UNIT IV

ML SUPERVISED LEARNING

Unit IV: ML Supervised Learning	4a. Describe supervised learning. 4b. Explain the procedure of Implementation of Simple Linear Regression Algorithm 4c. Differentiate Binary, Multi class and Multi label 4d. List key points of Logistic regression for classification problems. 4e. List key points of the Decision tree for classification problems. 4f. List key points of Random forest for classification problems.	4.1 Supervised learning: Regression and Classification 4.2 Regression: Implementation of Simple Linear Regression Algorithm 4.3 Classification: Binary, Multi class and Multi label(Only definition) 4.4 Classification Algorithm: K-Nearest Neighbors, Logistic Regression, Support vector machine, Decision tree, Random forest(No mathematical derivation, only key points of each algorithm)

4.1 Supervised Learning:

Supervised learning is when the model is getting trained on a labelled dataset. A **labelled** dataset is one that has both input and output parameters. In this type of learning both training and validation datasets are labelled as shown in the figures below.

User ID	Gender	Age	Salary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	1
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	1
15728773	Male	27	58000	1
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	1
15727311	Female	35	65000	0
15570769	Female	26	80000	1
15606274	Female	26	52000	0
15746139	Male	20	86000	1
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1

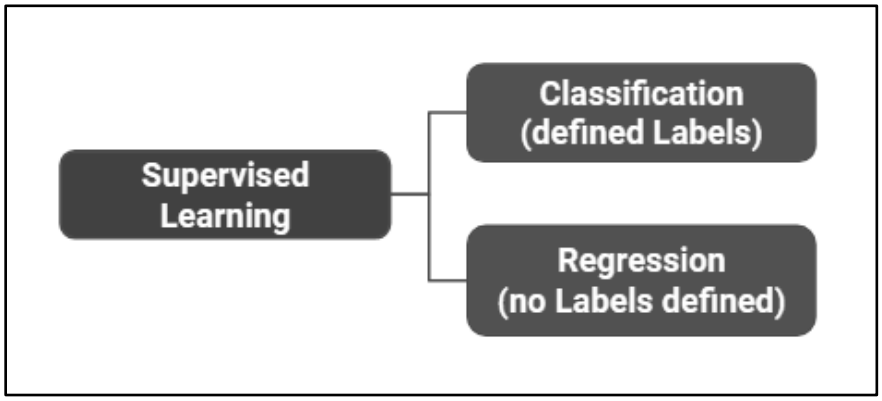
Figure A: CLASSIFICATION

Temperature	Pressure	Relative Humidity	Wind Direction	Wind Speed
10.69261758	986.882019	54.19337313	195.7150879	3.278597116
13.59184184	987.8729248	48.0648859	189.2951202	2.909167767
17.70494885	988.1119385	39.11965597	192.9273834	2.973036289
20.95430404	987.8500366	30.66273218	202.0752869	2.965289593
22.9278274	987.2833862	26.06723423	210.6589203	2.798230886
24.04233986	986.2907104	23.46918024	221.1188507	2.627005816
24.41475295	985.2338867	22.25082295	233.7911987	2.448749781
23.93361956	984.8914795	22.35178837	244.3504333	2.454271793
22.68800023	984.8461304	23.7538641	253.0864716	2.418341875
20.56425726	984.8380737	27.07867944	264.5071106	2.318677425
17.76400389	985.4262085	33.54900114	280.7827454	2.343950987
11.25680746	988.9386597	53.74139903	68.15406036	1.650191426
14.37810685	989.6819458	40.70884681	72.62069702	1.553469896
18.45114201	990.2960205	30.85038484	71.70604706	1.005017161
22.54895853	989.9562988	22.81738811	44.66042709	0.264133632
24.23155922	988.796875	19.74790765	318.3214111	0.329656571

Figure B: REGRESSION

Both the above figures have labelled data set as follows:

- **Figure A:** It is a dataset of a shopping store that is useful in predicting whether a customer will purchase a particular product under consideration or not based on his/ her gender, age, and salary.
  Input: Gender, Age, Salary
  Output: Purchased i.e. 0 or 1; 1 means yes the customer will purchase and 0 means that the customer won't purchase it.
- **Figure B:** It is a Meteorological dataset that serves the purpose of predicting wind speed based on different parameters.
  Input: Dew Point, Temperature, Pressure, Relative Humidity, Wind Direction
  Output: Wind Speed



Types of Supervised Learning:

A. Classification:

- It is a Supervised Learning task where output is having defined labels(discrete value).
- For example in above **Figure A**, Output – Purchased has defined labels i.e. 0 or 1; 1 means the customer will purchase, and 0 means that the customer won't purchase. The goal here is to predict discrete values belonging to a particular class and evaluate them on the basis of accuracy.
- It can be either binary or multi-class classification. In **binary** classification, the model predicts either 0 or 1; yes or no but in the case of **multi-class** classification, the model predicts more than one class.
- **Example:** Gmail classifies mails in more than one class like social, promotions, updates, and forums.

B. Regression:

- It is a Supervised Learning task where output is having continuous value.
- For example in above **Figure B**, Output – Wind Speed is not having any discrete value but is continuous in a particular range. The goal here is to predict a value as much closer to the actual output value as our model can and then evaluation is done by calculating the error value. The smaller the error the greater the accuracy of our regression model.

Example of Supervised Learning Algorithms:

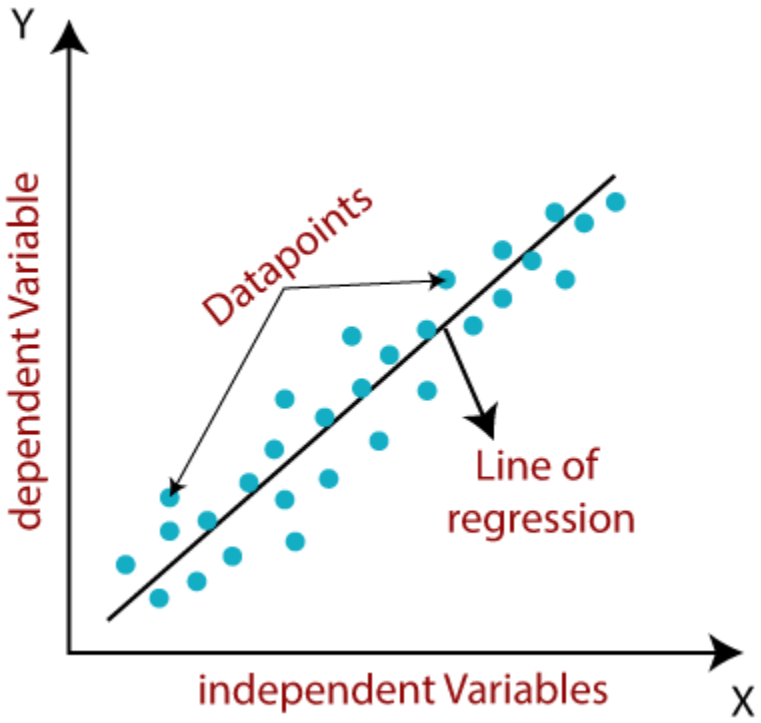
- Linear Regression
- Logistic Regression

- Nearest Neighbor
- Gaussian Naive Bayes
- Decision Trees
- Support Vector Machine (SVM)
- Random Forest

4.2 Regression Algorithm:

Linear Regression

- Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.



- Mathematically, we can represent a linear regression as:

$$y= a_0+a_1x+ \varepsilon$$

Y=Dependent Variable (Target Variable)
X=Independent Variable (predictor Variable)
a0=intercept of the line (Gives an additional degree of freedom)
a1=Linear regression coefficient (scale factor to each input value).
ε = random error

- **Types of Linear Regression**

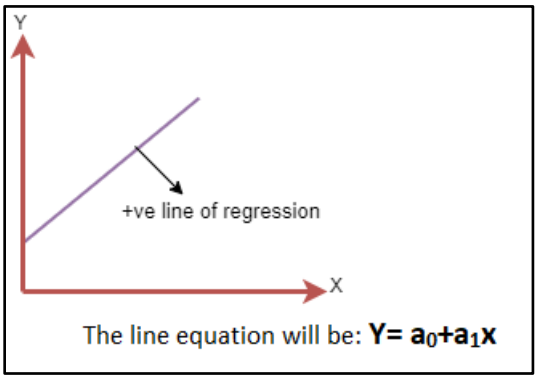
  Linear regression can be further divided into two types of the algorithm:

  - **Simple Linear Regression:**
    If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
  - **Multiple Linear regression:**
    If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

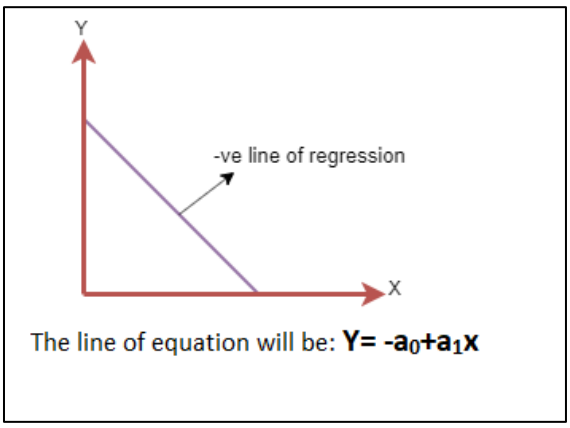
- **Linear Regression Line**

  A linear line showing the relationship between the dependent and independent variables is a **regression line**. A regression line can show two types of relationship:

  - **Positive Linear Relationship:**
    If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



  - **Negative Linear Relationship:**
    If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.



Implementation of simple Linear regression model:

Linear Regression

Importing Libraries

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

Importing the data

```
data = pd.read_csv('train_cleaned.csv')
data.head()
```

Segregating variables: Independent and Dependent Variables

```
x = data.drop(['Item_Outlet_Sales'], axis=1)
y = data['Item_Outlet_Sales']
x.shape, y.shape
```

Splitting the data into train set and the test set

 # Importing the train test split function

```
from sklearn.model_selection import train_test_split
train_x,test_x,train_y,test_y = train_test_split(x,y, random_state = 56)
```

 # Implementing Linear Regression

#importing Linear Regression and metric mean square error

```
from sklearn.linear_model import LinearRegression as LR
from sklearn.metrics import mean_absolute_error as mae
```

 # Creating instance of Linear Regresssion
lr = LR()

 # Fitting the model
lr.fit(train_x, train_y)

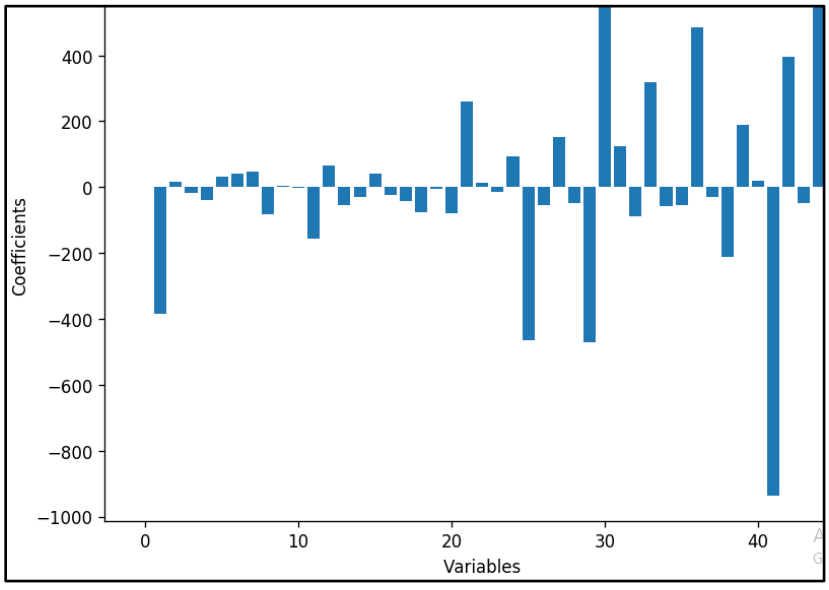
 # Predicting over the Train Set and calculating error
train_predict = lr.predict(train_x)
k = mae(train_predict, train_y)
print('Training Mean Absolute Error', k)

 # Predicting over the Test Set and calculating error
test_predict = lr.predict(test_x)
k = mae(test_predict, test_y)
print('Test Mean Absolute Error   ', k)

 # Predicting over the Test Set and calculating error
test_predict = lr.predict(test_x)
k = mae(test_predict, test_y)
print('Test Mean Absolute Error   ', k)

 # Parameters of Linear Regression
lr.coef_

 # Plotting the coefficients
plt.figure(figsize=(8, 6), dpi=120, facecolor='w', edgecolor='b')
x = range(len(train_x.columns))
y = lr.coef_
plt.bar(x, y)
plt.xlabel("Variables")
plt.ylabel('Coefficients')
plt.title('Coefficient plot')



Model Interpretability

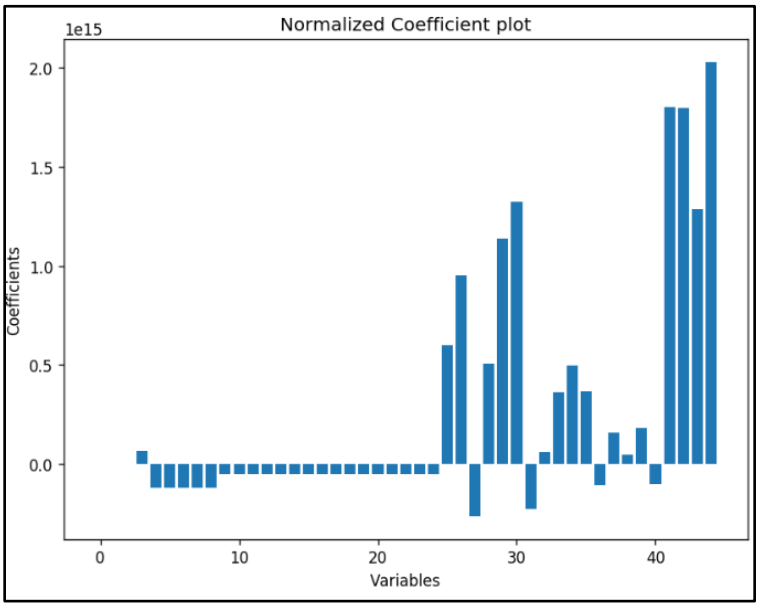
#So far we have simply been predicting the values using the linear regression, But in order to Interpret the model, the normalising of the data is essential.
 # Creating instance of Linear Regresssion
lr = LR(normalize = True)

 # Fitting the model
lr.fit(train_x, train_y)

 # Predicting over the Train Set and calculating error
train_predict = lr.predict(train_x)
k = mae(train_predict, train_y)
print('Training Mean Absolute Error', k)

 # Predicting over the Test Set and calculating error
test_predict = lr.predict(test_x)
k = mae(test_predict, test_y)
print('Test Mean Absolute Error ', k)

plt.figure(figsize=(8, 6), dpi=120, facecolor='w', edgecolor='b')
x = range(len(train_x.columns))
y = lr.coef_
plt.bar(x, y)
plt.xlabel("Variables")
plt.ylabel('Coefficients')
plt.title('Normalized Coefficient plot')



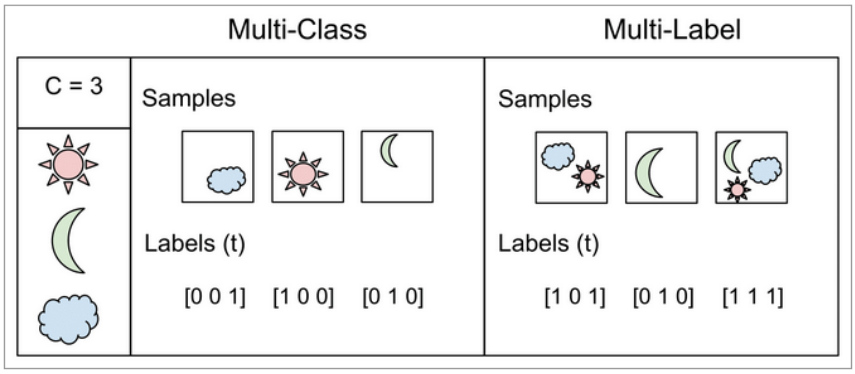
Now the coefficients we see are normalised and we can easily make final inferences out of it. Here we can see that there are a lot of Coefficients which are near to zero and not Significant.

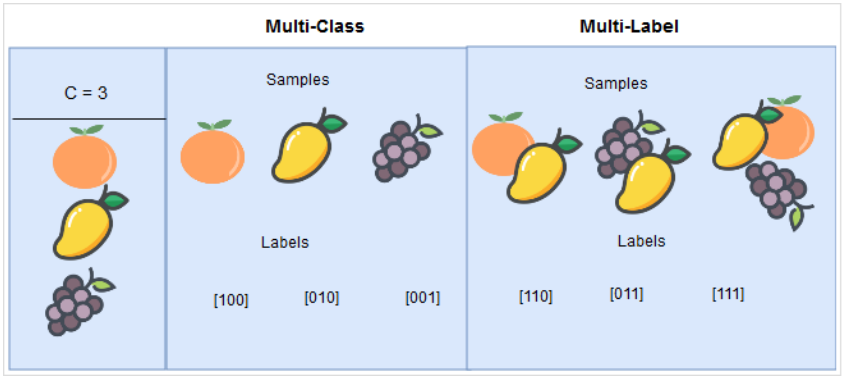
4.3 Binary, Multiclass and Multilabel classification:

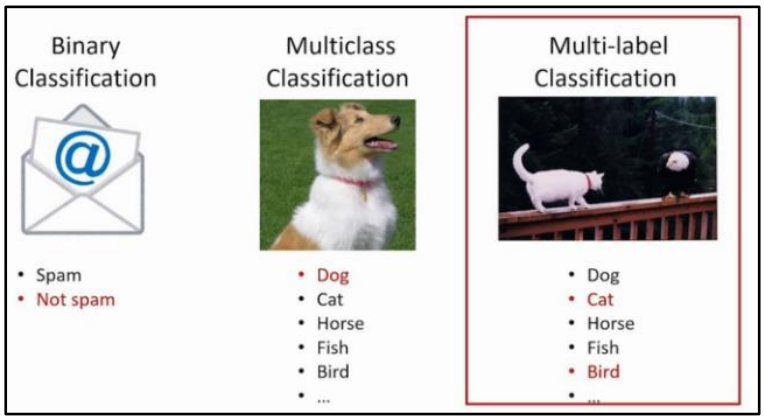
- **Binary classification:**
  - It refers to classification issues having two class labels.
  - Usually, binary classification tasks include two classes: one for the normal condition, denoted by the number 0, and another for the abnormal state, denoted by the number 1.
- **Multi-class classification**
  - It refers to classification issues in which each instance has only one of more than two class labels.
  - Multi-class classification does not have the concept of normal and abnormal outcomes, unlike binary classification. Instead, samples are grouped into one of several classes.
- **Multi-label classification**
  - It refers to classification tasks in which each instance has multiple classes.

Classification Type (CT)	Label(L)	Classification
CT=1	L=2	Binary
CT=1	L>2	Multiclass
CT>1	L>2	Multi Label

- Examples of multiclass and multilabel :







4.4 Classification Algorithm:

1. K-Nearest Neighbors:

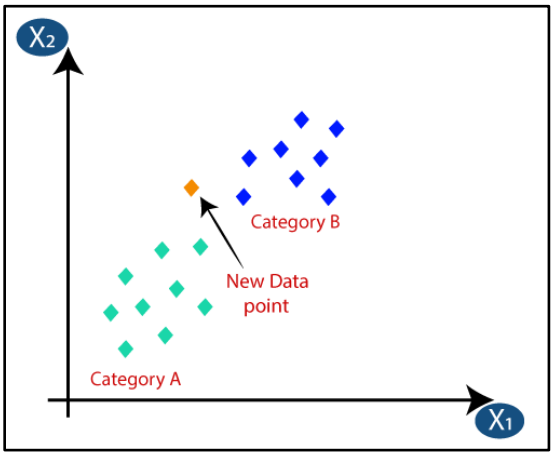
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- **How does K-NN work?**

  The K-NN working can be explained on the basis of the below algorithm:

  - **Step-1:** Select the number K of the neighbors
  - **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
  - **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
  - **Step-4:** Among these k neighbors, count the number of the data points in each category.
  - **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.

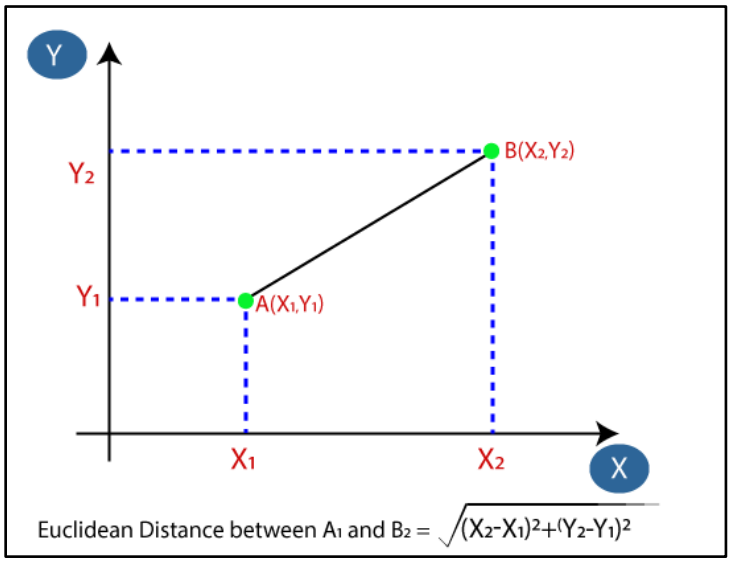
  Example:

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

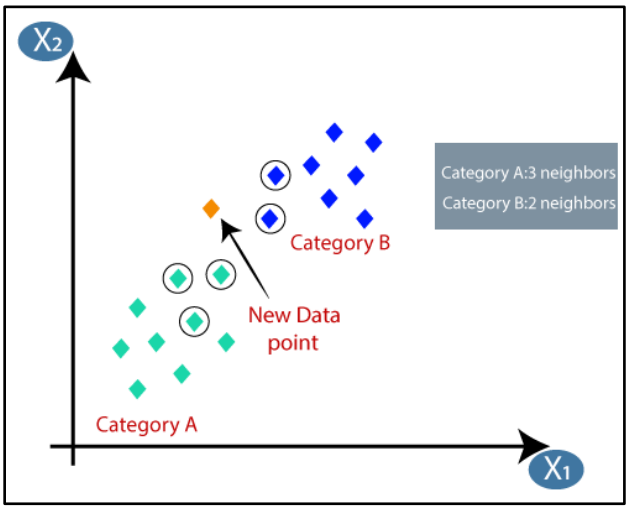


Firstly, we will choose the number of neighbors, so we will choose the k=5.

Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

Advantages :

- Easy and simple machine learning model.
- Few hyperparameters to tune.

Disadvantages :

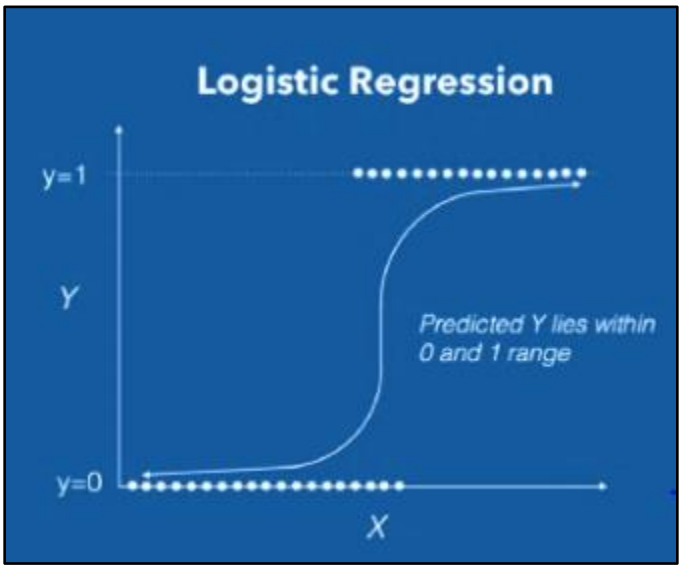
- computation cost during runtime if sample size is large.
- Proper scaling should k should be wisely selected.
- Large be provided for fair treatment among features.

2. Logistic Reression:

- Logistic regression is used for solving the classification problems**.**
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**. It is a predictive analysis algorithm and based on the **concept of probability.**

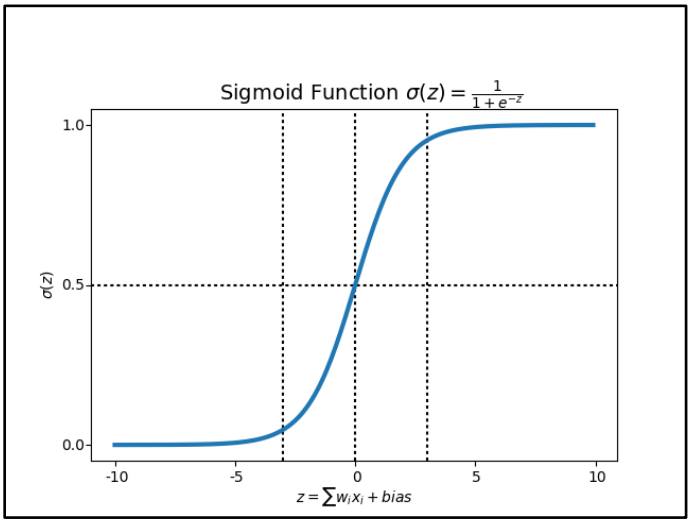
$$0 \leq h_\theta(x) \leq 1$$

Logistic regression hypothesis expectation



- Logistic Regression uses a more complex cost function, this cost function can be defined as the '**Sigmoid function**'.
- **What is the Sigmoid Function?**

  In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.



Sigmoid Function Graph

$$f(x) = \frac{1}{1+e^{-(x)}}$$

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

Advantages:

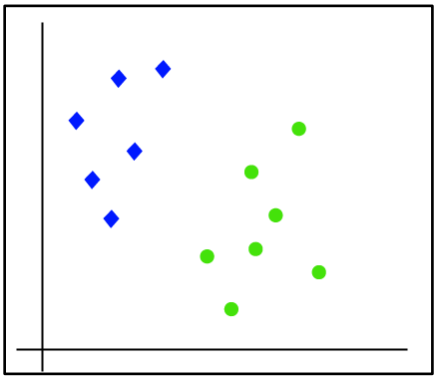
- Logistic regression is easier to implement, interpret, and very efficient to train.
- makes no assumptions about distributions of classes in feature space.
- It can easily extend to multiple classes(multinomial regression) and a natural probabilistic view of class predictions.
- It is very fast at classifying unknown records.
- Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.

Disadvantages

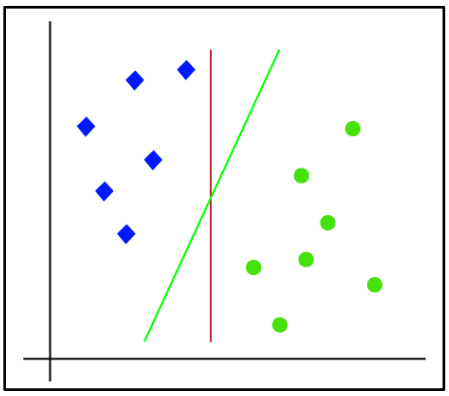
- If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.
- The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables.
- It can only be used to predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.

3. Support vector machine

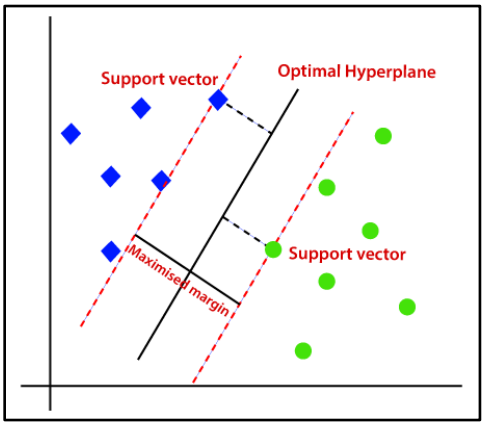
- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems.
- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a **hyperplane.**
- Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



- So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



- Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.

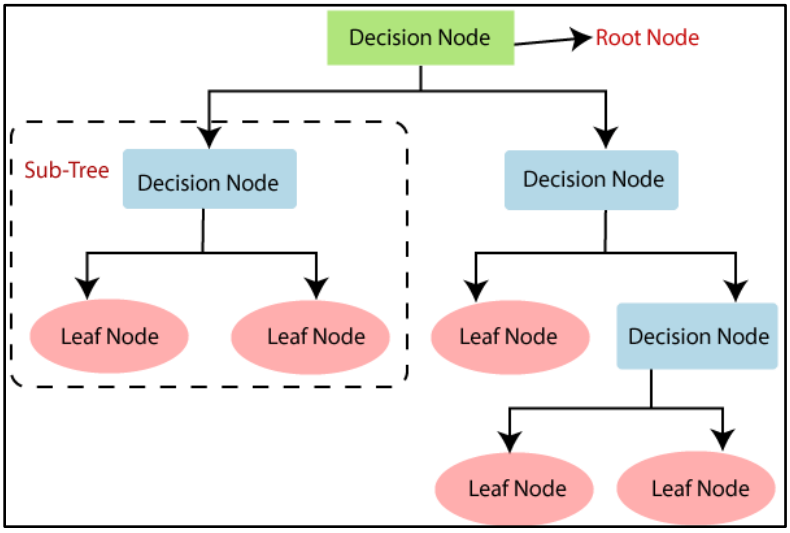


- **Advantages of SVM:**
  - Effective in high dimensional cases
  - Its memory efficient as it uses a subset of training points in the decision function called support vectors
  - Different kernel functions can be specified for the decision functions and its possible to specify custom kernels

4. Decision Tree Classification Algorithm

- 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.**

- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***



- 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.
- **How does the Decision Tree algorithm Work?**

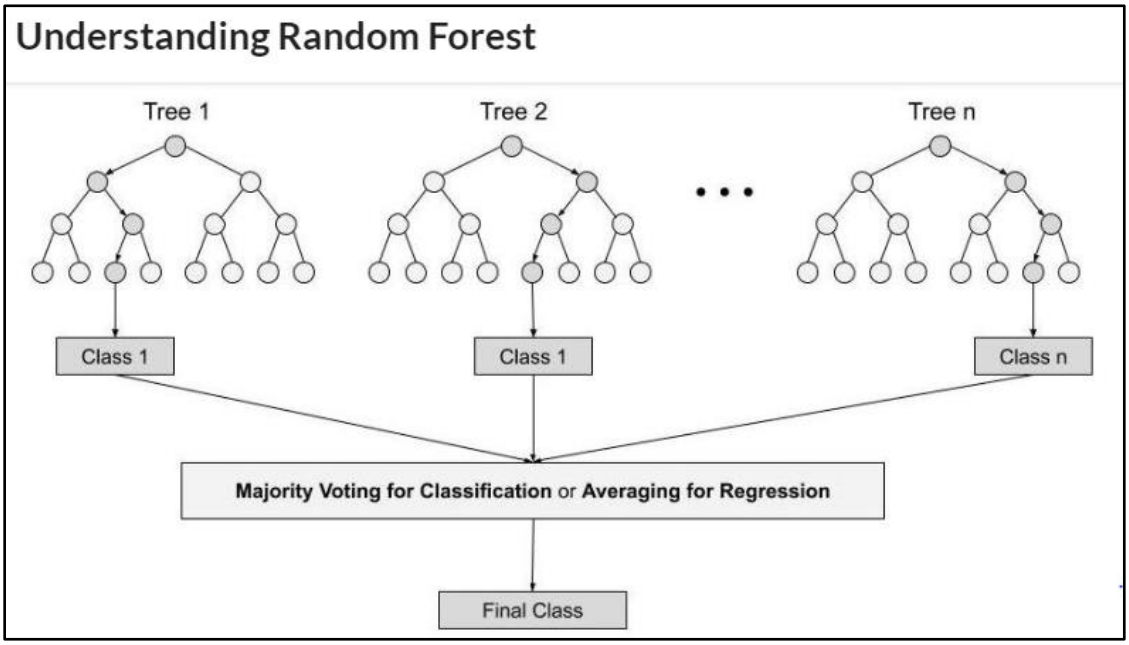
  - **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
  - **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
  - **Step-3:** Divide the S into subsets that contains possible values for the best attributes.

- **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.

- **Advantages:**
  - Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
  - A decision tree does not require normalization of data.
  - A decision tree does not require scaling of data as well.
  - Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
  - A Decision tree model is very intuitive and easy to explain to technical teams as well as stakeholders.
- Disadvantage:
  - A small change in the data can cause a large change in the structure of the decision tree causing instability.
  - For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
  - Decision tree often involves higher time to train the model.
  - Decision tree training is relatively expensive as the complexity and time has taken are more.

5. Random Forest Algorithm

- Random forest is a ***Supervised Machine Learning Algorithm*** that is ***used widely in Classification and Regression problems***.



- **Steps involved in random forest algorithm:**

- Step 1: In Random forest n number of random records are taken from the data set having k number of records.
- Step 2: Individual decision trees are constructed for each sample.
- Step 3: Each decision tree will generate an output.
- Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

- Advantages:
  - It reduces overfitting in decision trees and helps to improve the accuracy
  - It is flexible to both classification and regression problems
  - It works well with both categorical and continuous values
  - It automates missing values present in the data
  - Normalising of data is not required as it uses a rule-based approach.

- Disadvantages:
  - requires much computational power as well as resources as it builds numerous trees to combine their outputs.
  - It also requires much time for training as it combines a lot of decision trees to determine the class.
  - Due to the ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.

Difference Between Decision Tree & Random Forest

Parameters	Decision trees	Random Forest
Overfitting	1. Decision trees normally suffer from the problem of overfitting if it's allowed to grow without any control.	1. Random forests are created from subsets of data and the final output is based on average or majority ranking and hence the problem of overfitting is taken care of.
Speed	2. A single decision tree is faster in computation.	2. It is comparatively slower.
Formula	3. When a data set with features is taken as input by a decision tree it will formulate some set of rules to do prediction.	3. Random forest randomly selects observations, builds a decision tree and the average result is taken. It doesn't use any set of formulas.
Interpretability	Easy to interpret	Hard to interpret
Accuracy	Accuracy can vary	Highly accurate
Overfitting	Likely to overfit data	Unlikely to overfit data
Outliers	Can be highly affected by outliers	Robust against outliers
Computation	Quick to build	Slow to build (computationally intensive)

Differentiate Logistic regression and Support vector machine

S.No.	Logistic Regression	Support Vector Machine
1.	It is an algorithm used for solving classification problems.	It is a model used for both classification and regression.
2.	It is not used to find the best margin, instead, it can have different decision boundaries with different weights that are near the optimal point.	it tries to find the "best" margin (distance between the line and the support vectors) that separates the classes and thus reduces the risk of error on the data.
3.	It works with already identified identified independent variable.	It works well with unstructured and semi-structured data like text and images.
4.	It is based on statistical approach.	It is based on geometrical properties of the data.
5.	It is vulnerable to overfitting.	The risk of overfitting is less in SVM.
6.	Problems to apply logistic regression algorithm. 1. Cancer Detection: It can be used to detect if a patient has cancer(1) or not(0) 2. Test Score: Predict if the student is passed(1) or not(0). 3. Marketing: Predict if a customer will purchase a product(1) or not(0).	Problems that can be solved using SVM 1. Image Classification 2. Recognizing handwriting 3. Cancer Detection

2 MARKS QUSTIONS

1. State classification and regression.
2. Define supervised learning w.r.t., machine learning.
3. List any 4 supervised machine learning algorithms.
4. List any 2 advantages of Logistic regression algorithm.
5. List any 2 disadvantages of Logistic regression.
6. List any 2 advantages of SVM algorithm.
7. List any 2 disadvantages of SVM.
8. List any 2 advantages of K-NN algorithm.
9. List any 2 disadvantages of K-NN

10. List any 2 advantages of Decision Tree algorithm.
11. List any 2 disadvantages of Decision tree algorithm.
12. List any 2 advantages of Random Forest algorithm.
13. List any 2 disadvantages of Random Forest algorithm.

4 MARKS QUESTIONS

1. List any 4 key points of Linear regression
2. List any 4 key points of Logistic regression
3. List any 4 key points of K-NN.
4. List any 4 key points of SVM.
5. List any 4 key points of Decision tree.
6. List any 4 key points of Random Forest.
7. Explain Binary, Multiclass and Multilabel classification