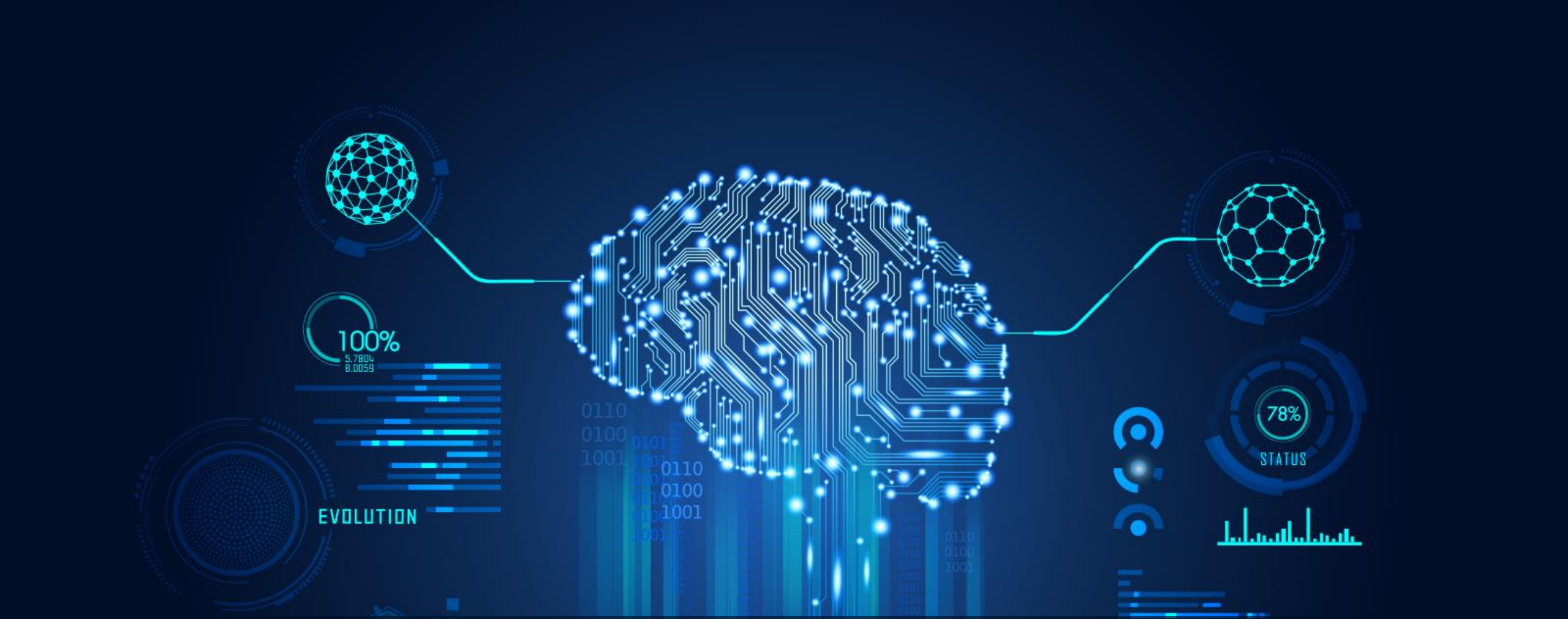


MACHINE LEARNING

Mohamad Irwan Afandi

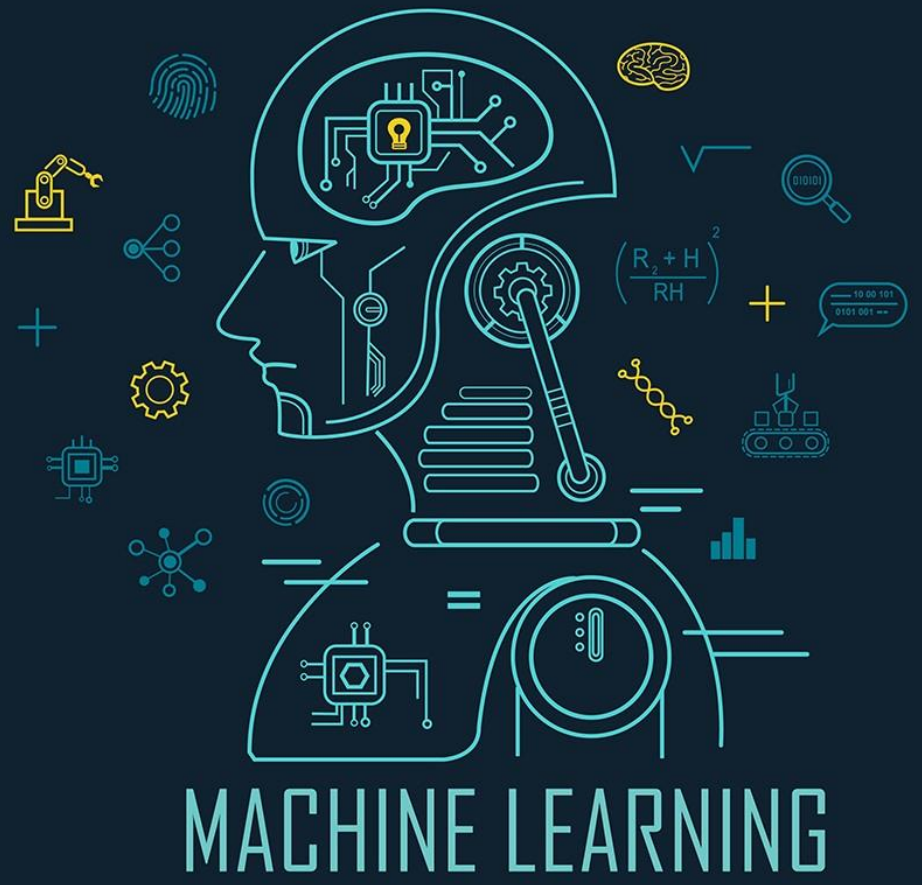
WHAT IS MACHINE LEARNING ?

Machine Learning is about making machines get better at some task by learning from data, instead of having to explicitly code rules.

How do we know that the system is working properly? Test the system with the data that you didn't use in training process, and see the accuracy.

Machine Learning Example:

- Email Spam Detection
- Customer Segmentation
- Fraud Detection
- Self Driving Car
- Image Recognition, etc.



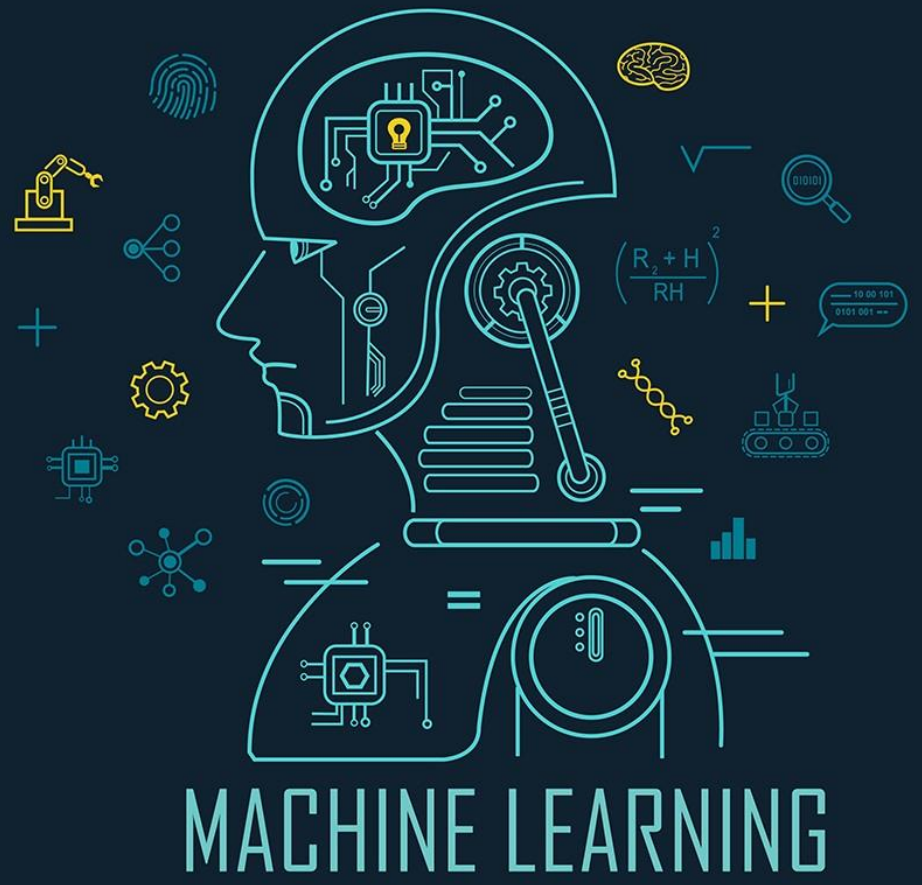
TYPE OF MACHINE LEARNING

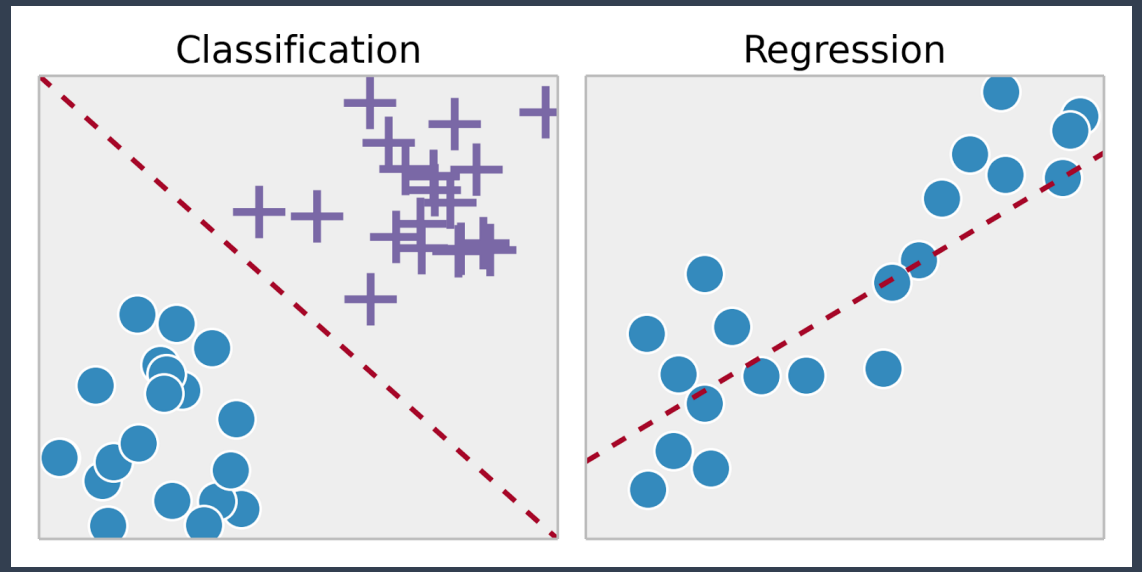
There are 4 types of machine learning:

- Supervised Learning (**data with label**)
- Unsupervised Learning (**data without label**)
- Semi-supervised Learning (**data with label or without label**)
- Reinforcement Learning (**using reward**)

Solve the problem in **supervised learning**:

- **Classification** : process of predicting class or category from observed values, the result is categorical data.
- **Regression** : is to predict output labels or responses which are continues numeric values.





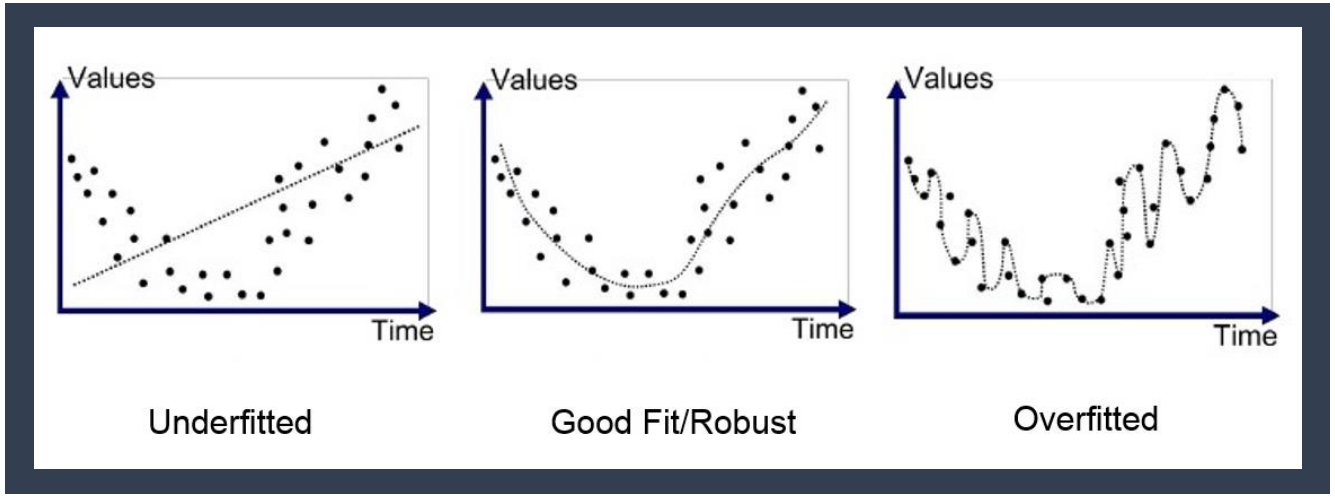
Classification vs Regression

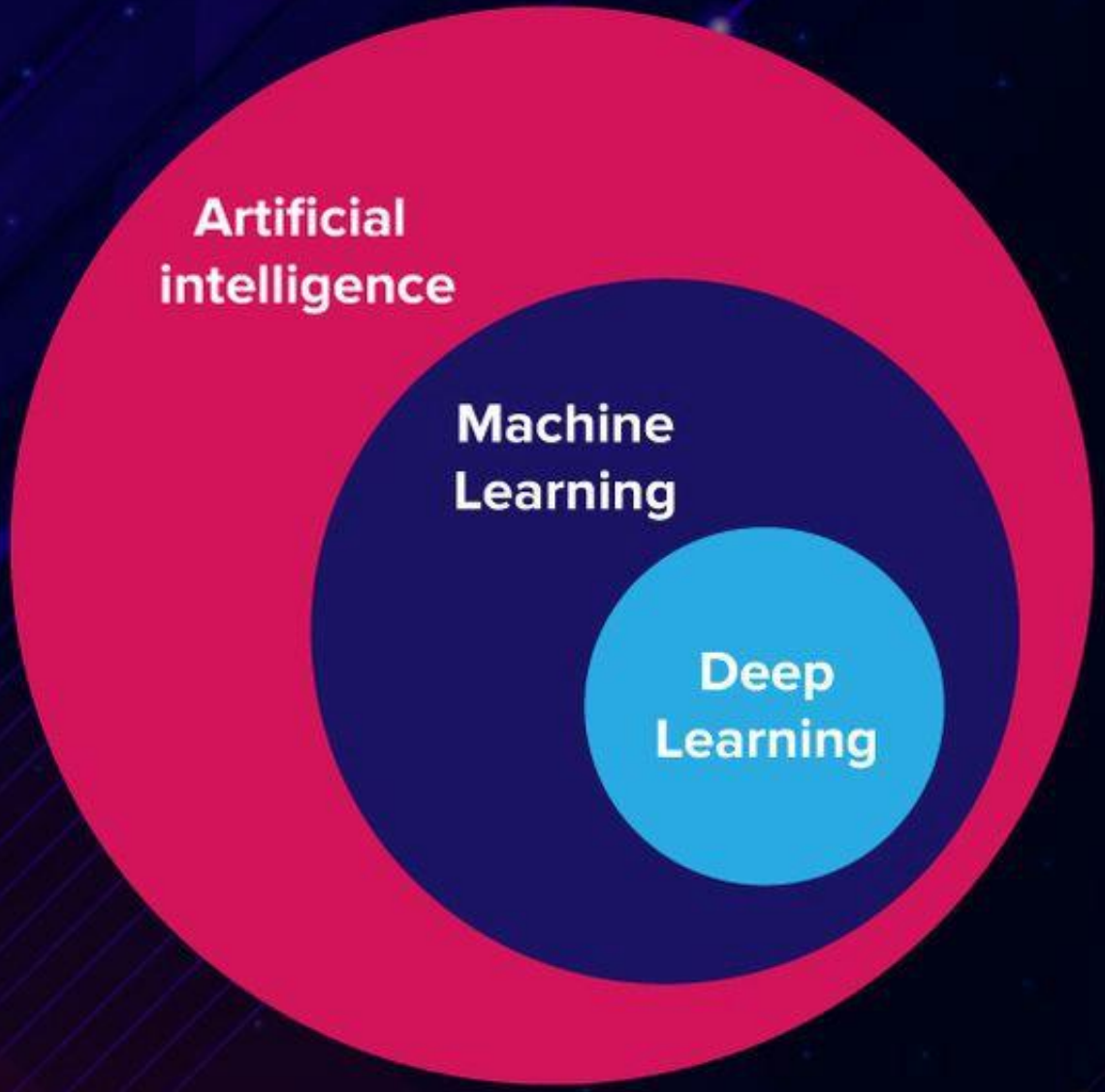
- **Classification** : predict the categorical value like yes or no, class A or class B, etc.
- **Regression** : Get the continue value from the data (dependent variable) based on another data (independent). Like salary or price

Main Problem

- **Underfitting** is the case where ***the model has not learned enough*** from the training data, resulting in low generalization and unreliable predictions. Factors: *we have lass data, build a linear modal with non linear data. Solution add more dataset.*

- **Overfitting** is the case where the model ***fits too well to the training set***. *Overfitting occurs if the model or algorithm shows low bias but high variance. Solution split and test data or using cross-validation.*





FEEL THE DIFFERENCE

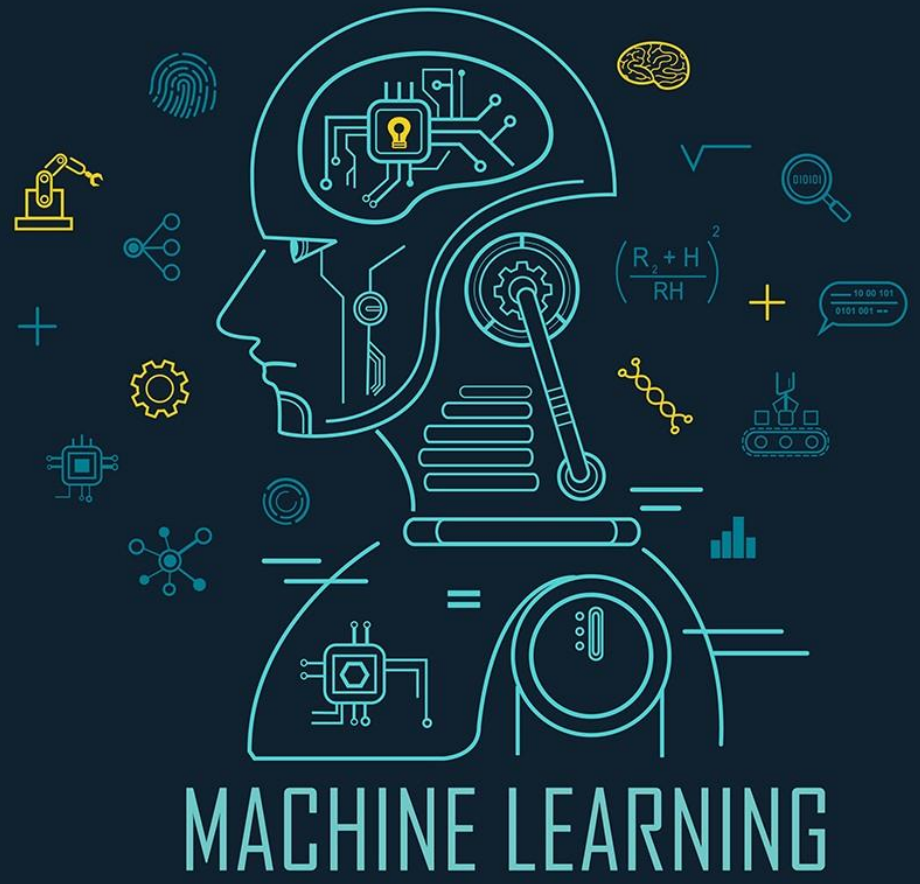
MACHINE ALGORITHM

Supervised Learning

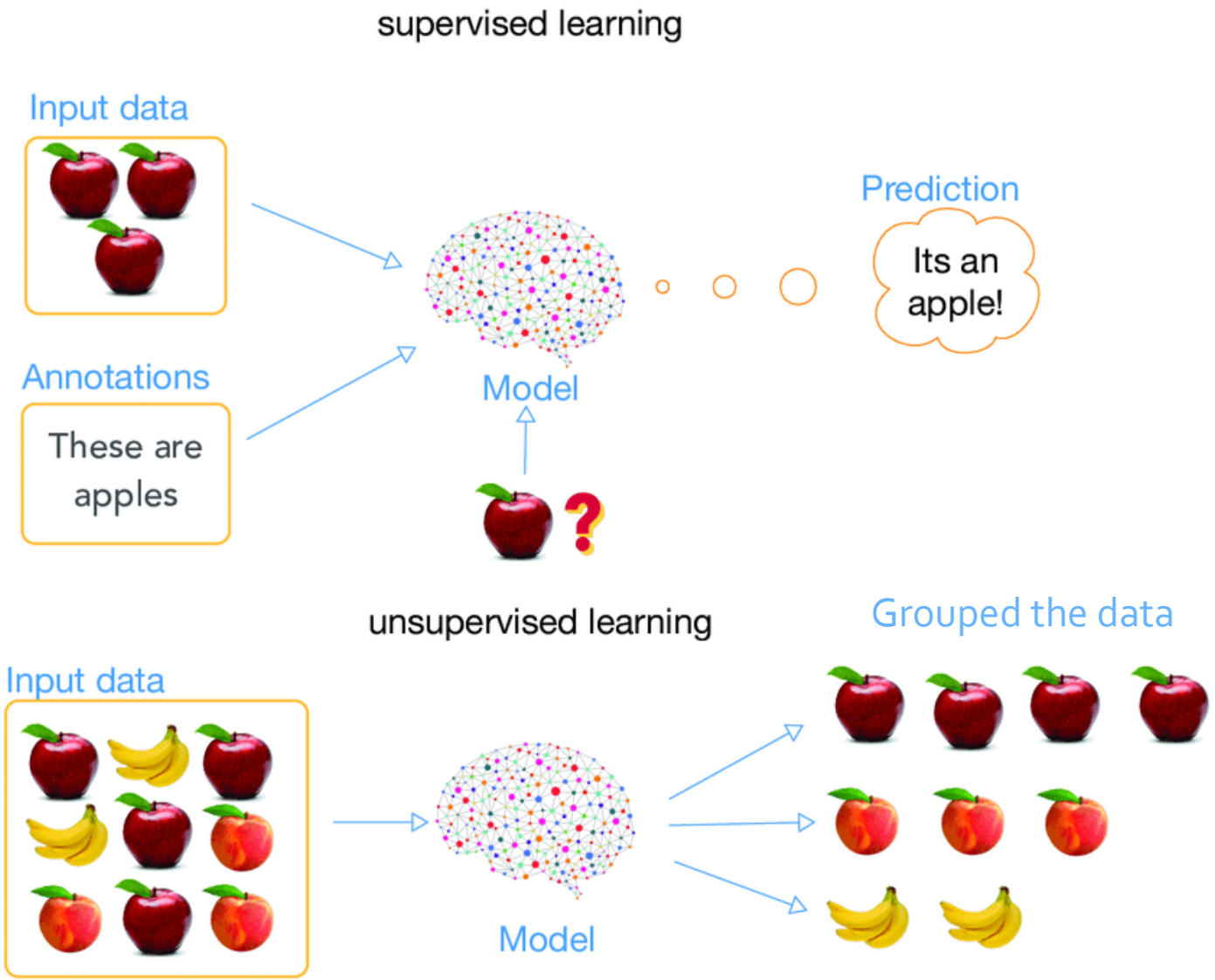
- Classification
  - Logistic Regression
  - K Nearest Neighbors (KNN)
  - Support Vector Machine (SVM)
  - Naïve Bayes (NB)
  - Decision Tree Classification
  - Random Forest Classification
  - Neural Networks
- Regression
  - Simple Linear Regression
  - Support Vector Regression
  - Decision Tree Regression
  - Random Forest Regression
  - Neural Networks

Unsupervised Learning

- K-Means, C-means, Fuzzy
- Hierarchical
- Gaussian Mixture
- Hidden Markov Model
- Neural Networks



How it Works..?


supervised learning
Input data
Annotations
These are apples
Model
Prediction
Its an apple!
unsupervised learning
Input data
Model
Grouped the data

Machine Learning

Build Machine Learning Model

- Import necessary python package
- Load the dataset
- Preprocessing and EDA
- Organizing data into training and test sets (only on supervised)
- Build machine learning model
- Evaluate the model

Details >>

Build Machine Learning Model

1 *Import Necessary Python Package*

Most of programmer puts it (all import code) at the beginning section

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
sns.set()
```

2 *Load the Dataset*

You can load dataset from csv, xlxs, txt, html, json, etc.

```
df = pd.read_csv("data.csv", na_values='?')
df.head()
```

3 *Preprocessing and EDA*

Generally you will dealing with missing value, data anomalies, outlier, imbalance data, scaling, visualization, etc.

Missing value handling

```
df = df.apply(lambda x: x.fillna(x.median())
              if x.dtype.kind in 'iuf' else
              x.fillna(df['num-of-doors'].mode()[0]))
```

Outlier Handling

```
def getIQR(feature):
  Q1 = feature.quantile(0.25)
  Q3 = feature.quantile(0.75)
  IQR = Q3-Q1
  return Q1-(1.5*IQR), Ub = Q3+(1.5*IQR) #return Q1 and Q3

def outlier_handling(lb, ub, x):
  if x < lb:
    return lb #set value as lower boundary
  if x > ub:
    return lb #set value as upper boundary
lb, ub = getIQR(df.bmi)
df[(df["bmi"]=>lb) | (df["bmi"]<=up)]
df["bmi"] = df["bmi"].apply(outlier_handling)
```

Build Machine Learning Model

4 *Organizing the data into training and test sets*

To avoid overfitting and know the accuracy score (only on supervised)

```
X_train, X_test, y_train, y_test = train_test_split(
    df_feature_scaler, df_label, test_size=0.2, random_state=27)
```

5 *Build a model*

Use hyperparameter tuning to get the best parameter (only on supervised)

Supervised

```
model = RandomForestClassifier(class_weight = 'balanced',
                               criterion = 'gini',
                               max_features = 'auto',
                               min_samples_leaf = 3,
                               min_samples_split = 4,
                               min_weight_fraction_leaf = 0.0,
                               n_estimators = 125)
model.fit(X_train, y_train)
```

Unsupervised

```
kmeans = KMeans(4)
kmeans.fit(feature4)
label = kmeans.predict(feature4)
rfm4['Label'] = label
rfm4.head()
```

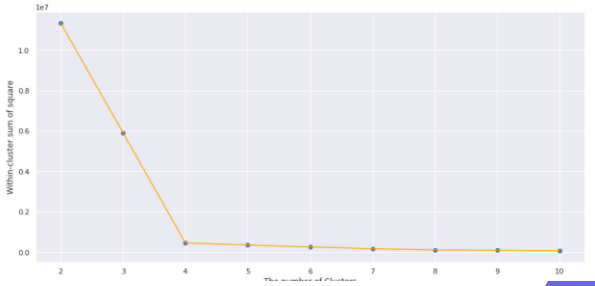
6 *Evaluate the model*

If the accuracy is good enough you can deploy, if isn't enough back to preprocessing or try another mode

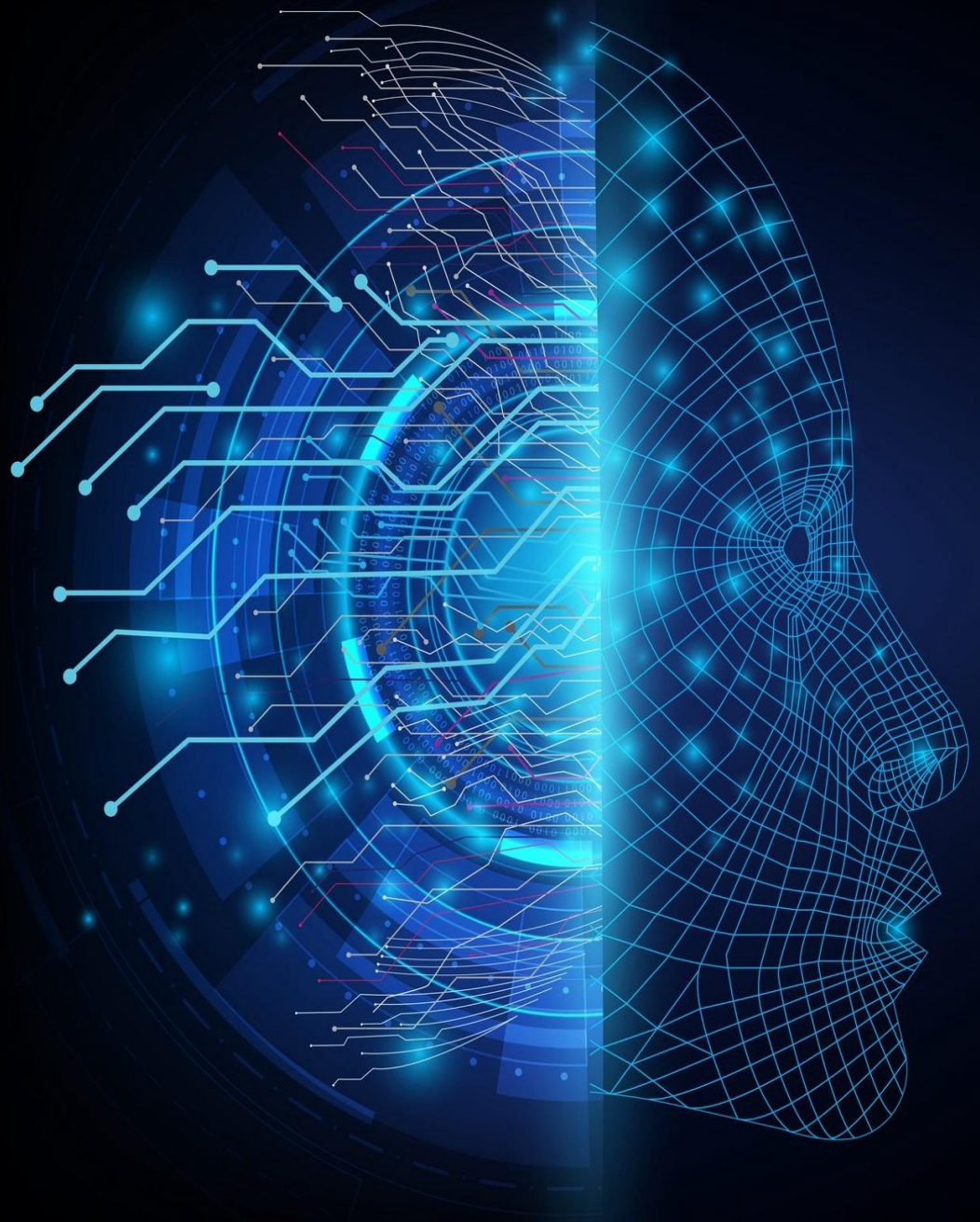
```
test_pred = model.predict(X_test)
accuracy_score(test_pred, y_test)
```

```
0.7732919254658385
```

```
from sklearn.cluster import KMeans
wcss = []
for i in range (2,11):
    kmeans = KMeans(i)
    kmeans.fit(feature4)
    wcss.append(kmeans.inertia_)
```



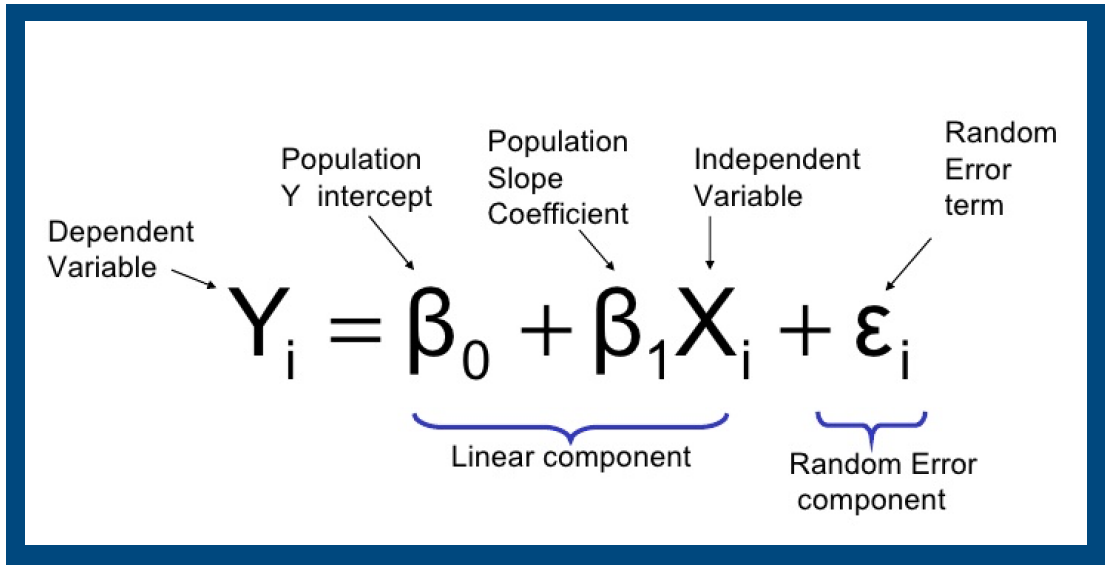
Choose the best K

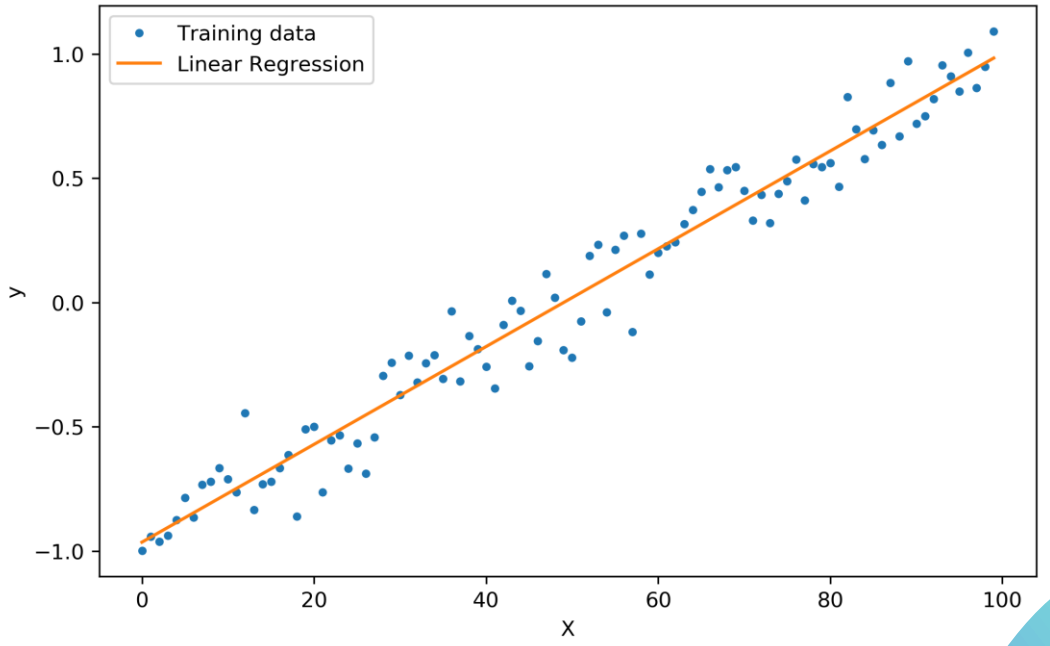


SUPERVISED LEARNING

1 Simple Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable (*independent*), and the other is considered to be a *dependent* variable.





Dependent : it is something that depends on other factors.
Independent : is the variable the experimenter changes or controls and is assumed to have a direct effect on the *dependent variable*.

implementation Example:

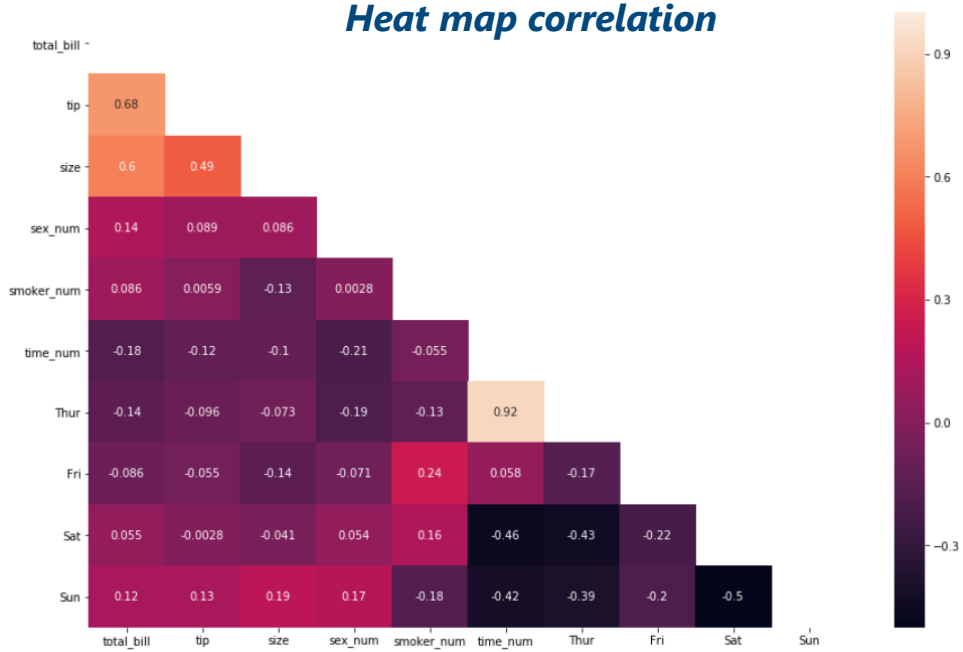
1. House Price Prediction
2. Salary Prediction

2 Multivariate Regression

Multivariate Regression *is supervised machine learning algorithm involving multiple data variables for analysis. A Multivariate regression is an extension of multiple regression with one dependent variable and multiple independent variables. Based on the number of independent variables, we try to predict the output. After get the values use hit map to see the data correlation.*

$$Y = mx1 + mx2 + mx3 + b$$

Heat map correlation

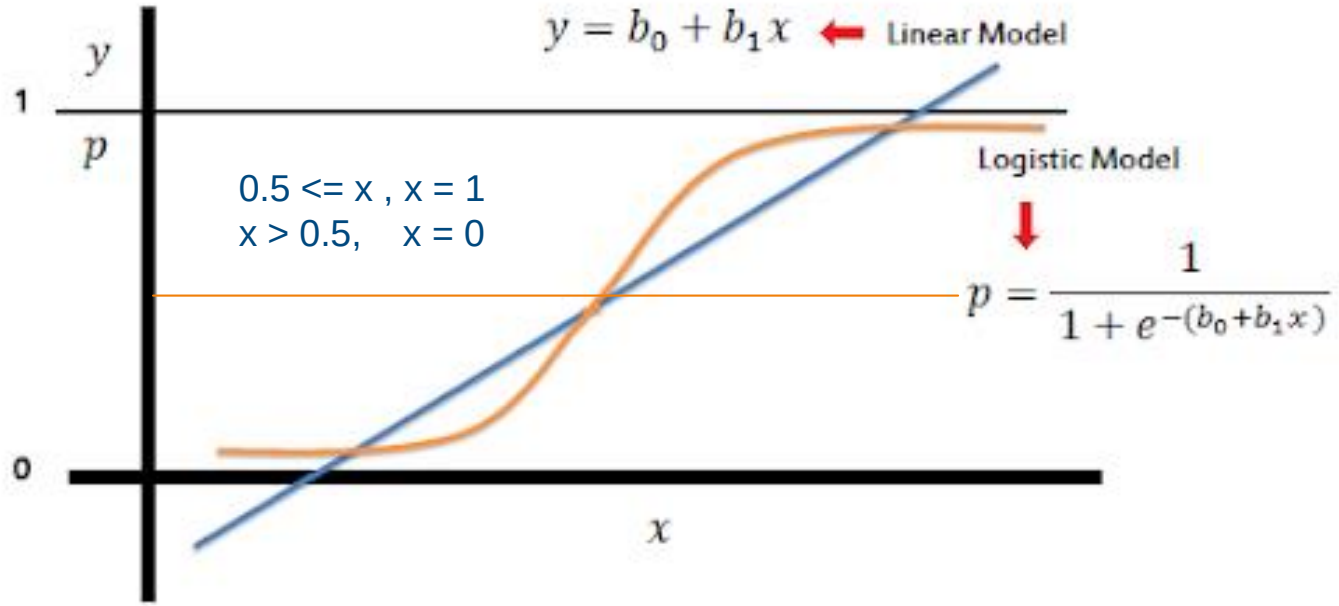


implementation Example:

1. How much Company has to pay to a new hire
2. Predict the total crop yield expected
3. Predict the GDP growth

3 Logistic Regression

Logistic regression *is a supervised learning classification algorithm used to predict category of the variable based on the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. Type of logistic regression: binary (0 or 1), categorical (more than 2 : A, B and C), and ordinal (ordered value more than 2: easy, medium, expert)*



If the logit value is not 0 or 1, see the probability because we can define the class based on it***. If the probability less then 0.5 it will classified to 0 and more than equal to 0.5 will classified to 1 (default library)***

implementation Example:

1. Email Spam Detection
2. Online transaction for Fraud detection

4 Naïve Bayes

Naïve Bayes *(using baye's theorem) is that the presence of a feature in a class is independent to the presence of any other feature in the same class. For example, a phone may be considered as smart if it is having touch screen, internet facility, good camera etc. Though all these features are dependent on each other, they contribute independently to the probability of that the phone is a smart phone.*

Likelihood

Class Prior Probability

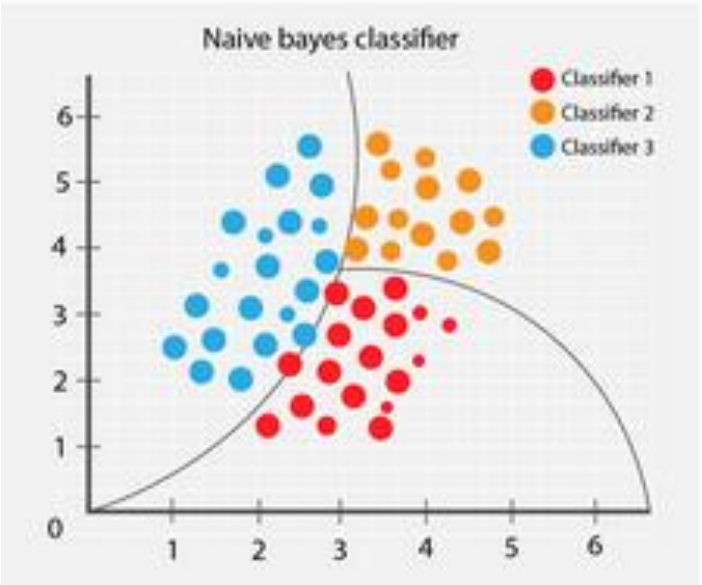
$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

The likelihood that an event (c) will happen given that another event (x) has already happened

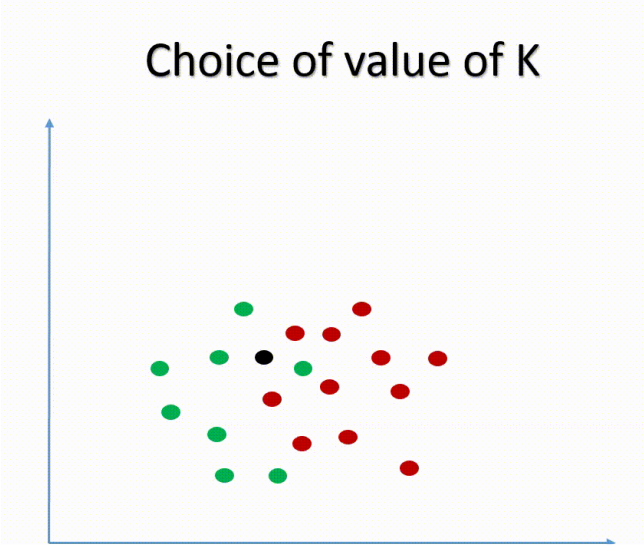


Implementation Example:

1. Recommendation System
2. Text Classification
3. Multiclass prediction

5 K Nearest Neighbour

KNN works *by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).*



Advantages:

1. Quick calculation time and simple
2. Versatile (regression and classification)
3. High Accuracy

Disadvantages:

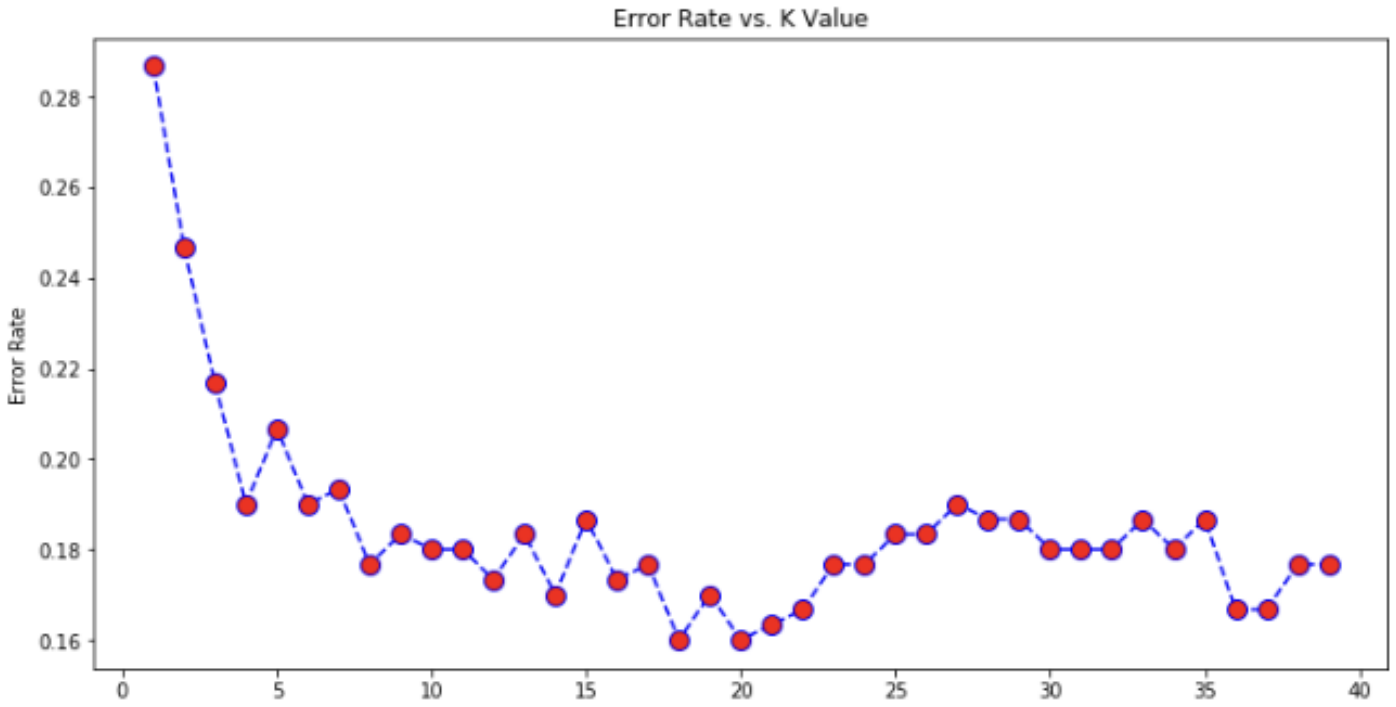
1. Accuracy depends on the quality of the data
2. With large data, the prediction stage might be slow

implementation Example:

1. predict the credit rating of customers
2. predict whether the loan is safe or risky (banking)
3. classifying potential voters in two classes will vote or won't vote (political)

5 K Nearest Neighbour

Choose the best number of K using Elbow method



Get the error rate for every K

```
error_rate = []

# Will take some time
for i in range(1,40):

 knn = KNeighborsClassifier(n_neighbors=i)
 knn.fit(X_train,y_train)
 pred_i = knn.predict(X_test)
 error_rate.append(np.mean(pred_i != y_test))
```

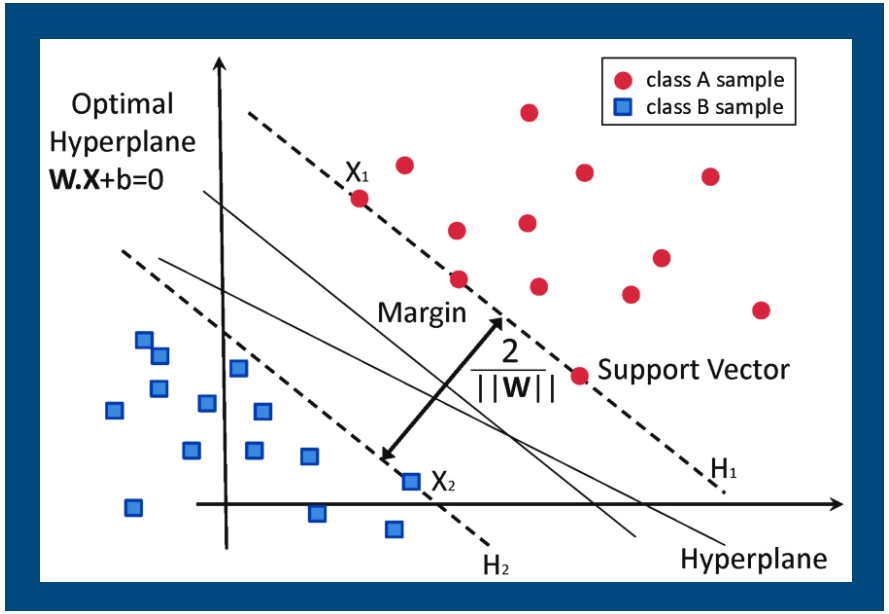
Get the error rate for every K

```
plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed',
marker='o',
 markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

6 Support Vector Machine

Classification & Regression

SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).



Advantages:

1. works well when there is clear margin of separation between classes.
2. more effective in high dimensional spaces.
3. Memory efficient

Disadvantages:

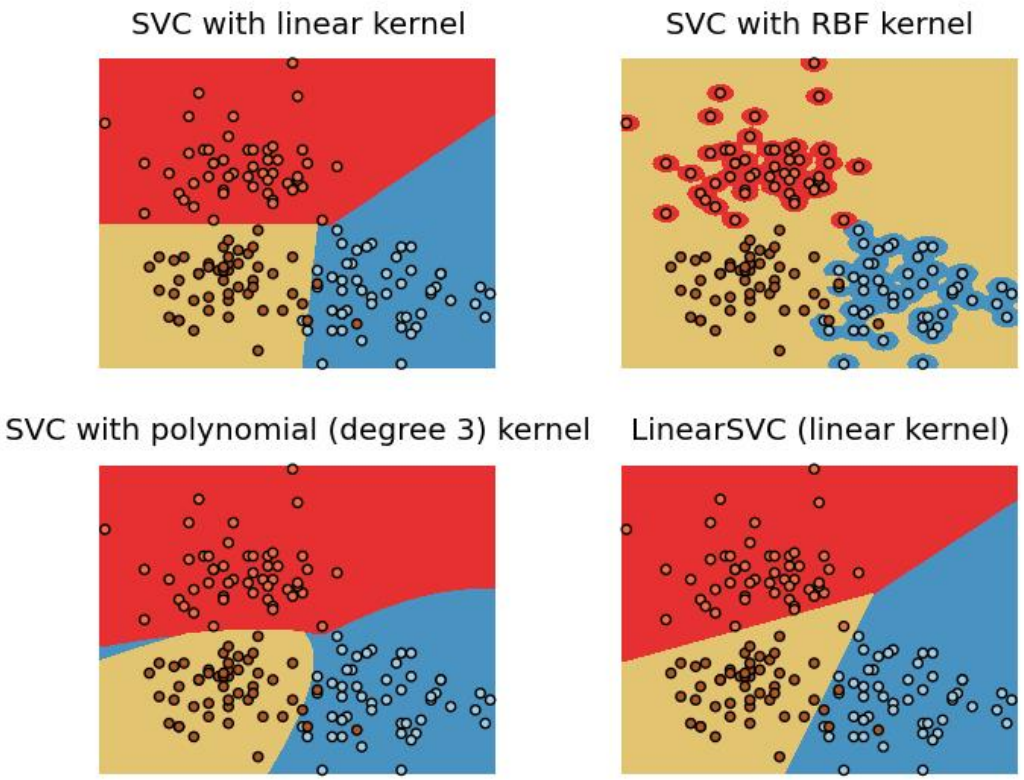
1. not suitable for large data.
2. does not perform very well, when the data set has more noise

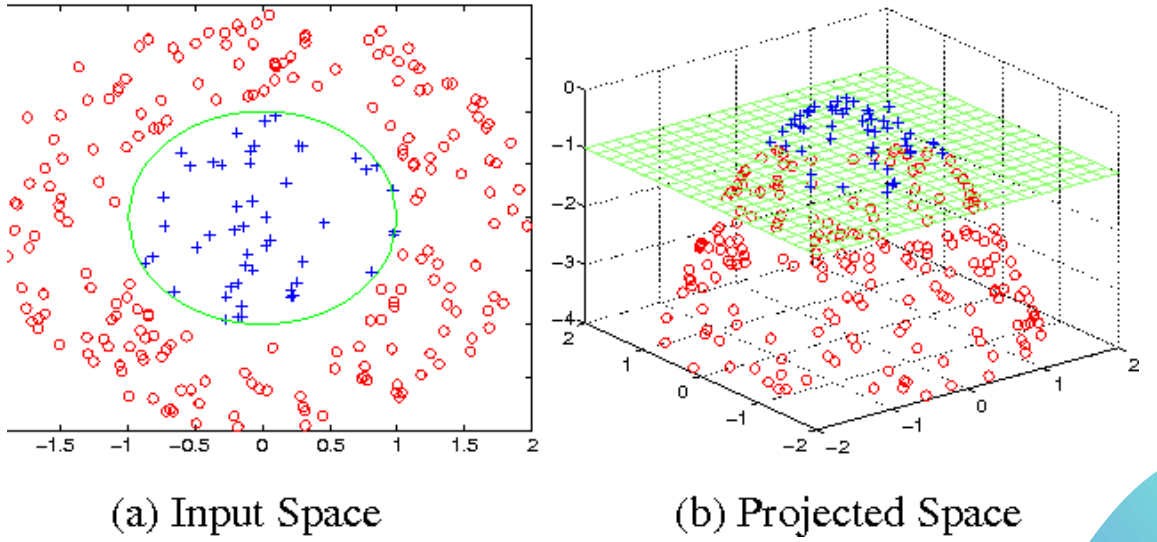
implementation Example:

1. Sentiment Analysis
2. Image Classification

6 Support Vector Machine

SVM Kernel Trick usually use to choose the right hyperplane for separate the data. You have to know that the class data distribution cannot be separated only by linear line. SVM Kernel : Linear, Radial Basis Function (RBF), Sigmoid, Polynomial.

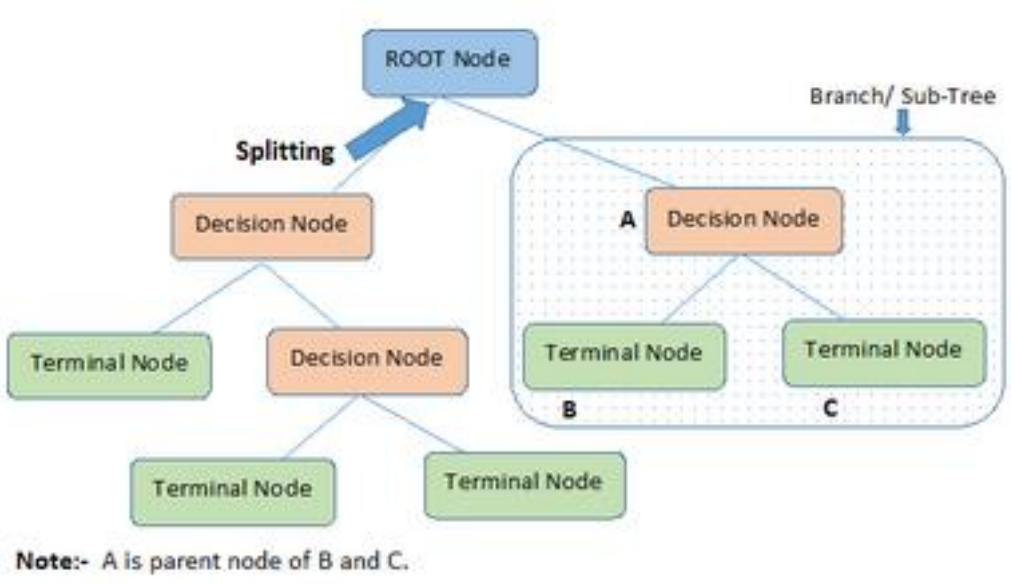




(a) Input Space (b) Projected Space

7 Decision Tree

A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g. whether a coin flip comes up heads or tails) , each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels and the paths from root to leaf represent classification rules.



Advantages:

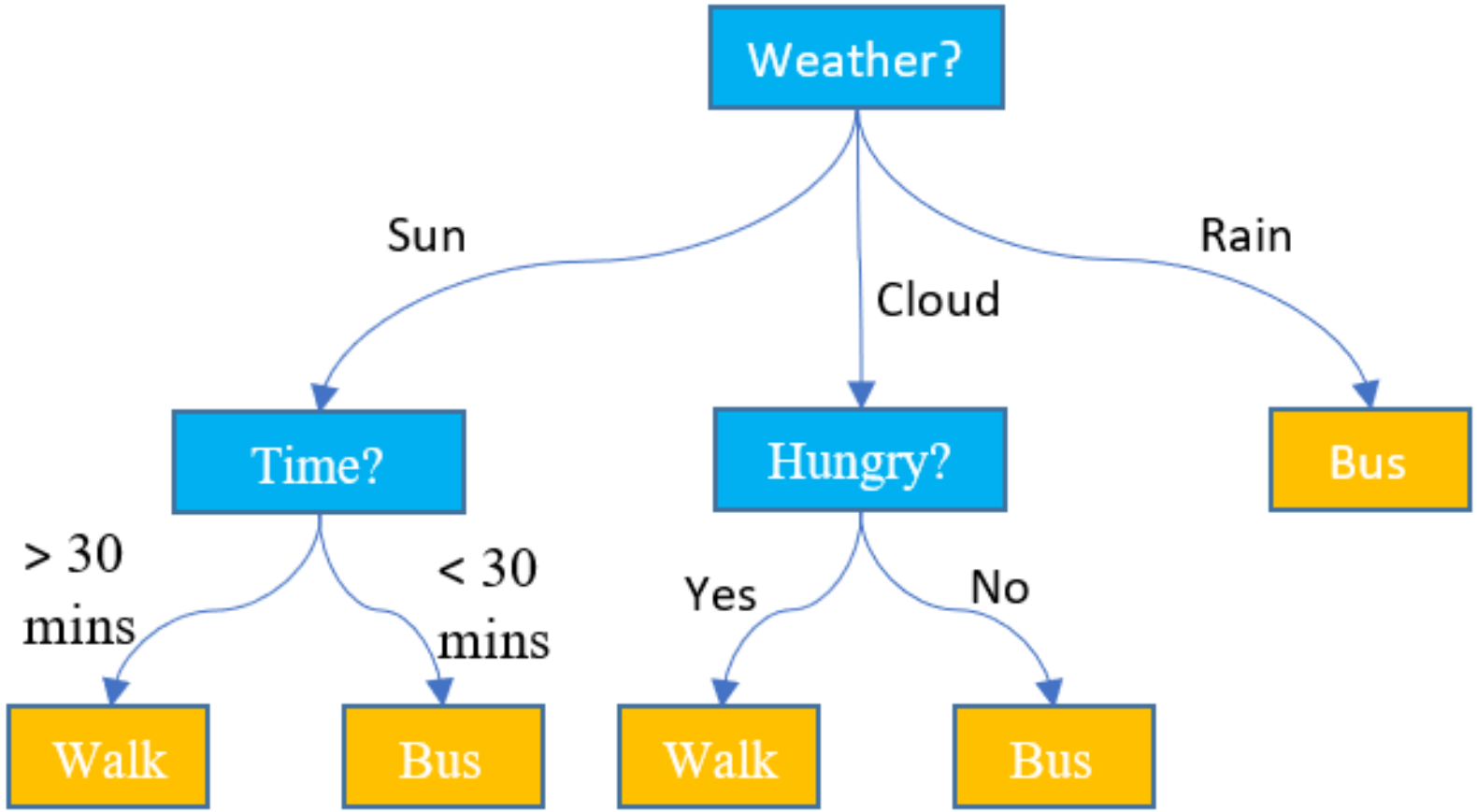
1. Easy to Understand
2. Less data cleaning required
3. Data type is not a constraint (handle both numerical and categorical variable)

Disadvantages:

1. Over fitting
2. Not fit for continuous variables

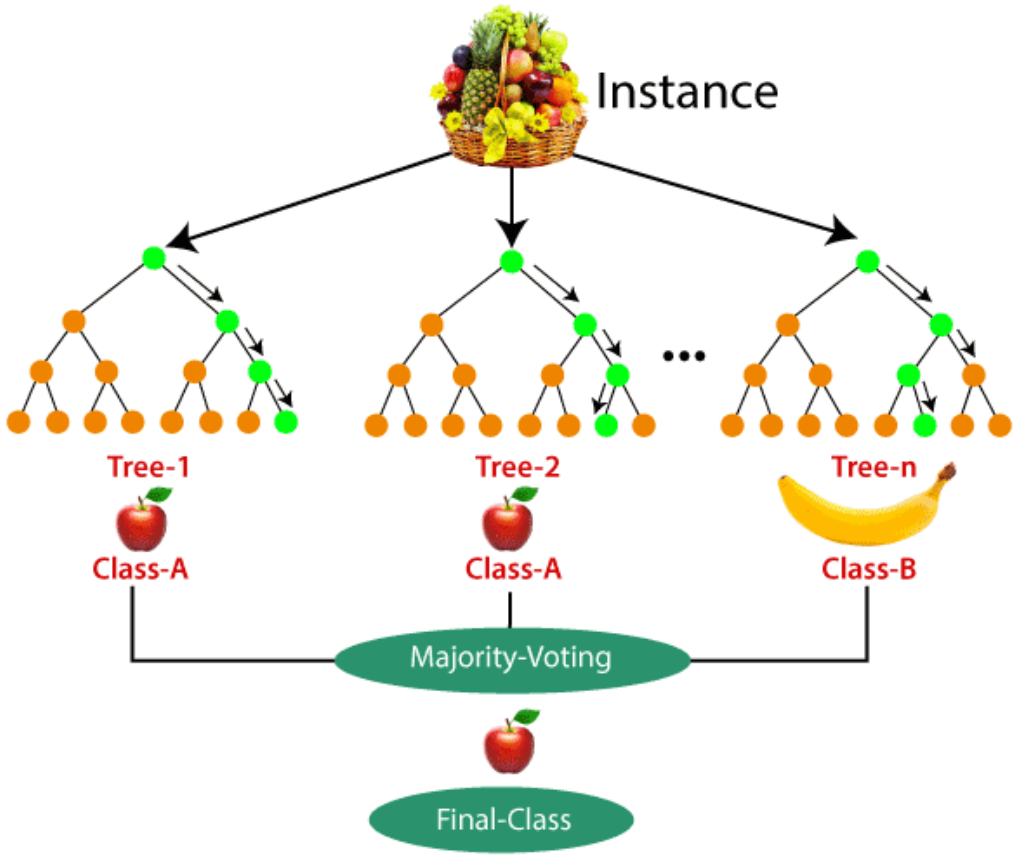
7 Decision Tree

Determine what should I do to back home



8 Random Forest

***Random forest**, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.*



Advantages:

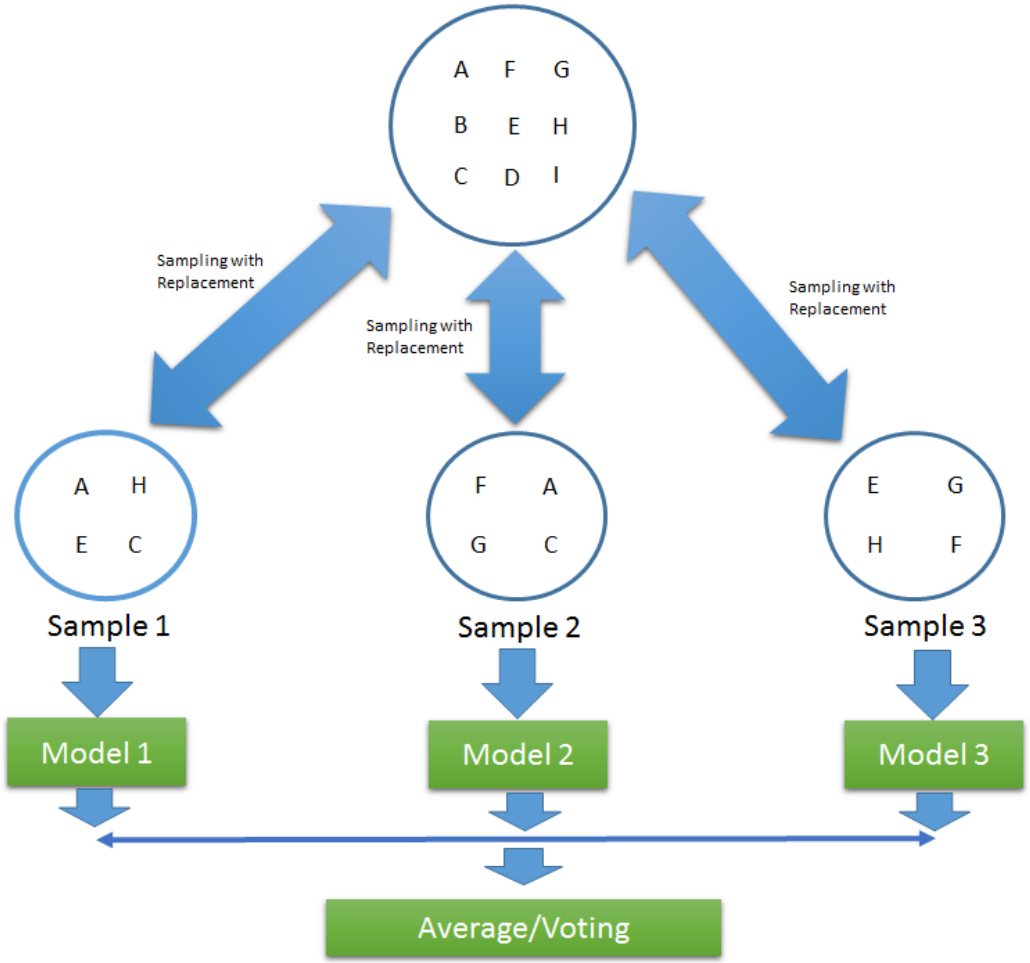
1. It reduces overfitting problem in decision trees and also reduces the variance
2. Random Forest can automatically handle missing values.

Disadvantages:

1. High Complexity
2. Longer Training Period

8 Random Forest

Classification & Regression

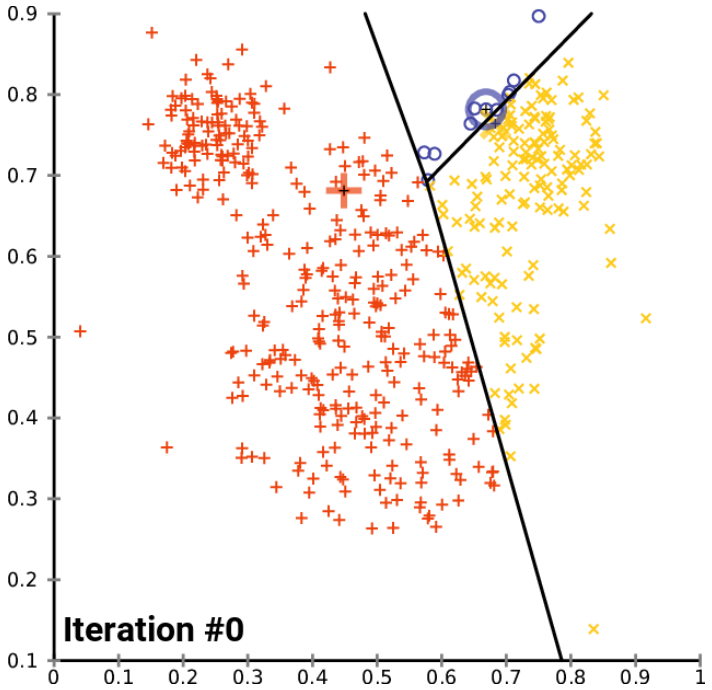


To generate the three from data training, RF using Bagging method. ***Bagging is an*** *approach is to use the same training algorithm for every predictor, but to train them on different random subsets of the training set. When sampling is performed with replacement, this method is called bagging (bootstrap aggregating).*

UNSUPERVISED LEARNING

1 K-Means

K-means *is a centroid-based algorithm, or a distance-based algorithm, where we calculate the distances to assign a point to a cluster. In K-Means, each cluster is associated with a centroid. With K-Means, we will grouped the data based on the nearest distance in with (x) centroid.*



Advantages:

1. Easy to understand and implement
2. If we have large number of variables then, K-means would be faster than Hierarchical clustering.

Disadvantages:

1. It is a bit difficult to predict the number of clusters
2. Order of data will have strong impact on the final output.

implementation Example:

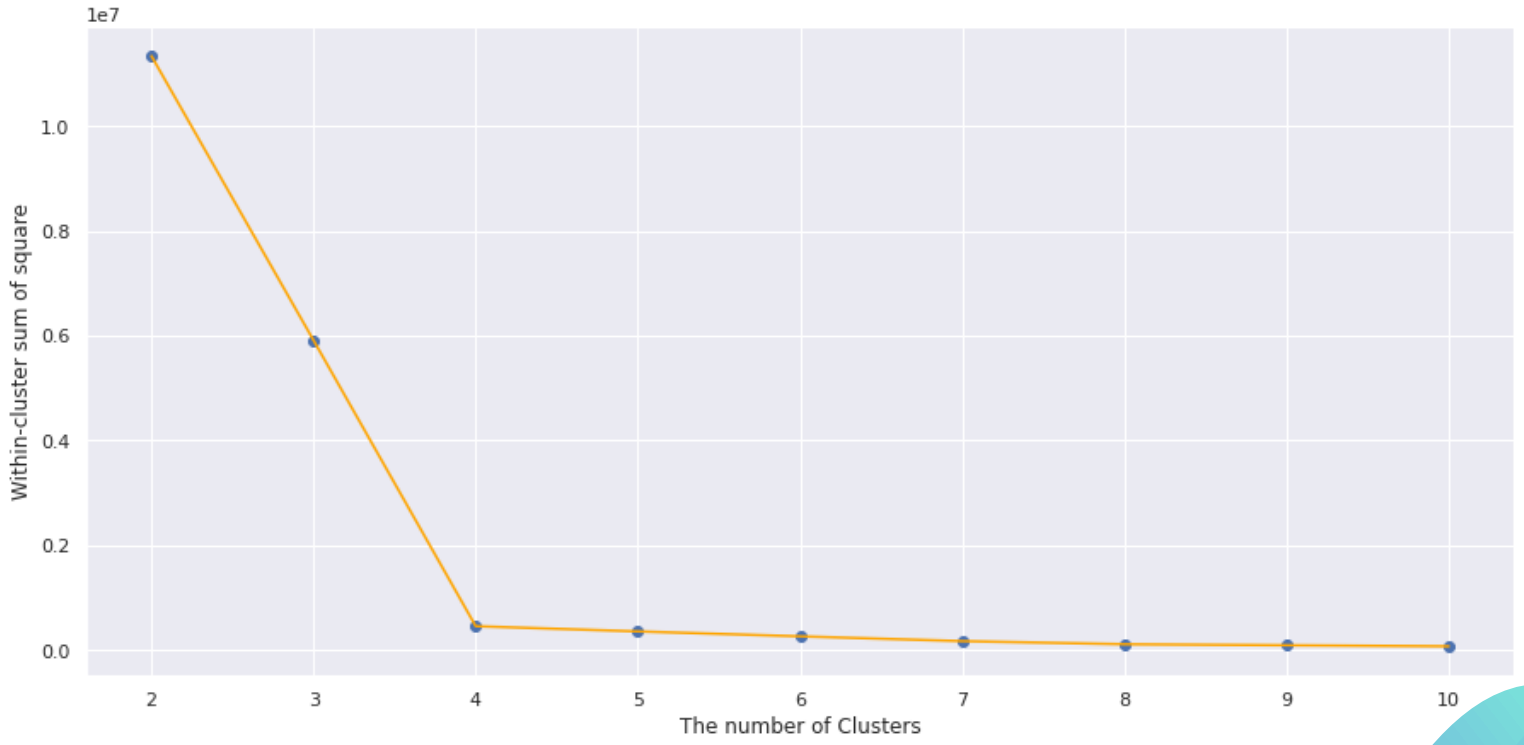
1. Customer Segmentation
2. Document Clustering
3. Recommendation Engineering

1 K-Means

Choose the best K-centroid for our cluster using Elbow Method

```
from sklearn.cluster import KMeans
wcss = []
for i in range (2,11):
    kmeans = KMeans(i)
    kmeans.fit(feature4)
    wcss.append(kmeans.inertia_)

#plot the elbow
clstr = range(2,11)
plt.figure(figsize=(15,7))
plt.scatter(clstr, wcss)
plt.plot(clstr,wcss, color='orange')
plt.xlabel("The number of Clusters")
plt.ylabel("Within-cluster sum of square")
plt.show()
```

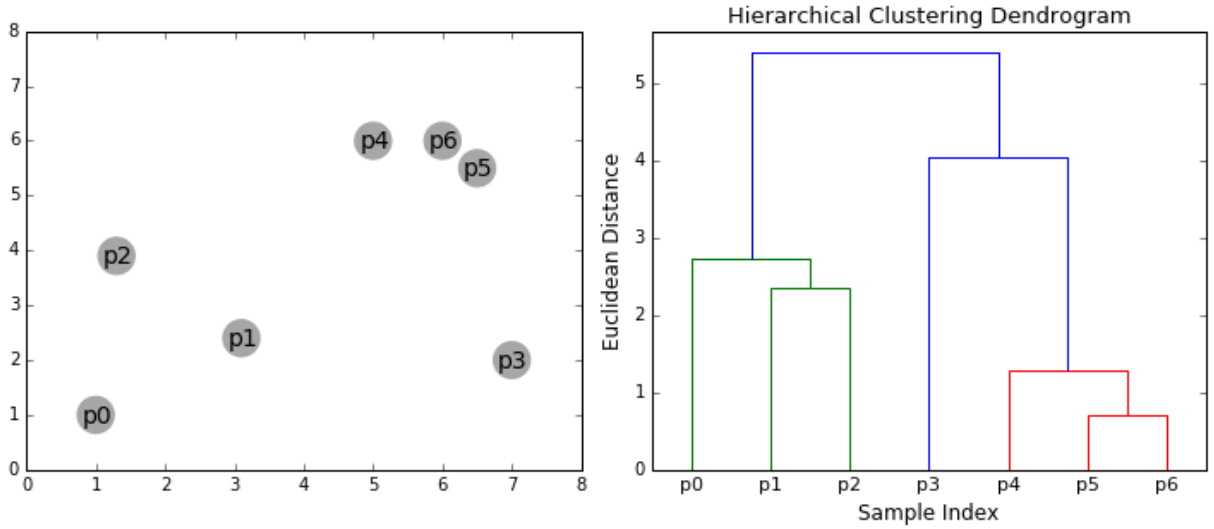


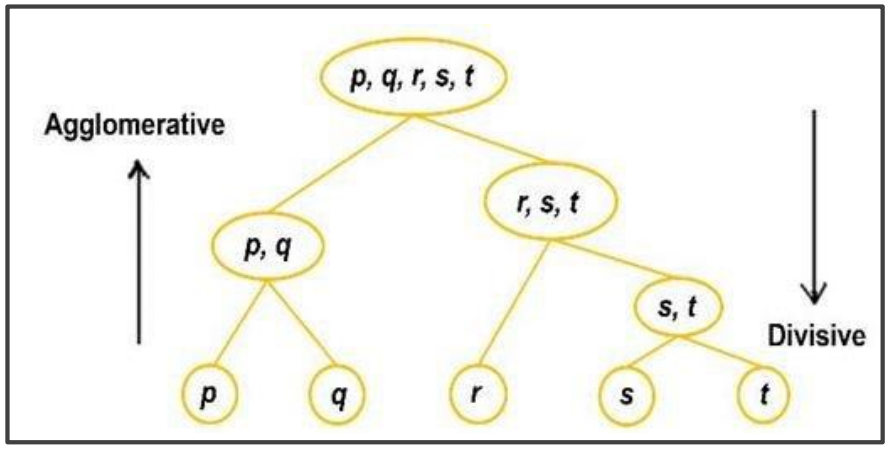
- **inertia** actually calculates the sum of distances of all the points within a cluster from the centroid of that cluster. Keeping this in mind, we can say that **the lesser the inertia value, the better our clusters are**.

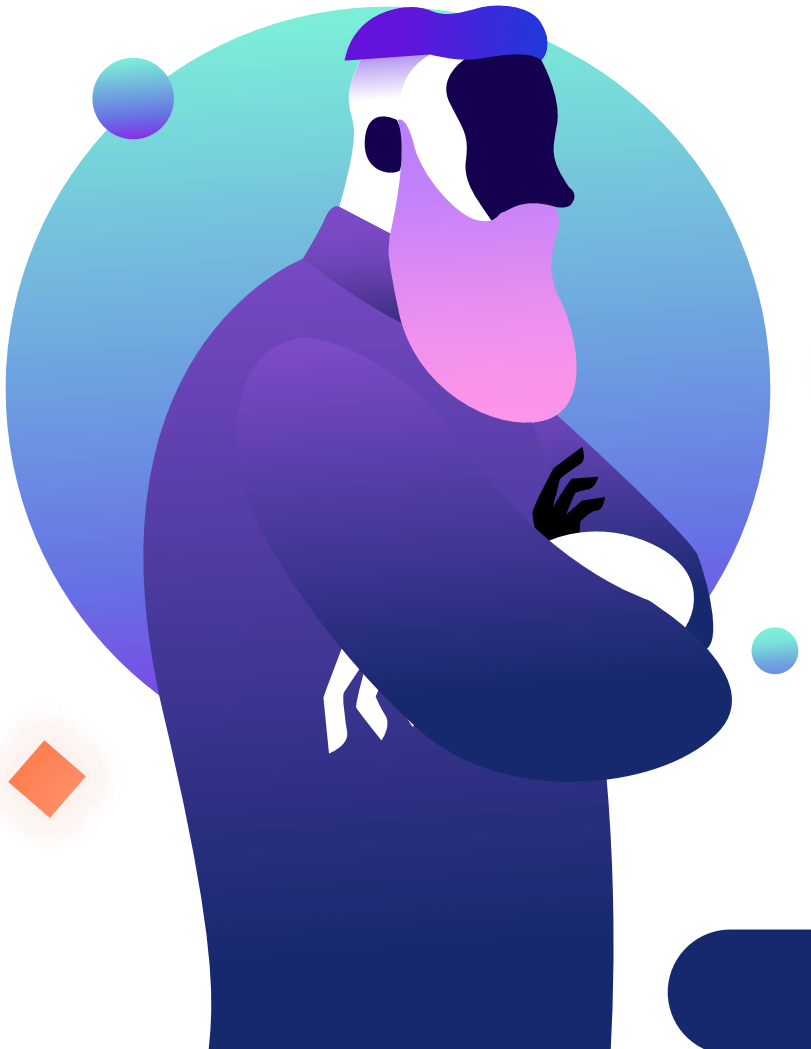
2 Hierarchical

Hierarchical clustering *is unsupervised learning that used to group together the unlabeled data points having similar characteristics. Hierarchical clustering algorithms falls into following two categories.*

- **Agglomerative,** each data point is treated as a single cluster and then successively merge or agglomerate (bottom-up approach) the pairs of clusters (*dendrogram or tree structure*)
- **Divisive,** all the data points are treated as one big cluster and the process of clustering involves dividing (Top-down approach) the one big cluster into various small clusters.







Thank You

We don’t need to memorize every rows of code.
But we need to practice every day, so ***start loving the code.***

References

- https://www.tutorialspoint.com/machine_learning_with_python/index.htm
- https://towardsdatascience.com/what-are-overfitting-and-underfitting-in-machine-learning-a96b30864690
- https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html
- https://medium.com/@moussadoumbia_90919/elbow-method-in-supervised-learning-optimal-k-value-99d425f229e7
- https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/
- https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/