Machine Learning with Python: An Eye opener Master class

Presented by

Mr. S. Peerbasha, MCA., M.Phil., SET., NET., MBA., M.Tech., (Ph.D)
Department of Computer Applications
Jamal Mohamed College (Autonomous)
Trichy-620 020.

Reach me at 9600466890
Mail me :- bashapeer2003@gmail.com / spb@jmc.edu

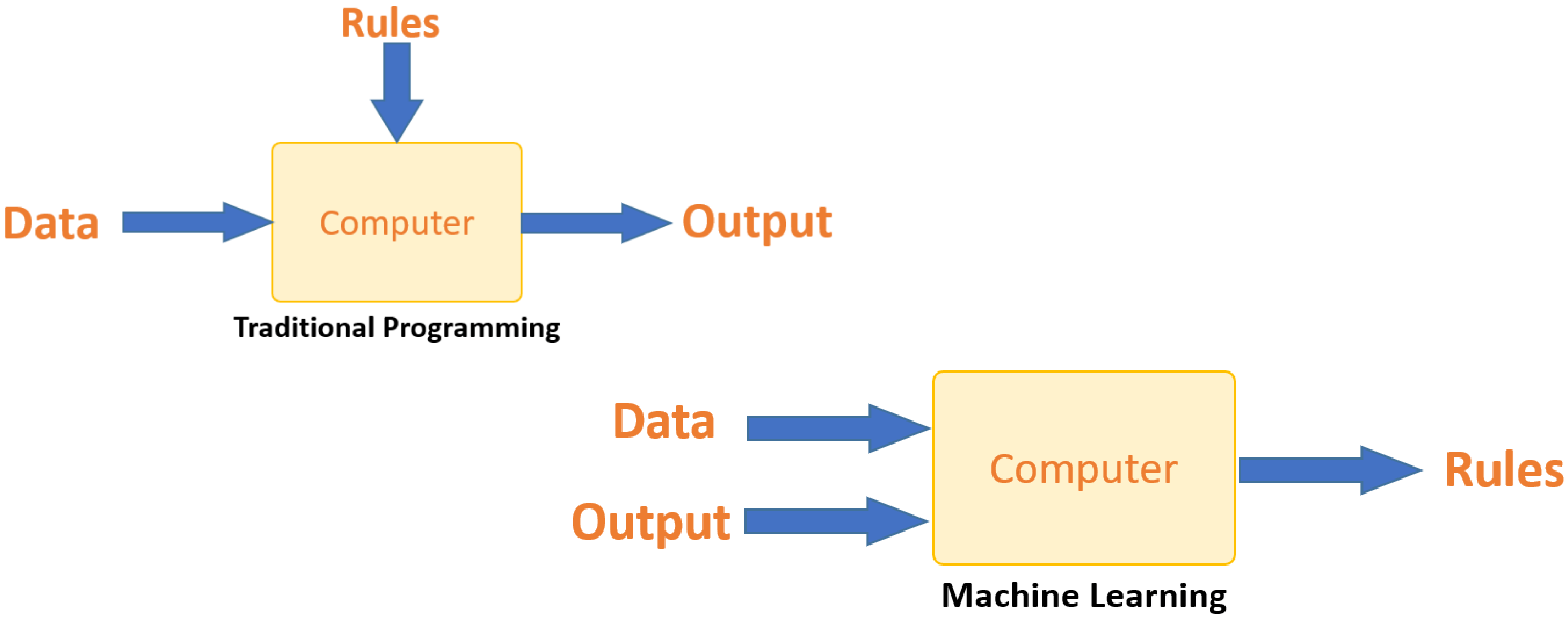
Outline

- Why use Machine Learning?
- Applications of Machine Learning
- How Machine Learning Works?
- Machine Learning Workflow
- Teachable Machine – Practical demo
- Steps to download Anaconda
- Working Python with Jupyter
- Top 5 Python Libraries
- Types of Machine Learning
- Linear Regression Demo

Machine learning

- “Machine learning is the science of getting computers to act without being explicitly programmed.” — Stanford University.
- Machine Learning is making the computer learn from studying data and statistics.
- Machine Learning is a step into the direction of artificial intelligence (AI). Learning refers to the ability of a computer system to improve its performance on a task over time, based on experience and feedback.
- Machine Learning is a program that analyses data and learns to predict the outcome.

Traditional Programming Vs Machine Learning

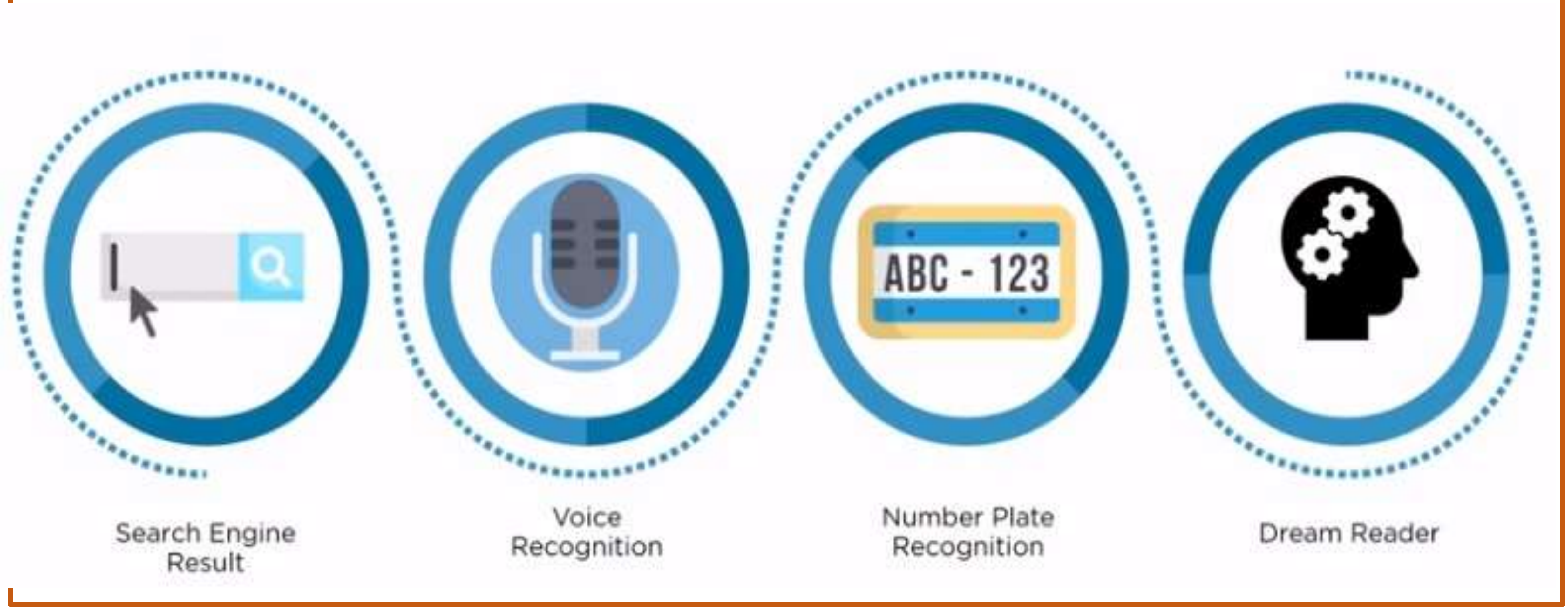


Example -Machine Learning Applications

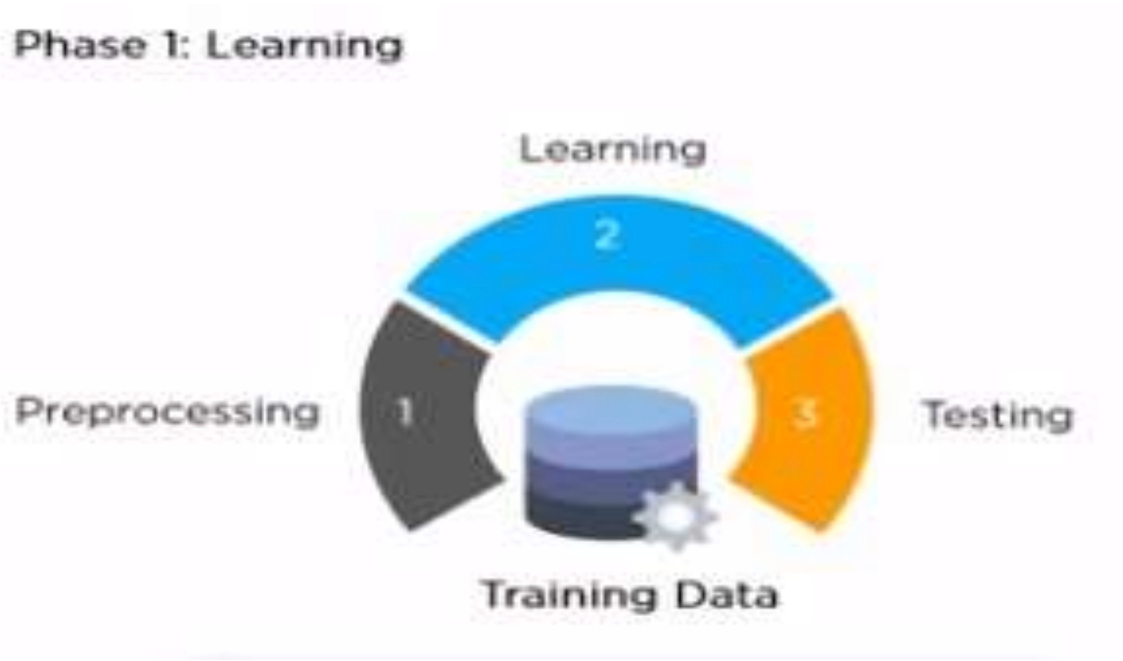
- Social Media – User activity-based recommendations
- Product Recommendations - basically a filtering system that seeks to predict and show the items that a user would like to purchase.
- Image Recognition - pixel and pattern analysis of an image to recognize the image as a particular object.
- Sentiment Analysis- exploration of subjective opinions or feelings collected from various sources about a particular subject.
- Healthcare - Clinical decision support

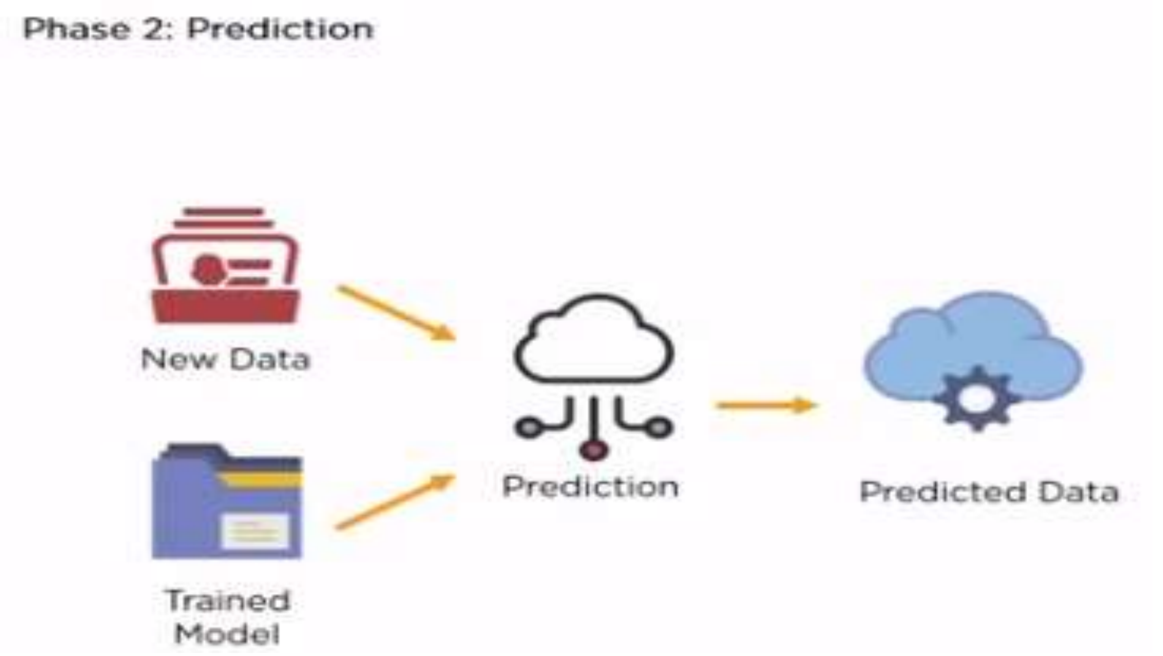
Why use Machine Learning?

Applications of Machine Learning

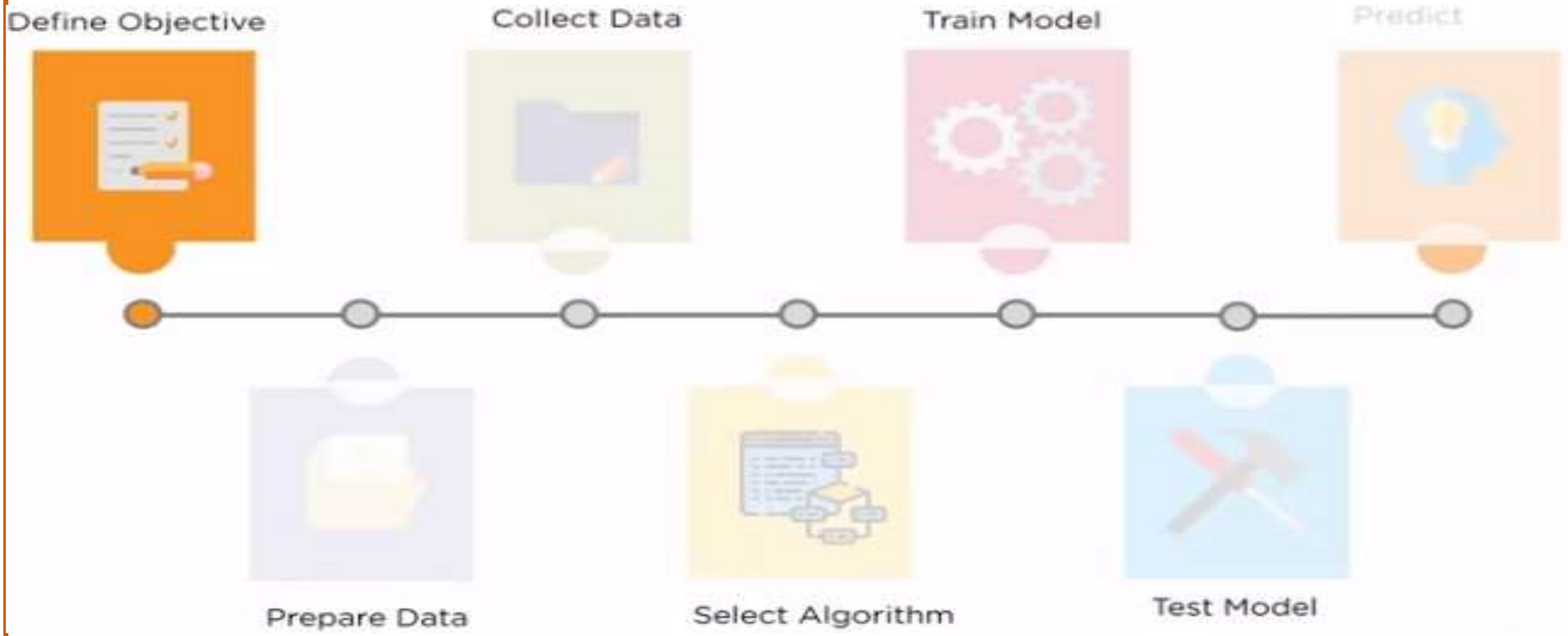


How Machine Learning Works?

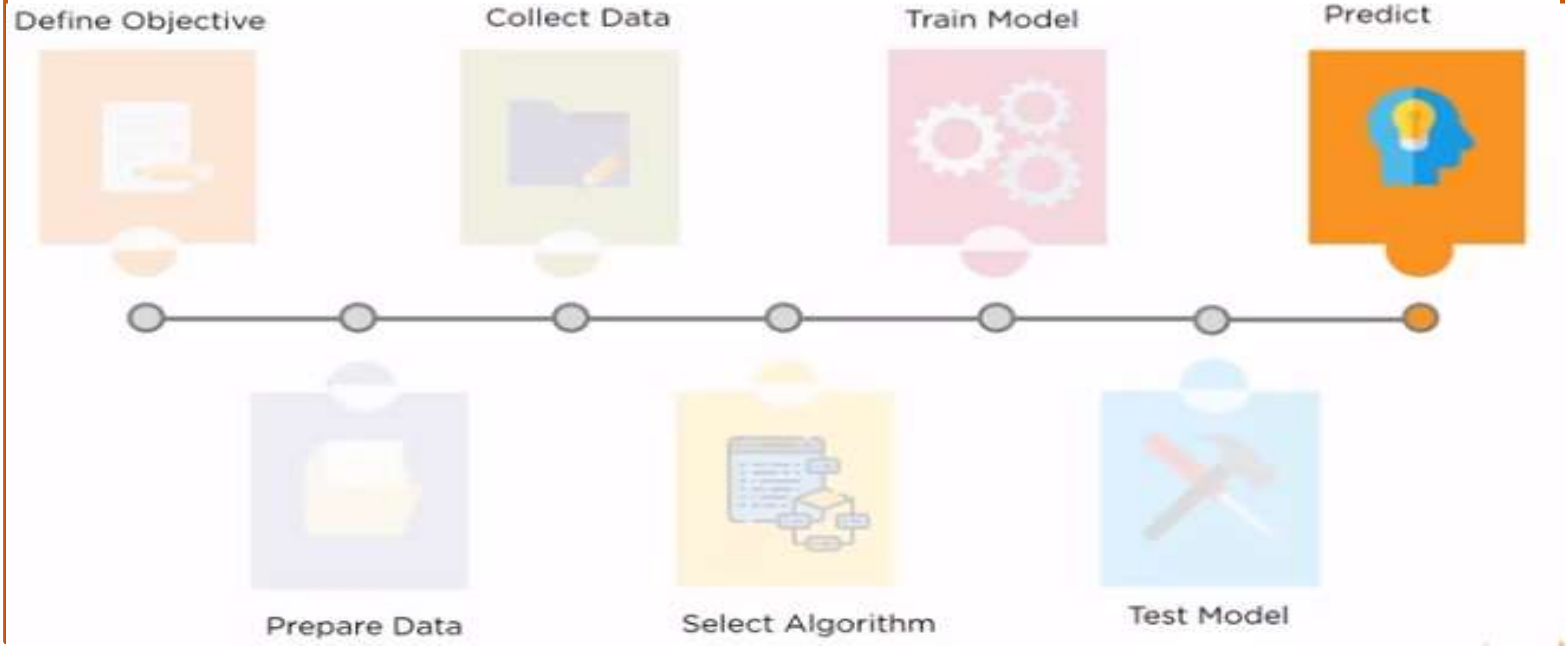




Machine Learning Workflow



Machine Learning Workflow



Python Libraries in Machine Learning

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

NumPy

❖NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions.

It is very useful for fundamental scientific computations in Machine Learning.

It is particularly useful for linear algebra, Fourier transform, and random number capabilities.

High-end libraries like Tensor Flow uses NumPy internally for manipulation of Tensors.

```
import numpy as np

# Creating two arrays of rank 2
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6], [7, 8]])

# Creating two arrays of rank 1
v = np.array([9, 10])
w = np.array([11, 12])

# Inner product of vectors
print(np.dot(v, w), "\n")

# Matrix and Vector product
print(np.dot(x, v), "\n")

# Matrix and matrix product
print(np.dot(x, y))
```

Output:

```
219
[29 67]
 [[19 22] [43 50]]
```

- SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics.

There is a difference between the SciPy library and the SciPy stack.

The SciPy is one of the core packages that make up the SciPy stack.

SciPy is also very useful for image manipulation.

```
# Python script using Scipy
# for image manipulation

from scipy.misc import imread, imsave, imresize

# Read a JPEG image into a numpy array
img = imread(‘E:/HCC_ML/ cat.jpg') # path of the image
print(img.dtype, img.shape)

# Tinting the image
img_tint = img * [1, 0.45, 0.3]

# Saving the tinted image
imsave(‘E:/HCC_ML/cat-tinted.jpg', img_tint)

# Resizing the tinted image to be 300 x 300 pixels
img_tint_resize = imresize(img_tint, (300, 300))

# Saving the resized tinted image
imsave(‘E:/HCC_ML/ cat-tinted_resized.jpg', img_tint_resize)
```

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google.

As the name suggests, TensorFlow is a framework that involves defining and running computations involving tensors.

It can train and run deep neural networks that can be used to develop several AI applications.

TensorFlow is widely used in the field of deep learning research and application.

```
#  Python program using TensorFlow
#  for multiplying two arrays

# import `tensorflow`
import tensorflow as tf

# Initialize two constants
x1 = tf.constant([1, 2, 3, 4])
x2 = tf.constant([5, 6, 7, 8])

# Multiply
result = tf.multiply(x1, x2)

# Initialize the Session
sess = tf.Session()

# Print the result
print(sess.run(result))

# Close the session
sess.close()
```

Output:

```
[ 5 12 21 32]
```

Keras

Keras is a very popular Machine Learning library for Python.

It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano.

It can run seamlessly on both CPU and GPU.

Keras makes it really for ML beginners to build and design a Neural Network.

One of the best thing about Keras is that it allows for easy and fast prototyping.

Pandas

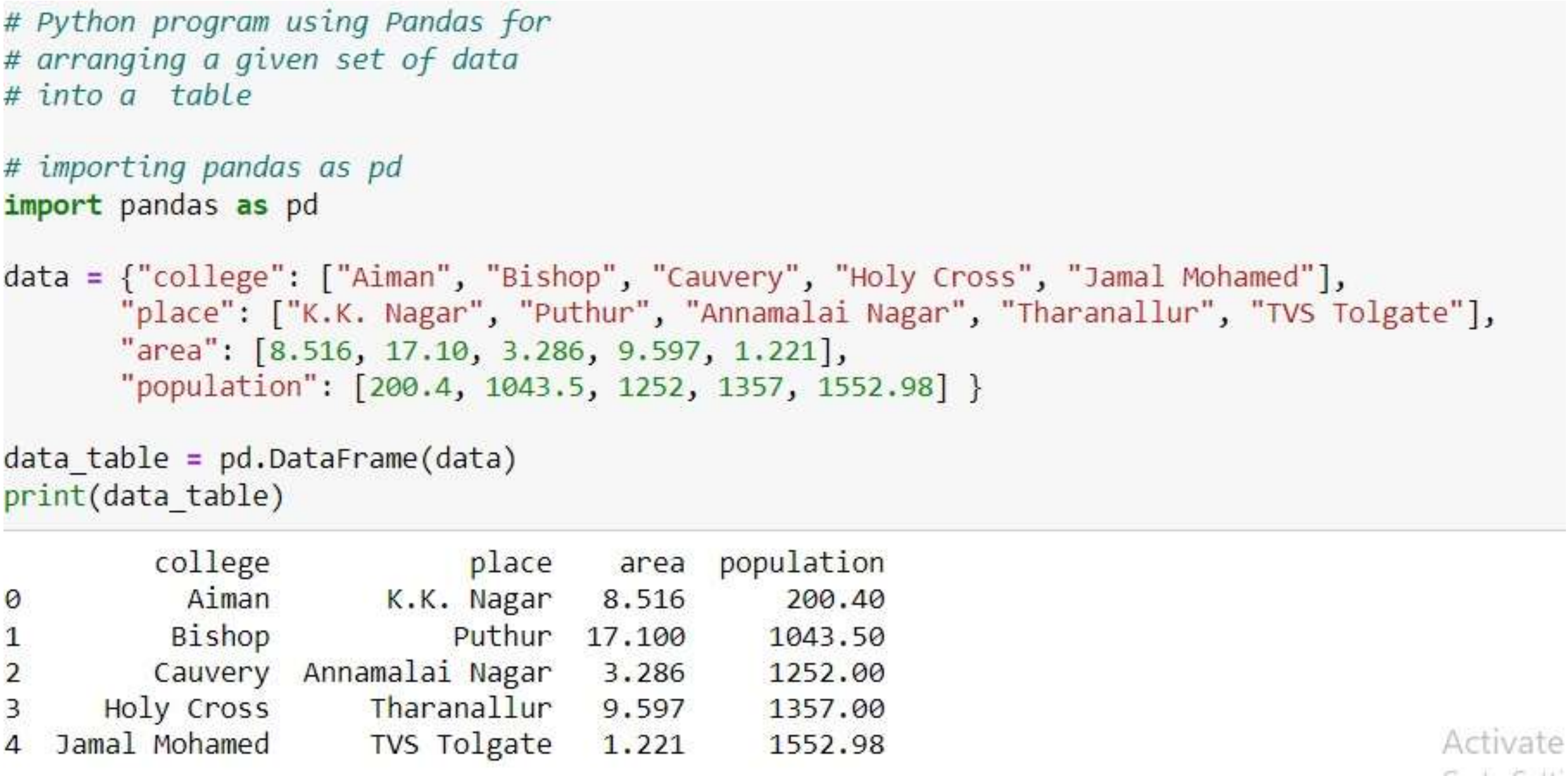
Pandas is a popular Python library for data analysis.

It is not directly related to Machine Learning.

As we know that the dataset must be prepared before training.

In this case, Pandas comes handy as it was developed specifically for data extraction and preparation.

It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.



```python
# Python program using Pandas for
# arranging a given set of data
# into a  table

# importing pandas as pd
import pandas as pd

data = {"college": ["Aiman", "Bishop", "Cauvery", "Holy Cross", "Jamal Mohamed"],
        "place": ["K.K. Nagar", "Puthur", "Annamalai Nagar", "Tharanallur", "TVS Tolgate"],
        "area": [8.516, 17.10, 3.286, 9.597, 1.221],
        "population": [200.4, 1043.5, 1252, 1357, 1552.98] }

data_table = pd.DataFrame(data)
print(data_table)
```

```
         college            place    area  population
0          Aiman       K.K. Nagar   8.516      200.40
1         Bishop           Puthur  17.100     1043.50
2        Cauvery  Annamalai Nagar   3.286     1252.00
3     Holy Cross      Tharanallur   9.597     1357.00
4  Jamal Mohamed      TVS Tolgate   1.221     1552.98
```

matplotlib

Version 3.0.2

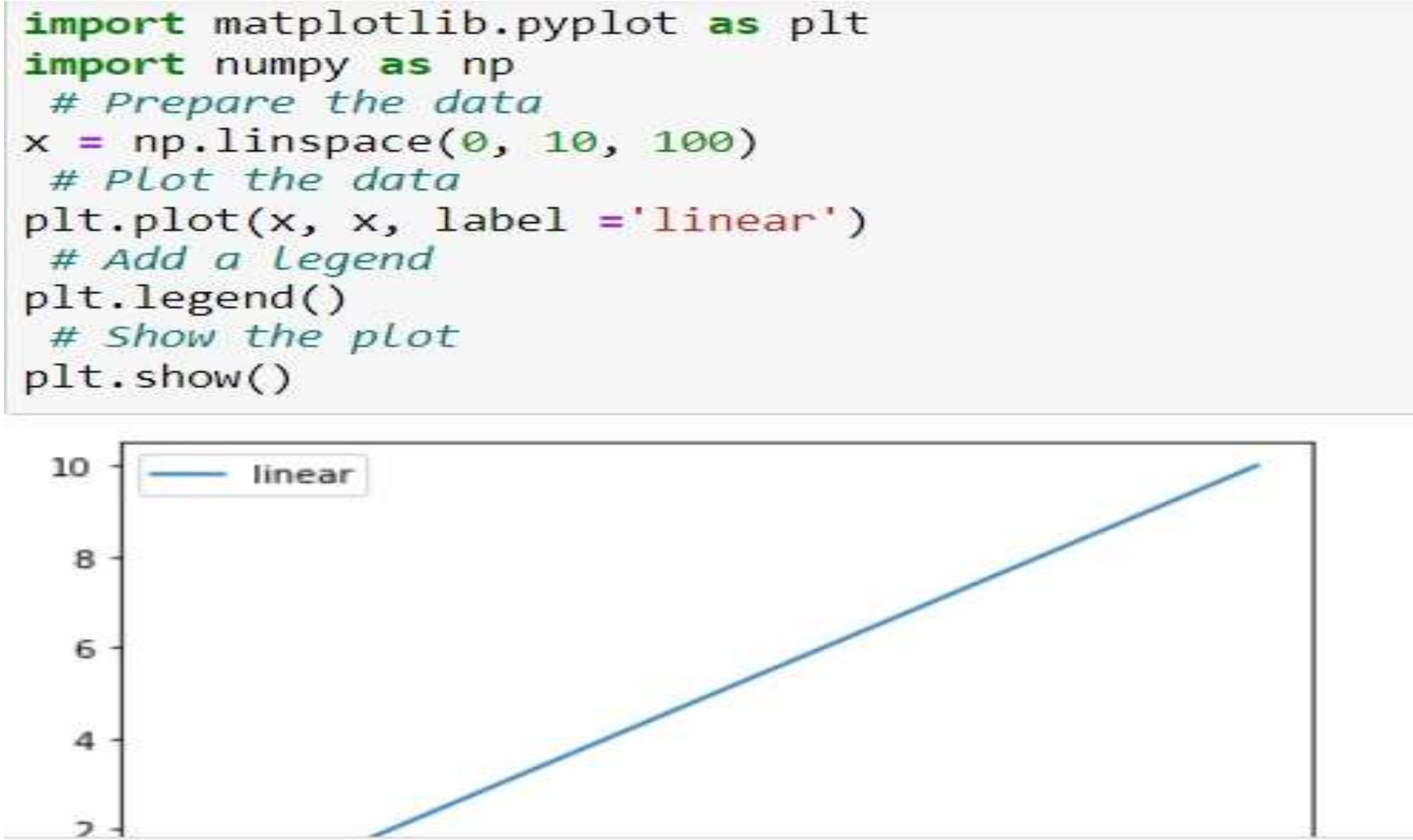
Matplotlib is a very popular Python library for data visualization.

Like Pandas, it is not directly related to Machine Learning.

It particularly comes in handy when a programmer wants to visualize the patterns in the data.

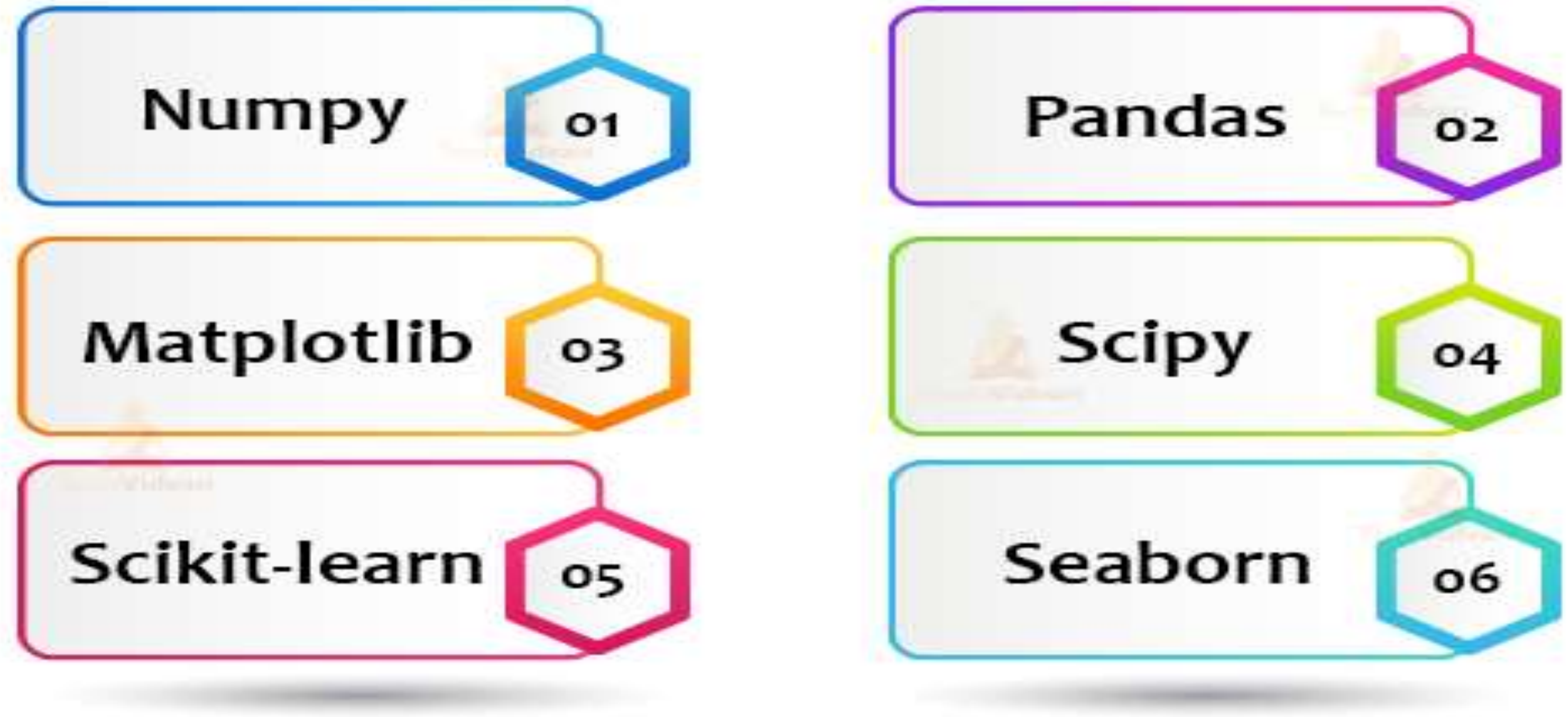
It is a 2D plotting library used for creating 2D graphs and plots.

It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats, etc,



```
import matplotlib.pyplot as plt
import numpy as np
 # Prepare the data
x = np.linspace(0, 10, 100)
 # Plot the data
plt.plot(x, x, label ='linear')
 # Add a legend
plt.legend()
 # Show the plot
plt.show()
```

Popular Python libraries for Data Science



Numpy 01

Pandas 02

Matplotlib 03

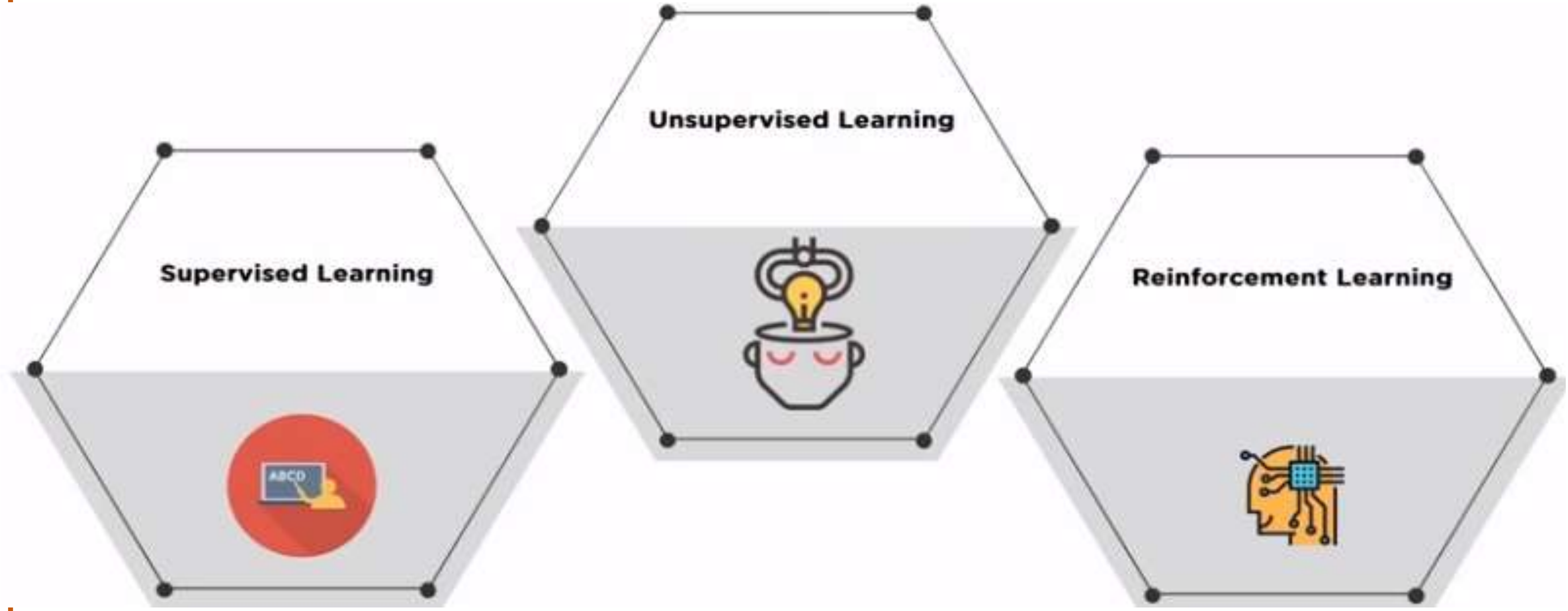
Scipy 04

Scikit-learn 05

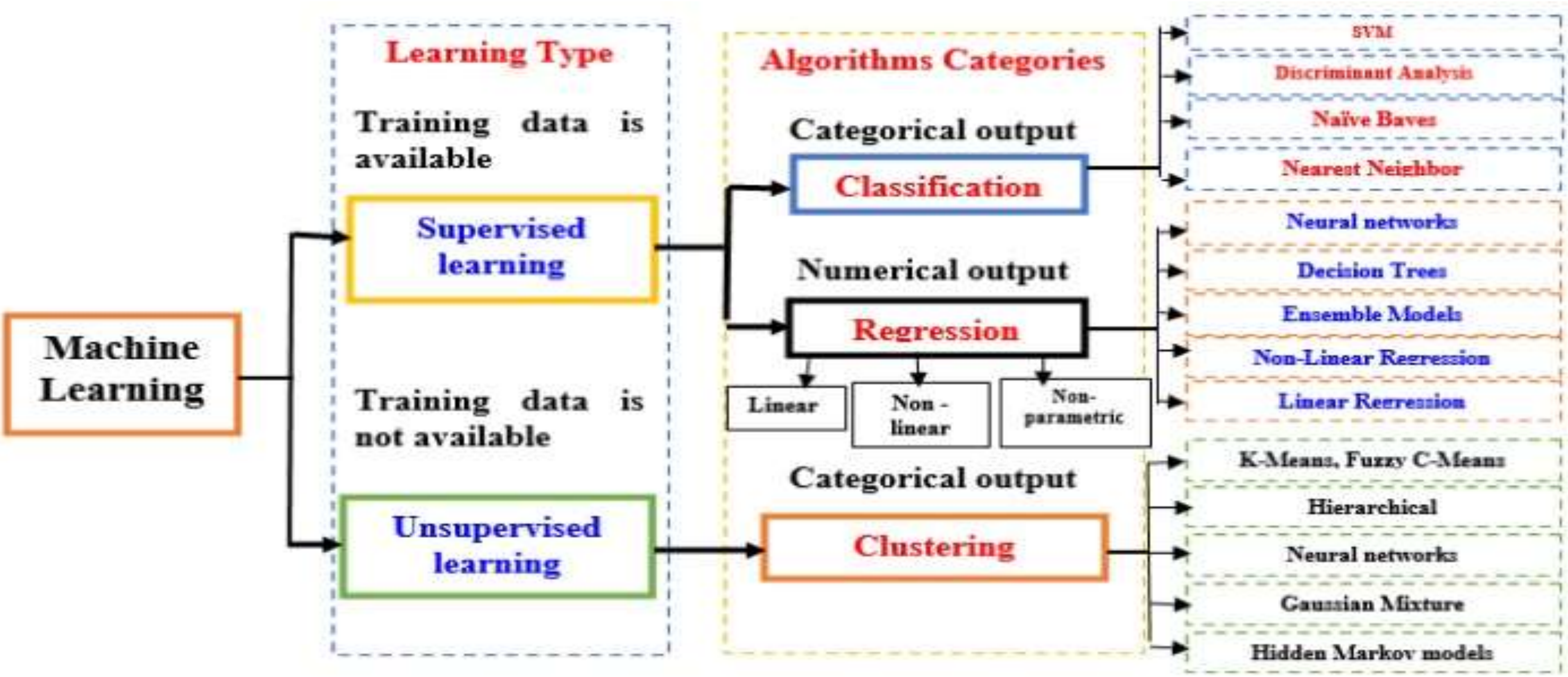
Seaborn 06

Running Anaconda through Jupyter
(A Practical demo)

Types of Machine Learning



Machine Learning Algorithms Overview

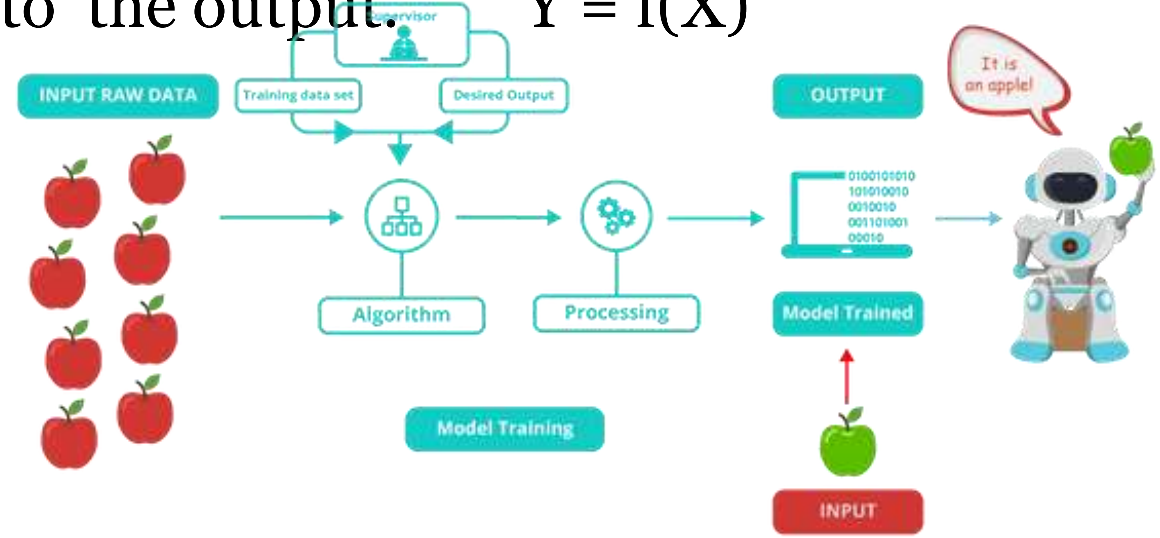


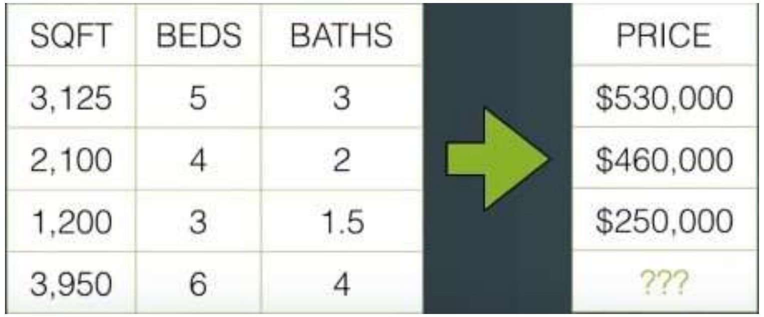
Machine Learning Algorithms

- Supervised Learning
  - Supervised learning is where you have input variables (x) and an output variable

(Y) and you use an algorithm to learn the mapping function from the input

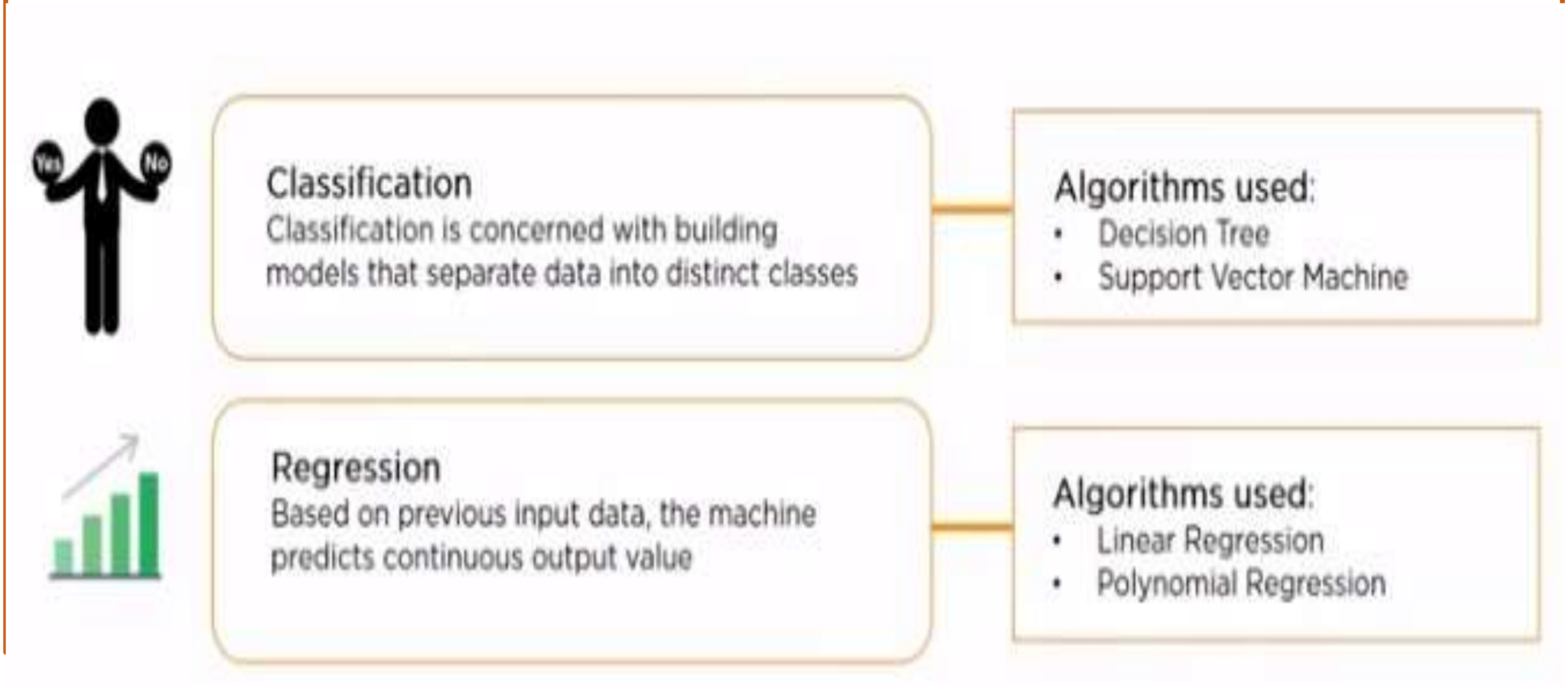
to the output. Y = f(X)





SQFT	BEDS	BATHS	PRICE
3,125	5	3	$530,000
2,100	4	2	$460,000
1,200	3	1.5	$250,000
3,950	6	4	???

Types of Supervised Learning



Classification

Classification is concerned with building models that separate data into distinct classes

Algorithms used:

- Decision Tree
- Support Vector Machine

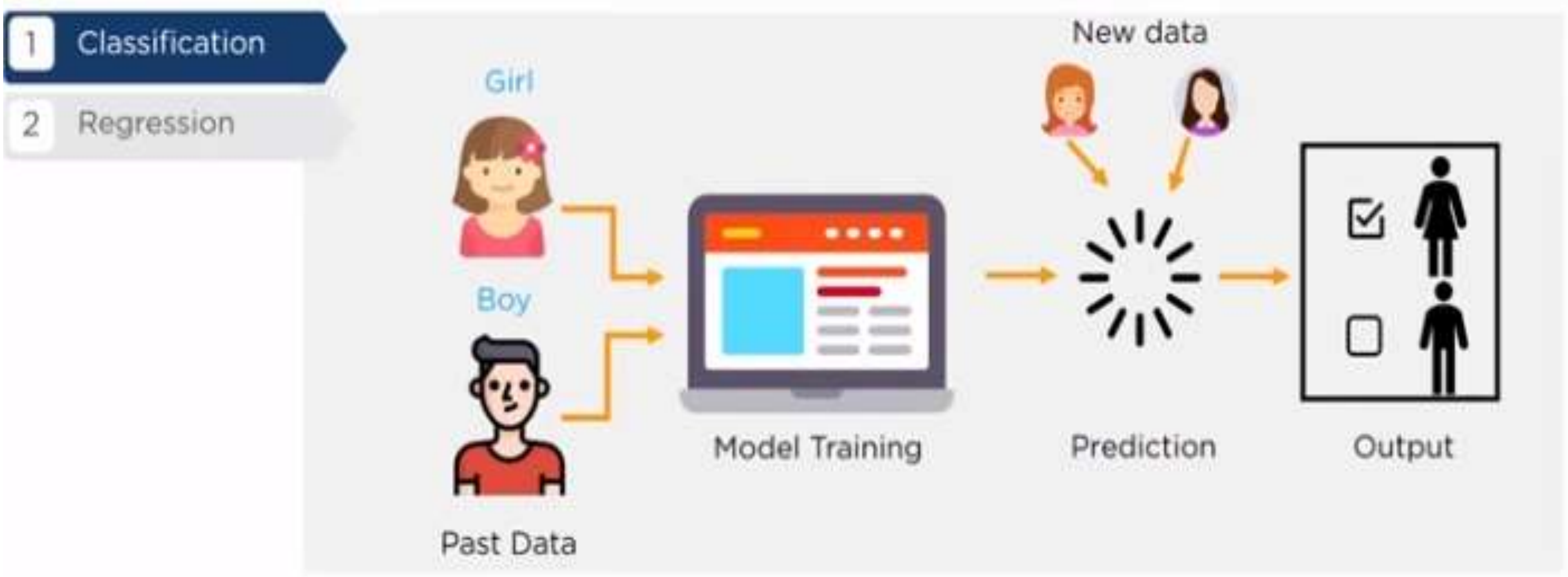
Regression

Based on previous input data, the machine predicts continuous output value

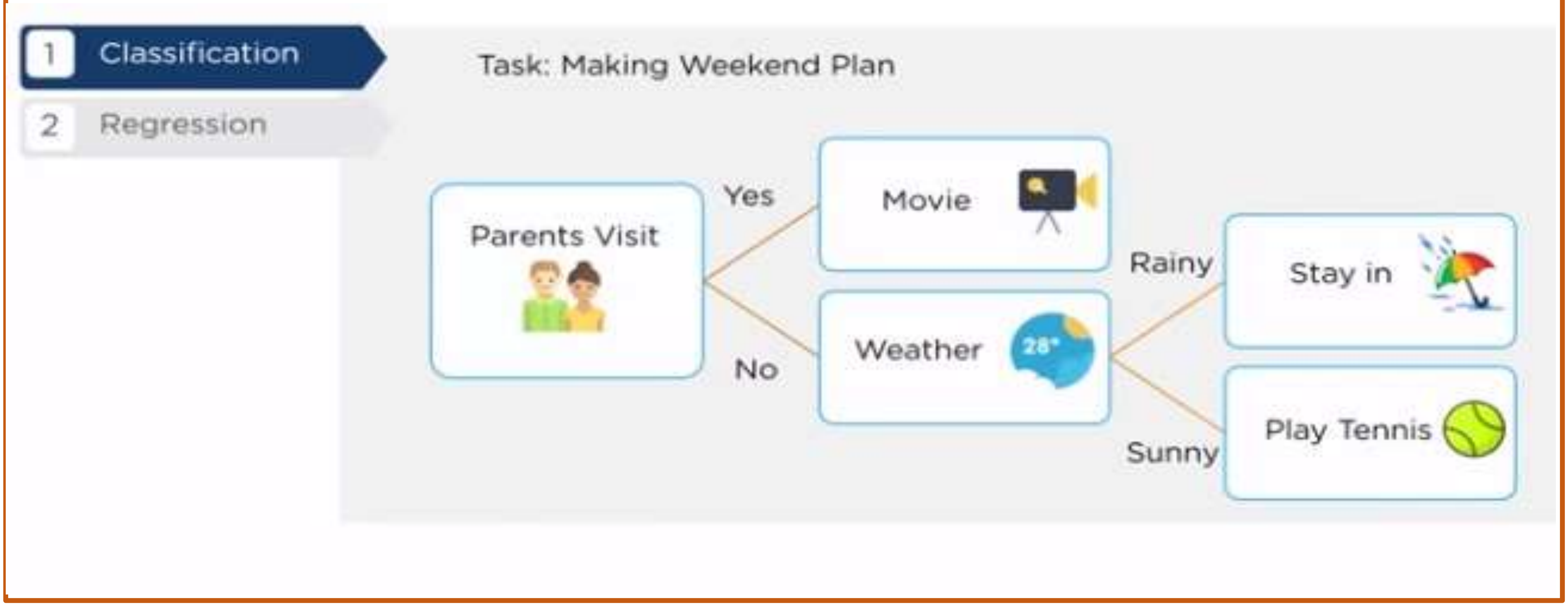
Algorithms used:

- Linear Regression
- Polynomial Regression

Supervised Learning - Classification



Supervised Learning – Decision Tree



1 Classification

2 Regression

Task: Making Weekend Plan

Parents Visit

Yes

Movie

No

Weather

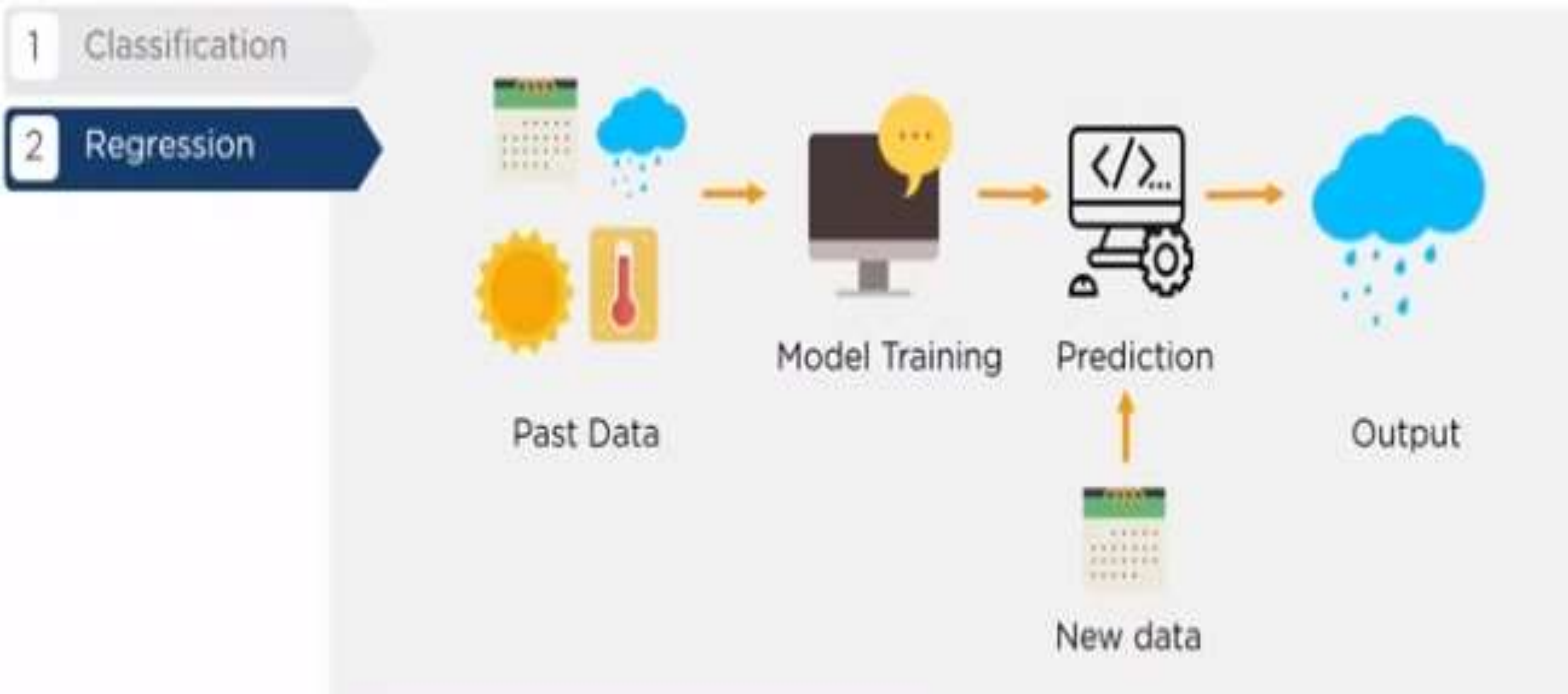
Rainy

Stay in

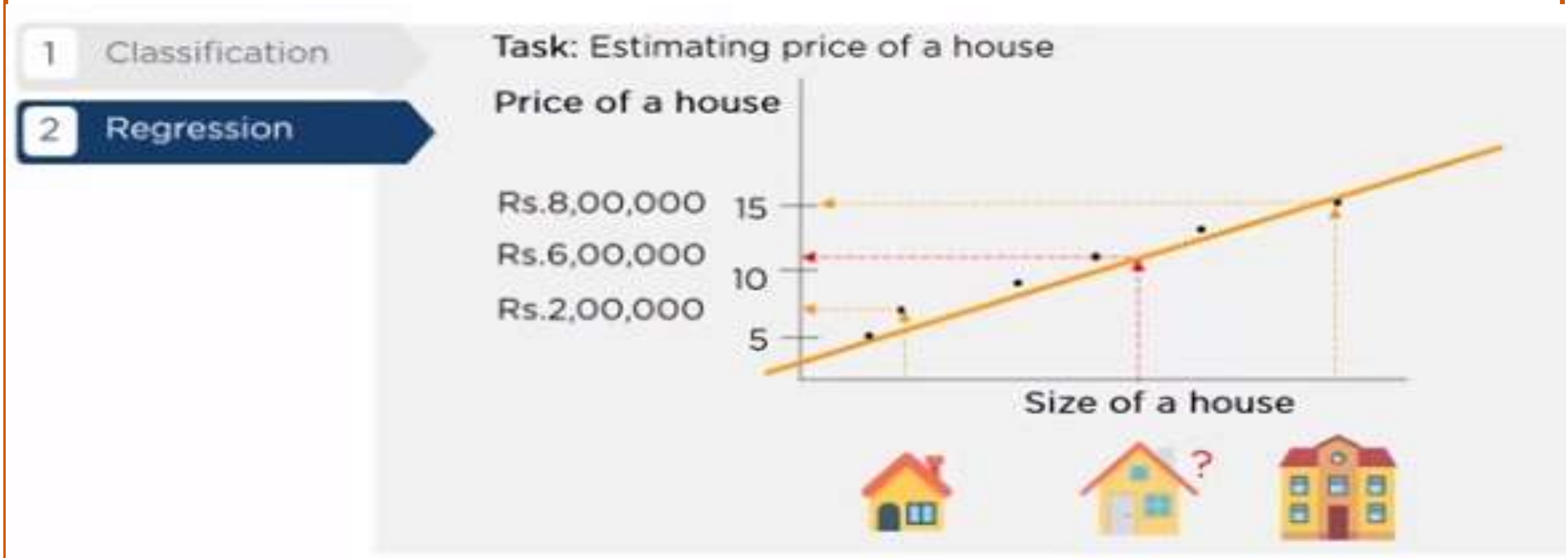
Sunny

Play Tennis

Supervised Learning – Regression



Supervised Learning – Regression

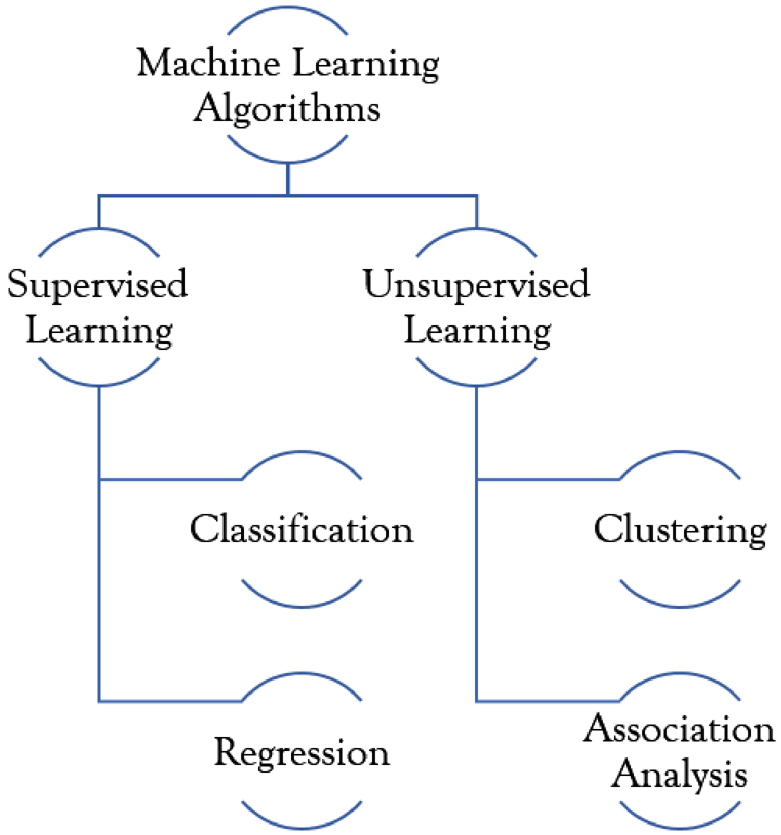


1 Classification

2 Regression

Task: Estimating price of a house

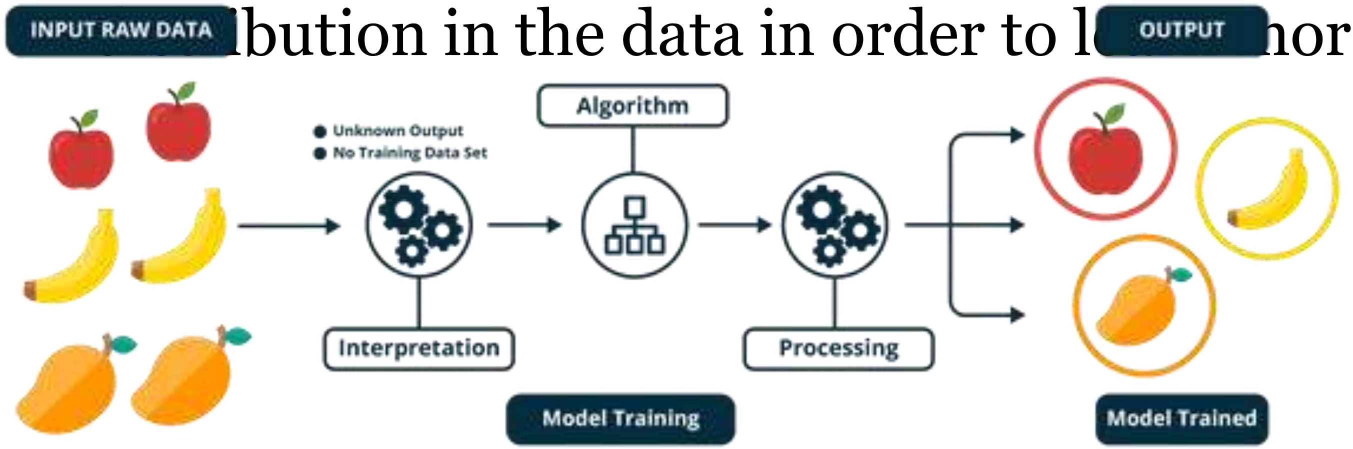
Machine Learning Algorithms

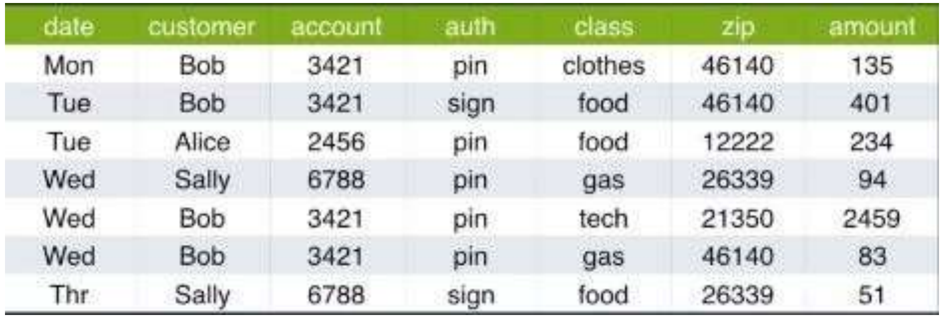


- Classification
  - A classification problem is when the output variable is a category, such as "Red" or "blue" or "disease" and "no disease".
- Regression
  - A regression problem is when the output variable is a real value, such as "dollars" or "weight".
- Clustering:
  - A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- Association:
  - An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Machine Learning Algorithms

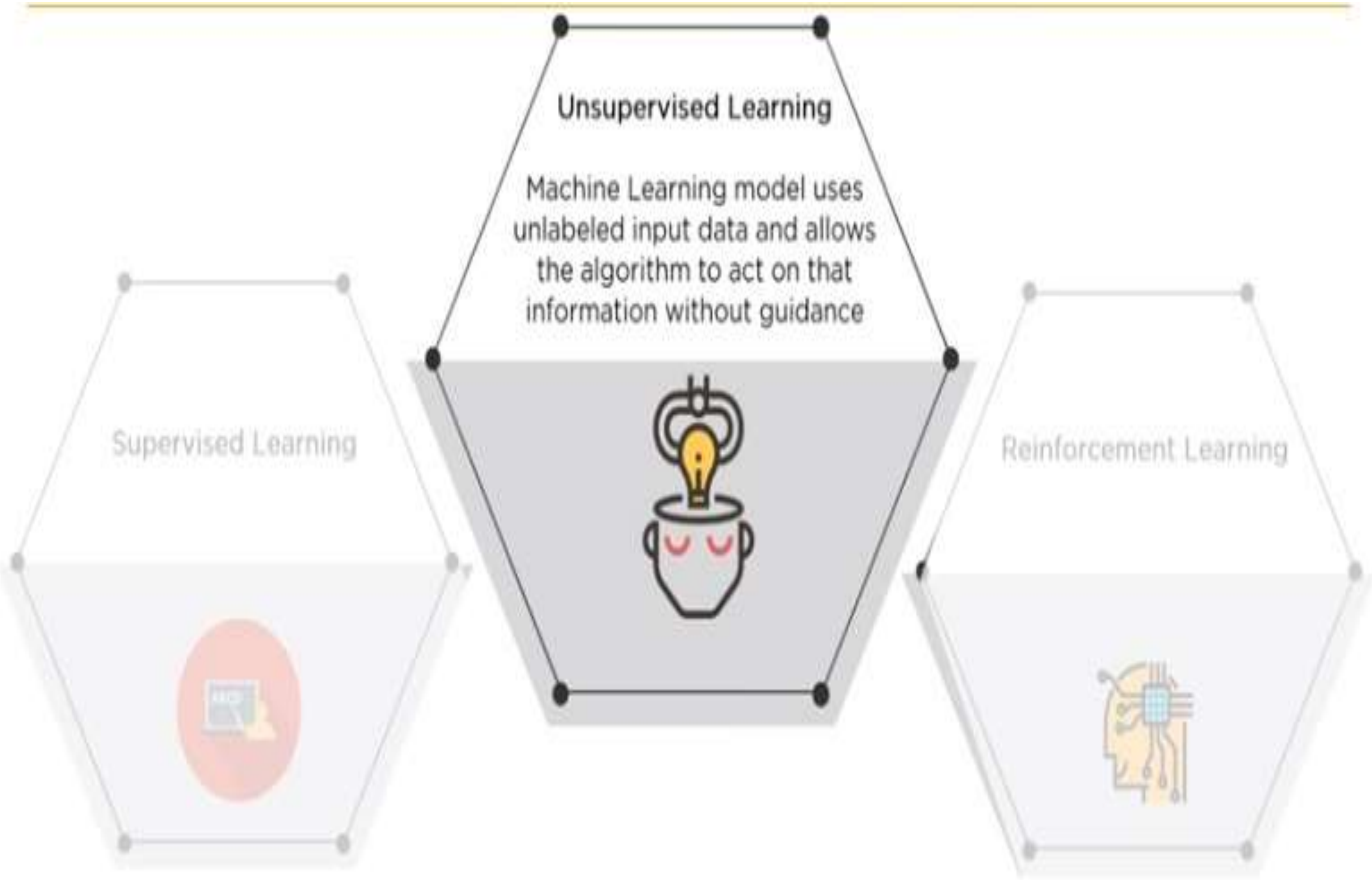
- Unsupervised Learning
  - Unsupervised learning is only havinginput data (X) and no corresponding output variables.
  - The goal for unsupervised learning is to model the underlying structure or ibution in the data in order to l hore about the data.





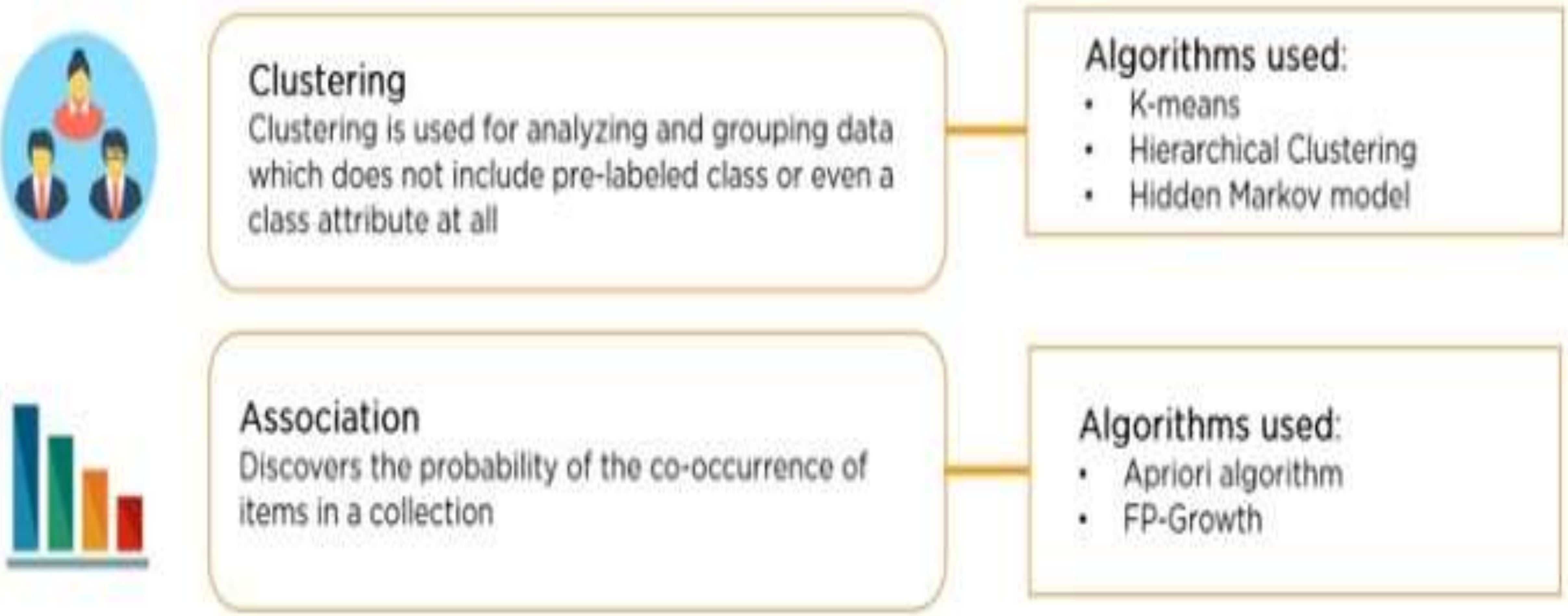
date	customer	account	auth	class	zip	amount
Mon	Bob	3421	pin	clothes	46140	135
Tue	Bob	3421	sign	food	46140	401
Tue	Alice	2456	pin	food	12222	234
Wed	Sally	6788	pin	gas	26339	94
Wed	Bob	3421	pin	tech	21350	2459
Wed	Bob	3421	pin	gas	46140	83
Thr	Sally	6788	sign	food	26339	51

Un Supervised Learning



Machine Learning model uses unlabeled input data and allows the algorithm to act on that information without guidance

Un Supervised Learning



Clustering

Clustering is used for analyzing and grouping data which does not include pre-labeled class or even a class attribute at all

Algorithms used:

- K-means
- Hierarchical Clustering
- Hidden Markov model

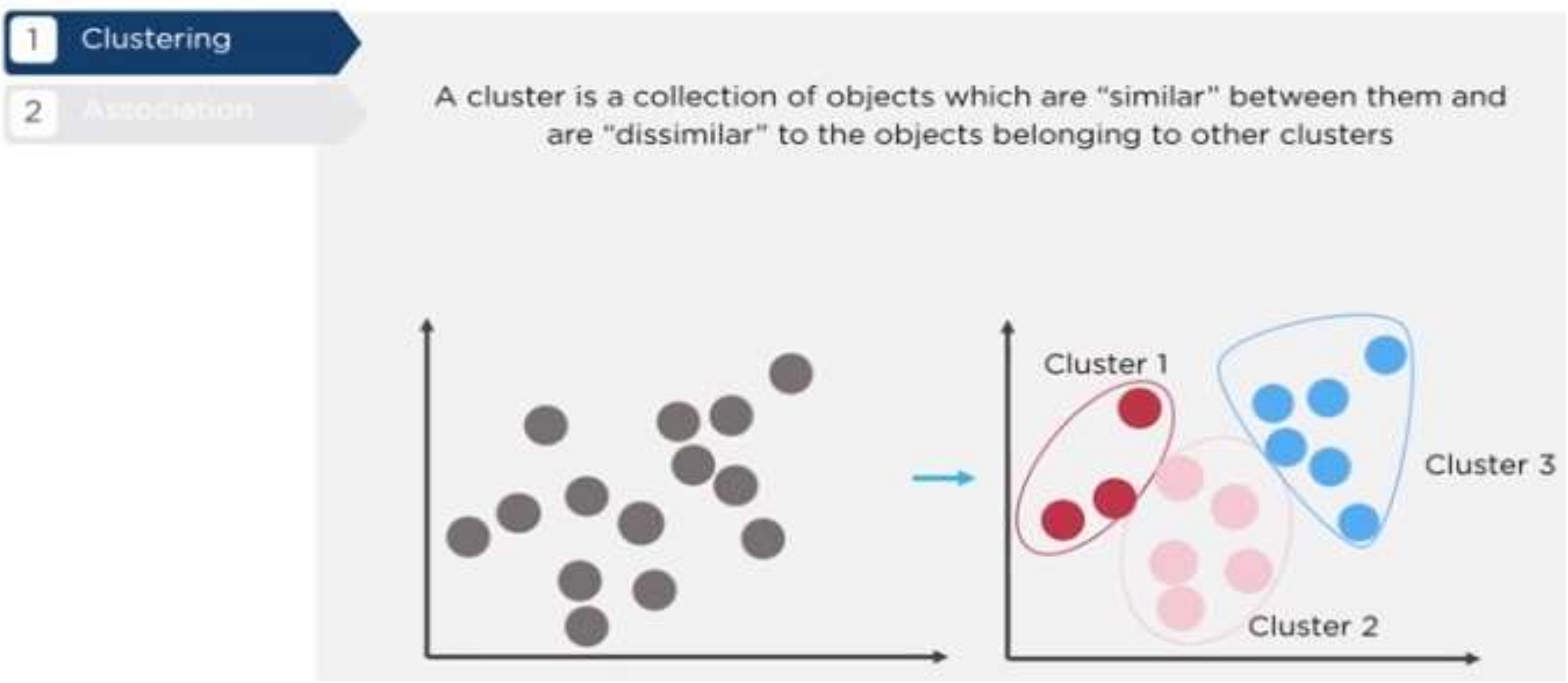
Association

Discovers the probability of the co-occurrence of items in a collection

Algorithms used:

- Apriori algorithm
- FP-Growth

Un Supervised Learning

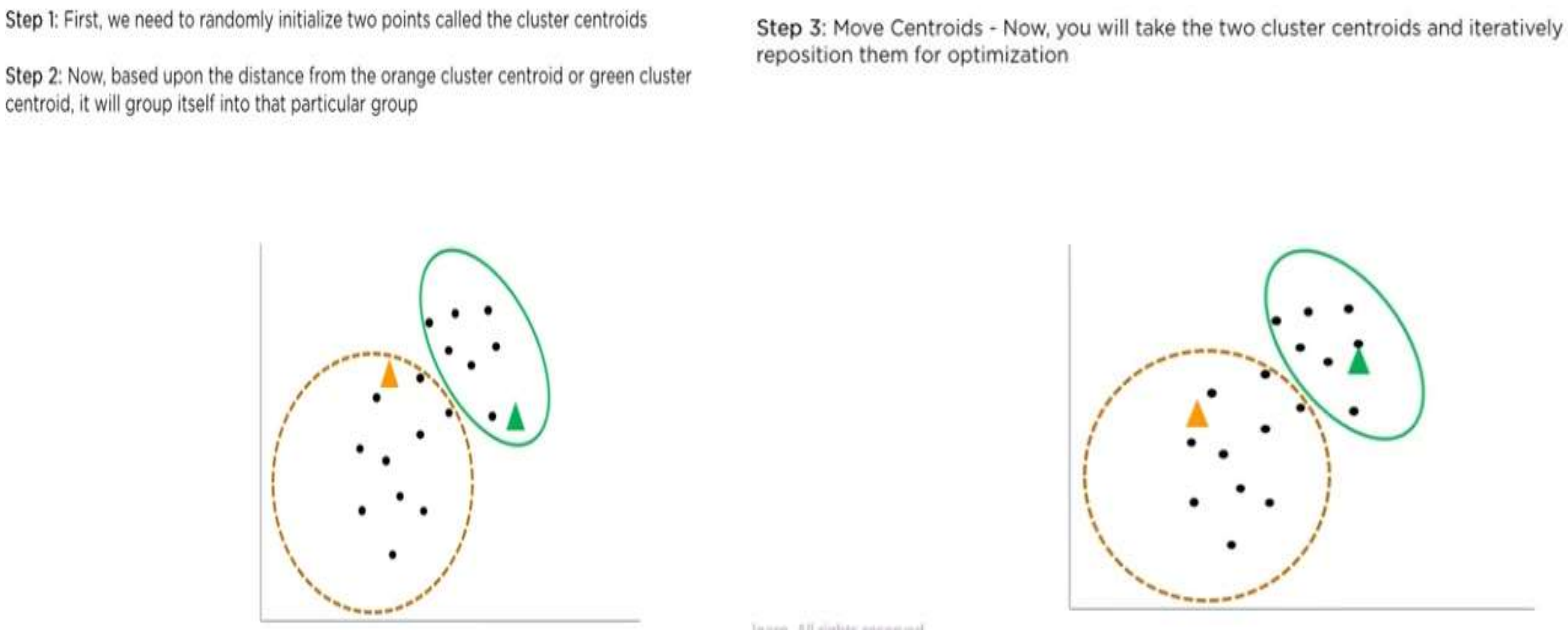


1 Clustering

2 Association

A cluster is a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters

Un Supervised Learning

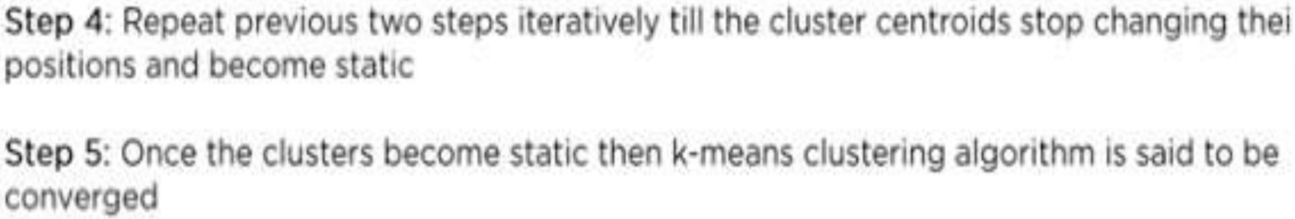


Step 1: First, we need to randomly initialize two points called the cluster centroids

Step 2: Now, based upon the distance from the orange cluster centroid or green cluster centroid, it will group itself into that particular group

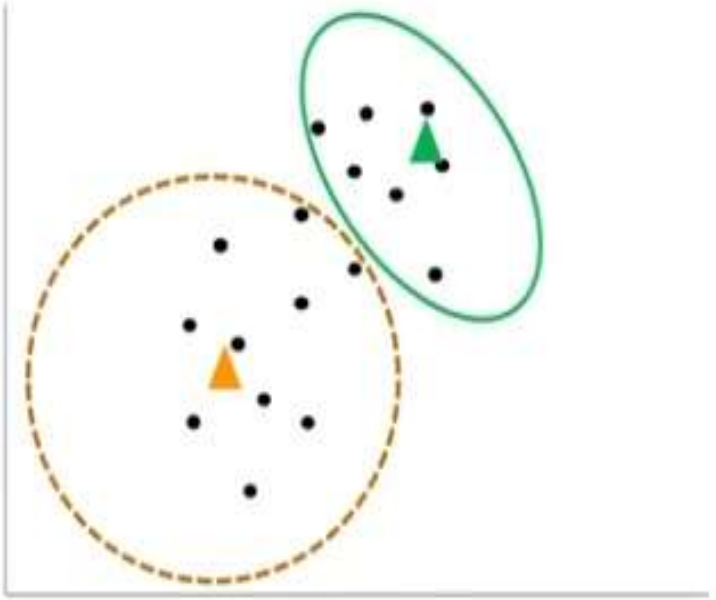
Step 3: Move Centroids - Now, you will take the two cluster centroids and iteratively reposition them for optimization

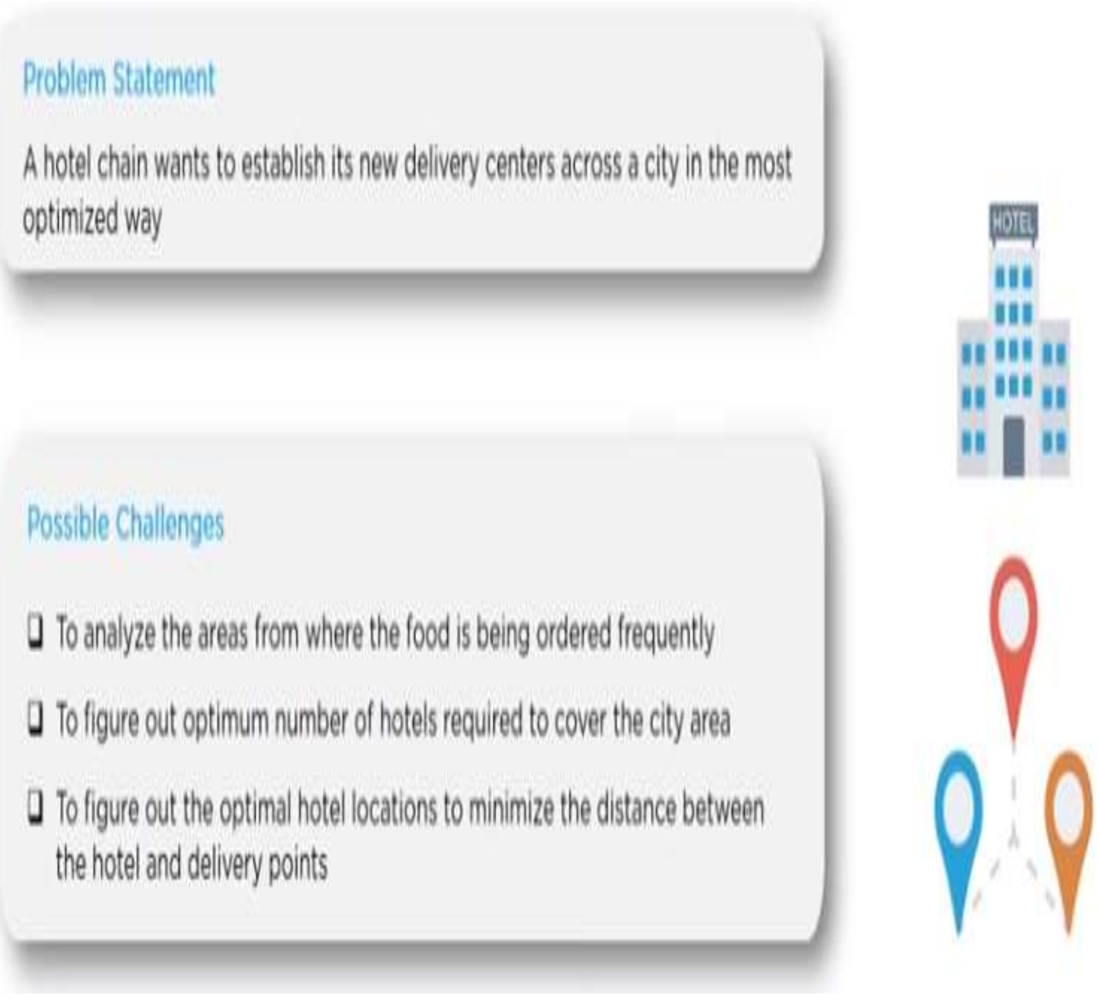
Un Supervised Learning



Step 4: Repeat previous two steps iteratively till the cluster centroids stop changing thei positions and become static

Step 5: Once the clusters become static then k-means clustering algorithm is said to be converged





Problem Statement

A hotel chain wants to establish its new delivery centers across a city in the most optimized way

Possible Challenges

- ❑ To analyze the areas from where the food is being ordered frequently
- ❑ To figure out optimum number of hotels required to cover the city area
- ❑ To figure out the optimal hotel locations to minimize the distance between the hotel and delivery points

Software Requirements

Image Processing Packages in Python

- Scikit image
  - scikit-image is a collection of algorithms for image processing.
  - scikit-image uses NumPy arrays as image objects by transforming the original pictures.
  - It is a fairly simple and straightforward library even for those who are new to Python’s ecosystem
  - It includes algorithms for:
    - Segmentation,
    - Geometric transformations,
    - Color space manipulation,
    - Analysis,
    - Filtering,
    - Morphology,
    - Feature detection, and more

Reference: https://analyticsindiamag.com/top-8-image-processing-libraries-in-python/

Image Processing Packages in Python

- OpenCV
  - OpenCV has become a popular library is focused on image processing, face detection, object detection.
  - Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc

Reference and Logo image : https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html

Image Processing Packages in Python

- Scipy
  - scipy is another of Python’s core scientific modules
  - Scipy is used for mathematical and scientific computations but can also perform multi- dimensional image processing.
  - In particular, the submodule scipy.ndimage provides functions operating on n- dimensional NumPy arrays.
  - Scipy offers the most commonly used image processing operations like:
    - Reading Images
    - Image Segmentation
    - Convolution
    - Face Detection
    - Feature Extraction and so on.

Reference and Logo image : https://neptune.ai/blog/image-processing-python-libraries-for-machine-learning

Image Processing Packages in Python

- Numpy
  - Numpy is one of the core libraries in Python programming and provides support for arrays.
  - An image is essentially an array of pixel values where each pixel is represented by 1 (greyscale) or 3 (RGB) values.
  - Therefore, NumPy can easily perform tasks such as image cropping, masking, or manipulation of pixel values.

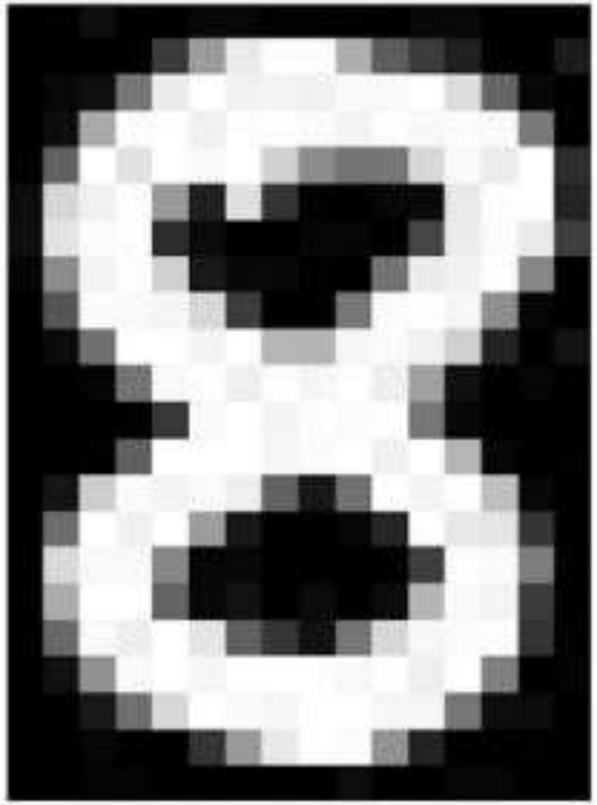
Reference and Logo image : https://neptune.ai/blog/image-processing-python-libraries-for-machine-learning

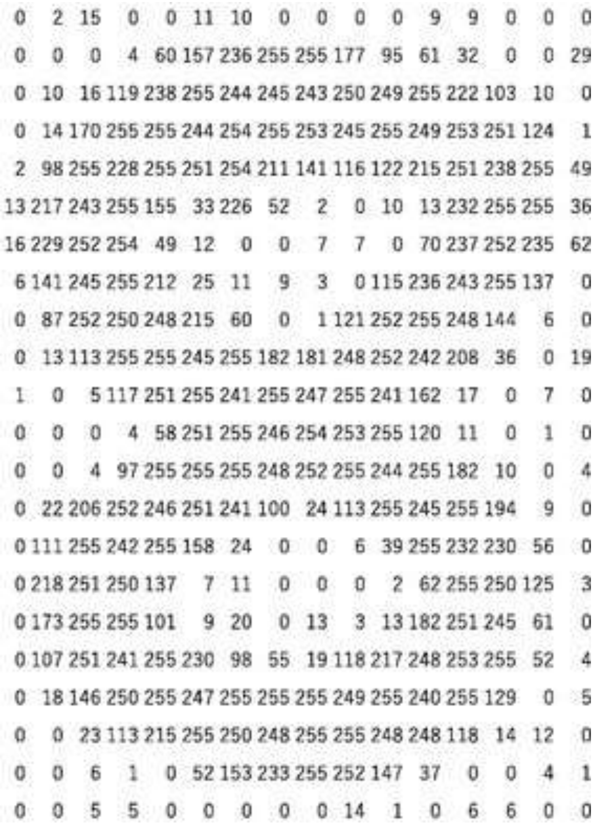
Software Requirements

- Anaconda
  - https://www.anaconda.com/products/individual
- Install numpy, scipy, matplotlib, scikit image , scikit learn packages
  - conda install -c anaconda numpy
  - conda install -c anaconda pandas
  - conda install -c anaconda scipy
  - conda install -c conda-forge matplotlib
  - conda install -c anaconda scikit-image
  - conda install -c anaconda scikit-learn
- After installation, open your jupyter notebook

Image Feature Extraction

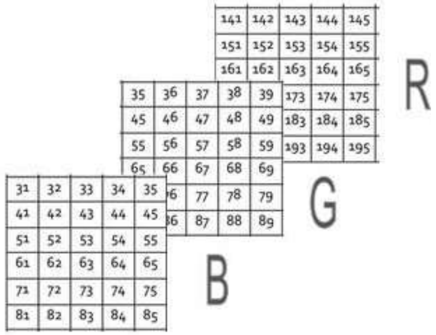
How do Machines Store Images?





0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

Colour Image



R

141	142	143	144	145
151	152	153	154	155
161	162	163	164	165
		173	174	175
		183	184	185
		193	194	195

G

35	36	37	38	39
45	46	47	48	49
55	56	57	58	59
65	66	67	68	69
	6	77	78	79
	6	87	88	89

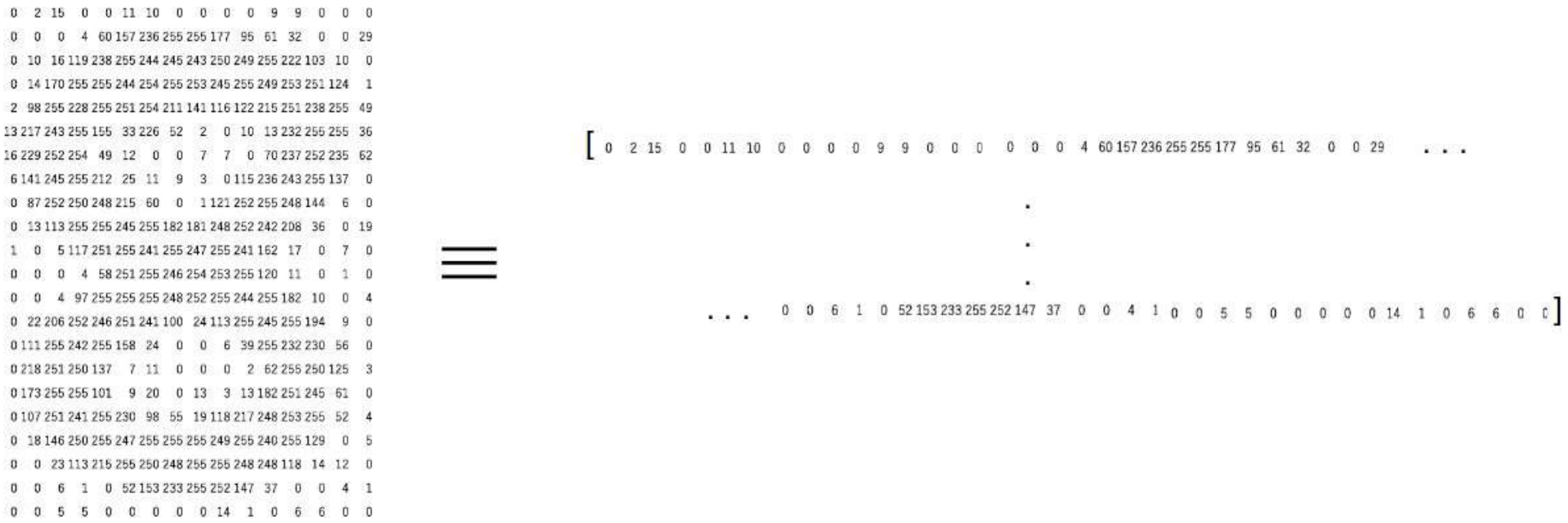
B

31	32	33	34	35
41	42	43	44	45
51	52	53	54	55
61	62	63	64	65
71	72	73	74	75
81	82	83	84	85

Image Source: https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python

Feature Extraction

- Method #1: Grayscale Pixel Values as Features
  - The simplest way to create features from an image is to use these raw pixel values as separate features.



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

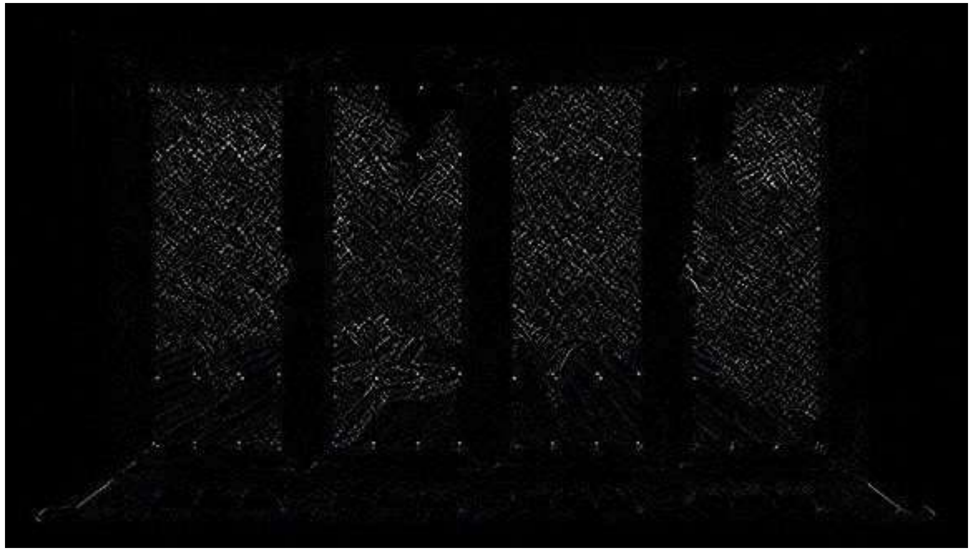
≡

```
[ 0 2 15 0 0 11 10 0 0 0 0 9 9 0 0 0  0 0 0 4 60 157 236 255 255 177 95 61 32 0 0 29   . . .
                                        .
                                        .
                                        .
        . . .   0 0 6 1 0 52 153 233 255 252 147 37 0 0 4 1 0 0 5 5 0 0 0 0 0 14 1 0 6 6 0 0 ]
```

Reference & Image Source: https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/

Feature Extraction

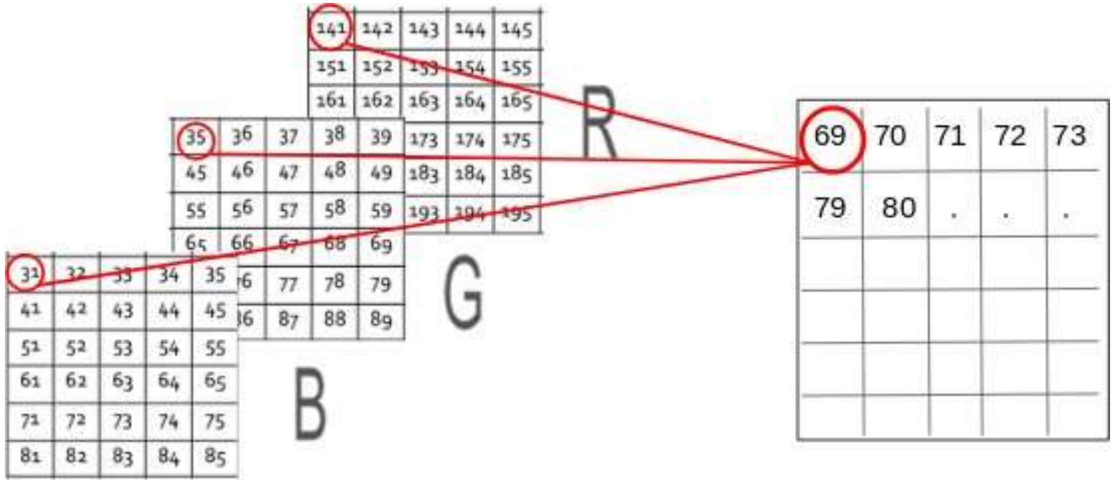
- Method #2: Extracting Edge Features
  - Extract edges as features and use that as the input for the model.



Reference & Image Source: https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/

Feature Extraction

- Method #3: Mean Pixel Value of Channels
  - Instead of using the pixel values from the three channels separately, we can generate a new matrix that has the mean value of pixels from all three channels.



Reference & Image Source: https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/

Scikit -Image

https://scikit-image.org/docs/stable/index.html

Image Processing Packages in scikit-image

- Read the image

```
from skimage import io
image = io.imread(‘image path’)
Image_list= io.imread_collection(“*.jpg")
```

- Converting images to greyscale

```
from skimage import io

image =io.imread('image path', as_grey=True)
```

```
from skimage import io

from skimage.color import rgb2gray

image = io.imread (‘image path’)

img_new = rgb2gray(image)
```

Reference: https://kapernikov.com/tutorial-image-classification-with-scikit-learn/

Image Processing Packages in scikit-image

- **Resizing Image**

```
from skimage.transform import resize

img = imread('images.jpeg')

img_resized = resize(img, (300, 300)) #resize image
```

- **Save Images**

```
from skimage import io

from skimage.color import rgb2gray  image = io.imread (‘image path’)  img_new =

rgb2gray(image)  io.imsave('local_logo.png', img_new)
```

Reference: https://kapernikov.com/tutorial-image-classification-with-scikit-learn/

Image Processing Packages in scikit-image

- **Displaying Image**

```
from skimage import io

img = io.imread('images.jpeg')

io.imshow(img)
```

- **Save as Images**

```
from skimage import io

logo = io.imread('logo.png’)

io.imsave('local_logo.png', logo)
```

Reference: https://kapernikov.com/tutorial-image-classification-with-scikit-learn/

Scikit -Learn

https://scikit-image.org/docs/stable/index.html

Image Processing Packages in scikit-image

- Load the Dataset

```
import pandas as pd  url ="mydata/sales.csv"

sales_data = pd.read_csv(url)
```

- Separate features and target variables

```
data = sales_data[cols]

target = sales_data['Opportunity Result']
```

Reference: https://kapernikov.com/tutorial-image-classification-with-scikit-learn/

Image Processing Packages in scikit-image

- **Split Training Set and Test Set**

```
from sklearn.model_selection import train_test_split
data_train, data_test, target_train, target_test = train_test_split(data,target, test_size = 0.30,
random_state = 10)
```

- **Build and Train a Model**

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()  model.fit(data_train, target_train)  pred = model.predict(data_test)
```

Reference: https://kapernikov.com/tutorial-image-classification-with-scikit-learn/

Image Processing Packages in scikit-image

- <u>Other Models</u>

```
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors=5)
```

```
from sklearn.svm import SVC

model = SVC(gamma='auto’)
```

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100)
```

Reference: https://kapernikov.com/tutorial-image-classification-with-scikit-learn/

Image Processing Packages in scikit-image

- <u>Accuracy</u>

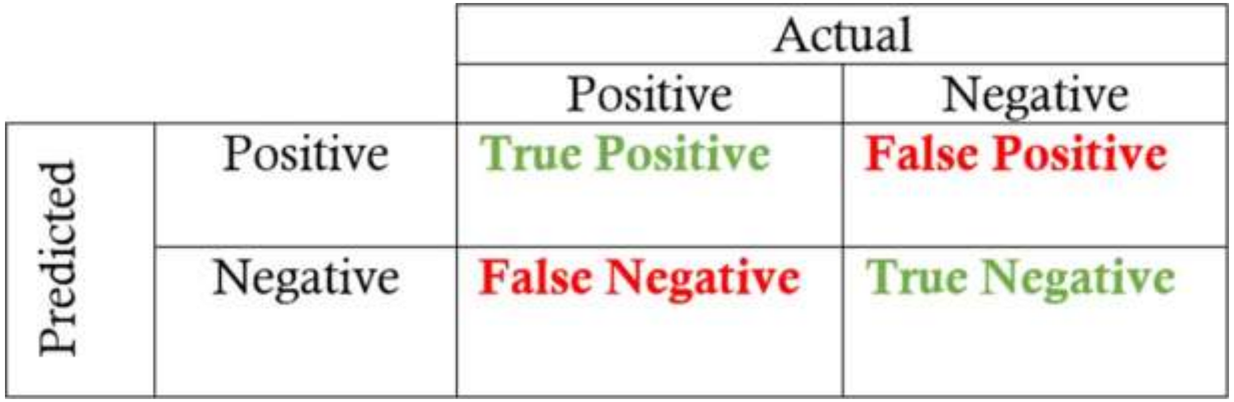
```
from sklearn.metrics import accuracy_score

print("Naive-Bayes accuracy : ",accuracy_score(target_test, pred, normalize = True))
```

Reference: https://kapernikov.com/tutorial-image-classification-with-scikit-learn/

Image Processing Packages in scikit-image

- <u>Accuracy</u>

- True positive:The prediction is correct and the actual value is positive
- False positive:The prediction is wrong and the actual value is positive
- True negative:The prediction is correct and the actual value is negative
- False negative:The prediction is wrong and the actual value is negative



		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Reference: https://kapernikov.com/tutorial-image-classification-with-scikit-learn/

THANK YOU