Experiment No- 2.2

Student Name: Yash Kumar **UID: 20BCS9256**
Branch: CSE **Section/Group: 616- B**
Semester: 5 **Date of Performance: 11-10-22**
Subject Name: Machine Learning **Subject Code: 20CSP -317**

1. **Aim/Overview of the practical:** Implementing Naive Bayes' Classifier on any dataset and analyse the accuracy.

2. Task to be done/ Which logistics used: Analysing accuracy by implementing the Naive Bayes' Classifier on any dataset.

3. **Steps of experiment/Code:**

    1. **Importing libraries such as matplotlib, pandas and seaborn and reading the dataset.**

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

In [2]:
```
dataset = pd.read_csv('NaiveBayes.csv')

# split the data into inputs and outputs
X = dataset.iloc[:, [0,1]].values
y = dataset.iloc[:, 2].values
```

2. Preprocessing the data for training the model and defining the model:

```
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)
```

```
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)
```

3. Implementing Naïve Bayes' Classifier using Bernoulli and calculating the accuracy.

```
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)
```

```
from sklearn.metrics import accuracy_score

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

```
0.8
```

4. Implementing Naïve Bayes' Classifier using Gaussian and calculating the accuracy

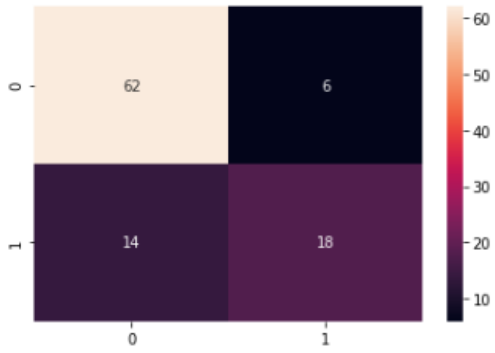
```
# import Gaussian Naive Bayes classifier
from sklearn.naive_bayes import GaussianNB

# create a Gaussian Classifier
classifer1 = GaussianNB()

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

# testing the model
y_pred1 = classifer1.predict(X_test)
```

```
from sklearn.metrics import accuracy_score

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

```
0.91
```

5. Displaying Confusion Matrices and Classification Reports:

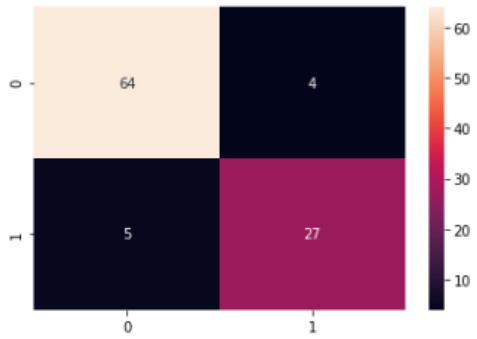
```
# importing the required modules
import seaborn as sns
from sklearn.metrics import confusion_matrix

# 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')
```



```
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
```

```
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')
```




```
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
```

6. Implementing Naive Bayes' Classifier for label encoding:

```
# assigning features and label variables
weather = ['Sunny','Sunny','Overcast','Rainy','Rainy','Rainy','Overcast','Sunny','Sunny', 'Rainy','Sunny','Overcast','Overcast',

# output class
play = ['No','No','Yes','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Yes','No']
```

```
from sklearn import preprocessing

# creating LabelEncoder
labelCode = preprocessing.LabelEncoder()

# Converting string labels into numbers.
weather_encoded=labelCode.fit_transform(weather)
```

```
print(weather_encoded)
```

```
[2 2 0 1 1 1 0 2 2 1 2 0 0 1]
```

```
from sklearn import preprocessing

# creating LabelEncoder
labelCode = preprocessing.LabelEncoder()

# converting string labels into numbers.
label=labelCode.fit_transform(play)
```

```
from sklearn.naive_bayes import GaussianNB

# create a Gaussian Classifier
model = GaussianNB()

# train the model using the training sets
model.fit(weather_encoded, label)
```

7. Getting Error as we need to reshape the given data:

```
from sklearn.naive_bayes import GaussianNB

# create a Gaussian Classifier
model = GaussianNB()

# train the model using the training sets
model.fit(weather_encoded, label)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-21-d1010b7d5489> in <module>
      5
      6 # train the model using the training sets
----> 7 model.fit(weather_encoded, label)

D:\Anaconda3\lib\site-packages\sklearn\naive_bayes.py in fit(self, X, y, sample_weight)
    208         self : object
    209         """
--> 210         X, y = self._validate_data(X, y)
    211         y = column_or_1d(y, warn=True)
    212         return self._partial_fit(X, y, np.unique(y), _refit=True,

D:\Anaconda3\lib\site-packages\sklearn\base.py in _validate_data(self, X, y, reset, valic
    430                 y = check_array(y, **check_y_params)
    431             else:
--> 432                 X, y = check_X_y(X, y, **check_params)
    433             out = X, y
    434

D:\Anaconda3\lib\site-packages\sklearn\utils\validation.py in inner_f(*args, **kwargs)
     70                           FutureWarning)
     71         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 72         return f(**kwargs)
     73     return inner_f
     74

D:\Anaconda3\lib\site-packages\sklearn\utils\validation.py in check_X_y(X, y, accept_spar
copy, force_all_finite, ensure_2d, allow_nd, multi_output, ensure_min_samples, ensure_mir
    793         raise ValueError("y cannot be None")
    794
```

8. Implementing Naive Bayes' Classifier and predicting the output:

```
import numpy as np

# converting 1D array to 2D
weather_2d = np.reshape(weather_encoded, (-1, 1))
```

```
from sklearn.naive_bayes import GaussianNB

# create a Gaussian Classifier
model = GaussianNB()

# train the model using the training sets
model.fit(weather_2d, label)
```

```
GaussianNB()
```

```
predicted= model.predict([[0]]) # 0:Overcast

# printing predicted value
print(predicted)
```

```
[1]
```

Learning Outcomes (What I have learnt):

1. I have learnt about implementing Naïve Bayes' Classifier for Bernoulli and Gaussian.

2. I have learnt about constructing confusion matrix.

3. I have learnt about various libraries which are supported by python such as sklearn, numpy, matplotlib.

4. I have learnt about the various functions provided by various libraries.

5. I have understood the experiment very well.

Evaluation Grid:

	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