Support vector machines (SVM)

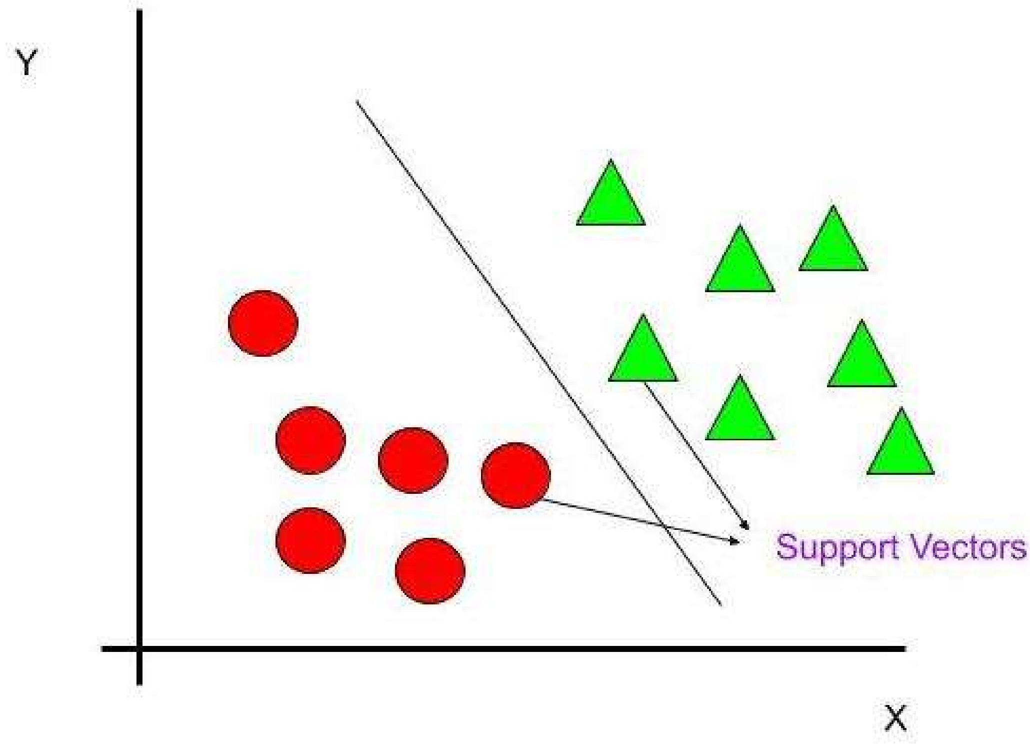
1990→AI called winter year. USA had invested huge amount in military but not successful due to failure of AI.

SVM →During that period SVM Invented →popular & successful More than 200 paper published.

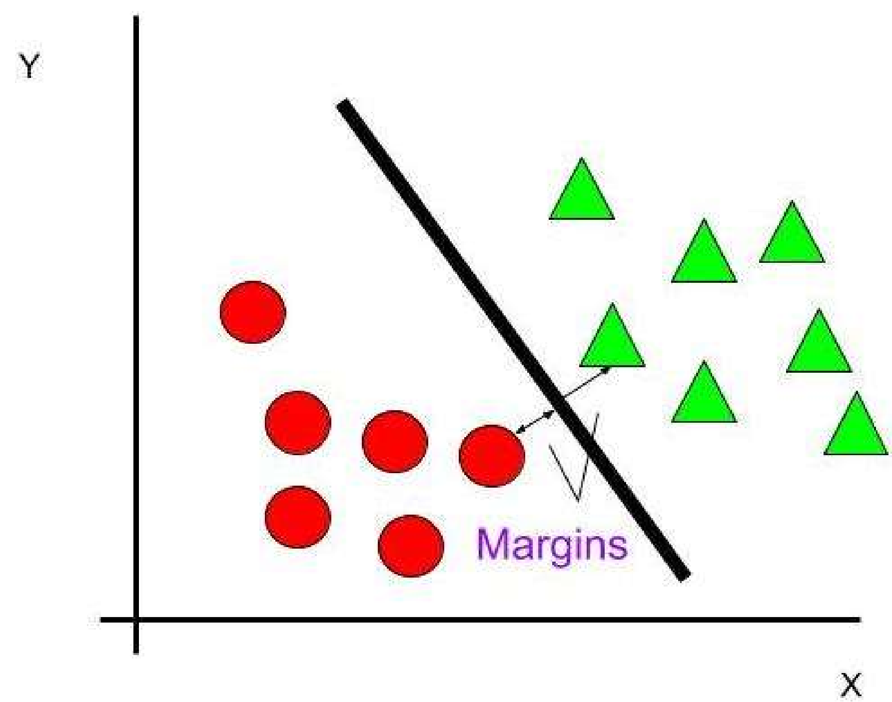
→SVM Most popular, Versatile MLA Used in Regression & Classification

- Preferred for Medium and small sized dataset.

- It separates data in two components using hyper plane by maximizing the margin (Also called large marginal classifier).



- Support Vectors are simply the coordinates of individual observation.
- Dotted line is hyperplane, separating blue and pink classes balls.



Hyper-plane

It is plane that linearly divide the n-dimensional data points in two components. In case of 2D, hyperplane is line,

in case of 3D it is plane.

It is also called as *n-dimensional line*.

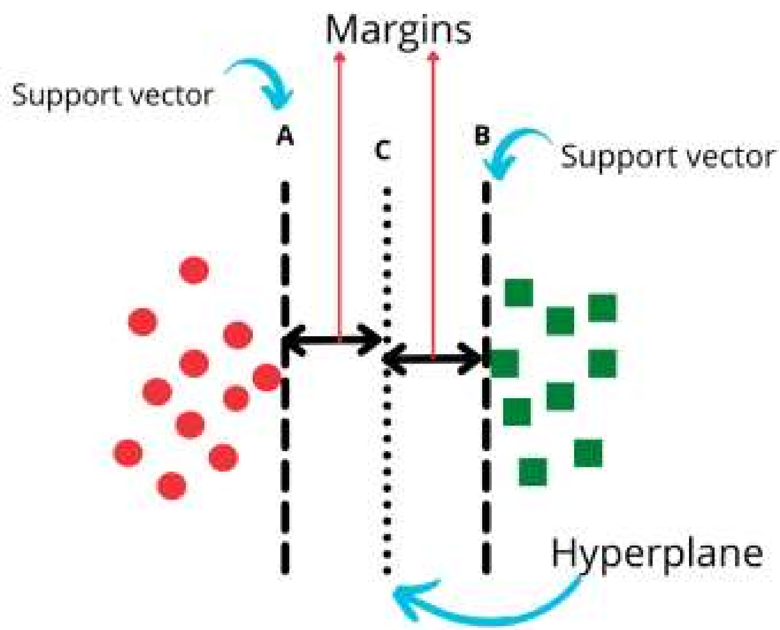
hyperplane line written as

$$y = a * x + b$$

$$a * x + b - y = 0$$

Let vector X=(x,y) and W=(a,-1) then in vector form hyperplane is

$$W . X + b = 0$$



Hyperplane: A-line classify with highest margin .

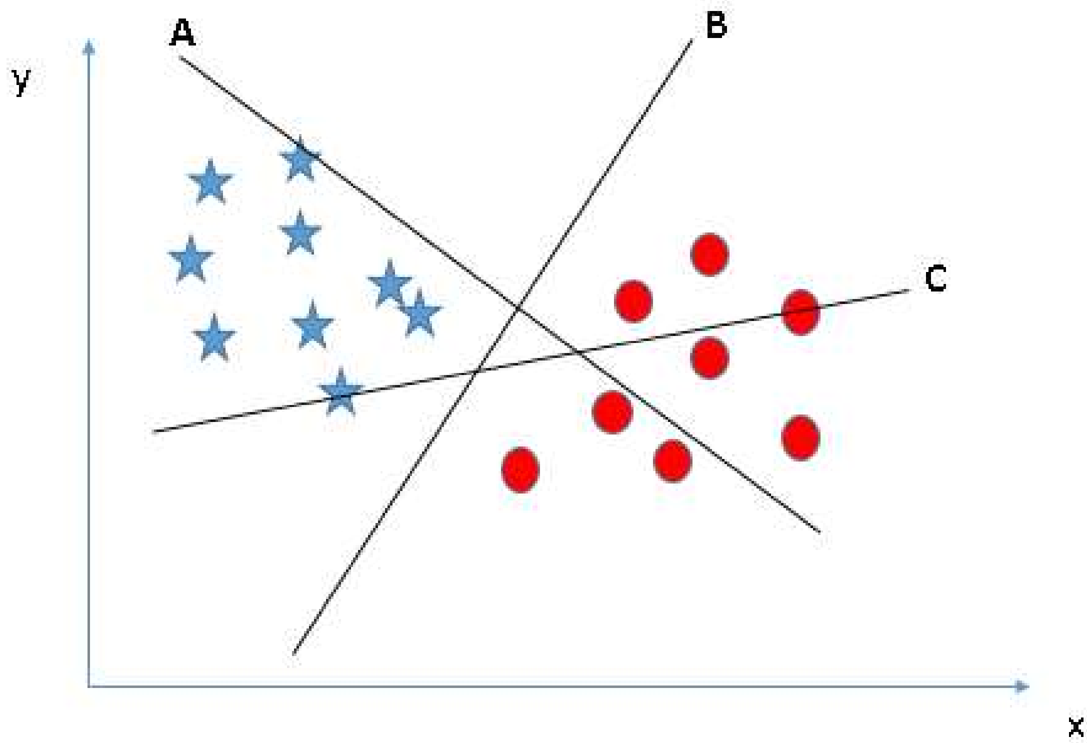
Kernel →means approach

There are different kernel functions available:

- linear
- Gaussian (RBF kernel- Radial Basis Function)
- Polynomial
- Sigmoid

(Scenario-1)

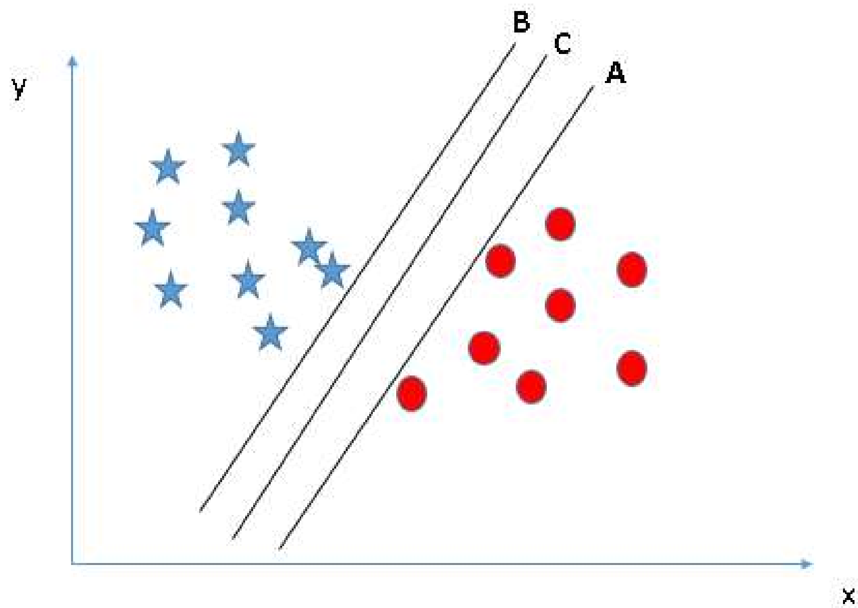
Identify the right hyper-plane: Here, we have three hyper-planes (A, B, and C). Now, identify the right hyper-plane to classify stars and circles.



:- "Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.

(Scenario-2)

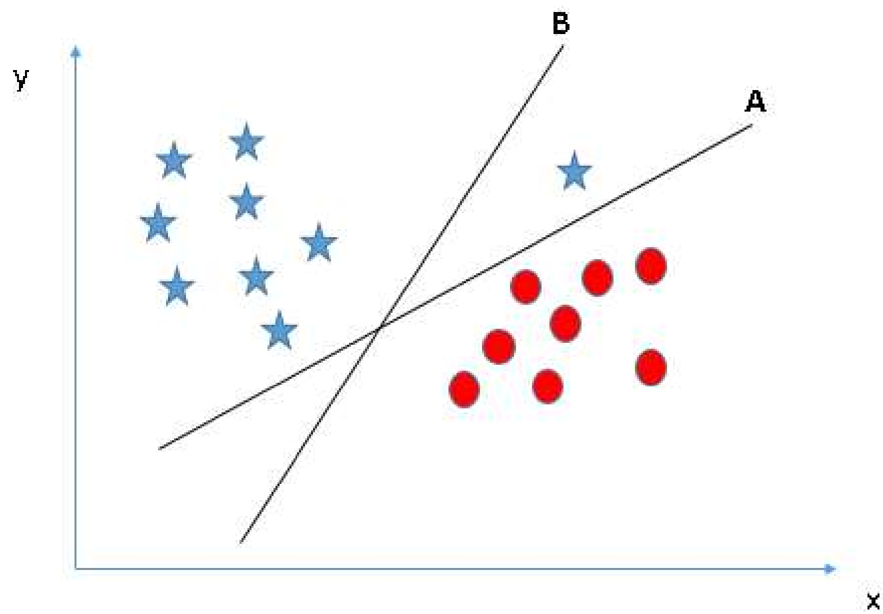
- **Identify the right hyper-plane:** Here, we have three hyper-planes (A, B, and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?



Here, maximizing the **Margin**. the right hyper-plane as C. If we select a hyper-plane having low margin then there is high chance of miss-classification.

(Scenario-3)

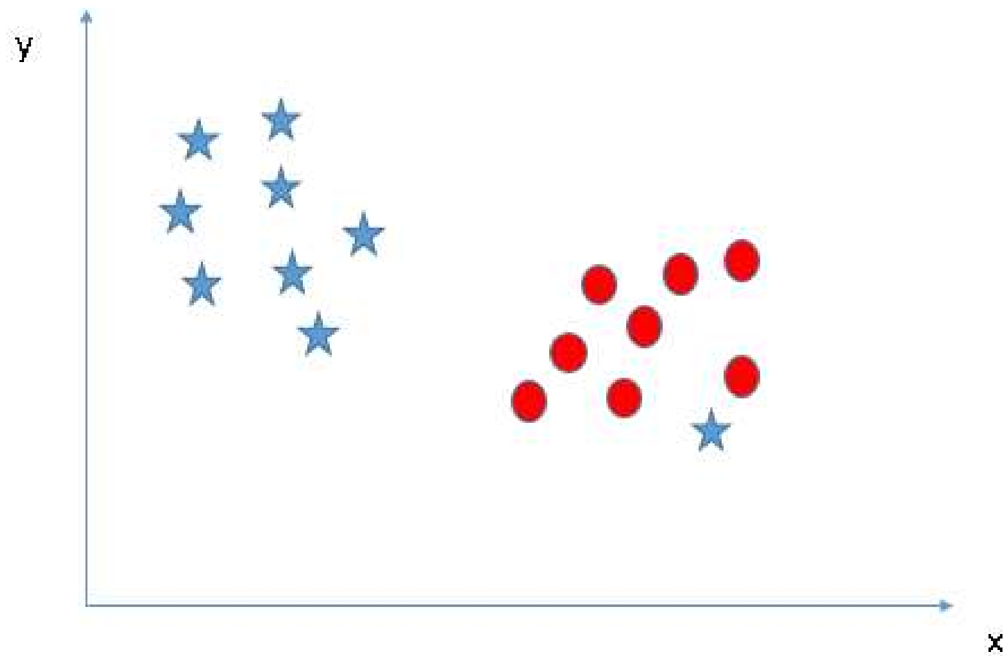
- **Identify the right hyper-plane:** Hint: Use the rules as discussed in previous section to identify the right hyper-plane



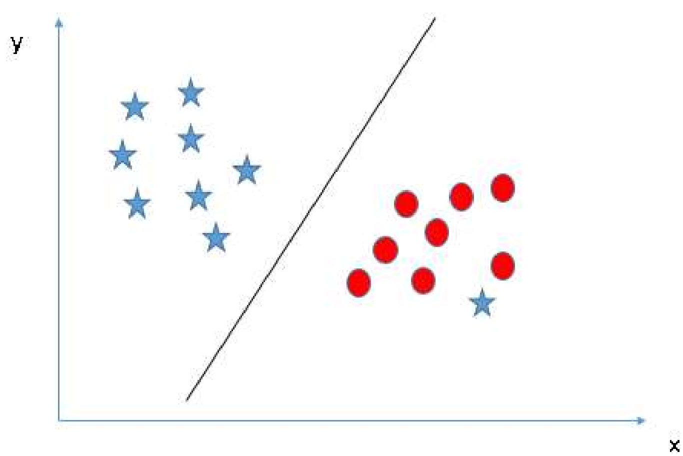
Note:- hyper-plane **B** as it has higher margin compared to **A.**

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.**

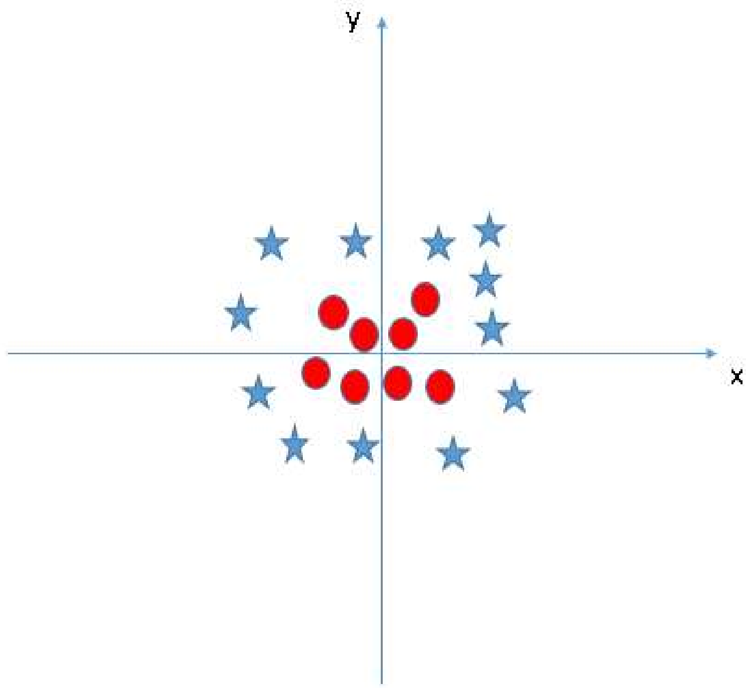
(Scenario-4)



The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, SVM classification is robust to outliers



(Scenario-5)
Find the hyper-plane to segregate to classes: In the scenario below, we can’t have linear hyper-plane between the two classes.

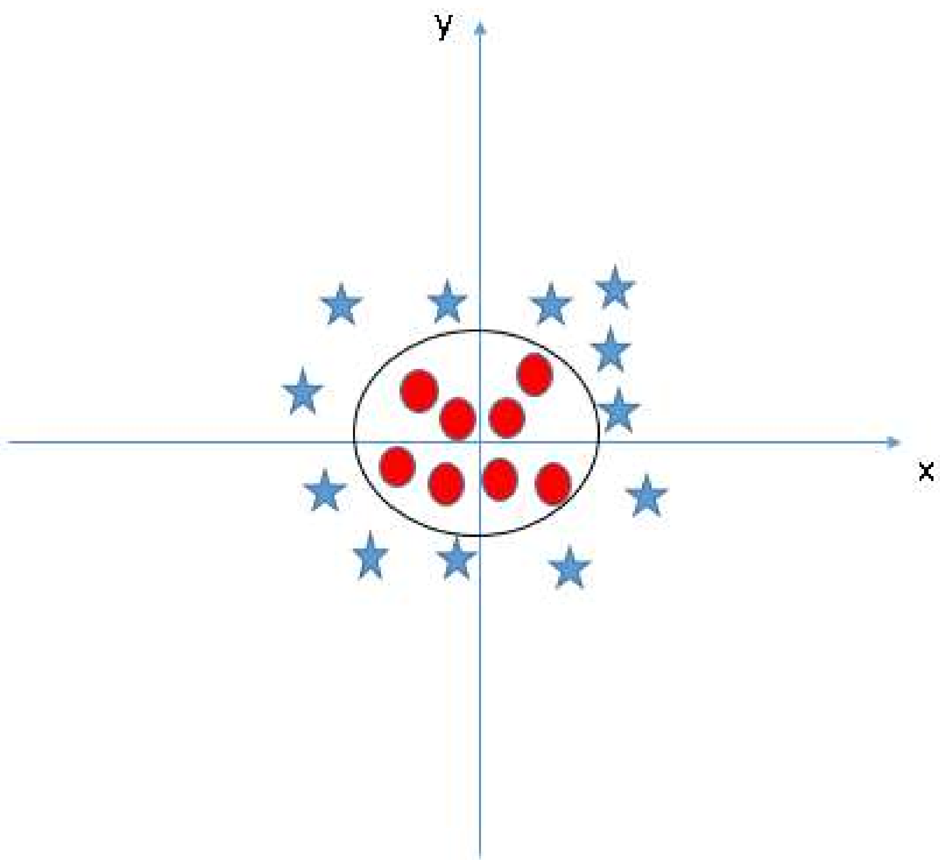


If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line..

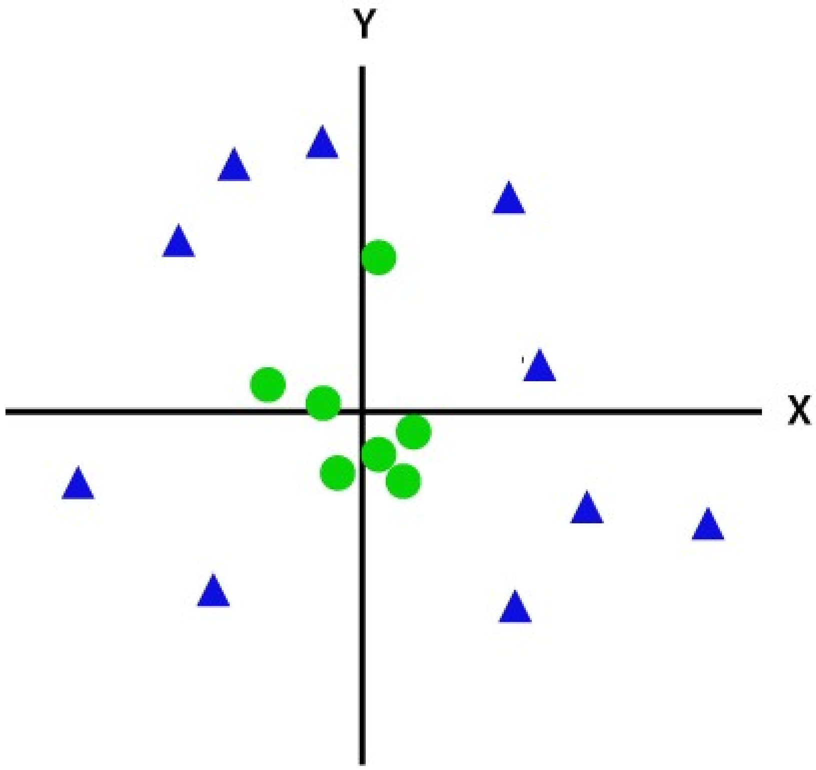
It is possible to solve in SVM *!!*. It will add one extra dimension to the data points to make it separable.

Kernel trick CONVERTING TO HIGH DIMENSIONALITY

It keep on increasing the dimensions unless the classes are separable.



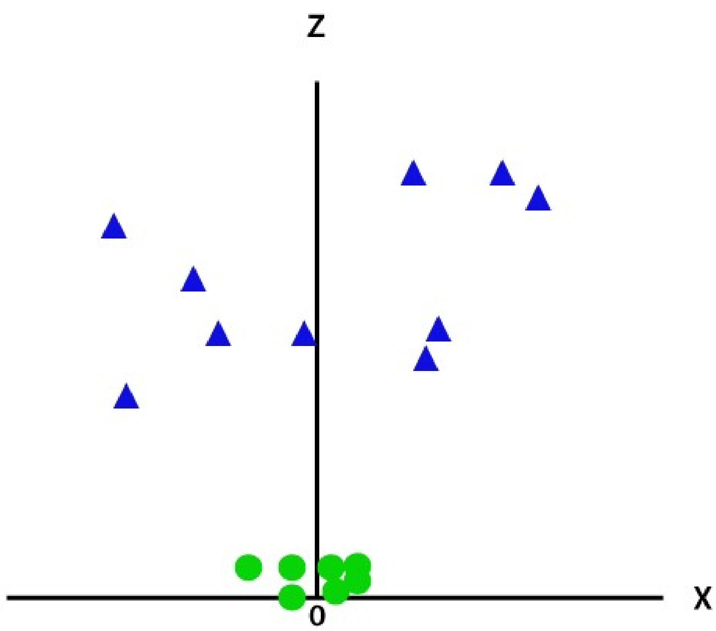
Concept Behind Non Linear Separation:

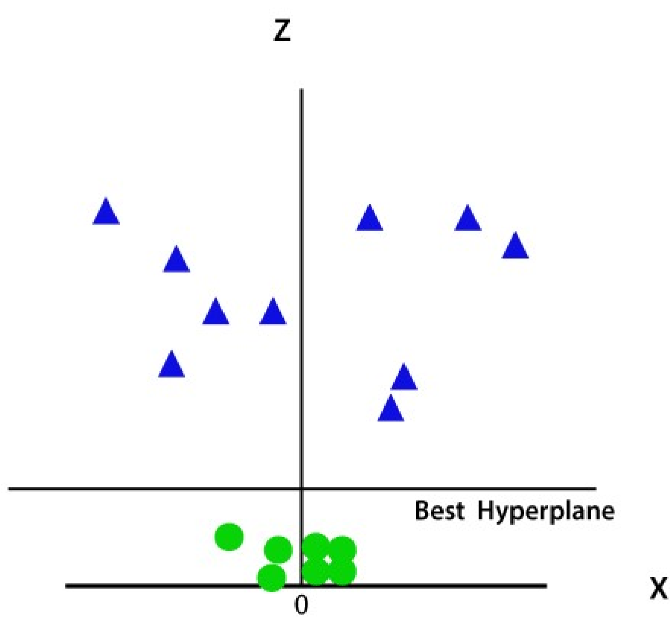


So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions' x and y, so for non-linear data, we will add a third dimension z. **It can be calculated as:**

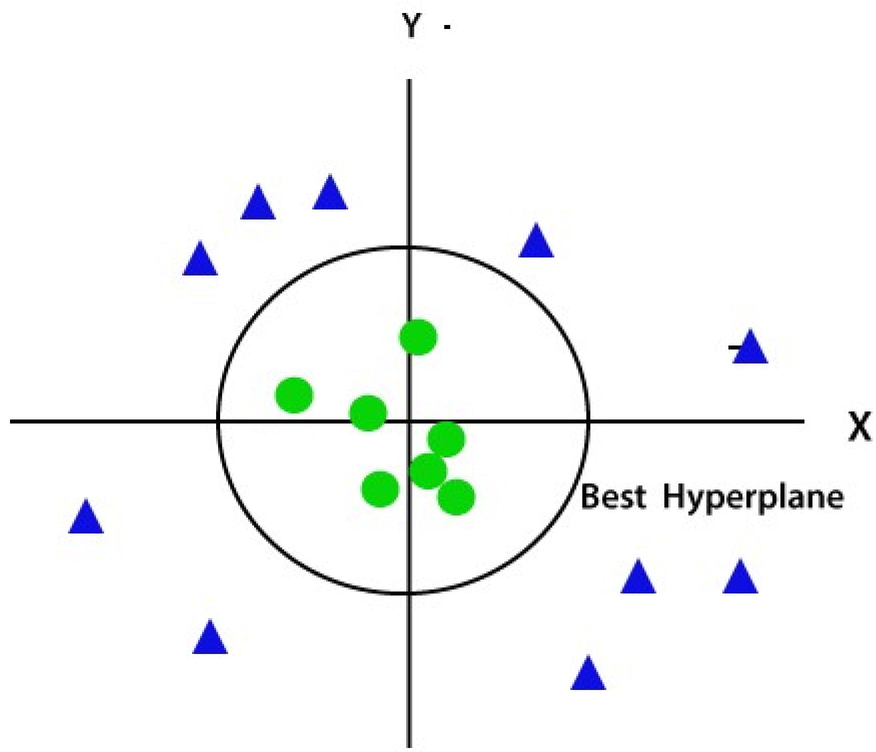
$$z=x^2 +y^2$$

By adding the third dimension, the sample space will become as below image:





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:



Hence we get a circumference of radius 1 in case of non-linear data.

Kernel functions. - transform non-linear spaces into linear spaces. It transforms data into higher dimension so that the data can be classified.

There are various options associated with SVM training; like changing kernel, gamma and C value.

kernel: various options available - “linear”, “rbf”,”poly” and sigmoid (default is “rbf”).

Here “rbf” and “poly” are useful for non-linear hyper-plane.

gamma: Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’. Higher the value of gamma, will try to exact fit the as per training data set i.e. generalization error and cause over-fitting problem.

gamma different gamma values like 1, 10 or 100.

C: Penalty parameter C of the error term. It also controls the trade-off between smooth decision boundaries and classifying the training points correctly.

Pro & Cons associated with SVM: -

Advantage: -

- Works really well with margin of separation.
- Effective in high dimensional space
- Accurate results.
- Useful for both linearly separable and non-linearly separable data.

Disadvantages: -

- It doesn't perform well when we have large dataset because the required training time is very high.

Applications of SVM

- **Sentiment analysis**
- **Spam Detection**
- **Handwritten digit recognition**
- **Image recognition**

```
from sklearn.svm import SVC #"Support vector classifier"
from sklearn.svm import SVR #"Support vector Regressor"

 # Building a Support Vector Machine on train data
svc_model = SVC(C=1, kernel='linear', gamma= 100)
svc_model.fit(X_train, Y_train)
 prediction = svc_model.predict(X_test)
```