# INTELLIGENT SYSTEMS LAB-6 (27/09/2021)

*NAME*: *Harshvardhan Agarwal* | ***SEC***: *A* | ***REG NO.***: *201800524*

## PROBLEM STATEMENT

Write a program to implement the naïve Bayesian classifier for a sample training data set stored as

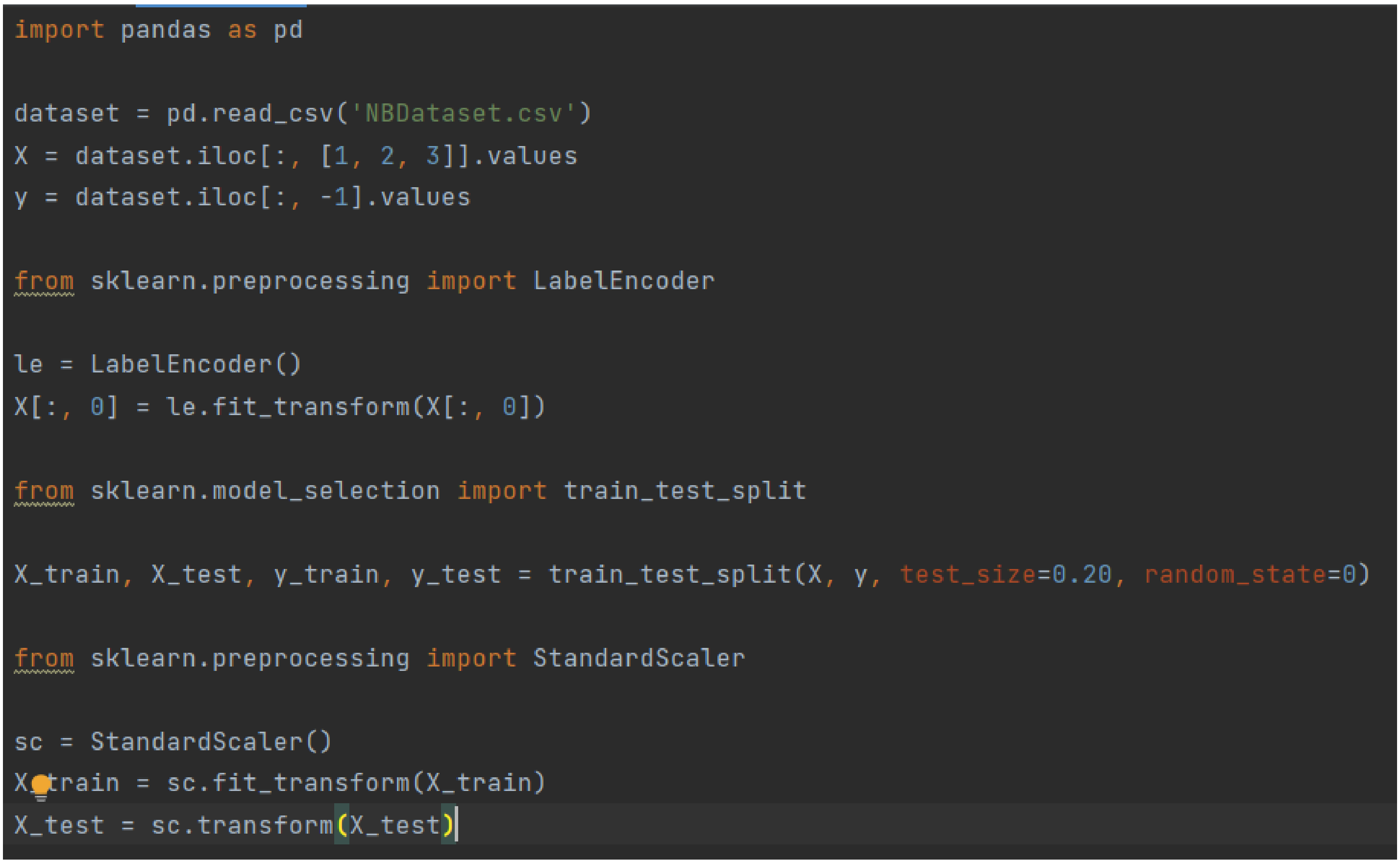
a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

1. Submit the Pdf file with code and graphs.
2. Perform the classifier on at least two (2) Dataset.
3. Perform the experiment with different test analysis.

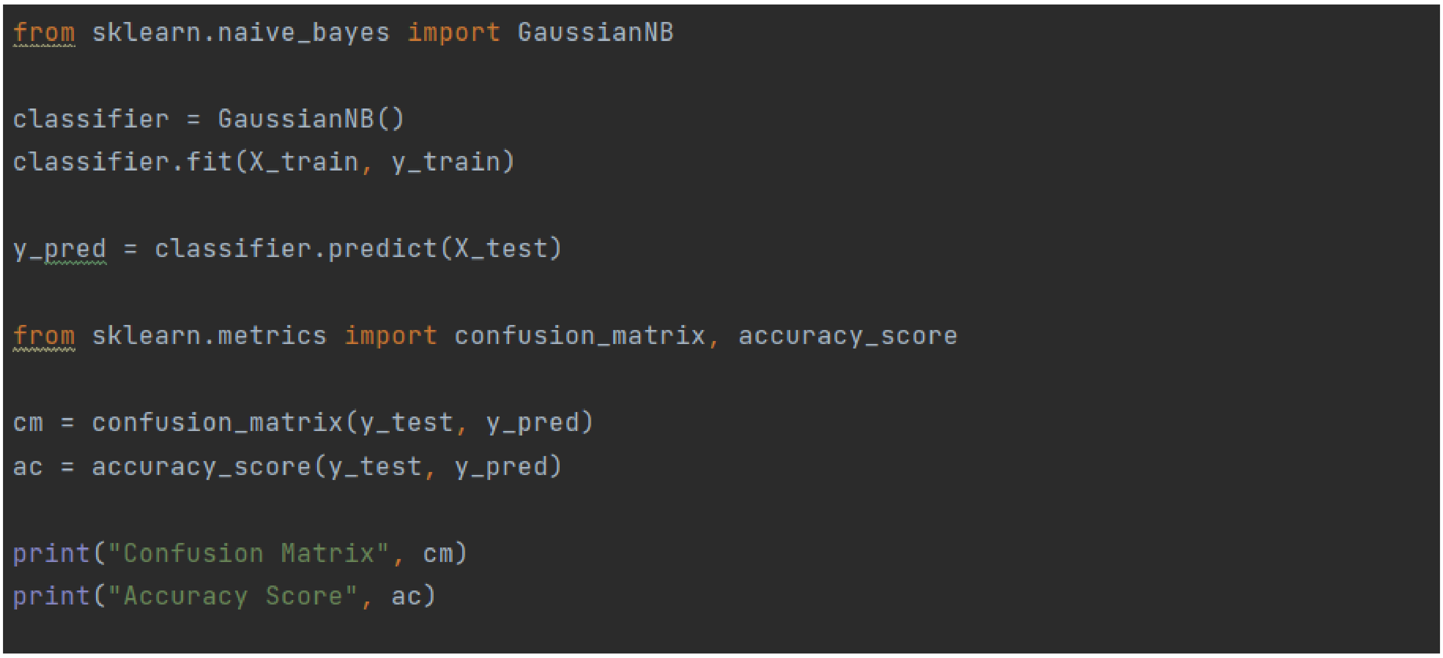
## PROBLEM SOLUTION

## SOURCE CODE

1. Using IRIS dataset and Confusion Matrix and Accuracy as the testing measure

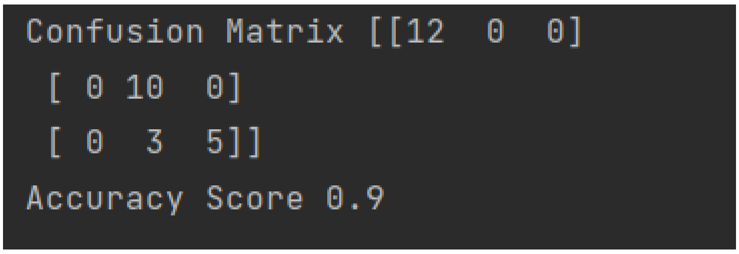


```
import pandas as pd

dataset = pd.read_csv('NBDataset.csv')
X = dataset.iloc[:, [1, 2, 3]].values
y = dataset.iloc[:, -1].values

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
X[:, 0] = le.fit_transform(X[:, 0])

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```



```python
from sklearn.naive_bayes import GaussianNB

classifier = GaussianNB()
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

from sklearn.metrics import confusion_matrix, accuracy_score

cm = confusion_matrix(y_test, y_pred)
ac = accuracy_score(y_test, y_pred)

print("Confusion Matrix", cm)
print("Accuracy Score", ac)
```

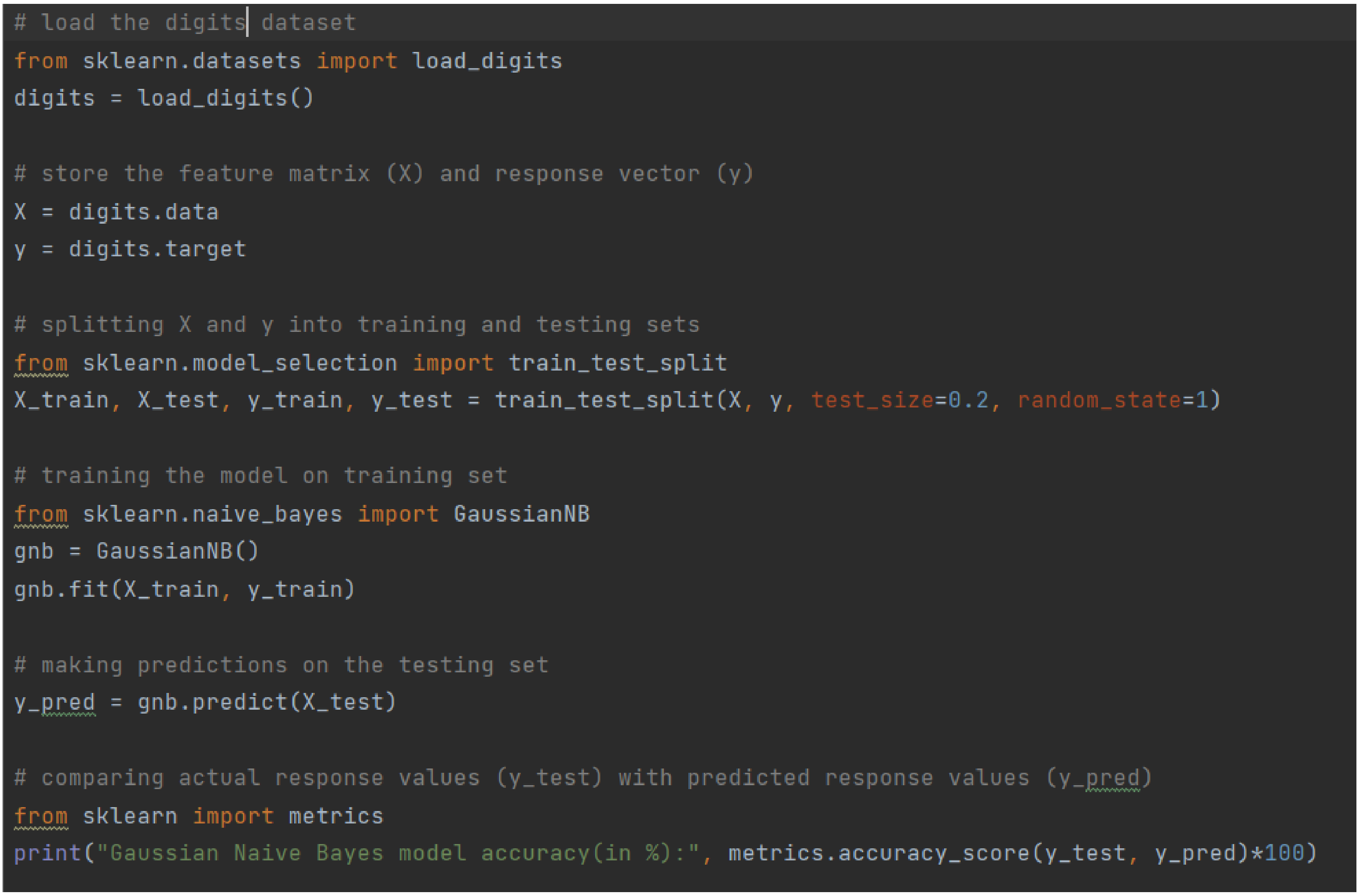
## OUTPUT



```
Confusion Matrix [[12  0  0]
 [ 0 10  0]
 [ 0  3  5]]
Accuracy Score 0.9
```

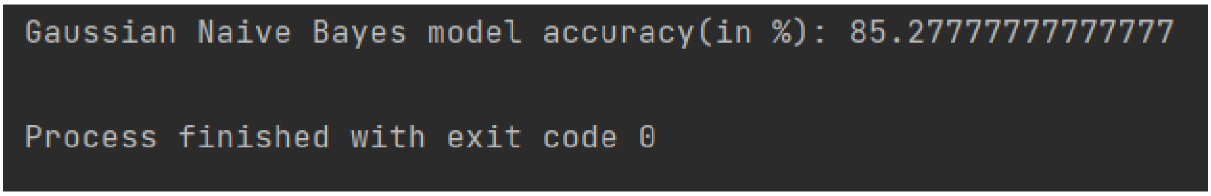
## SOURCE CODE

2. Using MNIST dataset



```
# load the digits dataset
from sklearn.datasets import load_digits
digits = load_digits()

# store the feature matrix (X) and response vector (y)
X = digits.data
y = digits.target

# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

# training the model on training set
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)

# making predictions on the testing set
y_pred = gnb.predict(X_test)

# comparing actual response values (y_test) with predicted response values (y_pred)
from sklearn import metrics
print("Gaussian Naive Bayes model accuracy(in %):", metrics.accuracy_score(y_test, y_pred)*100)
```

## OUTPUT



```
Gaussian Naive Bayes model accuracy(in %): 85.27777777777777


Process finished with exit code 0
```