Support Vector Machine

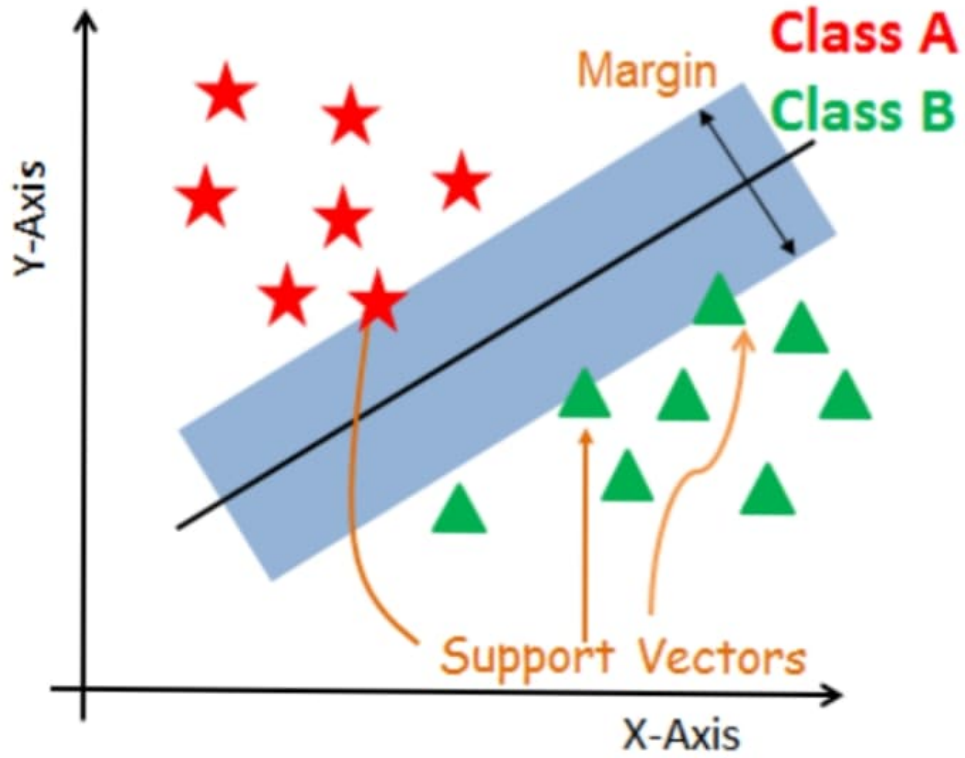
SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. It is known for its kernel trick to handle nonlinear input spaces. It is used in a variety of applications such as face detection, intrusion detection, classification of emails, news articles and web pages, classification of genes, and handwriting recognition.

Support Vector Machines

Support Vector Machines is considered to be a classification approach. 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.

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.

Hyperplane

A hyperplane is a decision plane which separates between a set of objects having different class memberships.

Margin

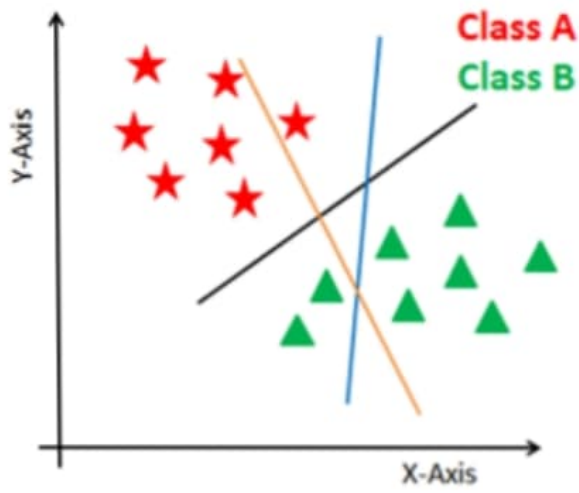
A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

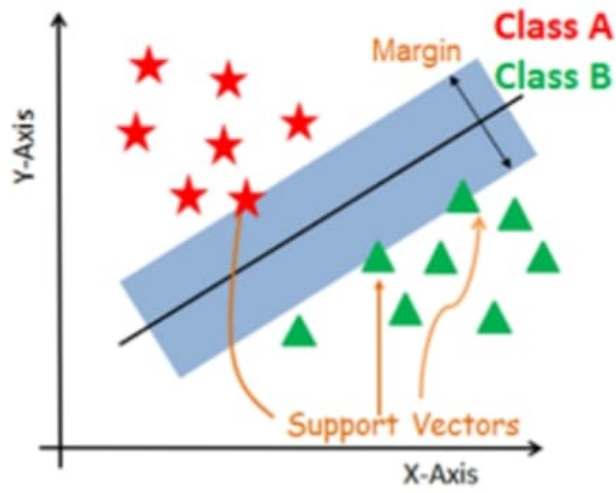
How does SVM work?

The main objective is to segregate the given dataset in the best possible way. The distance between the either nearest points is known as the margin. The objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum marginal hyperplane in the following steps:

1. Generate hyperplanes which segregates the classes in the best way. Left-hand side figure showing three hyperplanes black, blue and orange. Here, the blue and orange have higher classification error, but the black is separating the two classes correctly.

2. Select the right hyperplane with the maximum segregation from the either nearest data points as shown in the right-hand side figure.





Breast Cancer Dataset with Support Vector Machine

```
1  from sklearn import datasets
2  from sklearn.model_selection import train_test_split
3  from sklearn import svm
4  from sklearn import metrics
5
6  def MarvellousSVM():
7      #Load dataset
8      cancer = datasets.load_breast_cancer()
9
10     # print the names of the 13 features
11     print("Features of the cancer dataset : ", cancer.feature_names)
12
13     # print the label type of cancer('malignant' 'benign')
14     print("Labels of the cancer dataset : ", cancer.target_names)
15
16     # print data(feature)shape
17     print("Shape of dataset is :",cancer.data.shape)
18
19     # print the cancer data features (top 5 records)
20     print("First 5 records are : ")
21     print(cancer.data[0:5])
22
23     # print the cancer labels (0:malignant, 1:benign)
24     print("Target of dataset : ", cancer.target)
25
26     # Split dataset into training set and test set
27     X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target,
           test_size=0.3,random_state=109) # 70% training and 30% test
28
29     #Create a svm Classifier
30     clf = svm.SVC(kernel='linear') # Linear Kernel
31
32     #Train the model using the training sets
33     clf.fit(X_train, y_train)
34
35     #Predict the response for test dataset
36     y_pred = clf.predict(X_test)
37
38     # Model Accuracy: how often is the classifier correct?
39     print("Accuracy of the model is :",metrics.accuracy_score(y_test, y_pred)*100)
40
41 def main():
42     print("_______ Marvellous Support Vector Machine _________")
43
44     MarvellousSVM()
45
46 if __name__ == "__main__":
47     main()
48
```