Dikshita Kambri

118A2044

BE EXTC A2

**EXPERIMENT 5**

# SVM (SUPPORT VECTOR MACHINE)

# EXPERIMENT 5

## SVM (SUPPORT VECTOR MACHINE)

**AIM:** Implement support vector machine

**APPARATUS:** python

**THEORY:**

**Support vector machine**

Support vector machines so called as SVM is a ***supervised learning algorithm*** which can be used for classification and regression problems as support vector classification (SVC) and support vector regression (SVR).
SVM is based on the idea of finding a hyperplane that best separates the features into different domains.
**Terminologies used in SVM:**

The points closest to the hyperplane are called as the ***support vector points*** and the distance of the vectors from the hyperplane are called the ***margins***.

The basic intuition to develop over here is that more the farther SV points, from the hyperplane, more is the probability of correctly classifying the points in their respective region or classes. SV points are very critical in determining the hyperplane because if the position of the vectors changes the hyperplane's position is altered. Technically this hyperplane can also be called as ***margin maximizing hyperplane***.

**Pros and cons of SVM:**

**Pros:**

1. It is really effective in the higher dimension.
2. Effective when the number of features are more than training examples.
3. Best algorithm when classes are separable
4. The hyperplane is affected by only the support vectors thus outliers have less impact.
5. SVM is suited for extreme case binary classification.

**cons:**

1. For larger dataset, it requires a large amount of time to process.

2. Does not perform well in case of overlapped classes.
3. Selecting, appropriately hyperparameters of the SVM that will allow for sufficient generalization performance.
4. Selecting the appropriate kernel function can be tricky.

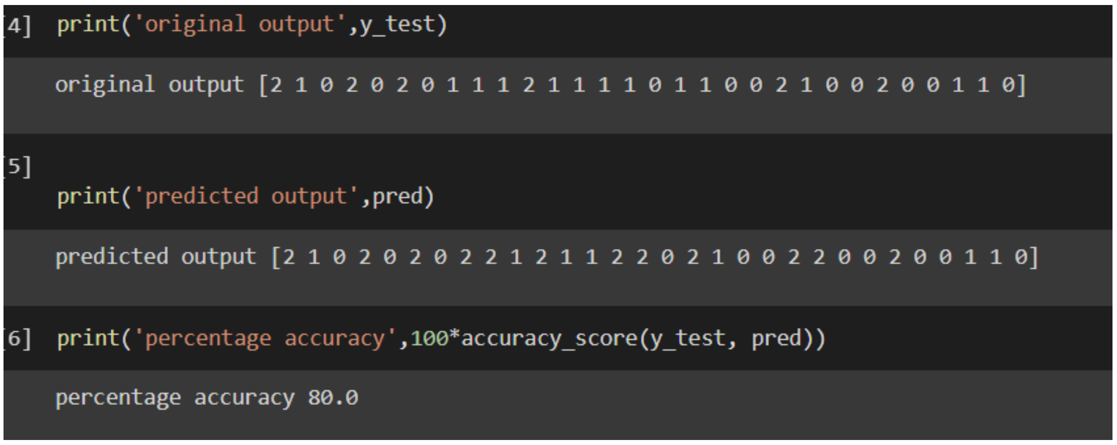
**OUTPUT:**

```
from sklearn import datasets
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score

iris = datasets.load_iris()
x = iris.data
y = iris.target
print(x)

print(y)

X_train, X_test, y_train, y_test = train_test_split(x,y,test_size
 = 0.2, random_state = 0)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
cls = svm.SVC()
cls.fit(X_train, y_train)
pred = cls.predict(X_test)
print('original output',y_test)
print('predicted output',pred)
print('percentage accuracy',100*accuracy_score(y_test, pred))
```



```
[4]  print('original output',y_test)

     original output [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0]

[5]
     print('predicted output',pred)

     predicted output [2 1 0 2 0 2 0 2 2 1 2 1 1 2 2 0 2 1 0 0 2 2 0 0 2 0 0 1 1 0]

[6]  print('percentage accuracy',100*accuracy_score(y_test, pred))

     percentage accuracy 80.0
```

**CONCLUSION:**

Studied that SVM is a learning algorithm for classification. We implemented a support vector machine with test size of 0.2 for Iris dataset. And observed that the percentage accuracy is 80.