Unit 2 Support Vector Machine

OUTLINE:

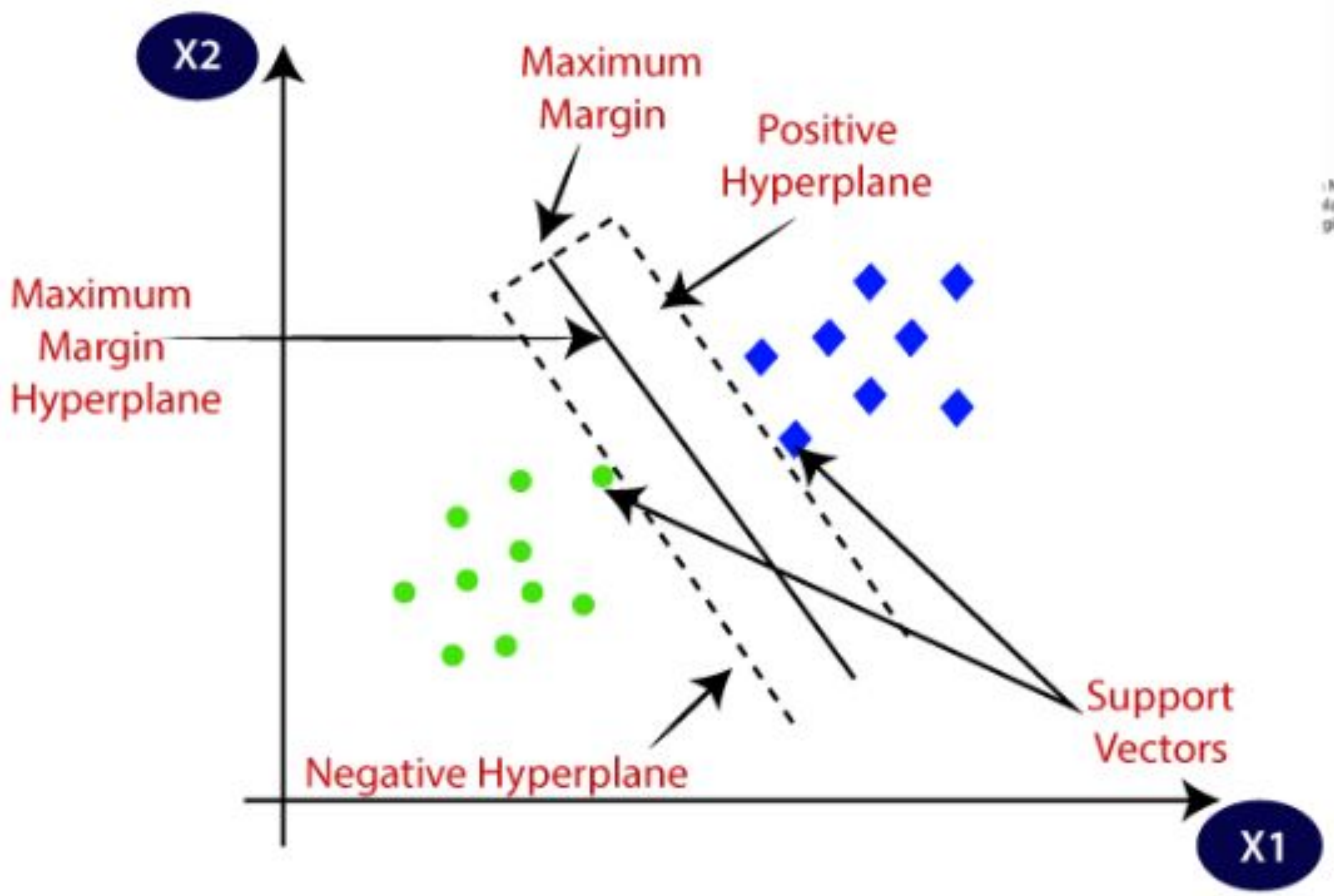
- ❖ Support Vector Machine

Support Vector Machine Algorithm

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

Support Vector Machine Algorithm



Support Vector Machine Algorithm

Types of SVM

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.

Support Vector Machine Algorithm

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.

Support Vectors:

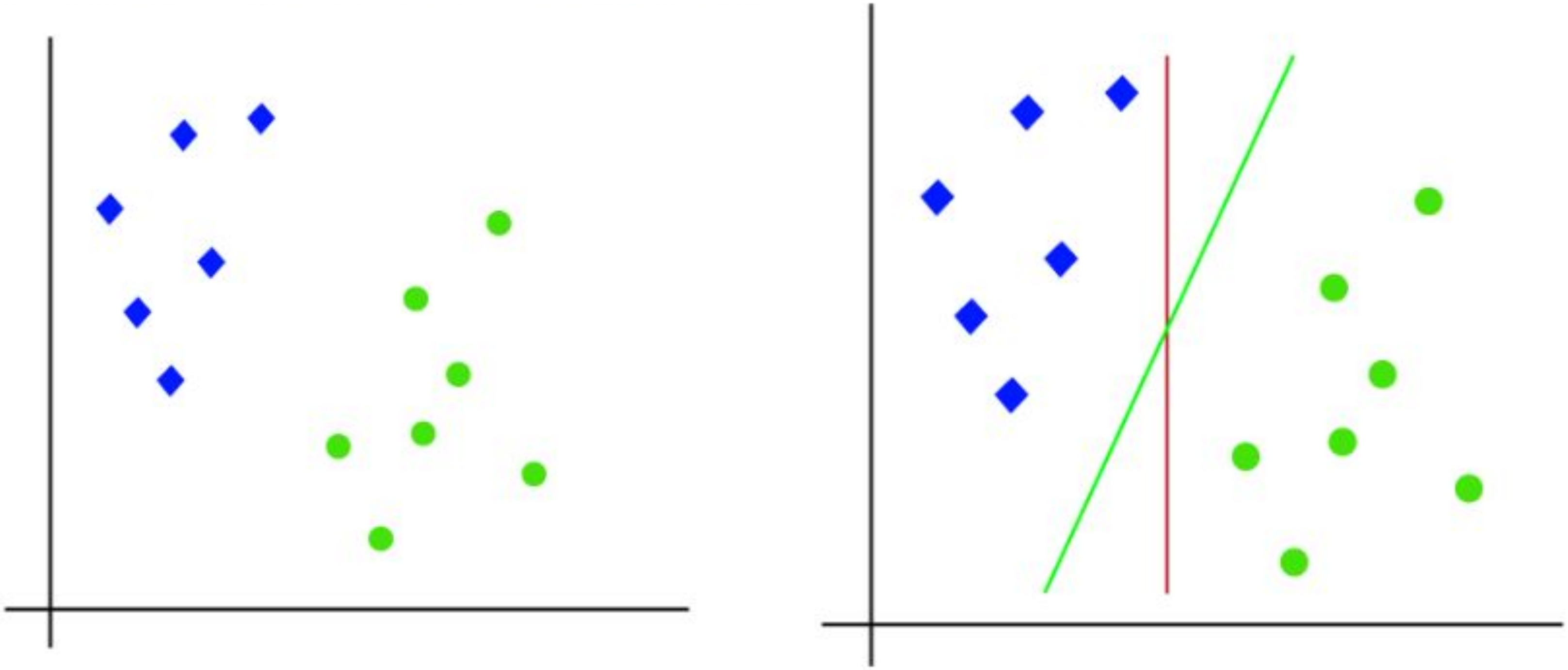
The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

Support Vector Machine Algorithm

How does SVM works?

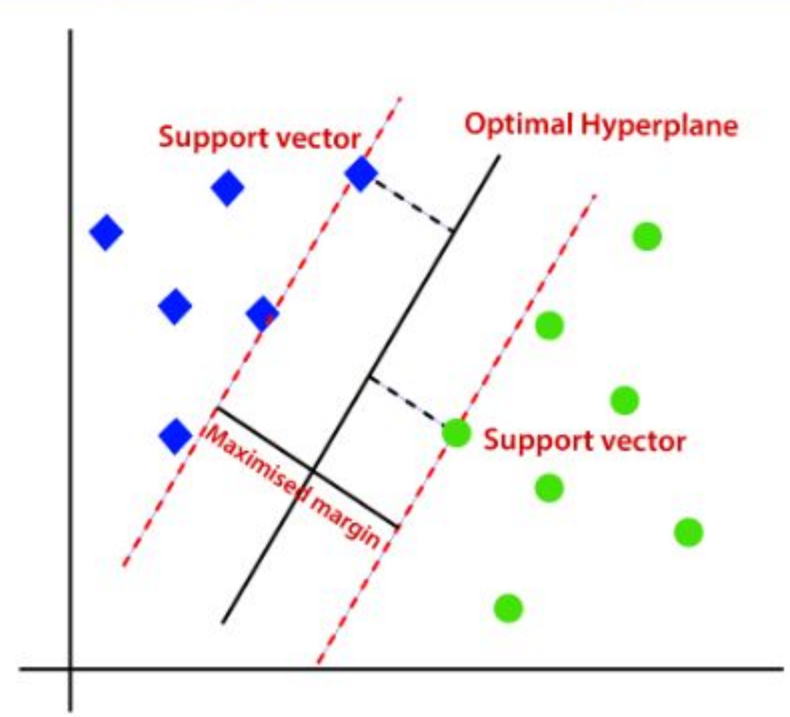
Linear SVM:

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:



Support Vector Machine Algorithm

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**.

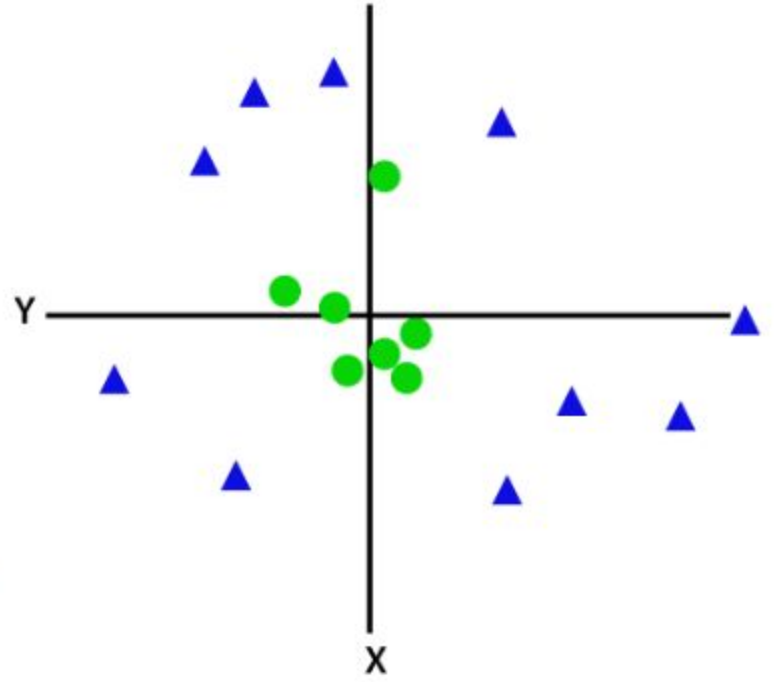


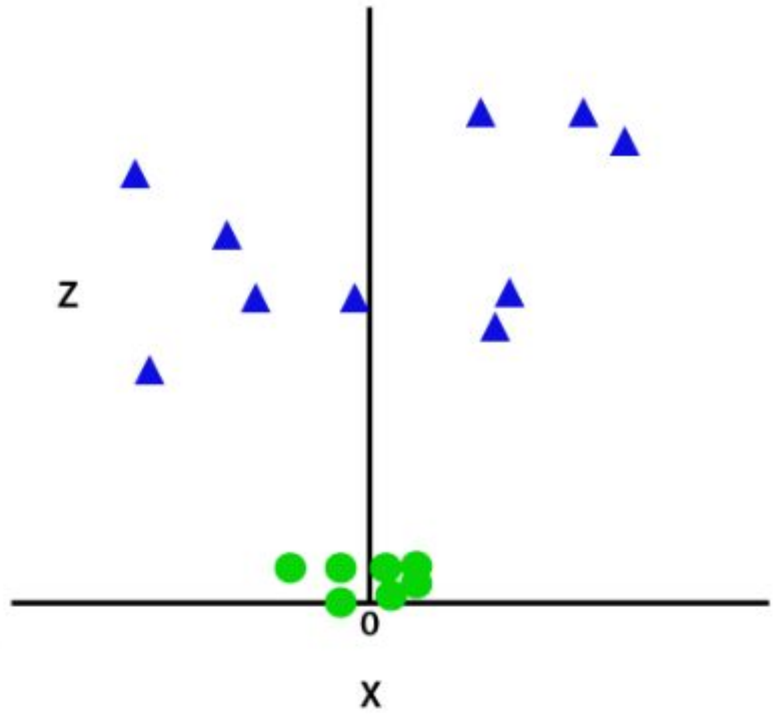
Support Vector Machine Algorithm

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:

o to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as: $z=x^2 +y^2$



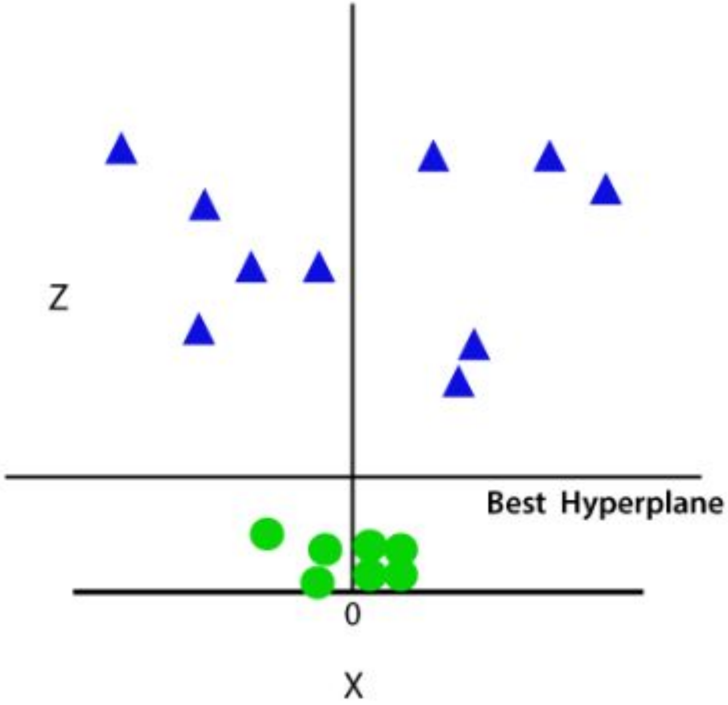


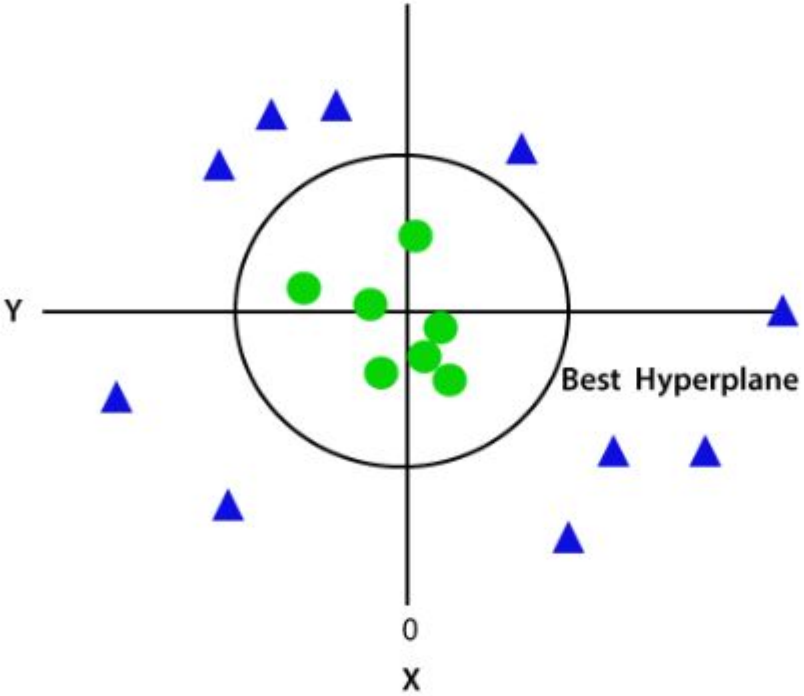
Support Vector Machine Algorithm

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:

o to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as: $z=x^2 +y^2$





Unit 2 SVM Kernel

OUTLINE:

- Support Vector Machine Kernel

Support Vector Machine Algorithm

SVM Kernel:

- In machine learning, kernel machines are a class of algorithms for pattern analysis, whose best known member is the support-vector machine (SVM). These methods involve using linear classifiers to solve nonlinear problems.

A set of techniques known as kernel methods are used in machine learning to address classification, regression, and other prediction issues. They are built around the idea of kernels, which are functions that gauge how similar two data points are to one another in a high-dimensional feature space.

Kernel methods' fundamental premise is used to convert the input data into a high-dimensional feature space, which makes it simpler to distinguish between classes or generate predictions. Kernel methods employ a kernel function to implicitly map the data into the feature space, as opposed to manually computing the feature space.

The most popular kind of kernel approach is the **Support Vector Machine (SVM),** a binary classifier that determines the best hyperplane that most effectively divides the two groups. In order to efficiently locate the ideal hyperplane, SVMs map the input into a higher-dimensional space using a kernel function.

Support Vector Machine Algorithm

Major Kernel Function in Support Vector Machine:

Linear Kernel:

A linear kernel is a type of kernel function used in machine learning, including in SVMs (Support Vector Machines). It is the simplest and most commonly used kernel function, and it defines the dot product between the input vectors in the original feature space.

The linear kernel can be defined as:

$K(x, y) = x .y$

Where x and y are the input feature vectors. The dot product of the input vectors is a measure of their similarity or distance in the original feature space.

When using a linear kernel in an SVM, the decision boundary is a linear hyperplane that separates the different classes in the feature space.

This linear boundary can be useful when the data is already separable by a linear decision boundary or when dealing with high-dimensional data, where the use of more complex kernel functions may lead to overfitting.

Support Vector Machine Algorithm

Gaussian (RBF) Kernel

The Gaussian kernel, also known as the radial basis function (RBF) kernel, is a popular kernel function used in machine learning, particularly in SVMs (Support Vector Machines). It is a nonlinear kernel function that maps the input data into a higher-dimensional feature space using a Gaussian function.

$$K(x, y) = exp(-gamma * ||x - y||^2)$$

Where x and y are the input feature vectors, gamma is a parameter that controls the width of the Gaussian function, and $||x - y||^2$ is the squared Euclidean distance between the input vectors.

When using a Gaussian kernel in an SVM, the decision boundary is a nonlinear hyper plane that can capture complex nonlinear relationships between the input features. The width of the Gaussian function, controlled by the gamma parameter, determines the degree of nonlinearity in the decision boundary.

One advantage of the Gaussian kernel is its ability to capture complex relationships in the data without the need for explicit feature engineering. However, the choice of the gamma parameter can be challenging, as a smaller value may result in under fitting, while a larger value may result in over fitting.

Support Vector Machine Algorithm

Laplace Kernel

The Laplacian kernel, also known as the Laplace kernel or the exponential kernel, is a type of kernel function used in machine learning, including in SVMs (Support Vector Machines). It is a non-parametric kernel that can be used to measure the similarity or distance between two input feature vectors.

K(x, y) = exp(-gamma * ||x - y||)

Where x and y are the input feature vectors, gamma is a parameter that controls the width of the Laplacian function, and ||x - y|| is the L1 norm or Manhattan distance between the input vectors.

When using a Laplacian kernel in an SVM, the decision boundary is a nonlinear hyperplane that can capture complex relationships between the input features. The width of the Laplacian function, controlled by the gamma parameter, determines the degree of nonlinearity in the decision boundary.

One advantage of the Laplacian kernel is its robustness to outliers, as it places less weight on large distances between the input vectors than the Gaussian kernel. However, like the Gaussian kernel, choosing the correct value of the gamma parameter can be challenging.

Unit 3 KNN Algorithm

OUTLINE:

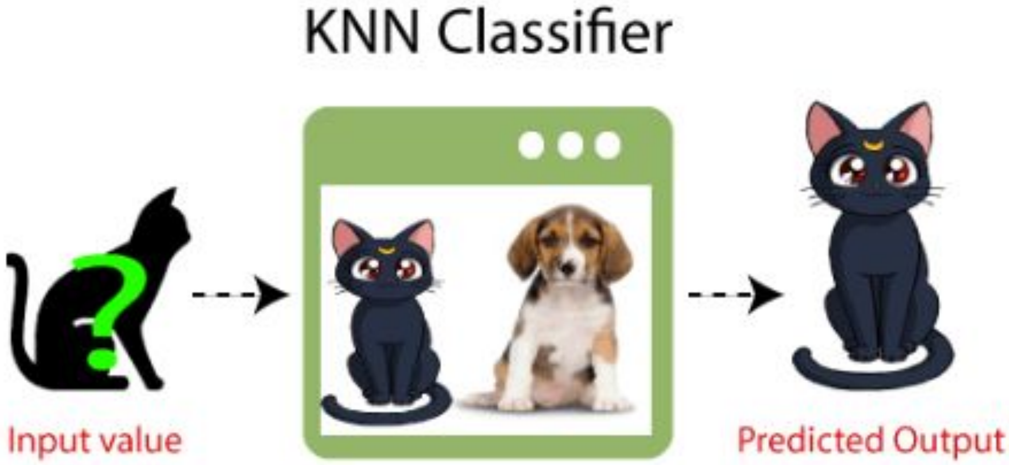
- ❖ KNN Algorithm

KNN Algorithm

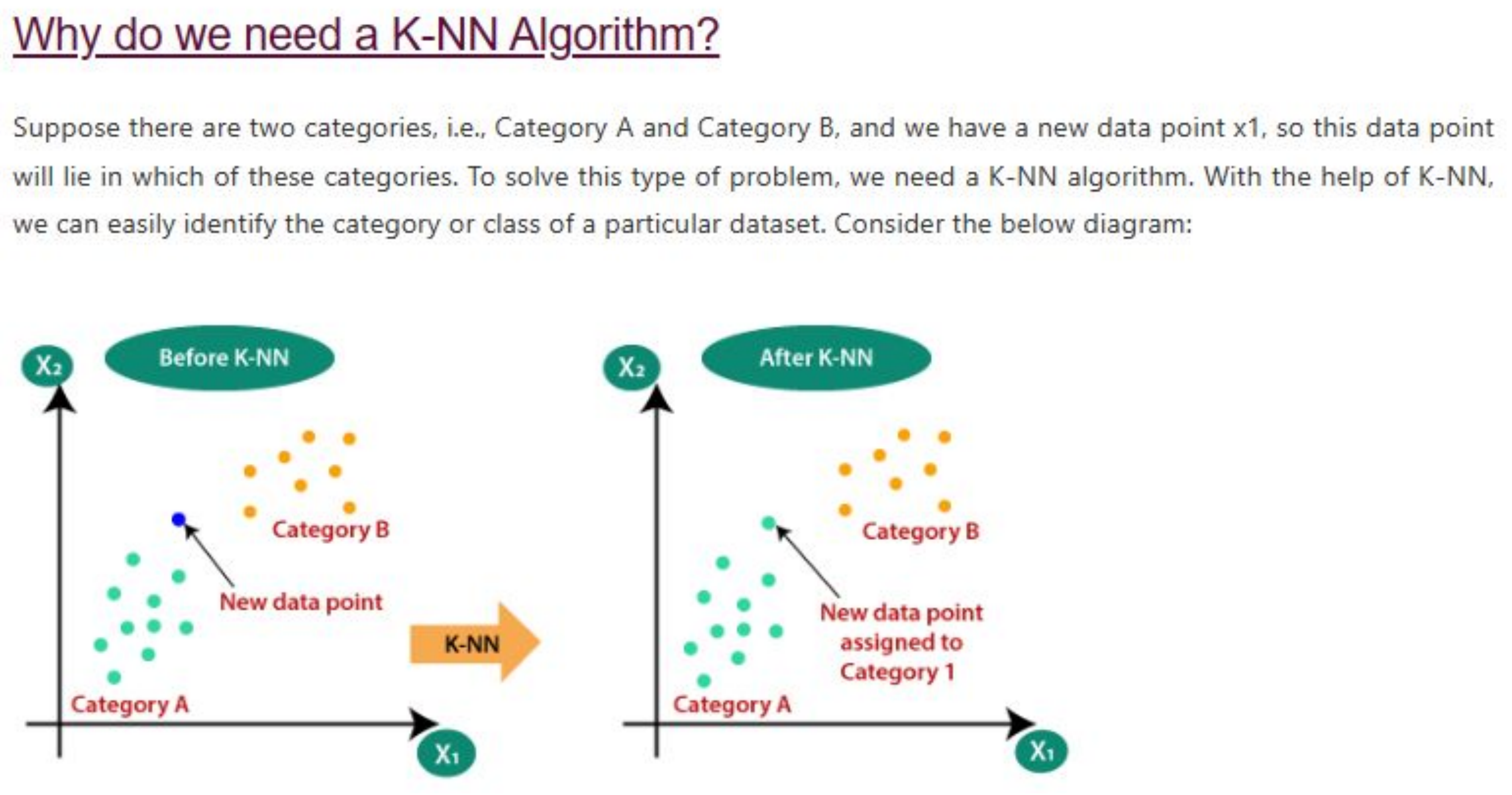
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm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

KNN Algorithm

- **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.



KNN Algorithm



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:

KNN Algorithm

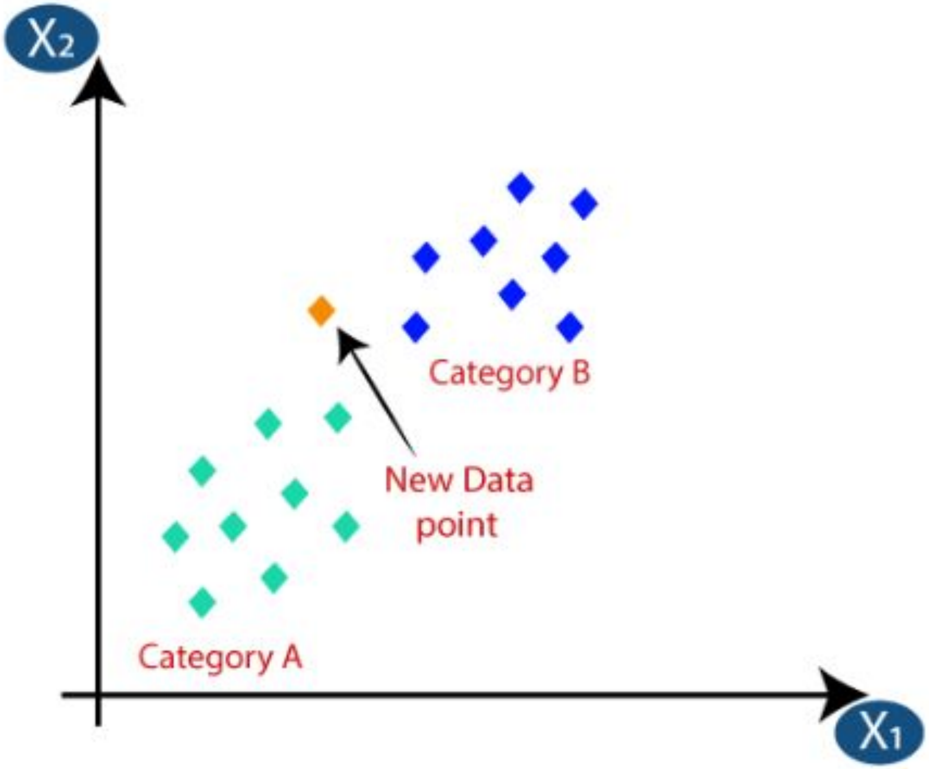
How does K-NN work?

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.

KNN Algorithm

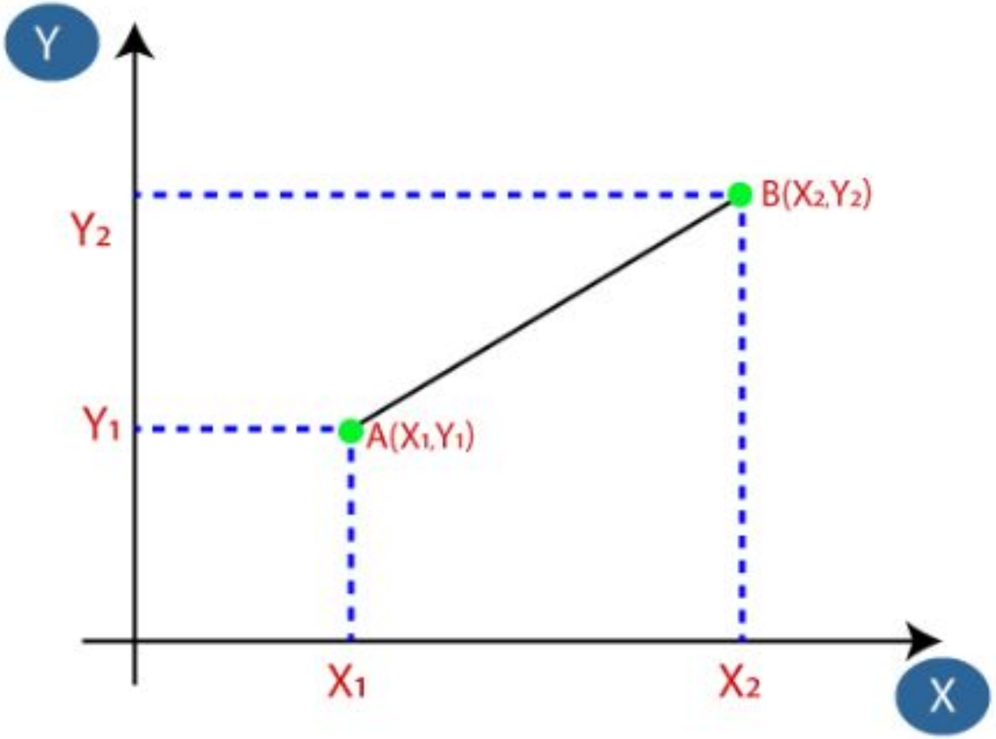
Suppose we have a new data point and we need to put it in the required category. Consider the below image:



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

KNN Algorithm

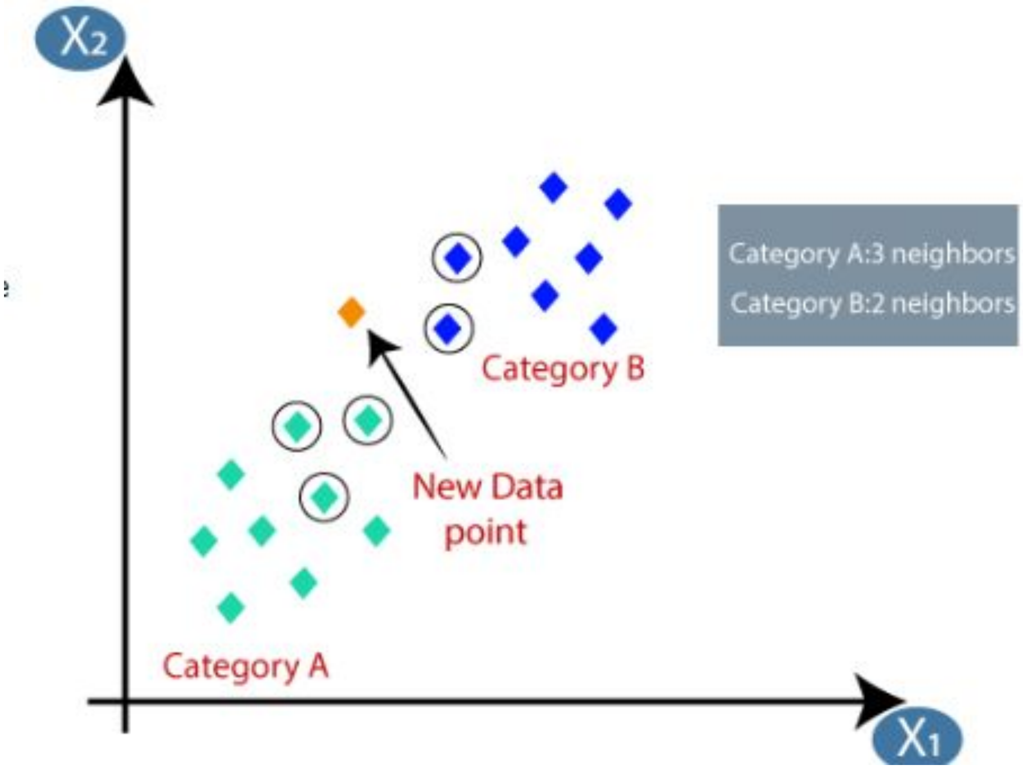
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



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

KNN Algorithm

- 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:



KNN Algorithm

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.

Unit 3 Multiclass Classification

OUTLINE:

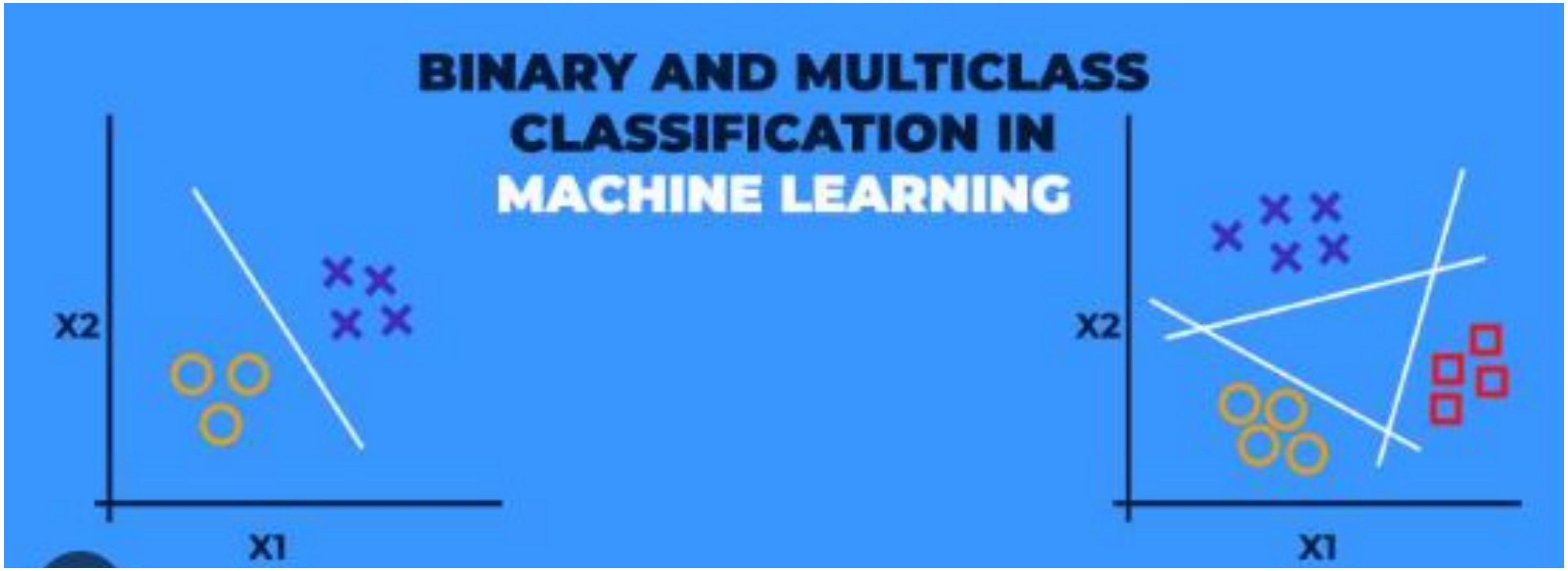
❖ Multiclass Classification

Multiclass Classification

In machine learning and statistical classification, multiclass classification or multinomial classification is the problem of classifying instances into one of three or more classes (classifying instances into one of two classes is called binary classification).

Which classifiers do we use in multiclass classification? When do we use them?
We use many algorithms such as Naïve Bayes, Decision trees, SVM, Random forest classifier, KNN, and logistic regression for classification. But we might learn about only a few of them here because our motive is to understand multiclass classification. So, using a few algorithms we will try to cover almost all the relevant concepts related to multiclass classification.

Multiclass Classification


BINARY AND MULTICLASS CLASSIFICATION IN MACHINE LEARNING
X2
X1
X2
X1