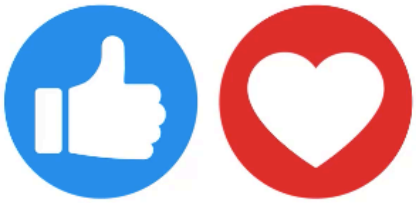
Udemy - 3 - Modeling - AI Services

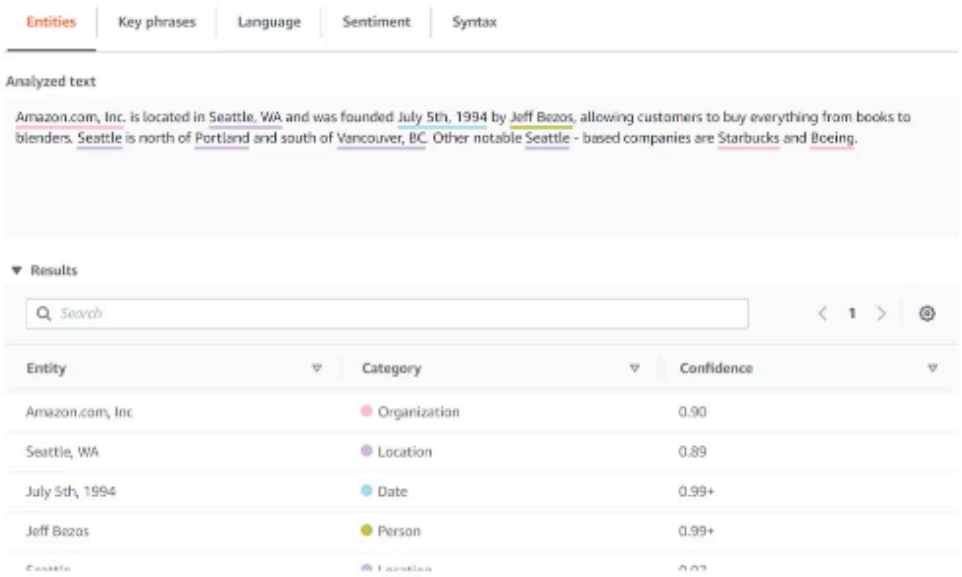
Amazon Comprehend

Amazon Comprehend

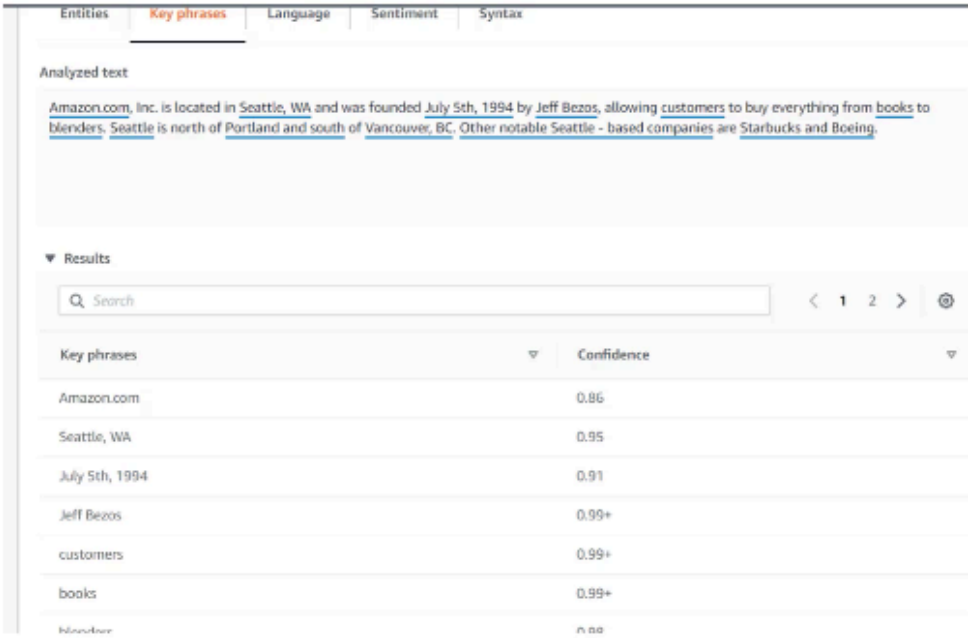
- Natural Language Processing and Text Analytics
- Input social media, emails, web pages, documents, transcripts, medical records (Comprehend Medical)
- Extract key phrases, entities, sentiment, language, syntax, topics, and document classifications
- Can train on your own data



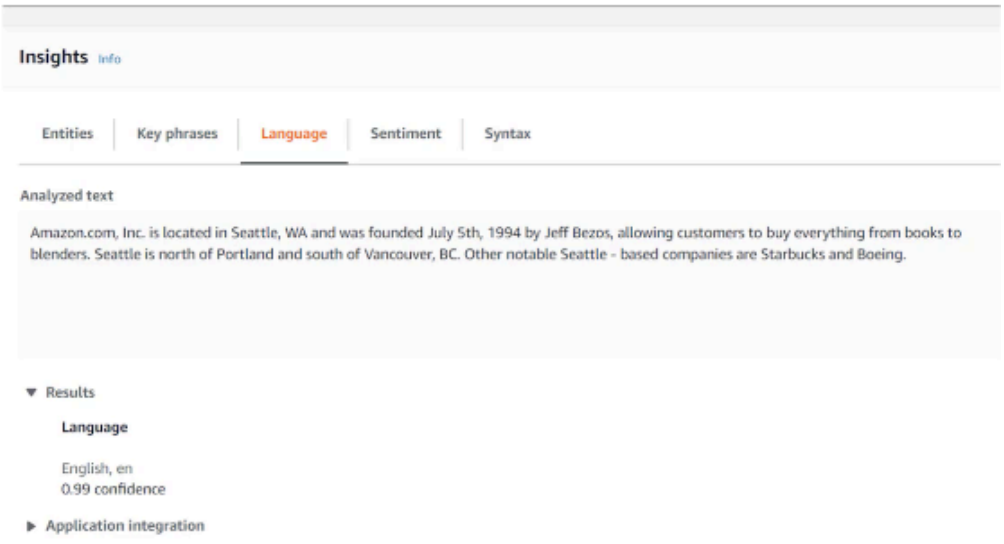
Entities



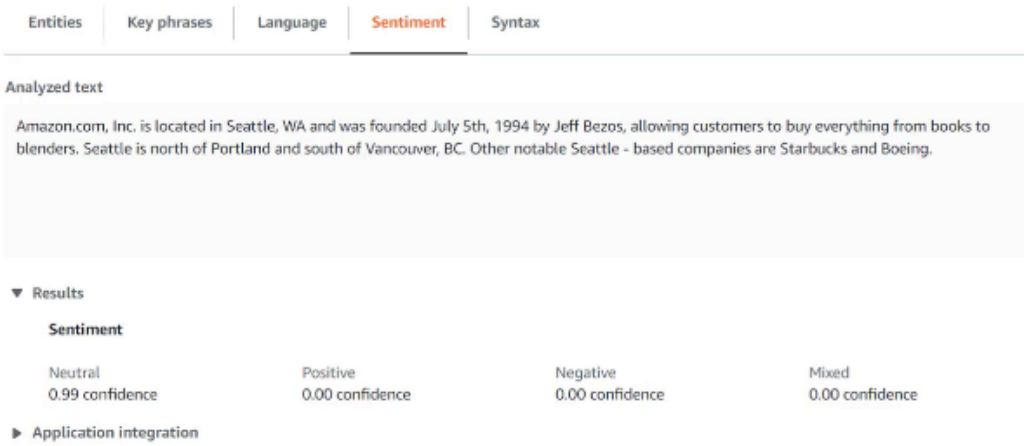
Key Phrases



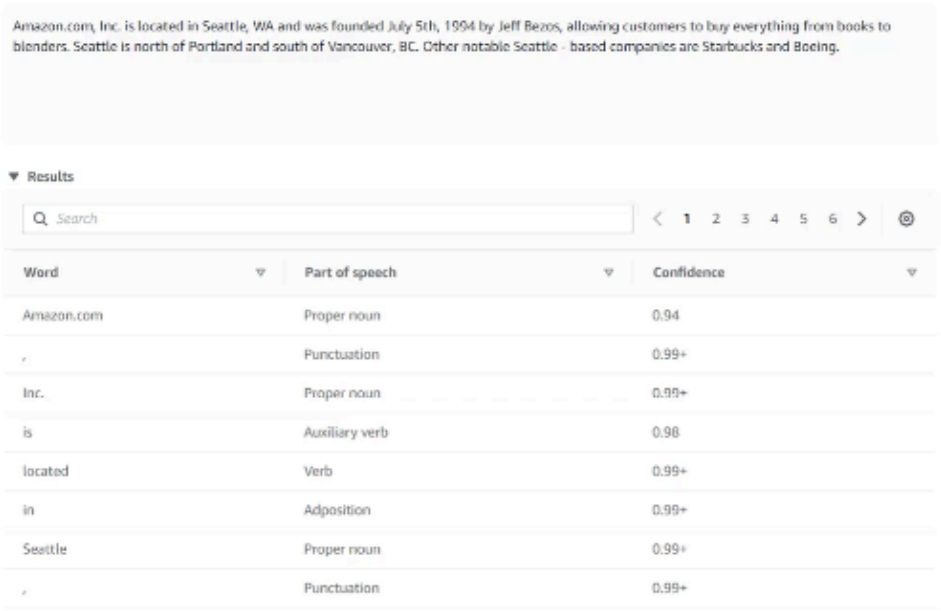
Language



Sentiment



Syntax

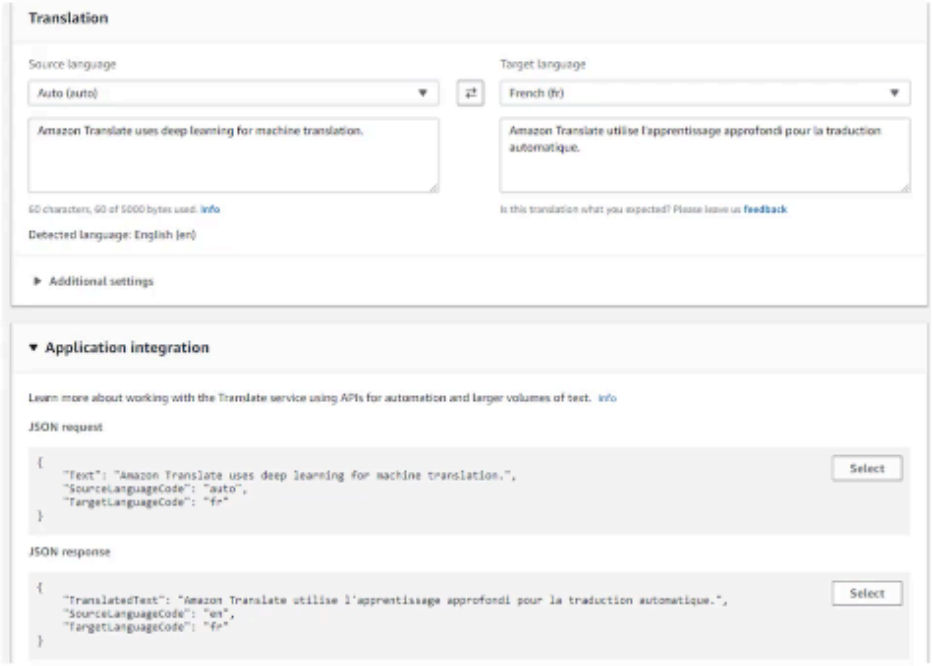


Amazon Translate

Amazon Translate

- Uses deep learning for translation
- Supports custom terminology
  - In CSV or TMX format
  - Appropriate for proper names, brand names, etc.

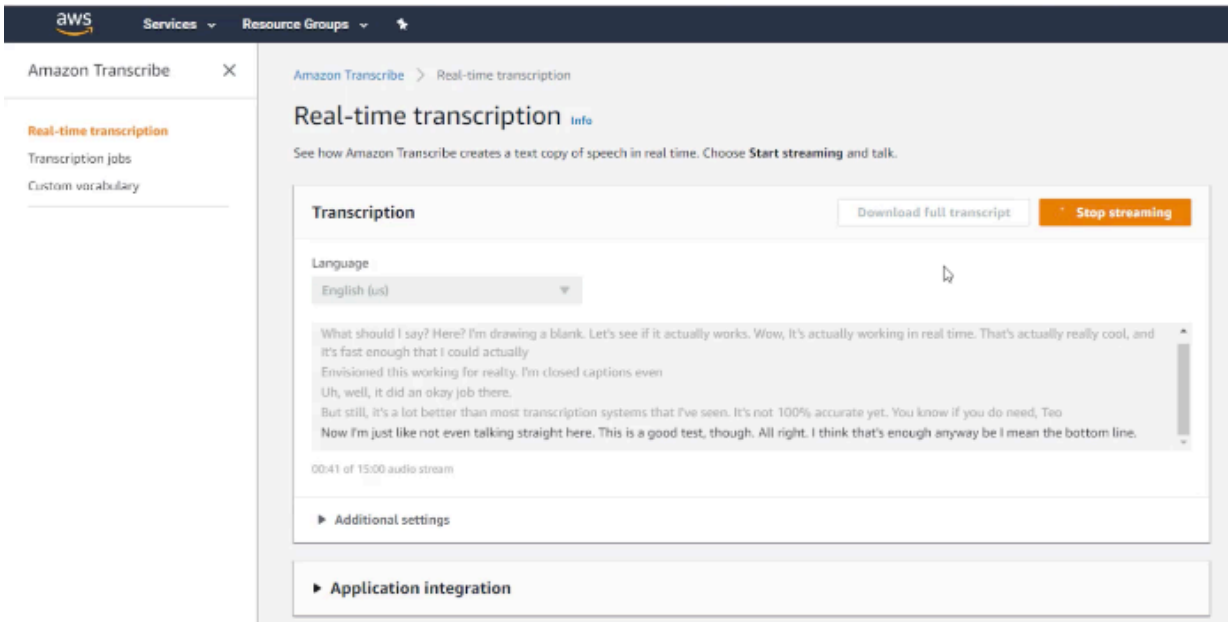
Amazon Translate



Amazon Transcribe

Amazon Transcribe

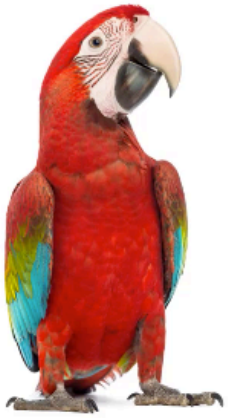
- Speech to text
  - Input in FLAC, MP3, MP4, or WAV, in a specified language
  - Streaming audio supported (HTTP/2 or WebSocket)
    - French, English, Spanish only
- Speaker Identificiation
  - Specify number of speakers
- Channel Identification
  - i.e., two callers could be transcribed separately
  - Merging based on timing of "utterances"
- Custom Vocabularies
  - Vocabulary Lists (just a list of special words – names, acronyms)
  - Vocabulary Tables (can include "SoundsLike", "IPA", and "DisplayAs")



Amazon Polly

Amazon Polly

- Neural Text-To-Speech, many voices & languages
- Lexicons
  - Customize pronunciation of specific words & phrases
  - Example: "World Wide Web Consortium" instead of "W3C"
- SSML
  - Alternative to plain text
  - Speech Synthesis Markup Language
  - Gives control over emphasis, pronunciation, breathing, whispering, speech rate, pitch, pauses.
- Speech Marks
  - Can encode when sentence / word starts and ends in the audio stream
  - Useful for lip-synching animation



Example of Lexicon

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa"
      xml:lang="en-US">
  <lexeme>
    <grapheme>pecan</grapheme>
    <phoneme>pɪˈkɑːn</phoneme>
  </lexeme>
</lexicon>
```

Other example:

```
  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>World Wide Web Consortium</alias>
  </lexeme>
  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>WWW Consortium</alias>
  </lexeme>
  <lexeme>
    <grapheme>Consortium</grapheme>
    <alias>Community</alias>
  </lexeme>
```

Example of SSML to add a pause

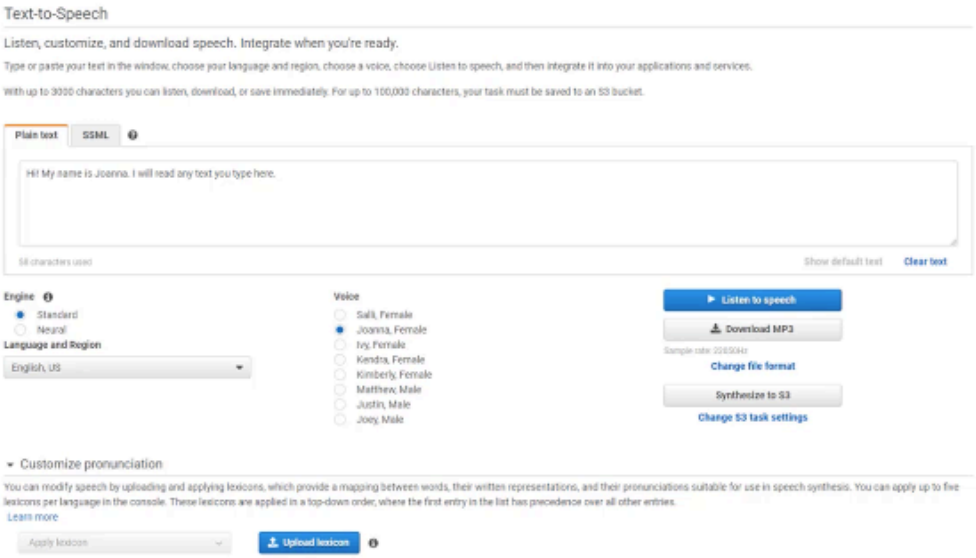
For example:

```
<speak>
     Mary had a little lamb <break time="3s"/>Whose fleece was white as snow.
</speak>
```

Or emphasize a word

```
<speak>
     I already told you I <emphasis level="strong">really like</emphasis> that person.
</speak>
```

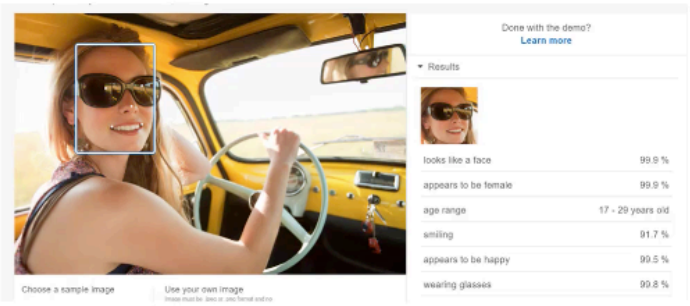
Amazon Polly



Amazon Rekognition

Rekognition

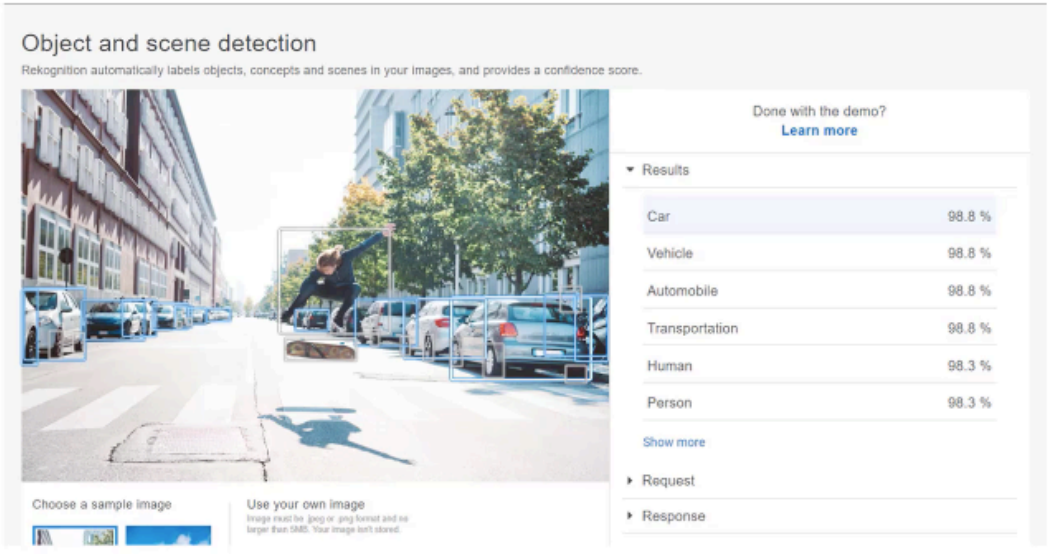
- Computer vision
- Object and scene detection
  - Can use your own face collection
- Image moderation
- Facial analysis
- Celebrity recognition
- Face comparison
- Text in image
- Video analysis
  - Objects / people / celebrities marked on timeline
  - People Pathing



Rekognition: The Nitty Gritty

- Images come from S3, or provide image bytes as part of request
  - S3 will be faster if the image is already there
- Facial recognition depends on good lighting, angle, visibility of eyes, resolution
- Video must come from Kinesis Video Streams
  - H.264 encoded
  - 5-30 FPS
  - Favor resolution over framerate
- Can use with Lambda to trigger image analysis upon upload

Rekognition



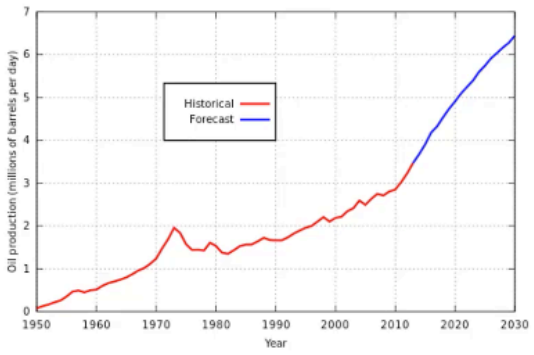
New in 2020: Rekognition Custom Labels

- Train with a small set of labeled images
- Use your own labels for unique items
- Example: the NFL (National Football League in the US) uses custom labels to identify team logos, pylons, and foam fingers in images.

Amazon Forecast

Amazon Forecast

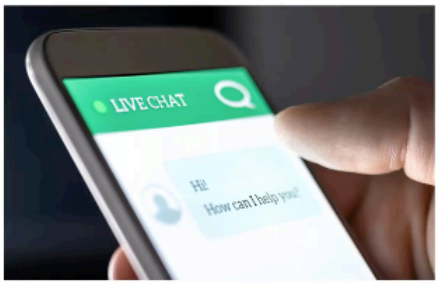
- Fully-managed service to deliver highly accurate forecasts with ML
- "AutoML" chooses best model for your time series data
  - ARIMA, DeepAR, ETS, NPTS, Prophet
- Works with any time series
  - Price, promotions, economic performance, etc.
  - Can combine with associated data to find relationships
- Inventory planning, financial planning, resource planning
- Based on "dataset groups," "predictors," and "forecasts."



Amazon Lex

Amazon Lex

- Billed as the inner workings of Alexa
- Natural-language chatbot engine
- A Bot is built around Intents
  - Utterances invoke intents ("I want to order a pizza")
  - Lambda functions are invoked to fulfill the intent
  - Slots specify extra information needed by the intent
    - Pizza size, toppings, crust type, when to deliver, etc.
- Can deploy to AWS Mobile SDK, Facebook Messenger, Slack, and Twilio



Utterance (phrase) ==> Invoke ==> **Intent** ==> Invoke ==> Lambda
The **Slots** provides more info for the Intent

Other ML Services

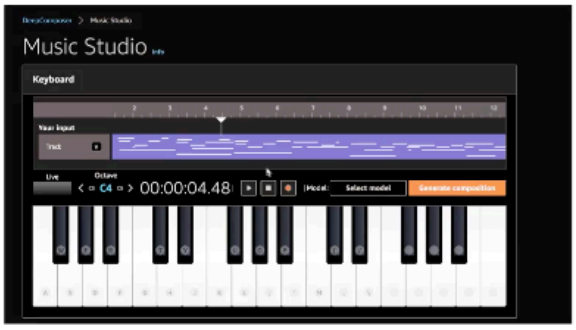
The Best of the Rest

- Amazon Personalize
  - Recommender system
- Amazon Textract
  - OCR with forms, fields, tables support
- AWS DeepRacer
  - Reinforcement learning powered 1/18-scale race car
- DeepLens
  - Deep learning-enabled video camera
  - Integrated with Rekognition, SageMaker, Polly, Tensorflow, MXNet, Caffe

New ML Services for 2020

AWS DeepComposer

- AI-powered keyboard
- Composes a melody into an entire song
- For educational purposes

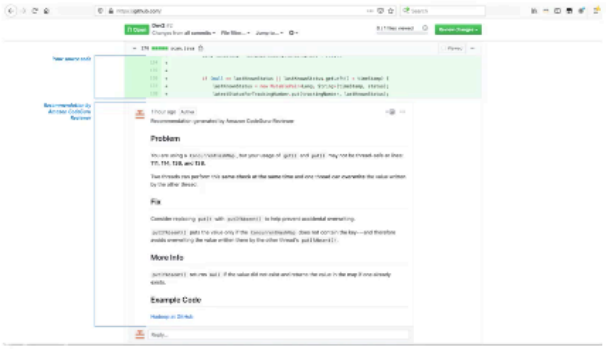


Amazon Fraud Detector

- Upload your own historical fraud data
- Builds custom models from a template you choose
- Exposes an API for your online application
- Assess risk from:
  - New accounts
  - Guest checkout
  - "Try before you buy" abuse
  - Online payments

Amazon CodeGuru

- Automated code reviews!
- Finds lines of code that hurt performance
- Resource leaks, race conditions
- Offers specific recommendations
- Powered by ML
- Currently Java-only (more coming soon)

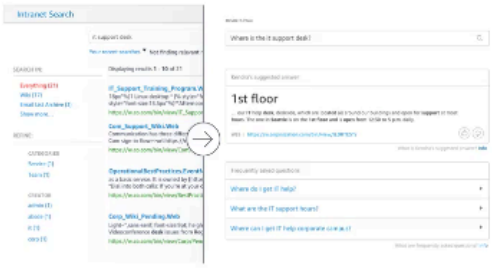


Contact Lens for Amazon Connect

- For customer support call centers
- Ingests audio data from recorded calls
- Allows search on calls / chats
- Sentiment analysis
- Find “utterances” that correlate with successful calls
- Categorize calls automatically
- Measure talk speed and interruptions
- Theme detection: discovers emerging issues

Amazon Kendra

- Enterprise search with natural language
- For example, “Where is the IT support desk?” “How do I connect to my VPN?”
- Combines data from file systems, SharePoint, intranet, sharing services (JDBC, S3) into one searchable repository
- ML-powered (of course) – uses thumbs up / down feedback
- Relevance tuning – boost strength of document freshness, view counts, etc.
- Alexa’s sister? I don’t know, but that’s one way to remember it ☺



Amazon Augmented AI (A2I)

- Human review of ML predictions
- Builds workflows for reviewing low-confidence predictions
- Access the Mechanical Turk workforce or vendors
- Integrated into Amazon Textract and Rekognition
- Integrates with SageMaker
- Very similar to Ground Truth

Putting these Services together

Putting the blocks together

- Build your own Alexa!
  - Transcribe -> Lex -> Polly
- Make a universal translator!
  - Transcribe -> Translate -> Polly
- Build a Jeff Bezos detector!
  - DeepLens -> Rekognition
- Are people on the phone happy?
  - Transcribe -> Comprehend

Quiz

Question 1:

Your deep neural network seems to converge on different solutions with different accuracy each time you train it. What's a likely explanation?

- ○ The learning rate is too small.
- ○ The batch size is too small
- ◉ The batch size is too large

Large batch sizes tend to get stuck, at random, inside "local minima" instead of the correct solution.

Question 2:

Your neural network's accuracy on its training data is increasing beyond the accuracy on test or validation data. What might be a valid thing to try to prevent this overfitting?

- (●) Use dropout
- ( ) Add more layers to the model
- ( ) Implement gradient checking

Dropout layers force the network to spread out its learning throughout the network, and can prevent overfitting resulting from learning concentrating in one spot. Early stopping would be another valid answer.

Question 3:

You're implementing a machine learning model for fraud detection, where most of your training data does not indicate fraud. The cost of a incorrectly identifying an actual fraudulent transaction is much higher than the cost of incorrectly identifying a non-fraudulent transaction. Which metric should you focus on for your model?

- ( ) Precision
- (●) Recall
- ( ) RMSE

Recall is appropriate when you care most about false negatives, which in this case is incorrectly identifying fraudulent transactions as non-fraudulent.

Question 4:

Where does the training code used by SageMaker come from?

- ○ Jupyter notebooks
- ◉ A Docker image registered with ECR
- ○ A GitHub repository

Whether it's your own code, a built-in algorithm from SageMaker, or a model you've purchased in the marketplace - all training code deployed to SageMaker training instances come from ECR.

Question 5:

Which SageMaker algorithm would be best suited for identifying topics in text documents in an unsupervised setting?

- ○ BlazingText can predict labels for sentences, but only if you've trained it in a supervised setting. It's not appropriate for working with entire documents.
- ○ Object2Vec
- ◉ LDA

Latent Dirichlet Allocation is a topic modeling technique. Neural Topic Model would also be a correct answer.