Support Vector Machine (SVM)

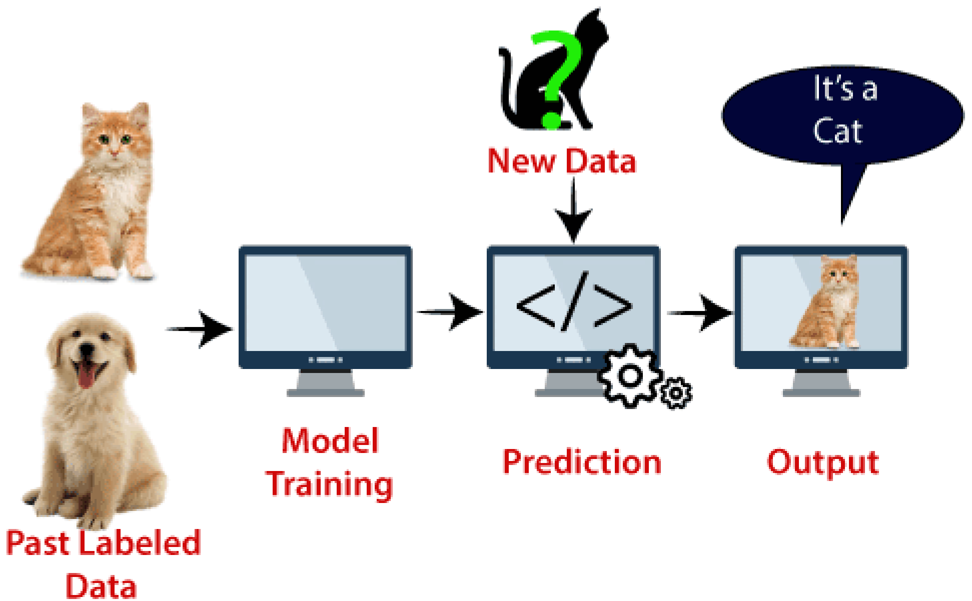
By
Sapna Yadav

Support vector machines

- Support Vector Machine (SVM) is a supervised machine learning algorithm capable of performing **classification, regression** and even **outlier detection**.
- The Support Vector Machine, created by **Vladimir Vapnik** in the 60s, still one of most popular machine learning classifiers
- used for classification and regression problems as **support vector classification** (SVC) and support **vector regression** (SVR).

Support vector machines

Example: SVM can be understood with the. 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:



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 machines

- SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees.
- It is known for its **kernel trick** to handle **nonlinear input spaces.**
- It is used in a variety of applications such as **face detection, intrusion detection, classification of emails, news articles and web pages, classification of genes, and handwriting recognition.**
- Classification of satellite data using supervised SVM.
- SVMs are helpful in text and hypertext categorization
- Classification of images can also be performed using SVMs

Support vector machines

- The classifier separates data points using a hyperplane with the **largest amount of margin**.
- That's why an SVM classifier is also known as a **discriminative classifier**. SVM finds an optimal hyperplane which helps in classifying new data points.
- It can easily handle **multiple continuous** and **categorical variables.**
- SVM constructs a **hyperplane in multidimensional space** to separate different classes.
- SVM generates optimal hyperplane in an **iterative manner,** which is used to minimize an error.

Hyperplane

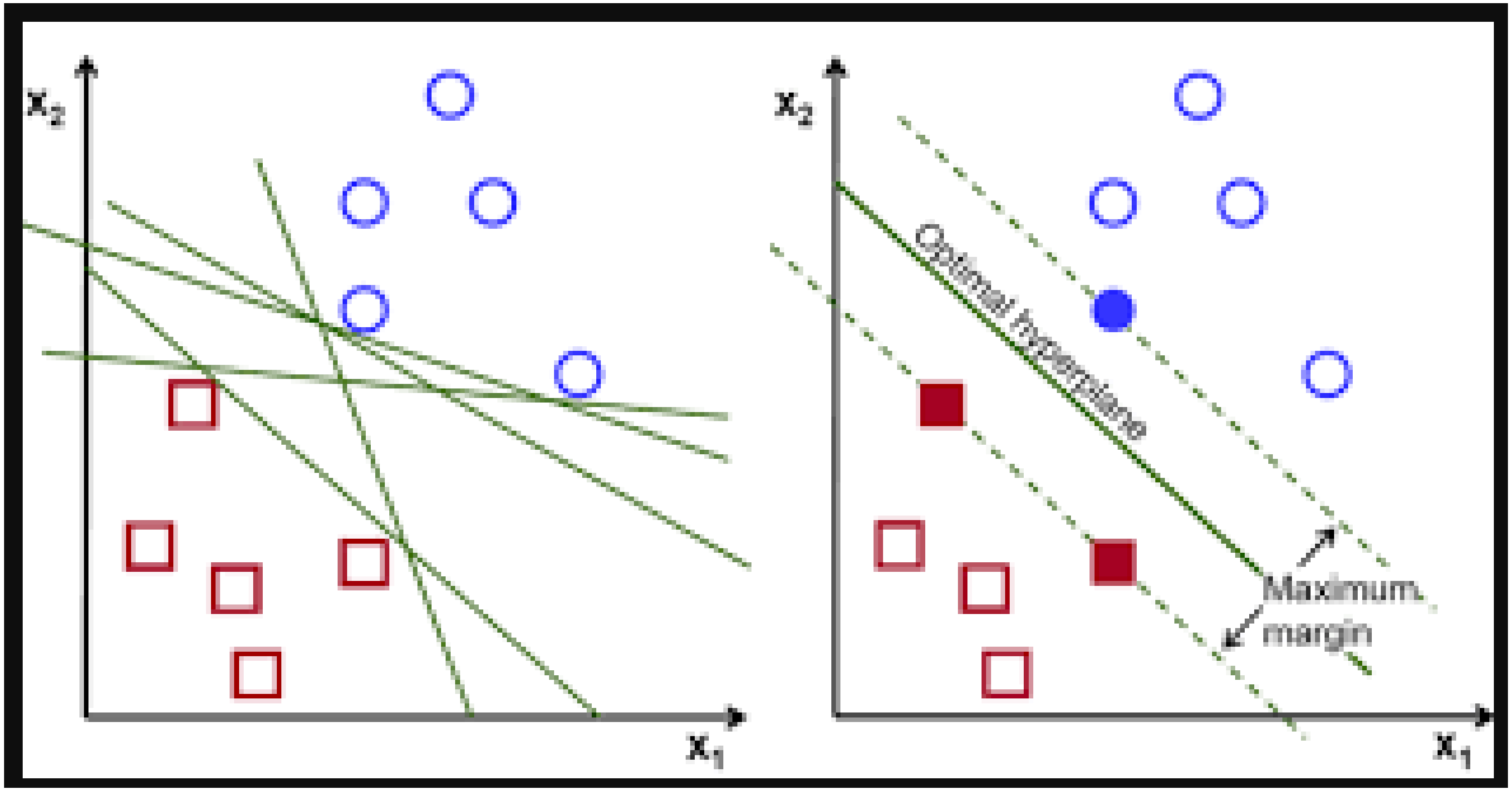
- A hyperplane is a decision plane which separates between a set of objects having different class memberships.
- There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. **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** then hyperplane will be a **straight line**. And if there are **3 features**, then hyperplane will be a **2-dimension plane.**

Support vector machines

Margin

- A margin is a gap between the two lines on the closest class points. This is calculated as the **perpendicular distance from the line to support vectors or closest points**.
- If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.
- We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support vector machines

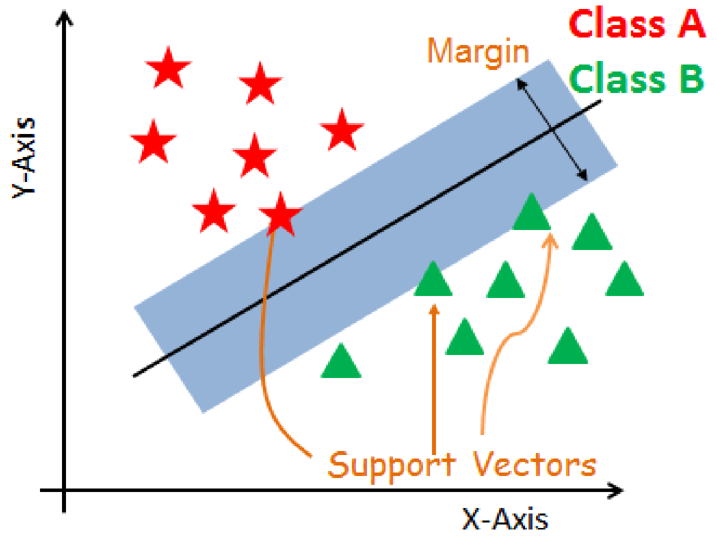


Support Vector machines

- The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

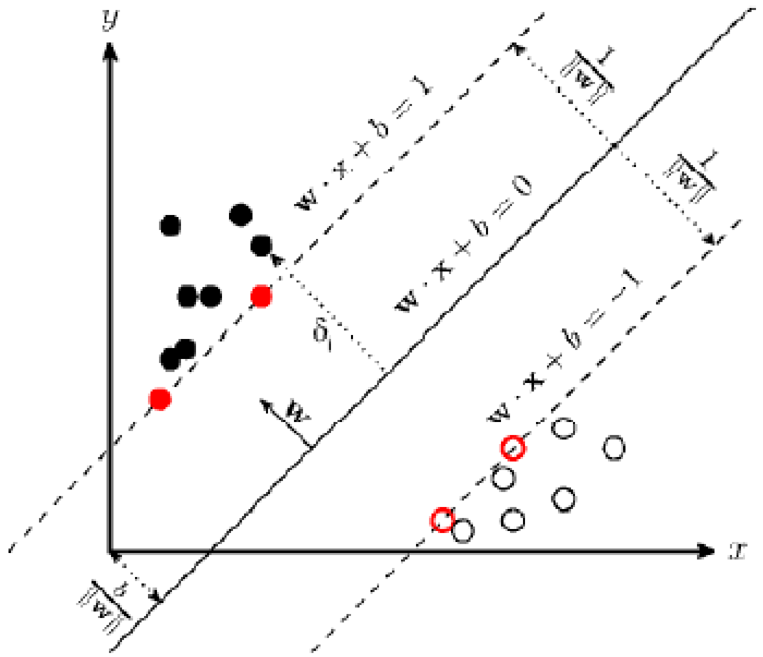
Support Vectors

Support vectors are the data points, which are closest to the hyperplane.



These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

So a hyperplane equation is obtained in the linear SVM for positive class:
w. (xi) + b ≤ + 1
Whereas for the negative class hyperplane equation in the linear SVM are:
w. (xi) + b ≥ - 1
Information:
w = weight (weight vector)
x = matrix input value (feature)
b = bias



y
x
$\mathbf{w}\cdot x + b = 1$
$\mathbf{w}\cdot \mathbf{x} + b = 0$
$\mathbf{w}\cdot \mathbf{x} + b = -1$
$\frac{1}{\|\mathbf{w}\|}$
$\frac{1}{\|\mathbf{w}\|}$
$\frac{b}{\|\mathbf{w}\|}$
δ_i
w

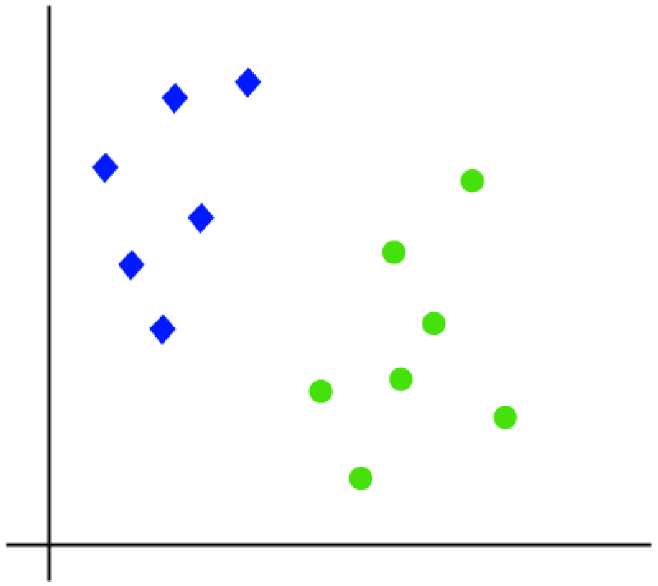
To calculate the largest margin value, it is done by optimizing the distance value between the hyperplane and the closest point in each class.

Quadratic Programming (QP) is used as a formula to find the minimal point of an equation with equation constraints:
τ (w) = 1/2 ǁwǁ ^ 2

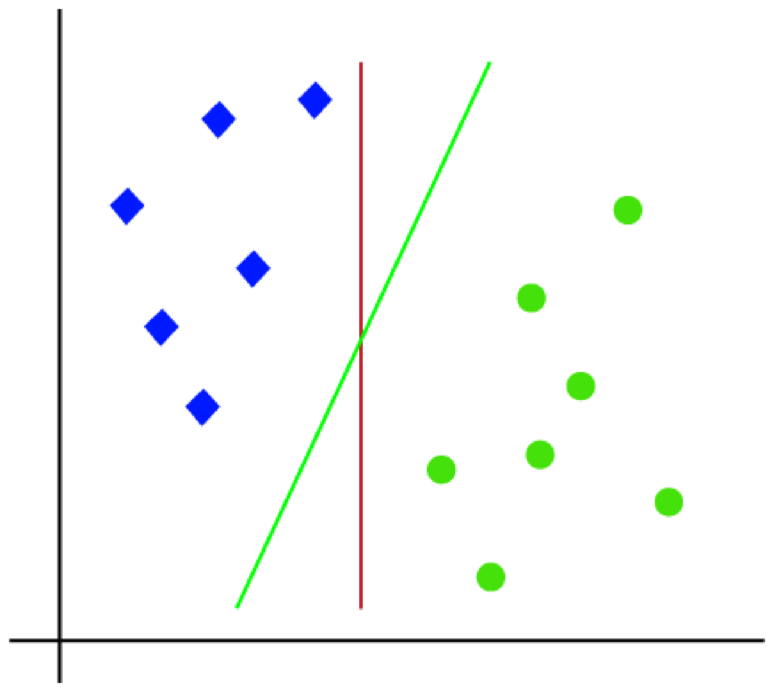
How does SVM works?

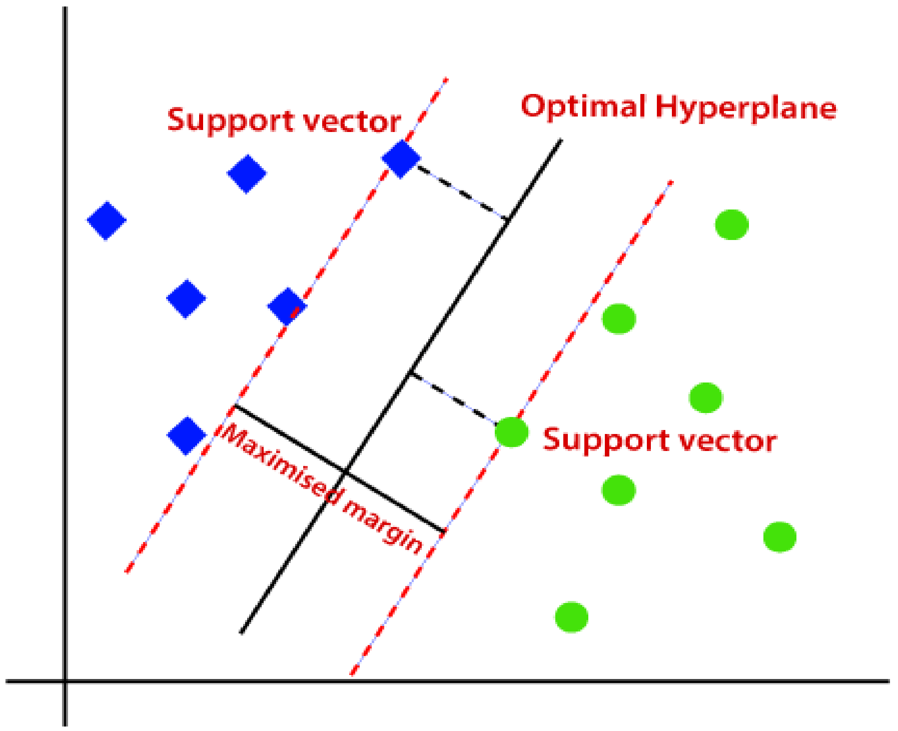
Linear SVM

- **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:**



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.

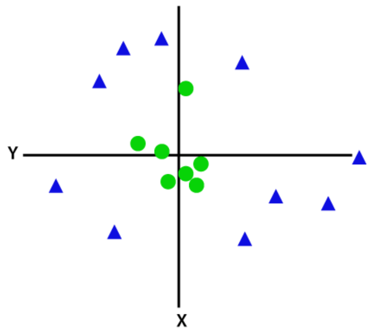




Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called **support vectors**. The distance between the vectors and the hyperplane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

How does Non-linear SVM works?

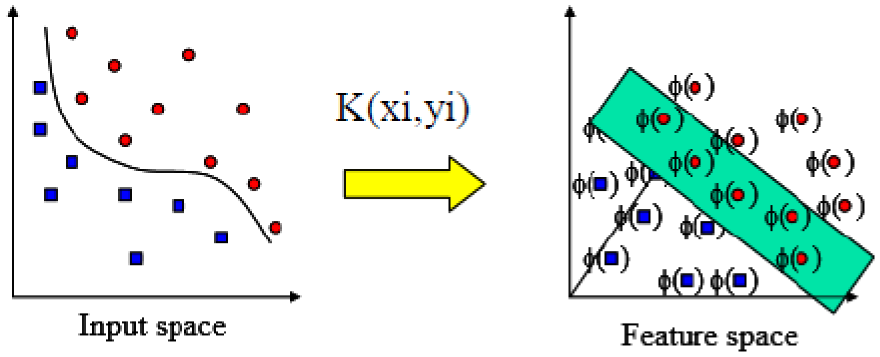
- **If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line.**



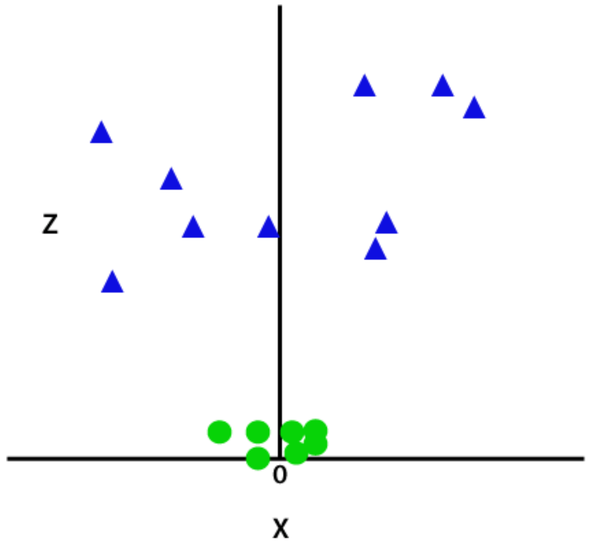
Non-linear classification is carried out using the kernel concept. The kernel concept in the non-linear case plays a role in determining the classification limits used as a model.

So 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:

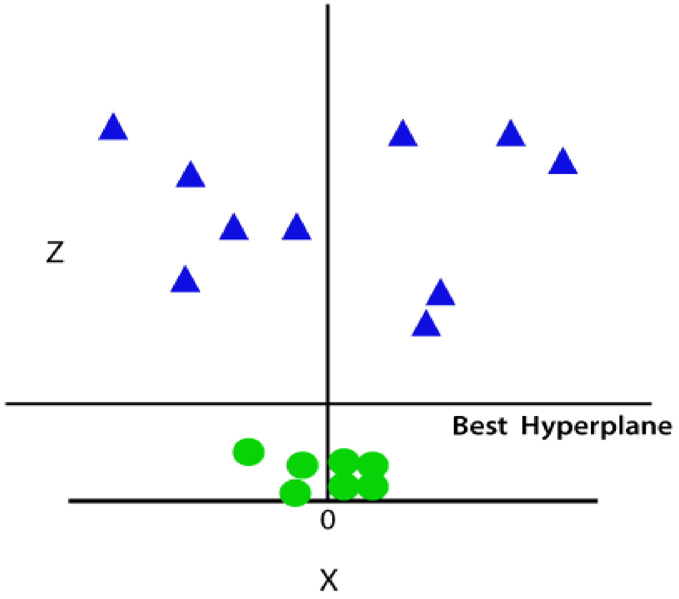
$z=x^2+y^2$



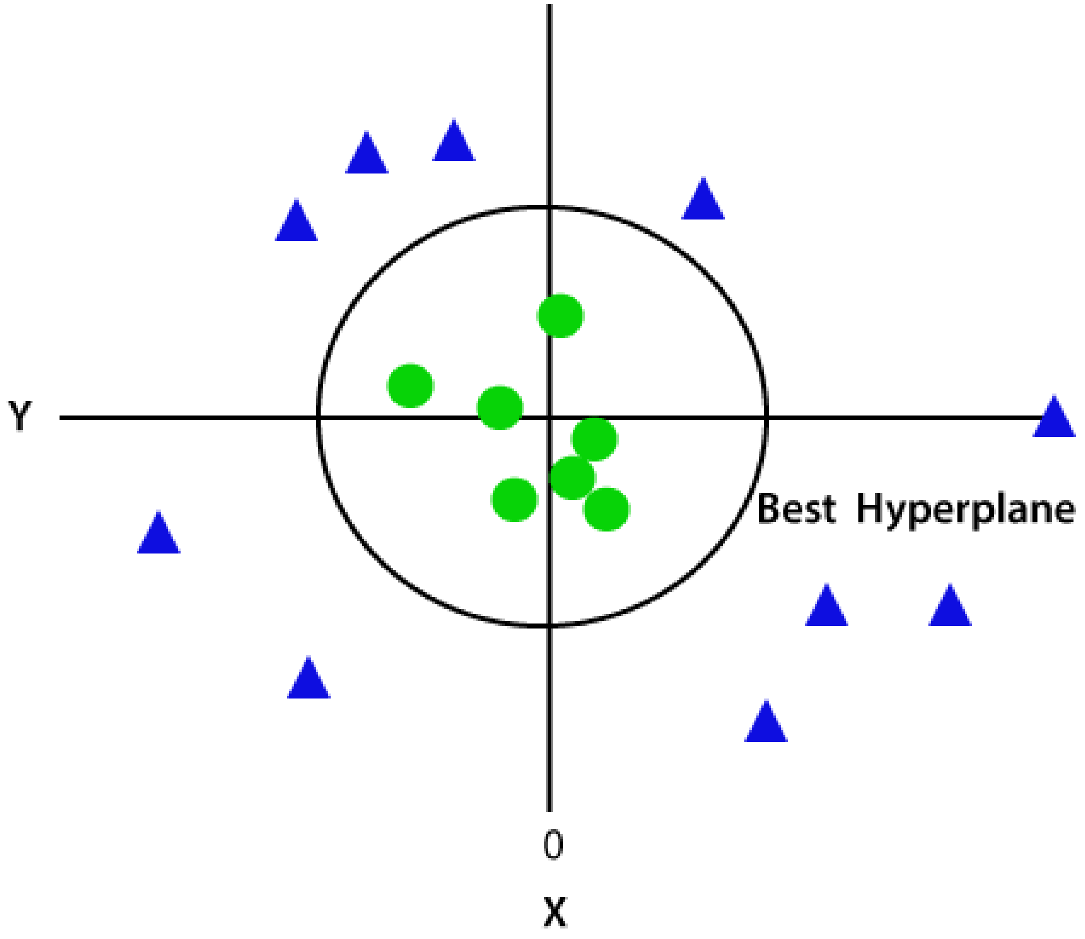
- By adding the third dimension, the sample space will become



So now, SVM will divide the datasets into classes in the following way



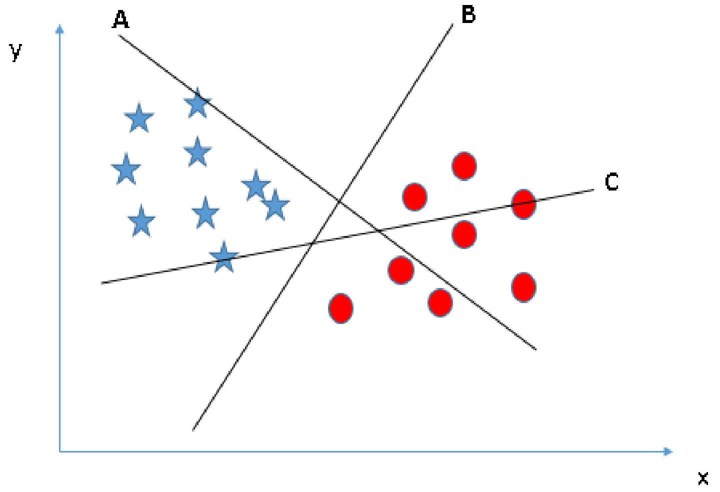
Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as



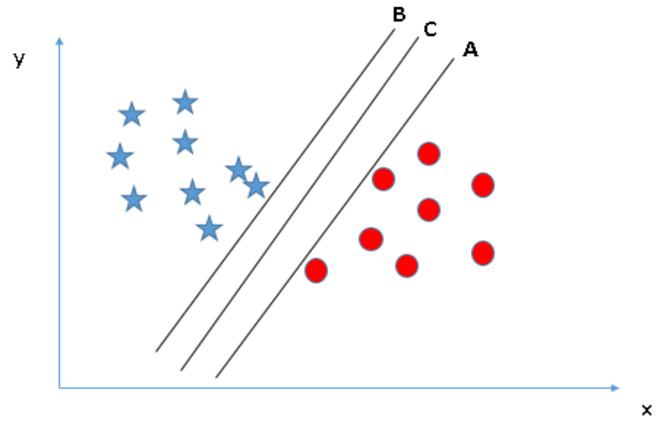
Identify the right hyper-plane

- **Identify the right hyper-plane (Scenario-1):** Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.

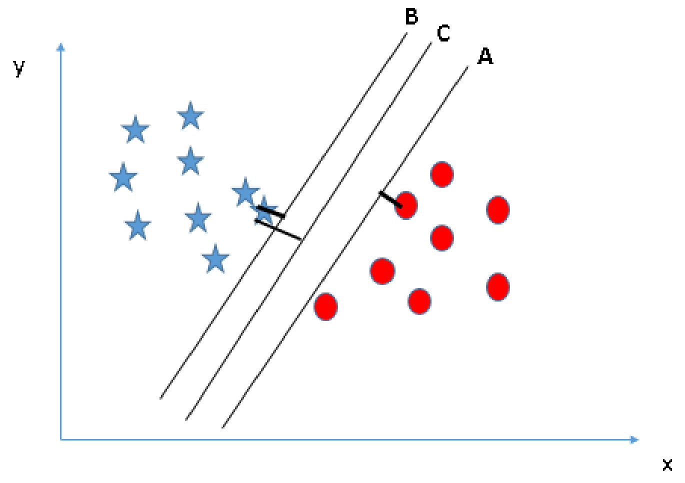
- **thumb rule to identify the right hyper-plane**: "Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.



Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?



Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**

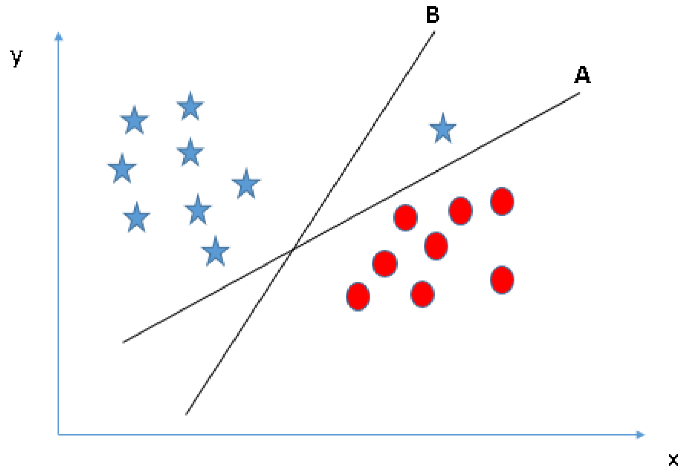


margin for hyper-plane C is high as compared to both A and B.

Support vector machines

Identify the right hyper-plane (Scenario-3):

you may have selected the hyper-plane **B** as it has higher margin compared to **A.**



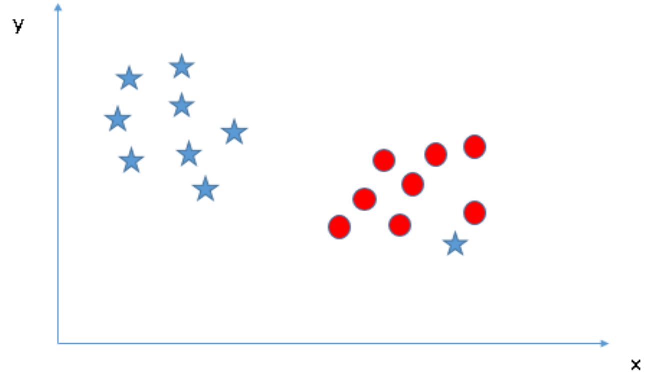
But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A.**

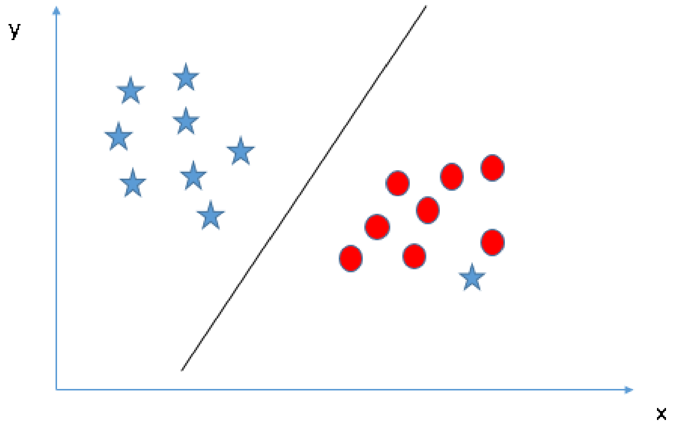
Support vector machines

Can we classify two classes (Scenario-4)?

unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.

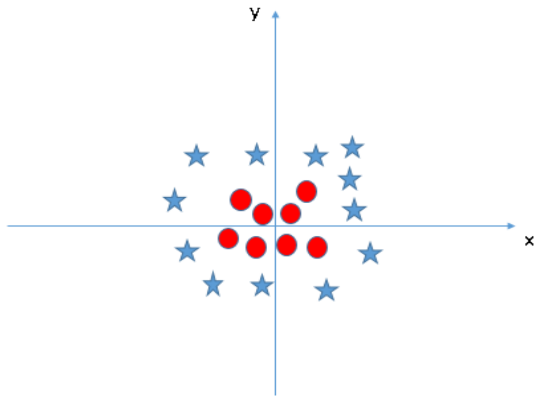
- **one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.**

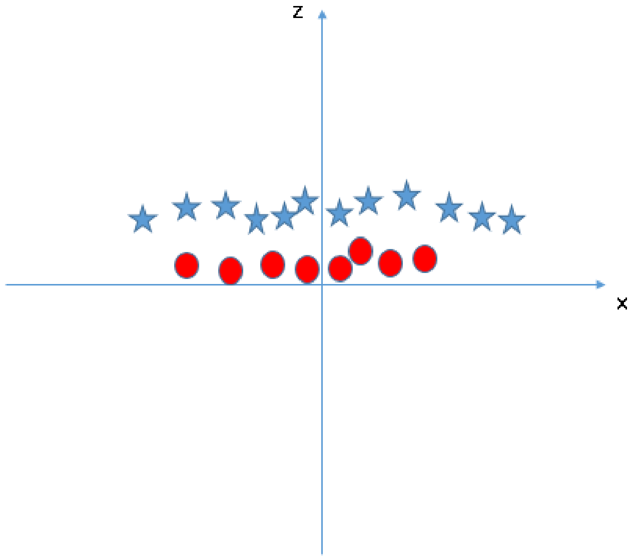




Find the hyper-plane to segregate to classes (Scenario-5): In the scenario below, we can't have linear hyper-plane between the two classes, **so how does SVM classify these two classes**? Till now, we have only looked at the linear hyper-plane.

- SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:

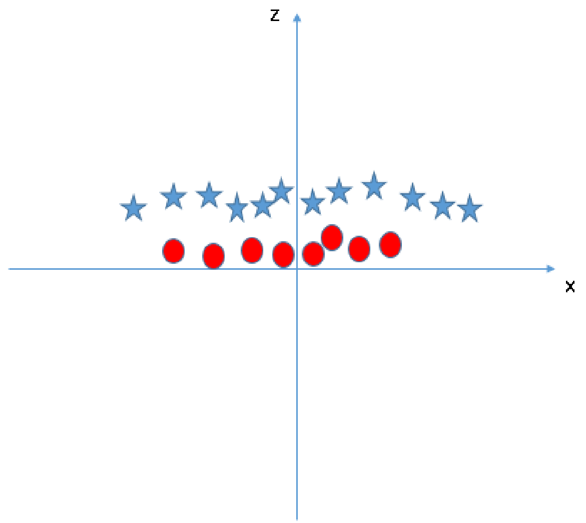




Support vector machines

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z ,away from the origin result to higher value of z.
- In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. **But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane.**

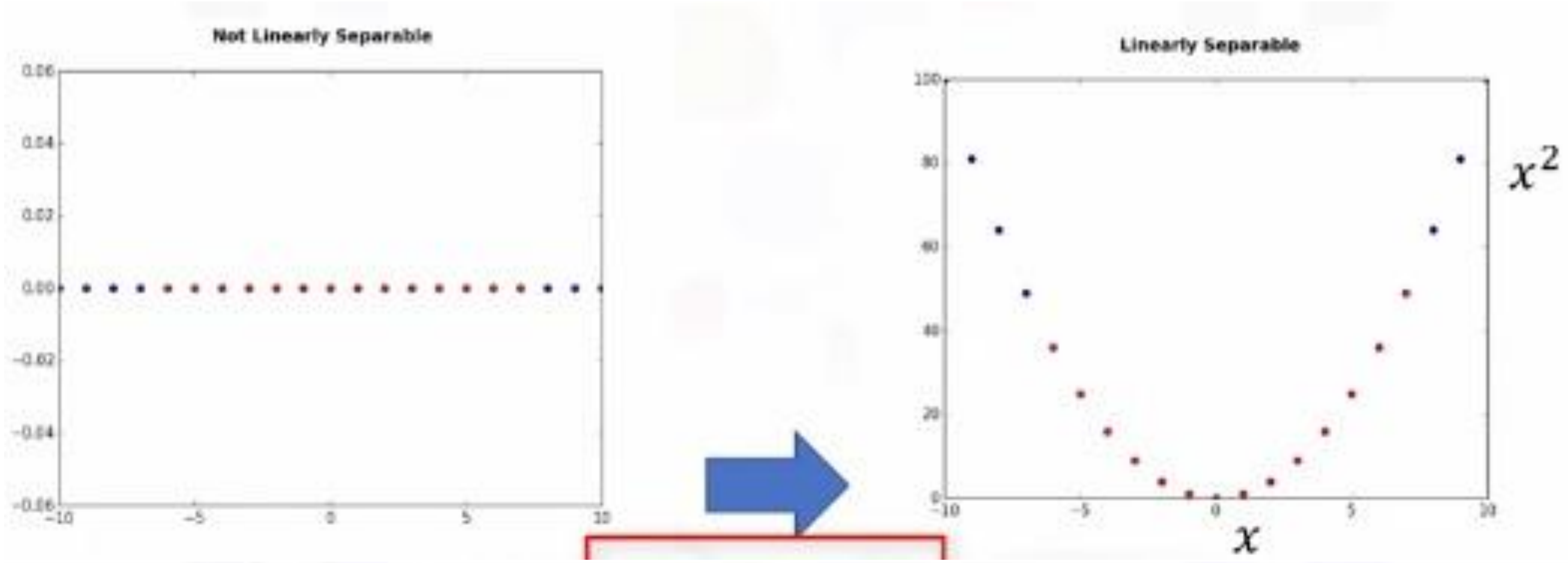
z=x^2+y^2

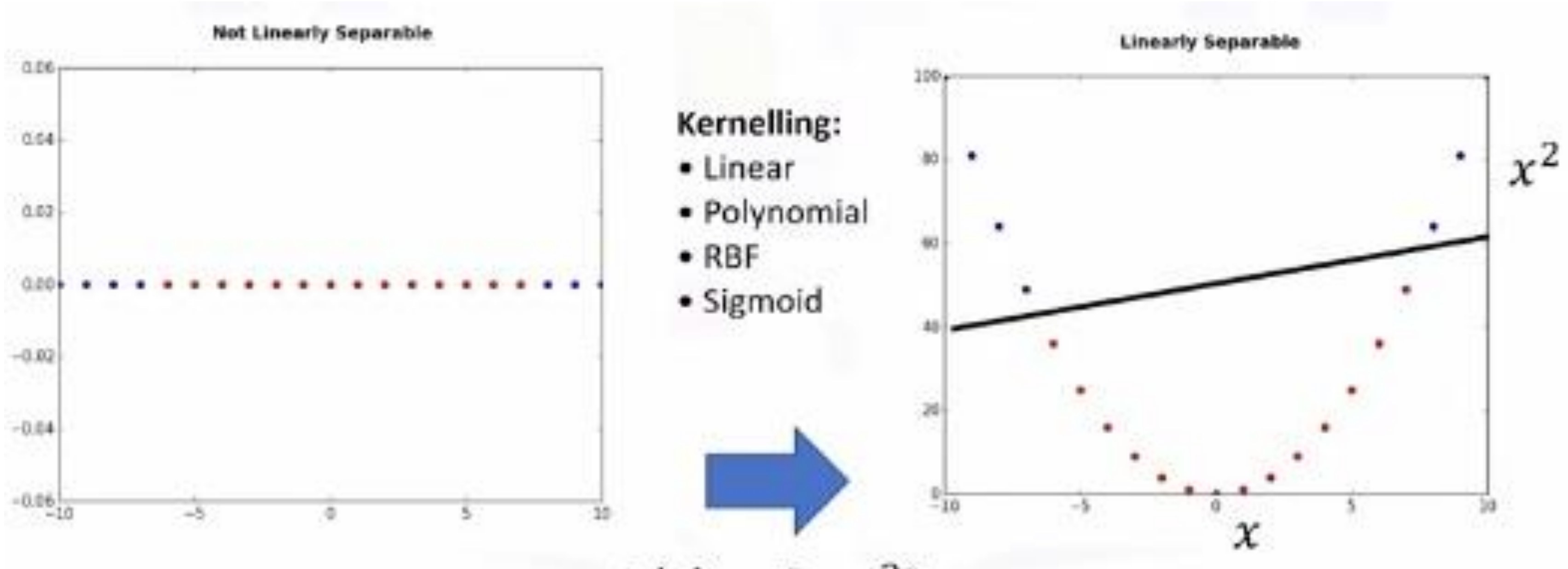


SVM Kernels

- The SVM algorithm has a technique called the **kernel trick**.
- A kernel transforms an input data space into the required form. SVM uses a technique called the kernel trick.
- Here, the kernel takes a low-dimensional input space and transforms it into a higher dimensional space.
- converts nonseparable problem to separable problems by adding more dimension to it.
- It is most useful in non-linear separation problem.
- Kernel trick helps you to build a more accurate classifier.
- Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

SVM Kernels





Kernelling:
- Linear
- Polynomial
- RBF
- Sigmoid

$$\phi(x) = [x, x^2]$$

- Different SVM algorithms use different types of kernel functions. These functions can be different types.
- For example ***linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid*.**
- Introduce Kernel functions for sequence data, graphs, text, images, as well as vectors.
- The most used type of kernel function is **RBF. Because it has localized and finite response along the entire x-axis**.
- **The kernel functions return the inner product between two points in a suitable feature space.**
- Thus by defining a notion of similarity, with little computational cost even in very high-dimensional spaces.

SVM Kernels

- **Linear Kernel** A linear kernel can be used as normal dot product any two given observations. The product between two vectors is the sum of the multiplication of each pair of input values.

K(x, xi) = sum(x * xi)

- **Polynomial Kernel** A polynomial kernel is a more generalized form of the linear kernel. The polynomial kernel can distinguish curved or nonlinear input space.

K(x,xi) = 1 + sum(x * xi)^d

Where d is the degree of the polynomial. d=1 is similar to the linear transformation. The degree needs to be manually specified in the learning algorithm.

SVM Kernels

- **Radial Basis Function Kernel** The Radial basis function kernel is a popular kernel function commonly used in support vector machine classification. RBF can map an input space in infinite dimensional space.

K(x,xi) = exp(-gamma * sum((x – xi^2))

$$K(x, x') = e^{-\gamma||x-x'||^2}$$

Here gamma is a parameter, which ranges from 0 to 1. A higher value of gamma will perfectly fit the training dataset, which causes over-fitting. Gamma=0.1 is considered to be a good default value. The value of gamma needs to be manually specified in the learning algorithm.

SVM Kernels

Gamma:

- A lower value of Gamma will loosely fit the training dataset, whereas a higher value of gamma will exactly fit the training dataset, which causes over-fitting.
- In other words, you can say a low value of gamma considers only nearby points in calculating the separation line, while the high value of gamma considers all the data points in the calculation of the separation line.

SVM Kernels-Tuning Hyper parameter C

- **Regularization**: Regularization parameter in python's Scikit-learn C parameter used to maintain regularization.
- Here C is the penalty parameter, which represents misclassification or error term.
- The misclassification or error term tells the SVM optimization how much error is bearable.
- This is how you can control the trade-off between decision boundary and misclassification term.
- A smaller value of C creates a small-margin hyperplane and a larger value of C creates a larger-margin hyperplane.

Support Vector Regression

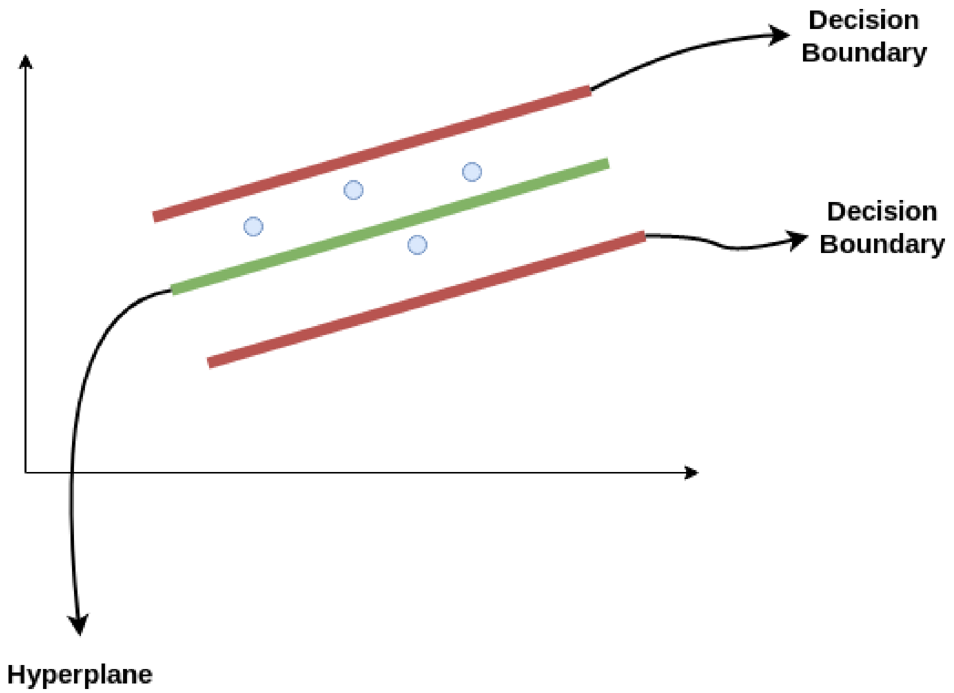
- Support Vector Regression is similar to Linear Regression in that the equation of the line is y= wx+b In SVR, this straight line is referred to as ***hyperplane***. The data points on either side of the hyperplane that are closest to the hyperplane are called ***Support Vectors*** which is used to plot the boundary line.

- Support Vector Regression (SVR) uses the same principle as SVM, but for regression problems. Let's spend a few minutes understanding the idea behind SVR.

- For a non-linear regression, the kernel function transforms the data to a higher dimensional and performs the linear separation. Here we will use the ***rbf*** kernel.

Support Vector Regression

The problem of regression is to find a function that approximates mapping from an input domain to real numbers on the basis of a training sample. So let’s now dive deep and understand how SVR works actually.



Consider these two red lines as the decision boundary and the green line as the hyperplane. **Our objective, when we are moving on with SVR, is to basically consider the points that are within the decision boundary line.** Our best fit line is the hyperplane that has a maximum number of points.

Support Vector Regression

The first thing that we'll understand is what is the decision boundary (the danger red line above!). Consider these lines as being at any distance, say 'a', from the hyperplane. So, these are the lines that we draw at distance '+a' and '-a' from the hyperplane. This 'a' in the text is basically referred to as epsilon.

Assuming that the equation of the hyperplane is as follows:

$Y = w*x + b$ (Equation of hyperplane)

Then the equations of decision boundary become:

$w*x + b = +a$

$w*x + b = -a$

Thus, any hyperplane that satisfies our SVR should satisfy:

$-a < Y - (w*x + b) < +a$

Support Vector Regression

Our main aim here is to decide a decision boundary at 'a' distance from the original hyperplane such that data points closest to the hyperplane or the support vectors are within that boundary line.

Hence, we are going to take only those points that are within the decision boundary and have the least error rate, or are within the Margin of Tolerance. This gives us a better fitting model.

Advantages & Disadvantages

Advantages

- SVM Classifiers offer **good accuracy** and perform **faster prediction** compared to others. They also use **less memory** because they **use a subset of training points** in the decision phase. SVM works well with a clear margin of separation and with high dimensional space.
- Training of the model is relatively easy
- The model scales relatively well to high dimensional data
- SVM is a useful alternative to neural networks
- Trade-off amongst classifier complexity and error can be controlled explicitly
- It is useful for both Linearly Separable and Non-linearly Separable data
- Assured Optimality: The solution is guaranteed to be the global minimum due to the nature of Convex Optimization

Advantages & Disadvantages

Disadvantages

- SVM is **not suitable for large datasets** as it takes more time in training. **It works poorly with overlapping classes** and is also sensitive to the type of kernel used.
- Picking right kernel and parameters can be computationally intensive
- In Natural Language Processing (NLP), a structured representation of text yields better performance. However, SVMs cannot accommodate such structures(word embedding)

Thank You