**EXPERIMENT NO. 6**

CLASS: TE CMPN A PID: 182027

NAME: REBECCA DIAS ROLL NO. : 19

**Aim: -** To implement SVM for

a. Iris data set

b. Dataset of your choice

**Theory:**

**What is SVM?**

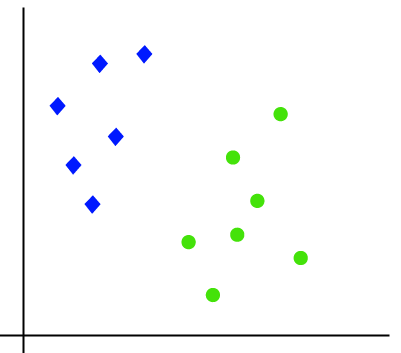
Support Vector Machine (SVM) is a relatively simple Supervised Machine Learning Algorithm used for classification and/or regression. It is more preferred for classification but is sometimes very useful for regression as well. Basically, SVM finds a hyper-plane that creates a boundary between the types of data. In 2-dimensional space, this hyper-plane is nothing but a line.  
In SVM, we plot each data item in the dataset in an N-dimensional space, where N is the number of features/attributes in the data. Next, find the optimal hyperplane to separate the data. So by this, you must have understood that inherently, SVM can only perform binary classification (i.e., choose between two classes). However, there are various techniques to use for multi-class problems.

Support Vector Machine for Multi-Class Problems  
To perform SVM on multi-class problems, we can create a binary classifier for each class of the data. The two results of each classifier will be:

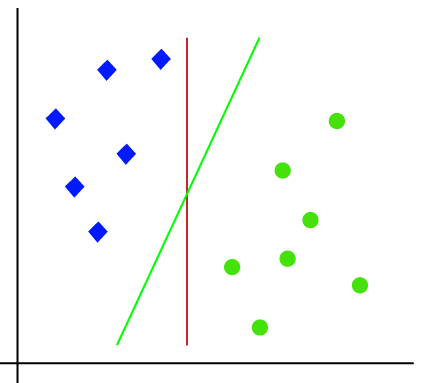
* The data point belongs to that class OR
* The data point does not belong to that class.
* For example, in a class of fruits, to perform multi-class classification, we can create a binary classifier for each fruit. For say, the ‘mango’ class, there will be a binary classifier to predict if it IS a mango OR it is NOT a mango. The classifier with the highest score is chosen as the output of the SVM.
* SVM for complex (Non Linearly Separable)  
  SVM works very well without any modifications for linearly separable data. **Linearly Separable Data**is any data that can be plotted in a graph and can be separated into classes using a straight line.

**How is it being used as a classifier?**

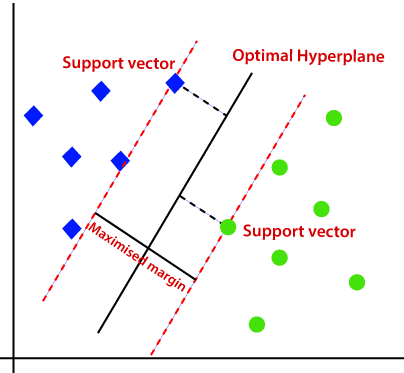
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



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. Consider the below image:

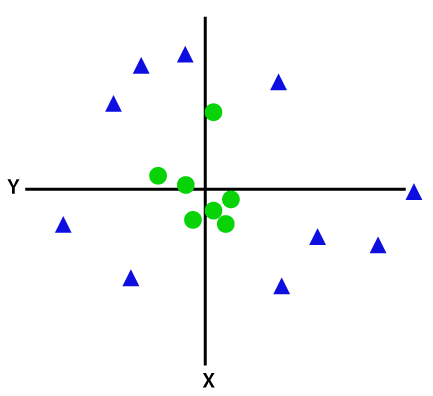


Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is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



**Non-Linear SVM:**

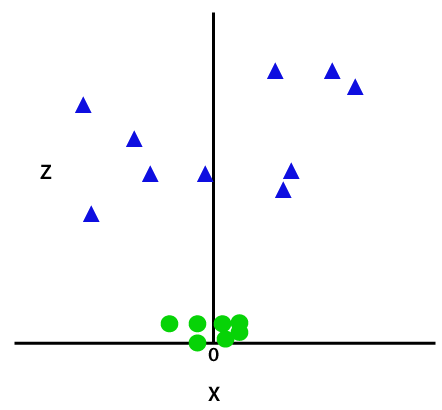
If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



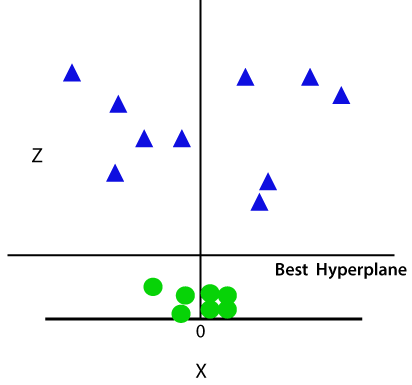
So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions’ x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

**z=x2 +y2**

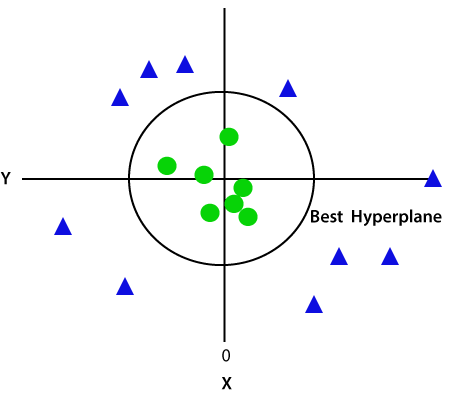
By adding the third dimension, the sample space will become as below image:



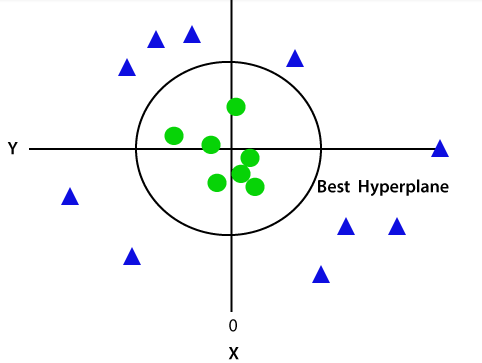
So now, SVM will divide the datasets into classes in the following way. Consider the below image:



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:



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:



**What is the optimization problem in the case of SVM?**

**1. Consider a dataset**

**2. Select two hyperplanes which separate the data with no points between them**

**3. Maximize their distance (the margin)**

**Consider a dataset**

• Data will be composed of n vectors xi.

• Each xi will also be associated with a value yi indicating if the element belongs to the class (+1) or not (-1).

• Consider xi is a p-dimensional vector if it has p dimensions.

• Finding two hyperplanes separating p-dimensional date is difficult because you can't draw it.

• Even if your data is 2-dimensional, we cannot find a separating hyperplane if your data is not linearly separable.

• Let us assume that our dataset D is linearly separable.

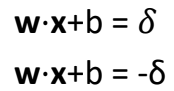
**Select two hyperplanes which separate the data with no points between them**

• The equation of a hyperplane can be written as wTx = 0

• Any hyperplane can be written as the set of points x satisfying w⋅x+b = 0.

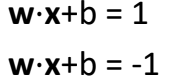
• Given a hyperplane H0 separating the dataset and satisfying: w⋅x+b = 0

Select two others hyperplanes H1and H2 which also separate the data and have the following equations:



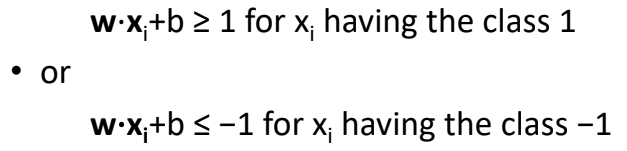
• So that H0 is equidistant from H1 and H2.

• We can set δ=1 to simplify the problem.



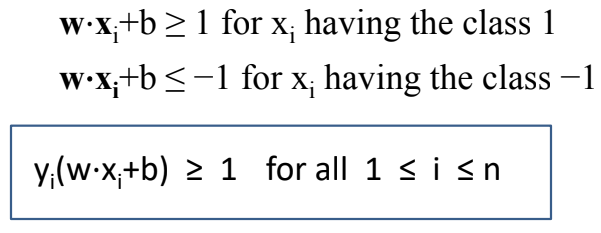
• We won't select any hyperplane; we will only select those who meet the two following constraints:

• For each vector xi either:



• Combining both constraints

• We can combine the following constraints



**Maximize the distance between the two hyperplanes**

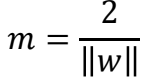
we need a vector

– to have a magnitude of m

– to be perpendicular to the hyperplane H1

• Fortunately, we already know a vector perpendicular to H1 , that is w. (because H1=w⋅x+b = 1)

To compute the margin:



The only variable we can change in this formula is

the norm of w. Let's try to give it different values:

– When ∥w∥=1 then m=2

– When ∥w∥=2 then m=1

– When ∥w∥=4 then m=0.5

• The bigger the norm is, the smaller the margin become.

**Code:**

<https://colab.research.google.com/drive/1ZJwHL2lMp8amZaYQfX7ebmCyIyRUHcu7?usp=sharing>

**Conclusion:**

In this experiment, we learnt about classification using SVM. SVM stands for support vector machine. It comes under the category of supervised learning. We implemented SVM on Iris dataset. We worked on different kernels like sigmoid, Gaussian and polynomial. The linear kernel is what you would expect, a linear model. The polynomial kernel is similar, but the boundary is of some defined but arbitrary order. RBF uses normal curves around the data points, and sums these so that the decision boundary can be defined by a type of topology condition such as curves where the sum is above a value of 0.5.

Iris species were classified using SVM. We also generated a classification report. We then applied SVM classifier for the diabetes dataset. We split our model into train data and test data and trained our model. We then provided an unknown sample to the system and it was correctly classified.

Accuracy was found to be 0.78.

Hence, the output was obtained successfully.