ELG 20225

APPLIED MACHINE LEARNING

Assignment 2

GROUP-4 MEMBERS:

- AMIRA ABU ISSA
- HEBA MOSTAFA
- AYA METWALLY

Part 1:

(1)

Example No.	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Blue	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Blue	Sports	Imported	Yes
11	Red	SUV	Domestic	No
12	Red	SUV	Domestic	No
13	Blue	Sports	Imported	No
14	Red	SUV	Imported	Yes

We have two classes C1=(stolen=Yes) ,C2=(stolen=No)

And we try to predict New Instance = (Blue, SUV, Domestic) into (Yes or No)

To calculate posterior probability of 2 classes

Posterior = (likelihood) (prior) / (evidence)

We must calculate prior, likelihood and evidence

Firstly, we calculate prior **:** the probability of each class

P(stolen=Yes) =Number of stolen cars /Total number of cars

$$= \frac{6}{14} = \frac{3}{7}$$

P(stolen=No) =Number of non-stolen cars /Total number of cars

$$= \frac{8}{14} = \frac{4}{7}$$

Another way P(stolen=Yes)+ P(stolen=No)=1

P(stolen=No)=$1-\frac{3}{7} = \frac{4}{7}$

Secondly, calculate likelihood: the probability of each factor (Blue, SUV, Domestic)

According 2 classes

P(Blue |stolen=Yes)= Number of Blue stolen cars/Number of cars stolen $= \frac{1}{6}$

P(Blue |stolen=No)= Number of Blue non-stolen cars/Number of cars stolen $= \frac{2}{8} = \frac{1}{4}$

P(SUV |stolen=Yes)= Number of SUV stolen cars/Number of cars stolen $= \frac{2}{6} = \frac{1}{3}$

P(SUV |stolen=No)= Number of SUV non-stolen cars/Number of cars stolen $= \frac{5}{8}$

P(Domestic |stolen=Yes)= Number of Domestic stolen cars/Number of cars stolen $= \frac{2}{6} = \frac{1}{3}$

P(Domestic |stolen=No)= Number of Domestic non-stolen cars/Number of cars stolen $= \frac{5}{8}$

Thirdly, we calculate the evidence : is the marginal probability that an observation x is seen

P(Blue,SUV,Domestic)=

P(Blue| stolen=Yes) * P(SUV|stolen=Yes)* P(Domestic|stolen=Yes)*P(stolen=Yes)

+ P(Blue |stolen=No) * P(SUV|stolen=No)* P(Domestic|stolen=No)*P(stolen=No)

$=(\frac{1}{6} * \frac{1}{3} * \frac{1}{3} * \frac{3}{7}) + (\frac{1}{4} * \frac{5}{8} * \frac{5}{8} * \frac{4}{7}) = \frac{257}{4032}$

Finally, we calculate posterior probabilities of 2 classes

P(stolen=Yes|Blue,SUV,Domestic) =

(P(Blue| stolen=Yes) * P(SUV|stolen=Yes)* P(Domestic|stolen=Yes)*P(stolen=Yes))/ P(Blue,SUV,Domestic)

$= (\frac{1}{6} * \frac{1}{3} * \frac{1}{3} * \frac{3}{7}) / \frac{257}{4032} = 0.125$

P(stolen=No|Blue,SUV,Domestic)=

P(Blue |stolen=No) * P(SUV|stolen=No)* P(Domestic|stolen=No)*P(stolen=No)/ P(Blue,SUV,Domestic)=

$(\frac{1}{4} * \frac{5}{8} * \frac{5}{8} * \frac{4}{7}) / \frac{257}{4032} = 0.875$

The Naïve Bayes classifier predicted that the new instance (Blue, SUV, Domestic) more likely to be non-stolen car with a probability 0.875

(2)

Target	Class 1	Class 2
a1 (Choose Class1)	0	6
a2 (Choose Class 2)	3	0
a3 (Rejection)	2	2

Firstly, we must calculate the expected risk of each class

$R(a_1|x) = \lambda_{11} * P(class_1|x) + \lambda_{12} * P(class_2|x) = 0* P(class_1|x) +6 *(1- P(class_1|x))$

$= 6 *(1- P(class_1|x))$

$R(a_2|x)= \lambda_{21} * P(class_1|x) + \lambda_{22} * P(class_2|x) =3* P(class_1|x) +0*(1- P(class_1|x))$

$=3* P(class_1|x)$

$R(a_3|x)= \lambda_{31} * P(class_1|x) + \lambda_{32} * P(class_2|x) = 2(P(class_1|x)+ P(class_2|x)) =2$

Such that $P(class_1|x)+ P(class_2|x)=1$

Secondly, we determine the Rejection area of $P(class_1|x)$

We choose a_1 if :

$R(a1|x) < 2$

$6-6* P(class_1|x) < 2$

$P(class_1|x) > \frac{2}{3}$

We choose a_2 if :

$R(a_2|x) < 2$

$3* P(class_1|x) < 2$

$P(class_1|x) < \frac{2}{3}$

We Reject if $\frac{2}{3} < P(class_1|x) < \frac{2}{3}$

So, we don't have Rejection Area

Part 2:

(1)

- Load the dataset and display the first 5 rows:

	make	address	all	3d	our	over	remove	internet	order	mail	...	char_semicolon	char_parenthesis	char_bracket	char_exclamation	char_dollar	char
0	0.00	0.64	0.64	0.0	0.32	0.00	0.00	0.00	0.00	0.00	...	0.00	0.000	0.0	0.778	0.000	
1	0.21	0.28	0.50	0.0	0.14	0.28	0.21	0.07	0.00	0.94	...	0.00	0.132	0.0	0.372	0.180	
2	0.06	0.00	0.71	0.0	1.23	0.19	0.19	0.12	0.64	0.25	...	0.01	0.143	0.0	0.276	0.184	
3	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.137	0.0	0.137	0.000	
4	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.135	0.0	0.135	0.000	

5 rows × 58 columns

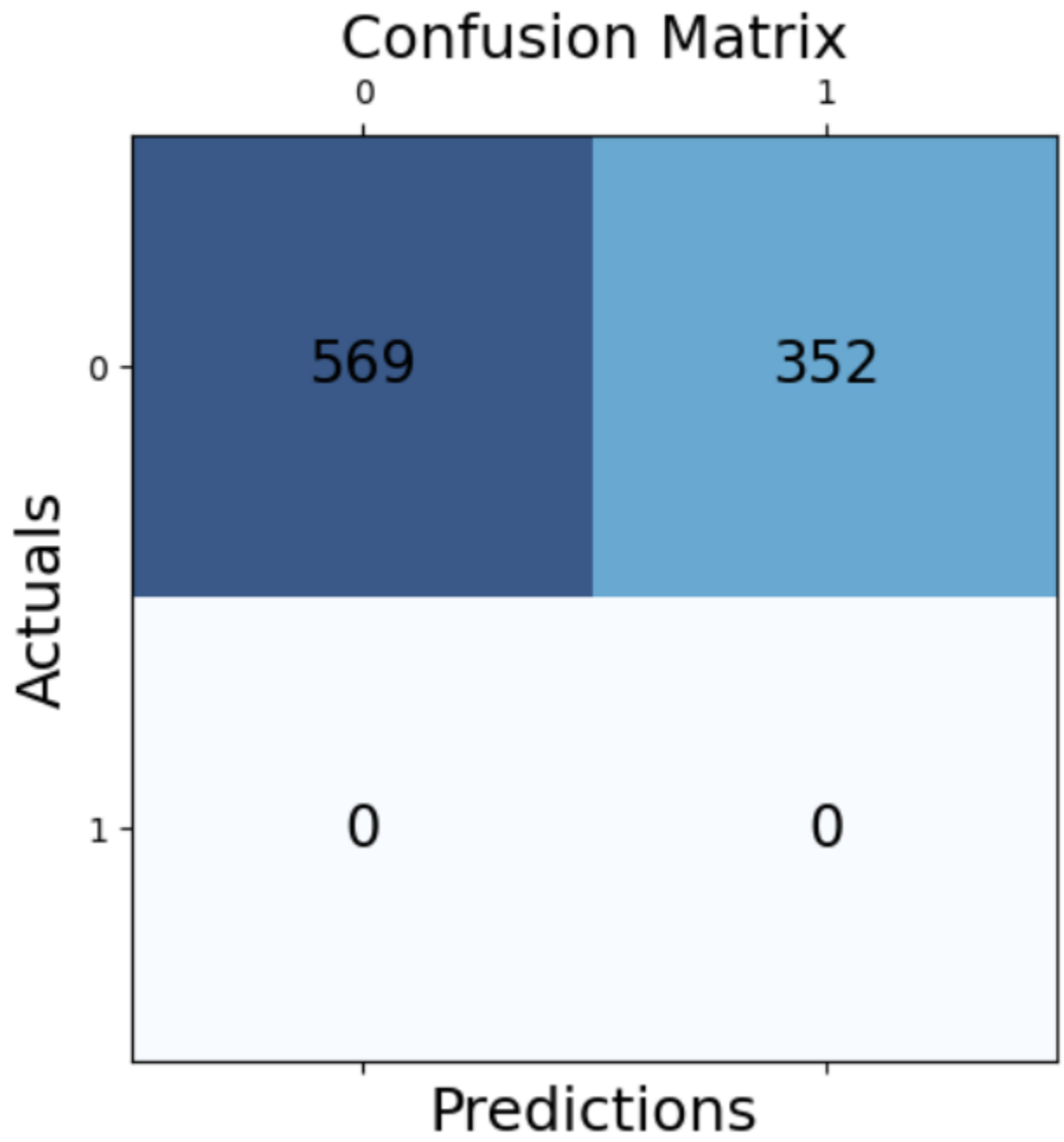
(a)

- Verifying the splitting manually process:

```
Dataset shape (4601, 58)
Training set shape (3680, 58)
Testing set shape (921, 58)
```

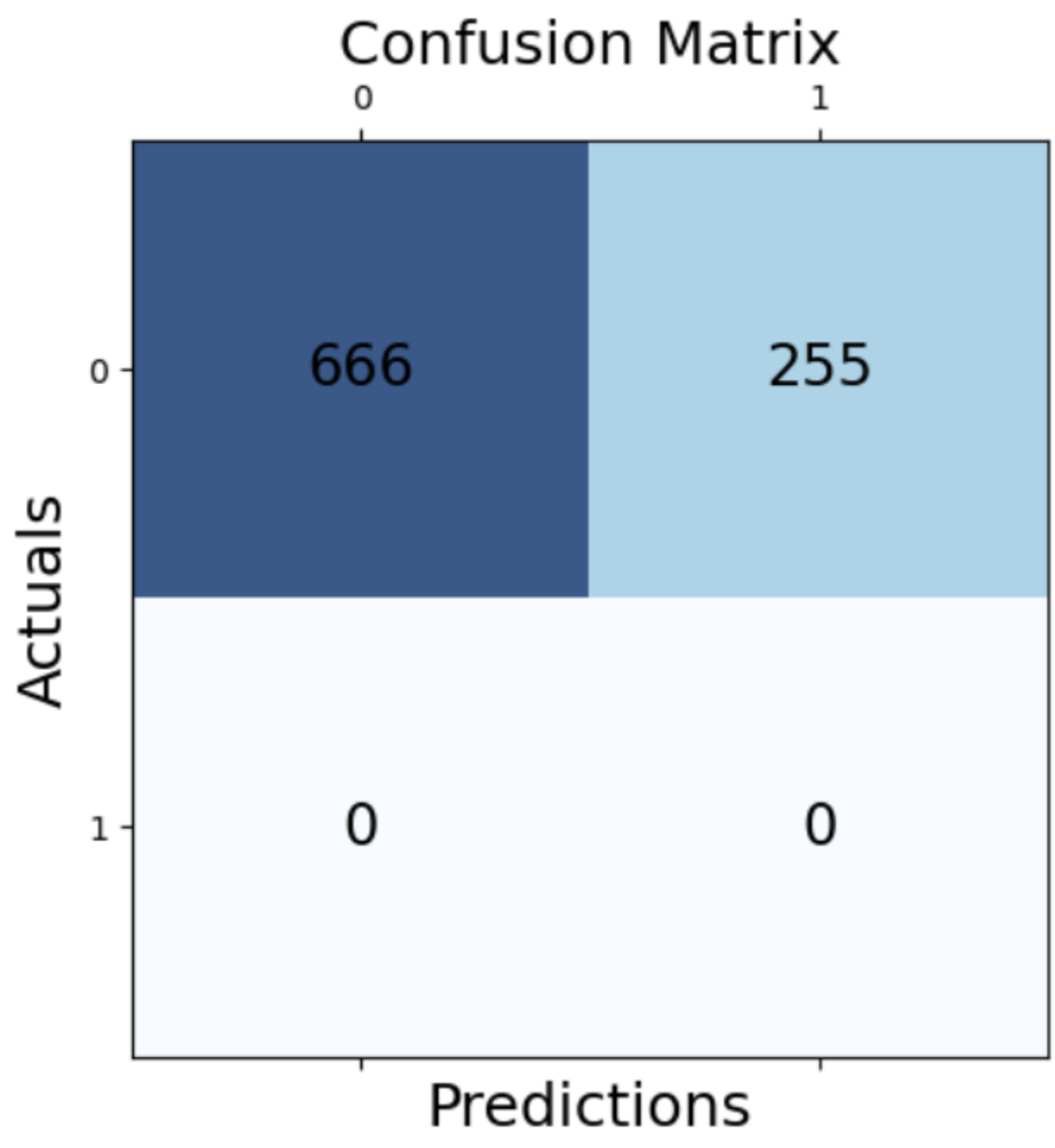
- the accuracy and confusion matrix for Gaussian classifier:

```
accuracy of test data for Gaussian:  0.6178067318132465
Confusion matrix for test set:
 [[569 352]
 [  0   0]]
```



- the accuracy and confusion matrix for Multinomial Naive Bayes Classifiers:

```
accuracy of test data for Multinomial:  0.7231270358306189
Confusion matrix for test set:
 [[666 255]
 [  0   0]]
```



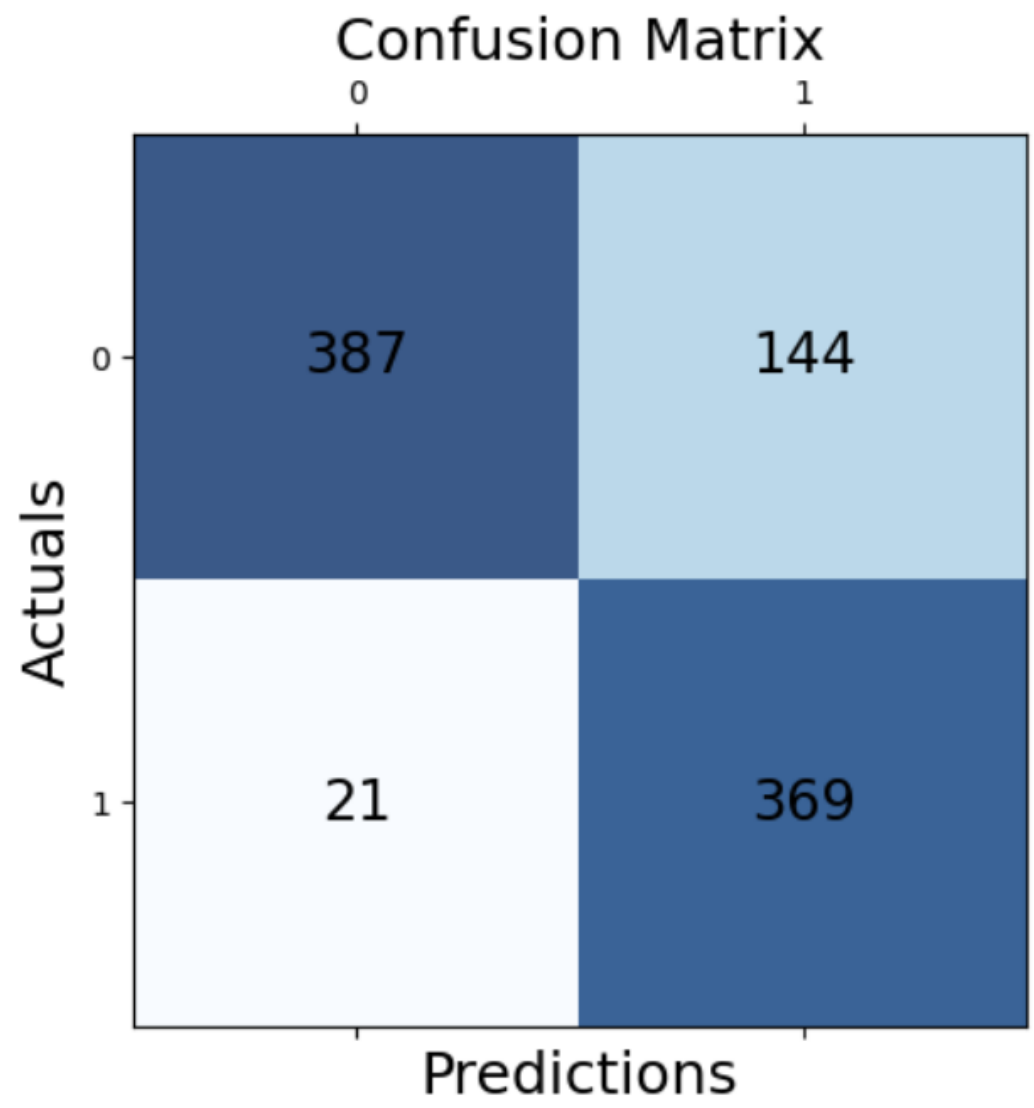
(b)

- Verifying the splitting) using train test split function) process

```
Training set shape (3680, 57)
Testing set shape (921, 57)
```

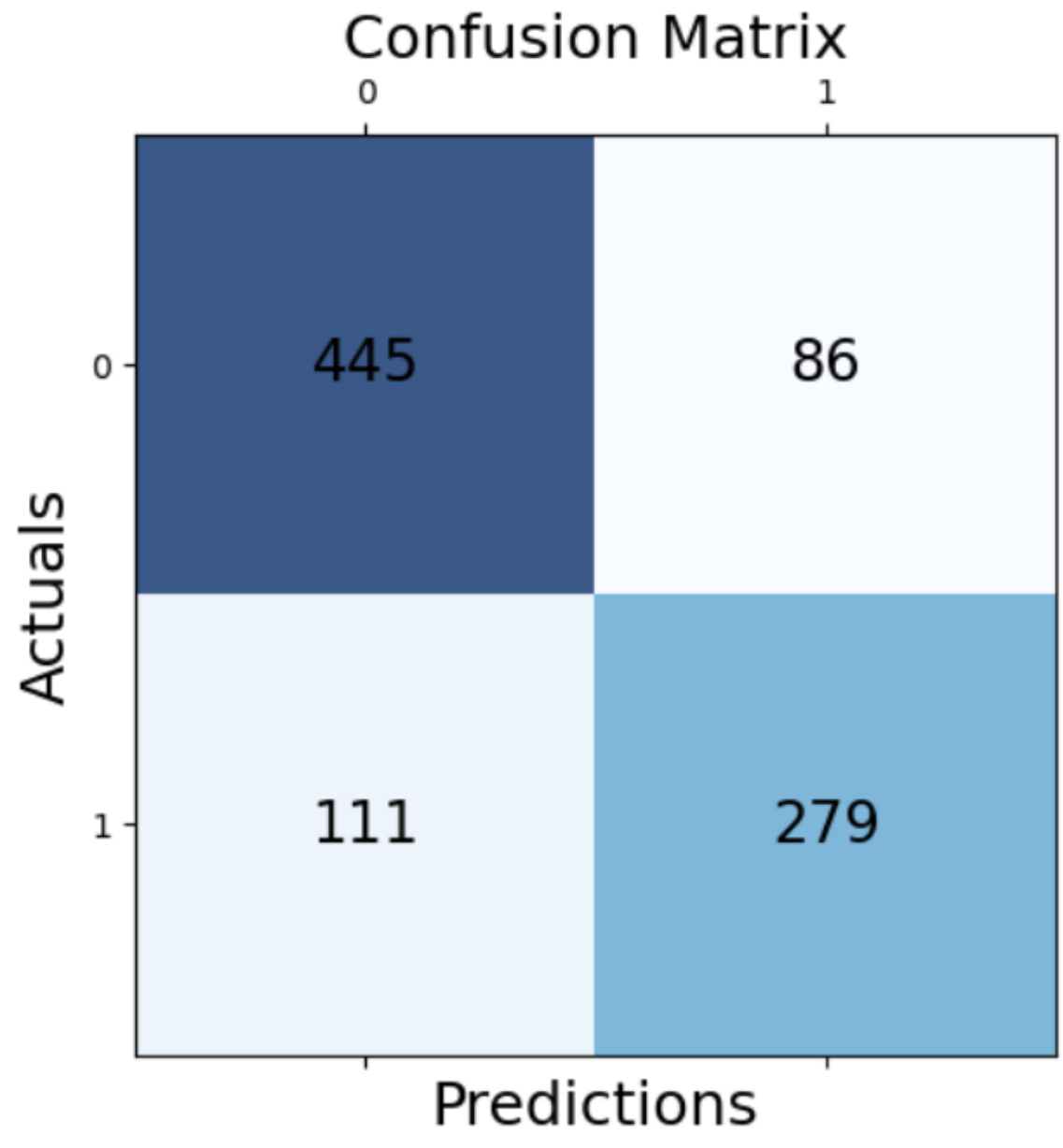
- the accuracy and confusion matrix for Gaussian classifier:

```
accuracy of test data for Gaussian:  0.8208469055374593
Confusion matrix for test set:
 [[387 144]
 [ 21 369]]
```



- the accuracy and confusion matrix for Multinomial Naive Bayes Classifiers:

```
accuracy of test data for Multinomial:  0.7861020629750272
Confusion matrix for test set:
 [[445  86]
 [111 279]]
```



(c)

We used Bernoulli classifier. The reason why the Bernoulli Naive Bayes classifier performs better in this case is because it is designed to work with binary data. In this dataset, the features are counts of words and characters, which can be considered binary data by thresholding them at zero. The Bernoulli Naive Bayes classifier models the probability of each feature being present or absent in each class, which is a better fit for this type of data than the Gaussian or Multinomial Naive Bayes classifiers.

The classification report for the Bernoulli Naive Bayes classifier shows that it achieves high precision and recall for both the spam and non-spam classes, with an F1-score of 0.85. The confusion matrix shows that the classifier correctly classified 499 non-spam emails and 312 spam emails, with only 32 false positives and 78 false negatives.

The accuracy and classification report:

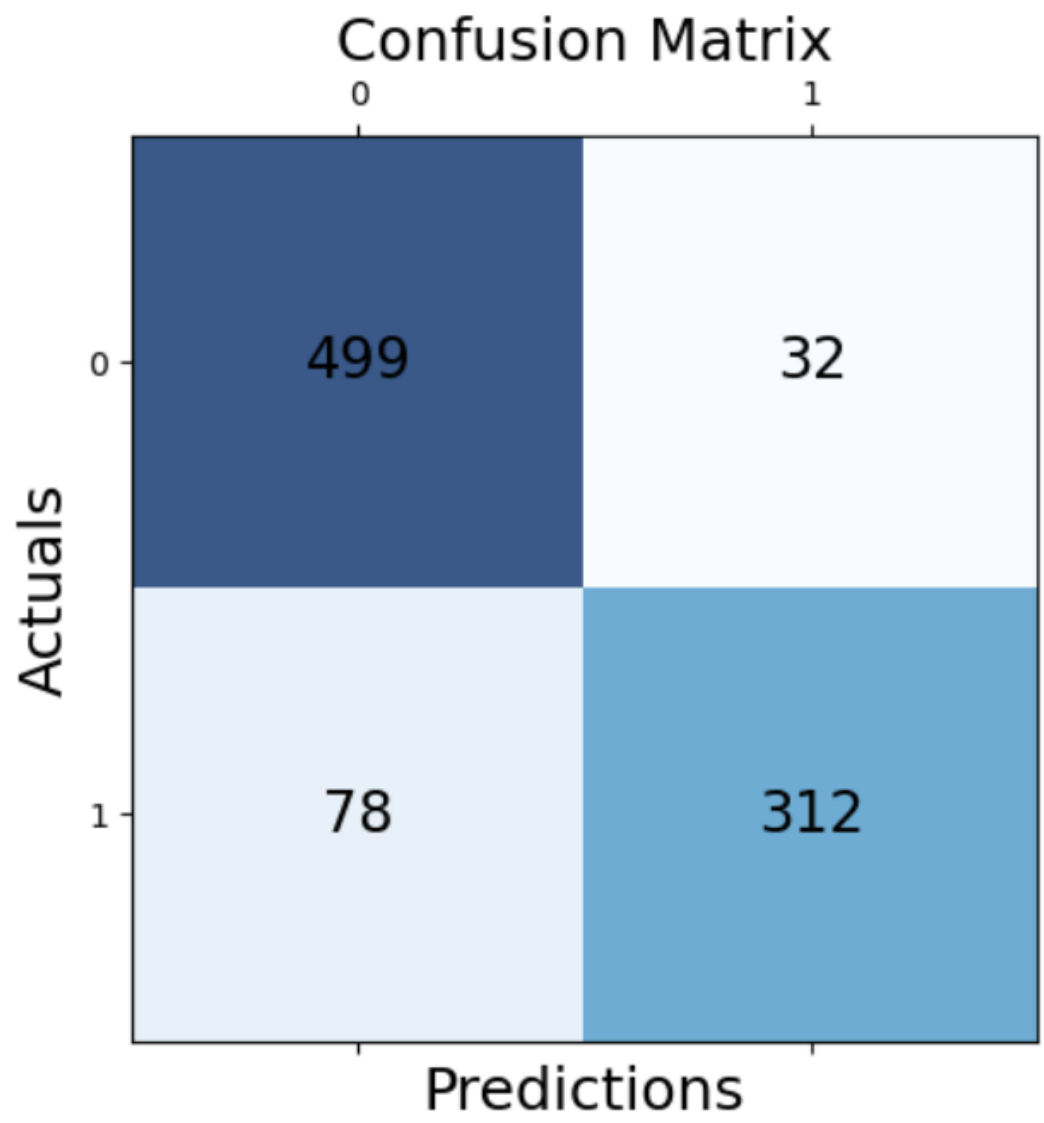
```
The accuracy for Bernoulli classifier 0.8805646036916395
Classification Report
              precision    recall  f1-score   support

           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
```

The confusion matrix:

```
Confusion matrix for test set:
 [[499  32]
 [ 78 312]]
```



(d)

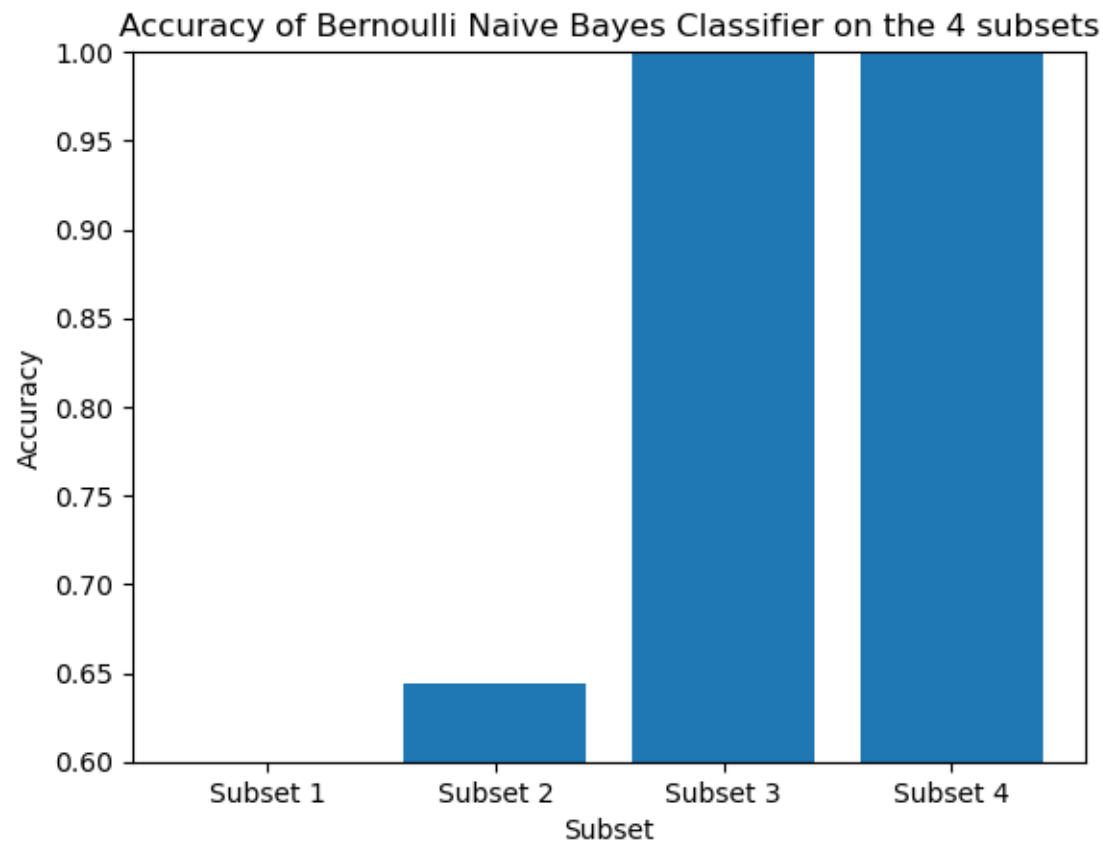
- Verifying splitting the data into four equal parts:

```
Training set of subset-1 (920, 57)
Training set of subset-2 (920, 57)
Training set of subset-3 (920, 57)
Training set of subset-4 (920, 57)
Test set of subset-1 (920,)
Test set of subset-2 (920,)
Test set of subset-3 (920,)
Test set of subset-4 (920,)
```

- The accuracy score for each subset:

```
Accuracies for the 4 subsets [0.0, 0.6438653637350705, 1.0, 1.0]
```

- Plot bar chart to show all subsets' accuracy:



The accuracies of the four subsets are as follows: subset1 = 0.0, subset2 = 0.6438653637350705, subset3 = 1.0, and subset4 = 1.0.

It is obvious that subsets 1 and 2 have the lowest accuracies due to data imbalance. Specifically, in subset1, all the data points have a value of 1. When the model attempts to compare these values with test data, which is also unbalanced and predominantly consists of zeros, this results in poor accuracy. Similarly, in subset2, nearly 893 data points have a value of 1 ("spam"), while only 27 have a value of 0 ("not spam"). As a result, when the model compares subset2 with test data, it yields an accuracy of 0.6438653637350705. In contrast, subsets 3 and 4 have accuracies of 100% as all their values are 1.