



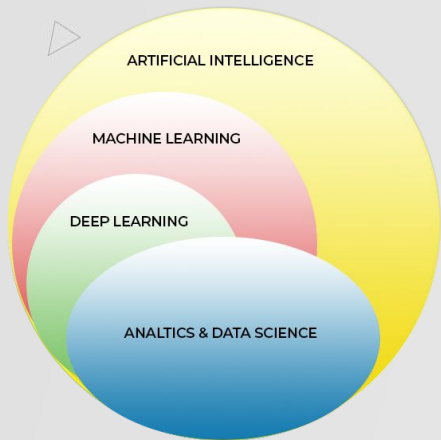
CSCRS

UHUZAM WEBINAR -1

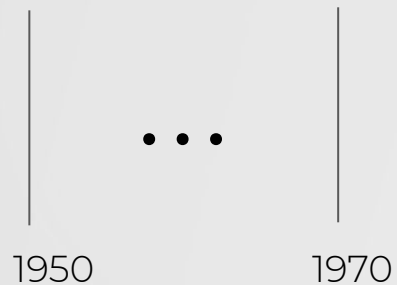
Basics - General Introduction

01

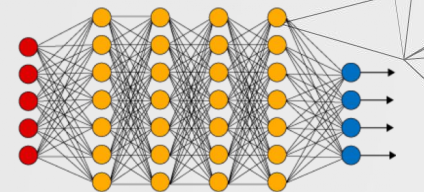
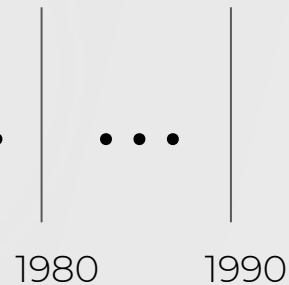
AI - DL - ML - Data Terms



Artificial Neural Network

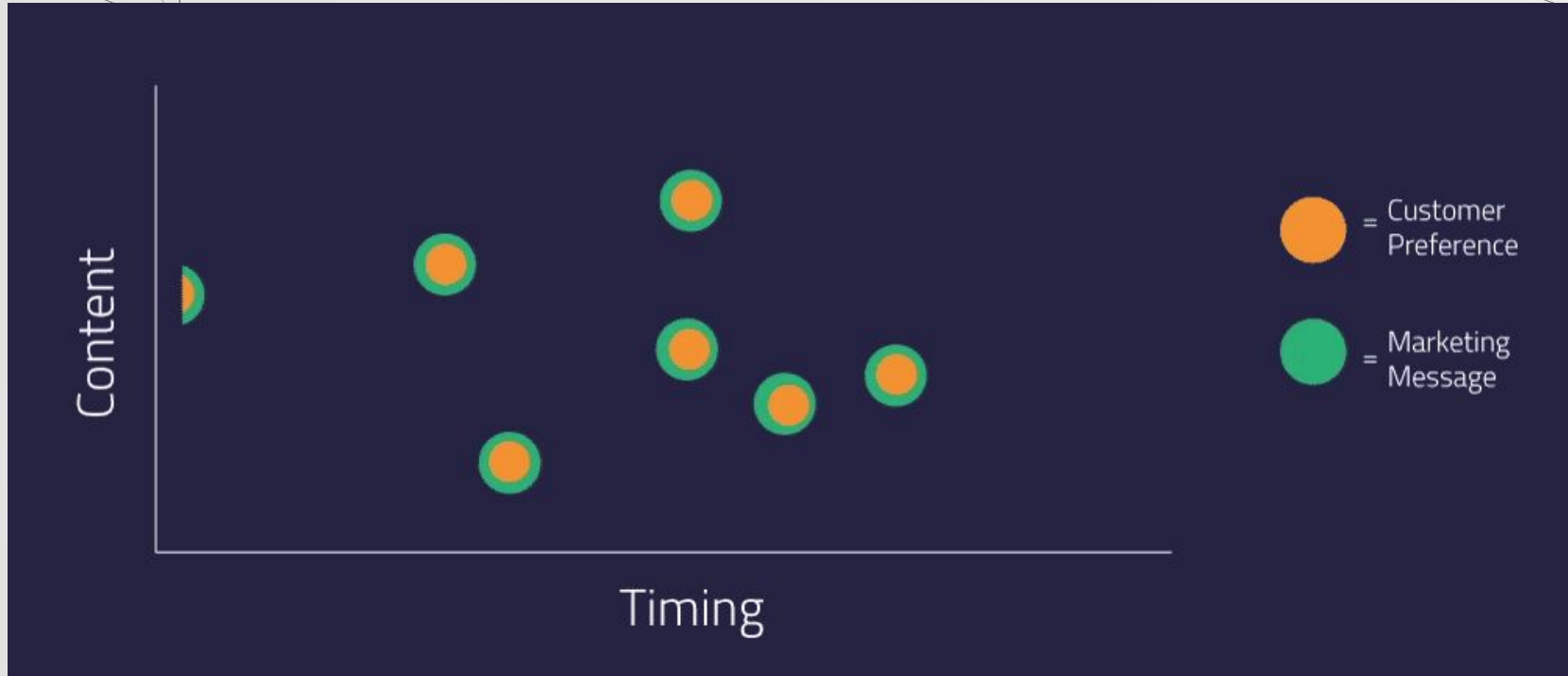


Machine Learning



Deep Learning

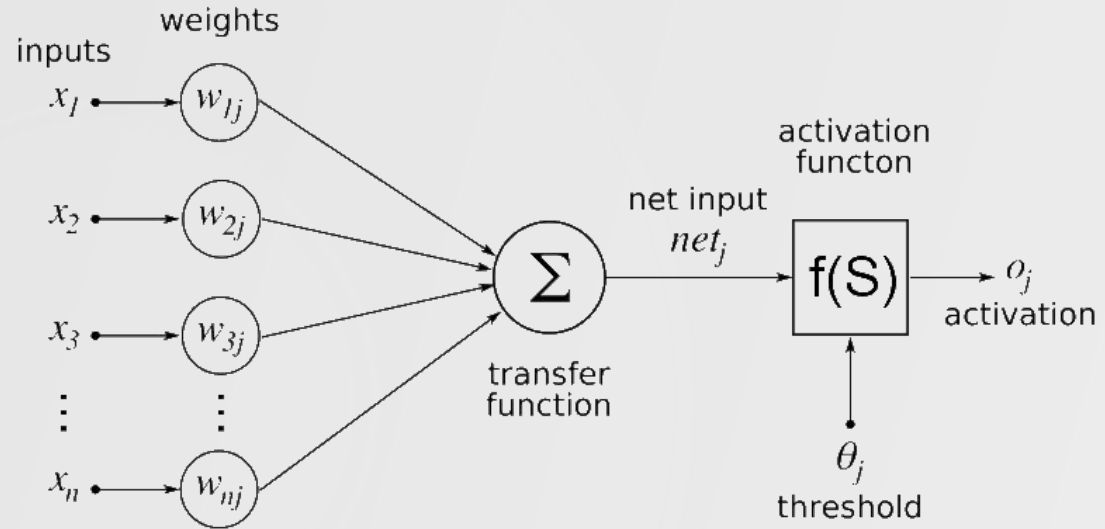




Artificial Neuron

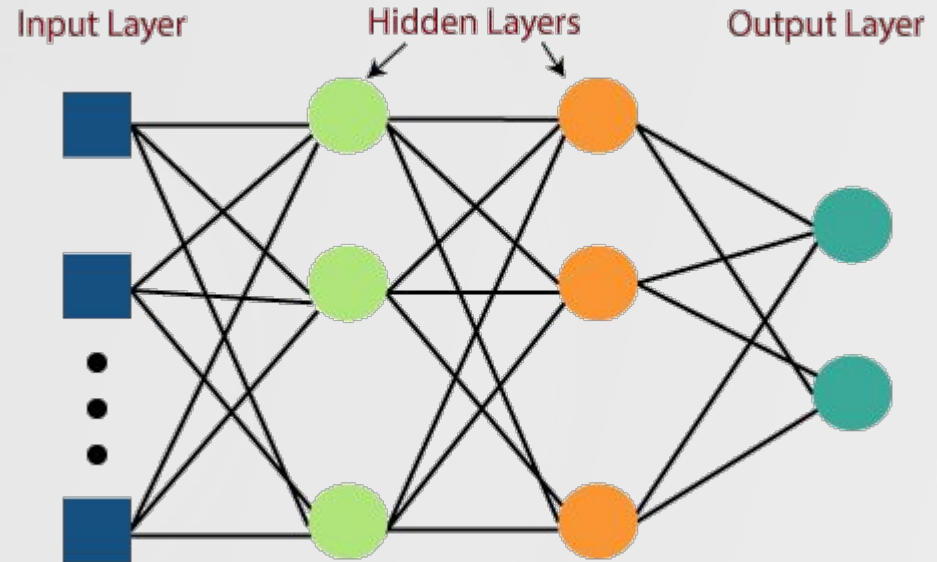
$$y = f \left(\sum_i w_i x_i \right)$$

x_i : inputs
 w_i : weights
 f : non-linearity



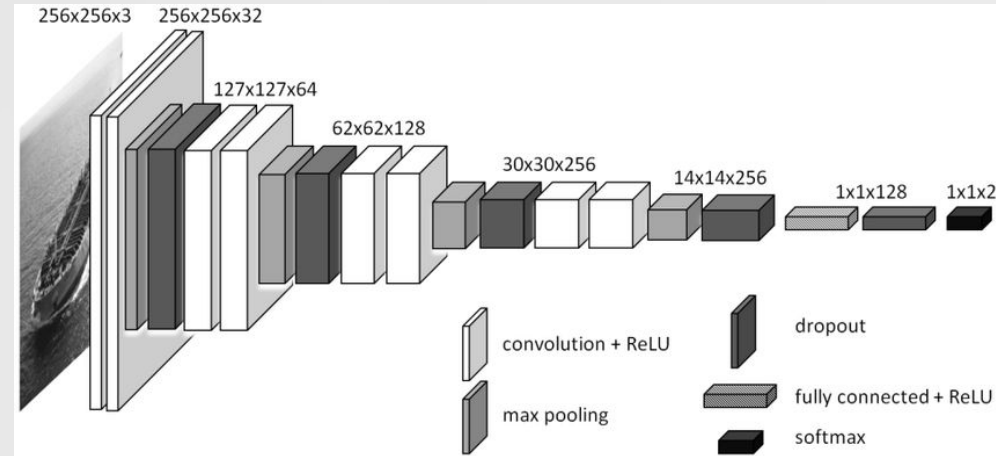
Multilayer Perception

Organization in layers

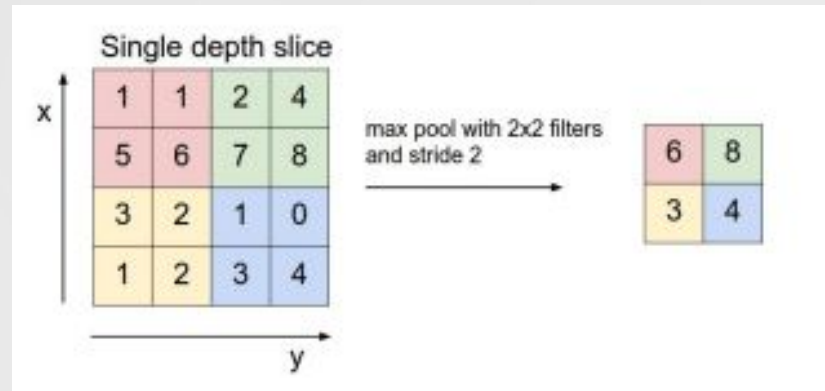


Types of Deep Learning Algorithms

- Convolutional Neural Networks (CNNs)
- Long Short Term Memory Networks (LSTMs)
- Recurrent Neural Networks (RNNs)
- Generative Adversarial Networks (GANs)
- Radial Basis Function Networks (RBFNs)
- Multilayer Perceptrons (MLPs)
- Self Organizing Maps (SOMs)
- Deep Belief Networks (DBNs)
- Restricted Boltzmann Machines(RBMs)



- Convolutional Layer
- Non-Linearity Layer
- Pooling (Downsampling Layer)
- Flattening Layer
- Fully-Connected Layer

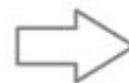


1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1



1	1	0
4	2	1
0	2	1



1
1
0
4
2
1
0
2
1



Input



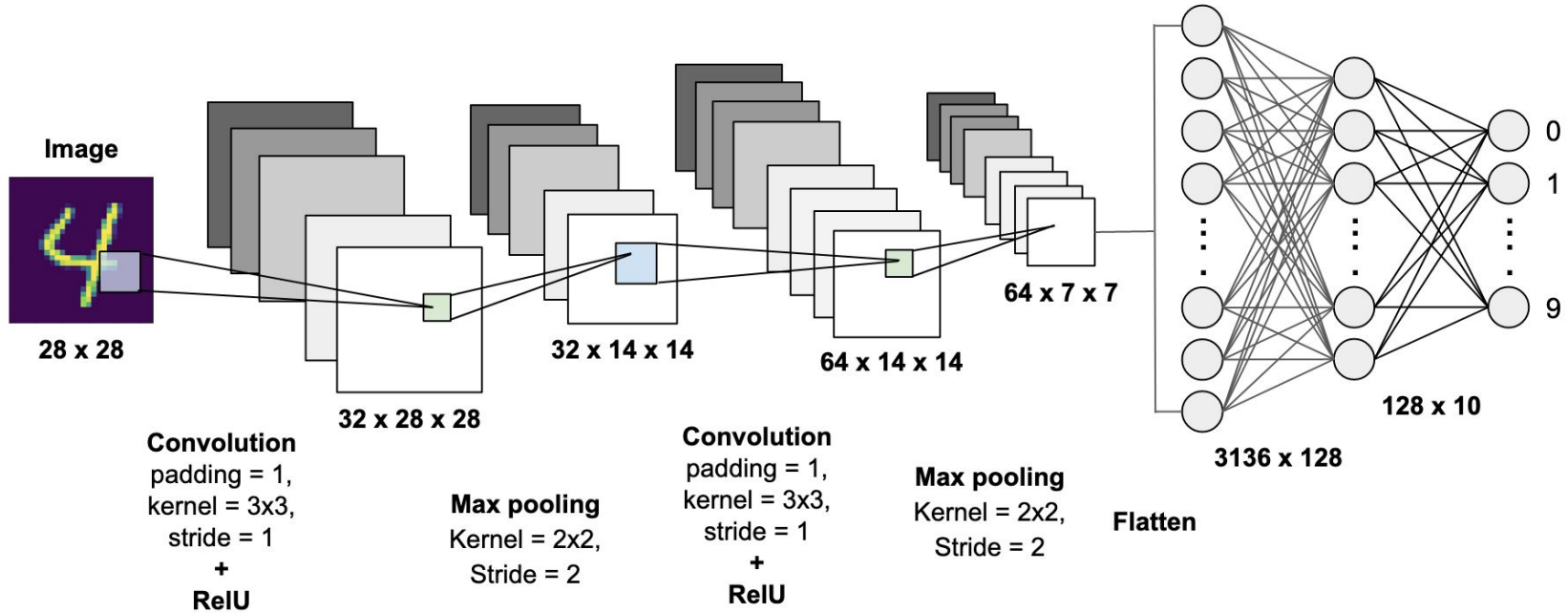
Original Image



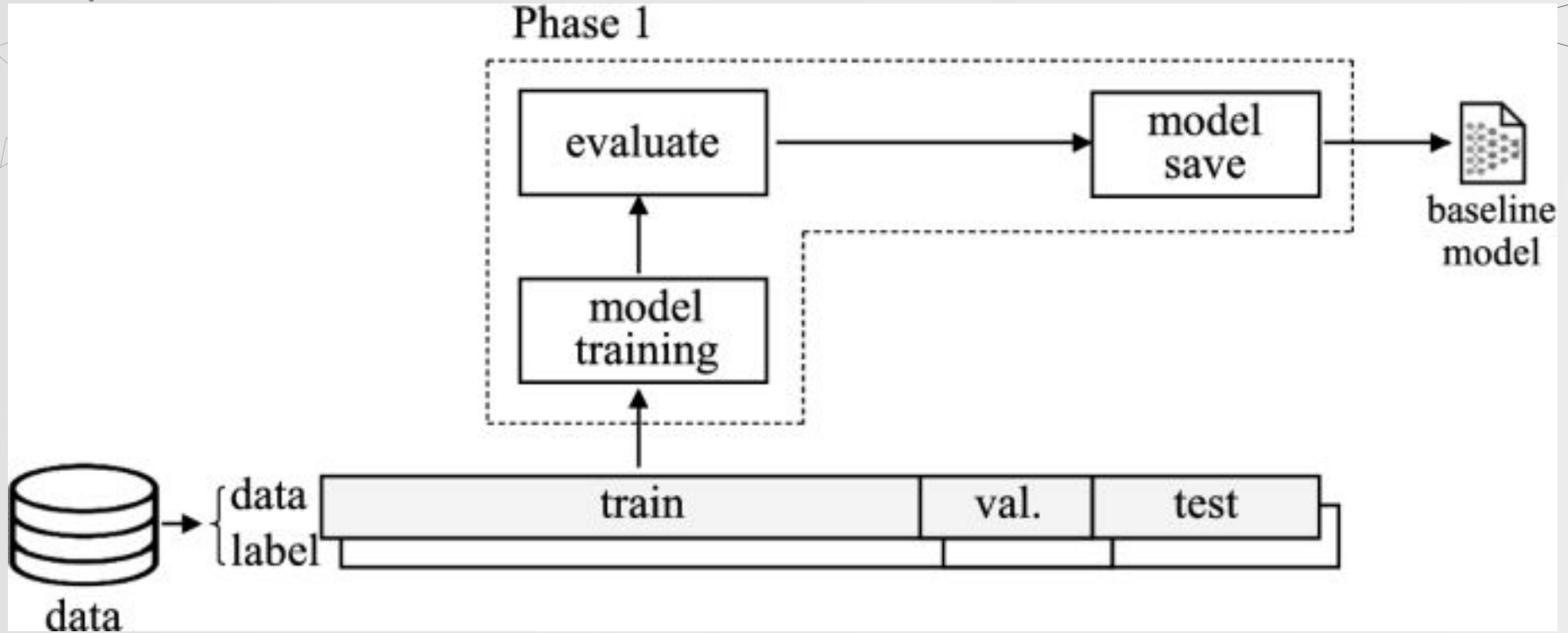
Feature Map



Non-Linear



Create General Baseline Model

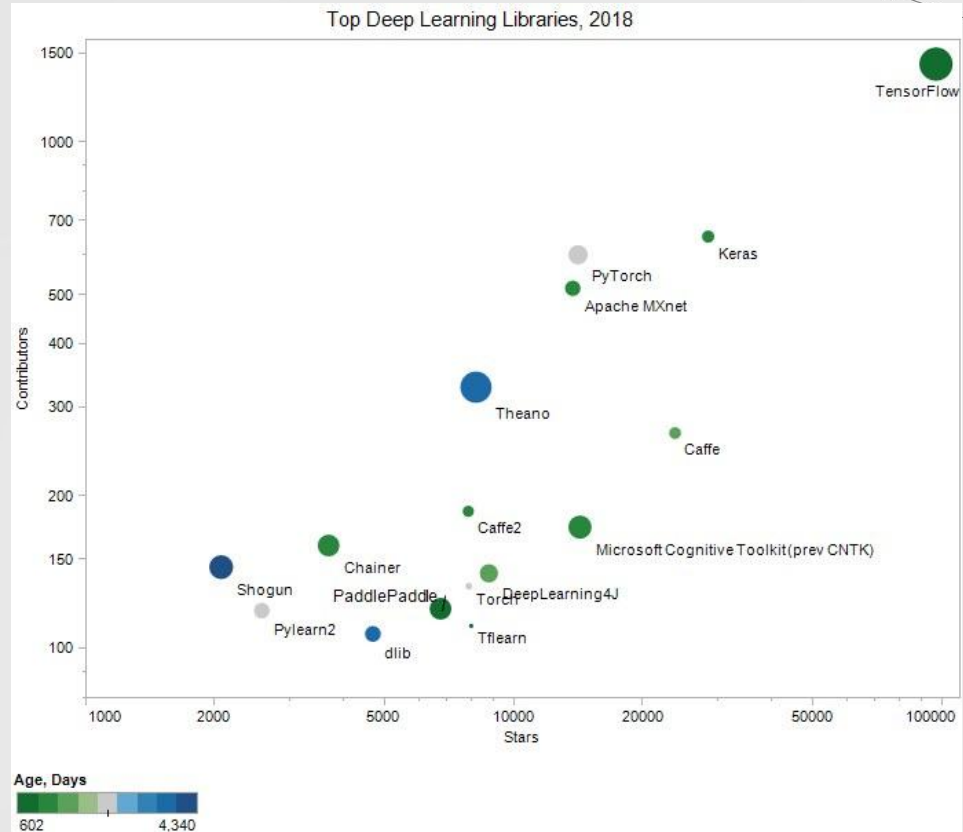


02

DL Frameworks and Libraries

Some DL frameworks

- Tensorflow
- Keras
- Pytorch
- MXNet
- Theano
- Caffe



Some Most Popular and Useful Python Libraries

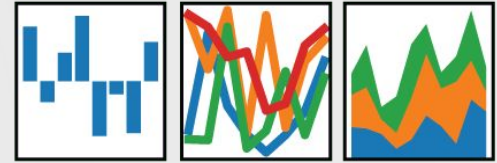


- Scikit-learn
- Pandas
- Numpy
- OpenCV
- Matplotlib



pandas

$$y_i t = \beta' x_{it} + \mu_i + \epsilon_{it}$$



NumPy



matplotlib

Numpy



The fundamental package for scientific computing with Python

```
import numpy as np

x = np.array([[1,2],[3,4]])
y = np.array([[5,6],[7,8]])

v = np.array([9,10])
w = np.array([11, 12])

# Inner product of vectors; both produce 219
print(v.dot(w))
print(np.dot(v, w))

# Matrix / vector product; both produce the rank 1 array [29 67]
print(x.dot(v))
print(np.dot(x, v))

# Matrix / matrix product; both produce the rank 2 array
# [[19 22]
#  [43 50]]
print(x.dot(y))
print(np.dot(x, y))
```



NumPy

Pandas



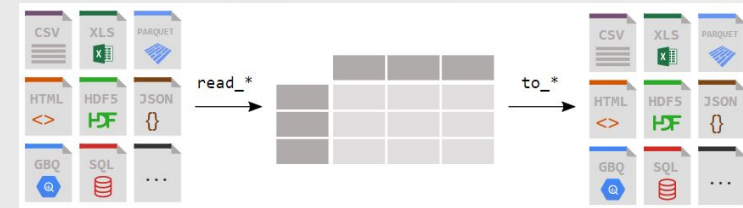
Pandas; open source data analysis and manipulation tool

```
In [2]: titanic = pd.read_csv("data/titanic.csv")
```

```
In [3]: titanic
Out[3]:
```

	PassengerId	Survived	Pclass	...	Fare	Cabin	Embarked
0	1	0	3	...	7.2500	NaN	S
1	2	1	1	...	71.2833	C85	C
2	3	1	3	...	7.9250	NaN	S
3	4	1	1	...	53.1000	C123	S
4	5	0	3	...	8.0500	NaN	S
..
886	887	0	2	...	13.0000	NaN	S
887	888	1	1	...	30.0000	B42	S
888	889	0	3	...	23.4500	NaN	S
889	890	1	1	...	30.0000	C148	C
890	891	0	3	...	7.7500	NaN	Q

[891 rows x 12 columns]



Scikit-learn



- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various context
- Built on Numpy, SciPy, and Matplotlib

```
$ python
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
```

```
>>> print(digits.data)
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0. 10. ... 12.  1.  0.]]
```

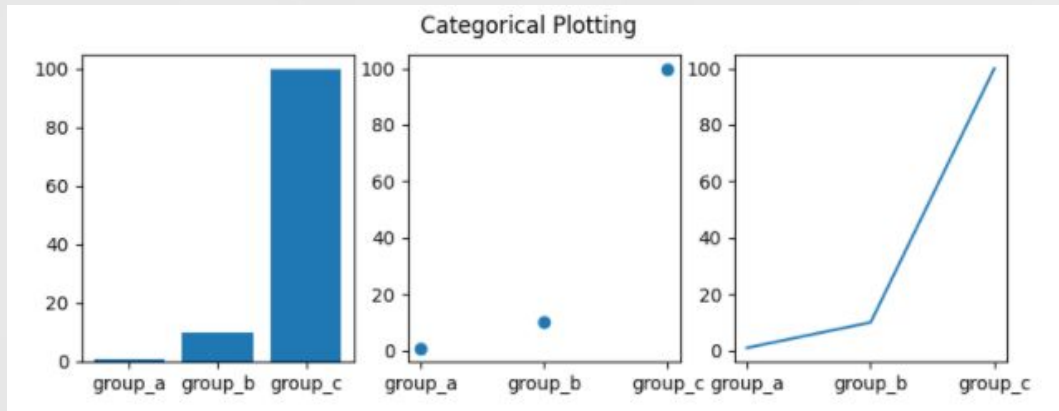
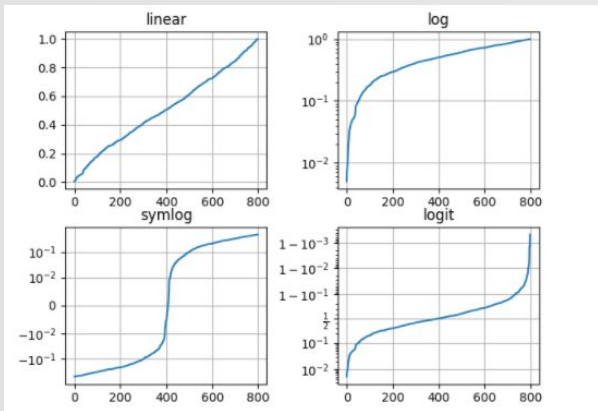
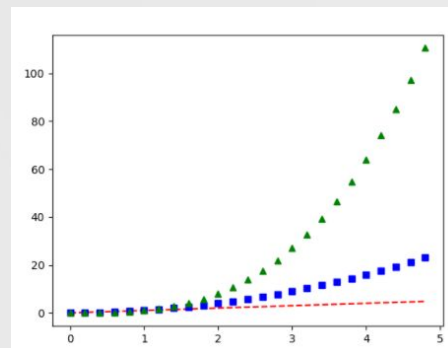
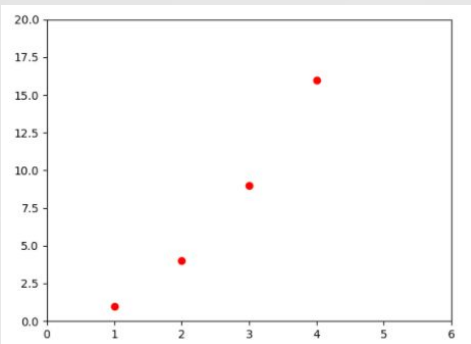
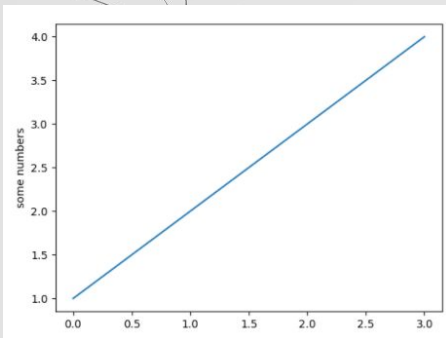
```
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
```

Matplotlib

Visualization with Python

matplotlib

CSCRS



Matplotlib

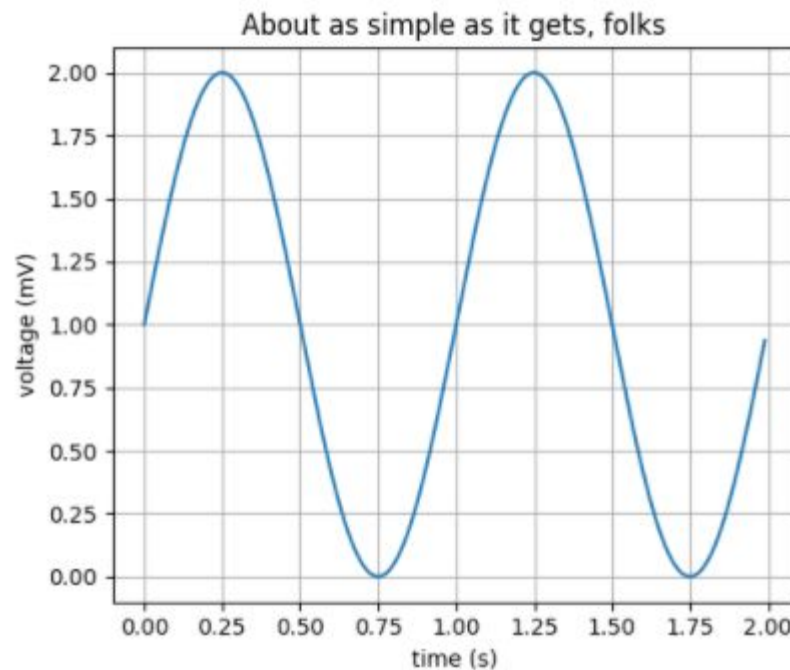
Visualization with Python

```
import matplotlib.pyplot as plt
import numpy as np

t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2*np.pi*t)
plt.plot(t, s)

plt.xlabel('time (s)')
plt.ylabel('voltage (mV)')
plt.title('About as simple as it gets, folks')
plt.grid(True)
plt.savefig("test.png")
plt.show()
```

matplotlib

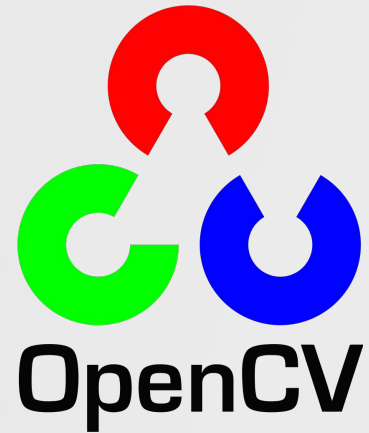


OpenCV



- OpenCV is an open source library used in real-time computer vision and machine learning software library.
- OpenCV has C++, Python, Java and MATLAB interfaces.

```
# read data  
image = cv2.imread(self.images_fps[i])  
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  
mask = cv2.imread(self.masks_fps[i], 0)
```



03

DL Working Environments

Deep Learning - Working Environment



Some environments that can be used while developing a deep learning model

Local Environment:

- Anaconda
- Spyder
- Jupyter Notebook

Cloud Environment:

- Google Colab
- AWS
- Azure

Computer Needed:

- GPU Supported, Cuda, cuDNN

Container Engine

- Docker

 Home

 Environments

 Learning

 Community

Documentation

Developer Blog



Applications on

base (root)

Channels



JupyterLab

1.2.6

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch



Notebook

6.0.3

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch



Qt Console

[4.6.0](#)

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch



Glueviz

0.15.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install



Orange 3

3.23.1

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install



RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Install

Spyder 3.2 (Python 3.6)

Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

untitled0

File Edit View Insert Cell Kernel Help Trusted Kernel

In []: 1+1

Uso

En este panel es posible obtener la ayuda de cualquier objeto al oprimir **Ctrl+I** estando al frente del mismo, bien sea en el Editor o en la Terminal.

Esta ayuda también se puede mostrar automáticamente después de escribir un paréntesis junto a un

Explorador de variables Explorador de archivos Ayuda

Terminal de IPython

Terminal 1/A

Python 3.6.1 [Anaconda custom (64-bit)] (default, May 11 2017, 13:25:24) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.

- ? -> Introduction and overview of IPython's features.
- %quickref -> Quick reference.
- help -> Python's own help system.
- object? -> Details about 'object', use 'object??' for extra details.

In [1]:

Historial de comandos Terminal de IPython

Permisos: RW Fin de línea: CRLF Codificación: UTF-8 Línea: 9 Columna: 1 Memoria: 83 %

jupyter Index Last Checkpoint: 13 minutes ago (unsaved changes)








[Visit repo](#)[Copy Binder link](#)

File Edit View Insert Cell Kernel Widgets Help

Trusted

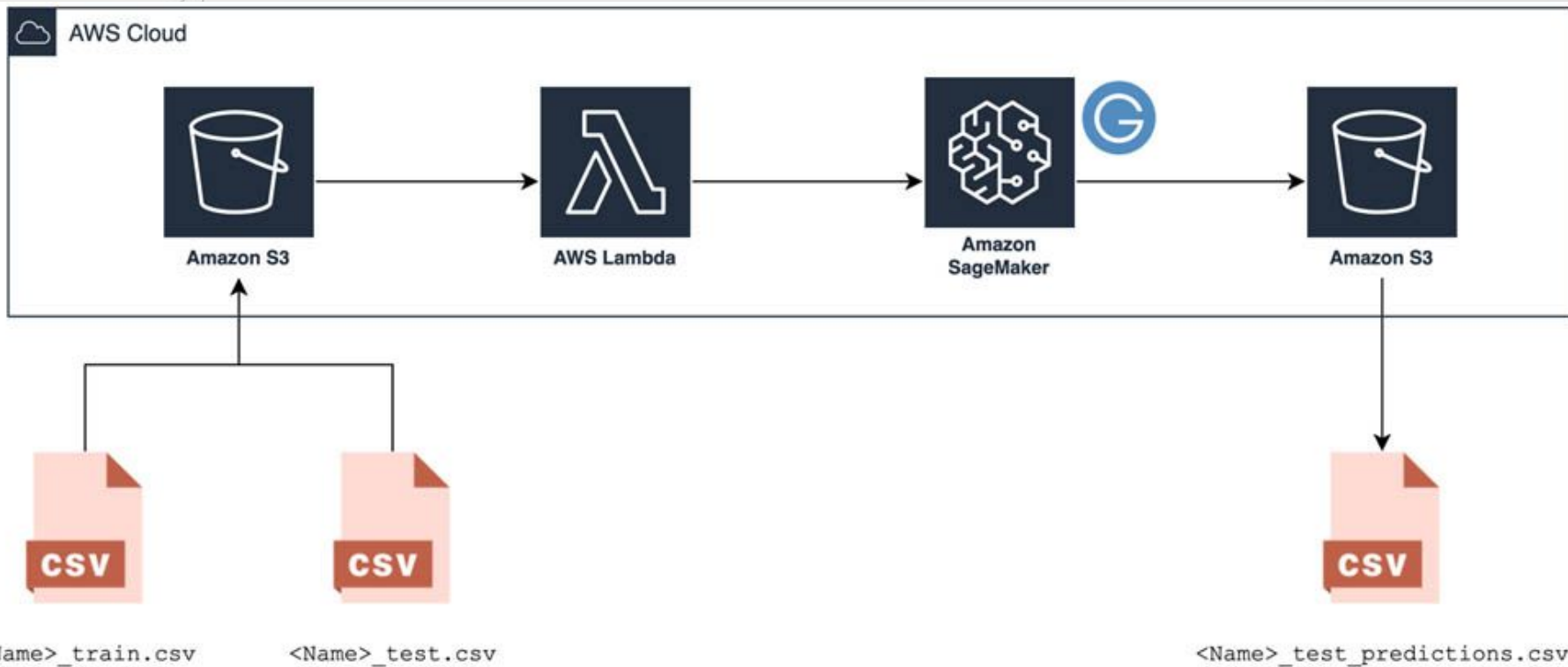


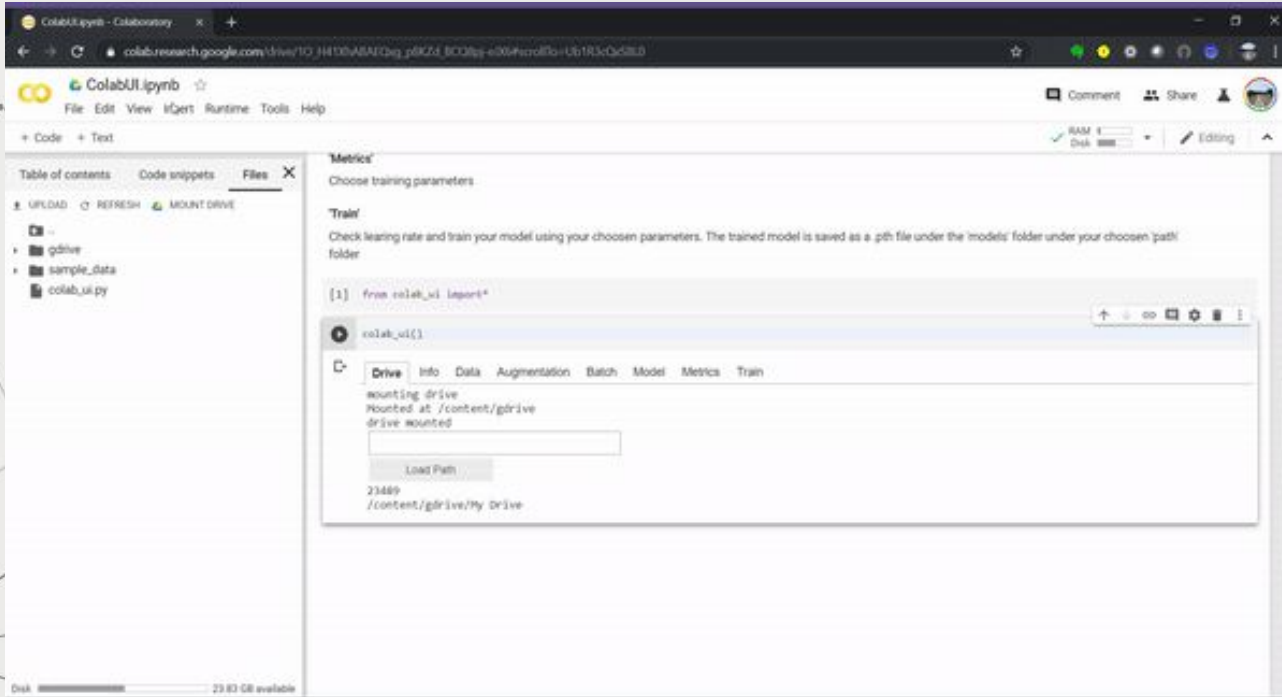
Python 3

        Run    Code 

Welcome to Jupyter!

In []: 





ColabUI.ipynb - Colaboratory

colab.research.google.com/drive/1O_H400VARNQig_p8K24_BCOlty-e0X#scrollTo=U61R3cQd38.8

ColabUI.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Table of contents Code snippets Files

UPLOAD REFRESH MOUNT DRIVE

-
- gdrive
- sample_data
- colab_ui.py

Metrics

Choose training parameters

Train

Check learning rate and train your model using your chosen parameters. The trained model is saved as a .pth file under the 'models' folder under your chosen 'path' folder

```
[1] from colab_ui import*
```

colab_ui{}

Drive Info Data Augmentation Batch Model Metrics Train

mounting drive
Mounted at /content/gdrive
drive mounted

Load Path

23489
/content/gdrive/My Drive

Disk 23.83 GB available





Home

Author



Automated ML



Designer



Notebooks

Assets



Datasets



Experiments



Pipelines



Models



Endpoints

Manage



Compute



Datastores



Data labeling

adb_automation_eastus2_ws > Welcome

Automated machine learning

Let automated machine learning train and find the best model based on your data without writing a single line of code. [Learn more](#)



New automated ML run

No recent automated ML runs to display.

Click "New automated ML run" to create your first run



[Learn more](#)

Documentation

[View all documentation](#)



Concept: What is automated machine learning?

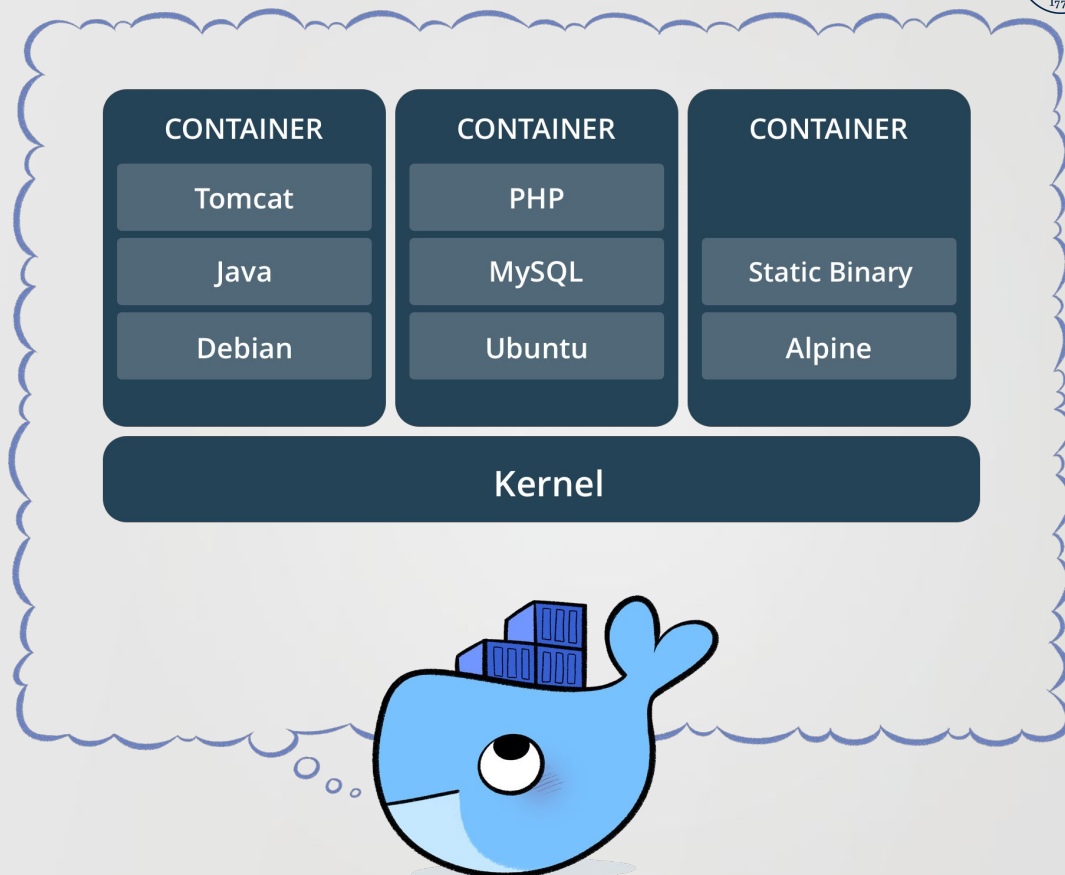
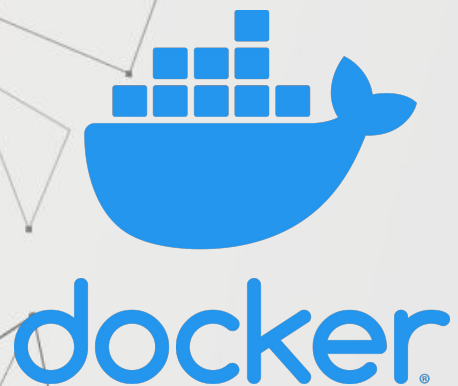


Tutorial: Create your first classification model with automated machine learning



Blog: Build more accurate forecasts with new capabilities in automated machine learning





04

Suggested Resources

>YazBel

pdf-destegi:sonlanmistir forum



next

Python Programlama Dili

Python 3 için Türkçe Kaynak

- 01. Bu Kitap Hakkında
 - 01. Bu Kitaptan Nasıl Yararlanabilirim?
 - 02. Nereden Yardım Alabilirim?
 - 03. Projeye Nasıl Yardımcı Olabilirim?
 - 04. Kullanım Koşulları
- 02. Python Hakkında
 - 01. Python Nedir?
 - 02. Neden Programlama Öğrenmek İsteyeyim?
 - 03. Neden Python?
 - 04. Python Nasıl Telaffuz Edilir?
 - 05. Platform Desteği
 - 06. Farklı Python Sürümleri
 - 07. Hangi Seriyi Öğrenmeliyim?
- 03. Python Nasıl Kurulur?
 - 01. GNU/Linux Kullanıcıları
 - 01. Kurulu Python Sürümü
 - 02. Paket Deposundan Kurulum
 - 03. Kaynaklı Kurulum



DEEP LEARNING with Python

François Chollet

MANNING



Deep Learning with PyTorch

Dr. Sepp Rein
Lucia D'Amico
Thomas Wolf
Foreword by Geoffrey Hinton

MANNING





THANKS

Does anyone have any questions?

irem@cscrs.itu.edu.tr
iremkomurcu.com



İTÜ Maslak Kampüsü, Uydu
Yolu, 34469, İstanbul, Türkiye



+90 212 285 68 13



info@cscrs.itu.edu.tr



www.cscrs.itu.edu.tr