Maitreya Kanitkar
BE-I-1 8084

# MLA Assignment 3

**Aim:-**
Implement SVM for performing classification and find its accuracy on the given data.

**Title:-**
Support Vector Machine.

**Theory:-**

Q1) What is Support Vector Machine?

→ Support Vector Machine

Support Vector Machine, or SVM is a supervised machine learning algorithm used for classification. It is sometimes also used for regression. SVM finds a hyperplane that creates a boundary between types of data. In 2-dimensional spaces, this hyperplane 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 in the data. Next, find the optimal hyperplane to separate the data. SVM is used for both binary and multi-class classification 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 willbe-

i) The data points belong to that class OR
ii) The data point doesn't belong to that class.

Q2) Explain Support Vectors.

→ Support vectors are data points that are closer the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

Q3) Differentiate between Hard margin and soft margin SVM.

→

| | Hard margin SVM | Soft Margin SVM |
|---|---|---|
| a) | Used if data is linearly separable. | If data is not linearly separable we use Soft margin SVM |
| b) | Used for proper classification with no misclassified data. | Used for better model generalization. |

Q4

Q4) Explain in brief gamma and regularization parameters.

→ Gamma-parameter-
Gamma is used when we use the Gaussian RBF kernel. If you use linear or polynomial kernel then you only need C hyperparameter.
Gamma is a hyperparameter which we have to set before training model. Gamma decides that how much curvature we want in a decision boundary.

High Gamma means more curvature and less low gamma means less curvature. What value of gamma one should use totally depends upon the data.

Regularization parameter -
The regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example.
For large values of C, the optimization will choose a smaller margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

In [1]:

```
#Support Vector Machine(SVM)
```

In [2]:

```
#importing required libraries
from sklearn import svm, metrics
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mlxtend.plotting import plot_decision_regions
```

In [3]:

```
#loading the dataset
data=pd.read_csv("SVM_DataSet_1.csv")
data.head()
```

Out[3]:

| | User ID | Gender | Age | EstimatedSalary | Purchased |
|---|---|---|---|---|---|
| **0** | 15624510 | Male | 19 | 19000 | 0 |
| **1** | 15810944 | Male | 35 | 20000 | 0 |
| **2** | 15668575 | Female | 26 | 43000 | 0 |
| **3** | 15603246 | Female | 27 | 57000 | 0 |
| **4** | 15804002 | Male | 19 | 76000 | 0 |

In [4]:

```
#Splitting the dataset into features and labels
X = data.iloc[:, [2, 3]].values
y = data.iloc[:, 4].values
```

In [5]:

```
#Splitting the dataset into training and testing set
X_train, X_test, y_train, y_test=train_test_split(X, y, train_size=0.8)
```

In [6]:

```
#SVM model(Kernel=Linear)
linear_model=svm.SVC(kernel='linear')
linear_model.fit(X_train, y_train)

#prediction
prediction=linear_model.predict(X_test)
```

```
print("Actual set:\n", y_test)
print("Predictions:\n", prediction)

#Confusion matrix
confusion=metrics.confusion_matrix(y_test, prediction)
print("Confusion matrix:\n", confusion)

#classification report
report=metrics.classification_report(y_test, prediction)
print("Classification report:\n", report)

#plot
plot_decision_regions(X=X, y=y, clf=linear_model)
plt.xlabel("Age", size=14)
plt.ylabel("EstimatedSalary", size=14)
plt.title('SVM Decision Region Boundary', size=16)
```

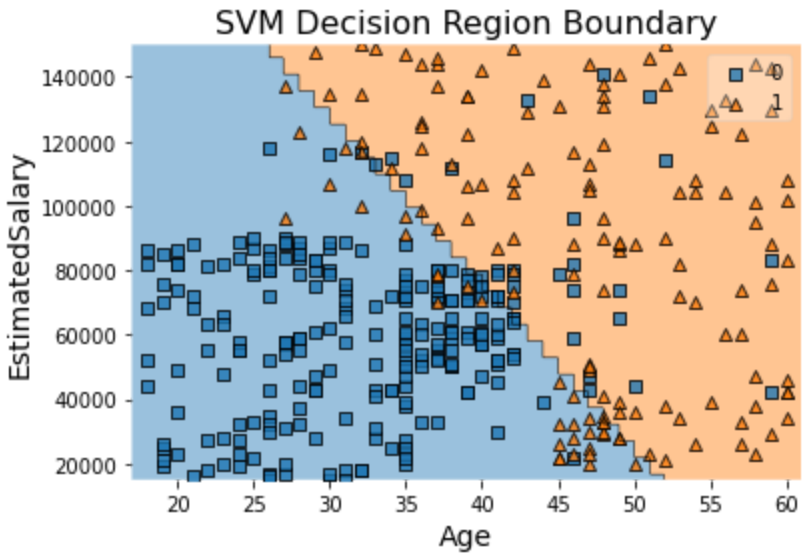
```
Actual set:
 [0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 1 0
 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1
 1 0 0 0 0 1]
Predictions:
 [0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 1 0
 1 0 1 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1
 0 0 0 0 0 1]
Confusion matrix:
 [[52  4]
 [ 2 22]]
Classification report:
              precision    recall  f1-score   support

           0       0.96      0.93      0.95        56
           1       0.85      0.92      0.88        24

    accuracy                           0.93        80
   macro avg       0.90      0.92      0.91        80
weighted avg       0.93      0.93      0.93        80
```

Out[6]:

```
Text(0.5, 1.0, 'SVM Decision Region Boundary')
```




In [7]:

```
#SVM model(kernel=rbf)
rbf_model=svm.SVC()
rbf_model.fit(X_train, y_train)

#prediction
prediction=rbf_model.predict(X_test)
print("Actual set:\n", y_test)
print("Predictions:\n", prediction)

#Confusion matrix
confusion=metrics.confusion_matrix(y_test, prediction)
print("Confusion matrix:\n", confusion)

#classification report
report=metrics.classification_report(y_test, prediction)
print("Classification report:\n", report)

#plot
plot_decision_regions(X=X, y=y, clf=rbf_model)
plt.xlabel("Age", size=14)
plt.ylabel("EstimatedSalary", size=14)
plt.title('SVM Decision Region Boundary', size=16)
```

```
Actual set:
 [0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 1 0
 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1
 1 0 0 0 0 1]
Predictions:
 [0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0
```

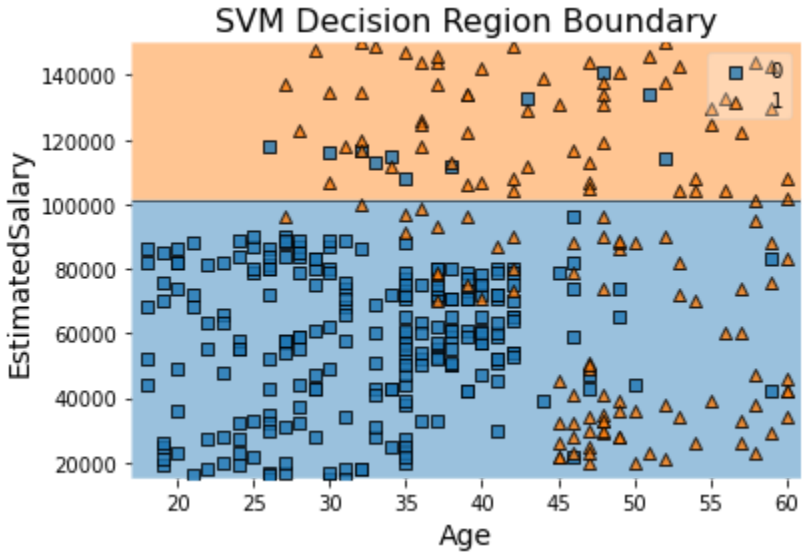
```
 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
 0 0 0 0 0 0]
Confusion matrix:
 [[55  1]
 [11 13]]
Classification report:
              precision    recall  f1-score   support

           0       0.83      0.98      0.90        56
           1       0.93      0.54      0.68        24

    accuracy                           0.85        80
   macro avg       0.88      0.76      0.79        80
weighted avg       0.86      0.85      0.84        80
```

Out[7]:

Text(0.5, 1.0, 'SVM Decision Region Boundary')




In [11]:

```
#SVM model(kernel=poly)
poly_model=svm.SVC(kernel='poly', C=4, gamma=0.00000001, max_iter=1e8)
poly_model.fit(X_train, y_train)

#prediction
prediction=poly_model.predict(X_test)
print("Actual set:\n", y_test)
print("Predictions:\n", prediction)

#Confusion matrix
confusion=metrics.confusion_matrix(y_test, prediction)
```

```
print("Confusion matrix:\n", confusion)

#classification report
report=metrics.classification_report(y_test, prediction)
print("Classification report:\n", report)

#plot
plot_decision_regions(X=X, y=y, clf=rbf_model)
plt.xlabel("Age", size=14)
plt.ylabel("EstimatedSalary", size=14)
plt.title('SVM Decision Region Boundary', size=16)
```

```
C:\Users\Maitreya\anaconda3\envs\tensorflow\lib\site-packages\sklearn\svm\_base.py:255:
ConvergenceWarning: Solver terminated early (max_iter=100000000).  Consider pre-processing
your data with StandardScaler or MinMaxScaler.
  warnings.warn('Solver terminated early (max_iter=%i).'
```

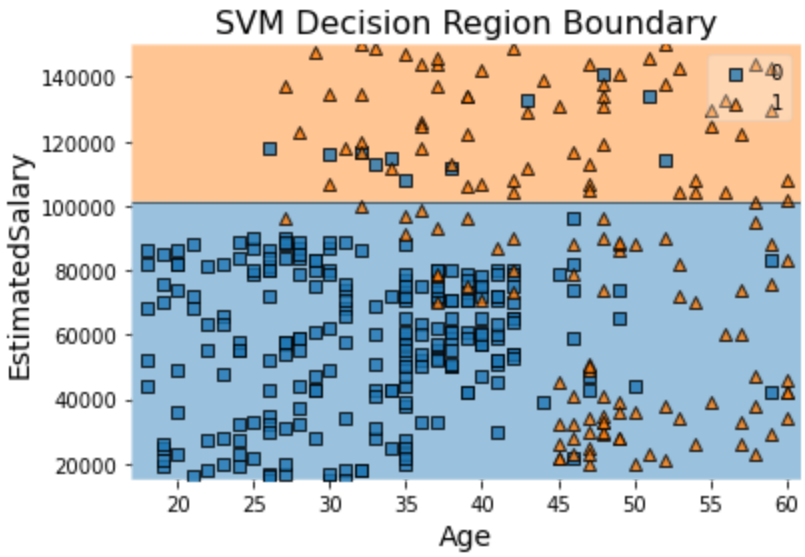
```
Actual set:
 [0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 1 0
 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1
 1 0 0 0 0 1]
Predictions:
 [0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0
 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1
 0 0 0 0 0 0]
Confusion matrix:
 [[55  1]
 [ 7 17]]
Classification report:
              precision    recall  f1-score   support

           0       0.89      0.98      0.93        56
           1       0.94      0.71      0.81        24

    accuracy                           0.90        80
   macro avg       0.92      0.85      0.87        80
weighted avg       0.90      0.90      0.90        80
```

Out[11]:

Text(0.5, 1.0, 'SVM Decision Region Boundary')


SVM Decision Region Boundary
EstimatedSalary
Age
0
1
140000
120000
100000
80000
60000
40000
20000
20
25
30
35
40
45
50
55
60