# Experiment 5

**Student Name:Upwan** **UID: 21BCS8105**
**Branch:CSE** **Section/Group: 20BCS_WM-702A**
**Semester: 5th** **Date of Performance: 12/10/2022**
**Subject Name: Machine Learning Lab** **Subject Code: 20CSP-317**

**Aim/Overview of the practical:**

**Implement Naïve Bayes on any dataset.**

**Task to be done:**

**To implement Naïve Bayes on any data set.**

**Apparatus/Simulator Used:**

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

## Code and Output:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

# importing the dataset

dataset = pd.read_csv ('NaiveBayes.csv')

# split the data into inputs and outputs
X = dataset.iloc[:, [0,1]].values
y = dataset.iloc[:, 2].values
# training and testing data
from sklearn.model_selection import train_test_split

# assign test data size 25%
X_train, X_test, y_train, y_test =train_test_split(X,y,test_size= 0.25, random_state=0)
# importing standard scaler
from sklearn.preprocessing import StandardScaler
```

```
# scalling the input data
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
# importing classifier
from sklearn.naive_bayes import BernoulliNB

# initializaing the NB
classifer = BernoulliNB()

# training the model
classifer.fit(X_train, y_train)

# testing the model
y_pred = classifer.predict(X_test)
# importing accuracy score
from sklearn.metrics import accuracy_score

# printing the accuracy of the model
print(accuracy_score(y_pred, y_test))
```

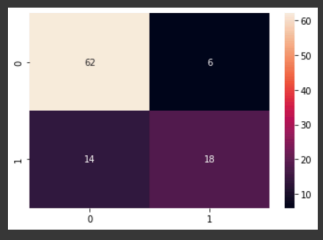
```
0.8
```

```python
# import Gaussian Naive Bayes classifier
from sklearn.naive_bayes import GaussianNB

# create a Gaussian Classifier
classifer1 = GaussianNB()

# training the model
classifer1.fit(X_train, y_train)

# testing the model
y_pred1 = classifer1.predict(X_test)
# importing accuracy score
from sklearn.metrics import accuracy_score

# printing the accuracy of the model
print(accuracy_score(y_test,y_pred1))
```

```
0.91
```

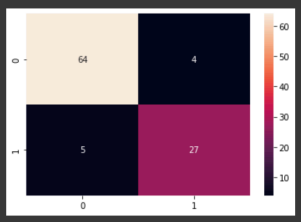
```python
# importing the required modules
import seaborn as sns
from sklearn.metrics import confusion_matrix

# passing actual and predicted values
cm = confusion_matrix(y_test, y_pred)

# true write data values in each cell of the matrix
sns.heatmap(cm, annot=True)
plt.savefig('confusion.png')

# importing classification report
from sklearn.metrics import classification_report

# printing the report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.91      0.86        68
           1       0.75      0.56      0.64        32

    accuracy                           0.80       100
   macro avg       0.78      0.74      0.75       100
weighted avg       0.79      0.80      0.79       100
```



```
# importing the required modules
import seaborn as sns
from sklearn.metrics import confusion_matrix

# passing actual and predicted values
cm = confusion_matrix(y_test, y_pred1)

# true write data values in each cell of the matrix
sns.heatmap(cm,annot=True)
plt.savefig('confusion.png')
# importing classification report
from sklearn.metrics import classification_report
# printing the report
print(classification_report(y_test, y_pred1))
```

```
              precision    recall  f1-score   support

           0       0.93      0.94      0.93        68
           1       0.87      0.84      0.86        32

    accuracy                           0.91       100
   macro avg       0.90      0.89      0.90       100
weighted avg       0.91      0.91      0.91       100
```




**Learning outcomes (What I have learnt):**

1. **Learnt how to implement Naïve Bayes**
2. **Learnt about numpy, seaborn, pandas libraries.**
3. **Learnt how to analyse in Naïve Bayes.**

**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 |