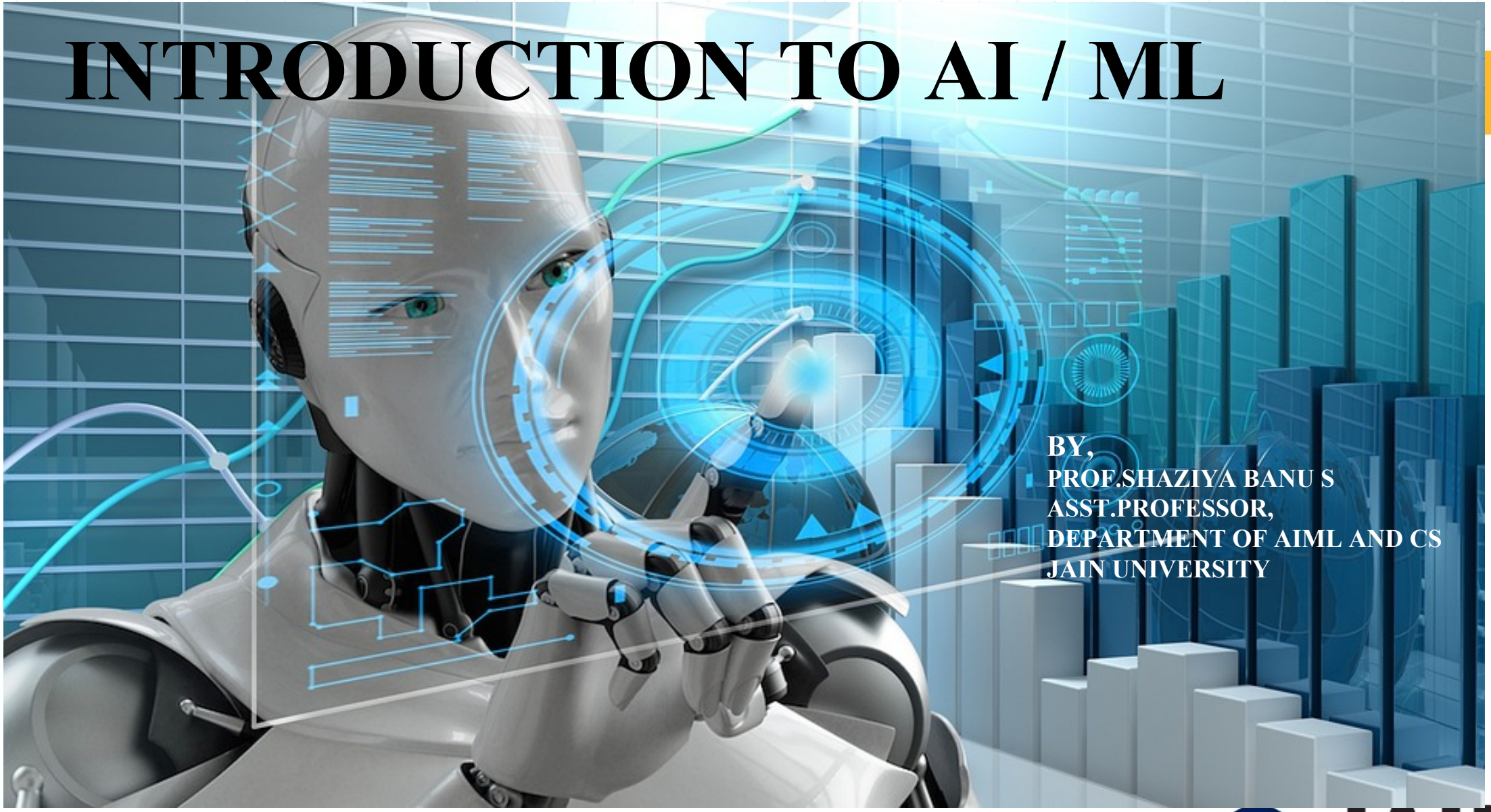
INTRODUCTION TO AI / ML



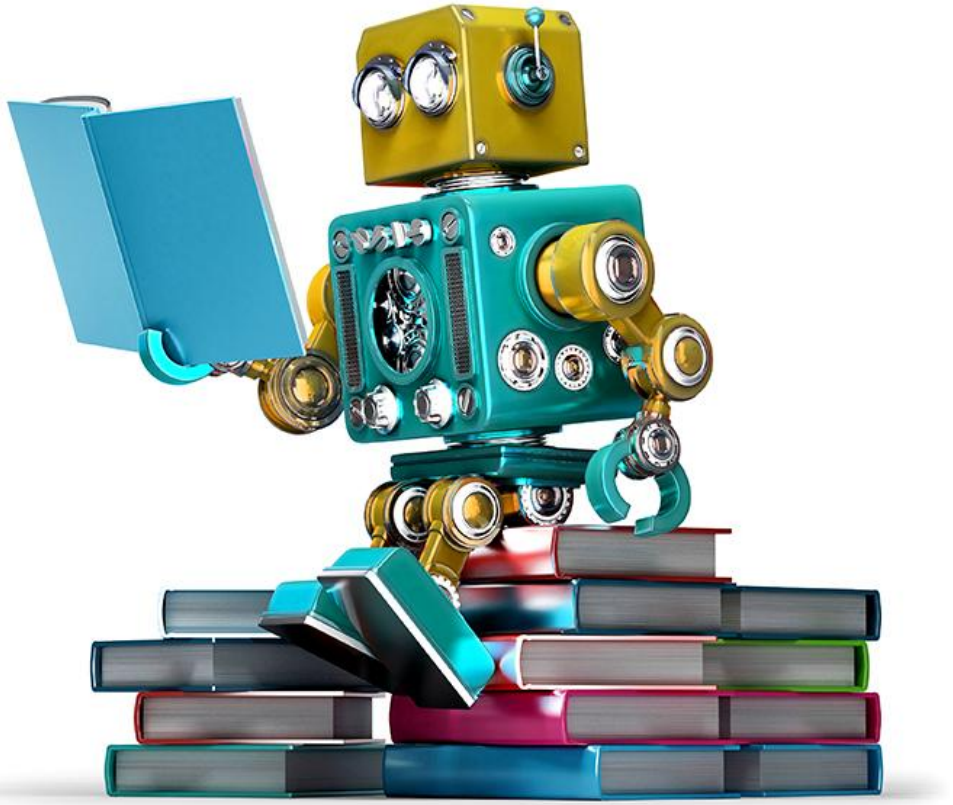
BY,
PROF.SHAZIYA BANU S
ASST.PROFESSOR,
DEPARTMENT OF AIML AND CS
JAIN UNIVERSITY

JGI JAIN DEEMED-TO-BE UNIVERSITY

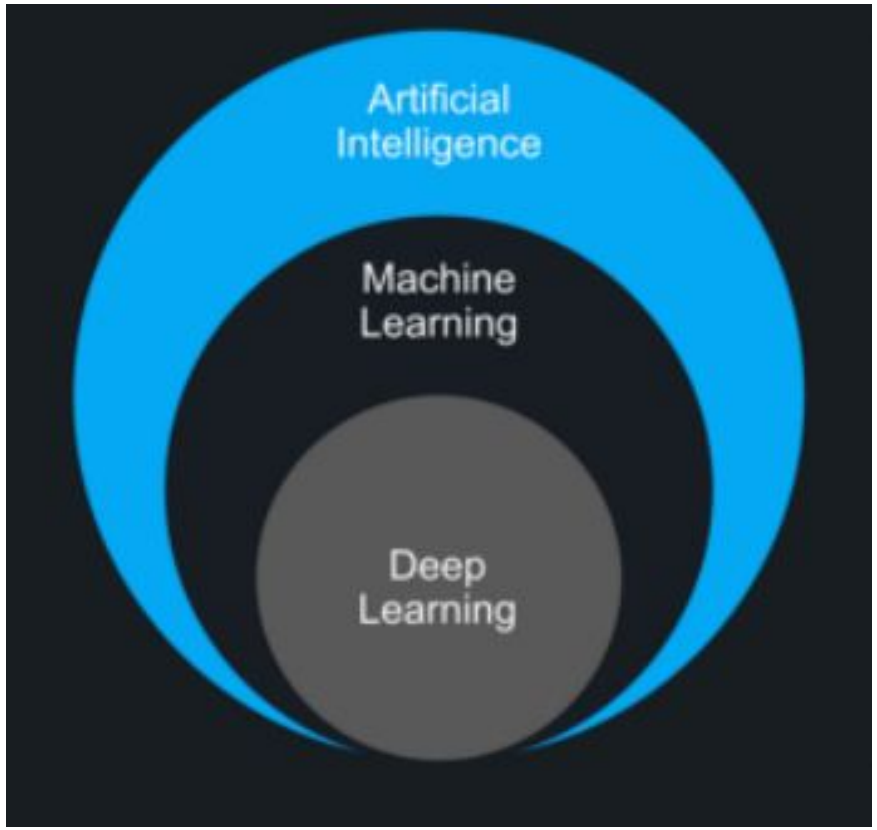
Powered By Futurense

Introduction

- What is AI?
- AI vs ML vs DL
- Types of ML
- Real time use cases



- **Artificial Intelligence** is a technique for building systems that mimic human behavior or decision-making.
- **Machine Learning** is a subset of AI that uses data to solve tasks. These solvers are trained models of data that learn based on the information provided to them.
  - This information is derived from probability theory and linear algebra.
  - ML algorithms use our data to learn and automatically solve predictive tasks.
- **Deep Learning** is a subset of machine learning which relies on multilayered neural networks to solve these tasks.



Mimics Human Behavior

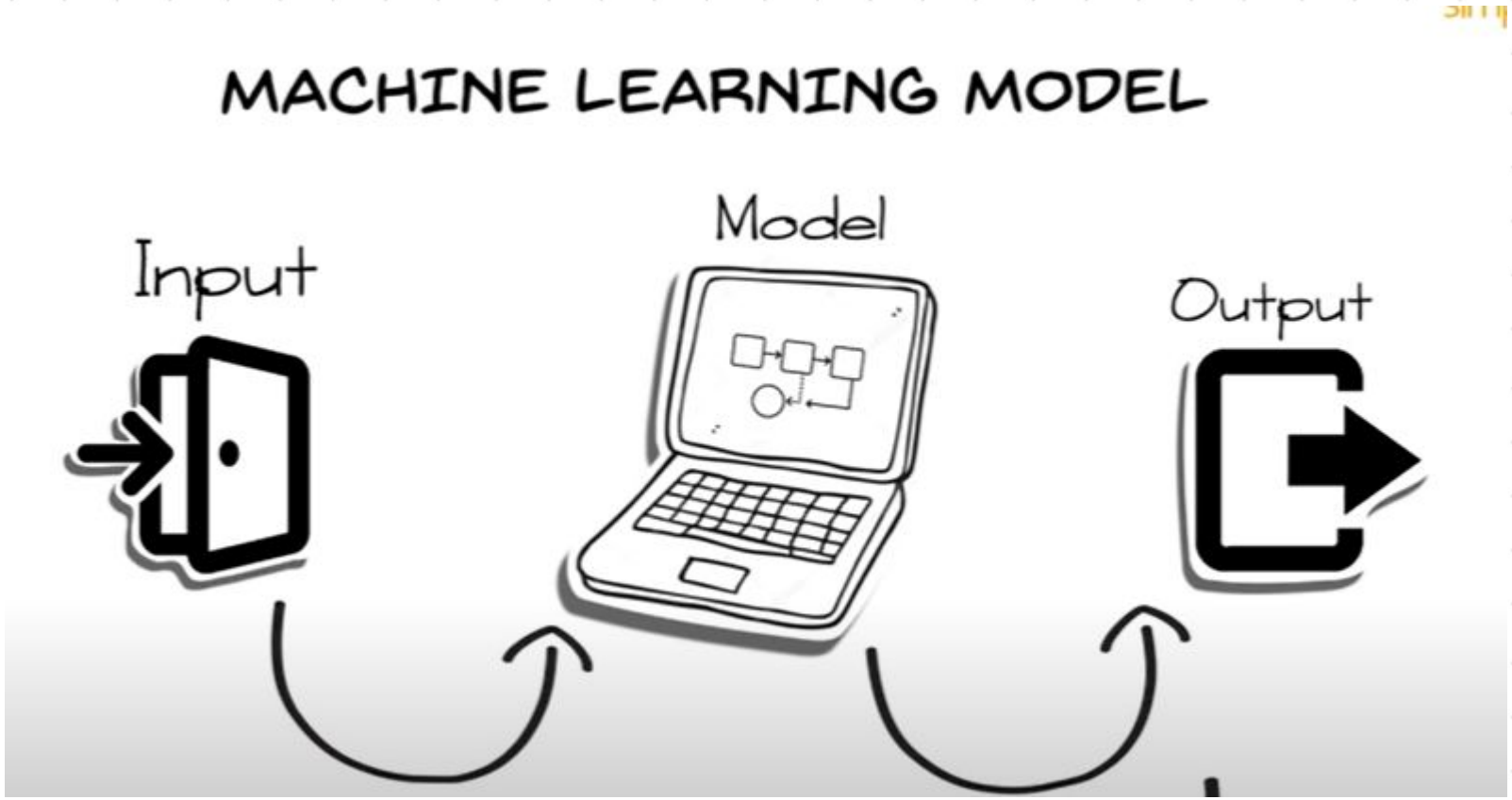
Building Trained Models of Data

- Probabilistic
- Linear Algebraic

Model built by multilayered neural networks.

In what all types can a machine learn?

JGI JAIN DEEMED-TO-BE UNIVERSITY | Powered By Futurense



Common terms to consider

- **Dataset**
- **Sample Data**
- **Variables**
  - **Types of variables**
    - **Independent variable:** Another name for Independent variable is **Predictor variable**.
    - **Dependent variable**: Another name for Dependent variable is **Output variable**. Mainly because the resulting variable value varies depending on the independent variable.
    - **Continuous variable:** Any variable with numerical value is considered as continuous. Temperature, humidity, and chances of rain are considered continuous variables.
    - **Categorical variable:** Any variable with non-numerical value is considered as categorical. Weather is considered a categorical variable.
- **Descriptive data**

Powered By

Futurense

EDA

Steps to analyse the data.

- Data Collection
- Data Cleaning
- EDA
- Modelling
- Deployment

- Exploratory Data Analysis or (EDA) is understanding the data sets by summarizing their main characteristics often plotting them visually.

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns                    #visualisation
import matplotlib.pyplot as plt          #visualisation
%matplotlib inline
sns.set(color_codes=True)
```

- 2.Load the dataset

```
df = pd.read_csv("car_dataset.csv")
```

- 3. Display the headers.

```
df.head(5)
```

- 4.Checking the types of data

```
df.dtypes
```

- 5. Dropping irrelevant columns

```
df = df.drop(['Engine Fuel Type', 'Market Category', 'Vehicl
e Style', 'Popularity', 'Number of Doors', 'Vehicle Size'],
axis=1)
df.head(5)
```

- 6. Renaming the columns

```
df = df.rename(columns={"Engine HP": "HP", "Engine Cylinder
s": "Cylinders", "Transmission Type": "Transmission", "Drive
n_Wheels": "Drive Mode","highway MPG": "MPG-H", "city mpg":
"MPG-C", "MSRP": "Price" })
df.head(5)
```

- 7. Dropping the duplicate rows

```
df.shape
```

```
duplicate_rows_df = df[df.duplicated()]
print("number of duplicate rows: ", duplicate_rows_df.shape)
```

```
df = df.drop_duplicates()
df.head(5)
```

- Dropping the missing or null values.

```
print(df.isnull().sum())
```

```
df = df.dropna()     # Dropping the missing values.
df.count()
```

```
print(df.isnull().sum())    # After dropping the values
```

- Detecting Outliers

```
sns.boxplot(x=df['Price'])
```

```
sns.boxplot(x=df['HP'])
```

```
sns.boxplot(x=df['Cylinders'])
```

- Plot different features against one another (scatter), against frequency (histogram)
- Histogram

```
df.Make.value_counts().nlargest(40).plot(kind='bar', figsize
=(10,5))
plt.title("Number of cars by make")
plt.ylabel('Number of cars')
plt.xlabel('Make');
```

- Heat map

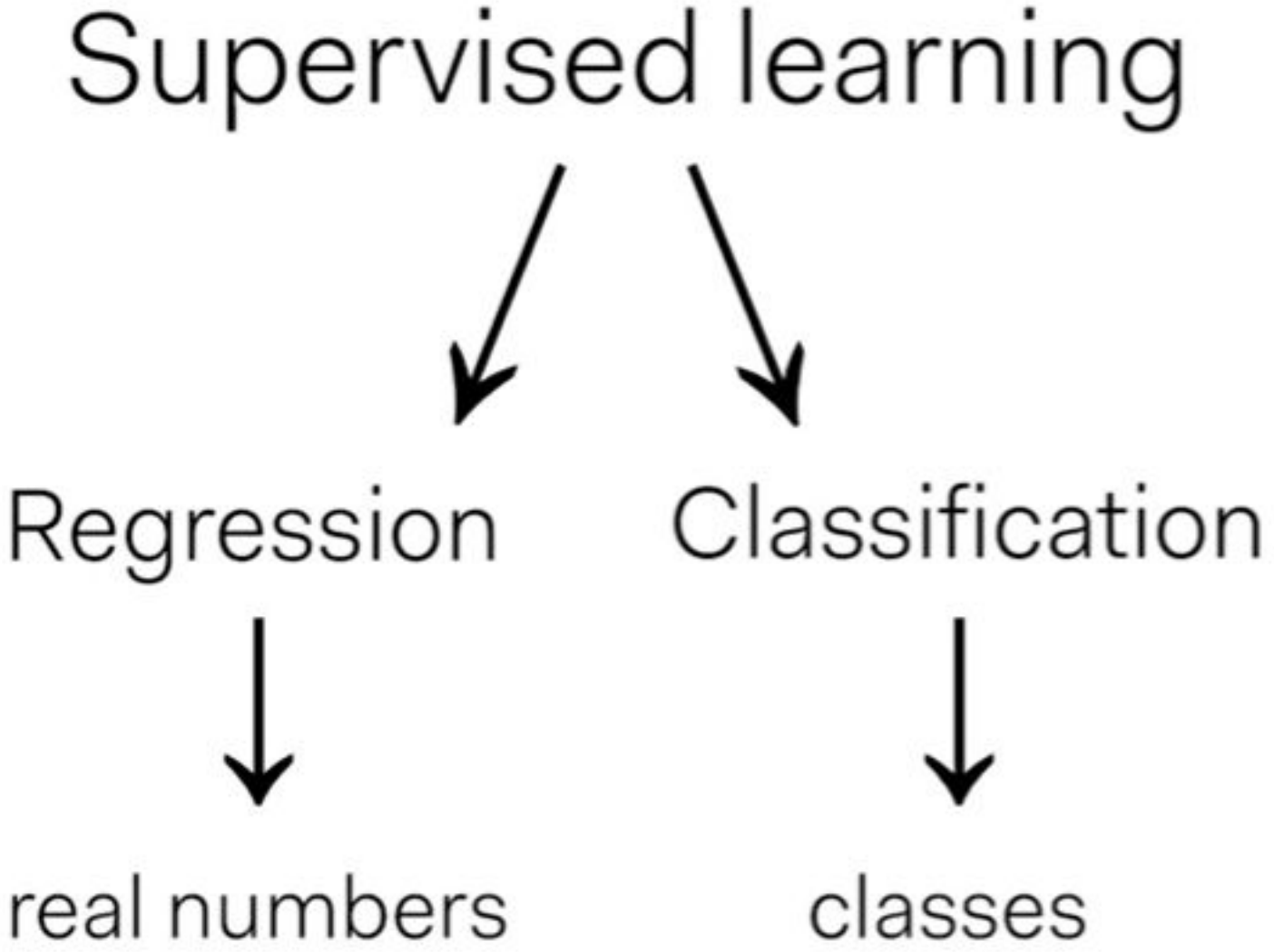
```
plt.figure(figsize=(10,5))
c= df.corr()
sns.heatmap(c,cmap="BrBG",annot=True)
c
```

- Scatter plot

```
fig, ax = plt.subplots(figsize=(10,6))
ax.scatter(df['HP'], df['Price'])
ax.set_xlabel('HP')
ax.set_ylabel('Price')
plt.show()
```

- Machine Learning Regression is a technique for investigating the relationship between independent variables or features and a dependent variable or outcome.
- It’s used as a method for predictive modeling in machine learning, in which an algorithm is used to predict continuous outcomes.

Regression Analysis



What is regression?

- Regression is defined as a statistical method that helps us to analyze and understand the relationship between two or more variables of interest.
- In regression, one dependent variable and one or more independent variables are available.
- We try to “regress” the value of the dependent variable “Y” with the help of the independent variables.

- **Dependent Variable:** This is the variable that we are trying to understand or forecast.
- **Independent Variable:** These are factors that influence the analysis or target variable and provide us with information regarding the relationship of the variables with the target variable.

What is Regression Analysis?

- Regression analysis is used for prediction and forecasting.

- **Financial Industry-** Understand the trend in the stock prices, forecast the prices, and evaluate risks in the insurance domain
- **Marketing-** Understand the effectiveness of market campaigns, and forecast pricing and sales of the product.
- **Manufacturing-** Evaluate the relationship of variables that determine to define a better engine to provide better performance
- **Medicine-** Forecast the different combinations of medicines to prepare generic medicines for diseases.

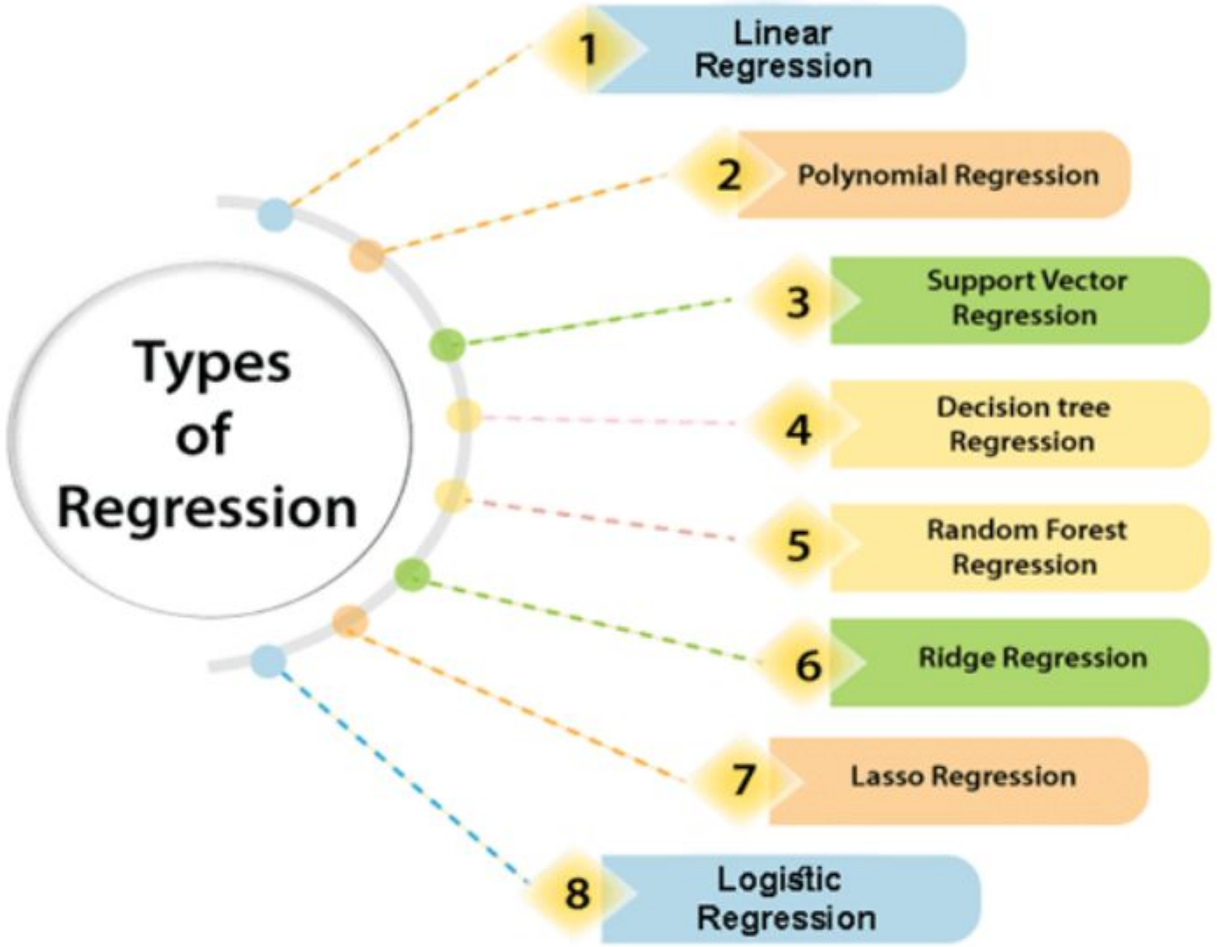
Overfit and Underfit

- When we use unnecessary explanatory variables, it might lead to overfitting. **Overfitting** means that our algorithm works well on the training set but is unable to perform better on the test sets. It is also known as a problem of **high variance**.

- When our algorithm works so poorly that it is unable to fit even a training set well, then it is said **to underfit** the data. It is also known as a problem of **high bias**.

Types of regression

- Linear Regression
- Logistic Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression
- Ridge Regression
- Lasso Regression:

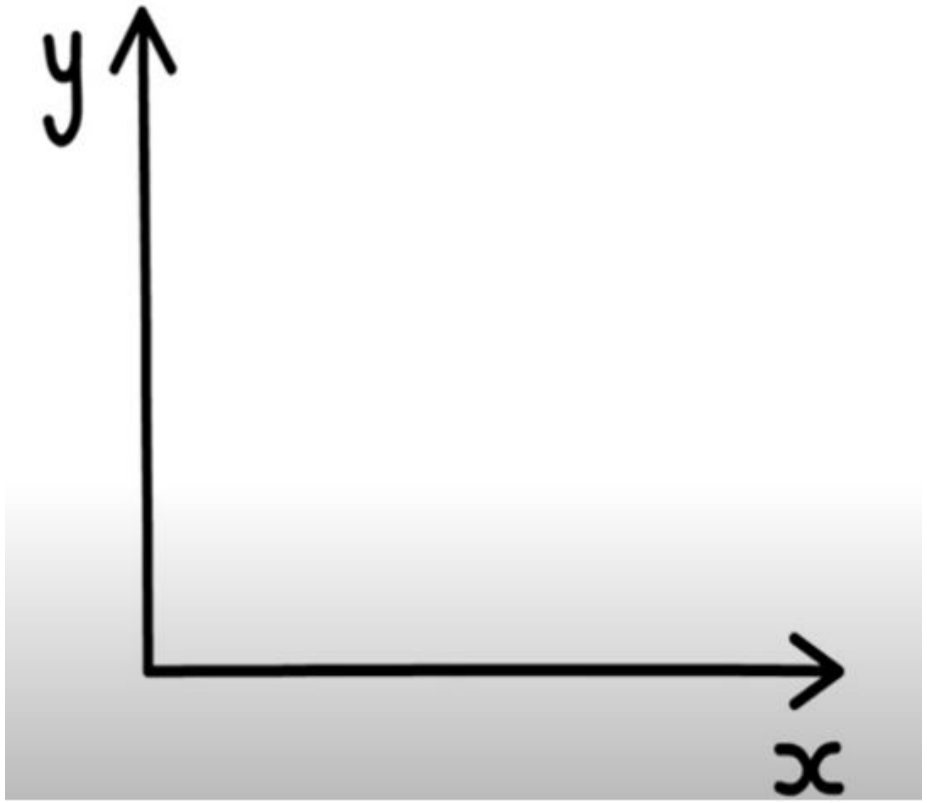


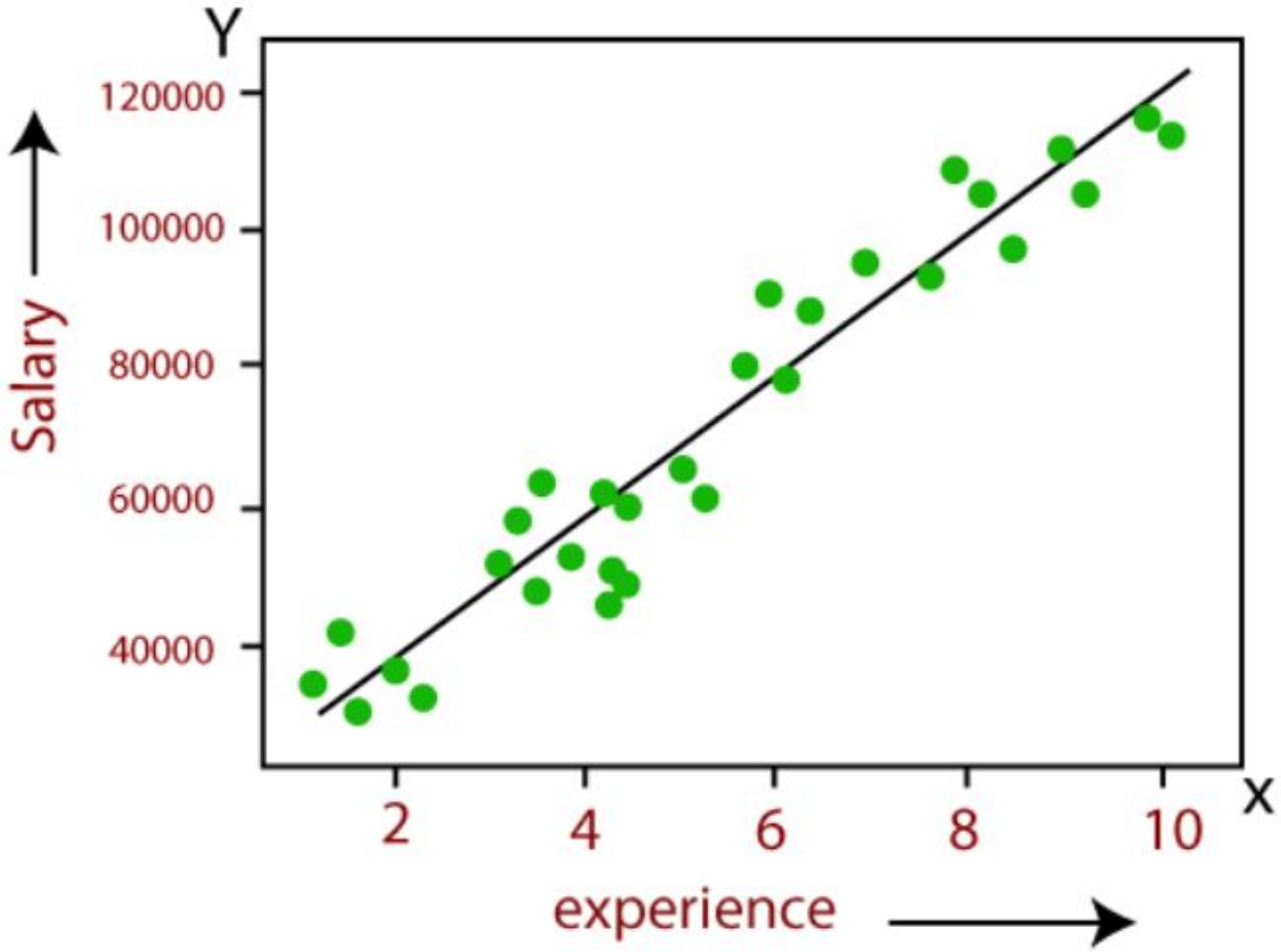
Linear regression (ordinary least square method)

- Linear regression is a statistical regression method that is used for predictive analysis.
- It is one of the very simple and easy algorithms which works on regression and shows the relationship between the **continuous variables.**
- Simple linear regression

Y= mX + c

- Multiple linear regression





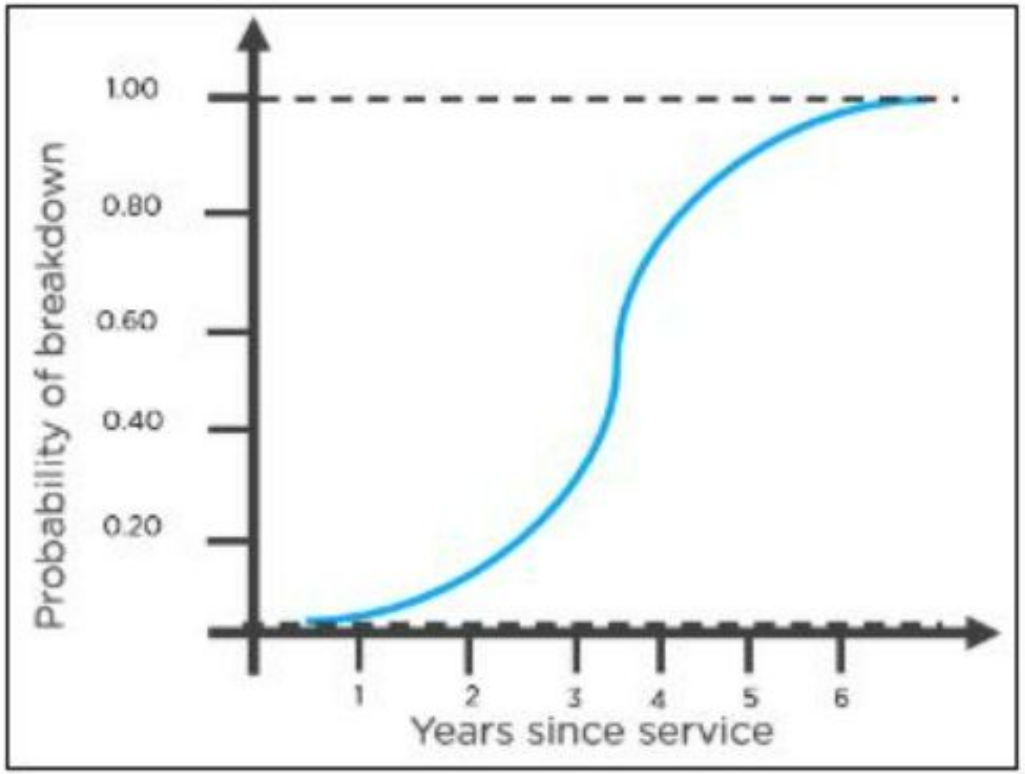
Logistic regression

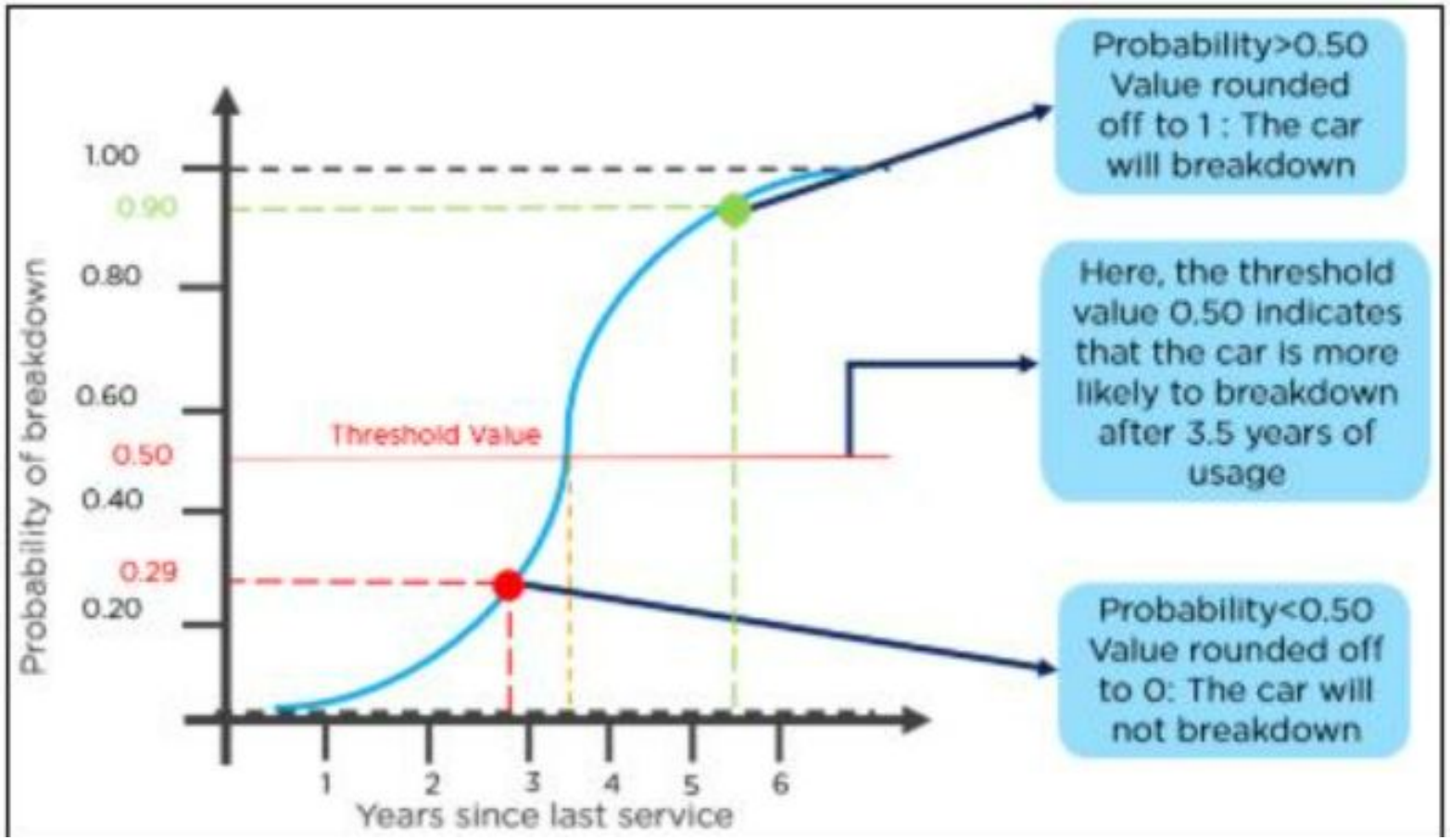
- In **classification problems**, we have dependent variables in a binary or discrete format such as 0 or 1.
- Logistic regression is a statistical method that is used for building machine learning models where the dependent variable is dichotomous: i.e. binary.
- Logistic regression is used to describe data and the relationship between one dependent variable and one or more independent variables.
- The independent variables can be nominal, ordinal, or of interval type.

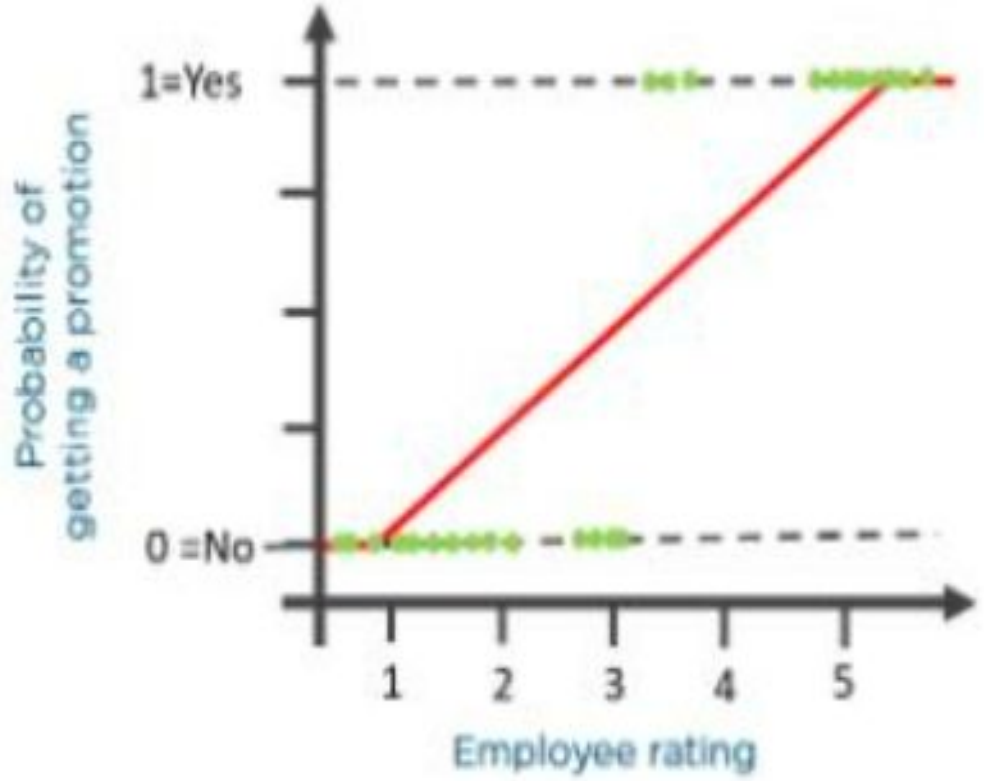
The equation of the sigmoid function is:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

- The following is an example of a logistic function







Decision tree

- Tree structured classifier.
- Can be used for both classification and regression.
- There are two types of nodes
  - Decision node(test)
  - Leaf node(result of classification)
- Test is performed on the basis of feature to get the output result

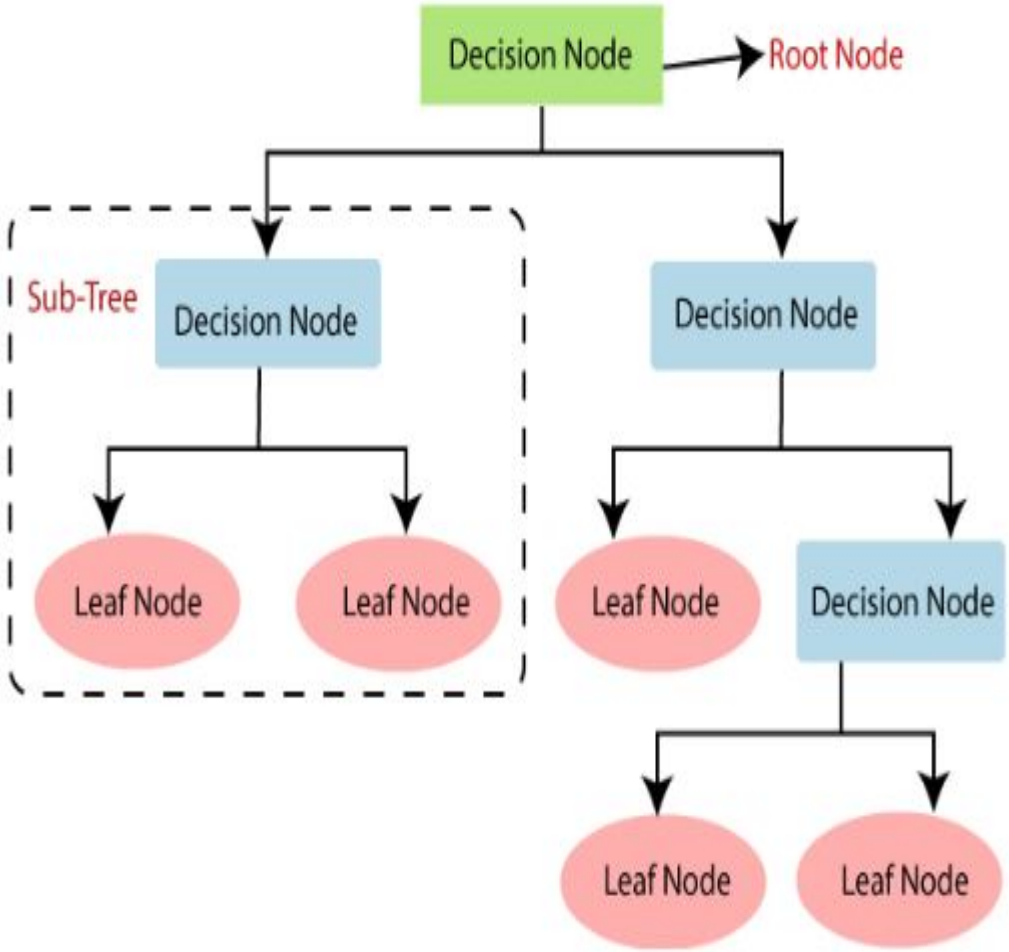
- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

Two Types of Decision Tree

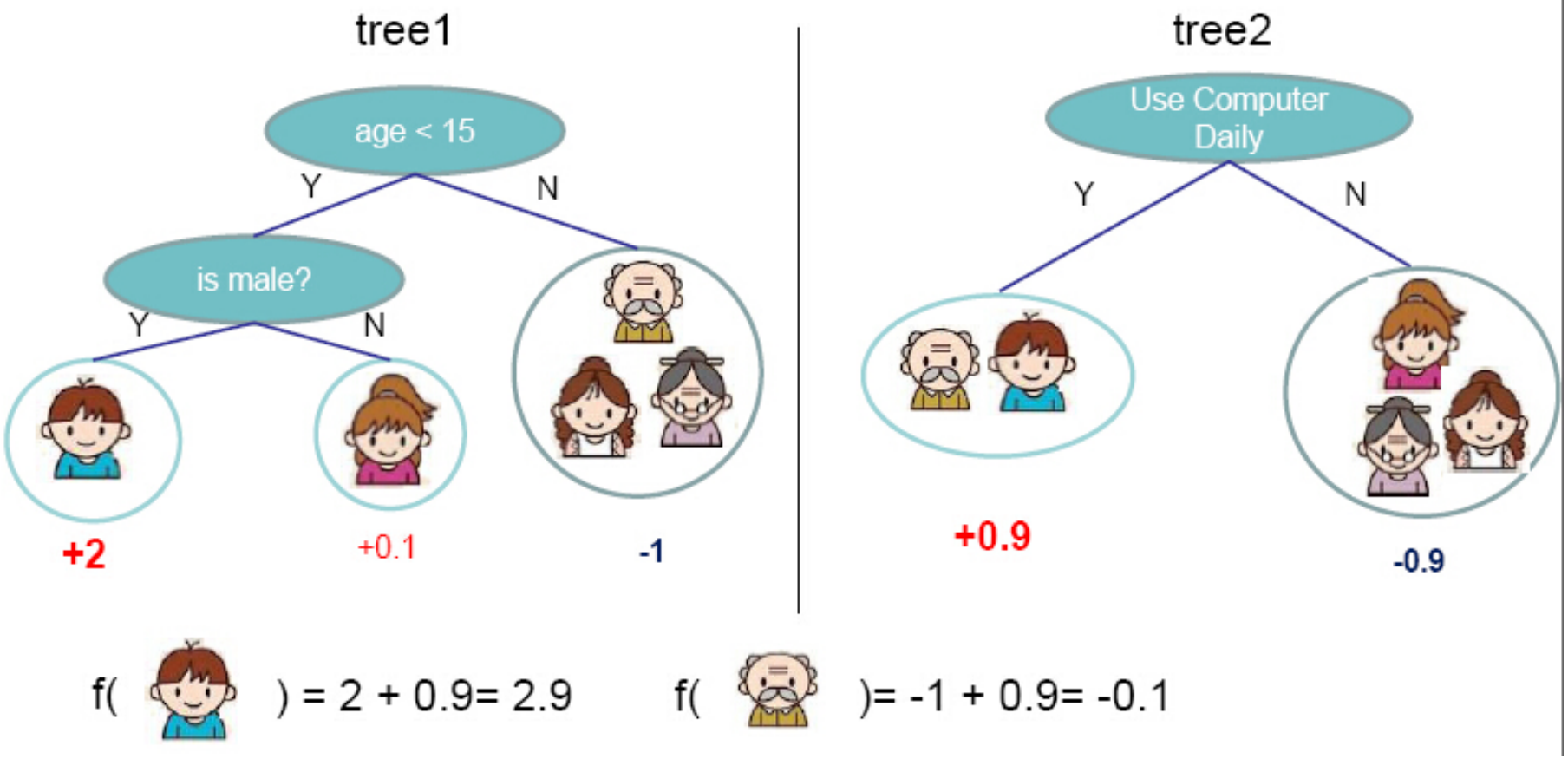
1. Classification
2. Regression

Decision Tree Terminologies

- **Root Node**: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.



- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the decision tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.



Attribute Selection Measures

- While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.** By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:
- **Information Gain**
- **Gini Index**

1. Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- According to the value of information gain, we split the node and build the decision tree.
- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:
- Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

- Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:
- Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

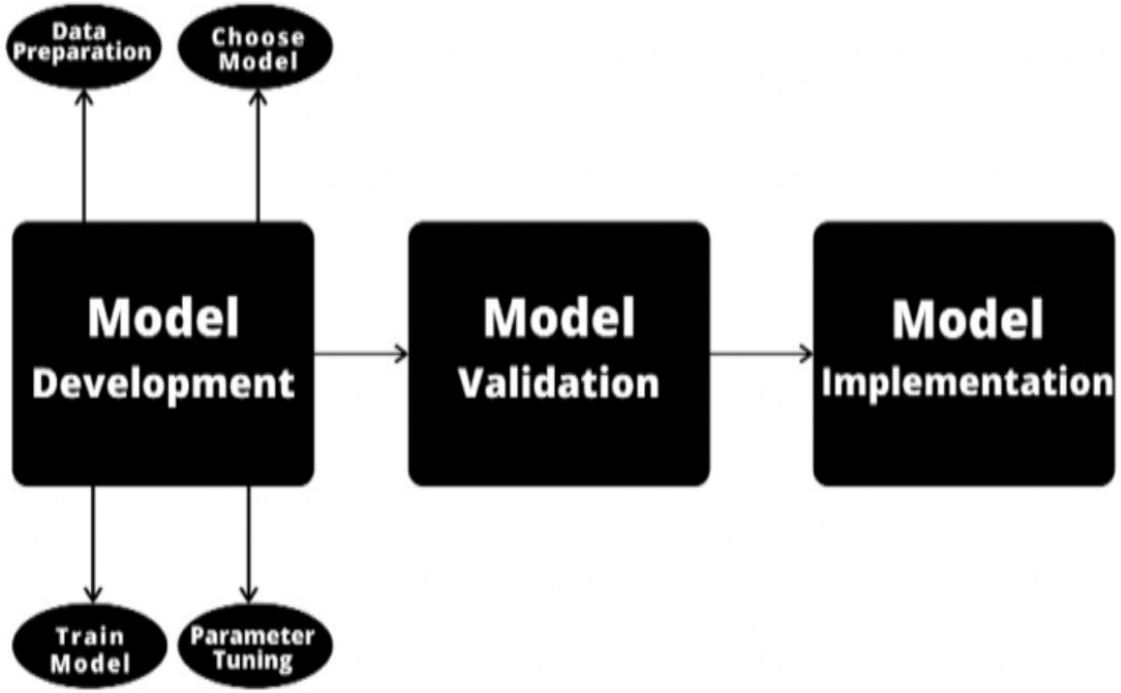
Where,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

Model Validation

"Machine Learning is about making predictions".

- These predictions come up after assorted processes like Data Preparation, Choosing a Model, Training the Model, Parameter Tuning, Model Validation, etc. So, only after carrying out the aforementioned operations, a Machine Learning Model (Regression or Classification) is efficient to make predictions.



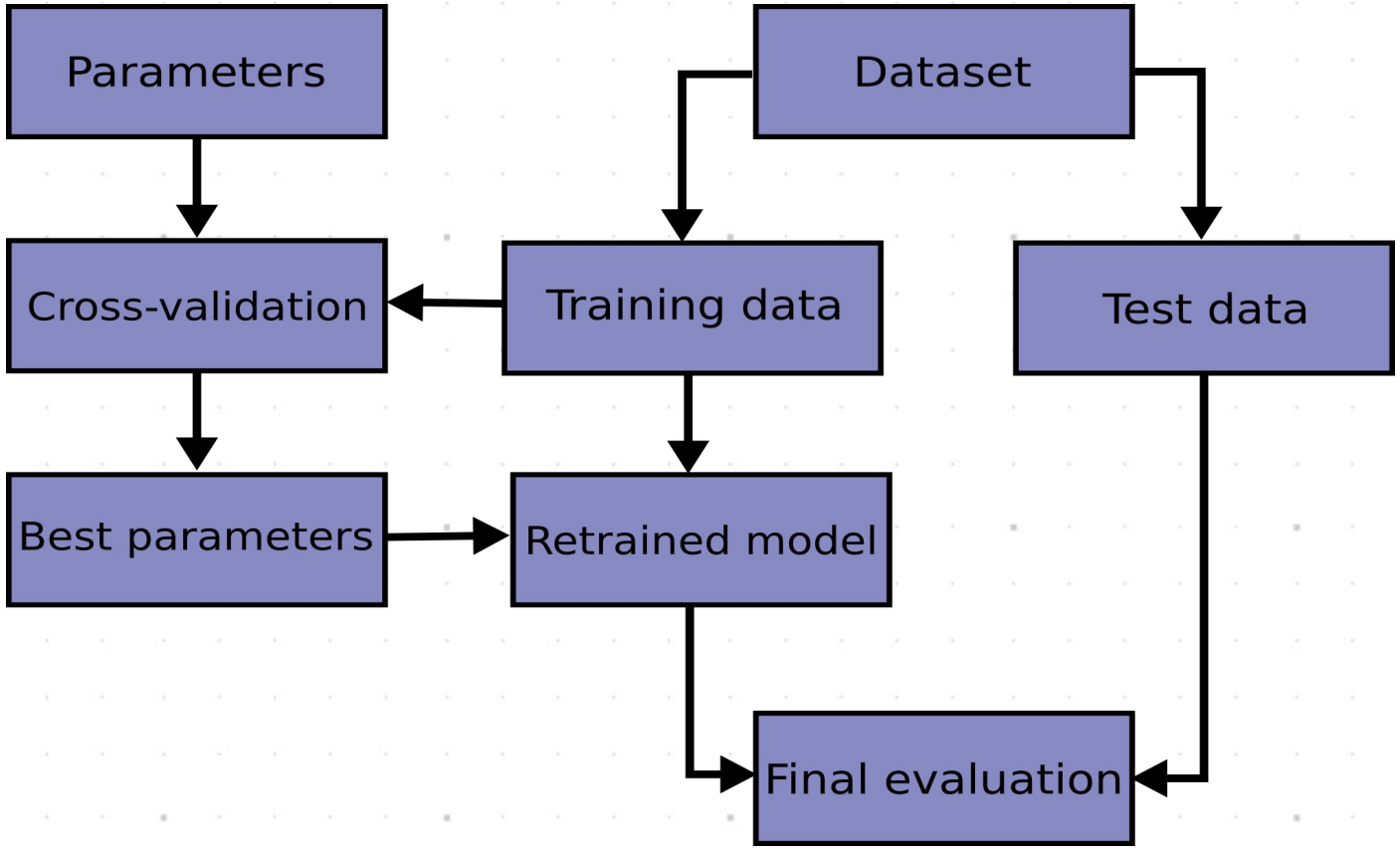
What is model validation?

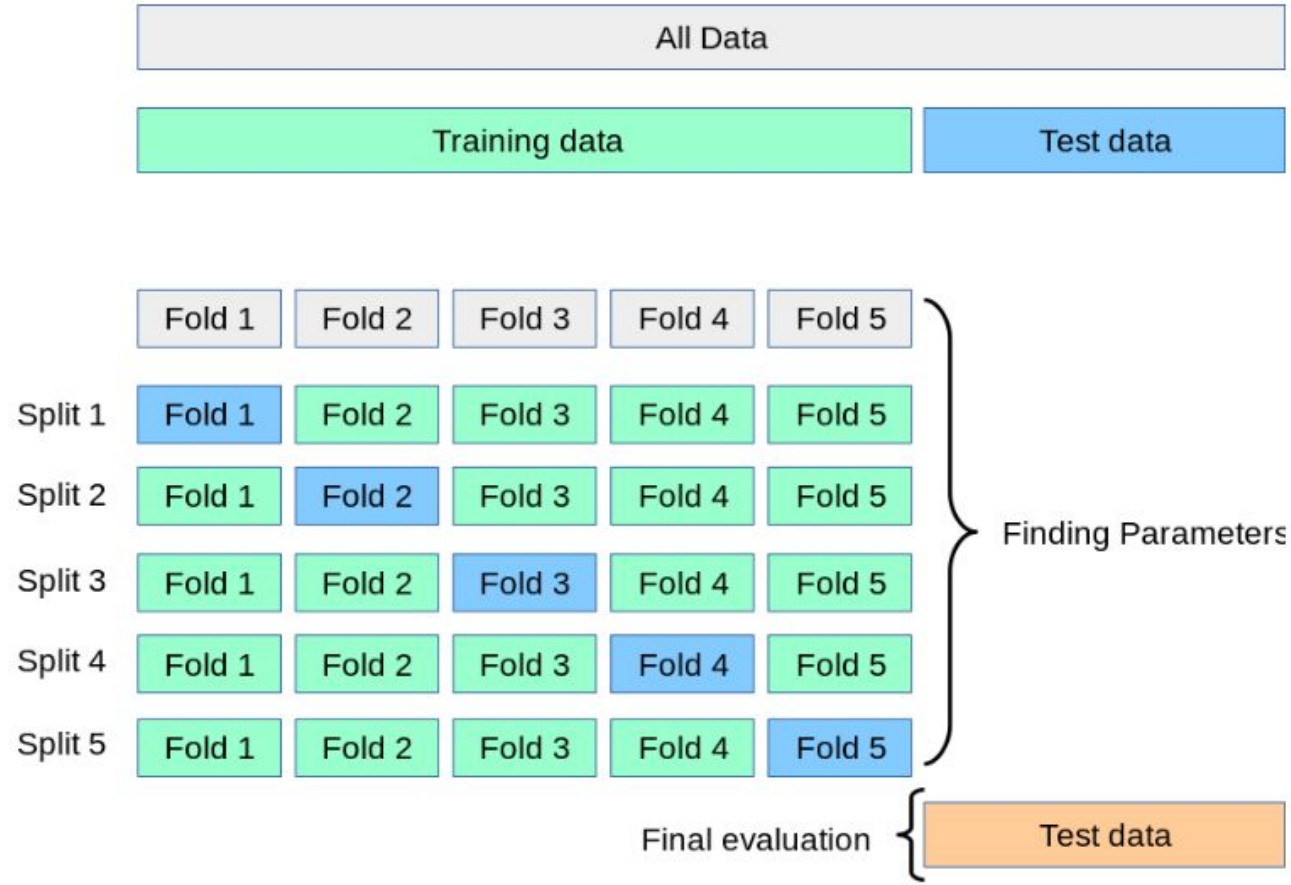
- **Model validation** is the process that is carried out after **Model Training** where the trained model is evaluated with a testing data set. The testing data may or may not be a chunk of the same data set from which the training set is procured.

Importance of Model Validation

- Scalability and flexibility
- Reduce the costs.
- Enhance the model quality.
- Discovering more errors
- Prevents the model from overfitting and underfitting.

Cross validation





K-fold :

- K Fold divides all the samples in groups of samples, called folds (if this is equivalent to the Leave One Out strategy), of equal sizes (if possible). The prediction function is learned using folds, and the fold left out is used for test.

- Example of 2-fold cross-validation on a dataset with 4 samples:

- >>>
- >>> import numpy as np
- >>> from sklearn.model_selection import KFold

- >>> X = ["a", "b", "c", "d"]
- >>> kf = KFold(n_splits=2)
- >>> for train, test in kf.split(X):
- ... print("%s %s" % (train, test))
- [2 3] [0 1]
- [0 1] [2 3]

Implementation:

- By sci-kit learn library,

```
>>> import numpy as np
>>> from sklearn.model_selection import
train_test_split >>> from sklearn import datasets
>>> from sklearn import svm

>>> X, y = datasets.load_iris(return_X_y=True)
>>> X.shape, y.shape
((150, 4), (150,))
```

- sample a training set while holding out 40% of the data for testing (evaluating) our classifier:

```
>>> X_train, X_test, y_train, y_test = train_test_split(
... X, y, test_size=0.4, random_state=0)
```

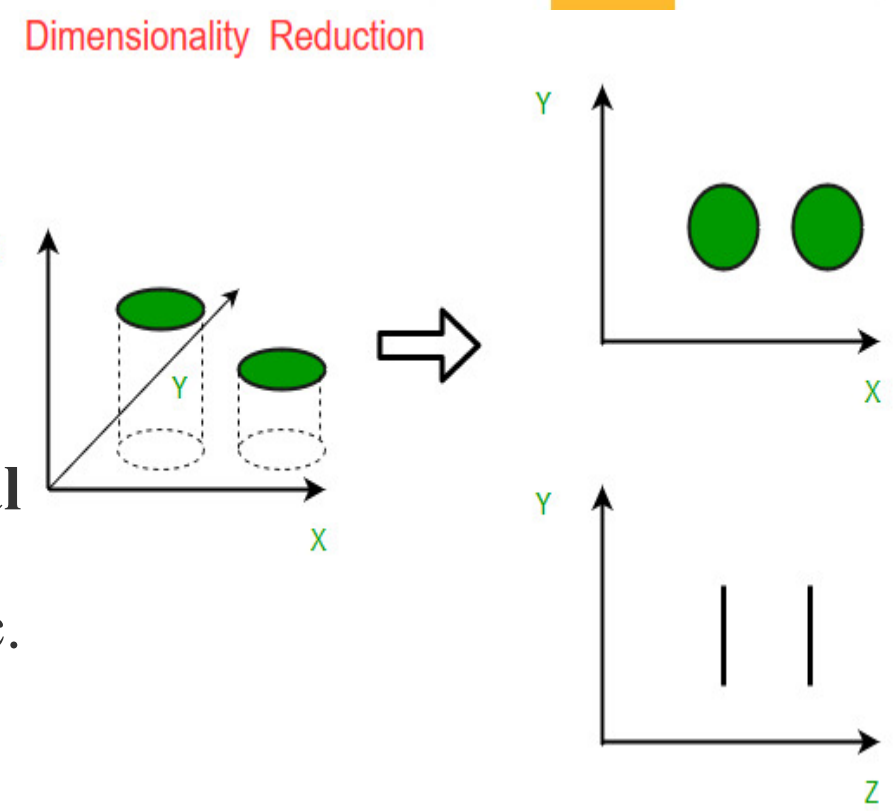
```
>>> X_train.shape, y_train.shape
((90, 4), (90,))
>>> X_test.shape, y_test.shape
((60, 4), (60,))
```

```
>>> clf = svm.SVC(kernel='linear', C=1).fit(X_train, y_train)
>>> clf.score(X_test, y_test)
0.96...
```

Dimensionality Reduction

Dimensionality reduction

- ***"It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information.“***
- Used obtaining a better fit predictive model while solving the classification and regression problems.
- It is commonly used in fields that deal with high-dimensional data, such as **speech recognition, signal processing, bioinformatics, etc. It can also be used for data visualization, noise reduction, cluster analysis**, etc.



Examples for dimensional reduction

Consider the following cubes illustrating temperature of certain days recorded weekly:

Temperature	64	65	68	69	70	71	72	75	80	81	83	85
Week1	1	0	1	0	1	0	0	0	0	0	1	0
Week2	0	0	0	1	0	0	1	2	0	1	0	0

Temperature	cool	mild	hot
Week1	2	1	1
Week2	2	1	1

Temperature	cool	mild	hot
Week1	2	1	1
Week2	2	1	1

Temperature	cool	mild	hot
Day 1	0	0	0
Day 2	0	0	0
Day 3	0	0	1
Day 4	0	1	0
Day 5	1	0	0
Day 6	0	0	0
Day 7	1	0	0
Day 8	0	0	0
Day 9	1	0	0
Day 10	0	1	0
Day 11	0	1	0
Day 12	0	1	0
Day 13	0	0	1
Day 14	0	0	0

The Curse of Dimensionality

- Handling high-dimensional data is very difficult in practice, commonly known as the *curse of dimensionality.*
- As the number of features increases, the number of samples also gets increased proportionally, and the chance of overfitting also increases.
- Hence, it is often required to reduce the number of features, which can be done with dimensionality reduction.

ADVANTAGES:

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.
- Less Computation training time is required for reduced dimensions of features.
- Reduced dimensions of features of the dataset help in visualizing the data quickly.
- It removes the redundant features (if present) by taking care of multicollinearity.

DISADVANTAGES:

- Some data may be lost due to dimensionality reduction.
- In the PCA dimensionality reduction technique, sometimes the principal components required to consider are unknown.

L1, L2 regularization

The term ‘regularization’ refers to a set of techniques that regularizes learning from particular features for traditional algorithms

Mathematically can be represented as,

$$y = w_0 + w_1x_1 + w_2x_2 + \ldots.. + w_nx_n$$

where y is the value to be predicted;

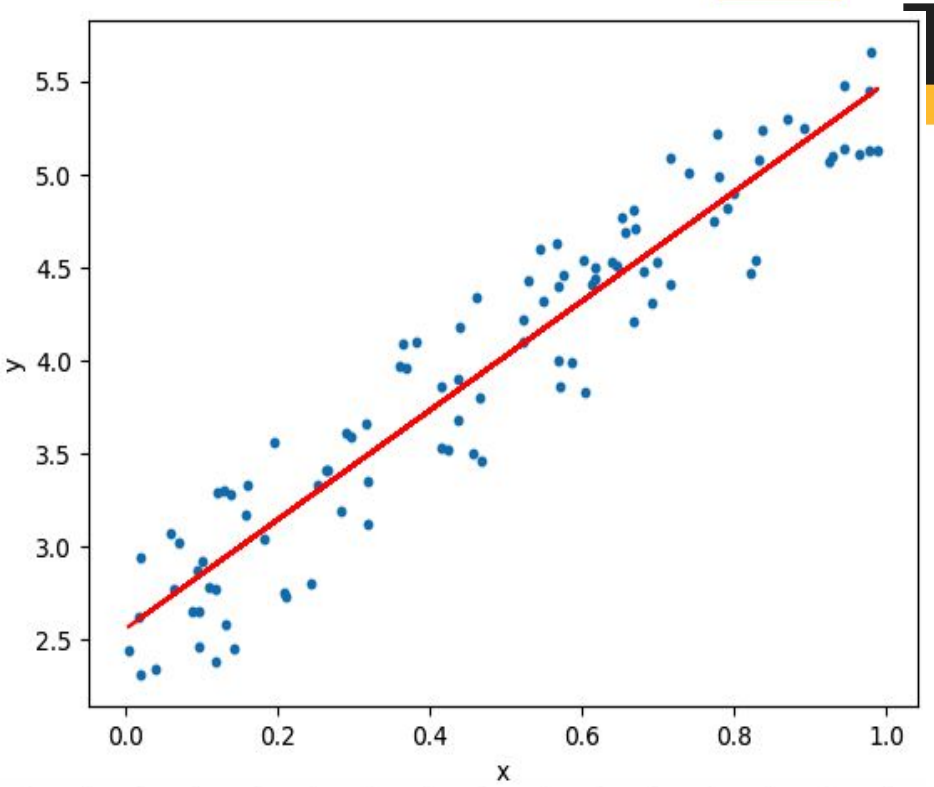
$x_1, x_2, \ldots., x_n$ are features that decides the value of y;

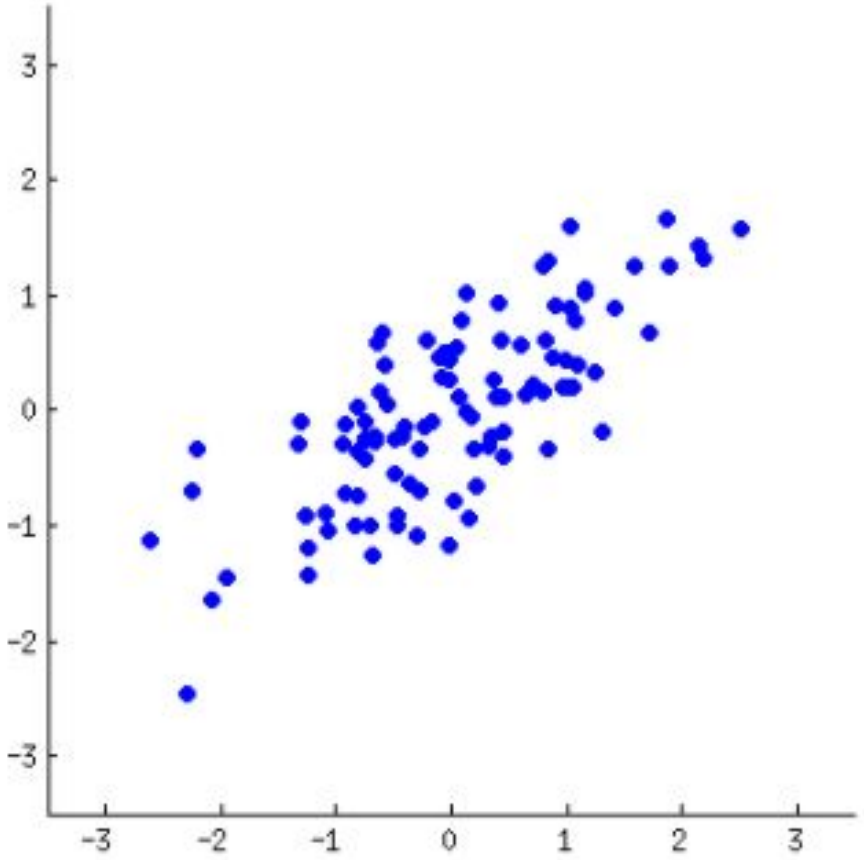
w_0 is the bias;

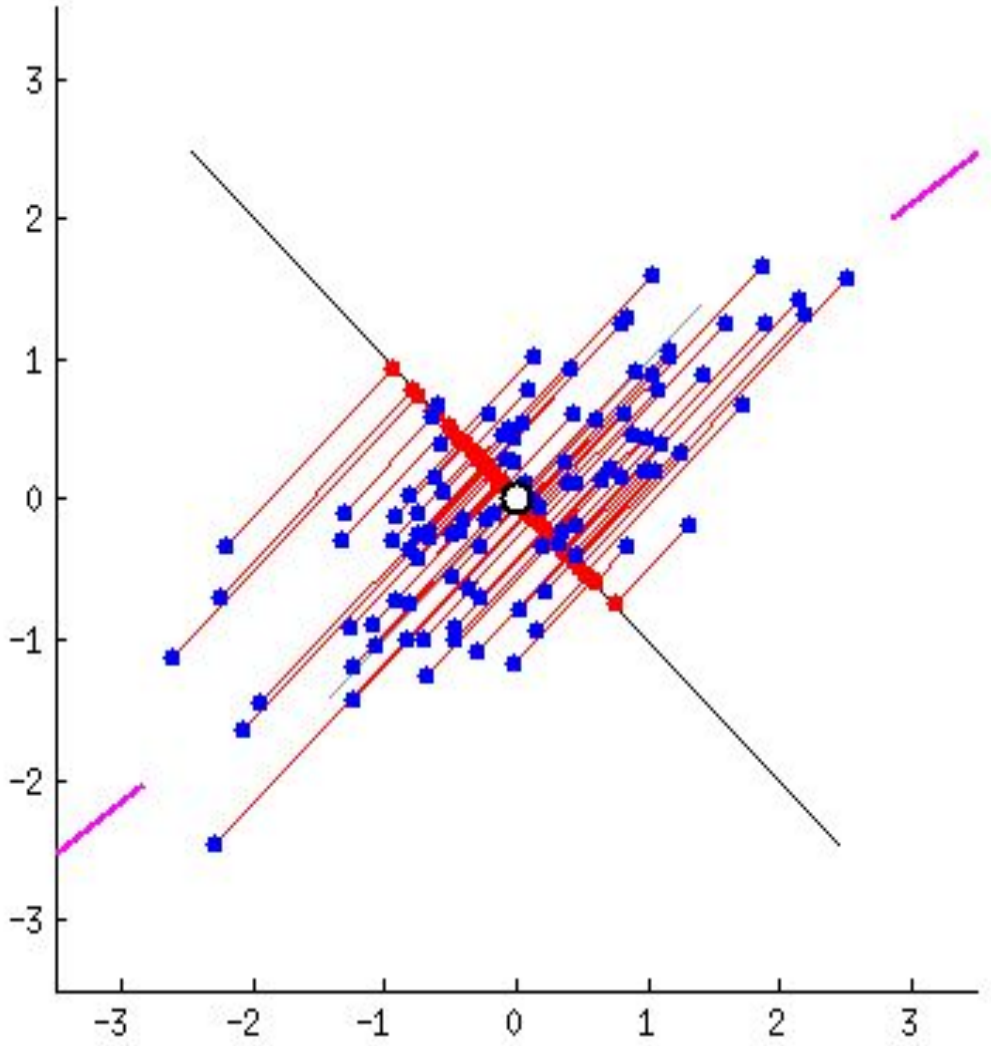
$w_1, w_2, \ldots.., w_n$ are the weights attached to $x_1, x_2, \ldots., x_n$ relatively.

PCA

- Principal Component Analysis(PCA) is one of the most popular linear dimension reduction algorithms. It is a projection-based method that transforms the data by projecting it onto a set of orthogonal(perpendicular) axes.
- That magenta straight line is called the **principal axis.**
- the dataset is projected onto these axes. The columns in the projected or transformed dataset are called **principal components**.





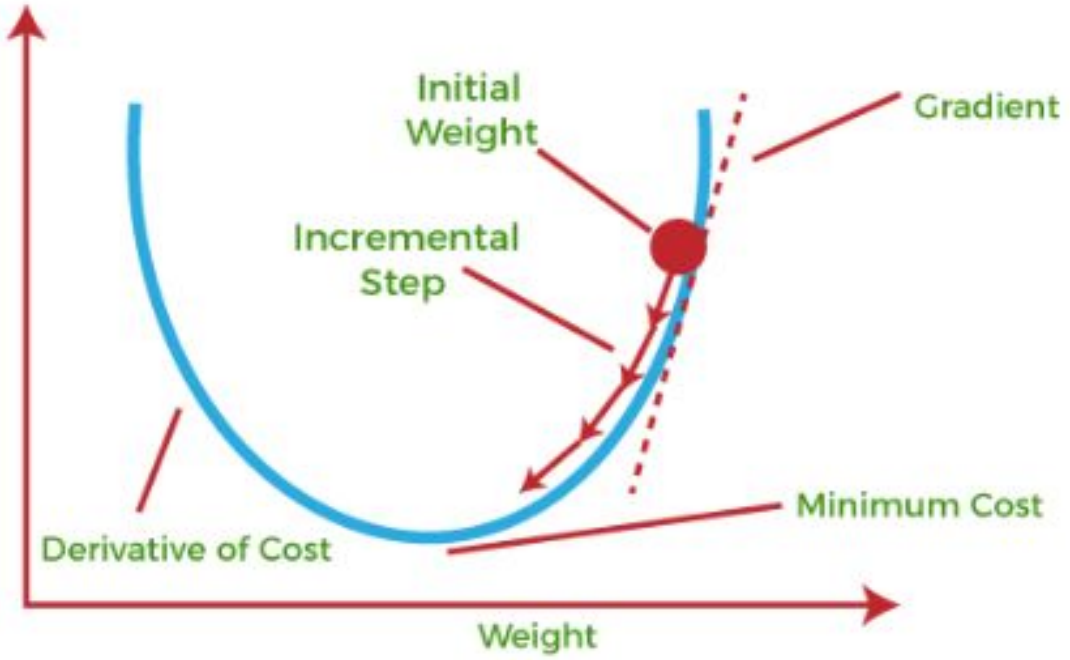


Gradient Descent in Machine Learning

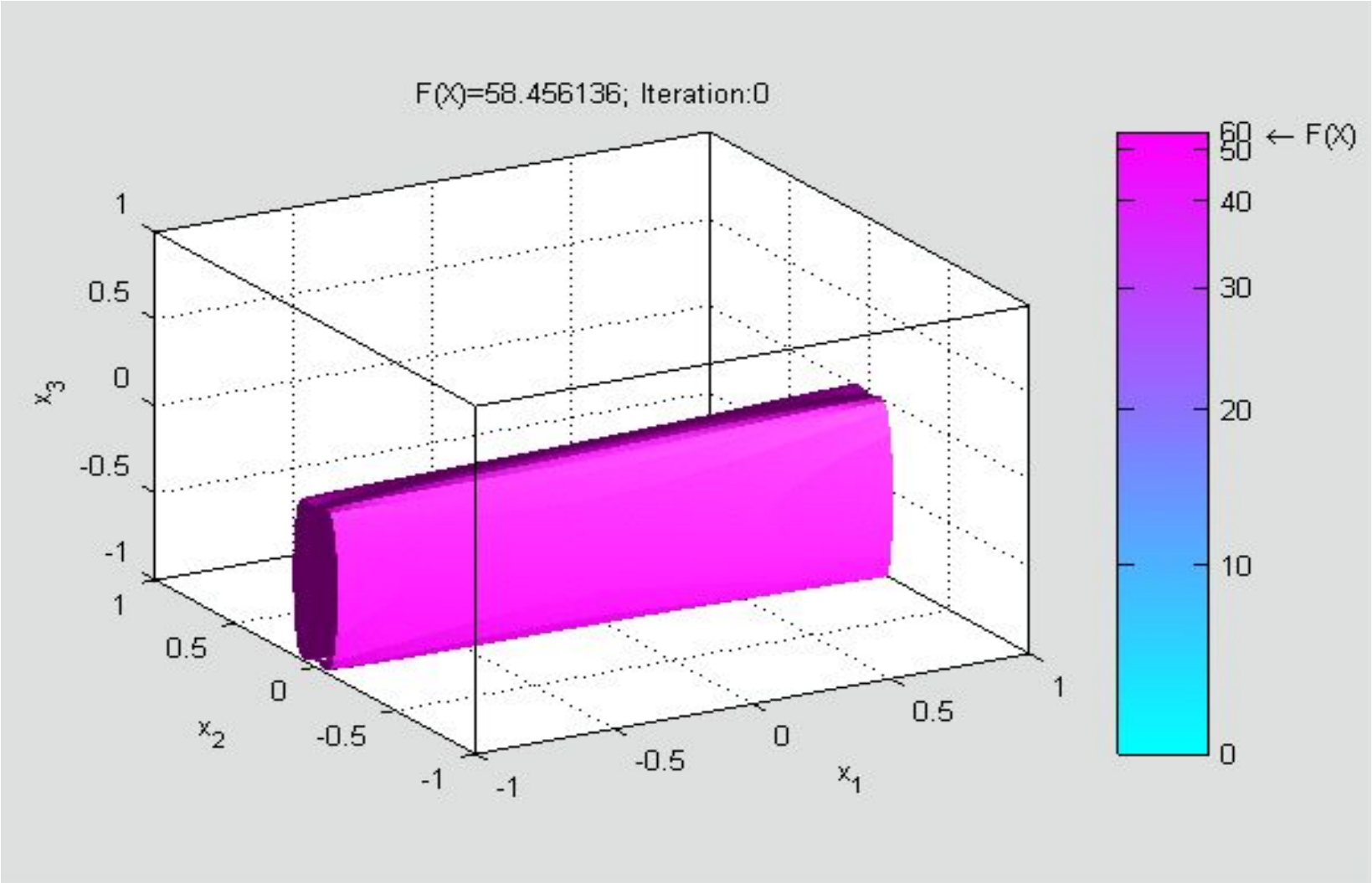
- Gradient Descent is known as one of the most commonly used **optimization algorithms** to train machine learning models by means of **minimizing errors** between **actual and expected results.**
- In mathematical terminology,
  - Optimization algorithm refers to the task of minimizing/maximizing an objective function f(x) parameterized by x.
  - Similarly, in machine learning, optimization is the task of minimizing the **cost function parameterized** by the model's parameters.
- The main objective of gradient descent is to minimize the convex function using iteration of parameter updates.

Cost-function

The cost function is defined as the measurement of difference or error between actual values and expected values at the current position and present in the form of a single real number.



Powered By Futurense



Thank you