Experiment 4

Student Name:Rajiv Paul **UID: 20BCS1812**
Branch:CSE **Section/Group: 20BCS_WM-702A**
Semester: 5th **Date of Performance: 28/09/2022**
Subject Name: Machine Learning Lab **Subject Code: 20CSP-317**

Aim/Overview of the practical:

Implement Support Vector Machine on any data set and analyze the accuracy with Logistic Regression.

Task to be done:

To implement Support Vector Machine on any data set.

Apparatus/Simulator Used:

- **Google Collab**
- **Python**
- **.csv file**

Code and Output:

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
sns.set_style('whitegrid')
```

```
[9] print(np.__version__)
    print(pd.__version__)
    print(matplotlib.__version__)
    print(sns.__version__)
    print(sklearn.__version__)
```

```
1.21.6
1.3.5
3.2.2
0.11.2
1.0.2
```

```
[10] df=sns.load_dataset('iris')
     df.head()
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
col=['petal_length','petal_width','species']
data=df.loc[:,col]
data.head()
```

	petal_length	petal_width	species
0	1.4	0.2	setosa
1	1.4	0.2	setosa
2	1.3	0.2	setosa
3	1.5	0.2	setosa
4	1.4	0.2	setosa

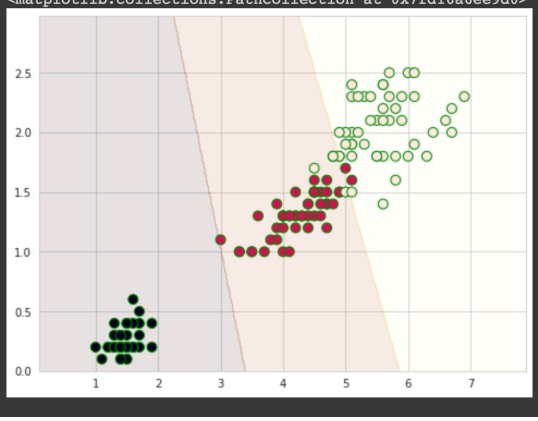
```
[12] df.species.unique()

     array(['setosa', 'versicolor', 'virginica'], dtype=object)

[13] colx=['petal_length','petal_width']
     x=df.loc[:,colx]
     x.head()
```

	petal_length	petal_width
0	1.4	0.2
1	1.4	0.2
2	1.3	0.2
3	1.5	0.2
4	1.4	0.2

```
label_to_num={'setosa' : 0,'versicolor' : 1,'virginica' : 2}
y=df.loc[:,col[2]].map(label_to_num)
y.head()

0    0
1    0
2    0
3    0
4    0
Name: species, dtype: int64
```

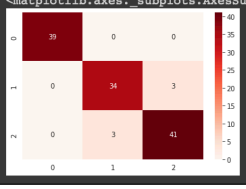
```
[15] from sklearn import svm
     C=0.001
     clf=svm.SVC(kernel='linear',C=C)
     clf.fit(x,y)

     SVC(C=0.001, kernel='linear')

[16] clf.predict([[6,2]])

     /usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feat
       "X does not have valid feature names, but"
     array([2])

     xv=x.values.reshape(-1,1)
     h=0.02
     xmin,xmax=xv.min(),xv.max()+1
     ymin,ymax=y.min(),y.max()+1
     xx,yy=np.meshgrid(np.arange(xmin,xmax,h),np.arange(ymin,ymax,h))
     z=clf.predict(np.c_[xx.ravel(),yy.ravel()])
     z=z.reshape(xx.shape)
     fig=plt.figure(figsize=(8,6))
     ax=plt.contourf(xx,yy,z,cmap='afmhot',alpha=0.1)
     plt.scatter(x.values[:,0],x.values[:,1],c=y,s=80,alpha=1,edgecolors='g')
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feat
  "X does not have valid feature names, but"
<matplotlib.collections.PathCollection at 0x7fd16a6ee9d0>
```




```
# Linear svm using train-test-split
df = sns.load_dataset('iris')
from sklearn.model_selection import train_test_split, cross_val_score, cross_val_predict
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score
from sklearn.preprocessing import StandardScaler
colx = ['petal_length','petal_width']
x = df.loc[:,colx] # get independent variable
label_to_num = {'setosa' : 0,
                'versicolor' : 1,
'virginica' : 2}
y = df.loc[:,col[2]].map(label_to_num) # get dependent variable
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size = 0.8,random_state = 0) # preform train_test_split

sc = StandardScaler() # scale the x_train and x_test
x_std_train = sc.fit_transform(x_train)

x_std_test = sc.fit_transform(x_test)

C = 1.0 # set hyperparameter 'C'

lModel = svm.SVC(kernel = 'linear', C = C) # instantiate model with kernel = 'linear' lModel.fit(x_std_train,y_train) # fit the model

# cross validation to get avg accuracy and std
score = cross_val_score(lModel,x_std_train,y_train,cv =10, scoring = 'accuracy' )

print("avg accuracy:\t{0:,.4f}".format(np.mean(score)))

print("avg std:\t{0:,.4f}".format(np.std(score)))

y_pred = cross_val_predict(lModel,x_std_train,y_train,cv =3) # get prediction values for train data

confusion_matrix(y_train,y_pred )# get CM for train data

sns.heatmap(confusion_matrix(y_train,y_pred),annot = True,cmap = 'Reds') # plot CM fo r train data
```

```
avg accuracy:   0.9500
avg std:        0.0553
<matplotlib.axes._subplots.AxesSubplot at 0x7fd167cd72d0>
```



Learning outcomes (What I have learnt):

1. **Learnt how to implement linear regression**
2. **Learnt about numpy, seaborn, pandas libraries.**
3. **Learnt how to use python for linear regression.**
4. **Learnt how to implement Support Vector Machine.**

Evaluation Grid :

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Student Performance (Conduct of experiment) objectives/Outcomes.		12
2.	Viva Voce		10
3.	Submission of Work Sheet (Record)		8
	Total		30