Kawar Nilesh Ramesh
Roll no : 59
Batch A

Practical 6

Implementation of classifying data using SVM's

Program with Output:

```
#import Libraries
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix, classification_report, ac
curacy_score
```

```
#import Libraries
import numpy as np
import pandas as pd
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

```
#import dataset
data=pd.read_csv('Social_Network.csv')
data.head()
```

```
#import dataset
data=pd.read_csv('Social_Network.csv')
data.head()
```

	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

```
#Extracting Independent and dependent variable
x = data.iloc[:,[2,3]].values
y = data.iloc[:,4].values
```

Kawar Nilesh Ramesh
Roll no : 59
Batch A

```
#Extracting Independent and dependent variable
x = data.iloc[:,[2,3]].values
y = data.iloc[:,4].values
```

```
#splitting the dataset into trainging ans testing 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)
```

```
#splitting the dataset into trainging ans testing 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)
```

```
#feature scaling
from sklearn.preprocessing import StandardScaler
st_x = StandardScaler()
x_train = st_x.fit_transform(x_train)
x_test = st_x.transform(x_test)
```

```
[6] #feature scaling
    from sklearn.preprocessing import StandardScaler
    st_x = StandardScaler()
    x_train = st_x.fit_transform(x_train)
    x_test = st_x.transform(x_test)
```

```
#Fitting train dataset
from sklearn.svm import SVC     #support vector classifier
classifier = SVC(kernel = 'linear',random_state = 0)       #linear SVM
classifier.fit(x_train,y_train)
```

```
#Fitting train dataset
from sklearn.svm import SVC     #support vector classifier
classifier = SVC(kernel = 'linear',random_state = 0)       #linear SVM
classifier.fit(x_train,y_train)

SVC(kernel='linear', random_state=0)
```

```
#predicting the test set result
y_pred=classifier.predict(x_test)
y_predTrain=classifier.predict(x_train)
```

Kawar Nilesh Ramesh
Roll no : 59
Batch A

```
print(y_pred)
print('\n',y_predTrain)
```

```
#predicting the test set result
y_pred=classifier.predict(x_test)
y_predTrain=classifier.predict(x_train)
print(y_pred)
print('\n',y_predTrain)
```

```
[0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0
 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0
 0 0 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1]

 [0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 1 1 0 1
 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 1 1 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 0 1 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 0 0 0 1 0 0
 1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0
 0 1 1 0 0 0 0 1 0 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0
 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0
 0 0 0 0]
```

```
#confusion matrix for test set
cm = confusion_matrix(y_test,y_pred)
print('confusion matrix for test data\n', cm)

#confusion matrix for train set
cm1 = confusion_matrix(y_train,y_predTrain)
print('\nconfusion matrix for train data\n', cm1)
```

```
#confusion matrix for test set
cm = confusion_matrix(y_test,y_pred)
print('confusion matrix for test data\n', cm)

#confusion matrix for train set
cm1 = confusion_matrix(y_train,y_predTrain)
print('\nconfusion matrix for train data\n', cm1)
```

```
confusion matrix for test data
 [[66  2]
 [ 8 24]]

confusion matrix for train data
 [[173  16]
 [ 37  74]]
```

```
#Accuracy score
print('Accuracy score for test data\n', accuracy_score(y_test,y_pred))
print('\nAccuracy Score for train data\n', accuracy_score(y_train,y_pre
dTrain))
```

Kawar Nilesh Ramesh
Roll no : 59
Batch A

```
#Accuracy score
print('Accuracy score for test data\n', accuracy_score(y_test,y_pred))
print('\nAccuracy Score for train data\n', accuracy_score(y_train,y_predTrain))
```

```
Accuracy score for test data
 0.9

Accuracy Score for train data
 0.8233333333333334
```

```
#Classification report
print('Classification report for test data \n', classification_report(y
_test,y_pred))
print('\nClassification report for train data\n', classification_report
(y_train,y_predTrain))
```

```
#Classification report
print('Classification report for test data \n', classification_report(y_test,y_pred))
print('\nClassification report for train data\n', classification_report(y_train,y_predTrain))
```

```
Classification report for test data
              precision    recall  f1-score   support

           0       0.89      0.97      0.93        68
           1       0.92      0.75      0.83        32

    accuracy                           0.90       100
   macro avg       0.91      0.86      0.88       100
weighted avg       0.90      0.90      0.90       100


Classification report for train data
              precision    recall  f1-score   support

           0       0.82      0.92      0.87       189
           1       0.82      0.67      0.74       111

    accuracy                           0.82       300
   macro avg       0.82      0.79      0.80       300
weighted avg       0.82      0.82      0.82       300
```