MACHINE LEARNING Notes - 201CS6T01

UNIT-III

TOPIC-1: Introduction- Ensemble Learning and Random Forest

Ensemble learning usually produces more accurate solutions than a single model would. This has been the case in a number of machine learning competitions and, where the winning solutions used ensemble methods.

Random forest is a popular supervised machine learning algorithm—used for both classification and regression problems. It is based on the concept of ensemble learning, which enables users to combine multiple classifiers to solve a complex problem and to also improve the performance of the model.

TOPIC-2: Voting Classifiers:

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.

It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting. The idea is instead of creating separate dedicated models and finding the accuracy for each them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class.

Voting Classifier supports two types of votings:

Hard Voting: In hard voting, the predicted output class is a class with the highest majority of votes i.e the class which had the highest probability of being predicted by each of the classifiers. Suppose three classifiers predicted the output class(A, A, B), so here the majority predicted A as output. Hence A will be the final prediction.

Soft Voting: In soft voting, the output class is the prediction based on the average of probability given to that class. Suppose given some input to three models, the prediction probability for class A = (0.30, 0.47, 0.53) and B = (0.20, 0.32, 0.40). So the average for class A is 0.4333 and B is 0.3067, the winner is clearly class A because it had the highest probability averaged by each classifier.

TOPIC-3: Ensemble Learning:

This is an adaptive learning methodology to combine several algorithms to extract better results than the individual performances.

The main motto of ensemble methods is as follows:

1. To decrease the variance(bagging&pasting).
2. To decrease the bias(boosting)
3. To improve predictions(stacking)

The Ensemble method can be applied in two ways:

1. **Sequential**: In this method, the dependency of base learners are exploited. This involves the evaluation and reweighing of the unimportant examples. e.g. Adaboost.

2. **Parallel:** In this method, the independence of the base learners are exploited. This involves the evaluation by simply averaging the outputs of the base learner. e.g. Random forest.

The ensemble can be of two types :

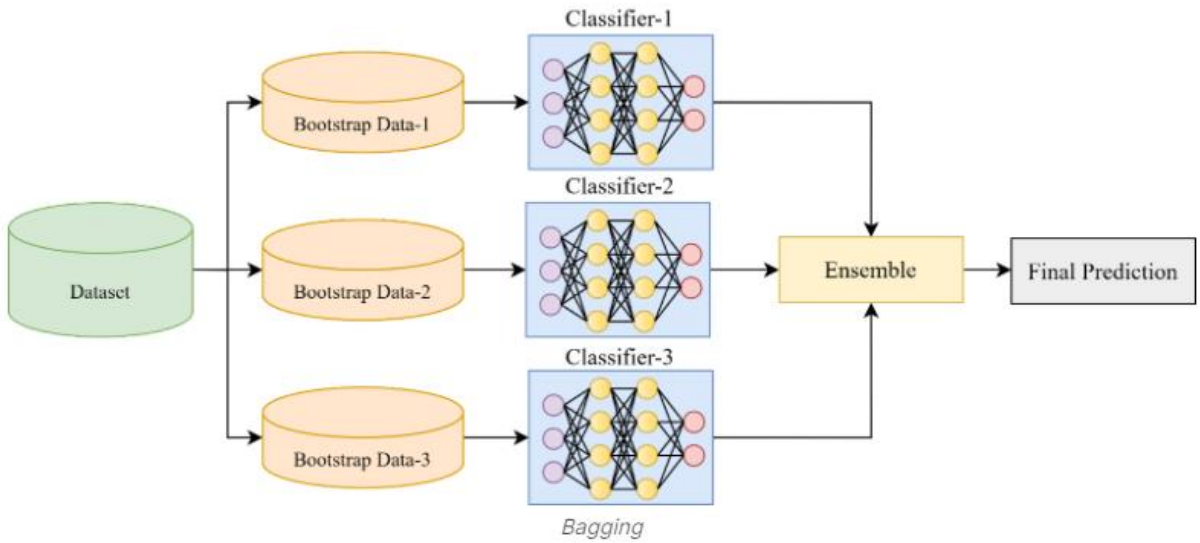
1. **Homogeneous ensemble:** In this, we have single base learning algorithm like random forest which only uses the decision tree algorithm.

2. **Heterogeneous ensemble :** In this, we have different base estimators algorithms.

Techniques used in the ensemble learning:

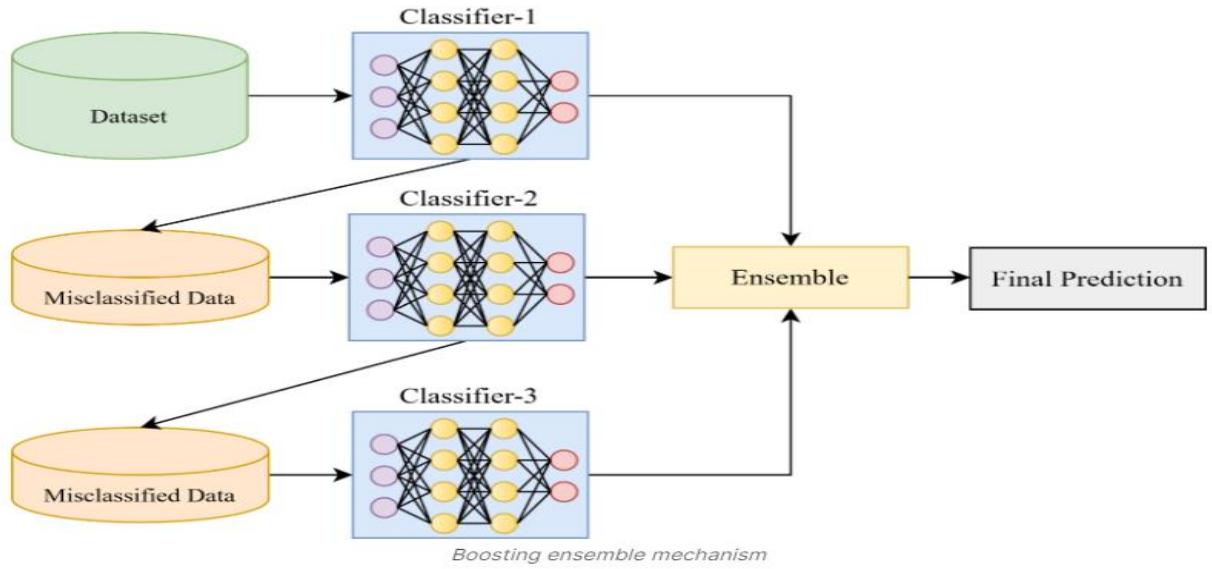
Bagging & Pasting:

Bagging means bootstrap+aggregating and it is a ensemble method in which we first bootstrap our data and for each bootstrap sample we train one model. After that, we aggregate them with equal weights. When it's not used replacement, the method is called pasting.



Boosting:

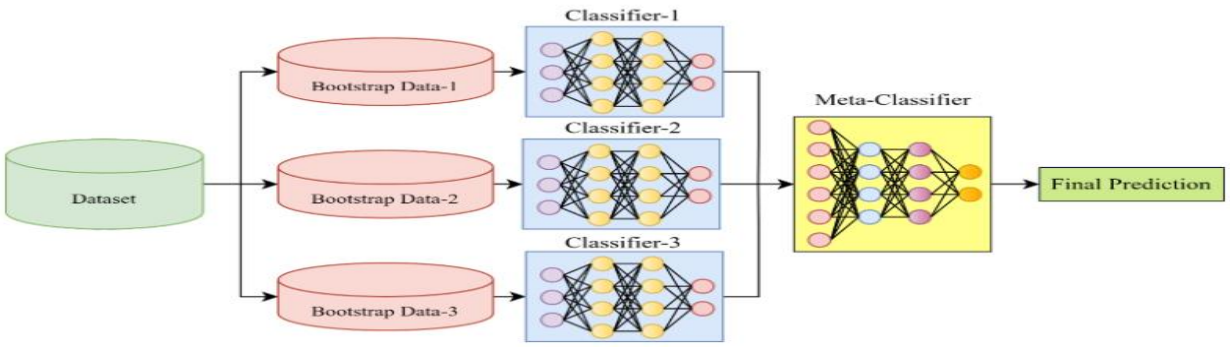
Boosting refers to a family of algorithms that are able to convert weak learners to strong learners. The predictions are then combined through a weighted majority vote (classification) or a weighted sum (regression) to produce the final prediction.



Boosting ensemble mechanism

Stacking:

Stacking is an ensemble learning technique that combines multiple classifications or regression models via a meta-classifier or a meta-regression. The base level models are trained based on a complete training set, then the meta-model is trained on the outputs of the base level model as features.



TOPIC-4: Random Forest Algorithm :

Random Forest algorithm is a supervised learning algorithm. There is a direct relationship between the number of trees in the forest and the results it can get. It uses a number of decision trees and predicts the more accurate result by averaging in case of regression and voting in case of classification.

How Random Forest algorithm works :

1. Randomly select "**K**" features from total "**m**" features where **k << m**
2. Among the "**K**" features, calculate the node "**d**" using the best split point
3. Split the node into n**odes** using the **best split**
4. Repeat the 1 **to 3** steps until "l" number of nodes has been reached
5. Build forest by repeating steps 1 **to 4** for "n" number times to create **"n" number of trees**

In the next stage, with the random forest classifier created, we will make the prediction.

1. Takes the **test features** and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)
2. Calculate the **votes** for each predicted target
3. Consider the **high voted** predicted target as the **final prediction** from the random forest algorithm

Advantages of Random Forest algorithm :

Compared with other classification techniques, there are three advantages :

1. For applications in classification problems, Random Forest algorithm will avoid the overfitting problem.
2. For both classification and regression task, the same random forest algorithm can be used.
3. The Random Forest algorithm can be used for identifying the most important features from the training dataset, in other words, feature engineering.
4. Versatile uses
5. Easy-to-understand hyperparameters
6. Classifier doesn't overfit with enough trees.

Disadvantages of Random Forest algorithm :

1. Increased accuracy requires more trees
2. More trees slow down model
3. Can't describe relationships within data

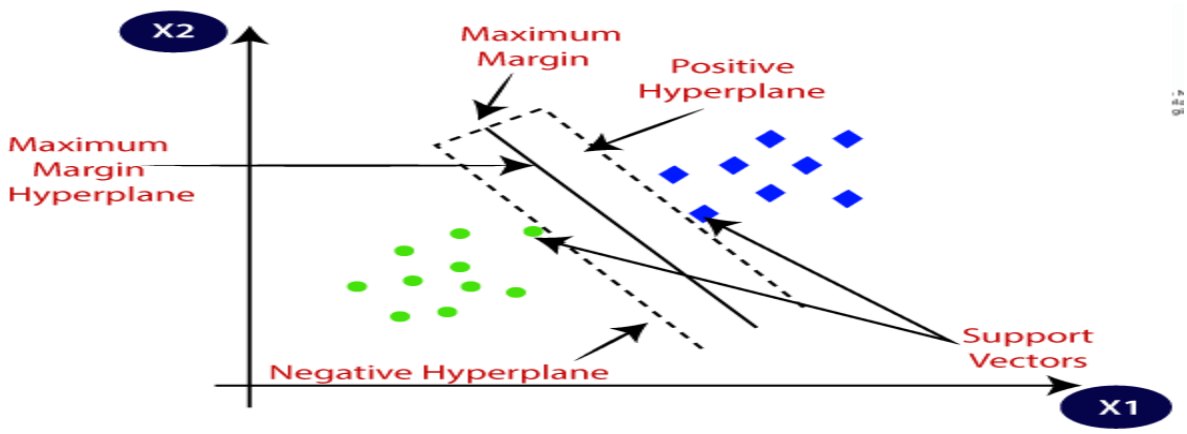
TOPIC -5: Support Vector Machine Algorithm:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

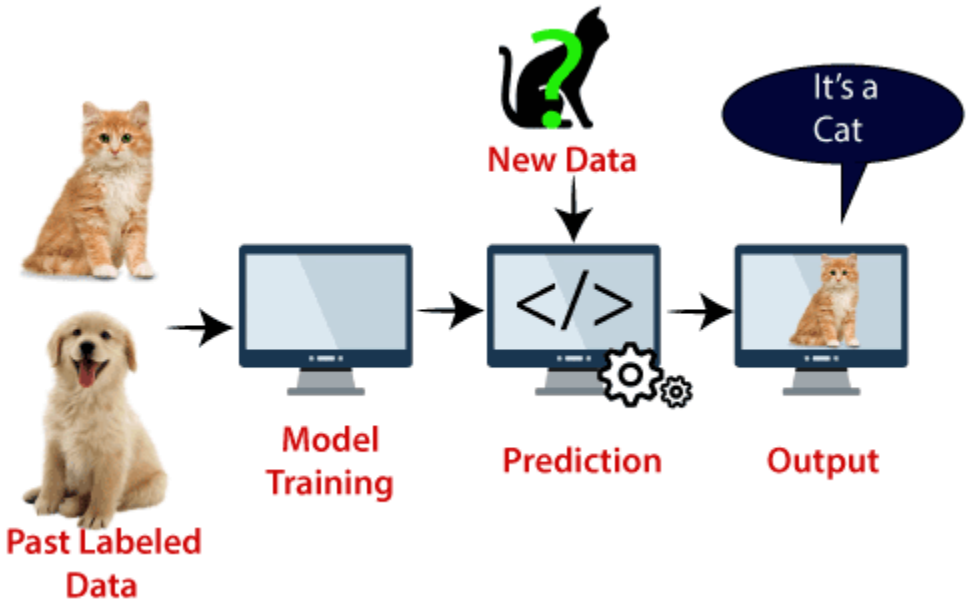
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Example: SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:Fullscreen



SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

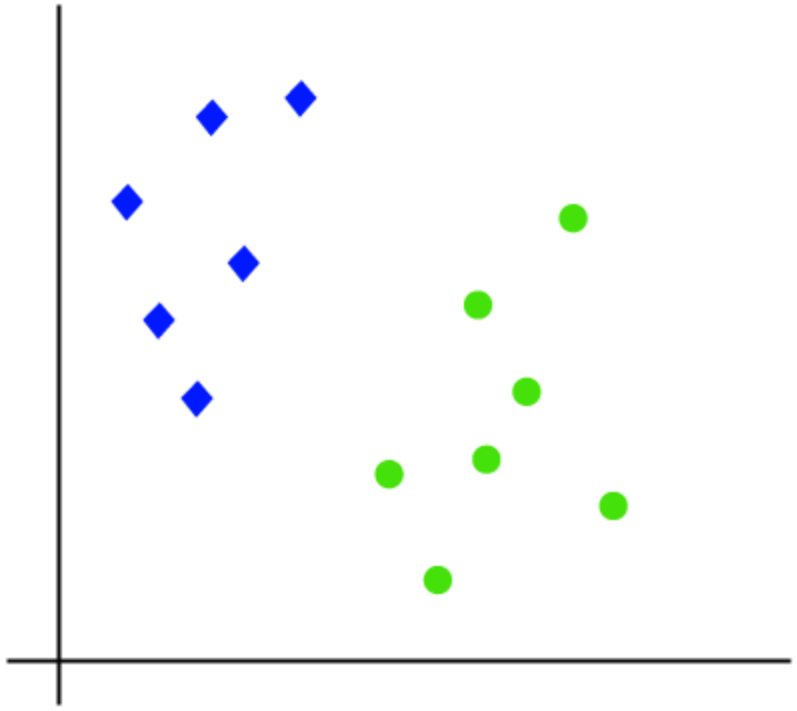
The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors: The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

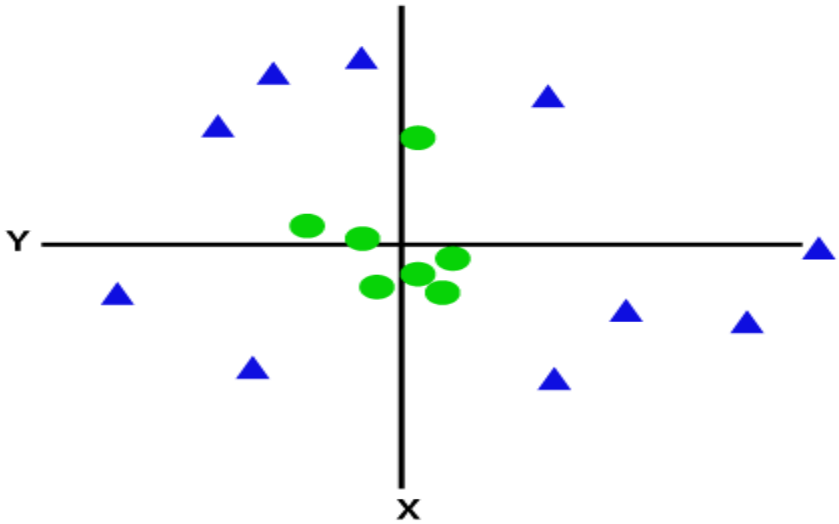
Linear SVM:

The working of the SVM algorithm can be understood by using an example. 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:



Non-Linear SVM:

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



TOPIC-6: Introduction to Support Vector Regression

It is one of the classic examples of supervised Machine learning technique. We could say it's one of the more powerful models which can be used in classification problems or assigning classes when the data is not linearly separable. I would give a classic kitchen example; I am sure most of us love chips? Of course, I do. I wanted to make homemade chips. I bought potatoes from the vegetable market and hit my kitchen. All I did was follow a YouTube video. I started slicing the potatoes in the hot oil, the result was a disaster, and I ended up getting chips which were dark brown/black.

What was wrong here? I had purchased potatoes, followed the procedure shown in the video and also maintained the right temperature of the oil, did I miss something? When I did my research by asking experts, I found that I did miss the trick, I had chosen potatoes with higher starch content instead of lower ones. All the potatoes looked the same for me, this is where the years and years of training comes into the picture, the experts had well-trained eyes on how to choose potatoes with lower starch. It has some specific features such as these potatoes would look fresh and will have some additional skin which could be peeled from our fingernails, and they look muddy.

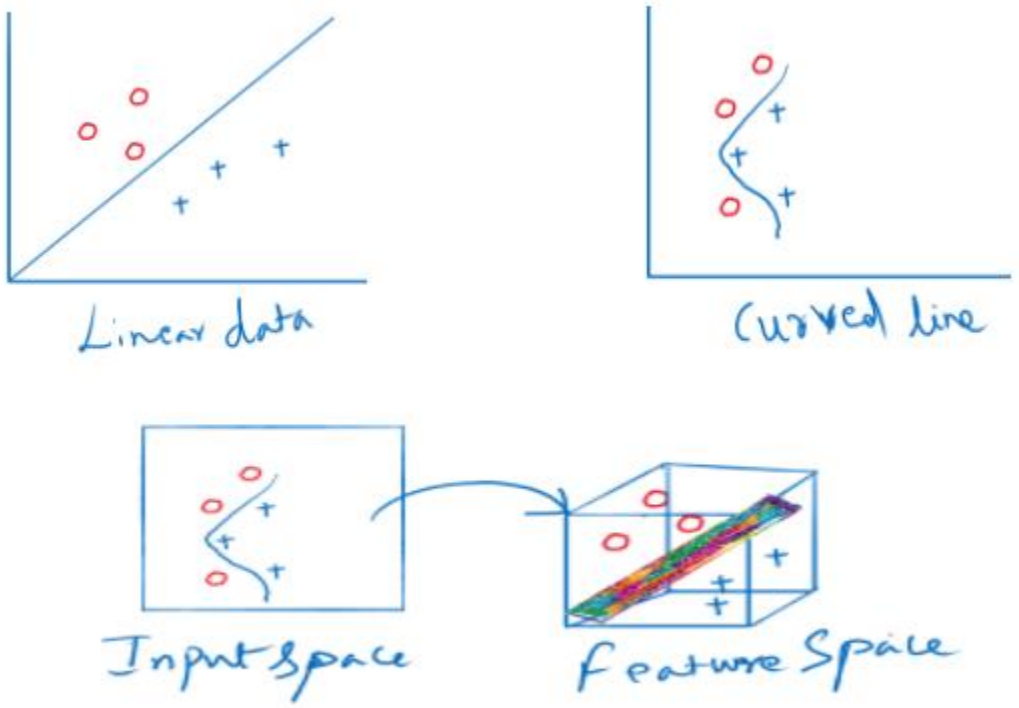
With the chips example, I was only trying to tell you about the nonlinear dataset. While many classifiers exist that can classify linearly separable data like logistic regression or linear regression, SVMs can handle highly non-linear data using an amazing technique called kernel trick. It implicitly maps the input vectors to higher dimensional (adds more dimensions) feature spaces by the transformation which rearranges the dataset in such a way that it is linearly solvable.

In short, a kernel is a function which places a low dimensional plane to a higher dimensional space where it can be segmented using a plane. In other words, it transforms linearly inseparable data to separable data by adding more dimensions to it.

There are three kernels which SVM uses the most

- **Linear kernel:** Dot product between two given observations

- **Polynomial kernel:** This allows curved lines in the input space
- **Radial Basis Function (RBF):** It creates complex regions within the feature space



In general, regression problems involve the task of deriving a mapping function which would approximate from input variables to a continuous output variable.

Support Vector Regression uses the same principle of Support Vector Machines. In other words, the approach of using SVMs to solve regression problems is called Support Vector Regression or SVR.

TOPIC-7: Naïve Bayes Classifier Algorithm:

ALGORITHM:

- 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 is it 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.Fullscreen

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

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