

Python for Machine Learning

UE19EC353

Department of ECE



People

Instructors:

1. Prof Niranjana B Krupa
2. Prof Raghavendra MJ

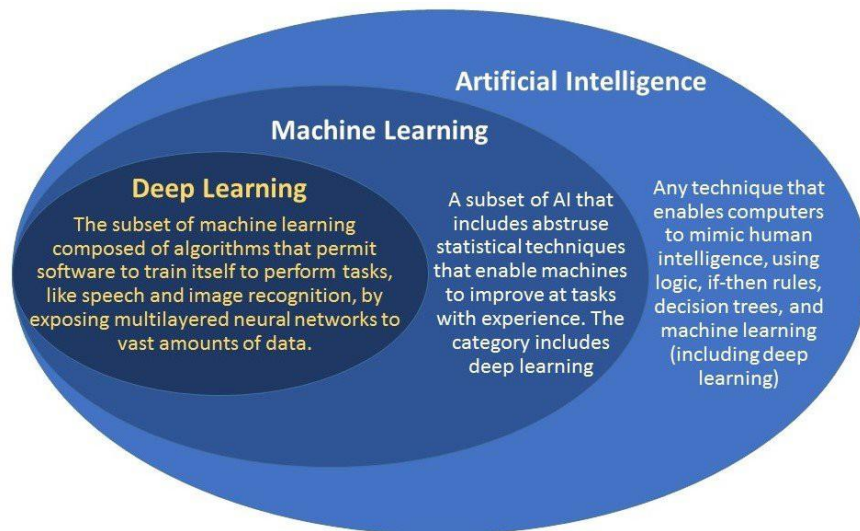
Teaching Assistant: K Venkat Ramnan

Course Description

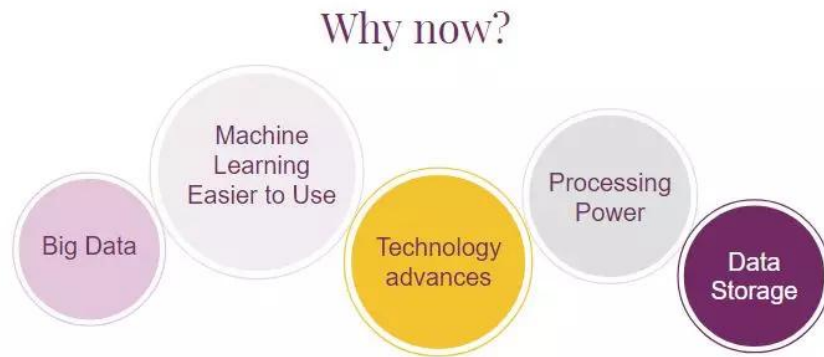
- Unit 1 : Foundations of Machine learning
- Unit 2 : Supervised Learning - Parametric methods
- Unit 3: Supervised Learning - Non Parametric methods
- Unit 4 : Kernel machines
- Unit 5: Clustering and Dimensionality Reduction

What is Machine Learning ?

- Branch of AI
- Data Analytics tool
- category of an algorithm
- use computational methods to “learn” information directly from data

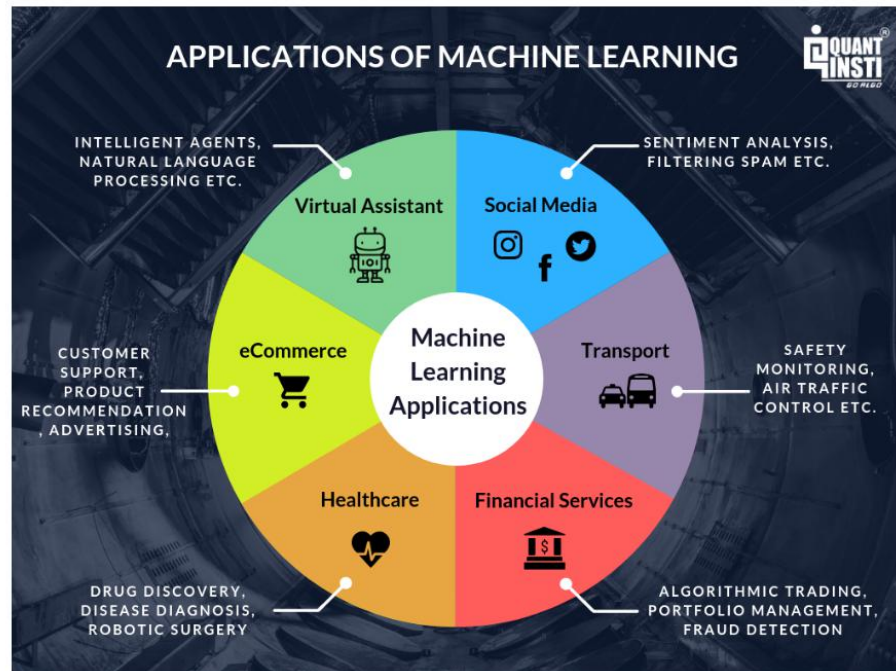
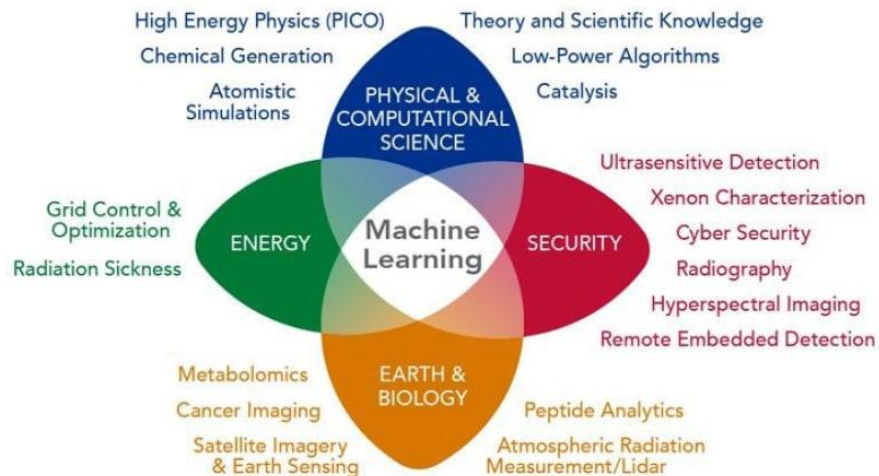


Why Machine Learning is booming?



- The field has matured both in terms of identity and in terms of methods and tools.
- abundance of data to learn from
- abundance of computation to run methods

Where is Machine Learning used?



Setting up your PC for ML

Installing packages using pip

Considering that you have installed python:

```
Anaconda Prompt (Anaconda3)

(base) C:\Users\venka>pip install numpy
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/22/a3/3a5469ebaca59100e50b4300dd011eed943f2aad7c6a80a07966b985e2c6/numpy-1.21.5-cp37-cp37m-win_amd64.whl (14.0MB)
    14.0MB 2.2MB/s
ERROR: tensorflow-gpu 2.3.0 has requirement numpy<1.19.0,>=1.16.0, but you'll have numpy 1.21.5 which is incompatible.
Installing collected packages: numpy
Successfully installed numpy-1.21.5

(base) C:\Users\venka>pip uninstall numpy
Uninstalling numpy-1.21.5:
  Would remove:
    c:\users\venka\anaconda3\lib\site-packages\numpy-1.21.5.dist-info\*
    c:\users\venka\anaconda3\lib\site-packages\numpy\*
    c:\users\venka\anaconda3\scripts\f2py.exe
Proceed (y/n)? y
  Successfully uninstalled numpy-1.21.5

(base) C:\Users\venka>pip uninstall numpy==1.18.0
WARNING: Skipping numpy as it is not installed.

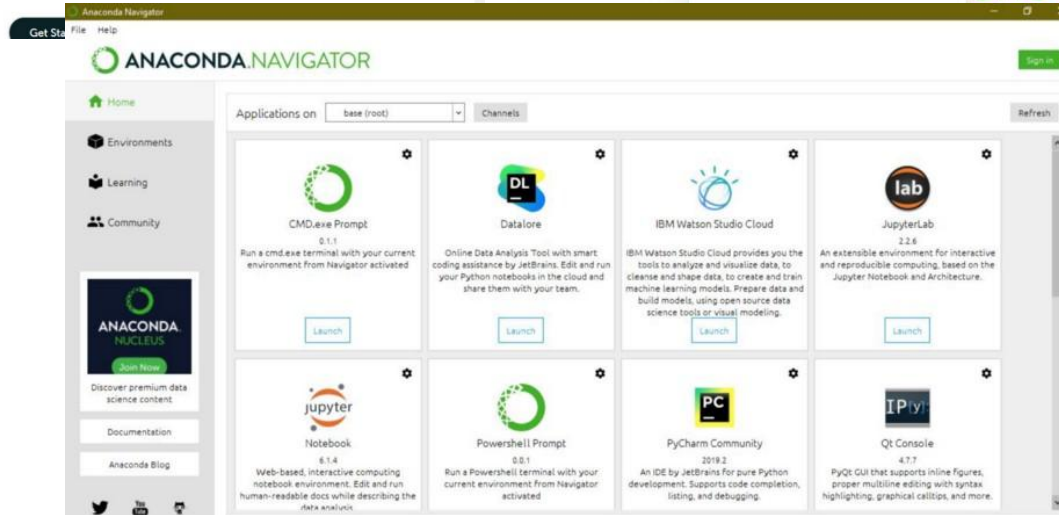
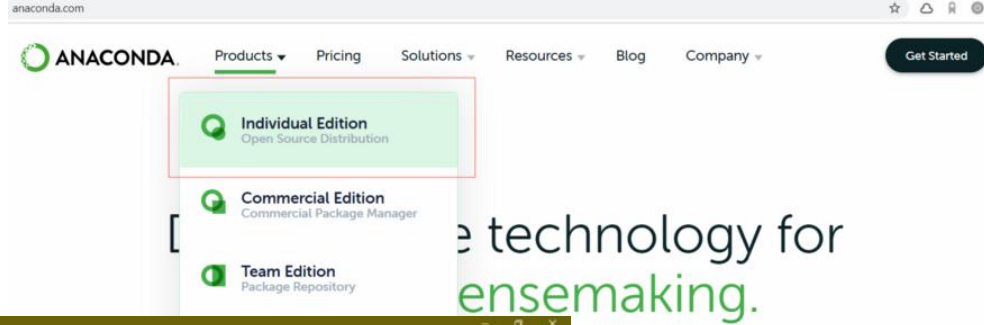
(base) C:\Users\venka>pip install numpy==1.18.0
Collecting numpy==1.18.0
  Downloading https://files.pythonhosted.org/packages/e9/0f/b1aa961980d5ea94243f28f91d3f6fc6f3b7e5047a9b8dc037541c2cc11/numpy-1.18.0-cp37-cp37m-win_amd64.whl (12.8MB)
    12.8MB 6.8MB/s
Installing collected packages: numpy
Successfully installed numpy-1.18.0

(base) C:\Users\venka>
```


Anaconda Navigator

Data science technology for
human sensemaking.

A movement that brings together millions of data science practitioners,
data-driven enterprises, and the open source community.



f data science practitioners,
i source community.

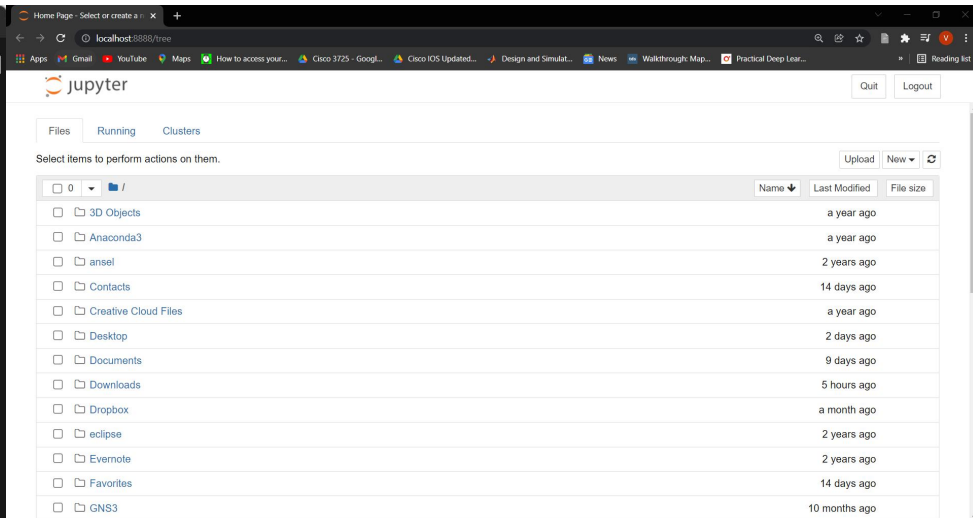


Jupyter notebook

Anaconda Prompt (Anaconda3) - jupyter notebook

```
(base) C:\Users\venka>jupyter notebook
[I 22:42:45.855 NotebookApp] JupyterLab extension loaded from C:\Users\venka\Anaconda3\lib\site-packages\jupyterlab
[I 22:42:45.855 NotebookApp] JupyterLab application directory is C:\Users\venka\Anaconda3\share\jupyter\lab
[I 22:42:45.857 NotebookApp] Serving notebooks from local directory: C:\Users\venka
[I 22:42:45.858 NotebookApp] The Jupyter Notebook is running at:
[I 22:42:45.858 NotebookApp] http://localhost:8888/?token=0041301ef574f146cb8194f22e521ce477195d6f6c0b6033
[I 22:42:45.858 NotebookApp] or http://127.0.0.1:8888/?token=0041301ef574f146cb8194f22e521ce477195d6f6c0b6033
[I 22:42:45.858 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 22:42:45.921 NotebookApp]
```

```
To access the notebook, open this file in a browser:
file:///C:/Users/venka/AppData/Roaming/jupyter/runtime/nbserver-5500-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=0041301ef574f146cb8194f22e521ce477195d6f6c0b6033
or http://127.0.0.1:8888/?token=0041301ef574f146cb8194f22e521ce477195d6f6c0b6033
[E 22:42:47.394 NotebookApp] Could not open static file ''
[W 22:42:47.456 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (::1) 11.97ms referer=http://localhost:8888/tree?token=0041301ef574f146cb8194f22e521ce477195d6f6c0b6033
[W 22:42:47.468 NotebookApp] 404 GET /static/components/react/react-dom.production.min.js (::1) 1.00ms referer=http://localhost:8888/tree?token=0041301ef574f146cb8194f22e521ce477195d6f6c0b6033
```



Home Page - Select or create a notebook

localhost:8888/tree

jupyter

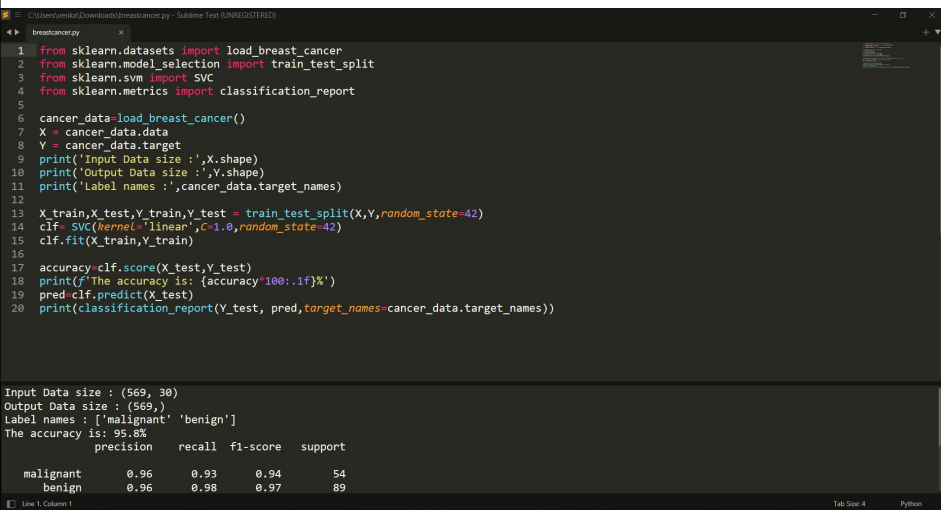
Files Running Clusters

Select items to perform actions on them.

Upload New

	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	3D Objects	a year ago	
<input type="checkbox"/>	Anaconda3	a year ago	
<input type="checkbox"/>	ansel	2 years ago	
<input type="checkbox"/>	Contacts	14 days ago	
<input type="checkbox"/>	Creative Cloud Files	a year ago	
<input type="checkbox"/>	Desktop	2 days ago	
<input type="checkbox"/>	Documents	9 days ago	
<input type="checkbox"/>	Downloads	5 hours ago	
<input type="checkbox"/>	Dropbox	a month ago	
<input type="checkbox"/>	eclipse	2 years ago	
<input type="checkbox"/>	Evernote	2 years ago	
<input type="checkbox"/>	Favorites	14 days ago	
<input type="checkbox"/>	GNS3	10 months ago	

Other cool code editors

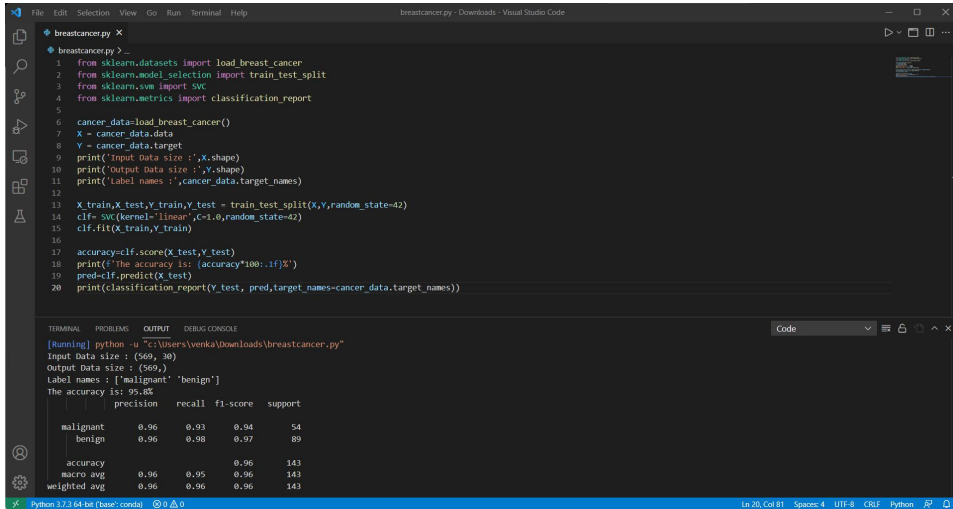


```

1 from sklearn.datasets import load_breast_cancer
2 from sklearn.model_selection import train_test_split
3 from sklearn.svm import SVC
4 from sklearn.metrics import classification_report
5
6 cancer_data=load_breast_cancer()
7 X = cancer_data.data
8 Y = cancer_data.target
9 print('Input Data size :',X.shape)
10 print('Output Data size :',Y.shape)
11 print('Label names : ',cancer_data.target_names)
12
13 X_train,X_test,Y_train,Y_test = train_test_split(X,Y,random_state=42)
14 clf= SVC(kernel='linear',C=1.0,random_state=42)
15 clf.fit(X_train,Y_train)
16
17 accuracy=clf.score(X_test,Y_test)
18 print(f'The accuracy is: {accuracy*100:.1f}%')
19 pred=clf.predict(X_test)
20 print(classification_report(Y_test, pred,target_names=cancer_data.target_names))
  
```

Input Data size : (569, 30)
 Output Data size : (569,)
 Label names : ['malignant' 'benign']
 The accuracy is: 95.8%

	precision	recall	f1-score	support
malignant	0.96	0.93	0.94	54
benign	0.96	0.98	0.97	89



```

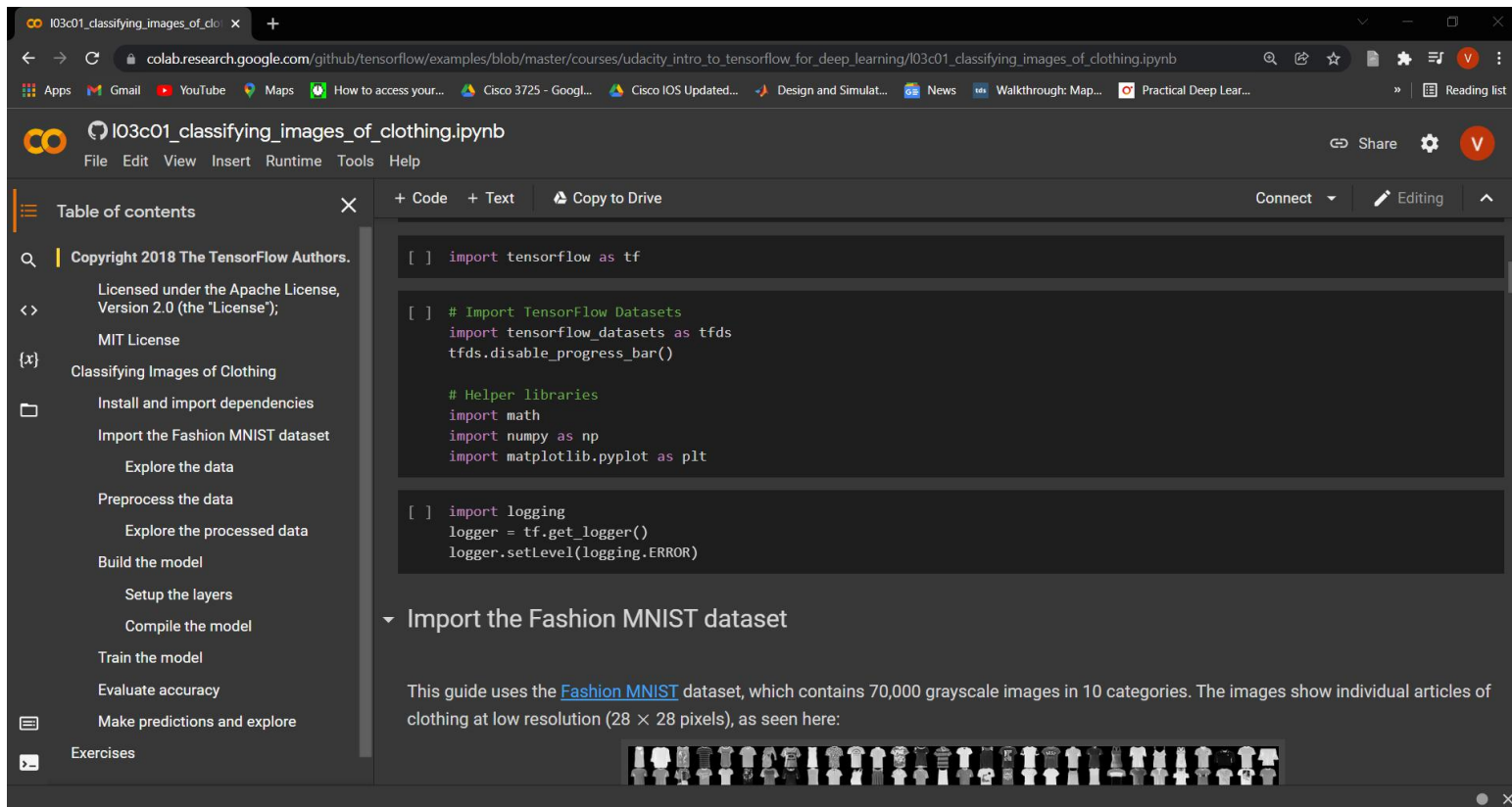
1 from sklearn.datasets import load_breast_cancer
2 from sklearn.model_selection import train_test_split
3 from sklearn.svm import SVC
4 from sklearn.metrics import classification_report
5
6 cancer_data=load_breast_cancer()
7 X = cancer_data.data
8 Y = cancer_data.target
9 print('Input Data size :',X.shape)
10 print('Output Data size :',Y.shape)
11 print('Label names : ',cancer_data.target_names)
12
13 X_train,X_test,Y_train,Y_test = train_test_split(X,Y,random_state=42)
14 clf= SVC(kernel='linear',C=1.0,random_state=42)
15 clf.fit(X_train,Y_train)
16
17 accuracy=clf.score(X_test,Y_test)
18 print(f'The accuracy is: {accuracy*100:.1f}%')
19 pred=clf.predict(X_test)
20 print(classification_report(Y_test, pred,target_names=cancer_data.target_names))
  
```

[Running] python -u "C:\Users\venka\Downloads\breastcancer.py"
 Input Data size : (569, 30)
 Output Data size : (569,)
 Label names : ['malignant' 'benign']
 The accuracy is: 95.8%

	precision	recall	f1-score	support
malignant	0.96	0.93	0.94	54
benign	0.96	0.98	0.97	89
accuracy			0.96	143
macro avg	0.96	0.95	0.96	143
weighted avg	0.96	0.96	0.96	143

Hmm.. Out of RAM and GPU?

Google Colab is the answer!!



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: `colab.research.google.com/github/tensorflow/examples/blob/master/courses/udacity_intro_to_tensorflow_for_deep_learning/I03c01_classifying_images_of_clothing.ipynb`. The notebook title is `I03c01_classifying_images_of_clothing.ipynb`. The left sidebar contains a 'Table of contents' with the following items: Copyright 2018 The TensorFlow Authors., Licensed under the Apache License, Version 2.0 (the "License"); MIT License, Classifying Images of Clothing, Install and import dependencies, Import the Fashion MNIST dataset, Explore the data, Preprocess the data, Explore the processed data, Build the model, Setup the layers, Compile the model, Train the model, Evaluate accuracy, Make predictions and explore, and Exercises. The main code area shows the following Python code:

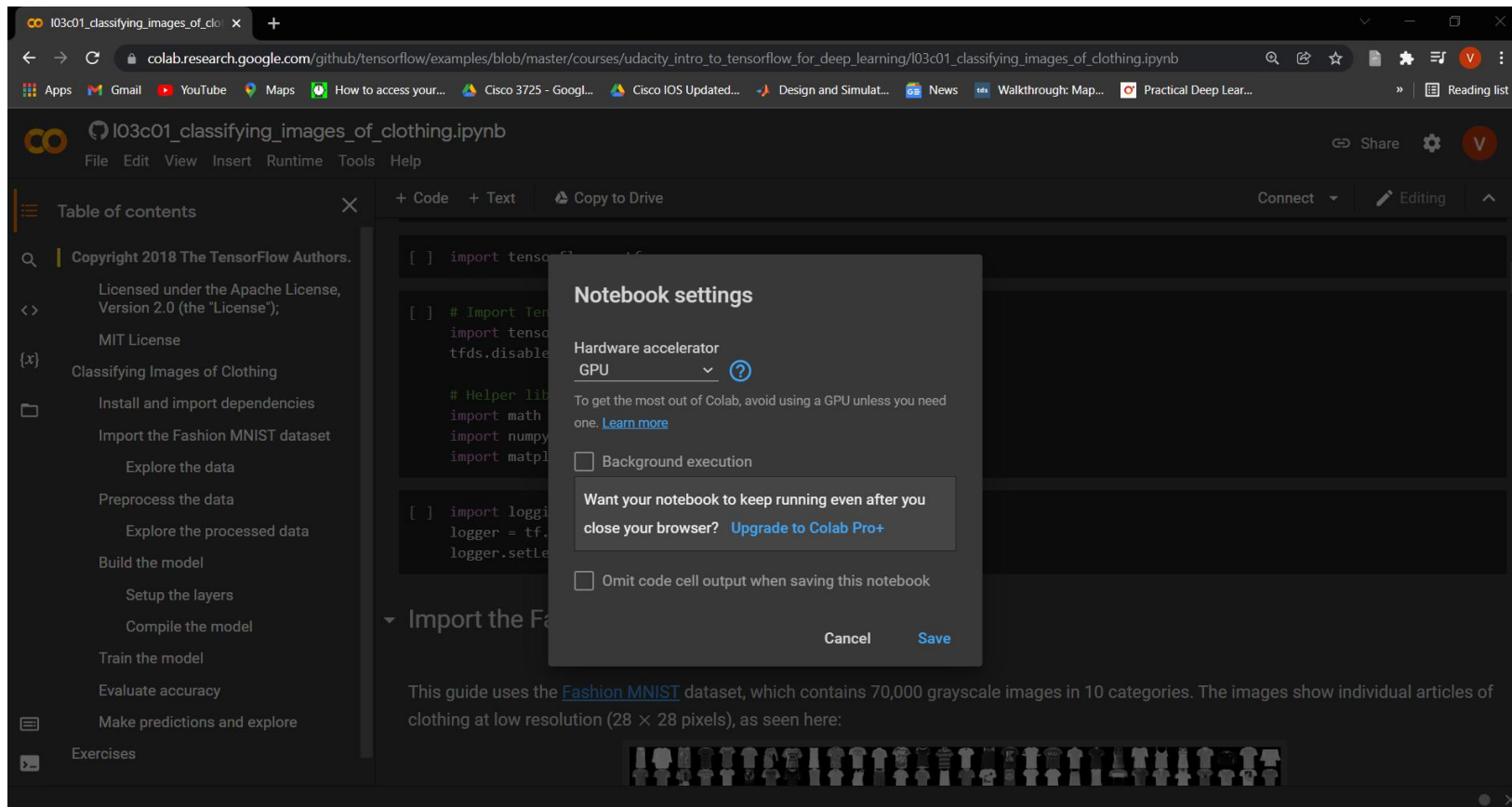
```
[ ] import tensorflow as tf

[ ] # Import TensorFlow Datasets
import tensorflow_datasets as tfds
tfds.disable_progress_bar()

# Helper libraries
import math
import numpy as np
import matplotlib.pyplot as plt

[ ] import logging
logger = tf.get_logger()
logger.setLevel(logging.ERROR)
```

Below the code, there is a section titled 'Import the Fashion MNIST dataset' with the following text: 'This guide uses the [Fashion MNIST](#) dataset, which contains 70,000 grayscale images in 10 categories. The images show individual articles of clothing at low resolution (28 × 28 pixels), as seen here:'. Below the text is a row of 28 small grayscale images of various clothing items.



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: `colab.research.google.com/github/tensorflow/examples/blob/master/courses/udacity_intro_to_tensorflow_for_deep_learning/I03c01_classifying_images_of_clothing.ipynb`. The notebook title is `I03c01_classifying_images_of_clothing.ipynb`. The left sidebar contains a 'Table of contents' with the following items: Copyright 2018 The TensorFlow Authors. Licensed under the Apache License, Version 2.0 (the "License"); MIT License; Classifying Images of Clothing; Install and import dependencies; Import the Fashion MNIST dataset; Explore the data; Preprocess the data; Explore the processed data; Build the model; Setup the layers; Compile the model; Train the model; Evaluate accuracy; Make predictions and explore; Exercises.

The main notebook area shows code cells. The first cell contains `import tensorflow as tf`. The second cell contains `# Import TensorFlow helper libraries`, `import tensorflow as tf`, `tf.disable_eager_execution()`, `# Helper libraries`, `import math`, `import numpy as np`, and `import matplotlib.pyplot as plt`. The third cell contains `import logging`, `logger = tf.get_logger()`, and `logger.setLevel(logging.INFO)`.

A 'Notebook settings' dialog box is open, showing the following options:

- Hardware accelerator: GPU (selected)
- To get the most out of Colab, avoid using a GPU unless you need one. [Learn more](#)
- ☐ Background execution
- ☐ Want your notebook to keep running even after you close your browser? [Upgrade to Colab Pro+](#)
- ☐ Omit code cell output when saving this notebook

The dialog box has 'Cancel' and 'Save' buttons. The 'Save' button is highlighted in blue.

Below the code cells, the text reads: 'This guide uses the [Fashion MNIST](#) dataset, which contains 70,000 grayscale images in 10 categories. The images show individual articles of clothing at low resolution (28 × 28 pixels), as seen here:'. Below this text is a row of 28 small grayscale images of clothing items.

Research and Research!!!

image classification - Google Sch

+

←

→

↺

scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=image+classification&btnG=&oq=image+class

🔖

☆

📄

⚙️

📖

V

⋮

Apps

Gmail

YouTube

Maps

How to access your...

Cisco 3725 - Googl...

Cisco IOS Updated...

Design and Simulat...

News

Walkthrough: Map...

Practical Deep Lear...

»

📖

Reading list

☰

Google Scholar

image classification

🔍

👤

📁

Articles

About 50,00,000 results (0.04 sec)

👤 My profile

★ My library

Any time

Since 2021

Since 2020

Since 2017

Custom range...

Sort by relevance

Sort by date

Any type

Review articles

☐ include patents

☒ include citations

☒ Create alert

A survey of **image classification** methods and techniques for improving **classification** performance

D.Lu, Q.Weng

- International journal of Remote sensing, 2007 - Taylor & Francis

Image classification is a complex process that may be affected by many factors. This paper examines current practices, problems, and prospects of **image classification**. The emphasis is placed on the summarization of major advanced **classification** approaches and the ...

☆ Save

🔗 Cite

Cited by 3140

Related articles

All 7 versions

[PDF] tandfonline.com

Textural features for **image classification**

RM.Haralick, K Shanmugam...

- IEEE Transactions on ..., 1973 - ieexplore.ieee.org

Texture is one of the important characteristics used in identifying objects or regions of interest in an **image**, whether the **image** be a photomicrograph, an aerial photograph, or a satellite **image**. This paper describes some easily computable textural features based on ...

☆ Save

🔗 Cite

Cited by 25841

Related articles

All 22 versions

[PDF] ieee.org

Indoor-outdoor **image classification**

M.Szummer, RW.Picard

- ... on Content-Based Access of **Image** ..., 1998 - ieexplore.ieee.org

We show how high-level scene properties can be inferred from **classification** of low-level **image** features, specifically for the indoor-outdoor scene retrieval problem. We systematically studied the features of: histograms in the Ohta color space; multiresolution ...

☆ Save

🔗 Cite

Cited by 999

Related articles

All 12 versions

[PDF] ieee.org

Image classification for content-based indexing

A.Vailaya, MAT.Figueiredo, AK.Jain...

- ... transactions on **image** ..., 2001 - ieexplore.ieee.org

Grouping images into (semantically) meaningful categories using low-level visual features is a challenging and important problem in content-based **image** retrieval. Using binary Bayesian classifiers, we attempt to capture high-level concepts from low-level **image** ...

☆ Save

🔗 Cite

Cited by 1121

Related articles

All 16 versions

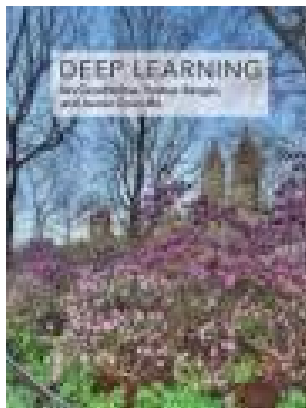
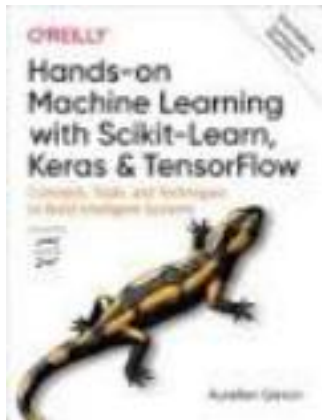
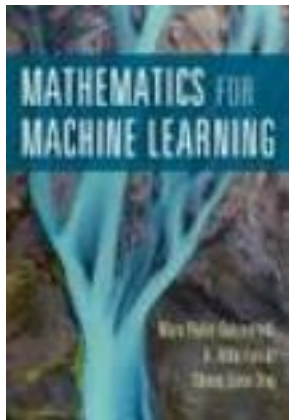
[PDF] ieee.org

Related searches

deep image classification

image classification features

Some amazing resources on ML



Agenda

- Day 1 : Introduction. Starting off with Numpy
- Day 2 : Tabular Data with Pandas; Visualization with matplotlib
- Day 3: Learning Scikit Learn; Into images with OpenCV
- Day 4: Going Deep with TF