**SVM**

**(Support vector machine)**

SVM is a powerful supervised algorithm that works best on smaller datasets but on complex ones.  Support Vector Machine, abbreviated as SVM can be used for both regression and classification tasks, but generally, they work best in classification problems.

SVM works best when the dataset is small and complex.

It is usually advisable to first use logistic regression and see how does it performs, if it fails to give a good accuracy you can go for SVM without any kernel (will talk more about kernels in the later section). Logistic regression and SVM without any kernel have similar performance but depending on your features, one may be more efficient than the other.

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.

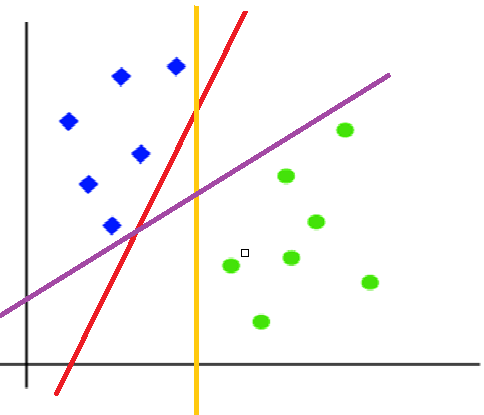


## Types of Support Vector Machine

#### SVM Linear

When the data is perfectly linearly separable only then we can use Linear SVM. Perfectly linearly separable means that the data points can be classified into 2 classes by using a single straight line(if 2D).

* **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 the classifier is used called Linear SVM classifier.



#### Non-Linear SVM

When the data is not linearly separable then we can use Non-Linear SVM, which means when the data points cannot be separated into 2 classes by using a straight line (if 2D) then we use some advanced techniques like kernel tricks to classify them. In most real-world applications we do not find linearly separable data points hence we use kernel tricks to solve them

**SVM can be of two types:**

* **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 the classifier used is called a Non-linear SVM classifier.

**Support Vectors:**These are the points that are closest to the hyperplane. A separating line will be defined with the help of these 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.

## Margin: it is the distance between the hyperplane and the observations closest to the hyperplane (support vectors). In SVM large margin is considered a good margin. There are two types of margins hard margin and soft margin.

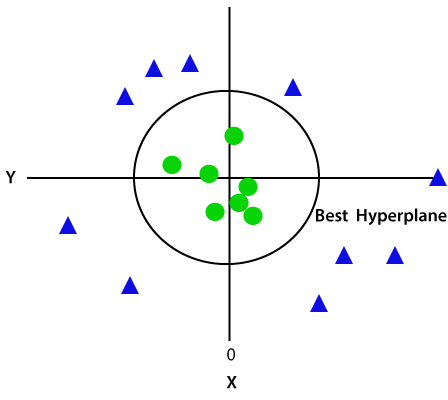
**NOTE:**Since we are plotting the data points in a 2-dimensional graph we call this decision boundary a **straight line** but if we have more dimensions, we call this decision boundary a **“hyperplane”**





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=x2 +y2



## Kernels in Support Vector Machine

The most interesting feature of SVM is that it can even work with a non-linear dataset and for this, we use “Kernel Trick” which makes it easier to classifies the points

#### Polynomial kernel

#### Sigmoid kernel

#### 3. RBF kernel

#### 4. Bessel function kernel

#### 5. Anova Kernel

Pros and Cons associated with SVM

* **Pros:**
  + It works really well with a clear margin of separation
  + It is effective in high dimensional spaces.
  + It is effective in cases where the number of dimensions is greater than the number of samples.
  + It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* **Cons:**
  + It doesn’t perform well when we have large data set because the required training time is higher
  + It also doesn’t perform very well, when the data set has more noise i.e. target classes are overlapping
  + SVM doesn’t directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is included in the related SVC method of Python scikit-learn library.

Library calling

from sklearn.svm import SVC

SVC**( \*,** C**=1.0,** kernel**='rbf',** degree**=3,** gamma**='scale',** coef0**=0.0,** shrinking**=True,** probability**=False,** tol**=0.001,** cache\_size**=200,** class\_weight**=None,** verbose**=False,** max\_iter**=-1,** decision\_function\_shape**='ovr',** break\_ties**=False,** random\_state**=None, )**