## 3.1 Regression vs Classification in Machine Learning

- Regression and Classification algorithms are Supervised Learning algorithms.
- Both the algorithms are used for prediction in Machine learning and work with the labeled datasets. But the difference between both is how they are used for different machine learning problems.

The main difference between Regression and Classification algorithms are as follows:

| Parameter | Regression Algorithm | Classification Algorithm |
|---|---|---|
| Basic | Regression is the process of finding a model or function for distinguishing the data into continuous real values instead of using classes. | Classification is the process of finding or discovering a model or function which helps in separating the data into multiple categorical classes. |
| Involves prediction of | Continuous values | Discrete values |
| Nature of the predicted data | Ordered | Unordered |
| Method of calculation | by measurement of root mean square error | by measuring accuracy |
| Classification | The regression Algorithm can be further divided into Linear and Non-linear Regression. | The Classification algorithms can be divided into Binary Classifier and Multi-class Classifier. |

| | | |
|---|---|---|
| Finding | In Regression, we try to find the best fit line, which can predict the output more accurately. | In Classification, we try to find the decision boundary, which can divide the dataset into different classes. |
| Example Algorithms | Decision tree, logistic regression, etc. | Regression tree (Random forest), Linear regression, etc |

**Examples for Classification and Regression**

Let's consider a dataset that contains student information of a particular university. A regression algorithm can be used in this case to predict the height of any student based on their weight, gender, diet, or subject major. We use regression in this case because height is a continuous quantity. There is an infinite number of possible values for a person's height. On the contrary, classification can be used to analyse whether an email is a spam or not spam. The algorithm checks the keywords in an email and the sender's address is to find out the probability of the email being spam.

## ***Regression***

The most extensively used modelling technique is linear regression, which assumes a linear connection between a dependent variable (Y) and an independent variable (X). It employs a regression line, also known as a best-fit line.

## Types of Regression

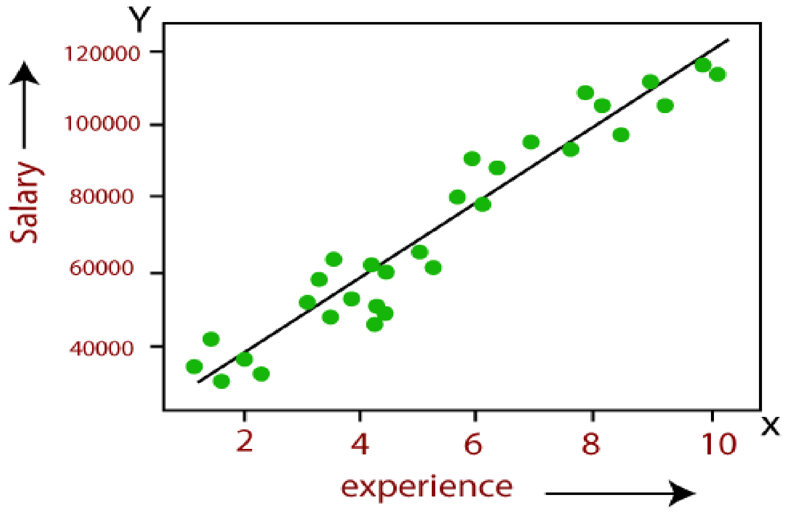
### 1. Simple Linear Regression

The simple linear regression is a model with only one dependent and one independent variable It tries to create a line that is as near to the data as possible by determining the slope and intercept, which define the line and reduce regression errors. There is a single x and y variable. The linear connection is defined by Equation: **Y = mX+c+e**

Where,
Y = Dependent Variable
m = Slope

X = Independent Variable
c = Intercept
e =error term.

- Linear Regression is one of the most simple Machine learning algorithm that comes under Supervised Learning technique and used for solving regression problems.
- It is used for predicting the continuous dependent variable with the help of independent variables.
- The goal of the Linear regression is to find the best fit line that can accurately predict the output for the continuous dependent variable.
- If single independent variable is used for prediction then it is called Simple Linear Regression and if there are more than two independent variables then such regression is called as Multiple Linear Regression.
- By finding the best fit line, algorithm establish the relationship between dependent variable and independent variable. And the relationship should be of linear nature.
- The output for Linear regression should only be the continuous values such as price, age, salary, etc. The relationship between the dependent variable and independent variable can be shown in below image:



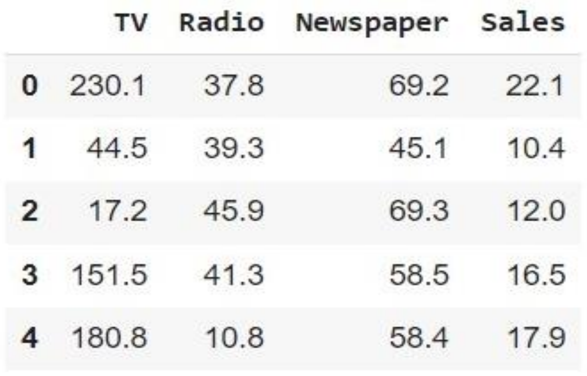
In above image the dependent variable is on Y-axis (salary) and independent variable is on x-axis (experience).

**2. Multiple Linear Regression**:

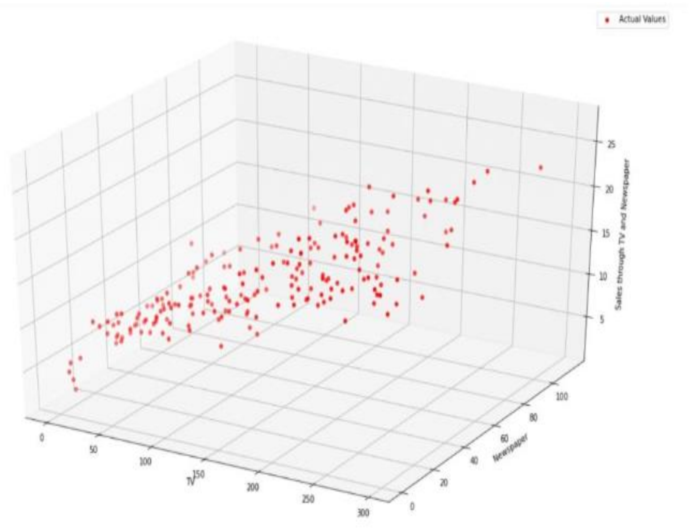
Multiple linear regression is a model with one or more x variables and one or more y variables, or one dependent variable and two or more independent variables .They are based on the presumption that both the dependent and independent variables, have a linear relationship. There are two types of multilinear regressions: linear and nonlinear.

Equation: **Y = m1X1+m2X2+m3X3+…..+c**

Multiple linear regression should be employed when numerous independent factors influence the outcome of a single dependent variable and when forecasting more complex interactions.



|   | TV | Radio | Newspaper | Sales |
|---|---|---|---|---|
| **0** | 230.1 | 37.8 | 69.2 | 22.1 |
| **1** | 44.5 | 39.3 | 45.1 | 10.4 |
| **2** | 17.2 | 45.9 | 69.3 | 12.0 |
| **3** | 151.5 | 41.3 | 58.5 | 16.5 |
| **4** | 180.8 | 10.8 | 58.4 | 17.9 |



## *3.2 Naïve Bayes Classifier*

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

### Why it is called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

### Bayes' Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

**Where,**

**P(A|B) is Posterior probability**: Probability of hypothesis A on the observed event B.

**P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.

**P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.

**P(B) is Marginal Probability**: Probability of Evidence.

## Working of Naïve Bayes' Classifier:

Working of Naïve Bayes' Classifier can be understood with the help of the below example:
Suppose we have a dataset of **weather conditions** and corresponding target variable "**Play**". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

**Problem**: If the weather is sunny, then the Player should play or not?

**Solution**: To solve this, first consider the below dataset:

| Instance | Weather | Play Tennis |
|---|---|---|
| **0** | Rainy | Yes |
| **1** | Sunny | Yes |
| **2** | Overcast | Yes |
| **3** | Overcast | Yes |
| **4** | Sunny | No |
| **5** | Rainy | Yes |
| **6** | Sunny | Yes |

| | | |
|---|---|---|
| **7** | Overcast | Yes |
| **8** | Rainy | No |
| **9** | Sunny | No |
| **10** | Sunny | Yes |
| **11** | Rainy | No |
| **12** | Overcast | Yes |
| **13** | Overcast | Yes |

**Frequency table for the Weather Conditions:**

| Weather | Yes | No |
|---|---|---|
| Overcast | 5 | 0 |
| Rainy | 2 | 2 |
| Sunny | 3 | 2 |
| Total | 10 | 5 |

**Likelihood table for weather condition:**

| Weather | No | Yes | |
|---|---|---|---|
| Overcast | 0 | 5 | 5/14= 0.35 |
| Rainy | 2 | 2 | 4/14=0.29 |
| Sunny | 2 | 3 | 5/14=0.35 |
| All | 4/14=0.29 | 10/14=0.71 | |

**Applying Bayes'theorem:**

**P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)**

P(Sunny|Yes)= 3/10= 0.3

P(Sunny)= 0.35

P(Yes)=0.71

So P(Yes|Sunny) = 0.3*0.71/0.35= **0.60**

**P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)**

P(Sunny|NO)= 2/4=0.5

P(No)= 0.29

P(Sunny)= 0.35

So P(No|Sunny)= 0.5*0.29/0.35 = **0.41**

So as we can see from the above calculation that **P(Yes|Sunny)>P(No|Sunny)**

**Hence on a Sunny day, Player can play the game.**

## Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for **text classification problems**.

## Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all predictors (or features) are independent, rarely happening in real life. This limits the applicability of this algorithm in real-world use cases.
- This algorithm faces the ‘zero-frequency problem’ where it assigns zero probability to a categorical variable whose category in the test data set wasn’t available in the training dataset. It would be best if you used a smoothing technique to overcome this issue.

- Its estimations can be wrong in some cases, so you shouldn't take its probability outputs very seriously.

## Applications of Naïve Bayes Classifier:

- It is used for **Credit Scoring**.
- It is used in **medical data classification**.
- It can be used in **real-time predictions** because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as **Spam filtering** and **Sentiment analysis**.

## Types of Naïve Bayes Model:

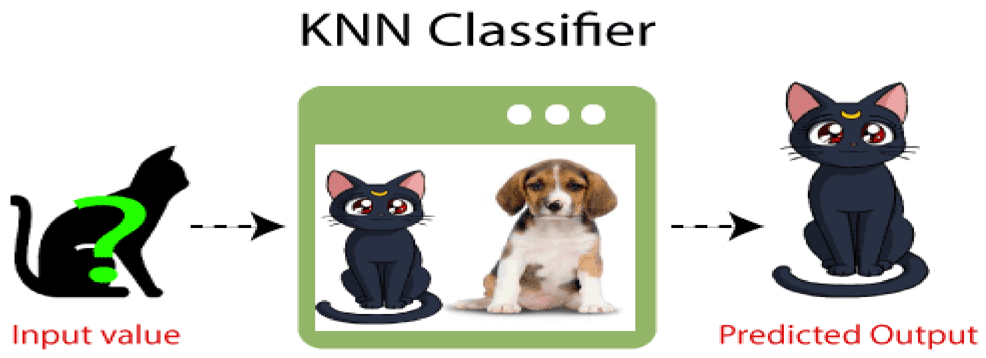
There are three types of Naive Bayes Model, which are given below:

- **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.
- **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.
- **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

## *3.3 K-Nearest Neighbours (KNN):*

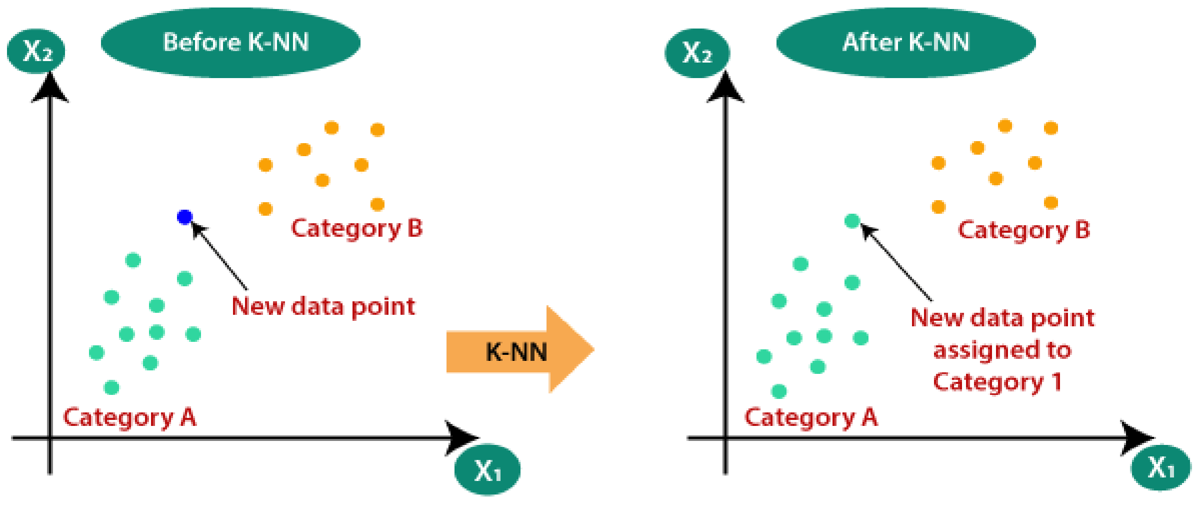
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- 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.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



## Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

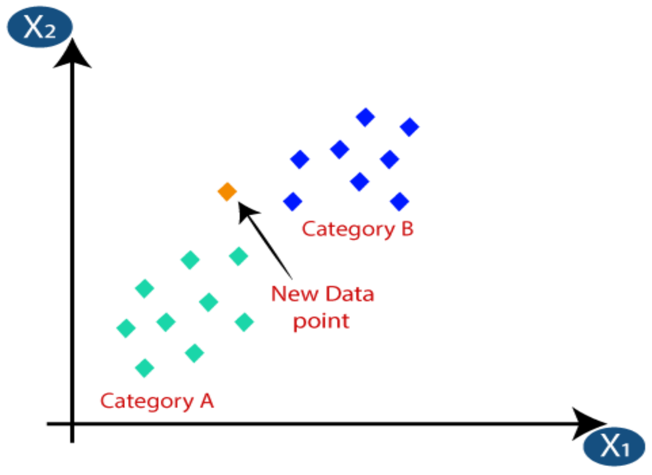


## How does K-NN work?

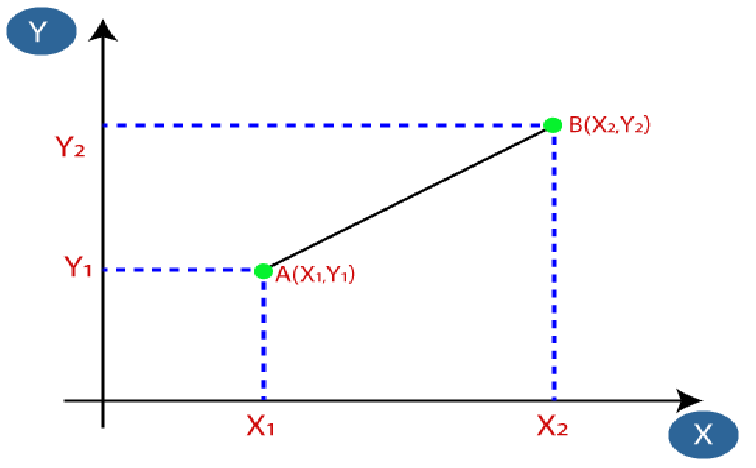
The K-NN working can be explained on the basis of the below algorithm:

- **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.
- **Step-6:** Our model is ready.

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

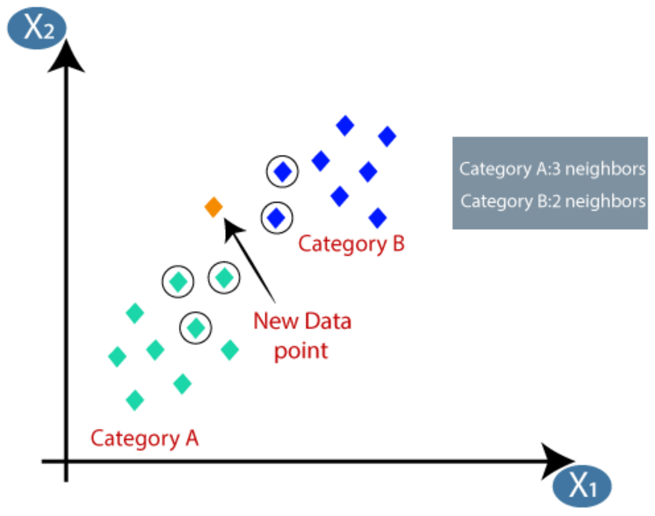


- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



Euclidean Distance between $A_1$ and $B_2$ = $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

- 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.

## How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm.

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

## Advantages of KNN Algorithm:

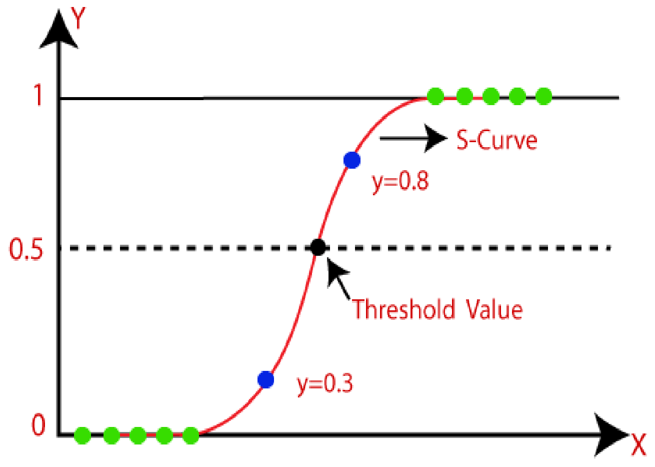
- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

## Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

## 3.4 Logistic Regression

- Logistic regression is one of the most popular Machine learning algorithm that comes under Supervised Learning techniques.
- It can be used for Classification as well as for Regression problems, but mainly used for Classification problems.
- Logistic regression is used to predict the categorical dependent variable with the help of independent variables.
- The output of Logistic Regression problem can be only between the 0 and 1.
- Logistic regression can be used where the probabilities between two classes is required. Such as whether it will rain today or not, either 0 or 1, true or false etc.
- Logistic regression is based on the concept of Maximum Likelihood estimation. According to this estimation, the observed data should be most probable.
- In logistic regression, we pass the weighted sum of inputs through an activation function that can map values in between 0 and 1. Such activation function is known as **sigmoid function** and the curve obtained is called as sigmoid curve or S-curve. Consider the below image:



- The equation for logistic regression is:

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

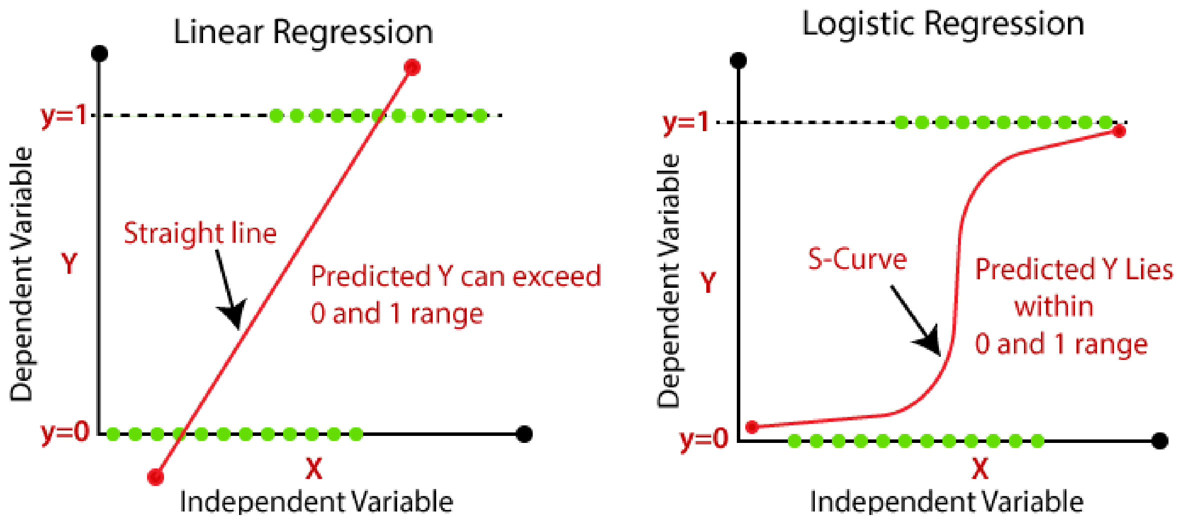
## Comparision between Linear Regression and Logistic Regression:

Linear Regression and Logistic Regression are the two famous Machine Learning Algorithms which come under supervised learning technique. Since both the algorithms are of supervised in nature hence these algorithms use labeled dataset to make the predictions. But the main difference between them is how they are being used. The Linear Regression is used for solving Regression problems whereas Logistic Regression is used for solving the Classification problems. The description of both the algorithms is given below along with difference table.

Difference between Linear Regression and Logistic Regression:

| Linear Regression | Logistic Regression |
|---|---|
| Linear regression is used to predict the continuous dependent variable using a given set of independent variables. | Logistic Regression is used to predict the categorical dependent variable using a given set of independent variables. |
| Linear Regression is used for solving Regression problem. | Logistic regression is used for solving Classification problems. |
| In Linear regression, we predict the value of continuous variables. | In logistic Regression, we predict the values of categorical variables. |
| In linear regression, we find the best fit line, by which we can easily predict the output. | In Logistic Regression, we find the S-curve by which we can classify the samples. |
| Least square estimation method is used for estimation of accuracy. | Maximum likelihood estimation method is used for estimation of accuracy. |
| The output for Linear Regression must be a continuous value, such as price, age, etc. | The output of Logistic Regression must be a Categorical value such as 0 or 1, Yes or No, etc. |

| | |
|---|---|
| In Linear regression, it is required that relationship between dependent variable and independent variable must be linear. | In Logistic regression, it is not required to have the linear relationship between the dependent and independent variable. |
| In linear regression, there may be collinearity between the independent variables. | In logistic regression, there should not be collinearity between the independent variable. |

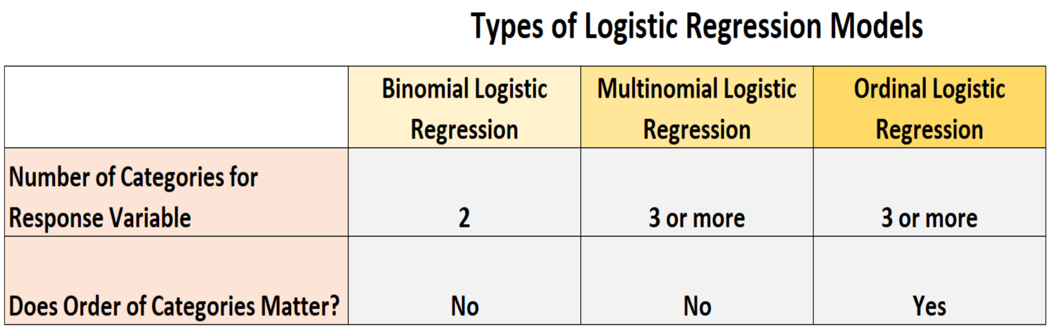


## Types of logistic regression

There are three types of logistic regression models:

- **Binary logistic regression**: The response variable can only belong to one of two categories.
- **Multinomial logistic regression**: The response variable can belong to one of three or more categories and there is no natural ordering among the categories.
- **Ordinal logistic regression**: The response variable can belong to one of three or more categories and there *is* a natural ordering among the categories.

The following table summarizes these differences:



**Types of Logistic Regression Models**

| | Binomial Logistic Regression | Multinomial Logistic Regression | Ordinal Logistic Regression |
|---|---|---|---|
| Number of Categories for Response Variable | 2 | 3 or more | 3 or more |
| Does Order of Categories Matter? | No | No | Yes |

**Binary Logistic Regression**

Binary logistic regression models are a type of logistic regression in which the response variable can only belong to two categories. Here are a couple examples:

<u>Example: Spam Detection</u>

Suppose a business wants to use the predictor variables (1) word count and (2) country of origin to predict the probability that a given email is spam.Since there are only two possible outcomes (spam or not spam) for the response variable, the business would use a binomial logistic regression model.

**Multinomial Logistic Regression**

Multinomial logistic regression models are a type of logistic regression in which the response variable can belong to one of three or more categories and there is no natural ordering among the categories.

<u>Example 1: Political Preference</u>

Suppose a political scientist wants to use the predictor variables (1) annual income and (2) years of education to predict the probability that an individual will vote for one of four different presidential candidates. Since there are more than two possible outcomes (there are four potential candidates) for the response variable and there is no natural ordering among the outcomes, the political scientist would use a multinomial logistic regression model.

<u>Example 2: Sports Preference</u>

Suppose a sports analyst wants to use the predictor variables (1) TV hours viewed per week and (2) age to predict the probability that an individual will pick either basketball, football, or baseball as their preferred sport.Since there are more than two possible outcomes (there are three sports) for the response variable, the sports analyst would use a multinomial logistic regression model.

**Ordinal Logistic Regression**

Ordinal logistic regression models are a type of logistic regression in which the response variable can belong to one of three or more categories and there *is* a natural ordering among the categories. Here are a couple examples:

Example 1: School Ratings

Suppose an academic advisor wants to use the predictor variables (1) GPA, (2) ACT score, and (3) SAT score to predict the probability that an individual will get into a university that can be categorized into “bad”, “mediocre”, “good”, or “great.” Since there are more than two possible outcomes (there are four classifications of school quality) for the response variable and there *is* a natural ordering among the outcomes, the academic advisor would use an ordinal logistic regression model.

Example 2: Movie Ratings

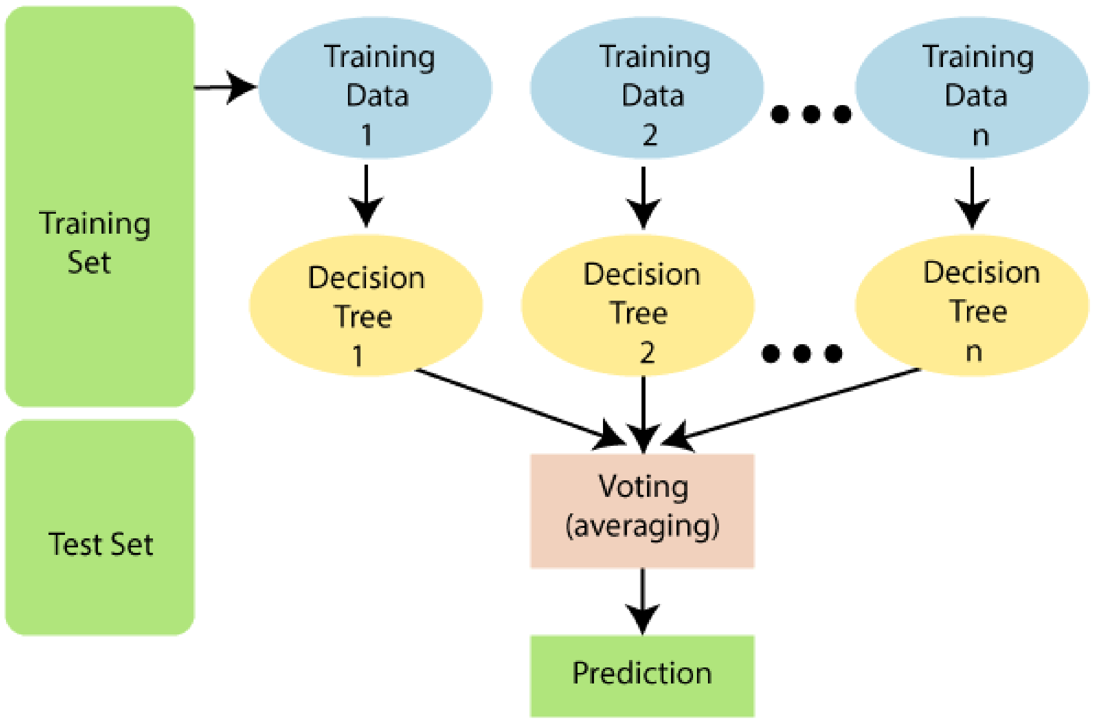
Suppose a movie critic wants to use the predictor variables (1) total run time and (2) genre to predict the probability that a given movie will receiving a rating between 1 and 10. Since there are more than two possible outcomes (there are 10 possible ratings) for the response variable and there *is* a natural ordering among the outcomes, the movie critic would use an ordinal logistic regression mode

## *3.5 Random Forest Algorithm*

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:



## Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

## Why to use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

## How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:
**Step-1:** Select random K data points from the training set.

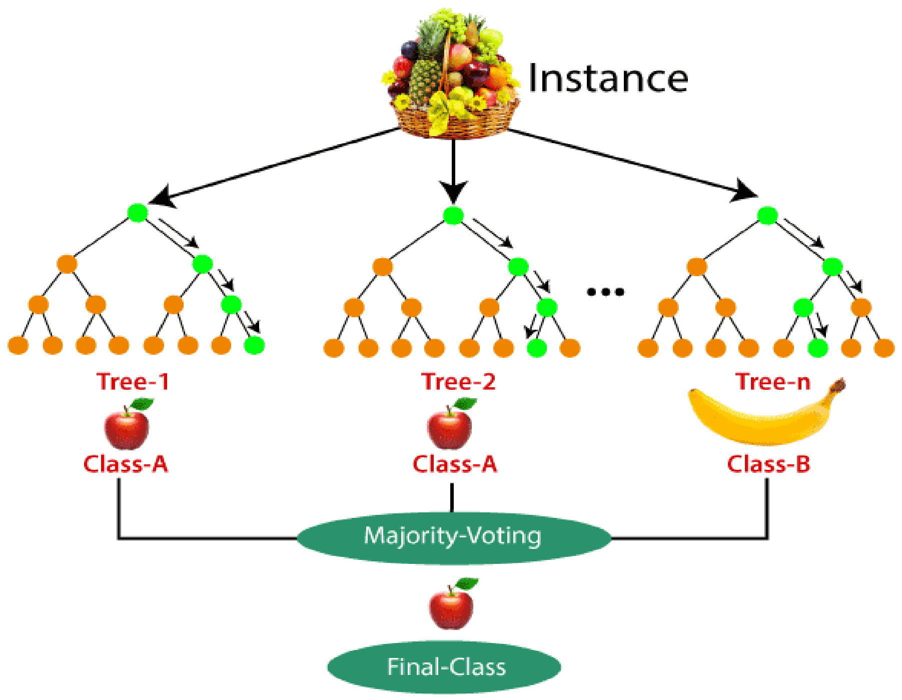
**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:
**Example:** Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



## Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

## Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

## Disadvantages of Random Forest

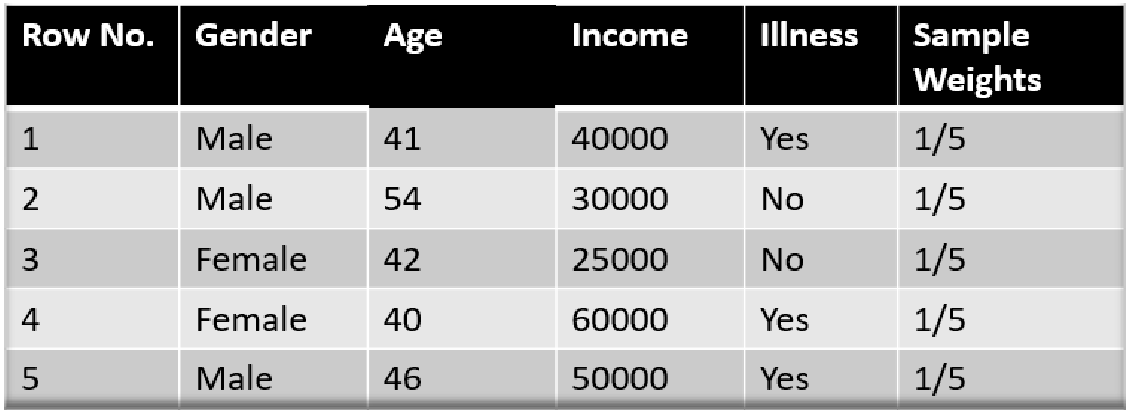
Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

## *3.5 Adaboost Algorithm*

AdaBoost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called **Decision Stumps.**What this algorithm does is that it builds a model and gives equal weights to all the data points. It then assigns higher weights to points that are wrongly classified. Now all the points which have higher weights are given more importance in the next model. It will keep training models until and unless a lowe error is received.

The principle behind boosting algorithms is first we built a model on the training dataset, then a second model is built to rectify the errors present in the first model. This procedure is continued until and unless the errors are minimized, and the dataset is predicted correctly.

**Step 1** – The Image is shown below is the actual representation of our dataset. Since the target column is binary it is a classification problem. First of all these data points will be assigned some weights. Initially, all the weights will be equal.



| Row No. | Gender | Age | Income | Illness | Sample Weights |
|---|---|---|---|---|---|
| 1 | Male | 41 | 40000 | Yes | 1/5 |
| 2 | Male | 54 | 30000 | No | 1/5 |
| 3 | Female | 42 | 25000 | No | 1/5 |
| 4 | Female | 40 | 60000 | Yes | 1/5 |
| 5 | Male | 46 | 50000 | Yes | 1/5 |

The formula to calculate the sample weights is:

$$w(x_i, y_i) = \frac{1}{N}, \ i = 1, 2, \ldots . n$$

Where N is the total number of datapoints.

Here since we have 5 data points so the sample weights assigned will be 1/5.

**Step 2** – We start by seeing how well "*Gender*" classifies the samples and will see how the variables (Age, Income) classifies the samples. We'll create a decision stump for each of the features and then calculate the ***Gini Index*** of each tree. The tree with the lowest Gini Index will be our first stump. Here in our dataset let's say ***Gender*** has the lowest gini index so it will be our first stump.

**Step 3** – We'll now calculate the **"Amount of Say"** or **"Importance"** or **"Influence"** for this classifier in classifying the datapoints using this formula:

$$\frac{1}{2}\log\frac{1 - Total\ Error}{Total\ Error}$$

The total error is nothing, but the summation of all the sample weights of misclassified data points.Here in our dataset let's assume there is 1 wrong output, so our total error will be 1/5, and alpha(performance of the stump) will be:

$$Performance\ of\ the\ stump = \frac{1}{2}\log_e\left(\frac{1-Total\ Error}{Total\ Error}\right)$$

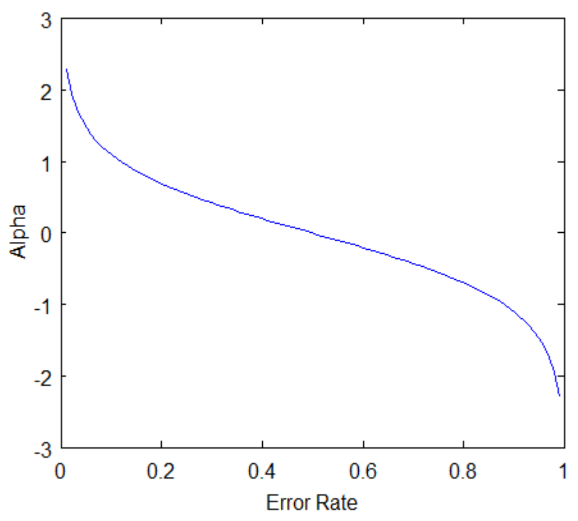
$$\alpha = \frac{1}{2}\log_e\left(\frac{1-\frac{1}{5}}{\frac{1}{5}}\right)$$

$$\alpha = \frac{1}{2}\log_e\left(\frac{0.8}{0.2}\right)$$

$$\alpha = \frac{1}{2}\log_e(4) = \frac{1}{2}*(1.38)$$

$$\alpha = 0.69$$

**Note**: Total error will always be between 0 and 1. 0 Indicates perfect stump and 1 indicates horrible stump.



From the graph above we can see that when there is no misclassification then we have no error (Total Error = 0), so the "amount of say (alpha)" will be a large number. When the classifier predicts half right and half wrong then the Total Error = 0.5 and the importance (amount of say) of the classifier will be 0. If all the samples have been incorrectly classified then the error will be very high (approx. to 1) and hence our alpha value will be a negative integer.

**Step 4** – You must be wondering why is it necessary to calculate the TE and performance of a stump? Well, the answer is very simple, we need to update the weights because if the same weights are applied to the next model, then the output received will be the same as what was received in the first model.

The wrong predictions will be given more weight whereas the correct predictions weights will be decreased. Now when we build our next model after updating the weights, more preference will be given to the points with higher weights.After finding the importance of the classifier and total error we need to finally update the weights and for this, we use the following formula:

$$New\ sample\ weight = old\ weight * e^{\pm Amount\ of\ say\ (\alpha)}$$

The amount of say (alpha) will be ***negative*** when the sample is **correctly classified**.

The amount of say (alpha) will be ***positive*** when the sample is **miss-classified.**

There are four correctly classified samples and 1 wrong, here the ***sample weight*** of that datapoint is *1/5* and the **amount of say/performance of the stump** of **Gender** is *0.69*.

New weights for correctly classified samples are:

$$New\ sample\ weight\ =\ \frac{1}{5}\ *\ \exp(-0.69)$$

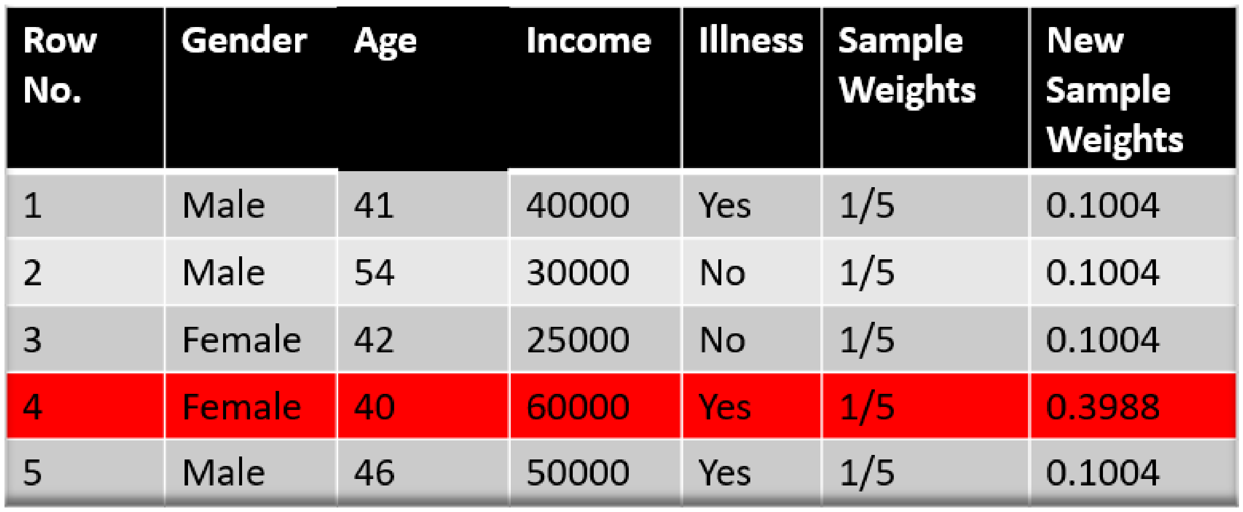
$$New\ sample\ weight\ =\ 0.2\ *\ 0.502\ =\ 0.1004$$

For *wrongly classified* samples the updated weights will be:

$$New\ sample\ weight\ =\ \frac{1}{5}\ *\ \exp(0.69)$$

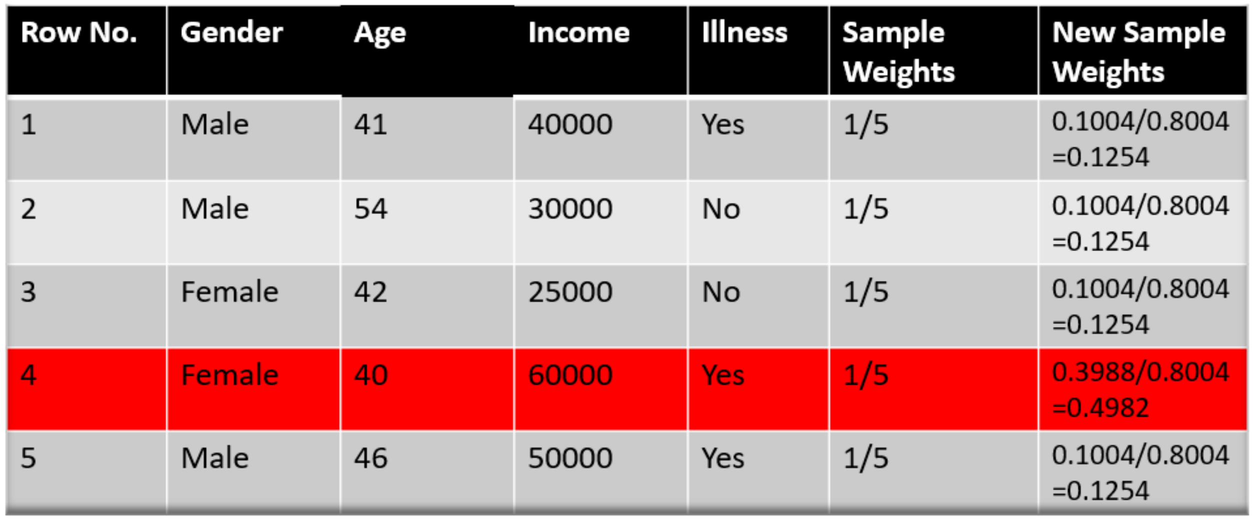
$$New\ sample\ weight\ =\ 0.2\ *\ 1.994\ =\ 0.3988$$

**Note:** See the sign of alpha when I am putting the values, the **alpha is negative** when the data point is correctly classified, and this *decreases the sample weight* from 0.2 to 0.1004. It is **positive** when there is **misclassification**, and this will *increase the sample weight* from 0.2 to 0.3988



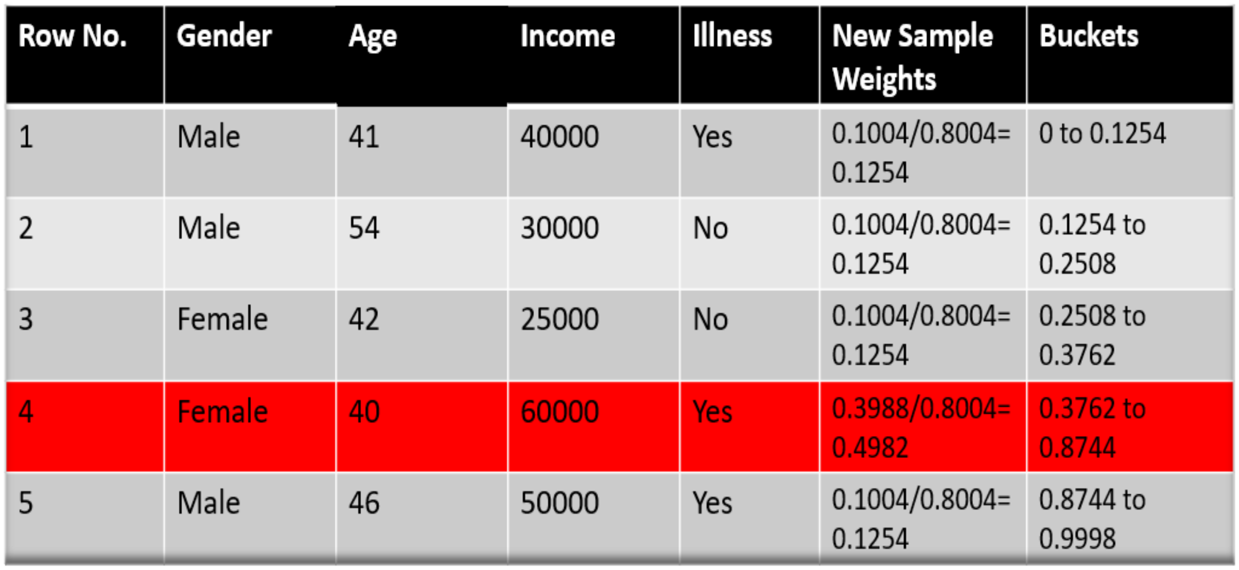
| Row No. | Gender | Age | Income | Illness | Sample Weights | New Sample Weights |
|---|---|---|---|---|---|---|
| 1 | Male | 41 | 40000 | Yes | 1/5 | 0.1004 |
| 2 | Male | 54 | 30000 | No | 1/5 | 0.1004 |
| 3 | Female | 42 | 25000 | No | 1/5 | 0.1004 |
| 4 | Female | 40 | 60000 | Yes | 1/5 | 0.3988 |
| 5 | Male | 46 | 50000 | Yes | 1/5 | 0.1004 |

We know that the total sum of the sample weights must be equal to 1 but here if we sum up all the new sample weights, we will get 0.8004. To bring this sum equal to 1 we will normalize these weights by dividing all the weights by the total sum of updated weights that is 0.8004. So, after normalizing the sample weights we get this dataset and now the sum is equal to 1.



| Row No. | Gender | Age | Income | Illness | Sample Weights | New Sample Weights |
|---|---|---|---|---|---|---|
| 1 | Male | 41 | 40000 | Yes | 1/5 | 0.1004/0.8004 =0.1254 |
| 2 | Male | 54 | 30000 | No | 1/5 | 0.1004/0.8004 =0.1254 |
| 3 | Female | 42 | 25000 | No | 1/5 | 0.1004/0.8004 =0.1254 |
| 4 | Female | 40 | 60000 | Yes | 1/5 | 0.3988/0.8004 =0.4982 |
| 5 | Male | 46 | 50000 | Yes | 1/5 | 0.1004/0.8004 =0.1254 |

**Step 5** – Now we need to make a new dataset to see if the errors decreased or not. For this we will remove the "sample weights" and "new sample weights" column and then based on the "new sample weights" we will divide our data points into buckets.



| Row No. | Gender | Age | Income | Illness | New Sample Weights | Buckets |
|---|---|---|---|---|---|---|
| 1 | Male | 41 | 40000 | Yes | 0.1004/0.8004= 0.1254 | 0 to 0.1254 |
| 2 | Male | 54 | 30000 | No | 0.1004/0.8004= 0.1254 | 0.1254 to 0.2508 |
| 3 | Female | 42 | 25000 | No | 0.1004/0.8004= 0.1254 | 0.2508 to 0.3762 |
| 4 | Female | 40 | 60000 | Yes | 0.3988/0.8004= 0.4982 | 0.3762 to 0.8744 |
| 5 | Male | 46 | 50000 | Yes | 0.1004/0.8004= 0.1254 | 0.8744 to 0.9998 |

**Step 6** – We are almost done, now what the algorithm does is selects random numbers from 0-1. Since incorrectly classified records have higher sample weights, the probability to select those records is very high. Suppose the 5 random numbers our algorithm take is

0.38,0.26,0.98,0.40,0.55. Now we will see where these random numbers fall in the bucket and according to it, we'll make our new dataset shown below.

| Row No. | Gender | Age | Income | Illness |
|---|---|---|---|---|
| 1 | Female | 40 | 60000 | Yes |
| 2 | Male | 54 | 30000 | No |
| 3 | Female | 42 | 25000 | No |
| 4 | Female | 40 | 60000 | Yes |
| 5 | Female | 40 | 60000 | Yes |

This comes out to be our new dataset and we see the datapoint which was wrongly classified has been selected 3 times because it has a higher weight.

**Step 9** – Now this act as our new dataset and we need to repeat all the above steps i.e.

1. Assign ***equal weights*** to all the datapoints
2. Find the stump that does the ***best job classifying*** the new collection of samples by finding their Gini Index and selecting the one with the lowest Gini index
3. Calculate the ***"Amount of Say"*** and ***"Total error"*** to update the previous sample weights.
4. Normalize the new sample weights.

Iterate through these steps until and unless a low training error is achieved.
Suppose with respect to our dataset we have constructed 3 decision trees (DT1, DT2, DT3) in a ***sequential manner.*** If we send our **test data** now it will pass through all the decision trees and finally, we will see which class has the majority, and based on that we will do predictions

## Advantages of Boosting Algorithms:

- Boosting algorithms follow ensemble learning which enables a model to give a more accurate prediction that cannot be trumped.

- Boosting algorithms are much more flexible than other algorithms as can optimize different loss functions and provides several hyperparameter tuning options.
- It does not require data pre-processing because it is suitable for both numeric as well as categorical variables.
- It does not require imputation of missing values in the dataset, it handles missing data automatically.

## Disadvantages of Boosting Algorithms:

- Below are a few disadvantages of boosting algorithms:
- Boosting algorithms may cause overfitting as well as overemphasizing the outliers.
- Gradient boosting algorithm continuously focuses to minimize the errors and requires multiple trees hence, it is computationally expensive.
- It is a time-consuming and memory exhaustive algorithm.
- Less interpretative in nature, although this is easily addressed with various tools.