APPLIED MACHINE LEARNING

Assignment 2

Names :
Phoebe Thabit Wadea
Sandy Adel Latef
Muhammad Mohsen Muhammad

Part 1:

1.

P(stolen=Yes) = 6/14 **P(stolen=No)** = 8/14

Feature analysis:

1. For color feature:

Color	Yes	No
Red	3/6	4/8
Yellow	2/6	2/8
Blue	1/6	2/8

---

2. For type feature:

Type	Yes	No
Sports	4/6	3/8
SUV	2/6	5/8

---

3. For origin feature:

Type	Yes	No
Domestic	2/6	5/8
Imported	4/6	3/8

Calculating the evidence:

x = (Blue, SUV, Domestic) = P(Blue, SUV, Domestic|Yes)*P(Yes)+ P(Blue, SUV, Domestic|No)*P(No)=. 0079+.056=.0639

Calculating posteriors for each class :

P(Yes|x) = [[P(Blue|Yes)*P(SUV|Yes)*P(Domestic|Yes)]*P(Yes)]

P(Yes|x) = [(1/6)*(2/6)*(2/6)]*(6/14)=.0079/ P(Blue, SUV, Domestic)=.0079/.0639= 0.12

P(No|x) = [P(Blue| No)*P(SUV| No)*P(Domestic| No)]*P(No)

P(No|x) = [(2/8)*(5/8)*(5/8)]*(8/14) = .056 /P(Blue, SUV, Domestic)=.056/.0639= 0.88

Classification Decision:

Comparing the probabilities, we find that **P(No|x)** > **P(Yes|x)**. Therefore, we label the new instance (x) as **"No"** indicating that it is not likely to be stolen.

2.

R(a1|x) = ʎ11.P(C1|x) + ʎ12.P(C2|x)

R(a1|x) = 0.P(C1|x) + 6.P(C2|x)

= 6.(1-P(C1|x)).

R(a2|x) = ʎ21.P(C1|x) + ʎ22.P(C2|x)

R(21|x) = 3.P(C1|x) + 0.P(C2|x)

= 3.P(C1|x).

R(ar|x) = 2.

We choose a1 if :

R(a1|x) < 2 ➔ 3.P(C1|x) < 2 ➔ P(C1|x) < 2/3.

We choose a2 if :

R(a2|x) < 2 ➔ 6-6.P(C1|x) < 2 ➔ -6.P(C1|x) < -4 ➔ P(C1|x) > 2/3.

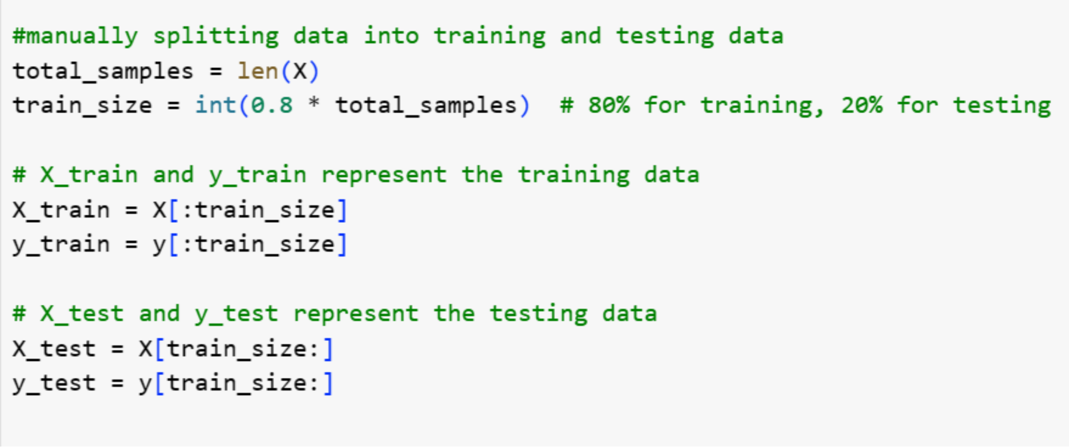
So , We reject if :

2/3 < R(a1|x) < 2/3.

Part 2:

a)

We used this function to split data into training data (80% of the data),
And testing data(20% of the data), using these lines of code:



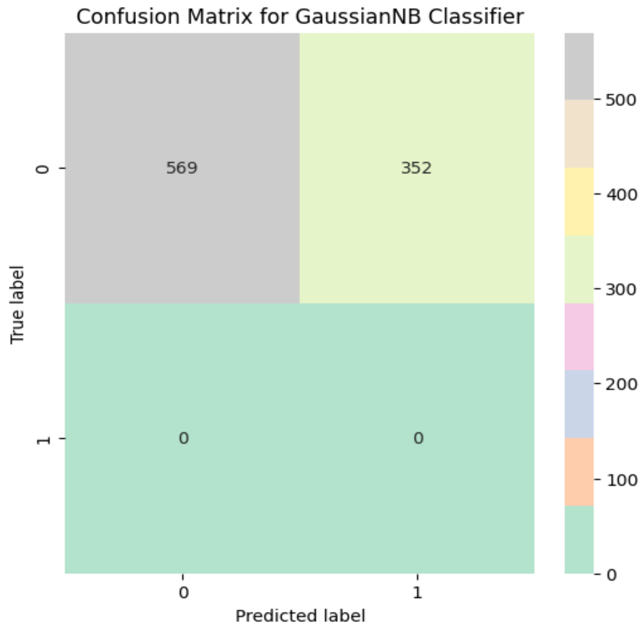
```
#manually splitting data into training and testing data
total_samples = len(X)
train_size = int(0.8 * total_samples)  # 80% for training, 20% for testing

# X_train and y_train represent the training data
X_train = X[:train_size]
y_train = y[:train_size]

# X_test and y_test represent the testing data
X_test = X[train_size:]
y_test = y[train_size:]
```

Naïve bayes classifiers:

Then we trained the training data on two Naïve bayes classifiers (Guassian,Multinominal) , and compute the accuracy and confusion matrix for each. **There are the results:**

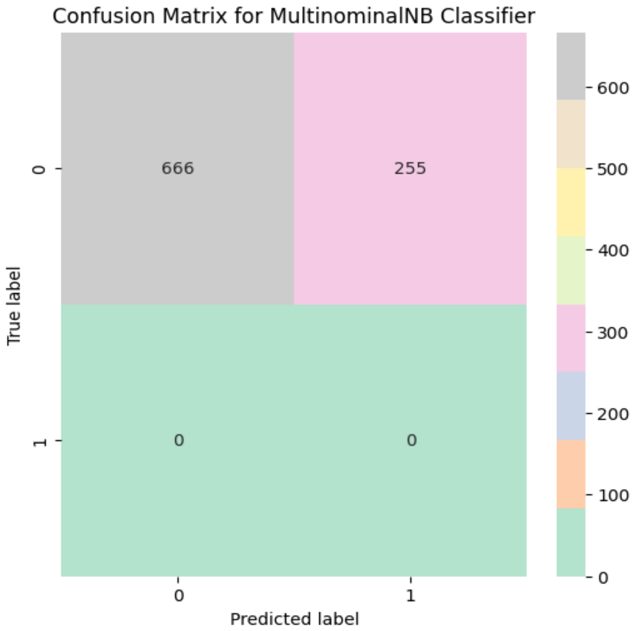
- **Guassian naïve bayes classifier results :**

```
accuracy for gaussian classifier:0.6178067318132465
```



- **Multinominal naïve bayes classifier results:**

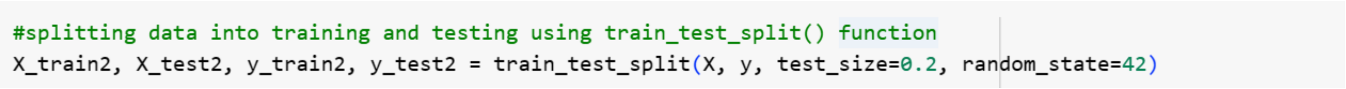
```
accuracy for multinominal classifier: 0.7231270358306189
```



---

b)

In this point ,we Splitted data into training data (80% of the data),And testing data(20% of the data) using **train_test_split()** function , from **sklearn.model_selection** module .



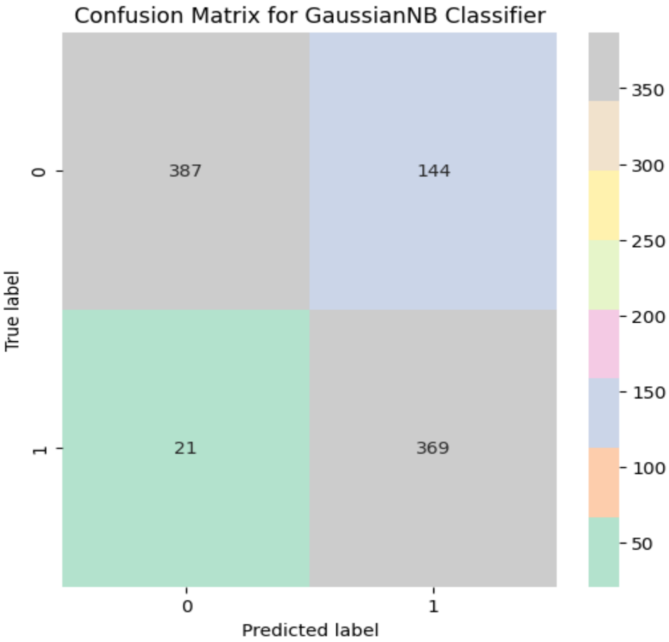
```
#splitting data into training and testing using train_test_split() function
X_train2, X_test2, y_train2, y_test2 = train_test_split(X, y, test_size=0.2, random_state=42)
```

The train_test_split function typically includes a randomization step when splitting the data. It shuffles the data before splitting it into training and testing sets. This randomization helps to ensure that the data is representative and avoids any potential bias in the ordering of the samples , so it suppose to give better results , as shown below :

We trained the same two classifiers as the previous point , and these are results :

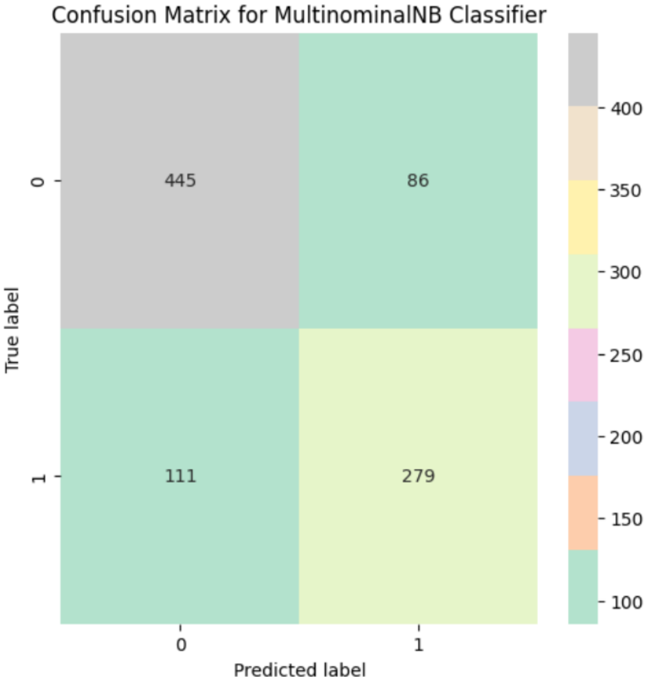
- **Results for Guassian naïve bayes classifier:**

```
accuracy for gaussian classifier: 0.8208469055374593
```



- **Results for Multinominal naïve bayes classifier:**

```
accuracy for multinominal classifier: 0.7861020629750272
```



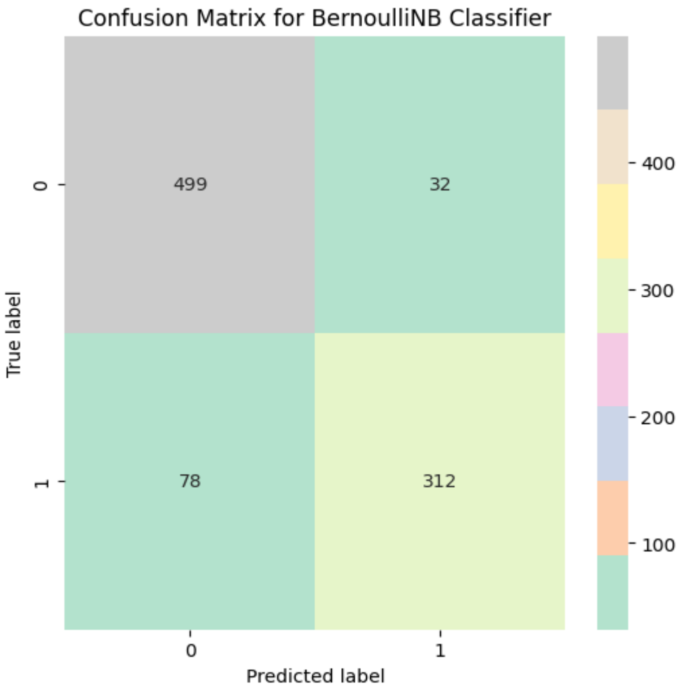
c)

Bernolli naïve bayes classifier:

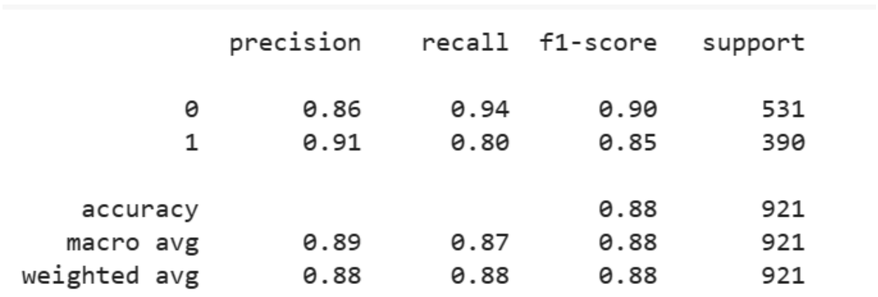
- is a probabilistic machine learning model that belongs to the family of naive Bayes classifiers. It is specifically designed for binary classification problems, the Bernoulli naive Bayes classifier is a popular choice for text classification tasks, such as sentiment analysis or spam detection, so in this data it gives the highest accuracy.

Bernolli results:

```
accuracy for Bernoulli classifier: 0.8805646036916395
```



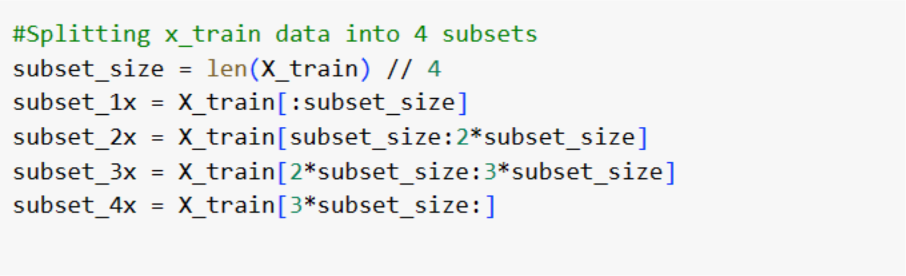
Classification report :



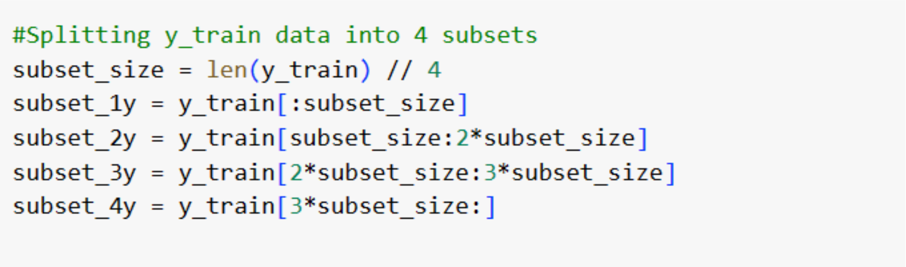
```
              precision    recall  f1-score   support

           0       0.86      0.94      0.90       531
           1       0.91      0.80      0.85       390

    accuracy                           0.88       921
   macro avg       0.89      0.87      0.88       921
weighted avg       0.88      0.88      0.88       921
```

d) Comparing subsets :

Splitting data into 4 subsets (25% for each subset):



```
#Splitting x_train data into 4 subsets
subset_size = len(X_train) // 4
subset_1x = X_train[:subset_size]
subset_2x = X_train[subset_size:2*subset_size]
subset_3x = X_train[2*subset_size:3*subset_size]
subset_4x = X_train[3*subset_size:]
```



```
#Splitting y_train data into 4 subsets
subset_size = len(y_train) // 4
subset_1y = y_train[:subset_size]
subset_2y = y_train[subset_size:2*subset_size]
subset_3y = y_train[2*subset_size:3*subset_size]
subset_4y = y_train[3*subset_size:]
```

Then comparing subsets with test data in the **point (a) ,** these are the results:

Accuracies for each subset:

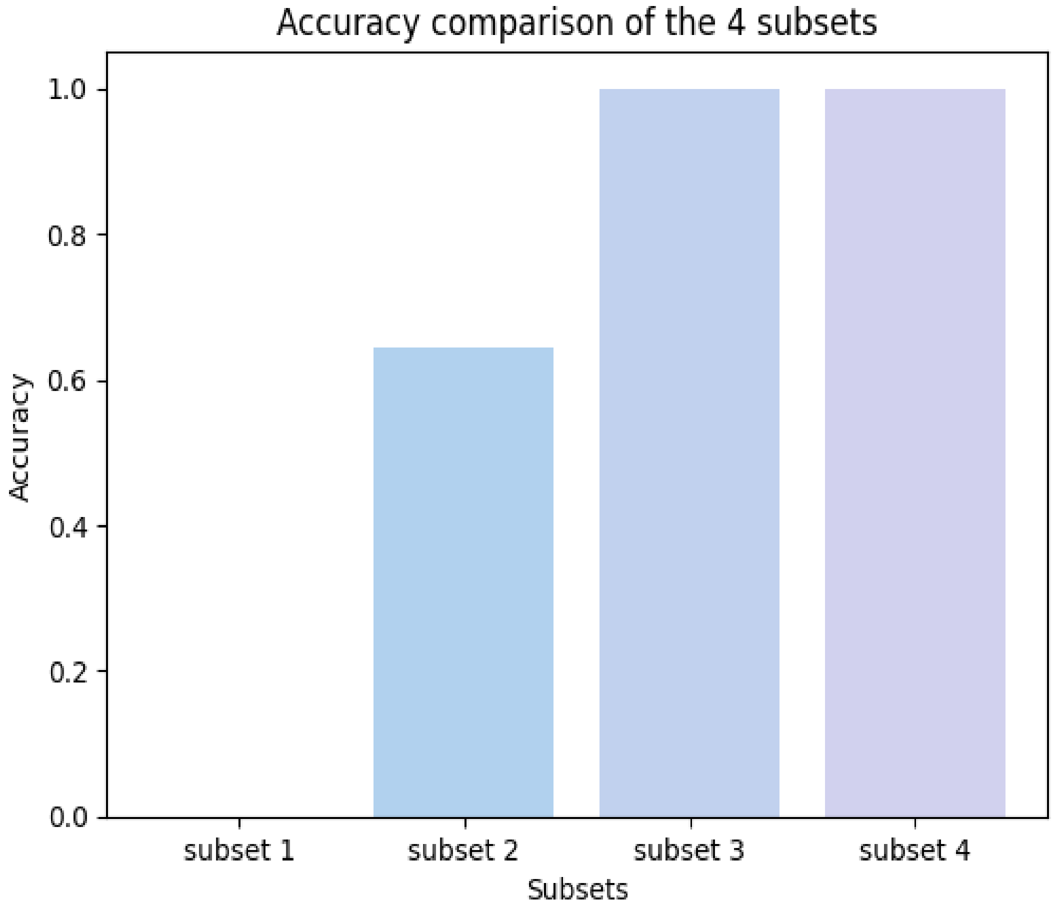
- Accuracy for subset1: 0.0 .
- Accuracy for subset2: 0.6438653637350705 .
- Accuracy for subset3: 1.0 .
- Accuracy for subset4: 1.0 ,

Subset_1 get accuracy of **"0"** , may be bacuase of Insufficient Training: The classifier may not have been adequately trained or trained on an insufficient amount of data. This may prevent the classifier from learning the underlying patterns and making accurate predictions.

Subset_2 has a good accuracy which means it trained on sufficient training data and raetreive a good accuracy.

Subset_3 , Subset_4 have accuracy of **"1",** which means the classifier is able to classify every instance correctly.,The training data used to train the classifier may be of high quality, free from errors, and representative of the true population. This ensures that the classifier can learn effectively and make accurate predictions.

Bar chart for comparing accuracies :


Accuracy comparison of the 4 subsets
Accuracy
subset 1
subset 2
subset 3
subset 4
Subsets