : A common occurrence in real-world forecasting
problems is missing values in the raw data.
audio-chunks/chunk55.wav : missing values makes it harder for a model to
generate a forecast
audio-chunks/chunk55.wav : Missing values makes it harder for a model to
generate a forecast.
audio-chunks/chunk56.wav : the primary example in retail is an out-of-stock
situation in demand forecasting
audio-chunks/chunk56.wav : The primary example in retail is an out-of-stock
situation in demand forecasting.
audio-chunks/chunk57.wav : If an item goes out of stock
audio-chunks/chunk57.wav : If an item goes out of stock.
audio-chunks/chunk58.wav : does sales for the day will Zero
audio-chunks/chunk58.wav : Does sales for the day will zero.
audio-chunks/chunk59.wav : if the forecast is generated based on those zero
sales values the forecast will be incorrect
audio-chunks/chunk59.wav : If the forecast is generated based on those zero
sales values the forecast will be incorrect.
audio-chunks/chunk60.wav : there are many reasons why values can be marked as
missing
audio-chunks/chunk60.wav : There are many reasons why values can be marked as
missing.
audio-chunks/chunk61.wav : missing values can occur because of no transaction
audio-chunks/chunk61.wav : Missing values can occur because of no transaction.
audio-chunks/chunk62.wav : they can also occur
audio-chunks/chunk62.wav : They can also occur.
audio-chunks/chunk63.wav : because of possible measurement errors
audio-chunks/chunk63.wav : Because of possible measurement errors.
audio-chunks/chunk64.wav : for example a service that monitored certain data
wasn't working correctly
audio-chunks/chunk64.wav : For example a service that monitored certain data
wasn't working correctly.
audio-chunks/chunk65.wav : or is another example
audio-chunks/chunk65.wav : Or is another example.
audio-chunks/chunk66.wav : the measurement couldn't happen correctly
audio-chunks/chunk66.wav : The measurement couldn't happen correctly.
audio-chunks/chunk67.wav : in retail the primary example for an inability to
take correct measurements is an out of stock situation in demand forecasting
audio-chunks/chunk67.wav : In retail the primary example for an inability to
take correct measurements is an out of stock situation in demand forecasting.
audio-chunks/chunk68.wav : this means that demand doesn't equal sales on that
day
audio-chunks/chunk68.wav : This means that demand doesn't equal sales on that
day.
audio-chunks/chunk69.wav : there are several ways you can calculate the missing
data
```

```
audio-chunks/chunk69.wav : There are several ways you can calculate the missing
data.
audio-chunks/chunk70.wav : the first method is forward fill
audio-chunks/chunk70.wav : The first method is forward fill.
audio-chunks/chunk71.wav : this uses the last known value for the missing value
audio-chunks/chunk71.wav : This uses the last known value for the missing value.
audio-chunks/chunk72.wav : building on that idea moving average uses the average
of the last known values to calculate the missing value
audio-chunks/chunk72.wav : Building on that idea moving average uses the average
of the last known values to calculate the missing value.
audio-chunks/chunk73.wav : backwards
audio-chunks/chunk73.wav : Backwards.
audio-chunks/chunk74.wav : uses the next known value after the missing value
audio-chunks/chunk74.wav : Uses the next known value after the missing value.
audio-chunks/chunk75.wav : the danger here is that you're using the future to
calculate the past which is bad in forecasting
audio-chunks/chunk75.wav : The danger here is that you're using the future to
calculate the past which is bad in forecasting.
audio-chunks/chunk76.wav : this method is also known as look ahead and should be
avoided
audio-chunks/chunk76.wav : This method is also known as look ahead and should be
avoided.
audio-chunks/chunk77.wav : interpolation uses an equation to calculate the
missing value
audio-chunks/chunk77.wav : Interpolation uses an equation to calculate the
missing value.
audio-chunks/chunk78.wav : you can also use a zero fill
audio-chunks/chunk78.wav : You can also use a zero fill.
audio-chunks/chunk79.wav : this is often used in retail
audio-chunks/chunk79.wav : This is often used in retail.
audio-chunks/chunk80.wav : because missing sales data shouldn't be calculated
audio-chunks/chunk80.wav : Because missing sales data shouldn't be calculated.
audio-chunks/chunk81.wav : the missing data represents that there were no orders
on that day
audio-chunks/chunk81.wav : The missing data represents that there were no orders
on that day.
audio-chunks/chunk82.wav : it would be wise to investigate why this happened
audio-chunks/chunk82.wav : It would be wise to investigate why this happened.
audio-chunks/chunk83.wav : but in this case you don't want to fill in the
missing value
audio-chunks/chunk83.wav : But in this case you don't want to fill in the
missing value.
audio-chunks/chunk84.wav : you might get data at different frequencies
audio-chunks/chunk84.wav : You might get data at different frequencies.
audio-chunks/chunk85.wav : for example you might have sales data that includes
the exact time stamp the sale was recorded
audio-chunks/chunk85.wav : For example you might have sales data that includes
the exact time stamp the sale was recorded.
```

```
audio-chunks/chunk86.wav : but have inventory data that only contains the year
month and day of the inventory level
audio-chunks/chunk86.wav : But have inventory data that only contains the year
month and day of the inventory level.
audio-chunks/chunk87.wav : when you have data that's at a different frequency
than other data sets
audio-chunks/chunk87.wav : When you have data that's at a different frequency
than other data sets.
audio-chunks/chunk88.wav : or data that's not compatible with your question you
might need to downsample
audio-chunks/chunk88.wav : Or data that's not compatible with your question you
might need to downsample.
audio-chunks/chunk89.wav : down sampling
audio-chunks/chunk89.wav : Down sampling.
audio-chunks/chunk90.wav : is moving from a more finely grained time to a less
finally grain time
audio-chunks/chunk90.wav : Is moving from a more finely grained time to a less
finally grain time.
audio-chunks/chunk91.wav : as the example shows this could be converting an
hourly data set
audio-chunks/chunk91.wav : As the example shows this could be converting an
hourly data set.
audio-chunks/chunk92.wav : to a daily data set
audio-chunks/chunk92.wav : To a daily data set.
audio-chunks/chunk93.wav : when downsampling you need to decide how to combine
the values
audio-chunks/chunk93.wav : When downsampling you need to decide how to combine
the values.
audio-chunks/chunk94.wav : in the previous case of sales data
audio-chunks/chunk94.wav : In the previous case of sales data.
audio-chunks/chunk95.wav : summing the quantity makes the most sense
audio-chunks/chunk95.wav : Summing the quantity makes the most sense.
audio-chunks/chunk96.wav : if the data is temperature
audio-chunks/chunk96.wav : If the data is temperature.
audio-chunks/chunk97.wav : you might want to find the average
audio-chunks/chunk97.wav : You might want to find the average.
audio-chunks/chunk98.wav : understanding your data helps you decide what's the
best course of action
audio-chunks/chunk98.wav : Understanding your data helps you decide what's the
best course of action.
audio-chunks/chunk99.wav : the opposite of downsampling is upsampling
audio-chunks/chunk99.wav : The opposite of downsampling is upsampling.
audio-chunks/chunk100.wav : when you move from a less finely grained time
audio-chunks/chunk100.wav : When you move from a less finely grained time.
audio-chunks/chunk101.wav : to a more filing grain time
audio-chunks/chunk101.wav : To a more filing grain time.
audio-chunks/chunk102.wav : the problem with upsampling is that it's extremely
difficult to achieve in most cases
```

```
audio-chunks/chunk102.wav : The problem with upsampling is that it's extremely
difficult to achieve in most cases.
audio-chunks/chunk103.wav : suppose you want to upsample your sales data from
daily sales to hourly sales
audio-chunks/chunk103.wav : Suppose you want to upsample your sales data from
daily sales to hourly sales.
audio-chunks/chunk104.wav : unless you have some other data source to reference
you wouldn't be able to do this
audio-chunks/chunk104.wav : Unless you have some other data source to reference
you wouldn't be able to do this.
audio-chunks/chunk105.wav : there are cases when you need to do something
perhaps to match the frequency of another time series
audio-chunks/chunk105.wav : There are cases when you need to do something
perhaps to match the frequency of another time series.
audio-chunks/chunk106.wav : or you might have an irregular time series or
specific domain knowledge that would help
audio-chunks/chunk106.wav : Or you might have an irregular time series or
specific domain knowledge that would help.
audio-chunks/chunk107.wav : in those cases
audio-chunks/chunk107.wav : In those cases.
audio-chunks/chunk108.wav : you need to be careful how you make the conversion
audio-chunks/chunk108.wav : You need to be careful how you make the conversion.
audio-chunks/chunk109.wav : for the retail example
audio-chunks/chunk109.wav : For the retail example.
audio-chunks/chunk110.wav : the best you can do is create a single order for the
day at a specified hour
audio-chunks/chunk110.wav : The best you can do is create a single order for the
day at a specified hour.
audio-chunks/chunk111.wav : for temperature
audio-chunks/chunk111.wav : For temperature.
audio-chunks/chunk112.wav : you could copy the daily temperature into each
hourly slot
audio-chunks/chunk112.wav : You could copy the daily temperature into each
hourly slot.
audio-chunks/chunk113.wav : or use a formula to calculate a curve
audio-chunks/chunk113.wav : Or use a formula to calculate a curve.
audio-chunks/chunk114.wav : in data science
audio-chunks/chunk114.wav : In data science.
audio-chunks/chunk115.wav : outliers have a mix of positive and negative
attributes
audio-chunks/chunk115.wav : Outliers have a mix of positive and negative
attributes.
audio-chunks/chunk116.wav : the same is true of Time series data
audio-chunks/chunk116.wav : The same is true of time series data.
audio-chunks/chunk117.wav : suppose you were examining sales data
audio-chunks/chunk117.wav : Suppose you were examining sales data.
audio-chunks/chunk118.wav : and you had an order that has an unusually high
number of items
```

```
audio-chunks/chunk118.wav : And you had an order that has an unusually high
number of items.
audio-chunks/chunk119.wav : you might not want to include that in your forecast
calcul
audio-chunks/chunk119.wav : You might not want to include that in your forecast
calcul.
audio-chunks/chunk120.wav : because the order size might never be repeated
audio-chunks/chunk120.wav : Because the order size might never be repeated.
audio-chunks/chunk121.wav : removing these outliers and anomalies is known as
smoothing
audio-chunks/chunk121.wav : Removing these outliers and anomalies is known as
smoothing.
audio-chunks/chunk122.wav : smoothing your data can help you deal with outliers
and other anomalies
audio-chunks/chunk122.wav : Smoothing your data can help you deal with outliers
and other anomalies.
audio-chunks/chunk123.wav : there are a few reasons why you might consider
smoothing
audio-chunks/chunk123.wav : There are a few reasons why you might consider
smoothing.
Error:
audio-chunks/chunk125.wav : during data preparation
audio-chunks/chunk125.wav : During data preparation.
audio-chunks/chunk126.wav : you remove error values and could also remove
outliers
audio-chunks/chunk126.wav : You remove error values and could also remove
outliers.
audio-chunks/chunk127.wav : you might also want to smooth your data to generate
features
audio-chunks/chunk127.wav : You might also want to smooth your data to generate
features.
audio-chunks/chunk128.wav : for visualization you could smooth your data to
reduce the noise in a plot
audio-chunks/chunk128.wav : For visualization you could smooth your data to
reduce the noise in a plot.
audio-chunks/chunk129.wav : it's important to understand why you are smoothing
the data and the impact that it might have
audio-chunks/chunk129.wav : It's important to understand why you are smoothing
the data and the impact that it might have.
audio-chunks/chunk130.wav : the outcome might be to reduce noise and create a
better model
audio-chunks/chunk130.wav : The outcome might be to reduce noise and create a
better model.
audio-chunks/chunk131.wav : but an equally important question is
audio-chunks/chunk131.wav : But an equally important question is.
audio-chunks/chunk132.wav : could your smoothing compromise the model
audio-chunks/chunk132.wav : Could your smoothing compromise the model.
audio-chunks/chunk133.wav : is the model expecting noisy data
```

```
audio-chunks/chunk133.wav : Is the model expecting noisy data.
audio-chunks/chunk134.wav : will you also be able to smooth the data in
production
audio-chunks/chunk134.wav : Will you also be able to smooth the data in
production.
audio-chunks/chunk135.wav : that's it for part 1 of this section will see you
again for part 2 where we'll review more time series specific challenges
audio-chunks/chunk135.wav : That's it for part 1 of this section will see you
again for part 2 where we'll review more time series specific challenges.
audio-chunks/chunk136.wav : and the tools and algorithms that can help us
Wrangle your data
audio-chunks/chunk136.wav : And the tools and algorithms that can help us
wrangle your data.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring video analysis by reviewing
how to create the training data set
audio-chunks/chunk2.wav : Will continue exploring video analysis by reviewing
how to create the training data set.
audio-chunks/chunk3.wav : data sets contain information that's needed to train
and test in Amazon recognition custom labels model
audio-chunks/chunk3.wav : Data sets contain information that's needed to train
and test in amazon recognition custom labels model.
audio-chunks/chunk4.wav : such as images labels and bounding boxes
audio-chunks/chunk4.wav : Such as images labels and bounding boxes.
audio-chunks/chunk5.wav : you can use images from Amazon S3
audio-chunks/chunk5.wav : You can use images from amazon s3.
audio-chunks/chunk6.wav : or you can upload them from your computer to S3 as
part of the process
audio-chunks/chunk6.wav : Or you can upload them from your computer to s3 as
part of the process.
audio-chunks/chunk7.wav : to train a model your data set should have at least
two labels
audio-chunks/chunk7.wav : To train a model your data set should have at least
two labels.
audio-chunks/chunk8.wav : with at least 10 images per label
audio-chunks/chunk8.wav : With at least 10 images per label.
audio-chunks/chunk9.wav : each image in your data set must be labeled
audio-chunks/chunk9.wav : Each image in your data set must be labeled.
audio-chunks/chunk10.wav : as we mentioned earlier you can use the Amazon
recognition custom labels console
audio-chunks/chunk10.wav : As we mentioned earlier you can use the amazon
recognition custom labels console.
audio-chunks/chunk11.wav : or Amazon sagemaker ground truth to label your images
audio-chunks/chunk11.wav : Or amazon sagemaker ground truth to label your
images.
audio-chunks/chunk12.wav : again to train an Amazon recognition custom labels
model
```

```
audio-chunks/chunk12.wav : Again to train an amazon recognition custom labels
model.
audio-chunks/chunk13.wav : your images must be labeled
audio-chunks/chunk13.wav : Your images must be labeled.
audio-chunks/chunk14.wav : a label indicates that an image contains an object
audio-chunks/chunk14.wav : A label indicates that an image contains an object.
audio-chunks/chunk15.wav : scene or concept
audio-chunks/chunk15.wav : Scene or concept.
audio-chunks/chunk16.wav : as we mentioned earlier a data set needs at least two
defined labels
audio-chunks/chunk16.wav : As we mentioned earlier a data set needs at least two
defined labels.
audio-chunks/chunk17.wav : also each image must have at least one assigned label
that identifies the object seen or Concept in the image
audio-chunks/chunk17.wav : Also each image must have at least one assigned label
that identifies the object seen or concept in the image.
audio-chunks/chunk18.wav : when you apply labels to an image as a whole
audio-chunks/chunk18.wav : When you apply labels to an image as a whole.
audio-chunks/chunk19.wav : these labels are known as Image level labels
audio-chunks/chunk19.wav : These labels are known as image level labels.
audio-chunks/chunk20.wav : they're useful for identifying scenes or concepts
that you want to detect
audio-chunks/chunk20.wav : They're useful for identifying scenes or concepts
that you want to detect.
audio-chunks/chunk21.wav : for example one of the images
audio-chunks/chunk21.wav : For example one of the images.
audio-chunks/chunk22.wav : show us a beach scene from Ko Olina
audio-chunks/chunk22.wav : Show us a beach scene from ko olina.
audio-chunks/chunk23.wav : it's on the island of Oahu in the US state of Hawaii
audio-chunks/chunk23.wav : It's on the island of oahu in the us state of hawaii.
audio-chunks/chunk24.wav : to train a model to detect beaches
audio-chunks/chunk24.wav : To train a model to detect beaches.
audio-chunks/chunk25.wav : you to add a beach label that applies to the entire
image
audio-chunks/chunk25.wav : You to add a beach label that applies to the entire
image.
audio-chunks/chunk26.wav : you can also apply labels to specific areas of an
image that contain an object you want to detect
audio-chunks/chunk26.wav : You can also apply labels to specific areas of an
image that contain an object you want to detect.
audio-chunks/chunk27.wav : for example
audio-chunks/chunk27.wav : For example.
audio-chunks/chunk28.wav : if you want your model to detect Amazon Echo devices
audio-chunks/chunk28.wav : If you want your model to detect amazon echo devices.
audio-chunks/chunk29.wav : it must identify the different types of echo devices
in an image
audio-chunks/chunk29.wav : It must identify the different types of echo devices
in an image.
```

```
audio-chunks/chunk30.wav : the model needs information about where the devices
are located in the image
audio-chunks/chunk30.wav : The model needs information about where the devices
are located in the image.
audio-chunks/chunk31.wav : and it needs a corresponding label that identifies
the type of the device
audio-chunks/chunk31.wav : And it needs a corresponding label that identifies
the type of the device.
audio-chunks/chunk32.wav : disinformation is known as localization information
audio-chunks/chunk32.wav : Disinformation is known as localization information.
audio-chunks/chunk33.wav : the location of the device is expressed as a bounding
box
audio-chunks/chunk33.wav : The location of the device is expressed as a bounding
box.
audio-chunks/chunk34.wav : the example objects with bounding boxes image
audio-chunks/chunk34.wav : The example objects with bounding boxes image.
audio-chunks/chunk35.wav : show us a bounding box that surrounds an Amazon Echo
Dot
audio-chunks/chunk35.wav : Show us a bounding box that surrounds an amazon echo
dot.
audio-chunks/chunk36.wav : the image also contains an Amazon Echo without a
bounding box
audio-chunks/chunk36.wav : The image also contains an amazon echo without a
bounding box.
audio-chunks/chunk37.wav : the output of the labeling process will be a manifest
file
audio-chunks/chunk37.wav : The output of the labeling process will be a manifest
file.
audio-chunks/chunk38.wav : the Manifest file for an image level label typically
contains the label or class name
audio-chunks/chunk38.wav : The manifest file for an image level label typically
contains the label or class name.
audio-chunks/chunk39.wav : along with some metadata about how the image was
labeled
audio-chunks/chunk39.wav : Along with some metadata about how the image was
labeled.
audio-chunks/chunk40.wav : for object detection
audio-chunks/chunk40.wav : For object detection.
audio-chunks/chunk41.wav : the Manifest contains information about each labeled
image
audio-chunks/chunk41.wav : The manifest contains information about each labeled
image.
audio-chunks/chunk42.wav : the bounding box identifies where the object is in
the image
audio-chunks/chunk42.wav : The bounding box identifies where the object is in
the image.
audio-chunks/chunk43.wav : along with the label that the bounding box belongs to
audio-chunks/chunk43.wav : Along with the label that the bounding box belongs
```

```
to.
audio-chunks/chunk44.wav : we've mentioned Amazon sagemaker ground truth a few
times
audio-chunks/chunk44.wav : We've mentioned amazon sagemaker ground truth a few
times.
audio-chunks/chunk45.wav : well now look at what it is and how it might help you
audio-chunks/chunk45.wav : Well now look at what it is and how it might help
you.
audio-chunks/chunk46.wav : with sagemaker ground truth
audio-chunks/chunk46.wav : With sagemaker ground truth.
audio-chunks/chunk47.wav : you can build high-quality training data sets for
your machine learning models
audio-chunks/chunk47.wav : You can build high-quality training data sets for
your machine learning models.
audio-chunks/chunk48.wav : to use it
audio-chunks/chunk48.wav : To use it.
audio-chunks/chunk49.wav : create a data set that needs labeling
audio-chunks/chunk49.wav : Create a data set that needs labeling.
audio-chunks/chunk50.wav : you then provide detailed instructions on what needs
to be labeled and submit the job
audio-chunks/chunk50.wav : You then provide detailed instructions on what needs
to be labeled and submit the job.
audio-chunks/chunk51.wav : you can decide who processes the images to create a
label data set
audio-chunks/chunk51.wav : You can decide who processes the images to create a
label data set.
audio-chunks/chunk52.wav : you can use workers from the Amazon Mechanical Turk
service
audio-chunks/chunk52.wav : You can use workers from the amazon mechanical turk
service.
audio-chunks/chunk53.wav : a vendor company
audio-chunks/chunk53.wav : A vendor company.
audio-chunks/chunk54.wav : or an internal Workforce with machine learning
audio-chunks/chunk54.wav : Or an internal workforce with machine learning.
audio-chunks/chunk55.wav : you can use the label data set output from sagemaker
ground truth to train your own models
audio-chunks/chunk55.wav : You can use the label data set output from sagemaker
ground truth to train your own models.
audio-chunks/chunk56.wav : or you can also use it with Amazon recognition custom
labels
audio-chunks/chunk56.wav : Or you can also use it with amazon recognition custom
labels.
audio-chunks/chunk57.wav : sagemaker ground truth can use active learning to
automate the labeling of your input data
audio-chunks/chunk57.wav : Sagemaker ground truth can use active learning to
automate the labeling of your input data.
audio-chunks/chunk58.wav : Active Learning is a machine learning technique that
identifies data that should be labeled by your workers
```

```
audio-chunks/chunk58.wav : Active learning is a machine learning technique that
identifies data that should be labeled by your workers.
audio-chunks/chunk59.wav : in sagemaker ground truth this functionality is
called automated data labeling
audio-chunks/chunk59.wav : In sagemaker ground truth this functionality is
called automated data labeling.
audio-chunks/chunk60.wav : automated data labeling
audio-chunks/chunk60.wav : Automated data labeling.
audio-chunks/chunk61.wav : can reduce the time and cost it takes to label your
data set
audio-chunks/chunk61.wav : Can reduce the time and cost it takes to label your
data set.
audio-chunks/chunk62.wav : compared to using only human workers
audio-chunks/chunk62.wav : Compared to using only human workers.
audio-chunks/chunk63.wav : when you use automated labeling
audio-chunks/chunk63.wav : When you use automated labeling.
audio-chunks/chunk64.wav : you incur Amazon sagemaker training and inference
costs
audio-chunks/chunk64.wav : You incur amazon sagemaker training and inference
costs.
audio-chunks/chunk65.wav : yes we just said that you can use machine learning to
label the images
audio-chunks/chunk65.wav : Yes we just said that you can use machine learning to
label the images.
audio-chunks/chunk66.wav : but you'll then use for machine learning
audio-chunks/chunk66.wav : But you'll then use for machine learning.
audio-chunks/chunk67.wav : we'll talk to how this works
audio-chunks/chunk67.wav : We'll talk to how this works.
audio-chunks/chunk68.wav : when sagemaker Ground truth starts an automated data
labeling job
audio-chunks/chunk68.wav : When sagemaker ground truth starts an automated data
labeling job.
audio-chunks/chunk69.wav : it selects a random sample of input data or object
audio-chunks/chunk69.wav : It selects a random sample of input data or object.
audio-chunks/chunk70.wav : and send it to human workers
audio-chunks/chunk70.wav : And send it to human workers.
audio-chunks/chunk71.wav : when the label data is returned
audio-chunks/chunk71.wav : When the label data is returned.
audio-chunks/chunk72.wav : sagemaker ground truth uses this data which is the
validation data
audio-chunks/chunk72.wav : Sagemaker ground truth uses this data which is the
validation data.
audio-chunks/chunk73.wav : to validate the models that were trained for
automated data labeling
audio-chunks/chunk73.wav : To validate the models that were trained for
automated data labeling.
audio-chunks/chunk74.wav : sagemaker ground truth runs a batch transform job
audio-chunks/chunk74.wav : Sagemaker ground truth runs a batch transform job.
```

```
audio-chunks/chunk75.wav : using the validated model for inference on the
validation data
audio-chunks/chunk75.wav : Using the validated model for inference on the
validation data.
audio-chunks/chunk76.wav : batch inference produces a confidence score and
quality metric for each object in the validation data
audio-chunks/chunk76.wav : Batch inference produces a confidence score and
quality metric for each object in the validation data.
audio-chunks/chunk77.wav : automated labeling determines that the confidence
score for each object which was produced in step 5
audio-chunks/chunk77.wav : Automated labeling determines that the confidence
score for each object which was produced in step 5.
audio-chunks/chunk78.wav : meet the required threshold which was determined in
step four
audio-chunks/chunk78.wav : Meet the required threshold which was determined in
step four.
audio-chunks/chunk79.wav : if the confidence score meets the threshold
audio-chunks/chunk79.wav : If the confidence score meets the threshold.
audio-chunks/chunk80.wav : the expected quality of automatic labeling exceeds
the requested level of accuracy
audio-chunks/chunk80.wav : The expected quality of automatic labeling exceeds
the requested level of accuracy.
audio-chunks/chunk81.wav : the object is then considered to be automatically
labeled
audio-chunks/chunk81.wav : The object is then considered to be automatically
labeled.
audio-chunks/chunk82.wav : step six
audio-chunks/chunk82.wav : Step six.
audio-chunks/chunk83.wav : produces a data set of unlabeled data with confidence
scores
audio-chunks/chunk83.wav : Produces a data set of unlabeled data with confidence
scores.
audio-chunks/chunk84.wav : sagemaker ground truth selects data points with low
confidence scores from this data set and sends them to human workers for
additional labeling
audio-chunks/chunk84.wav : Sagemaker ground truth selects data points with low
confidence scores from this data set and sends them to human workers for
additional labeling.
audio-chunks/chunk85.wav : sagemaker ground truth then uses the existing human
label data
audio-chunks/chunk85.wav : Sagemaker ground truth then uses the existing human
label data.
audio-chunks/chunk86.wav : and the additional human label data to train a new
model
audio-chunks/chunk86.wav : And the additional human label data to train a new
model.
audio-chunks/chunk87.wav : the process is repeated until the data set is fully
labeled or until another stopping condition is met
```

```
audio-chunks/chunk87.wav : The process is repeated until the data set is fully
labeled or until another stopping condition is met.
audio-chunks/chunk88.wav : for example
audio-chunks/chunk88.wav : For example.
audio-chunks/chunk89.wav : automatic labeling can stop when you meet your budget
for human annotation
audio-chunks/chunk89.wav : Automatic labeling can stop when you meet your budget
for human annotation.
audio-chunks/chunk90.wav : we recommend using automated data labeling on large
data sets
audio-chunks/chunk90.wav : We recommend using automated data labeling on large
data sets.
audio-chunks/chunk91.wav : the minimum number of objects allowed for automated
data labeling is 1,250
audio-chunks/chunk91.wav : The minimum number of objects allowed for automated
data labeling is 1,250.
audio-chunks/chunk92.wav : however we strongly suggest providing a minimum of
$5,000 object
audio-chunks/chunk92.wav : However we strongly suggest providing a minimum of
$5,000 object.
audio-chunks/chunk93.wav : that's it for part 2 of this section will see you
again for part 3 where we'll review how to evaluate and improve your model
audio-chunks/chunk93.wav : That's it for part 2 of this section will see you
again for part 3 where we'll review how to evaluate and improve your model.
audio-chunks/chunk1.wav : hi and welcome to module 6 of AWS Academy machine
learning introduction to natural language processing
audio-chunks/chunk1.wav : Hi and welcome to module 6 of aws academy machine
learning introduction to natural language processing.
audio-chunks/chunk2.wav : in this module we'll introduce natural language
processing which is also known as NLP
audio-chunks/chunk2.wav : In this module we'll introduce natural language
processing which is also known as nlp.
audio-chunks/chunk3.wav : this section includes a description of the major
challenges faced by NLP
audio-chunks/chunk3.wav : This section includes a description of the major
challenges faced by nlp.
audio-chunks/chunk4.wav : and the overall development process for NLP
applications
audio-chunks/chunk4.wav : And the overall development process for nlp
applications.
audio-chunks/chunk5.wav : will then review 5 AWS Services you can use
audio-chunks/chunk5.wav : Will then review 5 aws services you can use.
audio-chunks/chunk6.wav : to speed up the development of MLP based applications
audio-chunks/chunk6.wav : To speed up the development of mlp based applications.
audio-chunks/chunk7.wav : after completing this module you should be able to
audio-chunks/chunk7.wav : After completing this module you should be able to.
audio-chunks/chunk8.wav : describe the NLP use cases that are solved by using
managed Amazon ml services
```

```
audio-chunks/chunk8.wav : Describe the nlp use cases that are solved by using
managed amazon ml services.
Error:
audio-chunks/chunk10.wav : describe the managed Amazon ml services available for
NLP
audio-chunks/chunk10.wav : Describe the managed amazon ml services available for
nlp.
audio-chunks/chunk11.wav : let's get started
audio-chunks/chunk11.wav : Let's get started.
Error:
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring wrangling time series data
audio-chunks/chunk2.wav : Will continue exploring wrangling time series data.
audio-chunks/chunk3.wav : seasonality in data is any kind of repeating
observation
audio-chunks/chunk3.wav : Seasonality in data is any kind of repeating
observation.
audio-chunks/chunk4.wav : where the frequency of the observation is stable
audio-chunks/chunk4.wav : Where the frequency of the observation is stable.
audio-chunks/chunk5.wav : for example in sales you typically see higher sales at
the end of a quarter and into the fourth quarter
audio-chunks/chunk5.wav : For example in sales you typically see higher sales at
the end of a quarter and into the fourth quarter.
audio-chunks/chunk6.wav : consumer retail sees even higher sales in the fourth
quarter
audio-chunks/chunk6.wav : Consumer retail sees even higher sales in the fourth
quarter.
audio-chunks/chunk7.wav : be aware that data can have multiple types of
seasonality in the same data set
audio-chunks/chunk7.wav : Be aware that data can have multiple types of
seasonality in the same data set.
audio-chunks/chunk8.wav : there are many times when you should incorporate
seasonality information into your forecast
audio-chunks/chunk8.wav : There are many times when you should incorporate
seasonality information into your forecast.
audio-chunks/chunk9.wav : for instance localized holidays are a good example for
sales
audio-chunks/chunk9.wav : For instance localized holidays are a good example for
sales.
audio-chunks/chunk10.wav : the chart shows that the total revenue generated by
arcades has a strong correlation with the number of computer science doctorates
awarded in the US
audio-chunks/chunk10.wav : The chart shows that the total revenue generated by
arcades has a strong correlation with the number of computer science doctorates
awarded in the us.
audio-chunks/chunk11.wav : but correlations do not mean causation
audio-chunks/chunk11.wav : But correlations do not mean causation.
```

```
audio-chunks/chunk12.wav : if you disagree
audio-chunks/chunk12.wav : If you disagree.
audio-chunks/chunk13.wav : see the source for the chart
audio-chunks/chunk13.wav : See the source for the chart.
audio-chunks/chunk14.wav : there are many other correlations plotted on the site
and none of them make any sense
audio-chunks/chunk14.wav : There are many other correlations plotted on the site
and none of them make any sense.
audio-chunks/chunk15.wav : with your own data be careful that you're not seeing
and acting on correlations that don't have meaning in the real world
audio-chunks/chunk15.wav : With your own data be careful that you're not seeing
and acting on correlations that don't have meaning in the real world.
audio-chunks/chunk16.wav : here's an experiment
audio-chunks/chunk16.wav : Here's an experiment.
audio-chunks/chunk17.wav : if you generate two random time series data sets of
numbers between 0 and 1
audio-chunks/chunk17.wav : If you generate two random time series data sets of
numbers between 0 and 1.
audio-chunks/chunk18.wav : you'll find that they have a very low correlation
audio-chunks/chunk18.wav : You'll find that they have a very low correlation.
audio-chunks/chunk19.wav : but if you introduce the same slope to both data set
audio-chunks/chunk19.wav : But if you introduce the same slope to both data set.
audio-chunks/chunk20.wav : you'll see a very strong correlation
audio-chunks/chunk20.wav : You'll see a very strong correlation.
audio-chunks/chunk21.wav : you need to know how stable a system is
audio-chunks/chunk21.wav : You need to know how stable a system is.
audio-chunks/chunk22.wav : the level of stability or stationarity
audio-chunks/chunk22.wav : The level of stability or stationarity.
audio-chunks/chunk23.wav : can inform how much you should expect the system's
past Behavior to inform future Behavior
audio-chunks/chunk23.wav : Can inform how much you should expect the system's
past behavior to inform future behavior.
audio-chunks/chunk24.wav : a system with low stability won't be successful at
predicting the future
audio-chunks/chunk24.wav : A system with low stability won't be successful at
predicting the future.
audio-chunks/chunk25.wav : you'll often want to determine the trend for a Time
series
audio-chunks/chunk25.wav : You'll often want to determine the trend for a time
series.
audio-chunks/chunk26.wav : but if you adjust the series for the trend
audio-chunks/chunk26.wav : But if you adjust the series for the trend.
audio-chunks/chunk27.wav : it can be difficult to compare it with another series
that was also adjusted for the trend
audio-chunks/chunk27.wav : It can be difficult to compare it with another series
that was also adjusted for the trend.
audio-chunks/chunk28.wav : this is because the trends might dominate the values
in the series
```

```
audio-chunks/chunk28.wav : This is because the trends might dominate the values
in the series.
audio-chunks/chunk29.wav : this could then lead to overestimates in correlation
between the two series like we discussed previously
audio-chunks/chunk29.wav : This could then lead to overestimates in correlation
between the two series like we discussed previously.
audio-chunks/chunk30.wav : autocorrelation is one of the special problems you
face with time series data
audio-chunks/chunk30.wav : Autocorrelation is one of the special problems you
face with time series data.
audio-chunks/chunk31.wav : as you seen in other machine learning problems
audio-chunks/chunk31.wav : As you seen in other machine learning problems.
audio-chunks/chunk32.wav : the goal of building an ml model is to make sure
you're separating the signal from the noise
audio-chunks/chunk32.wav : The goal of building an ml model is to make sure
you're separating the signal from the noise.
audio-chunks/chunk33.wav : autocorrelation is a form of noise
audio-chunks/chunk33.wav : Autocorrelation is a form of noise.
audio-chunks/chunk34.wav : because separate observations aren't independent of
each other
audio-chunks/chunk34.wav : Because separate observations aren't independent of
each other.
audio-chunks/chunk35.wav : a Time series with autocorrelation might overstate
the accuracy of the model that's produced
audio-chunks/chunk35.wav : A time series with autocorrelation might overstate
the accuracy of the model that's produced.
audio-chunks/chunk36.wav : some of the algorithms you'll look at in this module
can help correct for autocorrelation
audio-chunks/chunk36.wav : Some of the algorithms you'll look at in this module
can help correct for autocorrelation.
audio-chunks/chunk37.wav : these factors along with seasonality
audio-chunks/chunk37.wav : These factors along with seasonality.
audio-chunks/chunk38.wav : will influence the model you'll select a produce your
forecast
audio-chunks/chunk38.wav : Will influence the model you'll select a produce your
forecast.
audio-chunks/chunk39.wav : some algorithms handle seasonality and
autocorrelation but others do not
audio-chunks/chunk39.wav : Some algorithms handle seasonality and
autocorrelation but others do not.
audio-chunks/chunk40.wav : pandas was developed with financial data analysis in
mind
audio-chunks/chunk40.wav : Pandas was developed with financial data analysis in
mind.
audio-chunks/chunk41.wav : as such it's good at handling time series data
audio-chunks/chunk41.wav : As such it's good at handling time series data.
Error:
audio-chunks/chunk43.wav : you can set the index for your pandas dataframe to be
```

```
a date time
audio-chunks/chunk43.wav : You can set the index for your pandas dataframe to be
a date time.
audio-chunks/chunk44.wav : you can then use date and time to select your data
audio-chunks/chunk44.wav : You can then use date and time to select your data.
audio-chunks/chunk45.wav : you can use ranges that contain partial dates
audio-chunks/chunk45.wav : You can use ranges that contain partial dates.
audio-chunks/chunk46.wav : you can also extract date Parts such as
audio-chunks/chunk46.wav : You can also extract date parts such as.
Error:
audio-chunks/chunk48.wav : weekday name and more
audio-chunks/chunk48.wav : Weekday name and more.
audio-chunks/chunk49.wav : for grouping and resampling tasks
audio-chunks/chunk49.wav : For grouping and resampling tasks.
audio-chunks/chunk50.wav : pandas has built-in functions to do both
audio-chunks/chunk50.wav : Pandas has built-in functions to do both.
Error:
audio-chunks/chunk52.wav : pandas can give you insights into autocorrelation
audio-chunks/chunk52.wav : Pandas can give you insights into autocorrelation.
audio-chunks/chunk53.wav : for more information about pandas in the time series
refer to the pandas documentation
audio-chunks/chunk53.wav : For more information about pandas in the time series
refer to the pandas documentation.
audio-chunks/chunk54.wav : one of the tasks in building a forecasting
application
audio-chunks/chunk54.wav : One of the tasks in building a forecasting
application.
audio-chunks/chunk55.wav : is to choose an appropriate algorithm
audio-chunks/chunk55.wav : Is to choose an appropriate algorithm.
audio-chunks/chunk56.wav : your choice of algorithm should be determined by the
type of data set you're using
audio-chunks/chunk56.wav : Your choice of algorithm should be determined by the
type of data set you're using.
audio-chunks/chunk57.wav : and the features of that data
audio-chunks/chunk57.wav : And the features of that data.
audio-chunks/chunk58.wav : Amazon forecast supports these five algorithms but
there are others
audio-chunks/chunk58.wav : Amazon forecast supports these five algorithms but
there are others.
audio-chunks/chunk59.wav : each algorithm can handle data with slightly
different characteristics
audio-chunks/chunk59.wav : Each algorithm can handle data with slightly
different characteristics.
audio-chunks/chunk60.wav : for example
audio-chunks/chunk60.wav : For example.
audio-chunks/chunk61.wav : take Auto regressive integrated moving average which
is also known as arima
audio-chunks/chunk61.wav : Take auto regressive integrated moving average which
```

```
is also known as arima.
audio-chunks/chunk62.wav : it removes Auto correlations that can influence the
pattern of observations
audio-chunks/chunk62.wav : It removes auto correlations that can influence the
pattern of observations.
audio-chunks/chunk63.wav : or take exponential smoothing which is also known as
ETS
audio-chunks/chunk63.wav : Or take exponential smoothing which is also known as
ets.
audio-chunks/chunk64.wav : this algorithm is useful for data sets with
seasonality
audio-chunks/chunk64.wav : This algorithm is useful for data sets with
seasonality.
audio-chunks/chunk65.wav : you can find out more about these algorithms in the
Amazon forecast documentation
audio-chunks/chunk65.wav : You can find out more about these algorithms in the
amazon forecast documentation.
audio-chunks/chunk66.wav : some key takeaways from this section of the module
include
audio-chunks/chunk66.wav : Some key takeaways from this section of the module
include.
audio-chunks/chunk67.wav : time series data is sequence data that includes a
time element which makes it different from regular data sets
audio-chunks/chunk67.wav : Time series data is sequence data that includes a
time element which makes it different from regular data sets.
audio-chunks/chunk68.wav : some of the time challenges include
audio-chunks/chunk68.wav : Some of the time challenges include.
audio-chunks/chunk69.wav : dealing with different time formats
audio-chunks/chunk69.wav : Dealing with different time formats.
audio-chunks/chunk70.wav : handling missing data through downsampling upsampling
and smoothing
audio-chunks/chunk70.wav : Handling missing data through downsampling upsampling
and smoothing.
audio-chunks/chunk71.wav : dealing with seasonality such as weekdays and yearly
Cycles
audio-chunks/chunk71.wav : Dealing with seasonality such as weekdays and yearly
cycles.
audio-chunks/chunk72.wav : avoiding bad correlations
audio-chunks/chunk72.wav : Avoiding bad correlations.
audio-chunks/chunk73.wav : pandas has excellent time series support with
functions for dealing with time
audio-chunks/chunk73.wav : Pandas has excellent time series support with
functions for dealing with time.
audio-chunks/chunk74.wav : there are five algorithms used by Amazon forecast
audio-chunks/chunk74.wav : There are five algorithms used by amazon forecast.
audio-chunks/chunk75.wav : arim
audio-chunks/chunk75.wav : Arim.
audio-chunks/chunk76.wav : deep are Plus
```

```
audio-chunks/chunk76.wav : Deep are plus.
audio-chunks/chunk77.wav : ETS
audio-chunks/chunk77.wav : Ets.
audio-chunks/chunk78.wav : NTTS and profit
audio-chunks/chunk78.wav : Ntts and profit.
audio-chunks/chunk79.wav : that's it for this section
audio-chunks/chunk79.wav : That's it for this section.
audio-chunks/chunk80.wav : we'll see you in the next video
audio-chunks/chunk80.wav : We'll see you in the next video.
audio-chunks/chunk1.wav : hi and welcome back
audio-chunks/chunk1.wav : Hi and welcome back.
audio-chunks/chunk2.wav : in this section we'll look at how you can use Amazon
forecast
audio-chunks/chunk2.wav : In this section we'll look at how you can use amazon
forecast.
audio-chunks/chunk3.wav : to create a predictor and generate forecast
audio-chunks/chunk3.wav : To create a predictor and generate forecast.
audio-chunks/chunk4.wav : when you generate forecast
audio-chunks/chunk4.wav : When you generate forecast.
audio-chunks/chunk5.wav : you can apply the machine learning development
pipeline you've seen throughout this course
audio-chunks/chunk5.wav : You can apply the machine learning development
pipeline you've seen throughout this course.
audio-chunks/chunk6.wav : but you still need data
audio-chunks/chunk6.wav : But you still need data.
audio-chunks/chunk7.wav : you need to import as much data as you have
audio-chunks/chunk7.wav : You need to import as much data as you have.
audio-chunks/chunk8.wav : both historical data and related data
audio-chunks/chunk8.wav : Both historical data and related data.
audio-chunks/chunk9.wav : you want to do some basic evaluation and feature
engineering before you use the data to train a model so you can meet the
requirements of Amazon forecast
audio-chunks/chunk9.wav : You want to do some basic evaluation and feature
engineering before you use the data to train a model so you can meet the
requirements of amazon forecast.
audio-chunks/chunk10.wav : to train a predictor
audio-chunks/chunk10.wav : To train a predictor.
audio-chunks/chunk11.wav : you need to choose an algorithm
audio-chunks/chunk11.wav : You need to choose an algorithm.
audio-chunks/chunk12.wav : if you're not sure which algorithm is the best for
your data Amazon forecast can choose for you
audio-chunks/chunk12.wav : If you're not sure which algorithm is the best for
your data amazon forecast can choose for you.
audio-chunks/chunk13.wav : to do this
audio-chunks/chunk13.wav : To do this.
audio-chunks/chunk14.wav : Select Auto ml as your algorithm
audio-chunks/chunk14.wav : Select auto ml as your algorithm.
audio-chunks/chunk15.wav : you also need to select a domain for your data
```

```
audio-chunks/chunk15.wav : You also need to select a domain for your data.
audio-chunks/chunk16.wav : if you're not sure what the best fit is
audio-chunks/chunk16.wav : If you're not sure what the best fit is.
audio-chunks/chunk17.wav : you can also select a custom domain
audio-chunks/chunk17.wav : You can also select a custom domain.
audio-chunks/chunk18.wav : domains have specific types of data they require
audio-chunks/chunk18.wav : Domains have specific types of data they require.
audio-chunks/chunk19.wav : when you have a trained model
audio-chunks/chunk19.wav : When you have a trained model.
audio-chunks/chunk20.wav : you can then use the model to make a forecast using
an input data set group
audio-chunks/chunk20.wav : You can then use the model to make a forecast using
an input data set group.
audio-chunks/chunk21.wav : after you've generated a forecast you can query the
forecast
audio-chunks/chunk21.wav : After you've generated a forecast you can query the
forecast.
audio-chunks/chunk22.wav : you can also export it to a bucket in Amazon S3
audio-chunks/chunk22.wav : You can also export it to a bucket in amazon s3.
audio-chunks/chunk23.wav : and finally you can encrypt the data in the forecast
before exporting it
audio-chunks/chunk23.wav : And finally you can encrypt the data in the forecast
before exporting it.
audio-chunks/chunk24.wav : the overall process for working with Amazon forecast
audio-chunks/chunk24.wav : The overall process for working with amazon forecast.
audio-chunks/chunk25.wav : is to import historical and related data
audio-chunks/chunk25.wav : Is to import historical and related data.
audio-chunks/chunk26.wav : Amazon forecast in specs the data
audio-chunks/chunk26.wav : Amazon forecast in specs the data.
audio-chunks/chunk27.wav : identifies key data
audio-chunks/chunk27.wav : Identifies key data.
audio-chunks/chunk28.wav : and selects an appropriate algorithm
audio-chunks/chunk28.wav : And selects an appropriate algorithm.
audio-chunks/chunk29.wav : it uses the algorithm to train and optimize a custom
model and produce a predictor
audio-chunks/chunk29.wav : It uses the algorithm to train and optimize a custom
model and produce a predictor.
audio-chunks/chunk30.wav : you create forecast
audio-chunks/chunk30.wav : You create forecast.
audio-chunks/chunk31.wav : by applying the predictor to your data set
audio-chunks/chunk31.wav : By applying the predictor to your data set.
audio-chunks/chunk32.wav : you can then retrieve these forecast
audio-chunks/chunk32.wav : You can then retrieve these forecast.
audio-chunks/chunk33.wav : in the AWS Management console
audio-chunks/chunk33.wav : In the aws management console.
audio-chunks/chunk34.wav : or you can export the forecast as comma delimited
files
audio-chunks/chunk34.wav : Or you can export the forecast as comma delimited
```

```
files.
audio-chunks/chunk35.wav : you can also use an API and AWS CLI commands to
create and retrieve forecasts
audio-chunks/chunk35.wav : You can also use an api and aws cli commands to
create and retrieve forecasts.
audio-chunks/chunk36.wav : when you work with Amazon forecast
audio-chunks/chunk36.wav : When you work with amazon forecast.
audio-chunks/chunk37.wav : you can select the domain you're working in
audio-chunks/chunk37.wav : You can select the domain you're working in.
audio-chunks/chunk38.wav : there are domains ranging from retail to web traffic
audio-chunks/chunk38.wav : There are domains ranging from retail to web traffic.
audio-chunks/chunk39.wav : and there's also a custom option for everything else
audio-chunks/chunk39.wav : And there's also a custom option for everything else.
audio-chunks/chunk40.wav : by selecting a domain you improve the efficiency of
the predictor
audio-chunks/chunk40.wav : By selecting a domain you improve the efficiency of
the predictor.
audio-chunks/chunk41.wav : each domain has specific types of data that you'll
Supply when you build the predictor
audio-chunks/chunk41.wav : Each domain has specific types of data that you'll
supply when you build the predictor.
audio-chunks/chunk42.wav : for example the retail domain expects data for
audio-chunks/chunk42.wav : For example the retail domain expects data for.
audio-chunks/chunk43.wav : the item identifiers
audio-chunks/chunk43.wav : The item identifiers.
audio-chunks/chunk44.wav : a timestamp for the observation
audio-chunks/chunk44.wav : A timestamp for the observation.
audio-chunks/chunk45.wav : the number of sales for that item
audio-chunks/chunk45.wav : The number of sales for that item.
audio-chunks/chunk46.wav : and the specified timestamp
audio-chunks/chunk46.wav : And the specified timestamp.
audio-chunks/chunk47.wav : here's an example of the data you need to provide for
a retail demand forecast
audio-chunks/chunk47.wav : Here's an example of the data you need to provide for
a retail demand forecast.
audio-chunks/chunk48.wav : for the time series you need
audio-chunks/chunk48.wav : For the time series you need.
audio-chunks/chunk49.wav : the time when the transaction took place ideally in
UTC format
audio-chunks/chunk49.wav : The time when the transaction took place ideally in
utc format.
audio-chunks/chunk50.wav : the item id of the item
audio-chunks/chunk50.wav : The item id of the item.
audio-chunks/chunk51.wav : and how many items were sold
audio-chunks/chunk51.wav : And how many items were sold.
audio-chunks/chunk52.wav : the metadata for the item might include the category
audio-chunks/chunk52.wav : The metadata for the item might include the category.
audio-chunks/chunk53.wav : the item color and other attributes
```

```
audio-chunks/chunk53.wav : The item color and other attributes.
audio-chunks/chunk54.wav : the link back to the time series data will be only
the item ID
audio-chunks/chunk54.wav : The link back to the time series data will be only
the item id.
audio-chunks/chunk55.wav : because item metadata typically doesn't change
audio-chunks/chunk55.wav : Because item metadata typically doesn't change.
audio-chunks/chunk56.wav : related data for creating a more useful forecast
audio-chunks/chunk56.wav : Related data for creating a more useful forecast.
audio-chunks/chunk57.wav : could include the sales price
audio-chunks/chunk57.wav : Could include the sales price.
audio-chunks/chunk58.wav : or other promotion data
audio-chunks/chunk58.wav : Or other promotion data.
audio-chunks/chunk59.wav : to link this back to the item you must include the
timestamp and the item ID
audio-chunks/chunk59.wav : To link this back to the item you must include the
timestamp and the item id.
audio-chunks/chunk60.wav : here's an example of the data you need to provide for
a web traffic forecast
audio-chunks/chunk60.wav : Here's an example of the data you need to provide for
a web traffic forecast.
audio-chunks/chunk61.wav : for the time series you need the web page ID
audio-chunks/chunk61.wav : For the time series you need the web page id.
audio-chunks/chunk62.wav : the number of page views per month
audio-chunks/chunk62.wav : The number of page views per month.
audio-chunks/chunk63.wav : and the timestamp
audio-chunks/chunk63.wav : And the timestamp.
audio-chunks/chunk64.wav : related data for creating a more useful forecast
could include the page category such as navigation or content category
audio-chunks/chunk64.wav : Related data for creating a more useful forecast
could include the page category such as navigation or content category.
audio-chunks/chunk65.wav : you'll also need the geographic identifier for the
web client
audio-chunks/chunk65.wav : You'll also need the geographic identifier for the
web client.
audio-chunks/chunk66.wav : for metadata you might also need to provide the
region
audio-chunks/chunk66.wav : For metadata you might also need to provide the
region.
audio-chunks/chunk67.wav : and the sales promotion information
audio-chunks/chunk67.wav : And the sales promotion information.
audio-chunks/chunk68.wav : Amazon forecast predictors use an algorithm to train
a model
audio-chunks/chunk68.wav : Amazon forecast predictors use an algorithm to train
a model.
audio-chunks/chunk69.wav : they then use the model to make a forecast using an
input data set group
audio-chunks/chunk69.wav : They then use the model to make a forecast using an
```

```
input data set group.
audio-chunks/chunk70.wav : to help you get started
audio-chunks/chunk70.wav : To help you get started.
audio-chunks/chunk71.wav : Amazon forecast provides predefined algorithms
audio-chunks/chunk71.wav : Amazon forecast provides predefined algorithms.
audio-chunks/chunk72.wav : arima
audio-chunks/chunk72.wav : Arima.
audio-chunks/chunk73.wav : deep are Plus
audio-chunks/chunk73.wav : Deep are plus.
audio-chunks/chunk74.wav : ETS
audio-chunks/chunk74.wav : Ets.
audio-chunks/chunk75.wav : NBT
audio-chunks/chunk75.wav : Nbt.
audio-chunks/chunk76.wav : and profit
audio-chunks/chunk76.wav : And profit.
audio-chunks/chunk77.wav : you can also use the auto ml feature
audio-chunks/chunk77.wav : You can also use the auto ml feature.
audio-chunks/chunk78.wav : it will try all the algorithms to see which ones are
the best at predicting data
audio-chunks/chunk78.wav : It will try all the algorithms to see which ones are
the best at predicting data.
audio-chunks/chunk79.wav : when you prepare data for training and machine
learning you typically hold back data to use when you validate and score the
model
audio-chunks/chunk79.wav : When you prepare data for training and machine
learning you typically hold back data to use when you validate and score the
model.
audio-chunks/chunk80.wav : the data that you hold back is usually a random
sample of your available data
audio-chunks/chunk80.wav : The data that you hold back is usually a random
sample of your available data.
audio-chunks/chunk81.wav : with time series data you must process your data
differently because of a correlation between time
audio-chunks/chunk81.wav : With time series data you must process your data
differently because of a correlation between time.
audio-chunks/chunk82.wav : when you import your data
audio-chunks/chunk82.wav : When you import your data.
audio-chunks/chunk83.wav : Amazon forecast breaks it into training and test data
sets which the diagram shows
audio-chunks/chunk83.wav : Amazon forecast breaks it into training and test data
sets which the diagram shows.
audio-chunks/chunk84.wav : the training data is used to train the model
audio-chunks/chunk84.wav : The training data is used to train the model.
audio-chunks/chunk85.wav : which is then tested against the data that was held
back
audio-chunks/chunk85.wav : Which is then tested against the data that was held
back.
audio-chunks/chunk86.wav : you can specify multiple back test windows
```

```
audio-chunks/chunk86.wav : You can specify multiple back test windows.
audio-chunks/chunk87.wav : which will split the data multiple times
audio-chunks/chunk87.wav : Which will split the data multiple times.
audio-chunks/chunk88.wav : train the model
audio-chunks/chunk88.wav : Train the model.
audio-chunks/chunk89.wav : and use metrics to determine which model gives the
best results
audio-chunks/chunk89.wav : And use metrics to determine which model gives the
best results.
audio-chunks/chunk90.wav : the default back test window is one
audio-chunks/chunk90.wav : The default back test window is one.
audio-chunks/chunk91.wav : you can change how Amazon forecast splits the data by
setting the back test window offset parameter when you create the predictor
audio-chunks/chunk91.wav : You can change how amazon forecast splits the data by
setting the back test window offset parameter when you create the predictor.
audio-chunks/chunk92.wav : if you don't set this value the algorithms use
default values
audio-chunks/chunk92.wav : If you don't set this value the algorithms use
default values.
audio-chunks/chunk93.wav : after you've trained a model
audio-chunks/chunk93.wav : After you've trained a model.
audio-chunks/chunk94.wav : you will need to measure it accuracy which you'll
learn about next
audio-chunks/chunk94.wav : You will need to measure it accuracy which you'll
learn about next.
audio-chunks/chunk95.wav : the first Amazon forecast evaluation metric is the
weighted quantile loss or W quantile Los
audio-chunks/chunk95.wav : The first amazon forecast evaluation metric is the
weighted quantile loss or w quantile los.
audio-chunks/chunk96.wav : when Amazon forecast creates a forecast
audio-chunks/chunk96.wav : When amazon forecast creates a forecast.
audio-chunks/chunk97.wav : it provides probabilistic predictions at three
distinct quantile
audio-chunks/chunk97.wav : It provides probabilistic predictions at three
distinct quantile.
audio-chunks/chunk98.wav : 10%
audio-chunks/chunk98.wav : 10%.
audio-chunks/chunk99.wav : 50% and 90%
audio-chunks/chunk99.wav : 50% and 90%.
audio-chunks/chunk100.wav : these prediction quantiles show you how much
uncertainty is associated with each forecast
audio-chunks/chunk100.wav : These prediction quantiles show you how much
uncertainty is associated with each forecast.
audio-chunks/chunk101.wav : a P10 quantile predicts that 10% of the time the
True Value will be less than the predicted value
audio-chunks/chunk101.wav : A p10 quantile predicts that 10% of the time the
true value will be less than the predicted value.
audio-chunks/chunk102.wav : for example
```

```
audio-chunks/chunk102.wav : For example.
audio-chunks/chunk103.wav : suppose that you are retailer
audio-chunks/chunk103.wav : Suppose that you are retailer.
audio-chunks/chunk104.wav : you want to forecast product demand for winter
gloves that sell well only during the fall and winter
audio-chunks/chunk104.wav : You want to forecast product demand for winter
gloves that sell well only during the fall and winter.
audio-chunks/chunk105.wav : say that you don't have sufficient storage space and
the cost of invested capital is high
audio-chunks/chunk105.wav : Say that you don't have sufficient storage space and
the cost of invested capital is high.
audio-chunks/chunk106.wav : was it the price of being overstocked on winter
gloves concerns you
audio-chunks/chunk106.wav : Was it the price of being overstocked on winter
gloves concerns you.
audio-chunks/chunk107.wav : then you might use the P10 quantile to order a
relatively low number of winter gloves
audio-chunks/chunk107.wav : Then you might use the p10 quantile to order a
relatively low number of winter gloves.
audio-chunks/chunk108.wav : you know that the P10 forecast overestimates the
demand for your winter gloves only 10% of the time
audio-chunks/chunk108.wav : You know that the p10 forecast overestimates the
demand for your winter gloves only 10% of the time.
audio-chunks/chunk109.wav : so you'll be sold out of your winter gloves for 90%
of the time
audio-chunks/chunk109.wav : So you'll be sold out of your winter gloves for 90%
of the time.
audio-chunks/chunk110.wav : a p50 quantile
audio-chunks/chunk110.wav : A p50 quantile.
audio-chunks/chunk111.wav : predicts that 50% of the time the True Value will be
less than the predicted value
audio-chunks/chunk111.wav : Predicts that 50% of the time the true value will be
less than the predicted value.
audio-chunks/chunk112.wav : continuing the winter gloves example
audio-chunks/chunk112.wav : Continuing the winter gloves example.
audio-chunks/chunk113.wav : say you know that there will be a moderate amount of
demand for the gloves
audio-chunks/chunk113.wav : Say you know that there will be a moderate amount of
demand for the gloves.
audio-chunks/chunk114.wav : and you aren't concerned about being overstocked
audio-chunks/chunk114.wav : And you aren't concerned about being overstocked.
audio-chunks/chunk115.wav : then you might choose to use the p50 quantile to
order gloves
audio-chunks/chunk115.wav : Then you might choose to use the p50 quantile to
order gloves.
audio-chunks/chunk116.wav : a P90 quantile
audio-chunks/chunk116.wav : A p90 quantile.
audio-chunks/chunk117.wav : predicts that 90% of the time the True Value will be
```

```
less than the predicted value
audio-chunks/chunk117.wav : Predicts that 90% of the time the true value will be
less than the predicted value.
audio-chunks/chunk118.wav : suppose you determine that being understocked on
gloves will result in large amounts of lost Revenue
audio-chunks/chunk118.wav : Suppose you determine that being understocked on
gloves will result in large amounts of lost revenue.
audio-chunks/chunk119.wav : for example the cost of not selling gloves is
extremely high or the cost of invested capital is low
audio-chunks/chunk119.wav : For example the cost of not selling gloves is
extremely high or the cost of invested capital is low.
audio-chunks/chunk120.wav : in this case
audio-chunks/chunk120.wav : In this case.
audio-chunks/chunk121.wav : you might choose to use the P90 quantile to order
gloves
audio-chunks/chunk121.wav : You might choose to use the p90 quantile to order
gloves.
audio-chunks/chunk122.wav : Amazon forecast also calculates the associated loss
or error at each quantile
audio-chunks/chunk122.wav : Amazon forecast also calculates the associated loss
or error at each quantile.
audio-chunks/chunk123.wav : weighted quantile lost calculate how far off the
forecast a certain quantile is from actual demand in either direction
audio-chunks/chunk123.wav : Weighted quantile lost calculate how far off the
forecast a certain quantile is from actual demand in either direction.
audio-chunks/chunk124.wav : lower W quantile lost metrics mean that the models
forecasts are more reliable
audio-chunks/chunk124.wav : Lower w quantile lost metrics mean that the models
forecasts are more reliable.
audio-chunks/chunk125.wav : the root mean square error or rmse
audio-chunks/chunk125.wav : The root mean square error or rmse.
audio-chunks/chunk126.wav : is another method for evaluating the reliability of
your forecasts
audio-chunks/chunk126.wav : Is another method for evaluating the reliability of
your forecasts.
audio-chunks/chunk127.wav : like w quantile lost rmse calculates how far off the
forecasted values were from the actual test data
audio-chunks/chunk127.wav : Like w quantile lost rmse calculates how far off the
forecasted values were from the actual test data.
audio-chunks/chunk128.wav : the rmse finds the difference between the actual
Target value in the data set
audio-chunks/chunk128.wav : The rmse finds the difference between the actual
target value in the data set.
audio-chunks/chunk129.wav : and the forecasted value for that time period
audio-chunks/chunk129.wav : And the forecasted value for that time period.
audio-chunks/chunk130.wav : and then squares the differences
audio-chunks/chunk130.wav : And then squares the differences.
audio-chunks/chunk131.wav : the example shows how to calculate rmse
```

```
audio-chunks/chunk131.wav : The example shows how to calculate rmse.
audio-chunks/chunk132.wav : the rmse value represents the standard deviation of
the prediction errors
audio-chunks/chunk132.wav : The rmse value represents the standard deviation of
the prediction errors.
audio-chunks/chunk133.wav : this test is good for forecast validity when the
arrows are mostly of the same
audio-chunks/chunk133.wav : This test is good for forecast validity when the
arrows are mostly of the same.
audio-chunks/chunk134.wav : that is their there aren't many outliers
audio-chunks/chunk134.wav : That is their there aren't many outliers.
audio-chunks/chunk135.wav : lower rmsc metrics indicate that the models
forecasts are more reliable
audio-chunks/chunk135.wav : Lower rmsc metrics indicate that the models
forecasts are more reliable.
audio-chunks/chunk136.wav : here's an example of how a web retailer might use
the accuracy metric
audio-chunks/chunk136.wav : Here's an example of how a web retailer might use
the accuracy metric.
audio-chunks/chunk137.wav : to evaluate a forecast
audio-chunks/chunk137.wav : To evaluate a forecast.
audio-chunks/chunk138.wav : the retailer wants to predict the demand for sales
of a particular brand of shoes
audio-chunks/chunk138.wav : The retailer wants to predict the demand for sales
of a particular brand of shoes.
audio-chunks/chunk139.wav : the input the sales records for this brand into
Amazon forecast to create a predictor
audio-chunks/chunk139.wav : The input the sales records for this brand into
amazon forecast to create a predictor.
audio-chunks/chunk140.wav : the predictor provides a forecasted demand of 1,000
pairs with the P10 p50 and P90 values shown
audio-chunks/chunk140.wav : The predictor provides a forecasted demand of 1,000
pairs with the p10 p50 and p90 values shown.
audio-chunks/chunk141.wav : the weighted quantile lost values indicate that 10%
of the time there will be fewer than 880 pairs sold
audio-chunks/chunk141.wav : The weighted quantile lost values indicate that 10%
of the time there will be fewer than 880 pairs sold.
audio-chunks/chunk142.wav : 50% of the time
audio-chunks/chunk142.wav : 50% of the time.
audio-chunks/chunk143.wav : fewer than 1,050 pairs will be sold
audio-chunks/chunk143.wav : Fewer than 1,050 pairs will be sold.
audio-chunks/chunk144.wav : and 90% of the time
audio-chunks/chunk144.wav : And 90% of the time.
audio-chunks/chunk145.wav : fewer than 1,200 pairs will be sold
audio-chunks/chunk145.wav : Fewer than 1,200 pairs will be sold.
audio-chunks/chunk146.wav : the retailer can then use these values to determine
which level of inventory to hold
audio-chunks/chunk146.wav : The retailer can then use these values to determine
```

```
which level of inventory to hold.
audio-chunks/chunk147.wav : they can base their decision on their assessment of
the risk they won't be able to fulfill orders
audio-chunks/chunk147.wav : They can base their decision on their assessment of
the risk they won't be able to fulfill orders.
audio-chunks/chunk148.wav : or that they'll have excess inventory
audio-chunks/chunk148.wav : Or that they'll have excess inventory.
audio-chunks/chunk149.wav : some key takeaways from the section of the module
include
audio-chunks/chunk149.wav : Some key takeaways from the section of the module
include.
audio-chunks/chunk150.wav : you can use Amazon forecast to train and use a model
for time series data
audio-chunks/chunk150.wav : You can use amazon forecast to train and use a model
for time series data.
audio-chunks/chunk151.wav : there are specific schemas defined for domains such
as retail and ec2 capacity planning
audio-chunks/chunk151.wav : There are specific schemas defined for domains such
as retail and ec2 capacity planning.
audio-chunks/chunk152.wav : or you can use a custom schema
audio-chunks/chunk152.wav : Or you can use a custom schema.
audio-chunks/chunk153.wav : you need to supply at least the time series data but
can also provide metadata and related data to add more information to the model
audio-chunks/chunk153.wav : You need to supply at least the time series data but
can also provide metadata and related data to add more information to the model.
audio-chunks/chunk154.wav : as with most supervised machine learning problems
audio-chunks/chunk154.wav : As with most supervised machine learning problems.
audio-chunks/chunk155.wav : your data is split into training and testing data
audio-chunks/chunk155.wav : Your data is split into training and testing data.
audio-chunks/chunk156.wav : but takes new account the time element
audio-chunks/chunk156.wav : But takes new account the time element.
audio-chunks/chunk157.wav : use rmse and W quantile lost metrics to evaluate the
efficiency of the model
audio-chunks/chunk157.wav : Use rmse and w quantile lost metrics to evaluate the
efficiency of the model.
audio-chunks/chunk158.wav : that's it for this video we'll see you in the next
one
audio-chunks/chunk158.wav : That's it for this video we'll see you in the next
one.
audio-chunks/chunk1.wav : hi welcome back
audio-chunks/chunk1.wav : Hi welcome back.
audio-chunks/chunk2.wav : will continue exploring feature engineering by
reviewing how to clean your data set
audio-chunks/chunk2.wav : Will continue exploring feature engineering by
reviewing how to clean your data set.
audio-chunks/chunk3.wav : in addition to converting string data to numerical
data you'll need to clean your data set for several other potential problem
areas
```

```
audio-chunks/chunk3.wav : In addition to converting string data to numerical
data you'll need to clean your data set for several other potential problem
areas.
audio-chunks/chunk4.wav : before encoding the string data
audio-chunks/chunk4.wav : Before encoding the string data.
audio-chunks/chunk5.wav : make sure the strings are all consistent
audio-chunks/chunk5.wav : Make sure the strings are all consistent.
audio-chunks/chunk6.wav : you'll also need to make sure variables use a
consistent scale
audio-chunks/chunk6.wav : You'll also need to make sure variables use a
consistent scale.
audio-chunks/chunk7.wav : for example if one variable describes the number of
doors in a car
audio-chunks/chunk7.wav : For example if one variable describes the number of
doors in a car.
audio-chunks/chunk8.wav : the scale will probably be between 2 and 8
audio-chunks/chunk8.wav : The scale will probably be between 2 and 8.
audio-chunks/chunk9.wav : what if another variable describes the number of cars
of a particular type sold in the state of California
audio-chunks/chunk9.wav : What if another variable describes the number of cars
of a particular type sold in the state of california.
audio-chunks/chunk10.wav : the scale will probably be in the thousands
audio-chunks/chunk10.wav : The scale will probably be in the thousands.
audio-chunks/chunk11.wav : some data items might also capture more than one
variable in a single value
audio-chunks/chunk11.wav : Some data items might also capture more than one
variable in a single value.
audio-chunks/chunk12.wav : for instance suppose the data set includes variables
that combine safety and maintenance into a single variable such as
audio-chunks/chunk12.wav : For instance suppose the data set includes variables
that combine safety and maintenance into a single variable such as.
audio-chunks/chunk13.wav : say high maintenance
audio-chunks/chunk13.wav : Say high maintenance.
audio-chunks/chunk14.wav : you need to train your machine Learning System for
both variables
audio-chunks/chunk14.wav : You need to train your machine learning system for
both variables.
audio-chunks/chunk15.wav : and also
audio-chunks/chunk15.wav : And also.
audio-chunks/chunk16.wav : split that single variable into two separate
variables
audio-chunks/chunk16.wav : Split that single variable into two separate
variables.
audio-chunks/chunk17.wav : you might also encounter data sets that are missing
data for some variables
audio-chunks/chunk17.wav : You might also encounter data sets that are missing
data for some variables.
audio-chunks/chunk18.wav : and some data sets will include outliers
```

```
audio-chunks/chunk18.wav : And some data sets will include outliers.
audio-chunks/chunk19.wav : will cover techniques for dealing with these
situations in this section
audio-chunks/chunk19.wav : Will cover techniques for dealing with these
situations in this section.
audio-chunks/chunk20.wav : you might find that data is missing
audio-chunks/chunk20.wav : You might find that data is missing.
audio-chunks/chunk21.wav : for example some columns in your data set could be
missing data because of a data collection error
audio-chunks/chunk21.wav : For example some columns in your data set could be
missing data because of a data collection error.
audio-chunks/chunk22.wav : or maybe data wasn't collected on a particular
feature until the data collection process was underway
audio-chunks/chunk22.wav : Or maybe data wasn't collected on a particular
feature until the data collection process was underway.
audio-chunks/chunk23.wav : missing data can make it difficult to accurately
interpret the relationship between the related feature and the target variable
audio-chunks/chunk23.wav : Missing data can make it difficult to accurately
interpret the relationship between the related feature and the target variable.
audio-chunks/chunk24.wav : so regardless of how the data ended up being Miss
audio-chunks/chunk24.wav : So regardless of how the data ended up being miss.
audio-chunks/chunk25.wav : it's important for you to deal with this issue
audio-chunks/chunk25.wav : It's important for you to deal with this issue.
audio-chunks/chunk26.wav : unfortunately most machine learning algorithms can't
handle missing values automatically
audio-chunks/chunk26.wav : Unfortunately most machine learning algorithms can't
handle missing values automatically.
audio-chunks/chunk27.wav : you'll need to use human intelligence to update
missing values with data that's meaningful and relevant to the problem
audio-chunks/chunk27.wav : You'll need to use human intelligence to update
missing values with data that's meaningful and relevant to the problem.
audio-chunks/chunk28.wav : most python libraries for data manipulation include
functions for finding missing data
audio-chunks/chunk28.wav : Most python libraries for data manipulation include
functions for finding missing data.
audio-chunks/chunk29.wav : so how do you decide if you should drop or impute
missing values
audio-chunks/chunk29.wav : So how do you decide if you should drop or impute
missing values.
audio-chunks/chunk30.wav : this question is answered in part by better
understanding how those values came to be missing in the first place
audio-chunks/chunk30.wav : This question is answered in part by better
understanding how those values came to be missing in the first place.
audio-chunks/chunk31.wav : and how much data the missing values represent within
your larger data set
audio-chunks/chunk31.wav : And how much data the missing values represent within
your larger data set.
audio-chunks/chunk32.wav : for instance
```

```
audio-chunks/chunk32.wav : For instance.
audio-chunks/chunk33.wav : say the missing values are randomly spread throughout
your data set and don't represent a larger portion of its respective row or
column
audio-chunks/chunk33.wav : Say the missing values are randomly spread throughout
your data set and don't represent a larger portion of its respective row or
column.
audio-chunks/chunk34.wav : in this case
audio-chunks/chunk34.wav : In this case.
audio-chunks/chunk35.wav : imputation is most likely the better option
audio-chunks/chunk35.wav : Imputation is most likely the better option.
audio-chunks/chunk36.wav : in contrast
audio-chunks/chunk36.wav : In contrast.
audio-chunks/chunk37.wav : say that you have a column a road that has a large
percentage of missing values
audio-chunks/chunk37.wav : Say that you have a column a road that has a large
percentage of missing values.
audio-chunks/chunk38.wav : in this case
audio-chunks/chunk38.wav : In this case.
audio-chunks/chunk39.wav : dropping the entire row or column would be preferred
over amputation
audio-chunks/chunk39.wav : Dropping the entire row or column would be preferred
over amputation.
audio-chunks/chunk40.wav : if you decide to drop rows with missing data
audio-chunks/chunk40.wav : If you decide to drop rows with missing data.
audio-chunks/chunk41.wav : you can use built-in functions to do this
audio-chunks/chunk41.wav : You can use built-in functions to do this.
audio-chunks/chunk42.wav : for example pandas drop na function can drop all rows
with missing data
audio-chunks/chunk42.wav : For example pandas drop na function can drop all rows
with missing data.
audio-chunks/chunk43.wav : or you can drop specific data values by using a
subset
audio-chunks/chunk43.wav : Or you can drop specific data values by using a
subset.
audio-chunks/chunk44.wav : as an alternative to dropping missing values you can
impute values for those missing values
audio-chunks/chunk44.wav : As an alternative to dropping missing values you can
impute values for those missing values.
audio-chunks/chunk45.wav : there are different ways to impute a missing value
audio-chunks/chunk45.wav : There are different ways to impute a missing value.
audio-chunks/chunk46.wav : for categorical values
audio-chunks/chunk46.wav : For categorical values.
audio-chunks/chunk47.wav : the missing value is usually replaced with the mean
the median or the most frequent values
audio-chunks/chunk47.wav : The missing value is usually replaced with the mean
the median or the most frequent values.
audio-chunks/chunk48.wav : for numerical or continuous variables
```

```
audio-chunks/chunk48.wav : For numerical or continuous variables.
audio-chunks/chunk49.wav : the missing value is usually replaced with the mean
or the median
audio-chunks/chunk49.wav : The missing value is usually replaced with the mean
or the median.
audio-chunks/chunk50.wav : you can impute a single row of missing data which is
known as univariate
audio-chunks/chunk50.wav : You can impute a single row of missing data which is
known as univariate.
audio-chunks/chunk51.wav : you can also do this for multiple rows which is known
as multivariate
audio-chunks/chunk51.wav : You can also do this for multiple rows which is known
as multivariate.
audio-chunks/chunk52.wav : will now look at a univariate example
audio-chunks/chunk52.wav : Will now look at a univariate example.
audio-chunks/chunk53.wav : computer function is being used to impute some
missing values
audio-chunks/chunk53.wav : Computer function is being used to impute some
missing values.
audio-chunks/chunk54.wav : it's a fairly small data set but there are two
missing values
audio-chunks/chunk54.wav : It's a fairly small data set but there are two
missing values.
audio-chunks/chunk55.wav : the missing value was imputed by the strategy of the
mean
audio-chunks/chunk55.wav : The missing value was imputed by the strategy of the
mean.
audio-chunks/chunk56.wav : to do this you first calculate the mean
audio-chunks/chunk56.wav : To do this you first calculate the mean.
audio-chunks/chunk57.wav : here is the mean of 3 and 2 which is 2.5
audio-chunks/chunk57.wav : Here is the mean of 3 and 2 which is 2.5.
audio-chunks/chunk58.wav : then you'll impute the mean value for the missing
value
audio-chunks/chunk58.wav : Then you'll impute the mean value for the missing
value.
audio-chunks/chunk59.wav : some data libraries include an impute package that
provides more complex ways to impute data
audio-chunks/chunk59.wav : Some data libraries include an impute package that
provides more complex ways to impute data.
audio-chunks/chunk60.wav : examples include K nearest neighbor
audio-chunks/chunk60.wav : Examples include k nearest neighbor.
audio-chunks/chunk61.wav : soft and cute
audio-chunks/chunk61.wav : Soft and cute.
audio-chunks/chunk62.wav : multiple imputation by chained equations and others
audio-chunks/chunk62.wav : Multiple imputation by chained equations and others.
audio-chunks/chunk63.wav : that's it for part two of this section
audio-chunks/chunk63.wav : That's it for part two of this section.
audio-chunks/chunk64.wav : will see you again for part 3 where we'll review how
```

```
to work with outliers in your data
audio-chunks/chunk64.wav : Will see you again for part 3 where we'll review how
to work with outliers in your data.
audio-chunks/chunk1.wav : hi and welcome back
audio-chunks/chunk1.wav : Hi and welcome back.
audio-chunks/chunk2.wav : this is section 6 and we're going to look at hosting
and using the model
audio-chunks/chunk2.wav : This is section 6 and we're going to look at hosting
and using the model.
audio-chunks/chunk3.wav : in this section we'll look at how you can deploy your
train model so it can be consumed by applications
audio-chunks/chunk3.wav : In this section we'll look at how you can deploy your
train model so it can be consumed by applications.
audio-chunks/chunk4.wav : after you've trained tuned and tested your model
you'll learn more about testing in the next section
audio-chunks/chunk4.wav : After you've trained tuned and tested your model
you'll learn more about testing in the next section.
audio-chunks/chunk5.wav : you're now ready to deploy your model
audio-chunks/chunk5.wav : You're now ready to deploy your model.
audio-chunks/chunk6.wav : if you're thinking that we're looking at the phases
out of order here's why we're discussing deployment now
audio-chunks/chunk6.wav : If you're thinking that we're looking at the phases
out of order here's why we're discussing deployment now.
audio-chunks/chunk7.wav : if you want to test your model and get performance
metrics from it
audio-chunks/chunk7.wav : If you want to test your model and get performance
metrics from it.
audio-chunks/chunk8.wav : you first need to make an inference or prediction from
the model
audio-chunks/chunk8.wav : You first need to make an inference or prediction from
the model.
audio-chunks/chunk9.wav : requires deployment
audio-chunks/chunk9.wav : Requires deployment.
audio-chunks/chunk10.wav : deployment for testing is different from production
audio-chunks/chunk10.wav : Deployment for testing is different from production.
audio-chunks/chunk11.wav : although the mechanics are the same
audio-chunks/chunk11.wav : Although the mechanics are the same.
audio-chunks/chunk12.wav : Amazon sagemaker provides everything you need to host
your model for simple testing and evaluation
audio-chunks/chunk12.wav : Amazon sagemaker provides everything you need to host
your model for simple testing and evaluation.
audio-chunks/chunk13.wav : from a few requests to deployment handling tens of
thousands of requests
audio-chunks/chunk13.wav : From a few requests to deployment handling tens of
thousands of requests.
audio-chunks/chunk14.wav : there are two ways you can deploy your model
audio-chunks/chunk14.wav : There are two ways you can deploy your model.
audio-chunks/chunk15.wav : for single predictions you can deploy your model with
```

```
Amazon sagemaker hosting services
audio-chunks/chunk15.wav : For single predictions you can deploy your model with
amazon sagemaker hosting services.
audio-chunks/chunk16.wav : sagemaker will deploy multiple computers which run
your model behind a load balanced endpoint
audio-chunks/chunk16.wav : Sagemaker will deploy multiple computers which run
your model behind a load balanced endpoint.
audio-chunks/chunk17.wav : applications can call the API at the endpoint to make
predictions
audio-chunks/chunk17.wav : Applications can call the api at the endpoint to make
predictions.
audio-chunks/chunk18.wav : with this model you can scale the number of instances
up or down based on demand
audio-chunks/chunk18.wav : With this model you can scale the number of instances
up or down based on demand.
audio-chunks/chunk19.wav : to get predictions for an entire data set use Amazon
sagemaker batch transform
audio-chunks/chunk19.wav : To get predictions for an entire data set use amazon
sagemaker batch transform.
audio-chunks/chunk20.wav : instead of deploying and maintaining a permanent
endpoint
audio-chunks/chunk20.wav : Instead of deploying and maintaining a permanent
endpoint.
audio-chunks/chunk21.wav : sagemaker will spin up your model and perform the
predictions for the entire data set you provide
audio-chunks/chunk21.wav : Sagemaker will spin up your model and perform the
predictions for the entire data set you provide.
audio-chunks/chunk22.wav : it will then store the results in Amazon S3 before it
shuts down and terminates the compute instances
audio-chunks/chunk22.wav : It will then store the results in amazon s3 before it
shuts down and terminates the compute instances.
audio-chunks/chunk23.wav : it's useful for performing batch predictions when you
test the model
audio-chunks/chunk23.wav : It's useful for performing batch predictions when you
test the model.
audio-chunks/chunk24.wav : you can quickly run your entire validation set
against the model without writing any code to process and collate the individual
results
audio-chunks/chunk24.wav : You can quickly run your entire validation set
against the model without writing any code to process and collate the individual
results.
audio-chunks/chunk25.wav : the goal of the deployment phase
audio-chunks/chunk25.wav : The goal of the deployment phase.
audio-chunks/chunk26.wav : is to provide a managed environment to host models
for providing inference securely and with low latency
audio-chunks/chunk26.wav : Is to provide a managed environment to host models
for providing inference securely and with low latency.
audio-chunks/chunk27.wav : after your model is deployed into production
```

```
audio-chunks/chunk27.wav : After your model is deployed into production.
audio-chunks/chunk28.wav : you should monitor your production data and retrain
your model if necessary
audio-chunks/chunk28.wav : You should monitor your production data and retrain
your model if necessary.
audio-chunks/chunk29.wav : newly deployed models need to reflect the current
production data
audio-chunks/chunk29.wav : Newly deployed models need to reflect the current
production data.
audio-chunks/chunk30.wav : new data is accumulated over time and it could
potentially identify alternative or new outcomes
audio-chunks/chunk30.wav : New data is accumulated over time and it could
potentially identify alternative or new outcomes.
audio-chunks/chunk31.wav : and so deploying a model is not a one-time exercise
audio-chunks/chunk31.wav : And so deploying a model is not a one-time exercise.
audio-chunks/chunk32.wav : instead it's a continuous process
audio-chunks/chunk32.wav : Instead it's a continuous process.
audio-chunks/chunk33.wav : with one click
audio-chunks/chunk33.wav : With one click.
audio-chunks/chunk34.wav : you can deploy your model on Amazon ml instances that
can automatically scale across multiple availability zones for higher redundancy
audio-chunks/chunk34.wav : You can deploy your model on amazon ml instances that
can automatically scale across multiple availability zones for higher
redundancy.
audio-chunks/chunk35.wav : just specify the type of instance and the maximum and
minimum number of instances desired
audio-chunks/chunk35.wav : Just specify the type of instance and the maximum and
minimum number of instances desired.
audio-chunks/chunk36.wav : sagemaker will take care of the rest
audio-chunks/chunk36.wav : Sagemaker will take care of the rest.
audio-chunks/chunk37.wav : it will launch the instances
audio-chunks/chunk37.wav : It will launch the instances.
audio-chunks/chunk38.wav : deploy your model and set up the secure https
endpoint for your application
audio-chunks/chunk38.wav : Deploy your model and set up the secure https
endpoint for your application.
audio-chunks/chunk39.wav : your application only needs to include an API call to
this endpoint to achieve inference with low latency and high throughput
audio-chunks/chunk39.wav : Your application only needs to include an api call to
this endpoint to achieve inference with low latency and high throughput.
audio-chunks/chunk40.wav : with this architecture you can integrate your new
models into your application in minutes
audio-chunks/chunk40.wav : With this architecture you can integrate your new
models into your application in minutes.
audio-chunks/chunk41.wav : because changes to the model no longer need changes
to the application code
audio-chunks/chunk41.wav : Because changes to the model no longer need changes
to the application code.
```

```
audio-chunks/chunk42.wav : sagemaker manages your production compute
infrastructure on your behalf
audio-chunks/chunk42.wav : Sagemaker manages your production compute
infrastructure on your behalf.
audio-chunks/chunk43.wav : it can perform health checks
audio-chunks/chunk43.wav : It can perform health checks.
audio-chunks/chunk44.wav : apply security patches
audio-chunks/chunk44.wav : Apply security patches.
audio-chunks/chunk45.wav : and conduct other routine maintenance
audio-chunks/chunk45.wav : And conduct other routine maintenance.
audio-chunks/chunk46.wav : all with built-in Amazon Cloud watch monitoring and
logging
audio-chunks/chunk46.wav : All with built-in amazon cloud watch monitoring and
logging.
audio-chunks/chunk47.wav : after you've trained the model
audio-chunks/chunk47.wav : After you've trained the model.
audio-chunks/chunk48.wav : you can create the endpoint either in code
audio-chunks/chunk48.wav : You can create the endpoint either in code.
audio-chunks/chunk49.wav : or by using the sagemaker console
audio-chunks/chunk49.wav : Or by using the sagemaker console.
audio-chunks/chunk50.wav : if you're planning to host only a single model you
can create an endpoint for that model
audio-chunks/chunk50.wav : If you're planning to host only a single model you
can create an endpoint for that model.
audio-chunks/chunk51.wav : but if you're planning to host multiple models
audio-chunks/chunk51.wav : But if you're planning to host multiple models.
audio-chunks/chunk52.wav : you need to create a multi-model endpoint
audio-chunks/chunk52.wav : You need to create a multi-model endpoint.
audio-chunks/chunk53.wav : multimodal endpoints provide a scalable and cost-
effective solution for deploying large numbers of models
audio-chunks/chunk53.wav : Multimodal endpoints provide a scalable and cost-
effective solution for deploying large numbers of models.
audio-chunks/chunk54.wav : they use a shared serving container that's enabled to
host multiple models
audio-chunks/chunk54.wav : They use a shared serving container that's enabled to
host multiple models.
audio-chunks/chunk55.wav : this reduces hosting cost by improving endpoint
utilization
audio-chunks/chunk55.wav : This reduces hosting cost by improving endpoint
utilization.
audio-chunks/chunk56.wav : compared to using single model endpoint
audio-chunks/chunk56.wav : Compared to using single model endpoint.
audio-chunks/chunk57.wav : it also reduces deployment overhead
audio-chunks/chunk57.wav : It also reduces deployment overhead.
audio-chunks/chunk58.wav : because sagemaker manages loading models in memory
and scaling the models based on the traffic patterns to them
audio-chunks/chunk58.wav : Because sagemaker manages loading models in memory
and scaling the models based on the traffic patterns to them.
```

```
audio-chunks/chunk59.wav : when you deploy machine learning models into
production to make predictions on new data
audio-chunks/chunk59.wav : When you deploy machine learning models into
production to make predictions on new data.
audio-chunks/chunk60.wav : you need to make sure you apply the same data
processing steps that were used in training to each inference request
audio-chunks/chunk60.wav : You need to make sure you apply the same data
processing steps that were used in training to each inference request.
audio-chunks/chunk61.wav : otherwise you can get incorrect prediction results
audio-chunks/chunk61.wav : Otherwise you can get incorrect prediction results.
audio-chunks/chunk62.wav : by using inference pipelines you can reuse the data
processing steps from model training during infer
audio-chunks/chunk62.wav : By using inference pipelines you can reuse the data
processing steps from model training during infer.
audio-chunks/chunk63.wav : without maintaining two separate copies of the same
code
audio-chunks/chunk63.wav : Without maintaining two separate copies of the same
code.
audio-chunks/chunk64.wav : this helps ensure the accuracy of your predictions
audio-chunks/chunk64.wav : This helps ensure the accuracy of your predictions.
audio-chunks/chunk65.wav : and reduces development overhead
audio-chunks/chunk65.wav : And reduces development overhead.
audio-chunks/chunk66.wav : because sagemaker is a managed service
audio-chunks/chunk66.wav : Because sagemaker is a managed service.
audio-chunks/chunk67.wav : inference pipelines are completely managed
audio-chunks/chunk67.wav : Inference pipelines are completely managed.
audio-chunks/chunk68.wav : when you play the pipeline model the service installs
and runs the sequence of containers on each ec2 instance in the endpoint
audio-chunks/chunk68.wav : When you play the pipeline model the service installs
and runs the sequence of containers on each ec2 instance in the endpoint.
audio-chunks/chunk69.wav : or each batch transform job
audio-chunks/chunk69.wav : Or each batch transform job.
audio-chunks/chunk70.wav : Additionally the sequence of feature processing and
inference runs with low latency because the containers are correlated on the
same ec2 instances
audio-chunks/chunk70.wav : Additionally the sequence of feature processing and
inference runs with low latency because the containers are correlated on the
same ec2 instances.
audio-chunks/chunk71.wav : some key takeaways from the section of the module
include these points
audio-chunks/chunk71.wav : Some key takeaways from the section of the module
include these points.
audio-chunks/chunk72.wav : you can deploy your train model by using sagemaker to
handle API calls from applications
audio-chunks/chunk72.wav : You can deploy your train model by using sagemaker to
handle api calls from applications.
audio-chunks/chunk73.wav : or to perform predictions using a batch
transformation
```

```
audio-chunks/chunk73.wav : Or to perform predictions using a batch
transformation.
audio-chunks/chunk74.wav : the goal of your model is to generate predictions to
answer the business problem
audio-chunks/chunk74.wav : The goal of your model is to generate predictions to
answer the business problem.
audio-chunks/chunk75.wav : be sure that your model can generate good results
before you deploy to production
audio-chunks/chunk75.wav : Be sure that your model can generate good results
before you deploy to production.
audio-chunks/chunk76.wav : finally
audio-chunks/chunk76.wav : Finally.
audio-chunks/chunk77.wav : use multi-model endpoint support to save resources
when you have multiple models to deploy
audio-chunks/chunk77.wav : Use multi-model endpoint support to save resources
when you have multiple models to deploy.
audio-chunks/chunk78.wav : that's it for this section we'll see you in the next
video
audio-chunks/chunk78.wav : That's it for this section we'll see you in the next
video.
audio-chunks/chunk1.wav : hi and welcome back
audio-chunks/chunk1.wav : Hi and welcome back.
audio-chunks/chunk2.wav : this is Section 3
audio-chunks/chunk2.wav : This is section 3.
audio-chunks/chunk3.wav : and we're going to give you a quick high-level
overview of machine learning terminology and a typical workflow
audio-chunks/chunk3.wav : And we're going to give you a quick high-level
overview of machine learning terminology and a typical workflow.
audio-chunks/chunk4.wav : we will cover these topics in more detail later in
this course but for now we'll focus on the larger picture
audio-chunks/chunk4.wav : We will cover these topics in more detail later in
this course but for now we'll focus on the larger picture.
audio-chunks/chunk5.wav : so to begin you should always start with the business
problem you or your team believe could benefit from machine learning
audio-chunks/chunk5.wav : So to begin you should always start with the business
problem you or your team believe could benefit from machine learning.
audio-chunks/chunk6.wav : from there you want to do some problem formulation
audio-chunks/chunk6.wav : From there you want to do some problem formulation.
audio-chunks/chunk7.wav : in this phase one task is to articulate your business
problem and convert it to an ml problem
audio-chunks/chunk7.wav : In this phase one task is to articulate your business
problem and convert it to an ml problem.
audio-chunks/chunk8.wav : after you formulated the problem
audio-chunks/chunk8.wav : After you formulated the problem.
audio-chunks/chunk9.wav : you move on to the data preparation and pre-processing
phase
audio-chunks/chunk9.wav : You move on to the data preparation and pre-processing
phase.
```

```
audio-chunks/chunk10.wav : you pull data from one or more data sources
audio-chunks/chunk10.wav : You pull data from one or more data sources.
audio-chunks/chunk11.wav : these data sources might have differences in data or
types that need to be reconciled so you can form a single cohesive view of your
data
audio-chunks/chunk11.wav : These data sources might have differences in data or
types that need to be reconciled so you can form a single cohesive view of your
data.
audio-chunks/chunk12.wav : you'll need to visualize your data
audio-chunks/chunk12.wav : You'll need to visualize your data.
audio-chunks/chunk13.wav : and use statistics to determine if the data is
consistent and can be used for machine learning
audio-chunks/chunk13.wav : And use statistics to determine if the data is
consistent and can be used for machine learning.
audio-chunks/chunk14.wav : we'll look at some of the data sources later in the
course
audio-chunks/chunk14.wav : We'll look at some of the data sources later in the
course.
audio-chunks/chunk15.wav : this example data has four columns containing data
from three different data sources
audio-chunks/chunk15.wav : This example data has four columns containing data
from three different data sources.
audio-chunks/chunk16.wav : the sources had slightly different ways of
representing data and the results are shown in the table
audio-chunks/chunk16.wav : The sources had slightly different ways of
representing data and the results are shown in the table.
audio-chunks/chunk17.wav : in ml problems
audio-chunks/chunk17.wav : In ml problems.
audio-chunks/chunk18.wav : columns represent features
audio-chunks/chunk18.wav : Columns represent features.
audio-chunks/chunk19.wav : and rose represent instances
audio-chunks/chunk19.wav : And rose represent instances.
audio-chunks/chunk20.wav : there are some issues here with the data in some of
the instances
audio-chunks/chunk20.wav : There are some issues here with the data in some of
the instances.
audio-chunks/chunk21.wav : in some cases you'll need a subject matter expert or
a functional
audio-chunks/chunk21.wav : In some cases you'll need a subject matter expert or
a functional.
audio-chunks/chunk22.wav : understand the authenticity of the data
audio-chunks/chunk22.wav : Understand the authenticity of the data.
audio-chunks/chunk23.wav : for example the date that's represented as 11
audio-chunks/chunk23.wav : For example the date that's represented as 11.
audio-chunks/chunk24.wav : 2 1969
audio-chunks/chunk24.wav : 2 1969.
audio-chunks/chunk25.wav : could be November 2nd or February 11th in the year
1969
```

```
audio-chunks/chunk25.wav : Could be november 2nd or february 11th in the year
1969.
audio-chunks/chunk26.wav : someone who owns or manages the data pool would be
able to clarify this ambiguity
audio-chunks/chunk26.wav : Someone who owns or manages the data pool would be
able to clarify this ambiguity.
audio-chunks/chunk27.wav : also the word male can probably be attributed to an
import issue where sells shifted position
audio-chunks/chunk27.wav : Also the word male can probably be attributed to an
import issue where sells shifted position.
audio-chunks/chunk28.wav : but there could be an outside chance where it's the
actual location
audio-chunks/chunk28.wav : But there could be an outside chance where it's the
actual location.
audio-chunks/chunk29.wav : Molly a city that's the capital of the Republic of
Maldives
audio-chunks/chunk29.wav : Molly a city that's the capital of the republic of
maldives.
audio-chunks/chunk30.wav : at times this error identification isn't as simple
and you'll need an SME to review the data
audio-chunks/chunk30.wav : At times this error identification isn't as simple
and you'll need an sme to review the data.
audio-chunks/chunk31.wav : you'll learn about the role of experts later in this
audio-chunks/chunk31.wav : You'll learn about the role of experts later in this.
audio-chunks/chunk32.wav : remember that one of the largest impacts you can have
on the success of a machine learning project is to have consistent and correct
data
audio-chunks/chunk32.wav : Remember that one of the largest impacts you can have
on the success of a machine learning project is to have consistent and correct
data.
audio-chunks/chunk33.wav : after your data is in good shape it's time to train
your model
audio-chunks/chunk33.wav : After your data is in good shape it's time to train
your model.
audio-chunks/chunk34.wav : this is where the process gets very iterative and
fluid
audio-chunks/chunk34.wav : This is where the process gets very iterative and
fluid.
audio-chunks/chunk35.wav : you'll likely go through many multiple passes of
feature engineering
audio-chunks/chunk35.wav : You'll likely go through many multiple passes of
feature engineering.
audio-chunks/chunk36.wav : training evaluating and tuning before you find a
model that meets your business goals
audio-chunks/chunk36.wav : Training evaluating and tuning before you find a
model that meets your business goals.
audio-chunks/chunk37.wav : feature engineering is the process of selecting or
creating the features your model will be trained with
```

```
audio-chunks/chunk37.wav : Feature engineering is the process of selecting or
creating the features your model will be trained with.
audio-chunks/chunk38.wav : features of The Columns of data you have in your data
set
audio-chunks/chunk38.wav : Features of the columns of data you have in your data
set.
audio-chunks/chunk39.wav : the goal of the model is to correctly estimate the
target value for the new data
audio-chunks/chunk39.wav : The goal of the model is to correctly estimate the
target value for the new data.
audio-chunks/chunk40.wav : the ml algorithm will use the features to predict the
target
audio-chunks/chunk40.wav : The ml algorithm will use the features to predict the
target.
audio-chunks/chunk41.wav : in this example the target data is the average number
of steps taken in a week
audio-chunks/chunk41.wav : In this example the target data is the average number
of steps taken in a week.
audio-chunks/chunk42.wav : selecting the correct features
audio-chunks/chunk42.wav : Selecting the correct features.
audio-chunks/chunk43.wav : get involve adding removing or calculating new
features
audio-chunks/chunk43.wav : Get involve adding removing or calculating new
features.
audio-chunks/chunk44.wav : you might want to make the data formats consistent
audio-chunks/chunk44.wav : You might want to make the data formats consistent.
audio-chunks/chunk45.wav : be consistent formats could be later used in the
model
audio-chunks/chunk45.wav : Be consistent formats could be later used in the
model.
audio-chunks/chunk46.wav : or you can make these changes for cosmetic reasons
audio-chunks/chunk46.wav : Or you can make these changes for cosmetic reasons.
audio-chunks/chunk47.wav : depending on the problem you want to solve this data
you might not even need to include the name feature in the example data
audio-chunks/chunk47.wav : Depending on the problem you want to solve this data
you might not even need to include the name feature in the example data.
audio-chunks/chunk48.wav : what about the country feature
audio-chunks/chunk48.wav : What about the country feature.
audio-chunks/chunk49.wav : If This Were a traditional database you might want to
move country to a lookup table then reference it
audio-chunks/chunk49.wav : If this were a traditional database you might want to
move country to a lookup table then reference it.
audio-chunks/chunk50.wav : most ml algorithms want the data for an instance in a
single row
audio-chunks/chunk50.wav : Most ml algorithms want the data for an instance in a
single row.
audio-chunks/chunk51.wav : ml algorithms need numerical data to process
audio-chunks/chunk51.wav : Ml algorithms need numerical data to process.
```

```
audio-chunks/chunk52.wav : you could consider turning the country text into the
countries ISO code
audio-chunks/chunk52.wav : You could consider turning the country text into the
countries iso code.
audio-chunks/chunk53.wav : however the model might interpret the numerical value
is having meaning
audio-chunks/chunk53.wav : However the model might interpret the numerical value
is having meaning.
audio-chunks/chunk54.wav : so the UK is ISO code value of 44 would be more
significant than the iso code value of the US which is 01
audio-chunks/chunk54.wav : So the uk is iso code value of 44 would be more
significant than the iso code value of the us which is 01.
audio-chunks/chunk55.wav : in this case
audio-chunks/chunk55.wav : In this case.
audio-chunks/chunk56.wav : splitting the data into multiple columns is fine
audio-chunks/chunk56.wav : Splitting the data into multiple columns is fine.
audio-chunks/chunk57.wav : this is known as categorical and cod
audio-chunks/chunk57.wav : This is known as categorical and cod.
audio-chunks/chunk58.wav : and you'll learn about this later in the course
audio-chunks/chunk58.wav : And you'll learn about this later in the course.
audio-chunks/chunk59.wav : for other types of data you could convert the text
value into a numerical value
audio-chunks/chunk59.wav : For other types of data you could convert the text
value into a numerical value.
audio-chunks/chunk60.wav : for example you could use zero or one to represent
male or female
audio-chunks/chunk60.wav : For example you could use zero or one to represent
male or female.
audio-chunks/chunk61.wav : these numeric values can be used more easily by the
model
audio-chunks/chunk61.wav : These numeric values can be used more easily by the
model.
audio-chunks/chunk62.wav : what about the remaining features like age birth
audio-chunks/chunk62.wav : What about the remaining features like age birth.
audio-chunks/chunk63.wav : which is shown as BM in the table or day of week
which is shown as Dow
audio-chunks/chunk63.wav : Which is shown as bm in the table or day of week
which is shown as dow.
audio-chunks/chunk64.wav : extracting the age birth month and day of week might
be appropriate depending on the problem you're trying to solve
audio-chunks/chunk64.wav : Extracting the age birth month and day of week might
be appropriate depending on the problem you're trying to solve.
audio-chunks/chunk65.wav : does age impact the target variable
audio-chunks/chunk65.wav : Does age impact the target variable.
audio-chunks/chunk66.wav : what about which day of the week they were born on
audio-chunks/chunk66.wav : What about which day of the week they were born on.
audio-chunks/chunk67.wav : don't worry if this sounds complicated
audio-chunks/chunk67.wav : Don't worry if this sounds complicated.
```

```
audio-chunks/chunk68.wav : you'll learn more about feature engineering later in
this course
audio-chunks/chunk68.wav : You'll learn more about feature engineering later in
this course.
audio-chunks/chunk69.wav : after your data is cleaned and you've identified the
features you want to use it's time to train a model
audio-chunks/chunk69.wav : After your data is cleaned and you've identified the
features you want to use it's time to train a model.
audio-chunks/chunk70.wav : you won't use all the data to train your model
audio-chunks/chunk70.wav : You won't use all the data to train your model.
audio-chunks/chunk71.wav : in fact you need to hold some of the data so you have
some data to test with
audio-chunks/chunk71.wav : In fact you need to hold some of the data so you have
some data to test with.
audio-chunks/chunk72.wav : typically you'll use about 80% of the data to train
with
audio-chunks/chunk72.wav : Typically you'll use about 80% of the data to train
with.
audio-chunks/chunk73.wav : and you'll save the rest of the data for testing
audio-chunks/chunk73.wav : And you'll save the rest of the data for testing.
Error:
audio-chunks/chunk75.wav : you'll train a model with training data
audio-chunks/chunk75.wav : You'll train a model with training data.
audio-chunks/chunk76.wav : in the diagram the model uses the xgboost algorithm
audio-chunks/chunk76.wav : In the diagram the model uses the xgboost algorithm.
audio-chunks/chunk77.wav : the model itself has some parameters you can set
audio-chunks/chunk77.wav : The model itself has some parameters you can set.
audio-chunks/chunk78.wav : these parameters will alter how the algorithm works
audio-chunks/chunk78.wav : These parameters will alter how the algorithm works.
audio-chunks/chunk79.wav : and they're known as hyperparameters
audio-chunks/chunk79.wav : And they're known as hyperparameters.
audio-chunks/chunk80.wav : the output of the training job will be a trained
model
audio-chunks/chunk80.wav : The output of the training job will be a trained
model.
audio-chunks/chunk81.wav : with the train model you can use some of the test
data to see how well the model performs
audio-chunks/chunk81.wav : With the train model you can use some of the test
data to see how well the model performs.
audio-chunks/chunk82.wav : you'll take an instance the model hasn't seen
audio-chunks/chunk82.wav : You'll take an instance the model hasn't seen.
audio-chunks/chunk83.wav : and use it to perform a prediction
audio-chunks/chunk83.wav : And use it to perform a prediction.
audio-chunks/chunk84.wav : because you already know the Target in your test data
you can compare the two values
audio-chunks/chunk84.wav : Because you already know the target in your test data
you can compare the two values.
audio-chunks/chunk85.wav : from these comparisons
```

```
audio-chunks/chunk85.wav : From these comparisons.
audio-chunks/chunk86.wav : you can calculate metrics which give you data on how
well the model is performing
audio-chunks/chunk86.wav : You can calculate metrics which give you data on how
well the model is performing.
audio-chunks/chunk87.wav : you'll then make changes to the models data
audio-chunks/chunk87.wav : You'll then make changes to the models data.
audio-chunks/chunk88.wav : features or hyperparameters until you find the model
that yields the best results
audio-chunks/chunk88.wav : Features or hyperparameters until you find the model
that yields the best results.
audio-chunks/chunk89.wav : when training your model
audio-chunks/chunk89.wav : When training your model.
audio-chunks/chunk90.wav : there's a real danger of overfitting or underfitting
the model
audio-chunks/chunk90.wav : There's a real danger of overfitting or underfitting
the model.
audio-chunks/chunk91.wav : your model is overfitting your training data when you
see the model performs well on the training data but doesn't perform well on the
evaluation data
audio-chunks/chunk91.wav : Your model is overfitting your training data when you
see the model performs well on the training data but doesn't perform well on the
evaluation data.
audio-chunks/chunk92.wav : this is because the model is memorizing the data it
saw and can't generalize to unseen examples
audio-chunks/chunk92.wav : This is because the model is memorizing the data it
saw and can't generalize to unseen examples.
audio-chunks/chunk93.wav : your model is underfitting the training data when the
model performs poorly on the training data
audio-chunks/chunk93.wav : Your model is underfitting the training data when the
model performs poorly on the training data.
audio-chunks/chunk94.wav : this is because the model can't capture the
relationship between the input examples which are often called X
audio-chunks/chunk94.wav : This is because the model can't capture the
relationship between the input examples which are often called x.
audio-chunks/chunk95.wav : and the target values which are often called why
audio-chunks/chunk95.wav : And the target values which are often called why.
audio-chunks/chunk96.wav : understanding model fit is important for
understanding the root cause of poor model accuracy
audio-chunks/chunk96.wav : Understanding model fit is important for
understanding the root cause of poor model accuracy.
audio-chunks/chunk97.wav : misunderstanding will guide you to take corrective
steps
audio-chunks/chunk97.wav : Misunderstanding will guide you to take corrective
steps.
audio-chunks/chunk98.wav : you can determine whether a predictive model is
underfitting or overfitting the training data
audio-chunks/chunk98.wav : You can determine whether a predictive model is
```

```
underfitting or overfitting the training data.
audio-chunks/chunk99.wav : by looking at the prediction error on the training
data and the evaluation data
audio-chunks/chunk99.wav : By looking at the prediction error on the training
data and the evaluation data.
audio-chunks/chunk100.wav : will show you steps you can take to avoid this later
in this course
audio-chunks/chunk100.wav : Will show you steps you can take to avoid this later
in this course.
audio-chunks/chunk101.wav : after you've retrained the model and you're
satisfied with the results you deploy your model to deliver the best possible
predictions
audio-chunks/chunk101.wav : After you've retrained the model and you're
satisfied with the results you deploy your model to deliver the best possible
predictions.
audio-chunks/chunk102.wav : later in this course will walk you through these
different phases and give you hands-on experience with each of them
audio-chunks/chunk102.wav : Later in this course will walk you through these
different phases and give you hands-on experience with each of them.
audio-chunks/chunk103.wav : but knowing the process is also useful when using
the managed services will also explore later
audio-chunks/chunk103.wav : But knowing the process is also useful when using
the managed services will also explore later.
audio-chunks/chunk104.wav : were certain Amazon ml services will do the bulk of
the work for you
audio-chunks/chunk104.wav : Were certain amazon ml services will do the bulk of
the work for you.
audio-chunks/chunk105.wav : here are some of the key takeaways for this section
audio-chunks/chunk105.wav : Here are some of the key takeaways for this section.
Error:
audio-chunks/chunk107.wav : we looked at how the machine learning pipeline
process can guide you through the process of training and evaluating a model
audio-chunks/chunk107.wav : We looked at how the machine learning pipeline
process can guide you through the process of training and evaluating a model.
audio-chunks/chunk108.wav : the iterative process can be broken into three broad
step
audio-chunks/chunk108.wav : The iterative process can be broken into three broad
step.
audio-chunks/chunk109.wav : data processing
audio-chunks/chunk109.wav : Data processing.
audio-chunks/chunk110.wav : model training and model evaluation
audio-chunks/chunk110.wav : Model training and model evaluation.
audio-chunks/chunk111.wav : that's it for this video we'll see you in the next
one
audio-chunks/chunk111.wav : That's it for this video we'll see you in the next
one.
```

[12]:

```
print(len(transcribed_texts))
```

```
46
```

0.5 3. Normalizing the text

(Go to top)

[13]:

```
import speech_recognition as sr
nltk.download('stopwords')
import nltk
nltk.download('punkt')
nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re


# Normalize all transcribed texts at once
normalized_texts = []
for text in transcribed_texts:
    # Normalize the text by converting to lowercase
    normalized_text = text.lower()

    # Remove special characters from the text
    normalized_text = re.sub('[^A-Za-z0-9]+', ' ', normalized_text)

    # Remove stop words from the text
    if not nltk.corpus.stopwords.words('english'):
        nltk.download('stopwords')
    stop_words = set(stopwords.words('english'))
    words = normalized_text.split()
    filtered_text = [word for word in words if not word in stop_words]
    normalized_text = ' '.join(filtered_text)

    # Lemmatize the text
    if not nltk.corpus.wordnet:
        nltk.download('wordnet')
    lemmatizer = WordNetLemmatizer()
    words = nltk.word_tokenize(normalized_text)
    normalized_text = ' '.join([lemmatizer.lemmatize(word) for word in words])

    normalized_texts.append(normalized_text)

# Print the normalized text for each file
for normalized_text in normalized_texts:
    print(f'Transcript of {audio_file}: {normalized_text}')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/ec2-user/nltk_data…
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /home/ec2-user/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /home/ec2-user/nltk_data…
```

```
Transcript of Mod02_Sect03.wav: hi welcome back section 3 going cover evaluate
data section look different data format type also look visualize analyze data
feature engineering start running statistic data better understand working need
ensure right format analysis amazon sagemaker algorithm support training data
csv format many tool use explore visualize analyze data also read csv format
generally speaking need least domain knowledge problem trying solve machine
learning example developing model predict set symptom indicates disease need
know relationship symptom disease data typically need numeric form machine
learning algorithm use data make prediction look way convert text data next
section explore data try gain insight overall data set one popular open source
python library panda take data various format reformat load tabular
representation data presenting row column format panda reformat load include csv
excel pickle javascript object notation json panda also data analysis
manipulation feature use throughout module loading data simple example pull csv
file specified url load data panda stored panda dataframe panda documentation
data frame described general 2d labeled size mutable tabular structure
potentially heterogeneously typed column helpful way think data frame think
spreadsheet sql table like table spread data frame row also known instance
column also known attribute shape property data frame describes number row
column column data frame series series one dimensional labeled array series
store data type learn data structure panda see panda documentation along data
load data frame row label column label row label known index column label known
column loaded data csv file hetero column created first line file change
behavior however column name source file pas parameter performing data analysis
important make sure using correct data many case panda correctly infer correct
data type load data move domain knowledge access domain expert often identify
data type issue use either type info function obtain information column type
shown example correct data type need figure case often numeric column could
missing data could single text value example car data set number door 23 five
analyzed data convert column correct data type using panda part one section see
part 2 review describe data
Transcript of Mod02_Sect03.wav: hi welcome module 2 aws academy machine learning
module going introduce machine learning first look business problem solved
machine learning well talk terminology process tool challenge face completing
module able recognize machine learning deep learning part artificial
intelligence describe artificial intelligence machine learning terminology
identify machine learning used solve business problem describe machine learning
process list tool available data scientist identify use machine learning instead
traditional software development method ready get started section 1 see next
video
Transcript of Mod02_Sect03.wav: welcome back section explore image analysis
```

detail part 2 take closer look video analysis introduce main amazon service using amazon recognition amazon recognition computer vision service based deep learning use add image video analysis application many us amazon recognition including creating searchable image video library amazon recognition make image stored video searchable discover object scene appear use amazon recognition build face based user verification system application confirm user identity comparing live image reference image amazon recognition interprets emotional expression happy sad surprise also interpret demographic information facial image gender amazon recognition also detect inappropriate content image stored video finally amazon recognition recognize extract text content image go quick note security need check application build using amazon recognition fall regulatory restriction defined field country security compliance amazon recognition shared responsibility aws customer information topic see aws compliance page amazon recognition aws managed service managed service amazon host machine learning model maintains api scale meet demand benefit set model constantly learn improve also focus building application use api optionally training service understand unique business need various resource use access interact amazon recognition apis sdks command aws command line interface also known aws cli language supported sdk include javascript python php dotnet jav j amc plus plus finally amazon recognition integrates aws service example need storage use amazon simple storage service s3 authentication authorization use aws identity access management diagram illustrates image search feature user take picture get information real estate property viewing user take picture mobile device user initiate search cause application upload amazon s3 s3 configured call service right occurs case bucket pass s3 path new object aws lambda lambda function called us amazon recognition sdk call service amazon recognition analyzes image detect aspect property creates label pass information back lambda object formatted javascript object notation json lambda store label confidence score amazon elastic service also known amazon e application user identify aspect property using object detected image example architecture system check uploaded image inappropriate content like previous example processing begin user uploads content user uploads image amazon s3 second s3 bucket configured call lambda function object written bucket lambda call amazon recognition via sdk amazon recognition analyzes image inappropriate content sends response back lambda content appropriate content approved content appropriate content sent manual inspection finally content approved notification sent user final use case system analyzes video feed sentiment analysis store camera capture video sent back office cloud based application typically application like us amazon kinesis stream video second application us sdk send video amazon recognition analysis visual sentiment extracted along attribute age discovered attribute sent amazon kinesis lambda function extract data stream data written s3 next data loaded amazon redshift regular basis finally tool like amazon quicksight used generate report data amazon recognition designed integrate application api sdks api operation provided detecting label face recognizing celebrity detecting unsafe image perform prediction provide service image object amazon s3 upload bite stream image image jpeg png format amazon recognition process image performs prediction return json object result amazon recognition performs prediction often return multiple label label confidence

level confidence level indicates likely label found image like example show label also hierarchy find instance object need understand detected object image instance result amazon recognition include bounding box contains starting coordinate top left box dimension height like example use information determine location detected object image important note finding contain confidence score use confidence score application tune response prediction higher score likely object correctly labeled part 1 section see part 2 explore facial detection
Transcript of Mod02_Sect03.wav: welcome back time review module wrap summary module learn describe nlp use case better solved using managed amazon ml service describe managed ml service available nlp good job thanks watching see next module
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring data collection reviewing secure data important consider security data go data set used course public real data customer transaction health record need kept secure use aws identity access management also known iam service control access resource make sure securing data within aws correctly avoid data breach diagram show simple iam policy allows read access specific s3 bucket listed role addition controlling access data need make sure data secure good practice might also legally required certain data type financial data healthcare record aws provides encryption feature storage service typically data rest transit often meet encryption requirement enabling encryption object service want protect data transit must use secure transport like secure socket slayer transport layer security ssl tl another aspect consider compliance audit dealing data regulated industry often need audit access data aws cloud trail service enables governance compliance operational auditing risk auditing aws account cloudtrail log continuously monitor retain account activity related action across entire aws infrastructure cloudtrail provides event history aws account activity including action taken aws management console aws sdk command line tool aws service event history simplify security analysis resource change tracking troubleshooting also use cloudtrail detect unusual activity aws account feature help simplify operational analysis troubleshoot key takeaway section looked first step solving machine learning problem obtaining data required train machine learning model also reviewed etl used obtain data multiple source service like aws glue make easy obtain data multiple data store finally make sure understand security requirement based business need regulatory requirement also make sure data secure authorized user able access data encrypted possible section 2 see next video
Transcript of Mod02_Sect03.wav: welcome back aws academy machine learning module 5 great topic today computer vision module start overview computer vision space learn use case terminology next explore detail analyzing image video managed service amazon web service aws finally look use customized data set performing object detection end module able describe use case computer vision describe amazon management learning service available image video analysis list step required prepare custom data set object detection describe amazon sagemaker ground truth used prepare custom data set finally use amazon recognition perform facial detection thanks watching see next video
Transcript of Mod02_Sect03.wav: section look preparing custom data set computer vision detect custom object one challenge using pre built model find image

trained find though amazon recognition trained ten million image detect object trained example consider aid heart playing card run car amazon recognition result show various attribute however none label playing card eight heart want amazon recognition detect image problem domain must train model image section learn train amazon recognition image problem domain focus using amazon recognition encounter similar process use pre trained model training computer vision algorithm recognize image requires large input data set practical organization many machine learning problem today solved training existing model use managed service like amazon recognition custom label like machine learning process need train amazon recognition recognizes scene object specific domain need training data set test data set contain labeled image image need label use amazon recognition custom label simplify labeling task example provides ui labeling image includes feature use draw bounding box around image also help find object scene unique business need use classify image detect object within image say want identify specific machine part image turbochargers torque converter collect picture kind machine use train model amazon recognition custom label also includes automated machine learning capability handle machine learning process provide training image service automatically load inspect data select correct machine learning algorithm train model provide model performance metric finish training model evaluate custom model performance test set image test set side side comparison model prediction versus label assigned also detailed performance metric review start using model immediately image analysis iterate retrain new version image refine model start using model track prediction correct mistake use feedback data retrain new model version improve performance label image diagram show typical process training computer vision model includes amazon recognition custom label feature step detail process developing custom model analyze image requires time expertise resource often take month complete also require thousand ten thousand hand labeled image model enough data make accurate decision take month generate gather data require large team laborer prepare use machine learning amazon recognition custom label build existing capability amazon recognition already trained ten million image across many category instead thousand image upload small set training image specific use case typically use hundred image use aws management console upload training image image already labeled amazon recognition custom label begin training model label image directly labeling interface use amazon sagemaker ground truth label tell shortly amazon recognition custom label work best use different model different domain example need detect machine part plant health two different model image select training similar image used inference use image use various lighting condition background resolution ideally training image mirror image want perform detection use source like use production work best documentation includes additional guideline image type weather jpegs pngs property like image size resolution part 1 section see part 2 review create training data set Transcript of Mod02_Sect03.wav: hi welcome back section going look type business problem machine learning help solve machine learning used across digital life email spam filter result machine learning program trained example spam regular email message based book reading product bought machine learning program predict book product likely interested machine learning program trained data reader habit purchase detecting credit card fra machine learning program trained example transaction turned fraud along normal transaction probably think many example social medium application using facial detection group photo detecting brain tumor brain scan finding anomaly x ray three main type machine learning supervised learning model us known input output generalize future output unsupervised learning model know input output find pattern data without help reinforcement learning model interacts environment learns take action maximize reward important know different type ml type guide toward selecting algorithm make sense solving business problem let look type supervised learning popular type ml widely applicable called supervised learning need supervisor teacher show right answer speak like student supervised algorithm need learn example essentially need teacher us training data help determine pattern relationship input output want build application detect credit card fraud need training data includes example fraud example normal transaction within supervised learning different type problem classification regression two subtypes classification problem first binary classification think back example identifying fraudulent transaction target variable example limited two option fraudulent fraudulent binary classification problem also multi class classification problem ml problem classify observation one three category say ml model predicts customer calling store reduce number transfer needed customer get correct customer support department case different customer support department represent variety potential target variable could many different department much two also regression problem regression problem longer mapping input defined number category instead mapping input continuous value like integer one example ml regression problem predicting price company stock computer vision good example supervised learning tumor x ray computer vision often built deep learning model automates extraction analysis classification understanding useful information single image sequence image computer vision enables machine identify people place thing image accuracy human level greater speed efficiency image data take many form search single image video sequence view multiple camera three dimensional data learn computer vision later discus unsupervised machine learning sometimes data supervisor room unsupervised learning label provided like supervised learning know variable pattern instance machine uncover create label model use date presented detect emerging property entire data set construct pattern property clustering common subcategory unsupervised learning kind algorithm group data different cluster based similar feature better understand attribute specific cluster example analyzing customer purchasing habit unsupervised algorithm identify group customer associated size tear company advantage unsupervised algorithm enable see pattern data aware natural language processing also known nlp another area machine learning experiencing growth ever used alexa voice value nlp try answer question nlp speech nlp show many application example nlp used chat call center bot automated system help get bank balance order food restaurant use nlp translation tool convert text language example might use application translate menu real time nlp also used voice text translation convert spoken word text finally nlp used sentiment analysis use analyze sentiment comment review product music movie sentiment could used give movie audience rating learn nlp later another kind machine learning gaining popularity recently reinforcement learning unlike machine learning reinforcement learning continuously improves model mining feedback

previous iteration reinforcement learning agent continuously learns trial error interacts environment reinforcement learning broadly useful reward desired outcome known path achieving annette path requires lot trial error discover take example aws deepracer aws deep racer simulator agent virtual car environment virtual race track action throttle steering input car goal completing racetrack quickly possible without deviating track car need learn desired driving behavior reach goal completing track car learn aws deepracer team use reward incentivize model learn desired driving behavior reinforcement learning thing driving learning called agent case aws deep racer car environment place agent learns example would marked race track agent something environment provokes response crossing boundary cross called action response called reward penalty depending whether agent something reinforced discouraged model agent move within action start receiving reward fewer penalty meet desired business outcome self driving vehicle bring together many machine deep learning algorithm model solve problem driving point point b two main task continuous detection environment forecasting change involve detecting object localizing predicting movement detected object output finding act input system make decision vehicle various control use case self driving vehicle require real time response environment example previously hidden pedestrian walk behind obstacle vehicle break need applied immediately latency room error action every problem solved machine learning sometimes regular programming work well need interested exploring potential machine learning solution look existence large data set large number variable machine learning often best choice uncertain business logic procedure needed obtain answer accomplish task machine learning system complex supporting infrastructure management support technical expertise need place help ensure project success key takeaway section explored machine learning application already part everyday life machine learning problem grouped three category supervised learning training data already know answer unsupervised learning data looking insight within data reinforcement learning model learns based experience feedback business problem supervised learning problem section see next video
Transcript of Mod02_Sect03.wav: welcome module 7 course wrap congratulation completing aws academy machine learning course take minute review learned go going start review learned course learned describe machine learning implement machine learning pipeline use amazon machine learning service forecasting computer vision natural language processing well done although course designed prepare become certified aws certified machine learning specialty review continue work towards certification aws certification help build credibility confidence validating cloud expertise industry recognized credential also help organization identify skilled professional lead cloud initiative using aws must earn passing score taking proctored exam earn aws certification receiving passing score receive certification credential aws certification publish list service feature covered certification exam however exam guide exam list current topic area objective covered exam exam guide found prepare aws certification exam webpage required update certification recertify every 3 year view aws certification recertification page detail information slide current june 2020 however exam frequently updated also detail regarding exam available topic tested exam subject change aws certified machine learning specialty mean select justify appropriate machine learning approach given business problem also

identify appropriate aws service implement machine learning solution finally design implement scalable cost optimized reliable secure machine learning solution sitting aws certified machine learning specialty exam recommend following knowledge experience one two year experience developing architecting running ml deep learning workload aws cloud experience include performing basic hyperparameter optimization working machine learning deep learning framework also able express intuition behind basic ml algorithm finally able follow best practice model training addition best practice deployment operation thanks watching congratulation completing aws academy machine learning course Transcript of Mod02_Sect03.wav: welcome back section look tool using throughout rest course start list exhaustive list tool available today going cover high level good place get started jupyter notebook jupyter notebook open source web application use create share document contain live code equation visualization narrative text us include data cleaning transformation numerical simulation statistical modeling data visualization machine learning much jupiter lab web based interactive development environment jupyter notebook code data jupiter lab flexible use configure arrange user interface support wide range workflow data science scientific computing machine learning jupiter lab extensible modular write plugins add new component integrate existing one later course use amazon sagemaker host jupiter notebook jupiter lab panda open source python library used data handling analysis panda represents data table similar spreadsheet table known panda dataframe matplotlib python library creating scientific static animated interactive visualization python use generate plot data later course seaborne another data visualization library python built matplotlib provides high level interface drawing attractive informative statistical graph numpy one fundamental scientific computing package python contains function n dimensional ray object also useful math function linear algebra fourier transform random number capability psychic learn open source machine learning library support supervised unsupervised learning also provides various tool model fitting data preprocessing model selection evaluation many utility sci fi matplotlib good tool exploring machine learning although use borrow function course might want consider exploring complete moving individual library package also tool contain production ready framework already mentioned sidekick learn good library machine learning framework supported aws tensorflow kera also include library use machine learning framework listed supported aws used amazon sagemaker aws also provides compute instance tuned machine learning cloud edge compute instance optimized learning inference another aws resource use certain amazon machine image amiss offer pre packaged amis contain many popular framework finally amazon sagemaker aws service many capability sagemaker deploy machine learning instance running jupiter notebook jupiter lab manages deployment compute resource need connect jupiter environment sagemaker also provides tool labeling data training model hosting train model aws marketplace also provides selection ready use model package algorithm third party machine learning developer aws also provides set management machine learning service integrate application even substantial machine learning experience computer vision amazon recognition provides object facial recognition image video also amazon textract extract text image speech service include amazon polly speak text another speech service amazon transcribe convert spoken audio language amazon comprehend us nlp find insight relationship text also amazon translate translate text different language want work chat amazon alexa help build interactive conversational application use voice text forecasting amazon forecast us machine learning combine time series data additional variable build forecast finally like work recommendation amazon personalized help create individual personalized recommendation customer managed service already trained many aspect problem domain need provide specific data get started going look many managed service second half course learn thing key takeaway section include point python popular language performing machine learning task jupiter notebook provide web based hosted development environment machine learning use jupiter notebook frequently machine learning large number open source tool pand use often machine learning practitioner finally depending upon requirement might start low level framework create solution might also use tool amazon sagemaker help heavy lifting simply use adapt one managed amazon ml service specific problem domain video see next one

Transcript of Mod02_Sect03.wav: hi welcome back continue exploring evaluate model diagram show confusion matrix two different model performed data tell one better better good question ask mean better better mean making sure find cat even mean get many false positive better mean making sure model accurate difficult see looking two chart trying several model using multiple fold hundred data point compare need calculate metric first metric sensitivity sometimes referred recall hit rate true positive rate sensitivity percentage positive identification cat example represents percentage cat correctly identified calculate sensitivity take number true positive number positive identification cat divide total number actual cat example 60 cat cat correctly identified cat specificity sometimes referred selectivity true negative rate specificity percentage negative correctly identified cat example number image cat correctly identified cat calculate specificity take number true negative divide total number actual negative example number cat correctly identified divided total number actual cat mean example 64 cat identified cat metric model knowing business goal make easier decide model use model would choose wanted make sure identify many cat possible model b would good answer concerned many false positive concerned incorrectly identified cat model would choose wanted make sure identified animal cat model might work scenario would depend many false negative tolerate classification patient heart disease model would best get interesting fun website might get bad reputation identify cat correctly trying diagnose patience focus probably different important understand trade offs making decide model use also metric help make decision part 2 section see part 3 start looking threshold

Transcript of Mod02_Sect03.wav: get started reviewing natural language processing mean natural language processing also known nlp explain nlp consider example nlp amazon alexa alexa work device amazon echo record word recording speech sent amazon server analyzed efficiently amazon break phrase individual sound connects database containing pronunciation various word define word closely correspond combination individual sound amazon identifies important word make sense task carry corresponding function instance alexa notice word like outside temperature open weather alexa skill amazon server send information back device alexa speaks mlp broad term general set business computational problem solve machine learning ml nlp system predate machine learning example speech text older pretty smartphone cell phone used nlp screen reader many nlp system use form machine learning mlp considers hierarchical structure language word lowest layer hierarchy group word make phrase next level phrase make sentence ultimately sentence convey idea nlp system face several significant challenge look challenge next language precise word different meaning based word surround known context often word phrase multiple meaning example consider term weather could weather colloquial meaning english sick could say wonderful weather outside mean weather condition outside good phrase oh really could convey surprise disagreement many thing depends context inflection main challenge nlp one challenge discovering structure text one first task nlp application break text meaningful unit word phrase sentence another challenge labeling data system convert text data must apply label representing various part speech every language require different labeling scheme match language grammar nlp also face challenge representing context word meaning depend heavily context nlp system need way represent large challenge many contact difficult convert contact form computer understand finally although grammar defines structure language application grammar indescribably large scope handling variation language used human major challenge mlp system machine learning large impact apply nlp range problem common application include search application google bing human machine interaction like alexa sentiment analysis marketing political campaign social research based medium analysis chatbots mimic human speech application apply machine learning development pipeline seen throughout course developing mlp solution first task formulate problem collect label data nlp collecting data consists breaking text meaningful subset labeling set feature engineering major part mlp application process get complicated dealing highly irregular unstructured text example say building application classify document need able distinguish word common term different meaning labeling data nlp domain sometimes also called tagging labeling process assign individual text string different part speech specialized tool use nlp labeling first task mlp application convert text data analyzed convert removing word needed analysis input text example word removed leave phrase sample text removing stop word normalize text converting similar word common form example word run ran running different form word run normalize instance word block text using stemming limitation process limitation group different form word single term limitation version word run would group instance form single term stemming hand remove character stemming algorithm considers unnecessary standing might work run example form ran might recognized form word run normalised standardize removing word dictionary using analysis example could remove acronym slang special character natural language toolkit also known nltk python library provides function removing stop word normalizing text another first step creating mlp system convert text data collection data frame mlp library provide function assist process example show using word tokenize function nltk library cleaned text loaded data frame apply one nlp model create feature couple common model first model known bag word simple model capturing frequency word document model creates key word value key number time word occurs document second model term frequency inverse document frequency also known tf idf term frequency count many time word appears document inverse document frequency number time word occurs

group document two value used together calculate weight word word frequently appear many document lower weight many established model nlp field example show bag word model bag word vector model vector model convert sentence phrase vector mathematical object record directionality magnitude example simple sentence converted vector frequency word recorded word value 2 appears twice sentence bag word often used classify document different category also used derive attribute feed nlp application sentiment analysis three broad category text analysis classification text similar classification system seen course text provides input process extract feature send feature machine learning algorithm interacts classifier model infers classification many application text matching example autocorrect spelling grammar checking based text matching algorithm edit also known levinstein frequently used drive relationship different word phrase text using process called core reference resolution several nlp system provide python library deriving relationship one biggest challenge nlp describe context text consider example user searching term tablet word tablet least two distinct meaning search engine need know meaning user mind search engine rely commonly used context term qualified example adding another term like medicine computing search process extracting entity known named entity recognition armor following function identify noun phrase using dependency chart part speech tagging classify phrase using classification algorithm word vec finally disambiguate entity using knowledge graph example using ner extract entity titanic north atlantic text named entity extracted use knowledge graph extract meaning knowledge graph combine subject matter expertise machine learning drive meaning amazon recommendation engine example knowledge graph main point remember section nlp predates machine learning use ml workflow seen module nlp main use case nlp search query analysis human machine interaction marketing social research nlp complicated human language lack precision thanks watching see next video
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring image analysis closer look facial detection facial detection us model tuned perform prediction specifically detecting face facial feature facial detection many feature standard object detection bounding box coordinate box surrounding face detected include value representing confidence bounding box contains face list attribute found face beard appears male female also confidence score attribute also detect physical emotion like whether person smiling frowning important understand classification based visual clue might represent actual emotion person facial landmark component face eye mouth typical landmark also include x coordinate quality describes brightness sharpness face pose describes rotation face inside image confidence feature provided detected feature remember feature prediction based visual observation amazon recognition compare two image determine contain person comparison require source target image result include face found include information matching non matching face confidence score indicate likely prediction amazon recognition also search known face use feature need train model providing collection image use train model detect people image provide find known face first create collection add face collection amazon recognition perform facial recognition image provide return typical information like bounding box coordinate confidence score associate face image specify image id external image id request parameter could file name image another id create create collection use search face image operation search face collection

returned data contains array face matched information includes bounding box confidence score external image id value use id value link back source image learned facial detection feature amazon recognition summary guideline discussed far amazon recognition detects human face capture bounding box show face found video also detect attribute position eye nose mouth detect emot quality detection landmark might appear item associated confidence score higher score mean model greater confidence detection gender inferred image inferred identity similarly emotion also determined image might reflect subject actual emotional state apply facial recognition responsibly facial recognition never used way violates individual right including right privacy also never used make autonomous decision scenario require human analyze example suppose bank us tool like amazon recognition financial application verify customer identity bank always clearly disclosed use technology ask customer approve term condition information topic see aws web page fact facial recognition artificial intelligence explain use amazon recognition process video perform video processing stored video video stream stored video uploaded stored s3 bucket type detection start operation search people face label celebrity text inappropriate content amazon recognition publishes completion status topic amazon simple notification service also known amazon sn route message subscriber durability best practice route message message queue amazon simple queue service amazon sqs application monitor sqsq completion start operation corresponding get operation retrieving result call get detection result return array label contain information label found video label information includes label image detection also includes timestamp label detected millisecond start video addition stored video also use amazon recognition video detect recognize face streaming video typical use case detecting known face video stream amazon recognition video us amazon kinesis video stream receive process video stream analysis result output amazon recognition video kinesis datastream read client application amazon recognition video provides stream processor called create stream processor use start manage analysis streaming video use amazon recognition video streaming video application must implement resource need kinesis video stream send streaming video amazon recognition video need amazon recognition video stream processor manage streaming video analysis finally need kinesis datastream consumer read analysis result amazon recognition video sends datastream want find face video need create collection process creating collection still image amazon recognition video place json frame record analyzed frame kinesis output stream amazon recognition video analyze every frame passed kinesis video stream frame record sent kinesis datastream contains information video stream fragment frame frame fragment face recognized frame also includes status information stream processor wrap quick summary amazon recognition computer vision service based deep learning easily add image video analysis application amazon recognition detect face sentiment text unsafe content library search image video amazon recognition integrated aws service thanks watching see next video Transcript of Mod02_Sect03.wav: hi welcome section 4 section going look feature engineering feature engineering one impactful thing improve machine learning model look two thing help make model successful first feature selection second feature extraction process creating feature feature selection select relevant feature discard rest apply feature selection prevent redundancy irrelevance

existing feature also use limit number feature help prevent overfitting feature extraction build valuable information raw data reformatting combining transforming primary feature new one process continues yield new data set consumed model achieve goal diagram show feature extraction cover range activity dealing missing data converting text data numerical data although list exhaustive give idea data handling needed get data useful state many task different job working data want make sure data correct format consistently represented correctly spelled among example might combine data extract data multiple column could also remove column altogether specific machine learn need convert text column numerical value also need decide handle outlier potentially rescale data next look common task section machine learning algorithm work best numerical data need make sure column data set contain numeric data converting encoding might need make several pass datasheet encode example might variability text value rose contain medium med value categorical data ordered want encode text numerical value capture ordinal relationship say data showing maintenance cost might encode low one medium hi three high four made sure categorical data uniform use tool like skykit learn panda encode data categorical data order need break data multiple column help make sure introduce ordinal relationship data example suppose assigned value 1 first color red assigned next value say blue model could interpret blue important red blue higher numeric value encoding non ordinal data multiple column feature better way think new feature like checkbox consider example three feature generated volume one indicates instance feature like color section see next video
Transcript of Mod02_Sect03.wav: time review module main takeaway module looked defining machine learning fit broader ai landscape also looked type problem machine learning help u solve machine learning applies learning algorithm develop model large data set looked machine learning pipeline different stage developing machine learning application finally introduce tool service use discussing challenge machine learning summary module learned recognize machine learning deep learning part artificial intelligence describe artificial intelligence machine learning terminology identify machine learning used solve business problem describe machine learning process list tool available data scientist identify use machine learning instead traditional software development method thanks watching see next video
Transcript of Mod02_Sect03.wav: hi welcome back module 3 section 8 section going take look tune model hyperparameters improve model performance recall earlier module hyperparameters thought knob tune machine learning algorithm improve performance looking explicitly tuning model time look specifically different type hyperparameters perform hyperparameter optimization couple different category hyperparameters first kind model hyperparameters helped define model example consider neural network computer vision problem case additional attribute architecture need defined like filter size cooling stride padding second kind optimizer hyperparameters relate model learns pattern based data used neural network model type hyperparameters include optimizers like gradient desu stochastic gradient descent also include optimizers use momentum like atom initialize parameter weight method like xavier initialization initialization third kind data hyperparameters relate attribute data include attribute define different data augmentation technique like cropping resizing image related

problem often used enough data enough variation data tuning hyperparameters labor intensive traditionally done manually someone domain experience related hyperparameter use case person would manually select hyperparameters based intuition experience would train model score validation data process would repeated achieved satisfactory result manual process always thorough efficient way tuning hyperparameters sagemaker perform automated hyperparameter tuning amazon sagemaker automatic model tuning find best version model running multiple training job data set using algorithm hyperparameter range specify chooses hyperparameter value result model performs best measured metric choose us gaussian process regression predict hyperparameter value might effective improving fit also us bayesian optimization balance exploring hyperparameter space exploiting specific hyperparameter value appropriate importantly automatic model tuning used built algorithm sagemaker pre built deep learning framework bring algorithm container suppose want solve binary classification problem fraud data set goal maximize area auc curve metric algorithm training linear learner algorithm model know value learning rate beta one beta 2 epoxy use train best model find best value hyperparameters specify range value sagemaker hyperparameter tuning search find combination value result training job performs best measure objective metric chose example sagemaker hyperparameter tuning launch training job use hyperparameter value range specified return training job highest auc hyperparameter tuning might necessarily improve model advanced tool building machine solution considered part scientific method process build complex machine learning system like deep learning neural network exploring possible combination impractical improve optimization use following guideline create hyperparameters instead using hyperparameters limit number hyperparameters one think would give good result range value hyperparameters choose search significantly affect success hyperparameter optimization although might want specify large range cover every possible value hyperparameter get better result limiting search small range value get best metric value within part range consider limiting range part hyperparameter tuning sagemaker attempt figure hyperparameters log scaled linear scaled initially assumes hyperparameters linear scaled log scaled might take time sagemaker discover know hyperparameter long scaled convert improve hyperparameter optimization running hyperparameter tuning job concurrently get work done quickly tuning job improves successive round experiment typically running one training job time achieves best result least amount compute time say distributed training job run multiple instance case hyperparameter tuning us last reported objective metric instance training job value objective metric training job design distributed training job report objective metric want gone end end process training tuning machine learning model worth talking amazon sagemaker autopilot service help find good model little effort input part autopilot create job supply test training target autopilot analyze data select appropriate feature train tune model document metric find best model based provided data result include winning model metric jupyter notebook use investigate result although using autopilot remove need pre process data save time feature selection model tuning key takeaway section module include point model tuning important finding best solution business problem hyperparameters tuned model optimizer data sagemaker perform automatic hyperparameter tuning finally overall model development accelerated using

autopilot video see next one
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring describe data data readable format perform descriptive statistic data better understand descriptive statistic help gain valuable insight data effectively pre process data prepare ml model look discus important descriptive statistic organized different category overall statistic include number row number column data set information relates dimension data important example indicate many feature lead high dimensionality poor model performance attribute statistic another type descriptive statist specifically numeric attribute used get better sense shape attribute includes property like mean standard deviation var minimum maximum value need look relationship one variable consider multivariate statistic mostly relate correlation relationship attribute case multiple variable feature might want look correlation important identify correlation attribute high correlation two attribute sometimes lead poor model performance feature closely correlated used model predict response variable could problem example model loss might converge minimum state aware highly correlated feature data set mean median two different measure describing extent data clustered around value position mean useful method understanding data data symmetrical however data skewed contains outlier median tends provide better metric understanding data relates central tendency instance outlier large value mean skewed one way serve accurate representation value truly centered median affected outlier way talk outlier soon statistic available viewed numerical data using method described also method calculate mean median others also view statistic single multiple column even group data specific value categorical attribute look frequency attribute value data set information give idea inside categorical variable diagram show car data set made several categorical value lug bo safety class safety either low medium high describe function see three unique value low frequent looking class column appears top value four unacc stand unaccounted account 1 210 1 728 value 70 might suggest imbalance target variable also categorical type look class distribution see whether class imbalance data set imbalance data mark disproportionate ratio class instance data set made credit card transaction tenth percent labeled fraud case algorithm might learn well enough predict example credit card fraud visualization could help gain insight data might aware otherwise histogram often good visualization technique seeing overall behavior particular feature histogram answer question like feature data normally distributed many peak data skewness particular feature using histogram data visualization value bend taller peak histogram indicate common value numerical feature use density plot box plot addition histogram get idea inside particular feature like histogram visualization help answer question like range data peak data special feature answering question box plot method graphically depicting group numerical data quartile two numerical variable feature data set might want look relationship scatter plot good way identify special relationship among variable case left diagram sulfate alcohol two numerical variable suppose want show relationship variable use scatter plot help visualize plot scattered around correlation among might high data scattered however might find relatively positive relationship two variable scatter plot matrix help look relationship multiple different feature panda easily create scatter plot matrix based column want look example three column give pairwise scatter plot two column scatter

plot might want identify special region particular subset data fit example relationship alcohol sulfate quality plot value good poor quality wine like example plotting give idea useful particular variable using classification problem part two section see part 3 review correlation takeaway section Transcript of Mod02_Sect03.wav: hi welcome back continue exploring video analysis reviewing evaluate improve model general improve quality model larger quantity better quality data use training image clearly show object seen include many thing interested bounding box around object use training image show object fully visible hidden object make sure training test data set match type image eventually run france object training example like logo provide bounding box around logo test image image represent scenario want localize object reducing false positive often result better position reduce false positive check increasing confidence threshold enables keep correct prediction eliminating false positive increase confidence threshold eventually result diminishing gain trade precision recall given model check see need add additional class training example detecting cat often dog flagged cat add dog label training data set along image dog got false positive effectively helping model learn predict dog cat new training image might find model confused two custom label cat dog test image label cat predicted labeled dog vice versa case first check mislabeled image training test set also adding training image reflect confusion help retrain model learn better discriminate cat dog reducing false negative often result better recall reduce false negative first lower confidence threshold improve recall also use better example model variety object image appear finally split label two class easier learn example instead good cooky bad cooky might want good cooky burnt cooky broken help model learn unique concept better satisfied performance model make available use starting console using code model running perform inference aws cli sdk call api specify amazon resource name amazon recognition custom label model want use amazon resource name also known arn also specify image want model make prediction provide input image image byte array base 64 encoded image bite s3 object custom label returned array custom label object custom label represents single object seen concept found image custom label includes label object seen concept found image also includes bounding box object found image bounding box coordinate show object located source image coordinate value ratio overall image size finally custom label includes confidence score represents confident amazon recognition custom label accuracy label bounding box training model calculates threshold value determines prediction label true default detect custom label operation return label confidence value le model calculated threshold value filter return label specify value men confidence greater model calculated threshold get model calculated threshold model training result amazon recognition custom label console get label regardless confidence specify min confidence value 0 find confidence value returned detect custom label operation low consider retraining model restrict number custom label returned detect custom label operation specifying max result input parameter return result sorted highest confidence lowest confidence key takeaway section module model must trained specific domain want analyze looking turbochargers need many picture turbochargers train model set custom labeling specific business case looked custom labeling process tool use want object detected need label image create bounding box object use amazon sagemaker ground

truth build training data set model also use machine learning label image thanks watching see next video
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring evaluate model classification model going return probability target value input belonging target class 0 1 convert value class need determine threshold use might think 50 could change lower higher improve result seen sensitivity specificity trade correctly incorrectly identifying class changing threshold impact outcome going take look visualize receiver operating characteristic graph also known roc graph summarizes confusion matrix threshold produced build one calculate plot sensitivity true positive rate false positive rate graph threshold value calculate false positive rate subtracting specificity one plot point draw line dotted black line 00 11 mean sensitivity true positive rate equal false positive rate point 11 mean correctly identified cat also incorrectly identified cat bad point line mean proportion correctly classified sample proportion incorrectly classified sample point 00 true positive 0 five model high sensitivity low false positive rate usually goal considered better line threshold recording closer towards top left corner data two model could plot roc curve model compare however tedious another graph use look next another evaluation metric use area curve receiver operator cur also known auc roc auc part area plotted line auc higher mean model better predicting cat cat cat cat use auc quickly compare model four number confusion matrix calculate model accuracy also known score adding correct prediction dividing number total number prediction accuracy widely used metric classification problem limitation metric effective lot true negative case data set think cat cat example accuracy based true negative say model good predicting cat case might feel confident model ability predict cat roll production lead example important make sure metric choose model evaluation aligns business goal think credit card fraud example case using accuracy main metric probably good idea lot true negative high true negative number might hide fact model ability identify case fraud identify true positive ideal credit card company probably unacceptable le almost perfect performance identifying fraud case would drive customer away would opposite want achieve business standpoint two metric often used situation first one precision essentially remove negative prediction precision proportion positive prediction actually correct calculate taking true positive dividing true positive plus false positive cost false positive high particular business situation precision might good metric think classification model identifies email message spam case want model label email message spam thus prevent user seeing message actually legitimate consider example model need predict whether patient terminal illness case using precision evaluation metric account false negative model model successful crucial fall asleep predict absence illness patient actually illness sensitivity would better metric use situation always need one f1 score combine precision sensitivity together give one number quantifies overall performance particular ml algorithm consider using f1 score class imbalance want preserve equality precision sensitivity dealing regression problem case common metric use evaluate model including mean squared error mean squared error frequently used general purpose saw classification metric determine prediction model compare difference prediction actual outcome specifically take difference prediction actual value square difference sum square difference observation skykit learn use mean

squared error function directly metric library metric use linear model r squared trained model performed batch transformation test data calculated metric use metric help tune model could select different set feature train model retrain model ask better model metric help inform could also use different data retrain model feature remember k fold cross validation earlier module finally could tune parameter model subject next section key takeaway section module evaluate model need data model seen could either holdout set could use k fold cross validation different machine learning model use different metric classification use confusion matrix auc roc generate regression use mean squared section 7 see next video

Transcript of Mod02_Sect03.wav: hi welcome back continue exploring data collection reviewing extract transform load data data typically spread across many different system data provider present challenge need bring data source together something consumed machine learning model extract transform load also known etl step etl defined way extract pull data source single location extraction might need modify data combine matching record task transform data finally load data loaded repository amazon s3 typical etl framework several component example consider diagram first crawler program connects datastore source target progress ranked list classifier determine schema data creates metadata table aws glue data catalog job defines business logic needed perform etl work run job need use schedule event final note service discussed exist transformed partition etl process aws glue fully managed etl service make simple cost effective categorize data clean enrich move reliably various data store aws glue consists central metadata repository known aws glue data catalog etl engine automatically generates python scala code also provides flexible scheduler handle dependency resolution job monitoring retries aws glue server need set manage infrastructure use aws glue console discover data transform make available search inquiry console call underlying service orchestrate work needed transform data also use aws glue api operation interface aws glue service way edit debug test python scala apache spark etl code using familiar development environment aws glue well suited machine learning receive label data used training example say provide aws glue training data teach model duplicate record data source look like aws glue identify duplicate present analysis data engineer aws glue enables orchestration complex etl job example aws glue crawl data source present information client data catalog aws glue run etl job based event getting new data set example use aws lambda function trigger etl job run soon new data becomes available amazon s3 also register new data set aws glue data catalog part etl job although manage tool available aws manipulate data data scientist also write script jupyter notebook handle data simple extract load script shown import variable section import library used note moto 3 library aws variable also set zip file web location local folder extraction download extract section make web request saving bite url stream stream passed zip file function used extract data extracted file folder upload s3 section enumerates folder file amazon s3 discover script used often migrated stand alone function imported python application part 2 section see part 3 review secure data

Transcript of Mod02_Sect03.wav: hi welcome back module 3 section 5 training section going look select model train data pre processed point done lot clean prepare data mean data completely ready train algorithm algorithm may able work training data data frame format file format like csv commonly used various algorithm make use optimization file format like record io protobuf use many amazon sagemaker algorithm support training data csv format amazon sagemaker requires csv file header record target variable first column amazon sagemaker algorithm work best use optimized protobuf record io format training data using format allows take advantage pipe mode training algorithm support pipe mode training job stream data directly amazon s3 using csv format target variable training data set first column left feature right target variable column evaluating model data trained lead overfitting recall overfitting model learns particular data set well essentially memorizing training data rather learning relationship feature label mean model learning relationship pattern apply new data future hold split data multiple set commonly set training data validation data testing data training data includes feature label feed algorithm selected produce model use model make prediction validation data set likely notice thing want tweak tune change ready run test data set includes feature since want label actually predicted performance get test data set reasonably expect see production common split using holdout method using 80 data training set 10 validation 10 test lot data split 70 training 15 validation 15 small data set use k fold cross validation utilize much data possible still relatively good metric order choose model better k fold cross validation randomly partition data k different segment segment use rest data outside training order validation particular segment let look example five fold cross validation available training data separated five different chunk training first model using chunk training data going calculate metric test second model going use piece training model trained apply test thing 5 time use training data tested five different model different chunk test data eventually testing data point one thing note splitting data data specific order lead bias model especially true working structured data example wine data ordered quality column run model test data ordered pattern applied biasing model might also mean target missing training data typically randomizing data set prior splitting sufficient many library provide function smaller set sometimes useful use stratified sampling stratified sampling ensures training test set approximately percentage sample target class complete set internet search give many way shuffle split data one easiest use train split function sklearn amazon sagemaker provides four different way train model built algorithm available easily deployed aws console cli jupyter notebook container used behind scene use one amazon sagemaker built algorithm deal directly amazon sagemaker supported framework provide pre built container support deep learning framework apache mxn tensorflow pie enchanter also support machine learning library skykit learn spark ml providing pre built docker image use amazon sagemaker python sdk deployed using respective amazon sagemaker sdk estimator class pre built amazon sagemaker container image use modify advanced scenario package scripter algorithm use amazon sagemaker use programming language framework develop container example team work build ml model build container train host algorithm well someone else may already developed tune model worth looking aws marketplace find available model amazon sagemaker provides high performance scalable machine learning algorithm optimized speed scale accuracy supervised learning amazon sagemaker includes xgboost linear learner algorithm classification quantitative regression problem also

factorization machine address recommendation time series prediction problem amazon sagemaker includes support unsupervised learning k mean clustering principal component analysis pca solve problem like identifying customer grouping based purchasing behavior finally selection specialized algorithm processing image deep learning task let look little closer three commonly used built algorithm use case xgboost extreme gradient boosting popular efficient open source implementation gradient boosted tree algorithm gradient boosting supervised learning algorithm attempt accurately predict target variable combining ensemble estimate set simpler weaker model xgboost done remarkably well machine learning competition robustly handle variety data type relationship distribution large number hyperparameters tweaked tuned improved fit flexibility make xgboost solid choice problem regression classification binary multaq ranking amazon sagemaker linear learner algorithm provides solution classification regression problem amazon sagemaker algorithm simultaneously explore different training objective choose best solution validation set also explore large number model choose best one need compare method provide solution continuous objective amazon sagemaker linear learner algorithm provides significant increase speed naive hyperparameter optimization technique k mean unsupervised learning algorithm attempt find discreet grouping within data member group similar possible one another different possible member group find attribute want algorithm use determine similarity train model amazon sagemaker create training job training job includes url amazon s3 bucket stored training data url s3 bucket want store output job amazon elastic container registry path training code stored compute resource want amazon sagemaker use model training compute resource ml compute instance managed amazon sagemaker amazon sagemaker provides selection instance type optimized fit different machine learning use case instance type comprise varying combination cpu gpu memory networking capacity give flexibility choose appropriate mix resource building training deploying ml model instance type includes one instant size allowing scale resource requirement target workload key takeaway section module include split data training testing set help validate model accuracy k fold cross validation help smaller data set two key algorithm supervised learning xgboost linear learner use k mean unsupervised learning use amazon sagemaker train model section 5 hope see next video
Transcript of Mod02_Sect03.wav: time review module wrap knowledge check module learned formulate problem business request obtain secure data machine learning build jupyter notebook using amazon sagemaker outline process evaluating data explain data need pre processed use open source tool examine pre process data use amazon sagemaker train host machine learning model use cross validation test performance ml model use hosted model inference create amazon sagemaker hyperparameter tuning job optimize model effectiveness concludes module thanks watching see next video
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring feature engineering describing work outlier might also need clean data based outlier exist outlier point data set lie abnormal distance value always something want clean add richness data set also make harder make accurate prediction skew value away normal value related feature outlier might also indicate data point actually belongs another column think outlier falling two broad category first

single variation single variable univariate outlier second variation two variable multivariate outlier one common way find univariate outlier box plot box plot show far data point mean variable box plot show data value within two quartile mean value outside range represented line extending box sometimes called whisker scatter plot effective way see multivariate outlier example diagram show amount sulfate alcohol collection wine scatter plot quickly visualize whether multivariate outlier two variable origin outlier likely inform deal pre processing phase pipeline possibly later feature engineering several different approach dealing outlier could delete outlier outlier based artificial error mean outlier natural introduced failure like incorrectly enter data also transform outlier taking natural log value turn reduces variation caused extreme outlier value would reduce outlier influence overall data set finally could use mean feature impute value replace outlier value would good approach outlier caused artificial error exhaustive list describes common option extracted feature need select appropriate feature training model three main feature selection method filter method used statistical method measure relevance feature correlation target variable rapper method measure useful subset feature training model feature measuring successful model filter faster cheaper rapper method involve training model repeatedly rapper typically find best subset feature risk overfitting compared using subset feature filter method embedded method algorithm specific might use combination filter wrapper filter method use proxy measure instead actual model performance fast compute still capture useful feature set common measure first pearson correlation coefficient measure statistical relationship association two continuous variable second linear discriminant analysis lda used find linear combination feature separate two class third analysis variance anova used analyze difference among group mean sample finally chi square single number tell much difference exists observed count count expect absolutely relationship population filter usually le computationally intensive rapper produce feature set tuned specific type predictive model lack tuning mean feature set filter general one rapper filter also usually lower prediction performance rapper however filter feature set contain assumption prediction model useful exposing relationship feature many filter provide feature ranking instead explicit best feature subset cutoff point ranking chosen cross validation filter also used preprocessing step wrapper enables wrapper used larger problem rapper method use predictive model score feature subset new subset used train tested holdout set score subset calculated counting number mistake made holdout set error rate model rapper train new model subset computationally intensive however usually provide best performing feature set particular type model problem forward selection start feature add best model found backward selection start feature drop one time select best model embedded method combine quality filter wrapper method implemented algorithm built feature selection method popular example method lasso ridge regression built penalization function reduce overfitting key takeaway section module feature engineering involves selecting best feature machine learning preprocessing give better data work better data typically provides better result two category pre processing converting data numerical value cleaning dirty data removing missing data cleaning outlier finally handle dirty data impact model section 4 see next video

Transcript of Mod02_Sect03.wav: welcome back aws academy machine learning module 3 going work entire machine learning pipeline using amazon sagemaker module discus typical process handling machine learning problem machine learning pipeline applied many machine learning problem focus supervised learning process learn module adapted type machine learning well large mod covering lot material end module able formulate problem business request obtain secure data machine learning build jupyter notebook using amazon sagemaker outline process evaluating data explain data need pre processed use open source tool examine pre process data use amazon sagemaker train host machine learning model use cross validation test performance machine learning model use hosted model inference finally create amazon sagemaker hyperparameter tuning job optimize model effectiveness ready get started see next video
Transcript of Mod02_Sect03.wav: hi welcome back section one going introduce computer vision computer vision exciting space machine learning think computer vision automated extraction information digital image using computer vision machine identify people place thing image accuracy human level greater speed efficiency computer vision often built deep learning model automates extraction analysis classification understanding useful information single image sequence image image data take many form single image video sequence view multiple camera three dimensional data computing power algorithm advanced last 10 year led increase capability easier access computer vision technology computer vision used primary use case computer vision use image facial recognition improve public safety home security way authenticate access personal device also use automatically classify image content management analysis autonomous driving partly enabled computer vision technology safety feature car lane detection collision avoidance medical image analysis computer vision improve accuracy speed patient medical diagnosis result better treatment outcome life expectancy patient finally manufacturing well trained computer vision incorporated robotics improve quality assurance operational efficiency example probably think computer vision problem broken area content recognition identifying thing image classification problem complex one several layer picture represented breakfast lunch dinner classification food answer depends model use perform classification model must trained training data provides algorithm data learn say model trained picture different type food might expect image output category milk peach mashed potato chicken nugget salad trained model different image could classify object trey cutlery napkin instead work image might want know kind object image location object object detection provides image category object located image set coordinate defining location box surrounding image known bounding box bounding box detection typically left height coordinate surrounding image use coordinate application object detected image confidence number usually associated object percentage indicates probability object belongs specific class confidence level important want determine action based object detection especially facial detection application case action significance object segmentation also known semantic segmentation go detail get fine boundary detected object basically fine grained inference predicting pixel image application require object segmentation include autonomous vehicle advanced computer human interaction object segmentation key problem field computer vision covering course video add another dimension computer vision video get data work

capture movement people object referred instance example detect people enter leave frame also deal moving camera use case computer vision building detection tracking analyze shopper behavior retail store studying path person follows use face analysis understand detail shopper average age range gender distribution expressed emotion without identifying another computer vision use case also analyze image identify action using motion video example activity delivering package dancing looking image baseball player example image could include capturing batter accuracy pitcher pitching style type pitch slow ball slider others inn batter performance versus specific pitcher manager could use data coach player improve performance coach also use data game make game time decision say want initiate various action based speed baseball leaving trajectory hit calculated ml model could lead audio visual warning possible foul ball crowd could result preemptive alarm hit high probability home run mean event following home run could well timed automated playing music setting firework home run hit home team wrap section key takeaway section covered computer vision automated extraction information image divide computer vision two distinct area image analysis video analysis image analysis includes object classification detection segmentation video analysis includes instance tracking action recognition motion estimation thanks watching see next video
Transcript of Mod02_Sect03.wav: hi welcome back module 3 section one going take look data set use module also look guidance formulate business problem get started reminder machine learning pipeline looked previous module map section module section section 1 cover formulate problem also cover data set use throughout module section 2 discus obtain secure data machine learning activity section 3 show tool technique gaining understanding data section 4 look pre processing data ready train model section 5 cover selecting training appropriate machine learning model section show deploy model make prediction section 7 examine process evaluating performance machine learning model finally section 8 look tuning model machine learning pipeline iterative process work real world problem might find iterating many time arrive solution meet business need first section examine think turning business requirement machine learning problem first step phase simply define problem want solve goal want reach understanding business goal key use measure performance solution unusual solidify business problem begin targeting solution lot question ask develop good understanding problem information problem begin framing approach problem even solved machine learning would traditional approach make sense supervised unsupervised machine learning problem labeled data train supervised model many question could ask business ultimately try validate use machine learning make sure access right people data also try come simplest solution problem example want identify fraudulent credit card transaction stop transaction process problem business goal outcome driving problem statement case say intended outcome reduction number customer end membership credit card result fraudulent transaction business perspective define success given problem desired outcome stage need move qualitative statement quantitative statement easily measured continue example metric could use define success problem might 10 reduction number customer file claim fraudulent transaction within 6 month period define business side problem time start thinking term machine learning model actual output want see model want specific statement reflects ml model could actually output

example might model output whether credit card transaction fraudulent fraudulent know want ml model actually achieve use information determine type ml working historical data customer filed report fraud transaction use data machine learning purpose historical data fall supervised learning approach label already defined recall earlier course supervised ml type categorized two group classification regression credit card example desired output categorizing transaction fraud fraud see dealing binary classification problem throughout module see several data set used access data set many uc irvine machine learning repository first data set contains numerical information composition wine along quality wine question might want ask data set based composition wine could predict quality therefore price addition question also use data set view statistic deal outlier scale numerical data second data set car evaluation database data set heavily text based enables explore encoding categorical data convert text value number processed machine learning third data set biomedical data set would also use lab question answer data set based biomechanical feature predict patient abnormality data set take entire end end process end trained model tuned use make prediction section looked business problem need converted ml problem also looked key question ask defining success measure outcome impact solution implemented business problem fall one two category first category classification binary multiclass ask target belong class second category regression ask predict numerical value section 1 see next video
Transcript of Mod02_Sect03.wav: welcome back section review five manage machine learning service use various use case service simplify process creating machine learning application start looking amazon transcribe use amazon transcribe recognize speech audio file produce transcription recognize specific voice audio file create customized vocabulary term specialized particular domain also add transcription service application integrating web internet protocol use two way communication application amazon transcript common use case amazon transcribe medical professional record note amazon transcribe capture spoken note text also video production organization generate subtitle automatically video could also done real time live feed add closed captioning medium company use amazon transcribe capture label content feed content amazon comprehend analysis last company record customer service sale call transcribe analyze result training strategic opportunity amazon polly convert text lifelike speech input either plain text file file formatted speech synthesis markup language ssml ssml markup language used provide special instruction speech sound example want introduce pause flow speech add ssml tag instructs amazon polly pause two word also output speech amazon polly mp3 vorbis pcm audio stream format amazon polly eligible use certain regulated workload example eligible use u health insurance portability accountability act 1996 hipaa amazon polly also eligible use payment card industry data security standard pci ds common use case amazon polly first example major news company using amazon polly generate vocal content directly written story also embedded mapping apis developer add voice geo based application language training company used amazon polly create system learning new language finally animator used add voice character amazon translate create multi language experience application create system read document one language render store another language also use part document analysis system amazon translate fully integrated machine learning service amazon comprehend amazon

transcribe amazon polly integration extract named entity sentiment key phrase integrating amazon comprehend create multilingual subtitle amazon transcribe speech translated content amazon polly common use case amazon translate first use case building international website use amazon translate quickly globalize website amazon translate also used develop multilingual chatbots chatbots used create human like interface application amazon translate create chatbot speaks multiple language another use case software localization localization major cost software aimed global audience amazon translate decrease software development time significantly reduce cost localizing software final example use case international medium management company manage medium global audience used amazon translate reduce cost localization amazon comprehend implement many nlp technique reviewed earlier module extract key ent perform sentiment analysis tag word part speech common use case amazon comprehend first example analyzing legal medical document legal insurance medical organization used amazon comprehend perform many nlp function reviewed module another large scale mobile app analysis mobile app developer use amazon comprehend look pattern usage apps design improvement financial fraud detection another use case amazon comprehend banking financial institution used examine large data set financial transaction uncover fraud look pattern illegal transaction finally used content management medium content company use amazon comprehend tag content analysis management amazon lex add human language front end application amazon lex let use conversational engine power amazon alexa automatically increase capacity amazon lex solution creating aws lambda function scale demand also store log file conversation analysis common use case amazon lex first use case building front end interface inventory management sale voice interface becoming common company used amazon lax add chatbots inventory sale application another use amazon lex creating customer service interface human like voice application quickly becoming standard many customer service application amazon alexa reduce time take develop chatbots increase quality amazon alexa also used develop interactive assist combining amazon lex ml service customer creating sophisticated assistance many different industry final example use case querying database human like language amazon lex combined aws database service create sophisticated data analysis application human like language interface main point take away module amazon transcribe automatically convert spoken language text amazon polly convert written text spoken language amazon translate create real time translation language amazon comprehend automate many nlp use case reviewed module finally amazon lex create human like interface application thanks watching see next video
Transcript of Mod02_Sect03.wav: time summarize main point module module learned describe use case computer vision describe amazon management learning service available image video analysis list step required prepare custom data set object detection describe amazon sagemaker ground tr used prepare custom data set use amazon recognition perform facial detection concludes introduction computer vision thanks watching see next video
Transcript of Mod02_Sect03.wav: hi welcome back review find correlation data set quantify linear relationship among variable seeing scatter plot correlation matrix good tool situation conveys strong weak linear relationship among numerical variable correlation go high one low 1 correlation one mean two

numerical feature perfectly correlated like saying proportional x correlation two variable 1 like saying proportional x linear relationship quantified correlation correlation 0 mean linear relationship mean relationship indication linear relationship two variable however looking number always straightforward often easier view number represented color highest number one dark green minus one dark brown color give positive negative direction also show strong correlation use seaborn heat map function show correlation matrix looking chart correlation citric acid fixed acidity would expected wine citric acid contributes acidity wine however much correlation fixed acidity ph ph measurement strength acid present fixed acidity measure quantity particular data set appear correlation key takeaway section module include point first step get data format used easily panda popular python library working data descriptive statistic help gain insight data use visualization examine data set detail section see next video

Transcript of Mod02_Sect03.wav: hi welcome section 1 section going talk machine learning course introduction machine learning also known ml first discus machine learning fit larger picture machine learning subset artificial intelligence ai broad branch computer science focused building machine human task deep learning subdomain machine learning understand fit together discus one mentioned machine learning subset broader computer science field known artificial intelligence ai focus building machine perform task human would typically perform contemporary popular culture probably seen ai movie television work fiction example might seen eye control world around start acting initiative ai started computer agent perceived action achieve specific goal maybe outcome created originally wished fictional ai interact extensively human helper worker generally better job working humanity general purpose kind ai example artificial general intelligence agi capacity learn understand task human ai problem typically span many field research natural language processing reason knowledge representation learning perception physical environment interaction eli yet reality u living simulation every year move closer area might also read seen commentary ethic creating ai view positive malicious fictional ai want destroy humanity use power source perhaps concerned risk mass unemployment intelligent machine could work 24 7 need break worry though going build next rogue ai course maybe next one search many definition machine learning universally agreed upon definition start looking couple definition example could say machine learning scientific study algorithm statistical model perform task using inference instead instruction bad starting point key point using algorithm statistical model instead instruction help better understand apply idea concrete example suppose need write application determines email message spam without machine learning need write complex series decision statement using else statement also need use word subject body number link length message determine email message spam would hard labor intensive build large set rule covering every possibility machine learning however could use list email message marked spam spam train machine learning model model would learn pattern word length attribute good indicator spam message presented model email message seen model would perform prediction say whether message spam spam deep learning represents significant leap forward capability artificial intelligence machine learning theory behind deep learning created human brain work artificial neural network inspired biological neuron

found brain although implementation become different artificial neuron one input single output neuron fire activate output based transformation input neural network composed layer artificial neuron connection layer typically input output hidden layer network output single neuron connects input neuron next layer network asked solve problem input layer populated training data neuron activate throughout layer answer presented output layer accuracy output measured output meet threshold training repeated slight change weight connection neuron neural network repeatedly time strengthens connection lead success diminishes connection lead failure see course machine learning practitioner spend lot time optimizing ml model selecting best data feature train contrast deep learning practitioner spend almost time task instead spend time modeling data different n architecture theory deep learning go back decade hardware needed run deep learning problem generally accessible recently available use deep learning address problem complex problem could worked mainstream machine learning recent occurrence rapid advancement machine deep learning started around mid 2000s partly moore law rise cloud computing resulted easier access larger faster cheaper compute storage capability rent computing power hour penny needed substantial investment buy operate large scale compute cluster 2012 neural network started used image net large scale visual recognition machine learning competition image recognition accuracy rate jumped 82 steadily climbing ever since fact exceeded human performance 2015 key takeaway section also machine learning subset ai focus using data train machine learning model make prediction deep learning technique inspired human biology us layer artificial neuron build network solve problem advancement technology cloud computing algorithm development led corresponding advance machine learning capability application section see next video
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring video analysis reviewing create test data set final step train model identify test data set use test data set validate evaluate model performance performing inference image test data set compare result labeling information training data set create test data set alternatively use amazon recognition custom label split training data set two data set using 80 20 split split mean 80 data used training 20 used testing define training test data set amazon recognition custom label automatically train model service automatically load inspects data select correct machine learning algorithm train model provides model performance metric charged amount time model take train data set contains image label take longer train training complete evaluate performance model testing amazon recognition custom label predicts test image contains custom label confidence score value quantifies certainty model prediction classification problem result mapped confusion matrix true positive model correctly predicts presence custom label test image predicted label also ground truth label image example amazon recognition custom label correctly return cat label cat present image false positive model incorrectly predicts presence custom label test image predicted label ground truth label image example amazon recognition custom label return cat label cat label ground truth image play false negative model predict custom label present image ground truth image includes label example amazon recognition custom label return cat custom label image contains cat true negative model correctly predicts custom label present test image example amazon recognition

custom label return cat label image contain cat console provides access true positive false positive false negative value image test data set prediction result used calculate various metric label aggregate metric entire test set definition apply prediction model make bounding box level bounding box metric calculated bounding box test image regardless whether box prediction ground truth help amazon recognition custom label provides various metric example view summary metric evaluation metric label also provides precision metric label average precision metric entire test data set precision proportion positive result correctly classified amazon recognition custom label provides average recall metric label average recall metric entire test data set recall fraction test set label correctly classified using previous example cat would many cat correctly classified service also provides average model performance score label average model performance score entire test data set f1 score combine precision recall together give one number quantifies overall performance particular machine learning algorithm might use f1 score class imbalance also want preserve equality precision sensitivity higher value mean better model performance recall precision satisfied accuracy model start using part 3 section see part 4 review evaluate improve model
Transcript of Mod02_Sect03.wav: hi welcome back module 3 section look evaluate model success predicting result point trained model time evaluate model determine good job predicting target new future data future instance unknown target value need ass model perform data already know target answer use assessment proxy performance future data reason hold sample data evaluating testing important part phase involves choosing appropriate metric business situation think back earlier section problem formulation phase define business problem outcome craft business metric evaluate success model metric choose phase linked business metric much possible often high correlation two metric addition considering business problem success metric type ml problem working influence model metric choose throughout rest module look example common metric used classification problem also look common metric used regression problem going start considering simple binary classification problem specific example imagine simple image recognition model labeling data either cat cat model trained use test data set held back perform prediction help examine performance model compare predicted value actual value plot value table like example start getting insight well model performed confusion matrix get high level comparison predicted class matched actual class actual label class cat identified p positive predicted label class also cat true positive good outcome model similarly actual label cat identified n negative predicted label class also cat true negative also good outcome model case model predicted correct outcome use testing data two possible outcome considered good outcome first one actual class neg predicted class positive cat called false positive prediction positive incorrect finally false negative happen actual class positive got cat predicted class negative cat part one section see part 2 review calculating classification metric
Transcript of Mod02_Sect03.wav: hi welcome back section 5 going discus challenge machine learning come across many challenge machine learning lot poor quality inconsistent data available significant portion job getting access generating enough good data representative problem want solve key issue watch overfitting

model data although mostly data science experience staffing team data scientist cost effective management support using machine learning business landscape look like problem complex formulate machine learning problem resulting model explained business explained might get adopted cost building updating operating machine learning solution finally technology map business unit access data needed data secured meet regulatory requirement tool framework used solution integrate system important question successful need able answer address many machine learning problem solved today using existing model without substantial machine learning knowledge already talked aws managed service machine learning add sophisticated machine learning capability application basic developer skill calling apis pre built model use adapt one example yolo mean look yolo popular computer vision model addition scenario use aws marketplace prefer buy model service independent software vendor instead developing key takeaway section face many machine learning challenge biggest one directly influence related data consider managed service solve machine learning problem within domain support using amazon recognition computer vision problem section see next video Transcript of Mod02_Sect03.wav: hi welcome amazon academy machine learning foundation module learn course objective various job role machine learning domain go learn machine learning completing module able identify course prerequisite objective indicate role data scientist business identify resource learning going look prerequisite taking course take course recommend first complete aws academy cloud foundation also general technical knowledge including foundational computer literacy skill like basic computer concept email file management good understanding internet also recommend intermediate skill python programming general knowledge applied statistic finally general business knowledge important course includes insight information technology used business also important business related skill set communication skill leadership skill orientation towards customer service course introduced key concept machine learning tool us also introduced work aws service machine learning learn recognize machine learning deep learning part artificial intelligence describe artificial intelligence machine learning terminology identify machine learning used solve business problem describe machine learning process list tool available data scientist identify use machine learning instead traditional software development method part course also learn implement machine learning pipeline includes formulate problem business request obtain secure data machine learning build jupyter notebook using amazon sagemaker outline process evaluating data explain data need pre processed use open source tool examine pre process data also use amazon sagemaker train host machine learning model use cross validation test performance machine learning model use hosted model inference create amazon sagemaker hyperparameter tuning job optimize model effectiveness finally use managed amazon machine learning service solve specific machine learning problem forecasting computer vision natural language processing review course outline achieve course objective complete following module start module 2 get introduction machine learning module 3 learn implement machine learning pipeline amazon sagemaker module 5 6 describe apply managed amazon machine learning service computer vision natural language processing finally module 7 summary course also includes overview step take work towards aws certified machine learning specialty next five slide provide detail subtopics covered module purpose module 2 introduce major concept understanding machine learning section 1 describes overall field machine learning machine learning relates artificial intelligence deep learning section 2 learn common business problem solve machine learning section 3 describes general workflow solving machine learning problem also learn common machine learning term section 4 review commonly used tool machine learning professional lastly section 5 get overview common challenge face working machine learning problem module 3 get introduction amazon sagemaker use implement machine learning pipeline module focus application machine learning solve problem several public domain data set example machine learning pipeline section 1 introduces defining business problem data set use module section 2 8 describe phase machine learning pipeline using computer vision example application section 2 learn collect secure data section 3 describes different technique evaluating data section 4 learn process feature engineering section 5 describe step take train model sagemaker section 6 get overview option sagemaker hosting using model finally section 7 8 cover evaluate tune model sagemaker module introduced using machine learning create forecast based time series data section 1 introduced forecasting common application section 2 outline pitfall using time series data make forecast finally section 3 get overview use amazon forecast module learn using machine learning computer vision section 1 describes general problem solve computer vision section 2 learn process analyzing image video section 3 learn step need take prepare data set computer vision module introduced natural language processing machine learning section 1 learn general set problem solve natural language processing section 2 review managed amazon machine learning service use address natural language processing problem service include amazon transcribe amazon translate amazon lex amazon comprehend amazon polly module 7 final module course module review learned throughout course also introduced next step take want achieve aws certified machine learning specialty section 1 module summarizes topic covered course section 2 learn aws documentation also review two common framework applying aws service finally section 3 describes step take want continue working towards aws certified machine learning special section learn common job role machine learning professional interested data scientist role focus developing analytical statistical programming skill data scientist use skill collect analyze interpret large data set university offered degree data science data scientist often degree related field like statistic computer science economics data scientist need technical competency statistic machine learning programming language add data analytics like career machine learning engineer skill need similar data scientist skill set like data scientist machine learning engineer also require technical competency statistic machine learning however focus programming skill software architecture analysis interpretation machine learning engineer apply programming architecture skill design develop machine learning system machine learning engineer often previous experience software development rely heavily programming software engineering machine learning role might also interested career science apply machine learning technology field machine learning impact everything astronomy zoology many different path open applied science researcher primary focus type science working need skill data scientist also need know apply skill chosen domain applied science rule also require technical competency statistic machine learning many software developer integrating machine learning application interested career software developer also include machine learning technology study machine learning developer primary focus software development skill also need skill data scientist make sure take coursework statistic applied mathematics final note module recommend reviewing student guide find link documentation resource use throughout course introduction thanks watching see next video

Transcript of Mod02_Sect03.wav: hi welcome back going look way collect secure data section explore technique challenge associated collecting securing data needed machine learning consider original example predicting credit card fraud formulated problem data need actually train model get desired output subsequently achieve intended business outcome access data much data solution use bring data one centralized repository answer question essential stage good news budding data scientist many place obtain data private data existing customer already exists including everything log file customary invoice database private data useful depending problem trying solve many case private data found many different system look bring source together shortly sometimes want use data collected made available commercial organization company reuters change healthcare dun bradstreet foursquare maintain database subscribe include curated news anonymized healthcare transaction global business record location data supplement data commercial data get useful insight gotten otherwise also many open source data set ranging wine quality movie review data set made available use research teaching purpose aws kaggle uci machine learning repository good place find open source data set government health organization source data could useful supervised machine learning problem need lot data also called observation already need know target answer prediction data kind data already know target answer prediction called labeled data observation data made two element target end feature target answer want predict credit card transaction example target given observation either fraud fraud feature attribute example use identify pattern predicting target answer feature credit card example could date transaction vendor amount dollar transaction might wonder source target fraud fraud typically information discovered transaction complete actual card owner notice fraudulent transaction statement information would recorded transaction exactly purpose using train future model given know element ml data set return one original question data need actually train model reach desired output subsequently intended business outcome example stage ml pipeline crucial get domain expertise help answer question domain knowledge start determining feature target data model need make accurate prediction data representative data using model make prediction example want predict credit card fraud need collect data positive fraudulent transaction also need collect data negative non fraudulent transaction need type data machine learning algorithm find pattern distinguish two type suppose average amount fraudulent transaction actually 3 put training data set includes small fraction fraudulent observation say 0 4 case difficult model truly learn pattern related fraudulent transaction might encounter production many different service aws find store data key service might use amazon simple storage service also known amazon s3 provides object level storage s3 store much data want form object think file could csv file file format need s3 accessed web based aws management console also access s3 programmatically api sdk third party solution also use api sdks training data already s3 planning run

training job several time different algorithm parameter could use amazon fsx lustre file system service speed training job serving s3 data amazon sagemaker high speed first time run training job fsx lustre automatically copy data s3 make available sagemaker use amazon fsx file system subsequent iteration training job prevents repeated downloads common s3 object alternatively training data might already amazon elastic file system amazon efs recommend using efs data source training data launch training job directly service without needing data movement result faster training start time often case environment data scientist home directory amazon efs quickly iterate model bringing new data sharing data colleague experimenting different field label data set example data scientist use jupyter notebook initial cleansing training set launched training job amazon sagemaker could use jupyter notebook drop column relaunch training job finally compare resulting model see one work better many aws service resource might find data example could use amazon relational database service amazon rds managed relational database service also use amazon redshift manage data warehouse service another option amazon timestream manage time series database designed specifically handle large amount data internet thing iot could even spin instance amazon elastic compute cloud also known amazon ec2 host database instance data source need extract useful data source assembling data machine learning look next part 1 section see part 2 review extract transform load data
Transcript of Mod02_Sect03.wav: hi welcome section 1 get started reviewing forecasting use case forecasting important area machine learning important many opportunity predicting future outcome based historical data many opportunity involve time component however time component add additional information also make time series problem difficult handle compared type prediction think time series data falling two broad category first univariate data mean one variable second one multivariate data mean one variable several common pattern time series data first pattern trend trend get pattern value increasing decreasing staying time seasonal pattern reflect time year month day pattern cyclical pattern similar seasonal pattern pattern repeat like large retail sale event happens time year finally change data time appear random discernible pattern many us forecasting use forecasting marketing application search sale forecasting demand projection could also used inventory management system anticipate required inventory level forecasting energy consumption help predict energy needed weather forecasting system used government commercial application agriculture section see next video
Transcript of Mod02_Sect03.wav: hi welcome module 4 aws academy machine learning module going look forecasting start introduction forecast look time series data different kind data going look amazon forecast service help simplify building forecast end module able describe business problem solved amazon forecast describe challenge working time series data list step required create forecast using amazon forecast use amazon forecast make prediction see next video
Transcript of Mod02_Sect03.wav: hi welcome back time review module wrap module learned describe business problem solved amazon forecast describe challenge working time series data list step required create forecast using amazon forecast use amazon forecast make prediction thanks participating see next mod
Transcript of Mod02_Sect03.wav: hi welcome back section 2 going focus processing

time series data different type data using far time series data data captured chronological sequence defined period time introducing time machine learning model positive impact model derive meaning change data point time time series data tends correlated mean dependency data point mixed result forecasting dealing regression problem regression assumes data point independent need develop method dealing data dependence increase validity prediction addition time series data add related data augmenta forecasting model example suppose want make prediction retail sale could include information product sold item identification sale along number unit sold per time period third type data metadata data set instance say retail data set might want include metadata like brand name genre music video group result better data work multiple data source face challenge handling timestamp data observe difference timestamp format challenge incomplete data however might able infer missing data case example say data contains month day year observe whether data seems sequence month number database repeating 12 00 could add year knew data started infer future year based order data much time stamp data stored utc form data check timestamp local universal time sometimes timestamp represent time think example suppose database car serviced garage timestamp indicate time car arrived completed picked indicate final entry entered system say trying model hourly caloric intake patient however daily data need adjust target time scale also data might time stamp could way extrapolate time series depending data domain example might wavelength measurement vector within image final note remember daylight saving different around world also daylight saving time might even occur twice year time zone common occurrence real world forecasting problem missing value raw data missing value make harder model generate forecast primary example retail stock situation demand forecasting item go stock sale day zero forecast generated based zero sale value forecast incorrect many reason value marked missing missing value occur transaction also occur possible measurement error example service monitored certain data working correctly another example measurement happen correctly retail primary example inability take correct measurement stock situation demand forecasting mean demand equal sale day several way calculate missing data first method forward fill us last known value missing value building idea moving average us average last known value calculate missing value backwards us next known value missing value danger using future calculate past bad forecasting method also known look ahead avoided interpolation us equation calculate missing value also use zero fill often used retail missing sale data calculated missing data represents order day would wise investigate happened case want fill missing value might get data different frequency example might sale data includes exact time stamp sale recorded inventory data contains year month day inventory level data different frequency data set data compatible question might need downsample sampling moving finely grained time le finally grain time example show could converting hourly data set daily data set downsampling need decide combine value previous case sale data summing quantity make sense data temperature might want find average understanding data help decide best course action opposite downsampling upsampling move le finely grained time filing grain time problem upsampling extremely difficult achieve case suppose want upsample sale data daily sale hourly sale unless data source reference able case need something perhaps match

frequency another time series might irregular time series specific domain knowledge would help case need careful make conversion retail example best create single order day specified hour temperature could copy daily temperature hourly slot use formula calculate curve data science outlier mix positive negative attribute true time series data suppose examining sale data order unusually high number item might want include forecast calcul order size might never repeated removing outlier anomaly known smoothing smoothing data help deal outlier anomaly reason might consider smoothing data preparation remove error value could also remove outlier might also want smooth data generate feature visualization could smooth data reduce noise plot important understand smoothing data impact might outcome might reduce noise create better model equally important question could smoothing compromise model model expecting noisy data also able smooth data production part 1 section see part 2 review time series specific challenge tool algorithm help u wrangle data
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring video analysis reviewing create training data set data set contain information needed train test amazon recognition custom label model image label bounding box use image amazon s3 upload computer s3 part process train model data set least two label least 10 image per label image data set must labeled mentioned earlier use amazon recognition custom label console amazon sagemaker ground truth label image train amazon recognition custom label model image must labeled label indicates image contains object scene concept mentioned earlier data set need least two defined label also image must least one assigned label identifies object seen concept image apply label image whole label known image level label useful identifying scene concept want detect example one image show u beach scene ko olina island oahu u state hawaii train model detect beach add beach label applies entire image also apply label specific area image contain object want detect example want model detect amazon echo device must identify different type echo device image model need information device located image need corresponding label identifies type device disinformation known localization information location device expressed bounding box example object bounding box image show u bounding box surround amazon echo dot image also contains amazon echo without bounding box output labeling process manifest file manifest file image level label typically contains label class name along metadata image labeled object detection manifest contains information labeled image bounding box identifies object image along label bounding box belongs mentioned amazon sagemaker ground truth time well look might help sagemaker ground truth build high quality training data set machine learning model use create data set need labeling provide detailed instruction need labeled submit job decide process image create label data set use worker amazon mechanical turk service vendor company internal workforce machine learning use label data set output sagemaker ground truth train model also use amazon recognition custom label sagemaker ground truth use active learning automate labeling input data active learning machine learning technique identifies data labeled worker sagemaker ground truth functionality called automated data labeling automated data labeling reduce time cost take label data set compared using human worker use automated labeling incur amazon sagemaker training inference cost yes said use machine learning label image use machine learning talk work sagemaker ground truth start

automated data labeling job selects random sample input data object send human worker label data returned sagemaker ground truth us data validation data validate model trained automated data labeling sagemaker ground truth run batch transform job using validated model inference validation data batch inference produce confidence score quality metric object validation data automated labeling determines confidence score object produced step 5 meet required threshold determined step four confidence score meet threshold expected quality automatic labeling exceeds requested level accuracy object considered automatically labeled step six produce data set unlabeled data confidence score sagemaker ground truth selects data point low confidence score data set sends human worker additional labeling sagemaker ground truth us existing human label data additional human label data train new model process repeated data set fully labeled another stopping condition met example automatic labeling stop meet budget human annotation recommend using automated data labeling large data set minimum number object allowed automated data labeling 1 250 however strongly suggest providing minimum 5 000 object part 2 section see part 3 review evaluate improve model
Transcript of Mod02_Sect03.wav: hi welcome module 6 aws academy machine learning introduction natural language processing module introduce natural language processing also known nlp section includes description major challenge faced nlp overall development process nlp application review 5 aws service use speed development mlp based application completing module able describe nlp use case solved using managed amazon ml service describe managed amazon ml service available nlp let get started
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring wrangling time series data seasonality data kind repeating observation frequency observation stable example sale typically see higher sale end quarter fourth quarter consumer retail see even higher sale fourth quarter aware data multiple type seasonality data set many time incorporate seasonality information forecast instance localized holiday good example sale chart show total revenue generated arcade strong correlation number computer science doctorate awarded u correlation mean causation disagree see source chart many correlation plotted site none make sense data careful seeing acting correlation meaning real world experiment generate two random time series data set number 0 1 find low correlation introduce slope data set see strong correlation need know stable system level stability stationarity inform much expect system past behavior inform future behavior system low stability successful predicting future often want determine trend time series adjust series trend difficult compare another series also adjusted trend trend might dominate value series could lead overestimate correlation two series like discussed previously autocorrelation one special problem face time series data seen machine learning problem goal building ml model make sure separating signal noise autocorrelation form noise separate observation independent time series autocorrelation might overstate accuracy model produced algorithm look module help correct autocorrelation factor along seasonality influence model select produce forecast algorithm handle seasonality autocorrelation others panda developed financial data analysis mind good handling time series data set index panda dataframe date time use date time select data use range contain partial date also extract date part

weekday name grouping resampling task panda built function panda give insight autocorrelation information panda time series refer panda documentation one task building forecasting application choose appropriate algorithm choice algorithm determined type data set using feature data amazon forecast support five algorithm others algorithm handle data slightly different characteristic example take auto regressive integrated moving average also known arima remove auto correlation influence pattern observation take exponential smoothing also known ets algorithm useful data set seasonality find algorithm amazon forecast documentation key takeaway section module include time series data sequence data includes time element make different regular data set time challenge include dealing different time format handling missing data downsampling upsampling smoothing dealing seasonality weekday yearly cycle avoiding bad correlation panda excellent time series support function dealing time five algorithm used amazon forecast arim deep plus ets ntts profit section see next video
Transcript of Mod02_Sect03.wav: hi welcome back section look use amazon forecast create predictor generate forecast generate forecast apply machine learning development pipeline seen throughout course still need data need import much data historical data related data want basic evaluation feature engineering use data train model meet requirement amazon forecast train predictor need choose algorithm sure algorithm best data amazon forecast choose select auto ml algorithm also need select domain data sure best fit also select custom domain domain specific type data require trained model use model make forecast using input data set group generated forecast query forecast also export bucket amazon s3 finally encrypt data forecast exporting overall process working amazon forecast import historical related data amazon forecast spec data identifies key data selects appropriate algorithm us algorithm train optimize custom model produce predictor create forecast applying predictor data set retrieve forecast aws management console export forecast comma delimited file also use api aws cli command create retrieve forecast work amazon forecast select domain working domain ranging retail web traffic also custom option everything else selecting domain improve efficiency predictor domain specific type data supply build predictor example retail domain expects data item identifier timestamp observation number sale item specified timestamp example data need provide retail demand forecast time series need time transaction took place ideally utc format item id item many item sold metadata item might include category item color attribute link back time series data item id item metadata typically change related data creating useful forecast could include sale price promotion data link back item must include timestamp item id example data need provide web traffic forecast time series need web page id number page view per month timestamp related data creating useful forecast could include page category navigation content category also need geographic identifier web client metadata might also need provide region sale promotion information amazon forecast predictor use algorithm train model use model make forecast using input data set group help get started amazon forecast provides predefined algorithm arima deep plus ets nbt profit also use auto ml feature try algorithm see one best predicting data prepare data training machine learning typically hold back data use validate score model data hold back usually random sample available data time series data must process data differently correlation time import data

amazon forecast break training test data set diagram show training data used train model tested data held back specify multiple back test window split data multiple time train model use metric determine model give best result default back test window one change amazon forecast split data setting back test window offset parameter create predictor set value algorithm use default value trained model need measure accuracy learn next first amazon forecast evaluation metric weighted quantile loss w quantile los amazon forecast creates forecast provides probabilistic prediction three distinct quantile 10 50 90 prediction quantiles show much uncertainty associated forecast p10 quantile predicts 10 time true value le predicted value example suppose retailer want forecast product demand winter glove sell well fall winter say sufficient storage space cost invested capital high price overstocked winter glove concern might use p10 quantile order relatively low number winter glove know p10 forecast overestimate demand winter glove 10 time sold winter glove 90 time p50 quantile predicts 50 time true value le predicted value continuing winter glove example say know moderate amount demand glove concerned overstocked might choose use p50 quantile order glove p90 quantile predicts 90 time true value le predicted value suppose determine understocked glove result large amount lost revenue example cost selling glove extremely high cost invested capital low case might choose use p90 quantile order glove amazon forecast also calculates associated loss error quantile weighted quantile lost calculate far forecast certain quantile actual demand either direction lower w quantile lost metric mean model forecast reliable root mean square error rmse another method evaluating reliability forecast like w quantile lost rmse calculates far forecasted value actual test data rmse find difference actual target value data set forecasted value time period square difference example show calculate rmse rmse value represents standard deviation prediction error test good forecast validity arrow mostly many outlier lower rmsc metric indicate model forecast reliable example web retailer might use accuracy metric evaluate forecast retailer want predict demand sale particular brand shoe input sale record brand amazon forecast create predictor predictor provides forecasted demand 1 000 pair p10 p50 p90 value shown weighted quantile lost value indicate 10 time fewer 880 pair sold 50 time fewer 1 050 pair sold 90 time fewer 1 200 pair sold retailer use value determine level inventory hold base decision assessment risk able fulfill order excess inventory key takeaway section module include use amazon forecast train use model time series data specific schema defined domain retail ec2 capacity planning use custom schema need supply least time series data also provide metadata related data add information model supervised machine learning problem data split training testing data take new account time element use rmse w quantile lost metric evaluate efficiency model video see next one
Transcript of Mod02_Sect03.wav: hi welcome back continue exploring feature engineering reviewing clean data set addition converting string data numerical data need clean data set several potential problem area encoding string data make sure string consistent also need make sure variable use consistent scale example one variable describes number door car scale probably 2 8 another variable describes number car particular type sold state california scale probably thousand data item might also capture one variable single value instance suppose data set includes variable combine safety maintenance single

variable say high maintenance need train machine learning system variable also split single variable two separate variable might also encounter data set missing data variable data set include outlier cover technique dealing situation section might find data missing example column data set could missing data data collection error maybe data collected particular feature data collection process underway missing data make difficult accurately interpret relationship related feature target variable regardless data ended miss important deal issue unfortunately machine learning algorithm handle missing value automatically need use human intelligence update missing value data meaningful relevant problem python library data manipulation include function finding missing data decide drop impute missing value question answered part better understanding value came missing first place much data missing value represent within larger data set instance say missing value randomly spread throughout data set represent larger portion respective row column case imputation likely better option contrast say column road large percentage missing value case dropping entire row column would preferred amputation decide drop row missing data use built function example panda drop na function drop row missing data drop specific data value using subset alternative dropping missing value impute value missing value different way impute missing value categorical value missing value usually replaced mean median frequent value numerical continuous variable missing value usually replaced mean median impute single row missing data known univariate also multiple row known multivariate look univariate example computer function used impute missing value fairly small data set two missing value missing value imputed strategy mean first calculate mean mean 3 2 2 5 impute mean value missing value data library include impute package provides complex way impute data example include k nearest neighbor soft cute multiple imputation chained equation others part two section see part 3 review work outlier data
Transcript of Mod02_Sect03.wav: hi welcome back section 6 going look hosting using model section look deploy train model consumed application trained tuned tested model learn testing next section ready deploy model thinking looking phase order discussing deployment want test model get performance metric first need make inference prediction model requires deployment deployment testing different production although mechanic amazon sagemaker provides everything need host model simple testing evaluation request deployment handling ten thousand request two way deploy model single prediction deploy model amazon sagemaker hosting service sagemaker deploy multiple computer run model behind load balanced endpoint application call api endpoint make prediction model scale number instance based demand get prediction entire data set use amazon sagemaker batch transform instead deploying maintaining permanent endpoint sagemaker spin model perform prediction entire data set provide store result amazon s3 shuts terminates compute instance useful performing batch prediction test model quickly run entire validation set model without writing code process collate individual result goal deployment phase provide managed environment host model providing inference securely low latency model deployed production monitor production data retrain model necessary newly deployed model need reflect current production data new data accumulated time could potentially identify alternative new outcome deploying model one time exercise instead continuous process one click deploy model amazon ml instance automatically scale across

multiple availability zone higher redundancy specify type instance maximum minimum number instance desired sagemaker take care rest launch instance deploy model set secure http endpoint application application need include api call endpoint achieve inference low latency high throughput architecture integrate new model application minute change model longer need change application code sagemaker manages production compute infrastructure behalf perform health check apply security patch conduct routine maintenance built amazon cloud watch monitoring logging trained model create endpoint either code using sagemaker console planning host single model create endpoint model planning host multiple model need create multi model endpoint multimodal endpoint provide scalable cost effective solution deploying large number model use shared serving container enabled host multiple model reduces hosting cost improving endpoint utilization compared using single model endpoint also reduces deployment overhead sagemaker manages loading model memory scaling model based traffic pattern deploy machine learning model production make prediction new data need make sure apply data processing step used training inference request otherwise get incorrect prediction result using inference pipeline reuse data processing step model training infer without maintaining two separate copy code help ensure accuracy prediction reduces development overhead sagemaker managed service inference pipeline completely managed play pipeline model service installs run sequence container ec2 instance endpoint batch transform job additionally sequence feature processing inference run low latency container correlated ec2 instance key takeaway section module include point deploy train model using sagemaker handle api call application perform prediction using batch transformation goal model generate prediction answer business problem sure model generate good result deploy production finally use multi model endpoint support save resource multiple model deploy section see next video
Transcript of Mod02_Sect03.wav: hi welcome back section 3 going give quick high level overview machine learning terminology typical workflow cover topic detail later course focus larger picture begin always start business problem team believe could benefit machine learning want problem formulation phase one task articulate business problem convert ml problem formulated problem move data preparation pre processing phase pull data one data source data source might difference data type need reconciled form single cohesive view data need visualize data use statistic determine data consistent used machine learning look data source later course example data four column containing data three different data source source slightly different way representing data result shown table ml problem column represent feature rose represent instance issue data instance case need subject matter expert functional understand authenticity data example date represented 11 2 1969 could november 2nd february 11th year 1969 someone owns manages data pool would able clarify ambiguity also word male probably attributed import issue sell shifted position could outside chance actual location molly city capital republic maldives time error identification simple need sme review data learn role expert later remember one largest impact success machine learning project consistent correct data data good shape time train model process get iterative fluid likely go many multiple pass feature engineering training evaluating tuning find model meet business goal feature engineering process selecting creating feature model trained feature column data

```
data set goal model correctly estimate target value new data ml algorithm use
feature predict target example target data average number step taken week
selecting correct feature get involve adding removing calculating new feature
might want make data format consistent consistent format could later used model
make change cosmetic reason depending problem want solve data might even need
include name feature example data country feature traditional database might
want move country lookup table reference ml algorithm want data instance single
row ml algorithm need numerical data process could consider turning country text
country iso code however model might interpret numerical value meaning uk iso
code value 44 would significant iso code value u 01 case splitting data multiple
column fine known categorical cod learn later course type data could convert
text value numerical value example could use zero one represent male female
numeric value used easily model remaining feature like age birth shown bm table
day week shown dow extracting age birth month day week might appropriate
depending problem trying solve age impact target variable day week born worry
sound complicated learn feature engineering later course data cleaned identified
feature want use time train model use data train model fact need hold data data
test typically use 80 data train save rest data testing train model training
data diagram model us xgboost algorithm model parameter set parameter alter
algorithm work known hyperparameters output training job trained model train
model use test data see well model performs take instance model seen use perform
prediction already know target test data compare two value comparison calculate
metric give data well model performing make change model data feature
hyperparameters find model yield best result training model real danger
overfitting underfitting model model overfitting training data see model
performs well training data perform well evaluation data model memorizing data
saw generalize unseen example model underfitting training data model performs
poorly training data model capture relationship input example often called x
target value often called understanding model fit important understanding root
cause poor model accuracy misunderstanding guide take corrective step determine
whether predictive model underfitting overfitting training data looking
prediction error training data evaluation data show step take avoid later course
retrained model satisfied result deploy model deliver best possible prediction
later course walk different phase give hand experience knowing process also
useful using managed service also explore later certain amazon ml service bulk
work key takeaway section looked machine learning pipeline process guide process
training evaluating model iterative process broken three broad step data
processing model training model evaluation video see next one
```

0.6 4. Extracting key phrases and topics

(Go to top)

[14]:

```
!pip install spacy
!python -m spacy download en_core_web_sm

import spacy
```

```python
# Load the pre-trained English language model
nlp = spacy.load('en_core_web_sm')

# Define a list to store the key phrases and topics for all texts
key_topics = []

# Loop over the normalized texts
for normalized_text in normalized_texts:
    # Process the normalized text with spaCy
    doc = nlp(normalized_text)

    # Extract noun chunks (key phrases)
    key_phrases = [chunk.text for chunk in doc.noun_chunks]

    # Extract named entities (topics)
    topics = list(set([entity.text for entity in doc.ents]))

    # Add the key phrases and topics for this text to the list
    key_topics.append((key_phrases, topics))

# Print the key phrases and topics for all texts
for i, (key_phrases, topics) in enumerate(key_topics):
    print(f"\nVideo {i+1}:")
    print("Key phrases:", key_phrases)
    print("Topics:", topics)
```

```
Collecting spacy
  Downloading
spacy-3.7.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(27 kB)
Collecting spacy-legacy<3.1.0,>=3.0.11 (from spacy)
  Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl.metadata (2.8 kB)
Collecting spacy-loggers<2.0.0,>=1.0.0 (from spacy)
  Downloading spacy_loggers-1.0.5-py3-none-any.whl.metadata (23 kB)
Collecting murmurhash<1.1.0,>=0.28.0 (from spacy)
  Downloading murmurhash-1.0.10-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_
64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.0 kB)
Collecting cymem<2.1.0,>=2.0.2 (from spacy)
  Downloading
cymem-2.0.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(8.4 kB)
Collecting preshed<3.1.0,>=3.0.2 (from spacy)
  Downloading preshed-3.0.9-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.m
anylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.2 kB)
Collecting thinc<8.3.0,>=8.2.2 (from spacy)
  Downloading
thinc-8.2.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
```

```
(15 kB)
Collecting wasabi<1.2.0,>=0.9.1 (from spacy)
  Downloading wasabi-1.1.2-py3-none-any.whl.metadata (28 kB)
Collecting srsly<3.0.0,>=2.4.3 (from spacy)
  Downloading
srsly-2.4.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(20 kB)
Collecting catalogue<2.1.0,>=2.0.6 (from spacy)
  Downloading catalogue-2.0.10-py3-none-any.whl.metadata (14 kB)
Collecting weasel<0.4.0,>=0.1.0 (from spacy)
  Downloading weasel-0.3.4-py3-none-any.whl.metadata (4.7 kB)
Collecting typer<0.10.0,>=0.3.0 (from spacy)
  Downloading typer-0.9.4-py3-none-any.whl.metadata (14 kB)
Collecting smart-open<7.0.0,>=5.2.1 (from spacy)
  Downloading smart_open-6.4.0-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from spacy)
(4.66.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from spacy)
(2.31.0)
Collecting pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 (from spacy)
  Downloading pydantic-2.6.4-py3-none-any.whl.metadata (85 kB)
                           85.1/85.1 kB
9.3 MB/s eta 0:00:00
Requirement already satisfied: jinja2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from spacy)
(3.1.3)
Requirement already satisfied: setuptools in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from spacy)
(69.1.0)
Requirement already satisfied: packaging>=20.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from spacy)
(21.3)
Collecting langcodes<4.0.0,>=3.2.0 (from spacy)
  Downloading langcodes-3.3.0-py3-none-any.whl.metadata (29 kB)
Requirement already satisfied: numpy>=1.19.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from spacy)
(1.22.4)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
packaging>=20.0->spacy) (3.1.1)
Collecting annotated-types>=0.4.0 (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy)
  Downloading annotated_types-0.6.0-py3-none-any.whl.metadata (12 kB)
Collecting pydantic-core==2.16.3 (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy)
  Downloading pydantic_core-2.16.3-cp310-cp310-manylinux_2_17_x86_64.manylinux20
```

```
14_x86_64.whl.metadata (6.5 kB)
Requirement already satisfied: typing-extensions>=4.6.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (4.9.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0.0,>=2.13.0->spacy) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0.0,>=2.13.0->spacy) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0.0,>=2.13.0->spacy) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0.0,>=2.13.0->spacy) (2024.2.2)
Collecting blis<0.8.0,>=0.7.8 (from thinc<8.3.0,>=8.2.2->spacy)
  Downloading
blis-0.7.11-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(7.4 kB)
Collecting confection<1.0.0,>=0.0.1 (from thinc<8.3.0,>=8.2.2->spacy)
  Downloading confection-0.1.4-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: click<9.0.0,>=7.1.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
typer<0.10.0,>=0.3.0->spacy) (8.1.7)
Collecting cloudpathlib<0.17.0,>=0.7.0 (from weasel<0.4.0,>=0.1.0->spacy)
  Downloading cloudpathlib-0.16.0-py3-none-any.whl.metadata (14 kB)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
jinja2->spacy) (2.1.5)
Downloading
spacy-3.7.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.6 MB)
                         6.6/6.6 MB
111.0 MB/s eta 0:00:0000:01
Downloading catalogue-2.0.10-py3-none-any.whl (17 kB)
Downloading
cymem-2.0.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (46 kB)
                         46.1/46.1 kB
5.0 MB/s eta 0:00:00
Downloading langcodes-3.3.0-py3-none-any.whl (181 kB)
                         181.6/181.6 kB
31.1 MB/s eta 0:00:00
Downloading murmurhash-1.0.10-cp310-cp310-manylinux_2_5_x86_64.manylinux1_
x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (29 kB)
Downloading preshed-3.0.9-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.man
ylinux_2_17_x86_64.manylinux2014_x86_64.whl (156 kB)
                         156.9/156.9 kB
29.4 MB/s eta 0:00:00
```

```
Downloading pydantic-2.6.4-py3-none-any.whl (394 kB)
                         394.9/394.9 kB
53.2 MB/s eta 0:00:00
Downloading
pydantic_core-2.16.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(2.2 MB)
                         2.2/2.2 MB
92.9 MB/s eta 0:00:00
Downloading smart_open-6.4.0-py3-none-any.whl (57 kB)
                         57.0/57.0 kB
7.0 MB/s eta 0:00:00
Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl (29 kB)
Downloading spacy_loggers-1.0.5-py3-none-any.whl (22 kB)
Downloading
srsly-2.4.8-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (493 kB)
                         493.0/493.0 kB
12.0 MB/s eta 0:00:0000:01
Downloading
thinc-8.2.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (922 kB)
                         922.3/922.3 kB
67.1 MB/s eta 0:00:00
Downloading typer-0.9.4-py3-none-any.whl (45 kB)
                         46.0/46.0 kB
3.8 MB/s eta 0:00:00
Downloading wasabi-1.1.2-py3-none-any.whl (27 kB)
Downloading weasel-0.3.4-py3-none-any.whl (50 kB)
                         50.1/50.1 kB
6.5 MB/s eta 0:00:00
Downloading annotated_types-0.6.0-py3-none-any.whl (12 kB)
Downloading
blis-0.7.11-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (10.2 MB)
                         10.2/10.2 MB
103.1 MB/s eta 0:00:0000:010:01
Downloading cloudpathlib-0.16.0-py3-none-any.whl (45 kB)
                         45.0/45.0 kB
4.1 MB/s eta 0:00:00
Downloading confection-0.1.4-py3-none-any.whl (35 kB)
Installing collected packages: cymem, wasabi, typer, spacy-loggers, spacy-
legacy, smart-open, pydantic-core, murmurhash, langcodes, cloudpathlib,
catalogue, blis, annotated-types, srsly, pydantic, preshed, confection, weasel,
thinc, spacy
Successfully installed annotated-types-0.6.0 blis-0.7.11 catalogue-2.0.10
cloudpathlib-0.16.0 confection-0.1.4 cymem-2.0.8 langcodes-3.3.0
murmurhash-1.0.10 preshed-3.0.9 pydantic-2.6.4 pydantic-core-2.16.3 smart-
open-6.4.0 spacy-3.7.4 spacy-legacy-3.0.12 spacy-loggers-1.0.5 srsly-2.4.8
thinc-8.2.3 typer-0.9.4 wasabi-1.1.2 weasel-0.3.4
Collecting en-core-web-sm==3.7.1
  Downloading https://github.com/explosion/spacy-
```

```
models/releases/download/en_core_web_sm-3.7.1/en_core_web_sm-3.7.1-py3-none-
any.whl (12.8 MB)
                           12.8/12.8 MB
108.7 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: spacy<3.8.0,>=3.7.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from en-
core-web-sm==3.7.1) (3.7.4)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.0.9)
Requirement already satisfied: thinc<8.3.0,>=8.2.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (8.2.3)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.0.10)
Requirement already satisfied: weasel<0.4.0,>=0.1.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.3.4)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.9.4)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (6.4.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (4.66.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.6.4)
Requirement already satisfied: jinja2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.1.3)
Requirement already satisfied: setuptools in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (69.1.0)
Requirement already satisfied: packaging>=20.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (21.3)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.3.0)
Requirement already satisfied: numpy>=1.19.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (1.22.4)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
packaging>=20.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.1.1)
Requirement already satisfied: annotated-types>=0.4.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.2->en-core-web-
sm==3.7.1) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.3 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.2->en-core-web-
sm==3.7.1) (2.16.3)
Requirement already satisfied: typing-extensions>=4.6.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.2->en-core-web-
sm==3.7.1) (4.9.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
requests<3.0.0,>=2.13.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2024.2.2)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
thinc<8.3.0,>=8.2.2->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
thinc<8.3.0,>=8.2.2->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.1.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
typer<0.10.0,>=0.3.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (8.1.7)
Requirement already satisfied: cloudpathlib<0.17.0,>=0.7.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
weasel<0.4.0,>=0.1.0->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from
jinja2->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.1.5)
Installing collected packages: en-core-web-sm
Successfully installed en-core-web-sm-3.7.1
  Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')

Video 1:
Key phrases: ['section', 'going cover evaluate data section', 'different data
format type', 'visualize analyze data feature engineering', 'statistic data',
'working need', 'right format analysis amazon sagemaker algorithm', 'training
data csv format', 'many tool use explore visualize analyze data', 'csv format',
'knowledge problem', 'machine', 'learning example developing model', 'set
symptom', 'disease', 'know relationship symptom disease data', 'numeric form
machine learning algorithm', 'data', 'prediction look', 'text data', 'next
section explore data', 'insight', 'overall data', 'one popular open source',
'python library panda', 'data', 'various format reformat load', 'representation
data', 'row column format panda reformat load', 'csv excel pickle javascript
object notation json panda', 'analysis manipulation feature use', 'module
loading data', 'simple example', 'csv file specified url load data panda',
'panda dataframe panda documentation data frame', 'general 2d', 'labeled size
mutable tabular structure', 'potentially heterogeneously typed column', 'data
frame', 'spreadsheet sql table', 'table', 'data frame row', 'attribute shape
property data frame', 'number row column column data frame series series', 'one
dimensional labeled array series store data type learn data structure panda',
'panda documentation', 'data load data frame row label column label row label
known index column label known column', 'loaded data csv file hetero column',
'first line file change behavior', 'data analysis', 'correct data', 'many case',
'panda', 'correct data type load data', 'data type issue', 'either type info
function', 'information column type', 'example correct data type need figure
case', 'numeric column', 'data', 'single text value example car data', 'number
door 23 five analyzed data convert column correct data type', 'panda part', 'one
section', 'part 2 review', 'data']
Topics: ['2d', 'one', 'first', 'csv', '2', '23 five', 'json panda', '3']

Video 2:
Key phrases: ['introduce machine', 'first look business problem', 'machine',
'well talk terminology process tool challenge face', 'module able recognize
machine', 'part artificial intelligence', 'artificial intelligence machine',
'learning terminology', 'machine learning', 'solve business problem', 'machine
learning process list tool available data scientist', 'traditional software
development method', 'section', 'next video']
Topics: ['first', '2', '1']

Video 3:
Key phrases: ['section', 'image analysis detail part', 'look video analysis',
'main amazon service', 'amazon recognition amazon recognition computer vision
service', 'deep learning use', 'image video analysis application', 'many us
amazon recognition', 'searchable image video library amazon recognition',
'image', 'amazon recognition build face based user verification system
application', 'user identity', 'live image reference image amazon recognition',
'emotional expression', 'happy sad surprise', 'demographic information facial
image gender amazon recognition', 'inappropriate content image stored video',
'finally amazon recognition recognize extract text content image', 'quick note
security need check application build', 'amazon recognition', 'regulatory
restriction defined field country security compliance amazon recognition',
'shared responsibility aws customer information topic', 'aws compliance page
amazon recognition aws managed service managed service amazon host machine
learning model', 'api scale meet demand benefit set model', 'improve', 'building
application use api optionally training service', 'unique business', 'various
resource use access interact amazon recognition apis sdks command aws command
line interface', 'aws cli language', 'sdk', 'javascript python php dotnet jav',
'j', 'storage use amazon simple storage service s3 authentication authorization
use aws identity access management diagram', 'image search feature user',
'picture', 'information real estate property viewing user', 'picture mobile
device user initiate search cause application upload amazon s3 s3', 'call
service right', 'case bucket', 'pass s3 path new object aws', 'lambda lambda
function', 'us', 'amazon recognition sdk', 'service amazon recognition',
'image', 'aspect property', 'label pass information', 'lambda object formatted
javascript object notation json lambda store label confidence score amazon
elastic service', 'amazon e application user', 'aspect property', 'object',
'image example architecture system', 'uploaded image inappropriate content',
'previous example processing', 'user uploads content user uploads image amazon
s3 second s3 bucket', 'amazon recognition', 'sdk amazon recognition', 'image',
'inappropriate content', 'response', 'lambda content appropriate content',
'approved content appropriate content', 'manual inspection', 'finally content
approved notification', 'user final use case system', 'video feed sentiment
analysis store camera capture video', 'office cloud based application',
'typically application', 'us amazon kinesis stream video second application',
'us', 'sdk', 'video amazon recognition analysis visual sentiment', 'attribute
age discovered attribute', 'amazon kinesis lambda function', 'extract', 'data
stream data', 'written s3', 'regular basis', 'amazon', 'generate report data
amazon recognition', 'integrate application api sdks api operation', 'label
face', 'celebrity', 'unsafe image perform prediction', 'provide service image
object amazon s3 upload bite stream image image jpeg png format amazon
```

```
recognition process image', 'amazon recognition', 'prediction', 'multiple label
label confidence level confidence level', 'likely label', 'example show label',
'instance object need understand', 'object image instance', 'amazon
recognition', 'bounding box', 'coordinate top left box dimension height',
'example', 'information determine location', 'object image important note',
'confidence score', 'higher score', 'part 1 section', 'part', 'facial
detection']
Topics: ['second', 'recognition', '2', '1']

Video 4:
Key phrases: ['time review module wrap summary module learn', 'nlp use case',
'managed amazon', 'ml service describe', 'available nlp good job', 'next
module']
Topics: []

Video 5:
Key phrases: ['data collection', 'secure data', 'security data', 'data', 'secure
use aws identity access management', 'iam service control access resource',
'securing data', 'aws', 'data breach diagram', 'simple iam policy', 'access',
'specific s3 bucket', 'listed role addition', 'access data', 'data', 'good
practice', 'certain data type financial data healthcare record aws', 'encryption
feature storage service', 'data rest transit', 'encryption requirement',
'encryption object service', 'protect data transit', 'secure transport', 'tl',
'another aspect', 'compliance audit', 'data', 'regulated industry', 'audit
access data aws cloud trail service', 'governance compliance operational
auditing risk auditing aws', 'retain account activity related action', 'entire
aws infrastructure cloudtrail', 'event history aws', 'account activity', 'action
taken aws management console aws', 'sdk command line tool aws service event
history', 'security analysis resource change tracking troubleshooting',
'cloudtrail', 'unusual activity aws account feature help', 'operational
analysis', 'troubleshoot key takeaway section', 'machine learning problem',
'data', 'etl', 'obtain data multiple source service', 'aws glue', 'easy obtain
data multiple data store', 'security requirement', 'based business', 'regulatory
requirement', 'data', 'possible section', 'next video']
Topics: ['section 2', 'simple iam', 'first']

Video 6:
Key phrases: ['back aws academy machine learning module', '5 great topic',
'computer vision space', 'use case terminology', 'explore detail', 'image video
managed service amazon web service aws', 'customized data', 'object detection',
'end module', 'able describe use case computer vision', 'amazon management
learning service available image video analysis list step', 'custom data',
'object detection', 'amazon sagemaker ground truth', 'custom data', 'amazon
recognition', 'facial detection thanks', 'next video']
Topics: ['5', 'today']

Video 7:
Key phrases: ['section', 'custom data', 'computer vision', 'custom object one
```

```
challenge', 'pre built model', 'image', 'trained find', 'amazon recognition',
'ten million image detect object trained example', 'aid heart', 'card run car
amazon recognition result', 'various attribute', 'none', 'playing card', 'eight
heart', 'amazon recognition', 'image problem domain', 'model image section',
'train amazon recognition image problem domain focus', 'amazon recognition',
'similar process', 'pre trained model training computer vision', 'algorithm',
'image', 'large input data', 'practical organization', 'many machine learning
problem', 'existing model use managed service', 'amazon recognition custom
label', 'machine learning process', 'train amazon recognition', 'scene',
'specific domain', 'training data', 'test data', 'contain', 'image image', 'need
label use amazon recognition custom label', 'labeling task example', 'labeling
image', 'feature use draw bounding box', 'image', 'object scene unique business
need use classify image', 'object', 'image', 'torque converter', 'picture kind
machine use train model amazon recognition custom label', 'automated machine
learning capability handle machine learning process', 'training image service',
'inspect data', 'correct machine learning algorithm train model', 'model
performance metric finish training model', 'image test', 'side side comparison
model prediction', 'label', 'also detailed performance metric review', 'model',
'analysis', 'iterate retrain new version image refine model', 'model track
prediction', 'correct mistake use feedback data retrain new model version',
'performance label image diagram', 'typical process', 'training computer vision
model', 'amazon recognition custom label feature step detail process', 'custom
model analyze image', 'time expertise resource', 'month', 'thousand ten thousand
hand labeled image model', 'enough data', 'accurate decision', 'month generate
gather data', 'large team laborer', 'use machine', 'amazon recognition custom
label', 'existing capability amazon recognition', 'ten million image', 'many
category', 'instead thousand image upload small set training image specific use
case', 'hundred image use aws management console upload training image image',
'amazon recognition custom label', 'training model label image', 'shortly amazon
recognition custom label work', 'different model', 'machine part plant health',
'similar image', 'used inference use image', 'various lighting condition
background resolution', 'image mirror image', 'detection use source', 'use
production work best documentation', 'additional guideline image type weather
jpegs pngs property', 'image size resolution part 1 section', 'part 2 review',
'training data']
Topics: ['eight', 'one', 'today', '1', 'thousand', 'ten million', 'hundred',
'2', 'two', 'ten thousand', 'month']

Video 8:
Key phrases: ['section', 'machine learning', 'digital life email spam filter
result machine learning program', 'trained example spam regular email message
based book reading product', 'machine learning program', 'book product likely
interested machine learning program trained data reader habit purchase detecting
credit card fra machine learning program trained example transaction', 'fraud',
'normal transaction', 'many example social medium application', 'facial
detection group photo detecting brain tumor brain', 'scan', 'anomaly x ray',
'three main type machine', 'supervised learning model', 'us known input output',
'future output', 'input output', 'pattern data', 'help', 'reinforcement learning
```

```
model', 'action', 'reward important know different type', 'type guide',
'algorithm', 'sense', 'business problem', 'popular type', 'widely applicable
called supervised learning need supervisor teacher', 'student', 'supervised
algorithm', 'learn example', 'us', 'pattern relationship input output', 'build
application', 'credit card fraud', 'need training data', 'example fraud example
normal transaction', 'different type problem classification regression two
subtypes classification problem', 'back example', 'fraudulent transaction target
variable example', 'two option fraudulent fraudulent binary classification
problem', 'class classification problem', 'ml problem', 'observation', 'one
three category', 'ml', 'model', 'customer', 'store', 'number transfer needed
customer', 'correct customer support department case', 'different customer
support department', 'variety potential target variable', 'many different
department', 'problem regression problem', 'input defined number category',
'input continuous value', 'integer one example', 'regression problem', 'price
company stock computer vision', 'good example', 'tumor x ray computer vision',
'deep learning model', 'extraction analysis classification', 'useful information
single image sequence image computer vision', 'machine', 'people', 'thing image
accuracy', 'human level greater speed efficiency image data', 'many form search
single image video sequence', 'multiple camera three dimensional data',
'computer vision', 'data supervisor room', 'learning label', 'supervised
learning', 'variable pattern instance machine uncover', 'label model use date',
'emerging property entire data', 'construct pattern property', 'similar
feature', 'attribute specific cluster example', 'customer purchasing habit',
'group customer associated size tear company advantage', 'algorithm enable',
'pattern data', 'aware natural language processing', 'growth', 'alexa voice
value', 'answer question nlp speech nlp', 'many application example nlp',
'center bot automated system', 'bank balance order food restaurant', 'text
language example', 'application translate menu', 'voice text translation convert
spoken word text', 'used sentiment analysis use analyze sentiment comment review
product music movie sentiment', 'movie audience rating', 'popularity', 'recently
reinforcement learning', 'reinforcement learning', 'model mining feedback
previous iteration reinforcement learning agent', 'trial error', 'environment
reinforcement', 'known path', 'annette path', 'lot trial error discover',
'example aws deepracer aws deep racer simulator agent virtual car environment',
'virtual race track action throttle', 'input car goal completing racetrack',
'track car need', 'behavior', 'goal completing track car', 'aws deepracer team',
'reward incentivize model', 'behavior reinforcement learning thing', 'agent
case', 'deep racer car environment', 'race track agent something environment',
'response', 'boundary cross', 'action response', 'reward penalty', 'agent
something', 'action', 'reward', 'fewer penalty', 'many machine', 'algorithm
model solve problem', 'point point b two main task continuous detection
environment forecasting change', 'object localizing predicting movement',
'object output', 'act input system', 'decision vehicle various control use case
self driving vehicle', 'previously hidden pedestrian walk', 'obstacle vehicle
break need', 'latency room error action', 'every problem', 'machine learning',
'sometimes regular programming work', 'potential machine learning solution',
'look existence', 'large data', 'often best choice uncertain business logic
procedure', 'obtain answer accomplish task machine learning system complex
```

```
supporting infrastructure management support technical expertise', 'place',
'project success key takeaway section', 'machine learning application', 'three
category', 'learning training data', 'answer', 'insight', 'data reinforcement
learning model', 'based experience feedback business problem', 'learning problem
section', 'next video']
Topics: ['anomaly x ray', 'one', 'much two', 'first', 'two', 'boundary cross',
'three']

Video 9:
Key phrases: ['welcome module', '7 course wrap congratulation completing aws
academy machine learning course', 'minute review', 'start review', 'course',
'implement machine learning pipeline', 'service forecasting', 'computer vision',
'natural language processing', 'course', 'prepare', 'certified aws certified
machine learning specialty review', 'work', 'certification aws certification
help', 'credibility confidence', 'cloud expertise industry', 'credential',
'organization', 'skilled professional lead cloud initiative', 'aws', 'passing
score', 'passing score', 'certification credential aws certification publish
list service feature covered certification exam', 'exam guide exam list current
topic area objective covered exam exam guide', 'prepare aws certification exam
webpage required update certification recertify', 'every 3 year view aws
certification recertification page detail information slide', 'exam', 'detail',
'exam available topic', 'exam subject change aws certified machine learning
specialty mean', 'select', 'appropriate machine learning approach', 'business
problem', 'appropriate aws service implement machine learning solution',
'scalable cost', 'reliable secure machine learning solution', 'specialty exam
recommend', 'knowledge experience', 'one two year experience developing
architecting', 'basic hyperparameter optimization working machine', 'deep
learning framework', 'also able express intuition', 'basic ml algorithm',
'finally able follow best practice model training addition best practice
deployment operation thanks', 'congratulation completing aws academy machine
learning course']
Topics: ['june 2020', 'two year', '7', 'every 3 year']

Video 10:
Key phrases: ['section look tool', 'rest course', 'exhaustive list tool', 'high
level good place', 'jupyter notebook jupyter notebook open source web
application use', 'share document', 'live code equation visualization narrative
text', 'us', 'transformation numerical simulation statistical modeling data
visualization machine', 'configure arrange user interface support wide range
workflow data science scientific computing machine', 'jupiter lab', 'extensible
modular write plugins', 'new component', 'lab panda open source', 'data',
'analysis panda', 'data table similar spreadsheet table known panda dataframe
matplotlib python library', 'scientific static', 'generate plot data', 'another
data visualization library python', 'high level interface', 'attractive
informative statistical graph numpy one fundamental scientific computing package
python', 'function', 'dimensional ray object', 'random number capability', 'open
source machine learning library support', 'unsupervised learning', 'various tool
model fitting data', 'model selection evaluation', 'many utility sci fi
```

```
matplotlib good tool', 'machine learning', 'use', 'function course', 'complete
moving individual library package', 'production ready framework', 'sidekick',
'good library machine learning framework', 'kera', 'library use machine learning
framework', 'supported aws', 'amazon sagemaker aws', 'machine learning cloud
edge compute instance', 'inference', 'another aws resource', 'certain amazon
machine image amiss offer', 'many popular framework', 'finally amazon sagemaker
aws', 'many capability sagemaker', 'machine learning instance', 'running jupiter
notebook jupiter lab', 'deployment compute resource need connect jupiter
environment sagemaker', 'tool labeling data training model', 'hosting train
model aws marketplace', 'selection ready use model package', 'algorithm third
party machine', 'learning developer aws', 'set management machine learning
service integrate application even substantial machine learning experience
computer vision amazon recognition', 'object facial recognition image video',
'also amazon textract', 'text image speech service', 'amazon', 'text', 'another
speech service amazon transcribe convert spoken audio language amazon', 'us',
'insight relationship text', 'text', 'different language', 'work', 'amazon alexa
help', 'interactive conversational application use voice text forecasting
amazon', 'combine time series data', 'additional variable build forecast', 'work
recommendation amazon', 'individual personalized recommendation customer managed
service', 'many aspect problem domain', 'specific data', 'look many managed
service', 'thing key takeaway section', 'machine', 'task jupiter notebook', 'web
based hosted development environment machine learning use jupiter notebook',
'large number open source tool pand', 'often machine learning practitioner',
'requirement', 'low level framework create solution', 'tool amazon sagemaker',
'heavy lifting', 'adapt one managed amazon', 'service specific problem domain
video', 'see']
Topics: ['one', 'today', 'tensorflow kera', 'half', 'jupiter', 'linear',
'third', 'task jupiter', 'sci', 'second', 'recognition']

Video 11:
Key phrases: ['evaluate model diagram show confusion matrix', 'two different
model', 'one better better good question', 'cat', 'sure model', 'two chart',
'several model', 'multiple fold hundred data point compare', 'metric first
metric sensitivity', 'recall', 'percentage cat', 'calculate sensitivity', 'take
number true positive number positive identification cat', 'total number actual
cat', '60 cat cat', 'cat specificity', 'correctly identified cat example number
image cat', 'cat calculate specificity', 'number', 'true negative divide total
number actual negative example number cat', 'divided total number actual cat
mean', '64 cat', 'cat metric model', 'business goal', 'easier decide model use
model', 'many cat possible model b', 'many false positive concerned incorrectly
identified cat model', 'sure identified animal cat model', 'scenario', 'many
false negative tolerate classification patient heart disease model',
'interesting fun website', 'bad reputation', 'cat', 'diagnose patience focus',
'probably different important understand trade offs', 'decide model use',
'metric help', 'decision part 2 section', 'part']
Topics: ['one', '60', '2', 'two', '64', '3']

Video 12:
```

```
Key phrases: ['natural language processing mean natural language processing',
'also known nlp', 'nlp', 'example', 'amazon server', 'individual sound',
'database', 'pronunciation', 'define word', 'closely correspond combination
individual sound amazon', 'important word', 'sense', 'task', 'corresponding
function instance alexa notice word', 'outside temperature open weather alexa
skill amazon server', 'information', 'device alexa', 'mlp broad term general set
business computational problem', 'pretty smartphone cell phone', 'nlp screen
reader', 'many nlp system use form machine learning mlp', 'hierarchical
structure language word lowest layer hierarchy group word', 'phrase', 'next
level phrase', 'sentence', 'convey idea nlp system', 'several significant
challenge look challenge', 'based word', 'known context', 'multiple meaning
example', 'term weather', 'colloquial meaning', 'wonderful weather', 'mean
weather condition', 'good phrase', 'surprise disagreement', 'many thing',
'context inflection main challenge', 'one challenge', 'structure text', 'one
first task nlp application break text meaningful unit word phrase', 'another
challenge labeling data system', 'label', 'various part speech', 'every
language', 'different labeling scheme match language grammar nlp', 'challenge',
'context word', 'heavily context nlp system need way', 'large challenge', 'many
contact difficult convert contact form computer', 'grammar', 'structure language
application grammar', 'indescribably large scope handling variation language',
'human major challenge mlp system machine', 'large impact', 'nlp range problem
common application', 'human machine interaction', 'alexa sentiment analysis
marketing political campaign social research based medium analysis chatbots',
'human speech application', 'machine learning development pipeline', 'course',
'mlp solution first task formulate problem', 'label data nlp collecting data',
'text', 'meaningful subset', 'highly irregular unstructured text example',
'building application classify document', 'able distinguish word common term',
'labeling data nlp domain', 'tagging labeling process', 'individual text
string', 'different part speech specialized tool', 'first task mlp application
convert text data', 'convert', 'word', 'needed analysis input text example
word', 'phrase sample text', 'stop', 'word', 'text', 'similar word', 'common
form example word run', 'different form word run normalize instance word block
text', 'stemming limitation process limitation group different form word single
term limitation version word run', 'group instance', 'hand remove character',
'stemming algorithm', 'unnecessary standing', 'form word run', 'word', 'analysis
example', 'acronym slang special character natural language toolkit', 'nltk
python library', 'function', 'stop word normalizing text', 'another first step',
'mlp system', 'text data collection data frame mlp library', 'word', 'function
nltk library', 'text loaded data frame', 'one nlp model create feature couple
common model first model known bag word simple model capturing frequency word
document model', 'key word value', 'key number time word', 'document second
model term frequency inverse document frequency', 'idf term frequency', 'word',
'document inverse document frequency number time word', 'group document two
value', 'calculate weight word word', 'many document lower weight', 'bag word
model bag word vector model vector model convert sentence phrase', 'vector
mathematical object record directionality', 'magnitude', 'simple sentence',
'converted vector frequency word', 'recorded word value', 'twice sentence bag
word', 'classify document', 'different category', 'derive attribute feed nlp
```

```
application sentiment analysis', 'three broad category text analysis
classification text similar classification system', 'course text', 'input
process extract feature', 'feature machine learning algorithm', 'classifier
model infers', 'many application text matching example autocorrect', 'spelling
grammar', 'based text matching algorithm edit', 'also known levinstein',
'frequently used drive relationship', 'different word phrase text', 'process',
'core reference resolution', 'several nlp system', 'python library deriving
relationship', 'one biggest challenge nlp describe context text', 'example user
searching term tablet word', 'least two distinct meaning search engine', 'user
mind search engine', 'commonly used context term qualified example', 'another
term', 'medicine computing search process', 'entity', 'entity recognition
armor', 'function', 'noun phrase', 'dependency chart part speech tagging
classify phrase', 'classification algorithm word', 'vec', 'entity', 'knowledge
graph example', 'ner extract entity titanic north atlantic text', 'entity', 'use
knowledge graph extract', 'graph', 'amazon recommendation engine', 'example
knowledge', 'graph main point', 'section nlp', 'machine learning use', 'workflow
seen module', 'main use case', 'nlp search query analysis', 'human machine
interaction marketing social research', 'next video']
Topics: ['one', 'first', '2', 'google bing', 'grammar', 'north atlantic', 'two',
'noun', 'english', 'second', 'three', 'vec']

Video 13:
Key phrases: ['image analysis', 'facial detection facial detection', 'us model',
'perform prediction', 'face facial feature facial detection', 'standard object
detection bounding box coordinate box surrounding face', 'value', 'confidence
bounding box', 'face list attribute', 'face beard', 'male female also confidence
score attribute', 'physical emotion', 'person', 'important understand
classification based visual clue', 'eye mouth', 'coordinate quality',
'brightness sharpness face pose', 'rotation face', 'image confidence feature',
'detected feature', 'feature prediction based visual observation amazon
recognition', 'two image determine contain person comparison', 'source target
image result', 'face', 'information matching non matching face confidence
score', 'likely prediction amazon recognition', 'known face use feature need
train model', 'collection image use train model', 'people image provide', 'known
face', 'collection', 'face collection amazon recognition', 'facial recognition
image', 'typical information', 'bounding box coordinate confidence score
associate face image specify image id external image id request parameter',
'name image', 'collection use search face image operation search face
collection', 'array face matched information', 'box confidence score external
image', 'id value', 'id value', 'link back source image', 'facial detection',
'human face capture bounding box show face', 'video', 'attribute position eye
nose mouth', 'emot quality detection landmark', 'item', 'higher score mean model
greater confidence detection gender inferred image inferred identity similarly
emotion', 'image', 'subject actual emotional state', 'facial recognition',
'responsibly facial recognition', 'individual right', 'right privacy',
'autonomous decision scenario', 'bank us tool', 'amazon recognition', 'financial
application', 'customer identity bank', 'use technology', 'customer approve term
condition information topic', 'aws web page fact', 'facial recognition
```

```
artificial intelligence', 'use amazon recognition process video', 'stored video
uploaded stored s3 bucket type detection', 'operation search people', 'label
celebrity text', 'inappropriate content amazon recognition', 'completion status
topic', 'simple notification service', 'amazon sn route message subscriber
durability best practice route message message queue amazon simple queue service
amazon sqs application monitor sqsq completion', 'start operation
corresponding', 'operation', 'result call', 'detection result return array label
contain information label', 'video label information', 'label image detection',
'timestamp label', 'millisecond start video addition', 'amazon recognition video
detect recognize', 'face streaming video typical use case', 'known face video
stream amazon recognition video', 'us amazon kinesis video stream', 'receive
process video stream analysis result output amazon recognition video kinesis
datastream read client application amazon recognition video', 'stream
processor', 'create stream processor use', 'analysis', 'streaming video use
amazon recognition video streaming video application', 'resource', 'need kinesis
video stream', 'streaming video amazon recognition video', 'amazon recognition
video stream processor', 'streaming video analysis', 'kinesis datastream
consumer read analysis', 'amazon recognition video', 'face', 'video need',
'collection process', 'collection', 'amazon recognition video place json frame
record analyzed frame kinesis output stream amazon recognition video analyze',
'every frame', 'kinesis video stream frame record', 'information video stream
fragment frame frame fragment face recognized frame', 'status information stream
processor wrap quick summary amazon recognition computer vision service based
deep learning', 'image video analysis application amazon recognition', 'face
sentiment text', 'unsafe content library search image video amazon recognition',
'integrated aws service thanks', 'next video']
Topics: ['two', 'first', 'recognition']

Video 14:
Key phrases: ['section', '4 section', 'feature engineering feature', 'one
impactful thing', 'machine learning model', 'model successful first feature
selection second feature extraction process', 'feature feature selection',
'relevant feature discard rest', 'feature selection prevent redundancy
irrelevance', 'existing feature', 'limit number feature help', 'overfitting
feature extraction', 'valuable information raw data', 'primary feature', 'new
one process', 'new data', 'consumed model', 'goal diagram show feature
extraction cover range activity', 'missing data', 'text data numerical data',
'idea data handling', 'data useful state', 'many task', 'different job working
data', 'sure data correct format', 'example', 'data', 'data multiple column',
'column', 'need convert text column numerical value', 'decide handle outlier
potentially rescale data', 'common task section machine learning algorithm',
'best numerical data', 'column data', 'numeric data', 'converting encoding',
'several pass datasheet encode example', 'variability text value', 'contain
medium med value categorical data', 'numerical value', 'ordinal relationship',
'data', 'maintenance cost', 'categorical data uniform', 'tool', 'skykit', 'panda
encode data categorical data order', 'ordinal relationship data example',
'assigned value', '1 first color red', 'next value', 'blue model', 'blue
important red blue higher numeric value', 'non ordinal data multiple column',
'new feature', 'checkbox', 'example', 'three feature generated volume one',
'color section', 'next video']
Topics: ['one', '1', 'first', 'three high four', 'section 4', 'two', 'second',
'three']

Video 15:
Key phrases: ['time review module main takeaway module', 'landscape', 'type
problem machine learning', 'u', 'machine learning', 'algorithm', 'machine
learning pipeline different stage developing machine learning application',
'challenge machine', 'summary module', 'machine', 'part artificial
intelligence', 'artificial intelligence machine learning terminology', 'machine
learning', 'solve business problem', 'machine learning process list tool
available data scientist', 'use machine learning instead traditional software
development method', 'next video']
Topics: []

Video 16:
Key phrases: ['module 3 section 8 section', 'tune model hyperparameters', 'model
performance recall', 'earlier module hyperparameters', 'algorithm',
'performance', 'specifically different type hyperparameters', 'hyperparameter
optimization couple different category hyperparameters', 'first kind model
hyperparameters', 'model example', 'neural network computer vision problem case
additional attribute architecture need', 'filter size', 'stride padding second
kind optimizer hyperparameters', 'model learns pattern based data', 'neural
network model type hyperparameters', 'optimizers', 'gradient desu stochastic
gradient descent', 'optimizers', 'momentum', 'atom initialize parameter weight
method', 'xavier initialization initialization', 'third kind data
hyperparameters', 'attribute data', 'attribute', 'different data augmentation
technique', 'cropping resizing image related problem', 'enough data', 'enough
variation data', 'hyperparameters labor', 'someone domain experience related
hyperparameter use case person', 'hyperparameters based intuition experience',
'model score validation data process', 'achieved satisfactory result manual
process', 'efficient way', 'hyperparameters sagemaker', 'automated
hyperparameter', 'amazon sagemaker', 'automatic model tuning', 'best version
model', 'multiple training job data', 'algorithm hyperparameter range specify',
'hyperparameter value result model', 'us gaussian process regression',
'hyperparameter value', 'improving fit', 'hyperparameter space', 'specific
hyperparameter value', 'built algorithm sagemaker', 'pre', 'algorithm
container', 'binary classification problem fraud data', 'goal maximize area auc
curve metric algorithm', 'linear learner algorithm model know value learning
rate', 'beta one beta 2 epoxy use train best model', 'best value
hyperparameters', 'combination value', 'training job', 'example sagemaker
hyperparameter', 'return training job highest auc hyperparameter tuning', 'model
advanced tool building machine solution', 'part scientific method process',
'complex machine learning system', 'deep learning neural network', 'possible
combination', 'optimization use', 'guideline', 'hyperparameters',
'hyperparameters', 'number hyperparameters', 'good result range value
hyperparameters', 'search', 'success hyperparameter optimization', 'every
```

```
possible value hyperparameter', 'search small range value', 'best metric value',
'part range', 'range part hyperparameter', 'sagemaker attempt figure
hyperparameters', 'hyperparameters', 'time sagemaker discover know
hyperparameter long scaled convert', 'hyperparameter optimization',
'hyperparameter', 'job', 'work', 'job', 'successive round experiment', 'one
training job time', 'best result', 'training job', 'multiple instance case
hyperparameter', 'us', 'metric training job design', 'training job report',
'objective metric want', 'machine learning model', 'amazon sagemaker autopilot
service', 'good model', 'little effort input part autopilot', 'job supply test
training target autopilot analyze data', 'appropriate feature train tune model
document metric find best model', 'provided data result', 'model metric jupyter
notebook', 'investigate result', 'autopilot remove', 'pre process data', 'time
feature selection model', 'key takeaway section module', 'point model', 'best
solution', 'model optimizer data sagemaker', 'automatic hyperparameter',
'finally overall model development', 'autopilot video']
Topics: ['one', 'section 8', 'third', 'first', 'linear', '2', 'second', '3']

Video 17:
Key phrases: ['data data', 'descriptive statistic data', 'descriptive statistic
help', 'valuable insight data', 'process data', 'ml model', 'important
descriptive statistic', 'organized different category overall statistic',
'number row number column data', 'information', 'dimension data', 'important
example', 'high dimensionality poor model performance attribute', 'another type
descriptive statist', 'specifically numeric attribute', 'better sense shape
attribute', 'property', 'var', 'look relationship', 'one variable',
'multivariate statistic', 'correlation relationship attribute case', 'multiple
variable feature', 'correlation', 'important identify correlation attribute high
correlation two attribute', 'poor model performance feature', 'closely
correlated used model', 'response variable', 'example model loss', 'minimum
state', 'aware highly correlated feature data', 'mean median', 'two different
measure', 'extent data', 'value position', 'useful method', 'data data',
'outlier median', 'better metric understanding data', 'central tendency
instance', 'large value', 'accurate representation value', 'available viewed
numerical data', 'method', 'calculate mean median others', 'even group data
specific value categorical attribute look frequency attribute value data',
'idea', 'categorical variable diagram show car data', 'several categorical value
lug bo safety class safety', 'either low medium high describe function', 'three
unique value low frequent looking class column', 'top value', 'unaccounted
account', '1 728 value', 'imbalance target variable also categorical type look
class distribution', 'class imbalance data', 'imbalance data', 'mark
disproportionate ratio class instance data', 'credit card transaction', 'tenth
percent', 'fraud case', 'algorithm', 'example credit card fraud visualization',
'insight data', 'often good visualization technique', 'overall behavior
particular feature histogram answer question', 'feature data', 'many peak data
skewness particular feature', 'histogram data visualization value', 'taller peak
histogram', 'common value numerical feature', 'density plot box plot addition
histogram', 'idea', 'particular feature', 'histogram visualization help',
'question', 'range data peak data special feature answering question box plot
```

```
method', 'group numerical data', 'relationship scatter plot', 'special
relationship', 'variable case', 'diagram sulfate alcohol', 'two numerical
variable', 'show relationship variable', 'plot', 'correlation', 'high data',
'relatively positive relationship', 'relationship', 'multiple different feature
panda', 'scatter plot matrix based column', 'three column', 'pairwise scatter
plot', 'two column scatter plot', 'special region particular subset data fit
example relationship alcohol sulfate quality plot value good poor quality wine',
'example', 'idea', 'useful particular variable', 'classification problem',
'part', 'part 3 review correlation takeaway section']
Topics: ['one', '1', 'two', '70', 'four', 'three', '3']

Video 18:
Key phrases: ['video analysis', 'evaluate', 'improve model', 'quality model
larger quantity', 'better quality data', 'training image', 'object', 'many thing
interested bounding box', 'fully visible hidden object', 'training test data',
'match type image', 'france object training example', 'logo', 'bounding box',
'logo test image image', 'represent scenario', 'localize object', 'better
position', 'false positive check increasing confidence threshold', 'correct
prediction', 'false positive increase confidence threshold', 'gain trade
precision recall', 'model check', 'see need', 'additional class training
example', 'cat', 'flagged cat', 'dog label training data', 'image dog', 'false
positive effectively helping model learn', 'dog cat new training image',
'model', 'two custom label cat dog test image label cat', 'labeled dog vice
versa case first check mislabeled image training test', 'training image',
'better discriminate cat dog', 'better recall', 'false negative first lower
confidence threshold', 'improve recall', 'better example model variety object
image', 'label', 'example', 'good cooky bad cooky', 'good cooky', 'unique
concept', 'better satisfied performance model', 'available use', 'console',
'code model', 'running perform inference aws cli sdk', 'api specify amazon
resource name amazon recognition custom label model', 'use amazon resource
name', 'arn', 'image want model', 'prediction', 'input image image byte array
base 64 encoded image bite s3 object custom label', 'array custom label object
custom label', 'single object seen concept', 'image custom label', 'label
object', 'seen concept', 'image', 'bounding box object', 'image bounding box
coordinate show object', 'finally custom label', 'confidence score', 'confident
amazon recognition custom label accuracy label bounding box training model',
'prediction label true default', 'calculated threshold value filter return label
specify value men', 'greater model', 'calculated threshold', 'model calculated
threshold model training result amazon recognition custom label console',
'label', 'confidence specify min confidence value', 'confidence value', 'custom
label operation', 'retraining model', 'number custom label', 'custom label
operation', 'max result input parameter return result', 'highest confidence
lowest confidence key takeaway section module model', 'specific domain',
'looking turbochargers', 'many picture turbochargers train model', 'custom
labeling specific business case', 'custom labeling process tool use', 'want
object', 'amazon sagemaker ground truth', 'training data set model', 'machine
learning label image thanks', 'next video']
Topics: ['france', 'byte', 'first', 'two', 'max', '64', 'recognition']
```

```
Video 19:
Key phrases: ['evaluate model classification model', 'target class', '50',
'lower higher improve result', 'sensitivity specificity trade', 'class changing
threshold impact outcome', 'visualize receiver operating characteristic graph',
'also known roc graph', 'build one calculate plot sensitivity true positive rate
false positive rate graph threshold value', 'false positive rate subtracting
specificity', 'false positive rate point', '11 mean correctly identified cat',
'cat bad point line mean proportion correctly classified sample proportion',
'incorrectly classified sample point 00 true positive 0 five model high
sensitivity low false positive rate', 'usually goal', 'better line threshold
recording', 'top left corner data', 'two model', 'another graph use', 'another
evaluation metric use area curve receiver operator', 'also known auc roc auc
part area', 'line auc higher mean model', 'cat cat cat cat use auc', 'model four
number confusion matrix calculate model accuracy', 'also known score', 'correct
prediction', 'number total number prediction accuracy', 'true negative case
data', 'model', 'cat case', 'confident model ability', 'cat roll production lead
example', 'metric choose model evaluation aligns business goal', 'credit card
fraud example case', 'probably good idea lot true negative high true negative
number', 'fact model ability', 'case fraud', 'true positive ideal credit card
company', 'le', 'almost perfect performance', 'fraud case', 'customer',
'business standpoint', 'situation', 'one precision', 'negative prediction
precision proportion positive prediction', 'calculate', 'true positive dividing
true positive plus false positive cost false positive high particular business
situation precision', 'classification model', 'email message spam case', 'model
label email message spam', 'user', 'message', 'example model need', 'precision
evaluation metric account', 'false negative model model successful crucial fall
asleep predict absence illness patient', 'metric use situation', 'one f1 score',
'precision sensitivity', 'one number quantifies', 'overall performance', 'ml',
'algorithm', 'f1 score class imbalance', 'common metric use', 'squared error',
'general purpose', 'classification metric determine prediction model compare
difference prediction actual outcome', 'difference prediction actual value
square difference sum square difference observation skykit', 'directly metric
library metric use linear model r squared trained model performed batch
transformation test data', 'metric use metric help tune model', 'different set
feature train model retrain model', 'better model metric help inform',
'different data retrain model feature', 'k', 'cross validation', 'parameter
model', 'key takeaway section module evaluate model need data model', 'k',
'cross validation different machine learning model', 'different metric
classification use confusion matrix auc roc generate regression use mean squared
section', 'next video']
Topics: ['50', 'one', 'five', '1', 'k fold cross', 'linear', 'first', 'two', '00
11', 'four', '0', '11', 'roc']

Video 20:
Key phrases: ['data collection reviewing extract transform load data data',
'many different system data provider', 'present challenge need', 'data source',
'something', 'machine learning model extract transform load', 'data source
```

```
single location extraction', 'modify data', 'matching record task transform
data', 'data', 'repository amazon s3 typical etl framework', 'several component
example', 'diagram first crawler program', 'datastore source target progress',
'ranked list classifier determine schema data', 'metadata table aws glue data
catalog job', 'business logic', 'perform etl work run job need use schedule
event final note service', 'partition etl process aws', 'fully managed etl
service', 'simple cost effective categorize data clean enrich', 'reliably
various data store aws glue', 'central metadata', 'python scala code', 'flexible
scheduler', 'dependency resolution job monitoring retries', 'available search
inquiry console', 'aws glue api operation interface aws glue service way edit
debug test python scala apache spark etl code', 'familiar development
environment aws', 'glue well suited machine learning', 'label data', 'training
example', 'aws glue training data', 'model duplicate record data source', 'aws
glue', 'duplicate present analysis data engineer aws glue', 'orchestration
complex etl job example aws glue crawl data source present information client
data catalog aws glue run etl job based event', 'new data', 'example use aws
lambda function trigger etl job', 'new data', 'available amazon s3', 'new data',
'aws glue data', 'tool available aws', 'script jupyter notebook', 'data simple
extract load script', 'import variable section import library', 'note', 'moto 3
library aws variable', 'zip file web location local folder extraction download',
'section', 'web request', 'bite url stream stream', 'zip file function',
'extract data', 'extracted file folder upload s3 section', 'folder file amazon
s3 discover script', 'function', 'python application part 2 section', 'part']
Topics: ['first', '2', 'metadata', 'determine schema', '3', 'scala']

Video 21:
Key phrases: ['module 3 section 5 training section', 'going look select model
train data', 'clean prepare data', 'data', 'work training data data frame format
file format', 'csv', 'various algorithm', 'use optimization file format',
'record io protobuf', 'many amazon sagemaker algorithm', 'training data csv
format amazon sagemaker', 'csv file header record target variable first column
amazon sagemaker algorithm', 'best use', 'optimized protobuf record io format
training data', 'advantage pipe mode training algorithm support pipe mode
training job stream data', 'directly amazon s3', 'csv format target variable
training data', 'first column', 'feature', 'right target variable column',
'model data', 'lead', 'recall overfitting model', 'particular data', 'training
data', 'relationship feature label mean model learning relationship pattern',
'new data future', 'hold split data multiple', 'feature label feed algorithm
selected produce model use model', 'prediction validation data', 'likely notice
thing', 'tweak tune', 'feature', 'label', 'performance', 'test data',
'production common split', 'holdout method', '80 data training', '10 validation
10 test lot data', '70 training 15 validation 15 small data set use', 'k', 'much
data', 'relatively good metric order', 'k', 'cross validation randomly partition
data k different segment segment use rest data', 'training order validation
particular segment', 'five fold cross validation available training data', 'five
different chunk training first model', 'chunk training data', 'metric test
second model', 'use piece training model', 'apply test thing', '5 time use
training data', 'five different model', 'different chunk test data', 'data',
'one thing note', 'bias model', 'especially true working structured data example
wine data', 'quality column', 'run model test data', 'pattern', 'biasing model',
'target missing training data', 'data', 'prior splitting', 'sufficient many
library', 'function', 'sometimes useful use', 'sampling', 'approximately
percentage sample target class complete set internet search', 'many way',
'shuffle', 'data', 'one easiest use train split function', 'sklearn amazon
sagemaker', 'four different way train model', 'algorithm available easily
deployed aws console cli jupyter notebook container', 'scene', 'one amazon
sagemaker', 'pre built container support deep learning framework apache mxn',
'pie enchanter', 'machine learning library skykit', 'spark', 'pre', 'docker
image use amazon sagemaker python sdk', 'respective amazon sagemaker sdk
estimator class', 'amazon sagemaker container image use', 'advanced scenario
package scripter algorithm', 'amazon sagemaker use programming language
framework', 'container example team work build', 'ml model build container train
host algorithm', 'someone', 'tune model', 'looking aws marketplace', 'available
model amazon sagemaker', 'high performance scalable machine learning algorithm
optimized speed scale accuracy', 'amazon sagemaker', 'xgboost linear learner
algorithm classification quantitative regression problem', 'also factorization
machine address recommendation time series prediction problem amazon sagemaker',
'support', 'clustering principal component analysis pca', 'customer', 'based
purchasing behavior', 'specialized algorithm processing image', 'deep learning
task', 'three commonly used built algorithm', 'case', 'xgboost extreme
gradient', 'popular efficient open source implementation gradient', 'tree
algorithm gradient', 'target variable combining ensemble estimate', 'simpler
weaker model', 'xgboost', 'machine learning competition', 'variety data type
relationship distribution', 'large number hyperparameters', 'improved fit
flexibility', 'xgboost solid choice problem', 'solution classification
regression problem amazon sagemaker algorithm', 'different training objective',
'best solution validation', 'large number model', 'one need compare method',
'solution', 'continuous objective amazon sagemaker linear learner algorithm',
'significant increase', 'speed naive hyperparameter optimization technique',
'unsupervised learning algorithm attempt', 'discreet grouping', 'data member
group', 'similar possible one', 'another different possible member group find
attribute', 'algorithm', 'determine similarity train model amazon sagemaker',
'training job training job', 'url amazon s3 bucket stored training data url s3
bucket', 'store output job amazon elastic container registry path training
code', 'amazon sagemaker', 'amazon sagemaker amazon sagemaker', 'selection
instance type', 'fit different machine learning use case instance type comprise
varying combination cpu gpu memory networking capacity', 'flexibility',
'appropriate mix resource building training', 'ml model instance type', 'one
instant size', 'scale resource requirement target workload key takeaway section
module', 'split data training testing', 'help', 'model accuracy', 'k', 'cross
validation', 'smaller data', 'two key algorithm', 'xgboost linear learner use',
'unsupervised learning use amazon sagemaker train model section 5 hope', 'next
video']
Topics: ['csv format amazon', '10', 'five', 'one', '80', 'section 5', 'k fold
cross', '15', 'first', '5', 'linear', 'two', '70', 'four', 'second', 'three',
'five fold', '3']
```

```
Video 22:
Key phrases: ['review module wrap knowledge check module', 'formulate problem
business request', 'secure data machine learning build jupyter notebook',
'amazon sagemaker outline process', 'data', 'data', 'pre processed use open
source tool', 'pre process data', 'amazon sagemaker train host machine',
'learning model', 'cross validation test performance', 'ml model use hosted
model inference', 'amazon sagemaker hyperparameter', 'next video']
Topics: []

Video 23:
Key phrases: ['feature engineering', 'describing work outlier', 'clean data
based outlier', 'lie abnormal distance value', 'something', 'accurate prediction
skew', 'data point', 'another column', 'outlier', 'first single variation single
variable univariate', 'outlier second variation', 'one common way', 'univariate
outlier box plot box plot show far data point mean variable box plot show data
value', 'two quartile mean value', 'box', 'whisker scatter plot effective way',
'multivariate outlier example diagram show amount', 'two variable origin
outlier', 'deal pre processing phase pipeline', 'possibly later feature
engineering', 'several different approach dealing outlier', 'outlier outlier
based artificial error mean outlier natural introduced failure', 'data',
'outlier', 'natural log value turn', 'variation', 'extreme outlier value',
'outlier influence', 'overall data', 'feature impute value', 'outlier value',
'good approach outlier', 'artificial error', 'exhaustive list', 'common option
extracted feature', 'select appropriate feature training model three main
feature selection method filter method', 'variable rapper method', 'useful
subset feature training model feature', 'successful model filter', 'faster
cheaper rapper method', 'training model', 'subset feature filter method',
'combination filter wrapper filter method', 'proxy measure', 'actual model
performance fast compute', 'useful feature', 'measure statistical relationship
association', 'two continuous variable second linear discriminant analysis',
'linear combination', 'separate two class third analysis variance', 'anova',
'analyze difference', 'group mean sample', 'chi square single number', 'much
difference', 'observed count count', 'absolutely relationship population
filter', 'computationally intensive rapper produce feature', 'mean feature',
'filter general one rapper filter', 'prediction performance rapper', 'contain
assumption prediction model', 'useful exposing relationship', 'many filter
provide feature', 'instead explicit best feature', 'cutoff point ranking chosen
cross validation filter', 'preprocessing step wrapper', 'larger problem rapper
method', 'predictive model score feature', 'new subset', 'holdout', 'set score
subset', 'calculated counting number mistake', 'holdout', 'error rate model
rapper train new model', 'best performing feature set particular type model
problem forward selection start feature', 'add best model', 'backward
selection', 'feature drop', 'one time select best model embedded method',
'quality filter wrapper method', 'popular example method lasso ridge
regression', 'penalization function', 'key takeaway section module feature
engineering', 'best feature machine learning preprocessing', 'better data work',
'better data', 'better result', 'two category', 'converting data numerical
```

```
value', 'dirty data', 'missing data cleaning outlier', 'dirty data impact model
section', 'next video']
Topics: ['one', 'third', 'first', 'chosen cross', 'section 4', 'two', 'second',
'three']

Video 24:
Key phrases: ['back aws academy machine learning module 3 going work entire
machine learning pipeline', 'amazon sagemaker module discus typical process',
'machine learning problem machine learning pipeline', 'learning process',
'module adapted type machine', 'lot material end', 'able formulate problem
business request', 'build jupyter notebook', 'amazon sagemaker outline process',
'data', 'data', 'pre processed use open source tool', 'pre process data',
'amazon sagemaker train host machine', 'learning model', 'cross validation test
performance machine learning model use hosted model inference', 'amazon
sagemaker hyperparameter tuning job optimize model effectiveness', 'next video']
Topics: ['3']

Video 25:
Key phrases: ['section', 'computer vision computer vision', 'exciting space
machine learning', 'computer vision', 'computer vision machine', 'people',
'thing image accuracy human level greater speed efficiency computer vision',
'deep learning model', 'extraction analysis classification', 'useful information
single image sequence image image data', 'many form single image video
sequence', 'multiple camera three dimensional data computing power algorithm',
'last 10 year led increase capability easier access computer vision technology
computer vision', 'primary use case computer vision', 'image facial
recognition', 'public safety home security way authenticate access personal
device', 'image content management analysis', 'autonomous driving', 'computer
vision technology safety feature car lane detection collision avoidance medical
image analysis computer vision', 'accuracy speed patient medical diagnosis',
'better treatment outcome life expectancy patient', 'well trained computer
vision', 'robotics', 'quality assurance operational efficiency example',
'computer vision problem broken area content recognition', 'thing image
classification problem', 'breakfast lunch dinner classification food answer',
'model use', 'training data', 'model', 'image output category milk peach mashed
potato chicken nugget', 'trained model', 'different image', 'object trey cutlery
napkin', 'image', 'kind object image location object object detection', 'image
category object', 'location box', 'surrounding image known bounding box bounding
box detection', 'height coordinate', 'image use coordinate application object',
'image confidence number', 'object percentage', 'probability object', 'specific
class confidence level', 'action based object detection', 'semantic segmentation
go detail', 'fine boundary', 'object', 'basically fine grained inference',
'pixel image application', 'object segmentation', 'autonomous vehicle advanced
computer human interaction object segmentation key problem field computer
vision', 'course video', 'another dimension computer vision video', 'data work
capture movement people', 'instance example', 'people', 'leave frame', 'camera
use case computer vision', 'detection tracking analyze shopper behavior retail
store', 'path person', 'use face analysis understand detail shopper average age
```

```
range gender distribution', 'emotion', 'another computer vision use case',
'image', 'action', 'motion video example activity', 'package dancing', 'image
baseball player example image', 'batter accuracy pitcher pitching style type
pitch slow ball slider others', 'batter performance', 'specific pitcher
manager', 'data coach player', 'performance coach', 'data game', 'game time
decision', 'various action based speed baseball', 'trajectory', 'calculated ml
model', 'audio visual warning', 'possible foul ball crowd', 'preemptive alarm',
'high probability home run mean event', 'home run', 'automated playing music',
'firework home run', 'home team wrap section key takeaway section', 'computer
vision automated extraction information image divide computer vision', 'two
distinct area image analysis video analysis image analysis', 'object
classification detection segmentation video analysis', 'instance', 'action
recognition motion estimation thanks', 'next video']
Topics: ['one', 'two', 'accuracy pitcher', 'last 10 year', 'three']

Video 26:
Key phrases: ['module', '3 section', 'data set use module', 'guidance formulate
business problem', 'reminder machine learning pipeline', 'previous module map
section module section section 1 cover formulate problem', 'use', 'module
section', 'activity section', 'show tool technique', 'data section', 'training
appropriate machine learning model section show', 'model', 'prediction section',
'performance machine', 'model machine', 'learning pipeline iterative process',
'real world problem', 'many time arrive solution', 'need first section examine',
'business requirement machine learning problem', 'first step phase', 'problem',
'solve goal', 'business goal key use measure performance solution', 'business
problem', 'solution lot question', 'good understanding problem information
problem', 'approach problem', 'traditional approach', 'sense', 'unsupervised
machine learning problem', 'data train supervised model', 'many question',
'business', 'validate use machine learning', 'sure access', 'people data',
'fraudulent credit card transaction', 'transaction process problem business goal
outcome', 'problem statement case', 'intended outcome reduction number
customer', 'membership credit card', 'fraudulent transaction business
perspective', 'problem', 'outcome stage need move qualitative statement
quantitative statement', 'continue example', 'define success problem', '10
reduction number customer file', 'fraudulent transaction', '6 month period
define business side problem time', 'term machine', 'model', 'specific
statement', 'ml model', 'example', 'credit card transaction fraudulent
fraudulent know', 'ml', 'model', 'use information determine type', 'working
historical data customer', 'report fraud transaction', 'data machine',
'historical data', 'supervised learning approach label', 'output', 'transaction
fraud fraud', 'binary classification problem', 'module', 'several data', 'used
access data', 'numerical information composition wine', 'quality wine question',
'data set based composition wine', 'quality', 'price addition question', 'data
set view statistic deal outlier scale numerical data second data set car
evaluation database data', 'heavily text based enables', 'text value number
processed machine', 'third data', 'lab question answer data', 'set based
biomechanical feature', 'patient abnormality data', 'entire end end process',
'trained model', 'use', 'prediction section', 'business problem', 'ml problem',
'key question', 'defining success measure outcome impact solution', 'business
problem', 'one two category first category classification binary multiclass ask
target', 'numerical value section', 'next video']
Topics: ['10', 'second', 'one', 'section 5', '1', 'third', 'first', '6 month',
'section 4', 'two', '7', '8', 'section 2', '3']

Video 27:
Key phrases: ['section review five manage machine learning service', 'various
use case service simplify process', 'machine learning application', 'amazon
transcribe', 'speech audio file produce transcription', 'specific voice audio
file', 'customized vocabulary term specialized particular domain',
'transcription service application integrating web internet protocol',
'communication application amazon', 'common use case', 'amazon', 'also video
production organization', 'subtitle', 'automatically video', 'real time', 'live
feed', 'closed captioning medium company', 'amazon', 'transcribe capture label
content feed content amazon', 'analysis', 'last company record customer service
sale call transcribe analyze result', 'strategic opportunity amazon', 'text
lifelike speech input', 'either plain text file file formatted speech synthesis
markup language ssml ssml markup language', 'special instruction speech sound
example', 'introduce pause flow speech', 'amazon', 'two word', 'speech amazon',
'pcm audio stream format amazon', 'certain regulated workload example', 'u',
'health insurance portability accountability act', 'hipaa amazon', 'payment card
industry data security standard', 'pci', 'ds common use case', 'amazon', 'major
news company', 'amazon', 'vocal content', 'directly written story', 'mapping
apis developer', 'voice geo based application language training company',
'amazon', 'system', 'new language', 'finally animator', 'multi language
experience application create system', 'document', 'one language render store',
'another language', 'part document analysis system amazon', 'fully integrated
machine learning service amazon', 'amazon', 'amazon polly integration extract',
'entity sentiment key phrase', 'amazon', 'polly common use case amazon', 'first
use case', 'international website', 'amazon', 'translate quickly globalize
website amazon translate', 'multilingual chatbots chatbots', 'interface
application amazon', 'chatbot', 'multiple language', 'another use case software
localization localization major cost software', 'global audience amazon',
'translate decrease software development time', 'cost localizing software final
example use case international medium management company', 'medium global
audience', 'amazon', 'cost localization amazon', 'many nlp technique', 'earlier
module extract', 'key ent perform sentiment analysis tag word', 'part speech',
'common use case amazon', 'first example', 'legal medical document legal
insurance medical organization', 'amazon', 'many nlp function', 'module',
'another large scale mobile app analysis mobile app developer use amazon', 'look
pattern usage apps design improvement financial fraud detection', 'another use
case amazon', 'banking financial institution', 'large data', 'financial
transaction', 'fraud look pattern illegal transaction', 'content management
medium content company', 'amazon', 'tag content analysis management amazon lex',
'human language', 'front end application amazon lex', 'conversational engine
power amazon alexa', 'capacity amazon lex solution', 'aws', 'lambda function
scale demand', 'log file conversation analysis', 'common use case', 'amazon',
'lex', 'case', 'building front end interface inventory management sale voice
interface', 'common company', 'chatbots inventory sale application', 'another
use amazon lex', 'customer service interface', 'human like voice application',
'chatbots', 'quality amazon alexa', 'interactive assist', 'amazon lex ml service
customer', 'sophisticated assistance', 'many different industry final example
use case', 'database', 'language amazon lex combined aws database service',
'sophisticated data analysis application', 'language interface main point',
'module amazon', 'transcribe', 'spoken language text amazon', 'real time
translation language amazon', 'many nlp use case', 'amazon lex', 'interface
application thanks', 'next video']
Topics: ['five', 'transcription', 'first', 'amazon', 'two', '1996']

Video 28:
Key phrases: ['main point module module', 'describe use case computer vision',
'amazon management learning service available image video analysis list step',
'custom data', 'object detection', 'amazon sagemaker ground', 'custom data',
'use amazon recognition', 'facial detection', 'introduction computer vision
thanks', 'next video']
Topics: []

Video 29:
Key phrases: ['review', 'correlation data', 'quantify linear relationship',
'scatter plot correlation', 'good tool situation', 'strong weak linear
relationship', 'numerical variable correlation', 'one mean', 'two numerical
feature', 'proportional x correlation', 'two variable', 'proportional x linear
relationship', 'linear relationship mean relationship indication linear
relationship', 'two variable', 'however looking number', 'often easier view
number', 'color highest number', 'one dark brown color', 'positive negative
direction', 'strong correlation', 'seaborn heat map function show correlation
matrix', 'looking chart correlation citric acid fixed acidity', 'wine citric
acid', 'acidity wine', 'however much correlation fixed acidity ph ph measurement
strength acid present fixed acidity measure quantity particular data',
'correlation key takeaway section module', 'point first step', 'data format',
'popular python library working data descriptive statistic help', 'visualization
examine data', 'set detail section', 'next video']
Topics: ['one', '1', 'linear', 'first', 'two', '0']

Video 30:
Key phrases: ['hi welcome section 1 section', 'talk machine learning course
introduction machine learning', 'first discus machine learning fit larger
picture machine learning', 'artificial intelligence', 'broad branch computer
science', 'building machine human task deep learning subdomain machine
learning', 'one mentioned machine learning', 'broader computer science field
known artificial intelligence', 'building machine', 'contemporary popular
culture', 'movie television work fiction example', 'eye control world', 'acting
initiative', 'computer agent perceived action', 'specific goal', 'human helper
worker', 'generally better job working humanity general purpose kind',
'example', 'artificial general intelligence agi capacity', 'task human', 'ai
```

```
problem', 'many field research natural language processing reason', 'knowledge
representation learning perception physical environment interaction', 'u',
'closer area', 'view', 'positive malicious fictional', 'humanity use power
source', 'perhaps concerned risk mass unemployment intelligent machine', '24 7
need break worry', 'next one search', 'definition', 'couple definition example',
'scientific study algorithm statistical model', 'task', 'inference', 'bad
starting point key point', 'algorithm statistical model', 'help', 'idea',
'concrete example', 'write application', 'email message spam', 'machine
learning', 'complex series decision statement', 'else statement', 'use word',
'subject body number link length message', 'hard labor', 'large set rule',
'every possibility machine learning', 'list email message marked spam spam train
machine learning model model', 'pattern word length attribute good indicator
spam message', 'model email message', 'prediction', 'message spam spam deep
learning', 'significant leap forward capability artificial intelligence machine
learning theory', 'deep learning', 'created human brain work artificial neural
network', 'biological neuron', 'brain', 'implementation', 'different artificial
neuron', 'one input single output neuron fire activate output based
transformation input neural network', 'composed layer artificial neuron
connection layer', 'output hidden layer network output single neuron', 'input
neuron', 'next layer network', 'solve problem input layer populated training
data neuron', 'layer answer', 'output layer accuracy output', 'slight change
weight connection neuron neural network', 'connection lead success', 'connection
lead failure', 'course machine learning practitioner', 'lot time', 'best data
feature train contrast deep learning practitioner', 'almost time', 'task',
'time', 'data', 'different n architecture theory', 'deep learning', 'deep
learning problem', 'deep learning address problem complex problem', 'mainstream
machine', 'recent occurrence rapid advancement machine', 'deep learning', 'mid
2000s', 'partly moore law rise cloud computing', 'easier access', 'faster
cheaper compute storage capability rent computing power hour penny',
'substantial investment buy', 'large scale compute cluster 2012 neural network',
'image net large scale visual recognition machine', 'learning competition image
recognition accuracy rate', 'fact', 'human performance', '2015 key takeaway
section', 'data train machine learning model', 'prediction', 'deep learning
technique', 'human biology us layer artificial neuron build network solve
problem advancement technology cloud computing algorithm development',
'corresponding advance machine learning capability application section', 'next
video']
Topics: ['hour penny', 'one', 'every year', '1', 'first', '2012', 'mid 2000s',
'82', '24 7', '2015']

Video 31:
Key phrases: ['video analysis', 'create test data', 'final step train model',
'test data', 'use test data', 'validate evaluate model performance', 'inference
image test data', 'compare result', 'labeling information training data',
'create test data', 'amazon recognition custom label split training data', 'two
data', '80 20 split split mean 80 data', '20 used testing define training test
data', 'amazon recognition custom label', 'model service', 'inspects data',
'correct machine learning algorithm train model', 'model performance metric
```

```
charged amount time model', 'train data set', 'image label', 'complete evaluate
performance model testing amazon recognition custom label', 'test image',
'custom label confidence score value quantifies certainty model prediction
classification problem', 'confusion matrix', 'true positive model', 'presence
custom label test image', 'label', 'truth label image example amazon recognition
custom label', 'cat label cat present image', 'false positive model', 'presence
custom label test image', 'label ground truth label image example amazon
recognition custom label return cat label cat label ground truth image', 'false
negative model', 'custom label present image ground truth image', 'label example
amazon recognition custom label return cat custom label image', 'cat true
negative model', 'custom label present test image example amazon recognition
custom label return cat label image', 'access', 'true positive false positive
false negative value image test data', 'prediction result', 'various metric
label aggregate metric entire test set definition', 'prediction model',
'bounding box level bounding box metric calculated bounding box test image',
'box prediction ground truth', 'amazon recognition custom label', 'various
metric example', 'summary metric evaluation metric label', 'precision metric
label average precision metric entire test data', 'precision proportion',
'positive result', 'correctly classified amazon recognition custom label',
'average recall metric label average recall metric entire test data', 'recall
fraction test', 'set label', 'previous example cat', 'many cat correctly
classified service', 'average model performance score label average model
performance score entire test data', 'f1 score', 'one number quantifies',
'overall performance particular machine learning algorithm', 'f1 score class
imbalance', 'equality precision sensitivity higher value', 'better model
performance recall precision satisfied accuracy model', 'part 3 section', 'part
4 review', 'improve model']
Topics: ['80', '80 20', 'two', '4', '20', '3']

Video 32:
Key phrases: ['module', '3 section', 'model success', 'result point trained
model time evaluate model', 'good job', 'target new future data future instance
unknown target value', 'ass model', 'target answer use assessment proxy
performance future data reason', 'sample data', 'important part phase',
'appropriate metric business situation', 'earlier section problem formulation
phase', 'business problem outcome craft business metric evaluate success model
metric choose phase', 'much possible often high correlation', 'two metric
addition', 'business problem success metric type', 'ml problem', 'working
influence model metric', 'rest module', 'look example common metric used
classification problem', 'common metric used regression problem', 'start',
'simple binary classification problem specific example', 'simple image
recognition model labeling data', 'either cat cat model trained use test data',
'performance model compare', 'value actual value plot value table', 'example',
'insight', 'well model', 'high level comparison', 'class', 'p positive predicted
label class', 'true positive good outcome model', 'similarly actual label cat',
'n negative predicted label class', 'true negative also good outcome model case
model', 'correct outcome use testing data', 'two possible outcome', 'one actual
class neg', 'class positive cat', 'false positive prediction positive
```

```
incorrect', 'cat', 'class negative cat part one section', 'part 2 review']
Topics: ['one', 'first', '2', 'two', '3']

Video 33:
Key phrases: ['section', 'going discus challenge machine learning', 'many
challenge machine', 'learning lot', 'poor quality inconsistent data',
'significant portion job', 'access', 'enough good data representative problem',
'key issue', 'overfitting model data', 'science experience staffing team data
scientist', 'effective management support', 'machine learning business
landscape', 'problem complex formulate machine learning problem', 'resulting
model', 'might get adopted cost building', 'operating machine', 'learning
solution', 'finally technology map business unit access data', 'data secured
meet regulatory requirement tool framework', 'solution', 'system', 'important
question', 'need able answer address', 'many machine learning problem',
'existing model', 'substantial machine learning knowledge', 'aws', 'managed
service machine learning', 'sophisticated machine learning capability
application basic developer skill', 'apis', 'pre built model use', 'one
example', 'yolo popular computer vision model addition scenario', 'aws
marketplace', 'buy model service independent software vendor', 'key takeaway
section', 'many machine learning challenge', 'related data', 'service', 'machine
learning problem', 'domain support', 'amazon recognition computer vision problem
section', 'next video']
Topics: ['one', 'recognition', 'section 5', 'today']

Video 34:
Key phrases: ['amazon academy machine', 'foundation module', 'course', 'various
job role machine learning domain', 'machine learning completing module',
'course', 'role data scientist business', 'resource learning', 'course', 'take
course', 'first complete aws academy cloud foundation', 'general technical
knowledge', 'foundational computer literacy skill', 'basic computer concept
email file management good understanding internet', 'intermediate skill python
programming general knowledge applied statistic', 'finally general business
knowledge', 'important course', 'insight information technology', 'business',
'also important business related skill set communication skill leadership skill
orientation', 'customer service course', 'key concept machine learning tool',
'us', 'work aws service machine learning', 'recognize machine', 'part artificial
intelligence', 'artificial intelligence machine', 'learning terminology',
'machine learning', 'solve business problem', 'machine learning process list
tool available data scientist', 'implement machine learning pipeline',
'formulate problem business request', 'build jupyter notebook', 'amazon
sagemaker outline process', 'data', 'data', 'pre processed use open source
tool', 'pre process data', 'amazon sagemaker train host machine learning model',
'cross validation test performance machine learning model use hosted model
inference', 'amazon sagemaker hyperparameter', 'managed amazon machine learning
service', 'computer vision natural language processing review course outline',
'course', 'introduction machine learning module', 'implement machine learning
pipeline amazon sagemaker module', 'service computer vision', '7 summary
course', 'work', 'aws certified machine learning specialty', 'five slide provide
```

```
detail subtopics', 'major concept', 'machine learning section', 'overall field
machine learning machine learning', 'artificial intelligence deep learning
section', 'common business problem', 'machine learning section', 'general
workflow', 'machine learning problem', 'common machine learning term section',
'4 review commonly used tool machine', 'professional lastly section', 'common
challenge face working machine learning problem module', 'introduction amazon
sagemaker', 'implement machine learning pipeline module focus application
machine', 'problem', 'several public domain data', 'example machine', 'pipeline
section', '1 introduces', 'business problem data', 'use module section', '8
describe phase machine learning pipeline', 'computer vision example application
section', 'secure data section', 'different technique', 'data section', 'learn
process feature engineering section', 'describe step', 'train model sagemaker
section', 'overview option sagemaker', 'model', '7 8 cover evaluate tune model
sagemaker module', 'machine learning create forecast based time series data
section', 'common application section', '2 outline pitfall', 'time series data',
'forecast', 'get overview use amazon forecast module learn', 'computer vision
section', 'general problem', 'computer vision section', 'learn process', 'image
video section', 'step need', 'take prepare data', 'computer vision module',
'natural language processing machine learning section', 'general set problem',
'natural language processing section', 'service use', 'amazon lex amazon',
'amazon', '7 final module course module review', 'course', 'next step take',
'specialty section 1 module summarizes topic', 'course section', 'aws
documentation', 'two common framework', 'aws service', 'finally section', 'aws
certified machine', 'special section', 'common job role machine', 'analytical
statistical programming skill data scientist', 'use skill collect analyze',
'university', 'degree data science data scientist', 'related field', 'statistic
computer science economics data scientist', 'technical competency statistic
machine learning programming language', 'data analytics', 'career machine',
'engineer skill', 'similar data scientist skill', 'data scientist machine
learning engineer', 'technical competency statistic machine learning',
'programming skill software architecture analysis interpretation machine
learning engineer', 'programming architecture skill design', 'machine learning
system machine', 'learning engineer', 'often previous experience software
development', 'heavily programming software engineering machine learning role',
'everything', 'many different path', 'open applied science researcher primary
focus type science working need skill data scientist', 'skill chosen domain',
'technical competency statistic machine', 'many software developer', 'machine
learning application interested career software developer', 'machine learning
technology study machine learning developer primary focus software development
skill', 'skill data scientist', 'coursework statistic', 'student guide', 'link
documentation resource', 'course', 'introduction', 'next video']
Topics: ['five', 'section 5', 'section 3', 'first', '2', '5 6', '4', '7',
'section 1', 'lex amazon', 'section 4', '3', '2 8', 'two', '7 8', 'section 2',
'amazon', '6', 'machine learning']

Video 35:
Key phrases: ['look', 'technique challenge', 'securing data', 'machine
learning', 'original example', 'credit card fraud formulated problem data',
'actually train model', 'intended business outcome access data', 'much data
solution use', 'data', 'one centralized repository answer question essential
stage good news budding data scientist', 'many place', 'data private data
existing customer', 'everything', 'log file customary invoice database private
data', 'depending problem', 'many case', 'private data', 'many different system
look', 'source', 'use data', 'available commercial organization company reuters
change healthcare dun bradstreet foursquare', 'database subscribe', 'healthcare
transaction', 'global business record location data', 'useful insight', 'many
open source data', 'wine quality movie review data', 'available use research
teaching purpose aws', 'uci machine', 'repository good place', 'open source
data', 'government health organization source data', 'useful supervised machine
learning problem', 'lot data', 'observation', 'know target answer prediction
data kind data', 'target answer prediction', 'labeled data observation data',
'two element target end feature target answer want', 'credit card transaction
example target', 'observation', 'either fraud fraud feature attribute example
use identify pattern', 'target answer feature credit card example', 'transaction
vendor amount dollar transaction', 'source target fraud fraud', 'typically
information', 'transaction', 'complete actual card owner', 'fraudulent
transaction statement information', 'transaction', 'exactly purpose', 'train
future model', 'element', 'data', 'one original question data', 'actually train
model', 'desired output', 'business outcome example stage', 'question domain
knowledge', 'feature target data model', 'accurate prediction data
representative data', 'model', 'prediction example', 'credit card fraud', 'data
positive fraudulent transaction', 'data negative non fraudulent transaction',
'type data machine learning', 'algorithm', 'pattern', 'two type suppose average
amount fraudulent transaction', 'training data set', 'small fraction fraudulent
observation', '0 4 case difficult model', 'pattern related fraudulent
transaction', 'production', 'many different service aws', 'store data key
service', 'amazon simple storage service', 'also known amazon s3', 'object level
storage s3 store', 'much data', 'form object', 'file', 'csv file file format
need s3 accessed web based aws management console', 's3', 'sdk third party
solution', 'api sdks training data', 'planning run training job', 'several time
different algorithm parameter', 'amazon fsx lustre file system service speed
training job', 'data amazon', 'high speed', 'training job', 'fsx lustre', 'data
s3', 'available sagemaker', 'repeated downloads common s3 object', 'data',
'elastic file system amazon efs', 'efs data source', 'data launch training job',
'data movement', 'often case environment data scientist home directory amazon',
'quickly iterate model', 'new data sharing data colleague', 'different field
label data', 'example data scientist', 'jupyter notebook', 'training job amazon
sagemaker', 'jupyter notebook drop column relaunch training job', 'resulting
model', 'one work better many aws service resource', 'data example', 'amazon
relational database service amazon rds managed relational database service',
'amazon redshift manage data warehouse service another option amazon timestream
manage time series database', 'large amount data internet thing iot',
'instance', 'amazon elastic compute cloud', 'amazon ec2 host database instance
data source', 'useful data source', 'data machine learning', 'look next part 1
section', 'part 2 review extract']
Topics: ['one', '1', 'third', 'first', '2', 'two', 'healthcare dun', 'everything
```

```
log file customary invoice database private', '4', 'kaggle uci machine
learning', 'directory amazon', '0', '3']

Video 36:
Key phrases: ['welcome section', 'forecasting use case', 'important area
machine', 'important many opportunity', 'future outcome', 'many opportunity',
'time component', 'however time component', 'additional information', 'time
series problem', 'type prediction', 'time series data', 'one variable second one
multivariate data', 'one variable several common pattern time series data first
pattern trend trend', 'pattern value', 'decreasing staying time', 'seasonal
pattern', 'day pattern cyclical pattern', 'similar seasonal pattern pattern',
'large retail sale event', 'time year', 'data time', 'random discernible
pattern', 'many us', 'use forecasting marketing application search sale
forecasting demand projection', 'inventory management system', 'required
inventory level forecasting energy consumption help', 'energy needed weather
forecasting system', 'government commercial application agriculture section',
'next video']
Topics: ['one', 'first', 'time year month day', 'two', 'section 1', 'second']

Video 37:
Key phrases: ['module', 'start introduction forecast look time series data',
'different kind data', 'building forecast end module', 'able describe business
problem', 'amazon forecast describe challenge working time series data list
step', 'create forecast', 'amazon forecast use amazon forecast', 'prediction',
'next video']
Topics: ['4']

Video 38:
Key phrases: ['back time review module wrap module', 'describe business
problem', 'amazon forecast describe challenge working time series data list
step', 'create forecast', 'amazon forecast use amazon forecast', 'prediction
thanks']
Topics: []

Video 39:
Key phrases: ['section', 'going focus processing time series data different type
data', 'far time series data data', 'chronological sequence', 'time machine',
'learning model positive impact model derive', 'meaning change data point time
time series data', 'correlated mean dependency data point mixed result',
'dealing regression problem regression', 'data', 'independent need', 'method',
'data dependence', 'increase validity prediction addition time series data',
'related data augmenta forecasting model example', 'prediction retail sale',
'information product', 'item identification sale', 'number unit', 'time',
'metadata', 'brand name genre music video group', 'better data', 'multiple data
source', 'timestamp data', 'observe difference timestamp format', 'missing data
case example', 'data', 'data', 'sequence month number database', 'year knew
data', 'stamp data', 'utc form data', 'example', 'database car serviced garage
timestamp', 'time car', 'final entry entered system', 'model hourly caloric
```

```
intake patient', 'daily data', 'adjust target time scale', 'data', 'might time
stamp', 'extrapolate time series', 'data domain example', 'measurement vector',
'image final note', 'daylight', 'world', 'twice year time zone common occurrence
real world forecasting problem', 'missing value raw data missing value', 'harder
model generate forecast primary example retail stock situation demand
forecasting item', 'stock sale day zero forecast', 'many reason value', 'missing
missing value', 'possible measurement error example service', 'certain data',
'correctly another example measurement', 'correctly retail primary example
inability', 'correct measurement stock situation demand forecasting mean demand
equal sale day', 'missing data', 'first method', 'us', 'last known value missing
value building idea', 'average us average last known value', 'missing value',
'us', 'next known value missing value danger', 'future calculate past bad
forecasting method', 'ahead avoided interpolation us equation', 'missing value',
'zero fill', 'retail missing sale data', 'missing data', 'order day', 'happened
case', 'want fill', 'missing value', 'data different frequency example', 'data',
'exact time stamp sale', 'inventory level data different frequency data', 'data
compatible question', 'downsample', 'finely grained time', 'grain time example
show', 'hourly data', 'daily data', 'downsampling need', 'combine value previous
case sale data summing quantity', 'sense data temperature', 'average
understanding data', 'best course action', 'downsampling upsampling move', 'le',
'time', 'grain time problem', 'upsample sale data daily sale hourly sale', 'data
source reference able case', 'something', 'frequency', 'another time series',
'irregular time series specific domain knowledge', 'case', 'conversion retail
example', 'single order day', 'specified hour temperature', 'daily temperature
hourly slot use formula calculate curve data science outlier mix', 'positive
negative attribute', 'true time series data', 'sale data order', 'unusually high
number item', 'forecast calcul order size', 'outlier anomaly reason', 'data
preparation', 'error value', 'outlier', 'smooth data', 'feature visualization',
'data', 'noise plot important understand smoothing data impact', 'noise',
'better model', 'equally important question', 'compromise model model', 'noisy
data', 'smooth data production part 1 section', 'part 2 review time series
specific challenge tool algorithm', 'u wrangle data']
Topics: ['metadata data', 'utc', 'forecast calcul', 'first', '2', '12 00',
'day', 'zero', 'month day year', 'wrangle', '1', 'hourly', 'third', 'year',
'daily', 'anomaly', 'future year', 'twice year', 'contains year month day',
'order day', 'hour', 'anomaly reason']

Video 40:
Key phrases: ['video analysis', 'create training data', 'data', 'contain
information', 'least two label least 10 image', 'label image data set', 'earlier
use amazon recognition custom label console amazon sagemaker ground truth label
image train amazon recognition custom label model image', 'label', 'image',
'object scene concept', 'earlier data', 'need least two defined label', 'also
image', 'must least one assigned label identifies object', 'concept image',
'label image whole label known image level label useful identifying scene
concept', 'example', 'one image', 'u', 'beach scene', 'ko olina island oahu u',
'beach', 'entire image', 'label specific area image contain object', 'example',
'model', 'amazon echo device', 'different type echo device image model',
'information device located image', 'corresponding label identifies type device
disinformation known localization information location device', 'box image',
'u', 'bounding box surround amazon echo dot image', 'amazon echo', 'box output
labeling process', 'manifest file manifest file image level label', 'label class
name', 'metadata image', 'object detection manifest', 'information', 'image
bounding box identifies', 'object image', 'label bounding box', 'sagemaker
ground truth', 'high quality training data', 'machine learning model use',
'data', 'need labeling', 'detailed instruction need', 'submit job decide process
image', 'label data', 'use worker', 'amazon mechanical turk service vendor
company', 'use label data', 'output sagemaker ground truth train model', 'amazon
recognition custom label sagemaker ground truth', 'active learning automate
labeling input data active learning machine learning technique identifies data',
'worker sagemaker ground truth functionality', 'automated data labeling
automated data labeling reduce time cost', 'label data', 'automated labeling
incur amazon sagemaker training inference cost', 'use machine learning label
image use machine learning talk work sagemaker ground truth', 'automated data
labeling job selects', 'random sample input data object', 'human worker label
data', 'us', 'automated data labeling sagemaker ground truth run batch transform
job', 'validated model inference validation data batch inference', 'confidence
score quality metric object validation data', 'confidence score object', 'step',
'four confidence score', 'threshold expected quality automatic labeling exceeds
requested level accuracy object', 'step six produce data', 'unlabeled data
confidence score sagemaker ground truth', 'data', 'low confidence score data',
'human worker', 'additional labeling sagemaker ground truth', 'us', 'existing
human label data', 'additional human label data', 'new model process', 'another
stopping condition', 'example', 'labeling stop', 'budget human annotation',
'automated data', 'large data', 'minimum number object', 'automated data',
'minimum 5 000 object part 2 section', 'part', '3 review', 'improve model']
Topics: ['one', '10', '5', '5 000', '2', '1 250', 'six', 'metadata', 'two',
'four', '3', 'ko olina island']

Video 41:
Key phrases: ['hi welcome module 6 aws academy machine learning introduction
natural language processing module introduce natural language processing',
'description major challenge', 'nlp overall development process nlp application
review 5 aws service use speed development mlp based application completing
module able describe nlp use case', 'managed amazon', 'ml service describe',
'managed amazon ml service available nlp']
Topics: ['5', '6']

Video 42:
Key phrases: ['wrangling time series data seasonality data kind repeating
observation frequency observation stable example sale', 'higher sale', 'fourth
quarter consumer retail', 'even higher sale fourth quarter aware data multiple
type seasonality data', 'many time', 'incorporate seasonality information
forecast instance', 'holiday good example sale chart', 'total revenue', 'u',
'source', 'many correlation plotted site', 'none', 'sense data', 'acting
correlation', 'real world experiment', 'two random time series data', 'number',
'low correlation', 'slope data', 'strong correlation', 'stable system level
stability stationarity inform', 'system', 'behavior', 'future behavior system',
'low stability', 'future', 'determine trend time series', 'series trend',
'difficult compare', 'another series', 'value series', 'overestimate
correlation', 'two series', 'one special problem', 'face time series data',
'machine learning problem goal building ml model', 'signal noise autocorrelation
form noise', 'separate observation', 'independent time series autocorrelation',
'accuracy model', 'algorithm look module help', 'autocorrelation factor',
'seasonality influence model', 'select produce forecast algorithm', 'seasonality
autocorrelation others panda', 'financial data analysis mind', 'good handling
time series data', 'index panda', 'select data', 'date', 'task panda', 'function
panda', 'insight autocorrelation information panda time series', 'panda
documentation', 'one task building forecasting application', 'appropriate
algorithm choice', 'type data', 'feature data amazon forecast support', 'five
algorithm others', 'algorithm', 'data', 'slightly different characteristic
example', 'auto regressive integrated moving average', 'also known arima', 'auto
correlation influence pattern observation', 'exponential smoothing',
'seasonality', 'time series data sequence data', 'time element', 'different
regular data set time challenge', 'different time format', 'missing data',
'upsampling smoothing dealing seasonality', 'weekday yearly cycle', 'bad
correlation panda', 'excellent time series support function', 'time', 'amazon
forecast arim', 'next video']
Topics: ['0 1', 'one', 'five', 'fourth quarter', 'quarter fourth quarter',
'weekday', 'two']

Video 43:
Key phrases: ['use amazon forecast', 'predictor generate forecast generate
forecast', 'machine learning development pipeline', 'course', 'data', 'import
much data historical data related data', 'basic evaluation feature engineering
use data train model', 'algorithm', 'algorithm best data amazon forecast',
'select auto', 'ml algorithm', 'select domain data', 'custom domain', 'trained
model use model', 'forecast', 'input data set group generated forecast query
forecast', 'bucket amazon s3', 'data forecast', 'overall process', 'working
amazon', 'import historical related data amazon', 'key data selects', 'us
algorithm', 'optimize custom model', 'predictor', 'create forecast', 'predictor
data', 'retrieve forecast aws management console export forecast comma delimited
file', 'api aws cli command', 'retrieve forecast work amazon', 'select domain
working domain', 'retail web traffic', 'everything', 'domain', 'efficiency
predictor', 'specific type data supply', 'data item identifier timestamp
observation number sale item', 'timestamp example data', 'retail demand forecast
time series need time transaction', 'place', 'ideally utc format item id item',
'many item', 'metadata item', 'category item color attribute link', 'back time
series data item id item metadata', 'related data', 'useful forecast', 'sale
price promotion data', 'back item', 'timestamp item', 'id example data',
'provide web traffic forecast time series need web page id number page view',
'month timestamp related data', 'useful forecast', 'page category navigation
content category', 'geographic identifier web client metadata', 'region sale
promotion information amazon forecast predictor', 'algorithm train model use
```

```
model', 'forecast', 'input data set group help', 'amazon forecast', 'predefined
algorithm arima', 'profit', 'auto ml feature', 'try algorithm', 'one best
predicting data', 'data', 'validate score model data', 'usually random sample
available data time series data', 'data', 'import data amazon forecast',
'diagram show training data', 'train model', 'data', 'multiple time train
model', 'metric determine model', 'one change amazon', 'test window offset
parameter', 'predictor set value', 'algorithm', 'default value trained model',
'measure accuracy', 'next first amazon forecast evaluation metric', 'los amazon
forecast creates forecast', 'probabilistic prediction three distinct quantile 10
50 90 prediction quantiles', 'much uncertainty', 'true value', 'value',
'retailer', 'forecast product demand winter glove', 'well fall winter',
'sufficient storage space cost invested capital high price overstocked winter
glove concern', 'p10 quantile order', 'relatively low number winter glove', 'p10
forecast overestimate demand winter glove', 'winter glove', '90 time p50',
'quantile predicts', 'true value', 'value', 'continuing winter glove example',
'moderate amount demand glove', 'p50 quantile order glove', 'p90 quantile
predicts', 'true value', 'value', 'understocked glove result', 'large amount',
'glove', 'extremely high cost invested capital low case', 'p90 quantile order
glove amazon forecast', 'associated loss error quantile weighted quantile lost
calculate', 'certain quantile actual demand', 'either direction lower w quantile
lost metric mean model forecast reliable root mean square error rmse', 'another
method', 'reliability forecast', 'w quantile lost rmse calculates',
'difference', 'actual target value data', 'calculate rmse rmse value', 'lower
rmsc metric', 'model forecast reliable example web retailer', 'accuracy metric
evaluate forecast retailer', 'demand sale particular brand shoe input sale
record brand amazon forecast', 'predictor predictor', 'forecasted demand 1 000
pair p10 p50 p90 value', 'quantile lost value', '10 time fewer 880 pair', '50
time fewer 1 050 pair', 'retailer use value determine level inventory', 'use',
'amazon forecast train use model time series data specific schema', 'domain
retail ec2 capacity planning use custom schema', 'supply least time series
data', 'metadata related data', 'information model', 'machine learning problem
data', 'training testing data', 'new account time', 'element use rmse w quantile
lost metric evaluate efficiency model video']
Topics: ['one', 'rmse', '90', '1 050', 'first', 'metadata', '1 200', '50', 'w
quantile', 'winter', '1 000', 'p90', 'los amazon', 'rmse rmse', 'three', 'rmse w
quantile', '10', 'amazon', 'square error', '880']

Video 44:
Key phrases: ['feature engineering', 'clean data', 'addition', 'string data
numerical data', 'clean data', 'several potential problem area', 'string data',
'sure string consistent', 'variable use', 'one variable', 'number door car
scale', 'another variable', 'state california scale', 'data set', 'variable
combine safety maintenance single variable', 'high maintenance', 'train machine
learning system variable', 'single variable two separate variable', 'data',
'outlier cover technique dealing situation section', 'data missing example
column data', 'data data collection error', 'data', 'particular feature data
collection process', 'underway missing data', 'relationship related feature
target variable', 'regardless data', 'important deal issue', 'unfortunately
```

```
machine learning algorithm', 'missing value', 'meaningful relevant problem',
'function', 'missing data', 'drop impute missing value question', 'better
understanding value', 'much data missing value', 'larger data', 'missing value',
'data', 'larger portion respective row column case imputation', 'likely better
option contrast', 'column road', 'large percentage missing value case', 'entire
row column', 'drop row missing data use', 'function drop row', 'data', 'specific
data value', 'subset alternative', 'missing value impute value missing value',
'impute missing value categorical value missing value', 'mean median frequent
value numerical continuous variable missing value', 'mean median impute single
row missing data known univariate also multiple row known multivariate look
univariate example computer function', 'impute missing value', 'fairly small
data', 'two missing value missing value imputed strategy mean first calculate
mean mean 3 2 2 5 impute mean value missing value data library', 'impute
package', 'complex way impute data example', 'k nearest neighbor soft cute
multiple imputation', 'equation others', 'two section', 'part 3 review work
outlier data']
Topics: ['clean data', 'first', '2 8', 'two', 'california', '3', 'thousand']

Video 45:
Key phrases: ['section', 'look', 'model section', 'train model', 'next section',
'ready deploy model', 'looking phase order', 'deployment want test model',
'performance metric first need', 'inference prediction model', 'deployment
deployment testing', 'different production', 'mechanic amazon sagemaker',
'everything', 'host model simple testing evaluation request deployment', 'ten
thousand request', 'model single prediction', 'model amazon sagemaker hosting
service sagemaker', 'multiple computer run model', 'load balanced endpoint
application call api endpoint', 'prediction model scale number instance based
demand', 'prediction entire data', 'use amazon sagemaker batch transform',
'permanent endpoint sagemaker spin model perform prediction entire data set
provide store result amazon s3 shuts', 'useful performing batch prediction test
model', 'entire validation set model', 'code process', 'individual result goal
deployment phase', 'managed environment host model', 'inference', 'securely low
latency model', 'production monitor production data retrain model', 'necessary
newly deployed model need', 'current production data new data accumulated time',
'alternative new outcome deploying model', 'model amazon', 'ml instance',
'sagemaker', 'care rest launch instance', 'model set secure http endpoint
application application need', 'api call endpoint achieve inference low latency
high throughput architecture', 'new model application minute change model',
'change application code sagemaker', 'production compute infrastructure behalf',
'health check', 'security patch', 'conduct', 'routine maintenance', 'amazon
cloud', 'trained model', 'create', 'endpoint', 'either code', 'sagemaker console
planning host single model', 'endpoint model', 'host multiple model need',
'multi model endpoint multimodal endpoint', 'scalable cost effective solution',
'large number model use shared serving container', 'host multiple model',
'hosting cost', 'endpoint utilization', 'single model endpoint', 'deployment
overhead sagemaker', 'loading model memory scaling model based traffic pattern',
'machine learning model production', 'prediction', 'new data', 'data processing
step', 'training inference request', 'incorrect prediction result', 'inference
```

```
pipeline', 'reuse', 'data processing step model training infer', 'two separate
copy code help', 'accuracy prediction', 'development overhead sagemaker managed
service inference pipeline', 'sequence container ec2 instance endpoint batch
transform job', 'additionally sequence feature processing inference', 'low
latency container', 'ec2 instance', 'point', 'train model', 'sagemaker handle
api call application perform prediction', 'batch transformation goal model
generate prediction answer business problem', 'sure model', 'good result',
'production', 'multi model endpoint support', 'resource', 'multiple model',
'deploy section', 'next video']
Topics: ['one', 'first', '6', 'two', 'ten thousand']

Video 46:
Key phrases: ['section', 'quick high level overview machine', 'terminology',
'typical workflow cover topic detail', 'larger picture', 'business problem
team', 'machine learning', 'problem formulation phase', 'ml problem formulated
problem', 'data', 'one data source data source', 'data type need', 'reconciled
form single cohesive view data', 'visualize data', 'statistic determine data',
'consistent used machine learning', 'look data source', 'data four column',
'data', 'data result', 'table', 'ml problem column represent feature',
'represent instance issue data instance case', 'subject matter expert functional
understand authenticity data example date', 'someone', 'data pool', 'male',
'import issue', 'shifted position', 'actual location molly city capital
republic', 'time', 'role expert', 'one largest impact success machine learning
project consistent correct data data good shape time train model process',
'iterative fluid', 'find model meet business goal feature engineering process',
'feature model trained feature column data data', 'goal model', 'target value
new data', 'ml', 'algorithm', 'feature', 'target example target data average
number step', 'correct feature', 'new feature', 'data format', 'consistent
consistent format', 'model', 'change cosmetic reason', 'problem', 'solve data',
'name feature example data country', 'traditional database', 'move country
lookup table reference', 'ml', 'algorithm', 'data instance single row', 'ml',
'algorithm', 'numerical data process', 'country text country', 'iso code',
'model', 'numerical value', 'uk', 'code value', 'would significant iso code
value u 01 case splitting data multiple column fine known categorical cod',
'later course type data', 'text value numerical value example', 'male female
numeric value', 'easily model remaining feature', 'age birth', 'bm', 'dow', 'age
birth', 'depending problem', 'age impact target variable day week born worry
sound complicated learn feature engineering', 'data', 'identified feature', 'use
time train model', 'data train model fact', 'hold data data test', 'rest data',
'known hyperparameters', 'training job', 'trained model train model use test
data', 'well model performs', 'instance model', 'seen use perform prediction',
'target test data', 'two value comparison', 'metric give data', 'well model',
'change model data feature hyperparameters', 'model yield', 'training model real
danger', 'underfitting model model', 'training data', 'model', 'well training
data', 'well evaluation data model memorizing data', 'generalize unseen example
model underfitting training data model', 'poorly training data model capture
relationship input example', 'x target value', 'understanding model', 'important
understanding root', 'poor model accuracy misunderstanding guide', 'corrective
```

```
step', 'overfitting training data', 'prediction error training data evaluation
data', 'avoid', 'model satisfied result', 'model', 'best possible prediction',
'different phase', 'hand experience knowing process', 'managed service',
'certain amazon', 'ml service', 'machine learning pipeline process guide process
training evaluating model iterative process', 'three broad step data processing
model training model evaluation video']
Topics: ['one', 'zero', '80', '01', '11', '44', 'age birth month day week', 'day
week', 'february 11th year 1969', 'two', 'november 2nd', 'four', 'three', '3',
'2 1969']
```

0.7 5. Creating the dashboard

(Go to top)

[16]:

```python
from ipywidgets import Label, widgets

# Create a search box for the key phrases and topics
search_box = widgets.Text(description='Search:')
search_description = Label(value='Enter a keyword or topic to search for:')

display(search_description,search_box)

# Define a function to handle the search
def handle_search(sender):
    query = sender.value.lower()
    found_video = False
    output.clear_output()  # clear previous output
    with output:
        for (i, ((key_phrases, topics), audio_file)) in␣
 ↪enumerate(zip(key_topics, audio_files)):
            if any(query in phrase.lower() for phrase in key_phrases) or␣
 ↪any(query in topic.lower() for topic in topics):
                found_video = True
                print(f"Searched keyword {query} is related to Transcript of␣
 ↪{audio_file}")
    if not found_video:
        with output:
            print(f"No video found for searched keyword'{query}'.")

# Attach the search function to the search box
search_box.on_submit(handle_search)

# Create an output widget to display search results
output = widgets.Output()
display(output)
```

```
Label(value='Enter a keyword or topic to search for:')
```

```
Text(value='', description='Search:')
```

```
Output()
```

[ ]:

```
```