EXPERIMENT NO. 7

AIM: Implement Support Vector Machine (SVM) Algorithm for the given dataset.

SOFTWARE: Python

LABORATORY OUTCOMES:

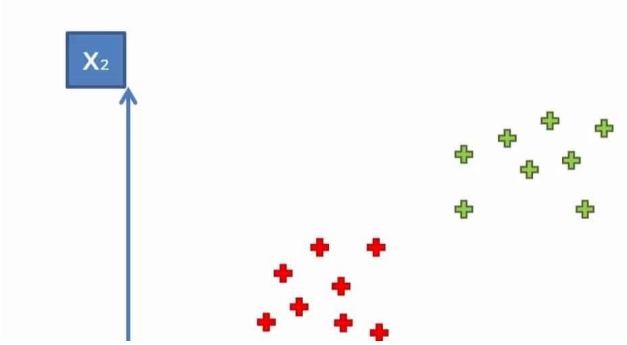
- Students will be able to implement Support Vector Machine algorithm for the given dataset.
- Students will be able to find the **optimal line or best decision boundary** to separate points into different spaces.
-

THEORY:

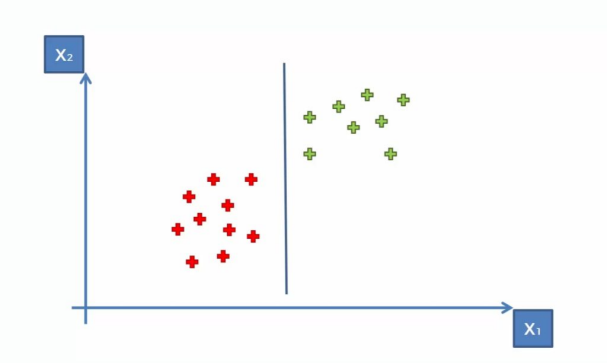
SVM was developed in the **1960s** and refined in the **1990s.** It becomes very popular in the machine learning field because SVM is very powerful compared to other algorithms. SVM (Support Vector Machine) is a **supervised machine learning algorithm**. That's why training data is available to train the model. SVM uses a classification algorithm to classify a two-group problem. SVM focus on **decision boundary and support vectors.**

How SVM Works?

Here, we have two points in two-dimensional space, we have two columns x1 and x2. We have some observations such as **red and green**, which are already classified. This is linearly separable data.

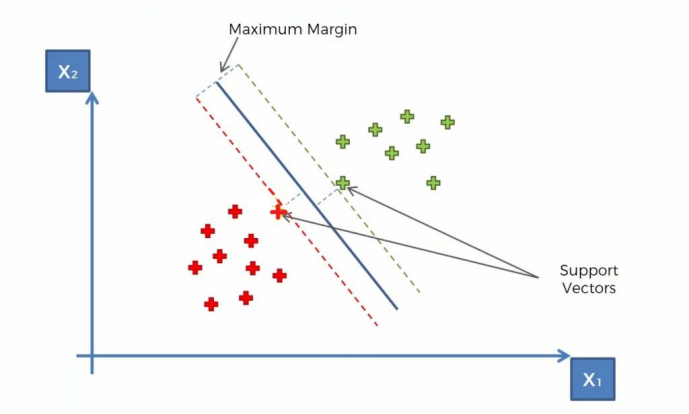


To classify new points, we need to create a boundary between two categories, and when in the future we will add new points and we want to classify them, then we know where they belong. Either in a Green Area or Red Area. One way to separate is to draw a vertical line between two areas, so anything on the right is Red and anything on the left is Green. Something like that-



However, there is one more way, draw a **horizontal line or diagonal line**. You can create multiple diagonal lines, which achieve similar results to separate our points into two classes. But our main task

is to find the **optimal line or best decision boundary.** And for this SVM is used. SVM finds the best decision boundary, which helps us to separate points into different spaces.
SVM finds the best or optimal line through the **maximum margin**, which means it has max distance and equidistance from both classes or spaces. The sum of these two classes has to be maximized to make this line the maximum margin.



These, two vectors are **support vectors.** In SVM, only support vectors are contributing. That's why these points or vectors are known as **support vectors**. Due to support vectors, this algorithm is called a **Support Vector Algorithm(SVM).** In the picture, the line in the middle is a **maximum margin hyperplane** or classifier. In a two-dimensional plane, it looks like a line, but in a multi-dimensional, it is a **hyperplane**.

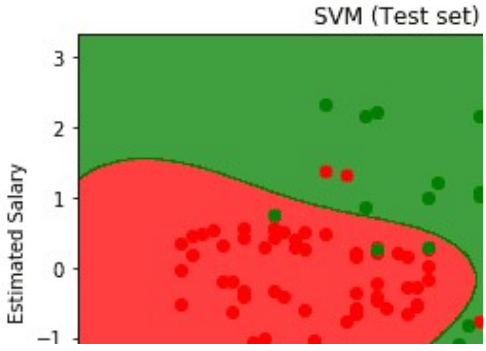
PROGRAM:

```
#Import libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
#Load the Dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
#Split Dataset into X and Y
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
#Split the X and Y Dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
#Perform Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
#Fit SVM to the Training set
from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(X_train, y_train)
#Predict the Test Set Results
y_pred = classifier.predict(X_test)
```

```
#Make the Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test,y_pred)
#Visualise the Test set results

from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step =
0.01),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('SVM (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

OUTPUTS:



CONCLUSION:

Implementation of SVM for classification of linear separable data is carried out here. In the output image, there are a total of 7 incorrect predictions. There are 3 green(Yes) predictions that were predicted as Red(No) and 4 Red(No) predictions that were predicted as Green(Yes). But overall we got 93% accuracy.

TEXT/REFERENCE BOOKS:

- "Neural Network a Comprehensive Foundation" By Simon Haykin
- "Introduction to Soft Computing" By Dr. S. N. Shivnandam, Mrs. S. N. Deepa
- "Neural Network: A classroom Approach" By Satish Kumar
- "Neural Network, Fuzzy Logic and Genetic Algorithms" By Rajshekharan S, Vijayalakshmi

Pai
- "Neural Network Design" by Hagan Demuth, Beale"Neural Network for Pattern Recognition", Christopher M. Bishop

WEB ADDRESS (URLS):

- https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_associate_memory.htm
- http://cns.bu.edu/~nicks/project/html/thesis/node76.html
- http://staff.fit.ac.cy/com.ke/files/acoe402/Neural_networks_2.pdf

----------x----------x----------