Assignment 1

2K18/SE/070
Janvi Arora

2K18/SE/078
Mayand Kumar

Introduction

Two questions were given and we had to apply Naive Bayes Models on the datasets given in both the questions. Dataset in first question was a MNIST dataset for digit recognition and second was of Bank Note Authentication.

1. Dataset Description, Pre-Processing, Model Training and Prediction performed on MNIST dataset for hand digit recognition

1.1. Dataset Description

MNIST dataset is a dataset of handwritten digits consisting of 60,000 images in training dataset and 10,000 images in test dataset. It was created using the samples of NIST dataset. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset.

1.2. Pre-processing

Initially dataset tuples were in the form of 28*28 matrix. So, to make dataset tuple suitable for input in our model, we flattened the tuples resulting in an array of 784 elements where each element corresponds to a pixel of the image.

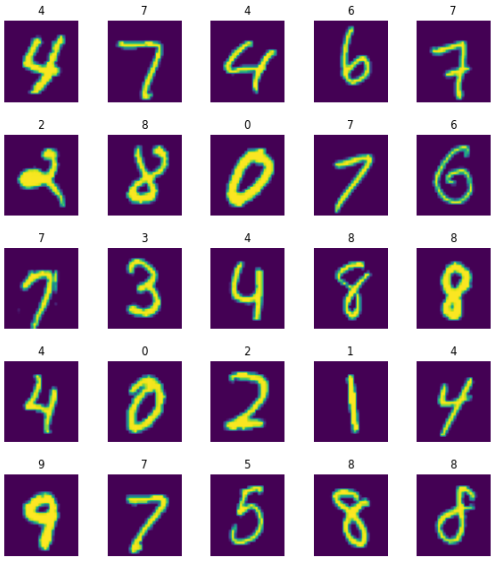


Fig 1. Some Randomly generated images

1.2.1. Making data frame for classification between 0 and 1

For solving this part of question, We made a separate data frame containing 0 label data and 1 label data for both training and testing set. So, now our training data size and testing data size got reduced to 12665 and 2115 respectively.

```
Training Data Shape:
(12665, 784)
(12665,)
```

```
Testing Data Shape:
(2115, 784)
(2115,)
```

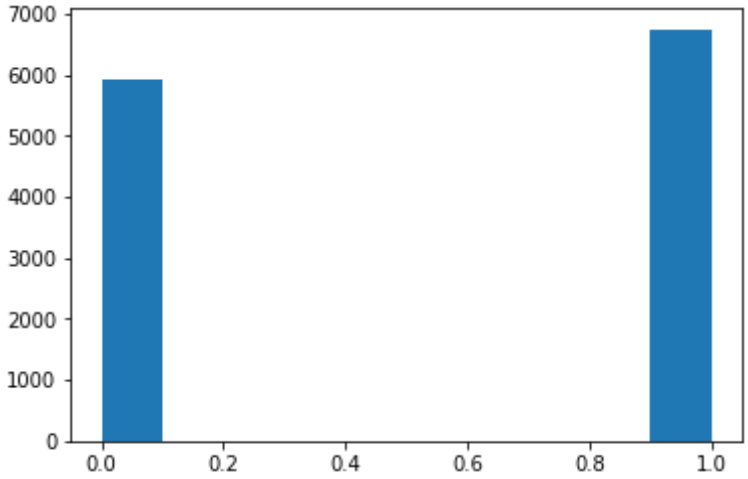


Fig 2. Histogram to visualize the frequency of 0 and 1 in dataset

1.2.2. Making dataframe for classification between 3 and 8

For solving this part of question, We made a separate dataframe containing 3 label data and 8 label data for both training and testing set. So, now our training data size and testing data size got reduced to 11982 and 1984 respectively.

```
Training Data Shape:
(11982, 784)
(11982,)
```

```
Testing Data Shape:
(1984, 784)
(1984,)
```

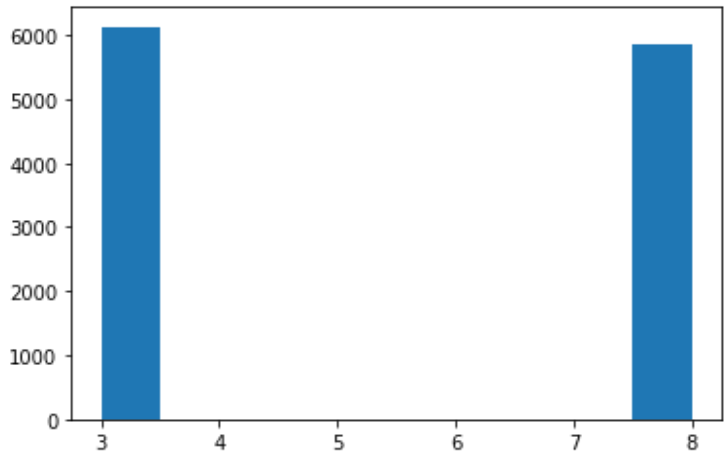


Fig 3. Histogram to visualize the frequency of 3 and 8 in dataset

1.3. Classifier Used

In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. Conditional probability is given by:-

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

1.4. Performance Measures used

We have used accuracy, precision , recall , f1-score as our performance measures . We have evaluated the confusion matrix . The other measure which we have used is ROC curve .

A ROC curve is a curve which is used to evaluate the performance of algorithms with false-positive rates on the x-axis and true positive rates being plotted on the y-axis. AUC (Area under ROC curve) is considered as the most appropriate measure to evaluate ROC Curves.

1.5. Results and Analysis

1.5.1. For Classification between 0 and 1

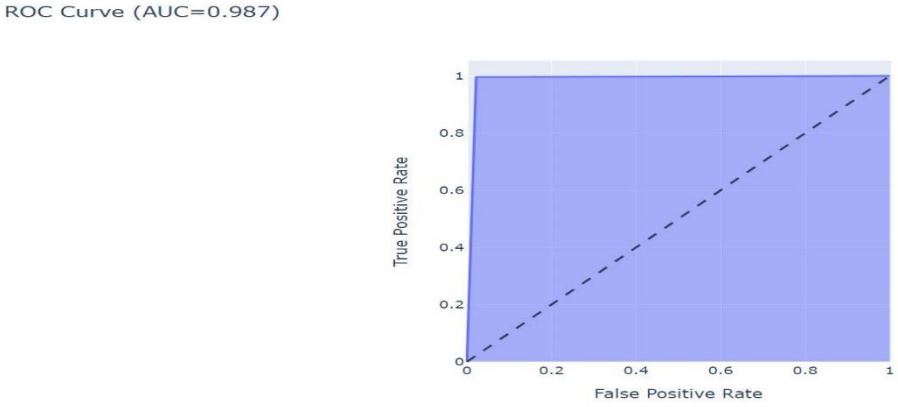
The Confusion Matrix, Classification Report, and Accuracy Score generated for this part :

```
Confusion Matrix:
[[ 976    4]
 [  22 1113]]

Classification Report
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       980
           1       1.00      0.98      0.99      1135

    accuracy                           0.99      2115
```

```
   macro avg       0.99      0.99      0.99      2115
weighted avg       0.99      0.99      0.99      2115
```

```
Accuracy:
0.9877068557919622
```

We have also generated ROC Curve and AUC score for this part came to be 0.987



We can analyse from the result of performance measures that we took into consideration for this problem that our Bayesian Model performed quite well in predicting the digits. The results are quite good because both the digits are very much distinct in appearance from each other. So, it was easy for the classifier to classify between them.

1.5.2. For Classification between 3 and 8

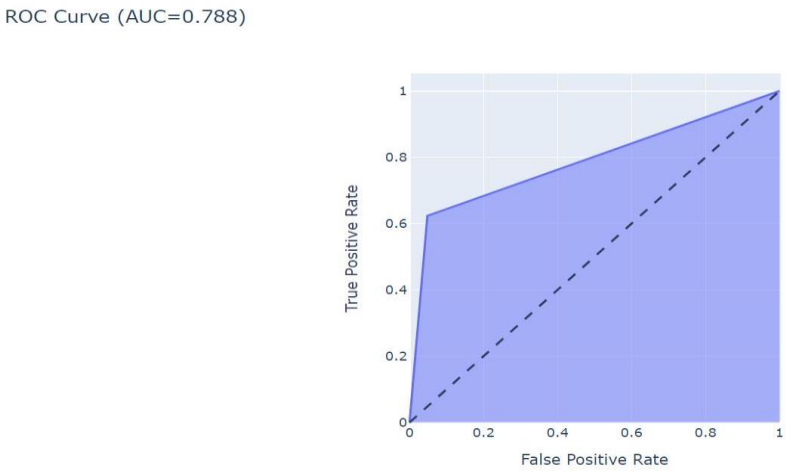
The Confusion Matrix, Classification Report, and Accuracy Score generated for this part :

```
Confusion Matrix:
[[435 575]
 [ 22 952]]
```

```
Classification Report
              precision    recall  f1-score   support

           3       0.95      0.43      0.59      1010
           8       0.62      0.98      0.76       974

    accuracy                           0.70      1984
   macro avg       0.79      0.70      0.68      1984
weighted avg       0.79      0.70      0.68      1984
```

```
Accuracy:
0.6990927419354839
```

We have also generated ROC Curve and AUC score for this part came to be 0.788



We can analyse from the result of performance measures that we took into consideration for this problem that our Bayesian Model performed poor in classifying the digits. The results are quite bad because both the digits are very much alike in appearance from each other. So, it was quite difficult for the classifier to classify between them. Precision in classification report clearly indicates the difficulty faced by the classifier to classify between them.

1.6. Results

The model performs well in classifying 0 and 1 since they both are quite distinct in appearance whereas 3 and 8 are quite similar. Thus performance measures are low for the 3 and 8 classification than of 0 and 1.

2. Dataset Description, Pre-Processing, Model Training and Prediction performed on Bank Note Authentication Dataset

2.1. Dataset Description

Data were extracted from images that were taken from genuine and forged banknote-like specimens. For digitization, an industrial camera usually used for print inspection was used. The final images have 400x 400 pixels. Due to the object lens and distance to the investigated object gray-scale pictures with a resolution of about 660 dpi were gained. Wavelet Transform tool were used to extract features from images.

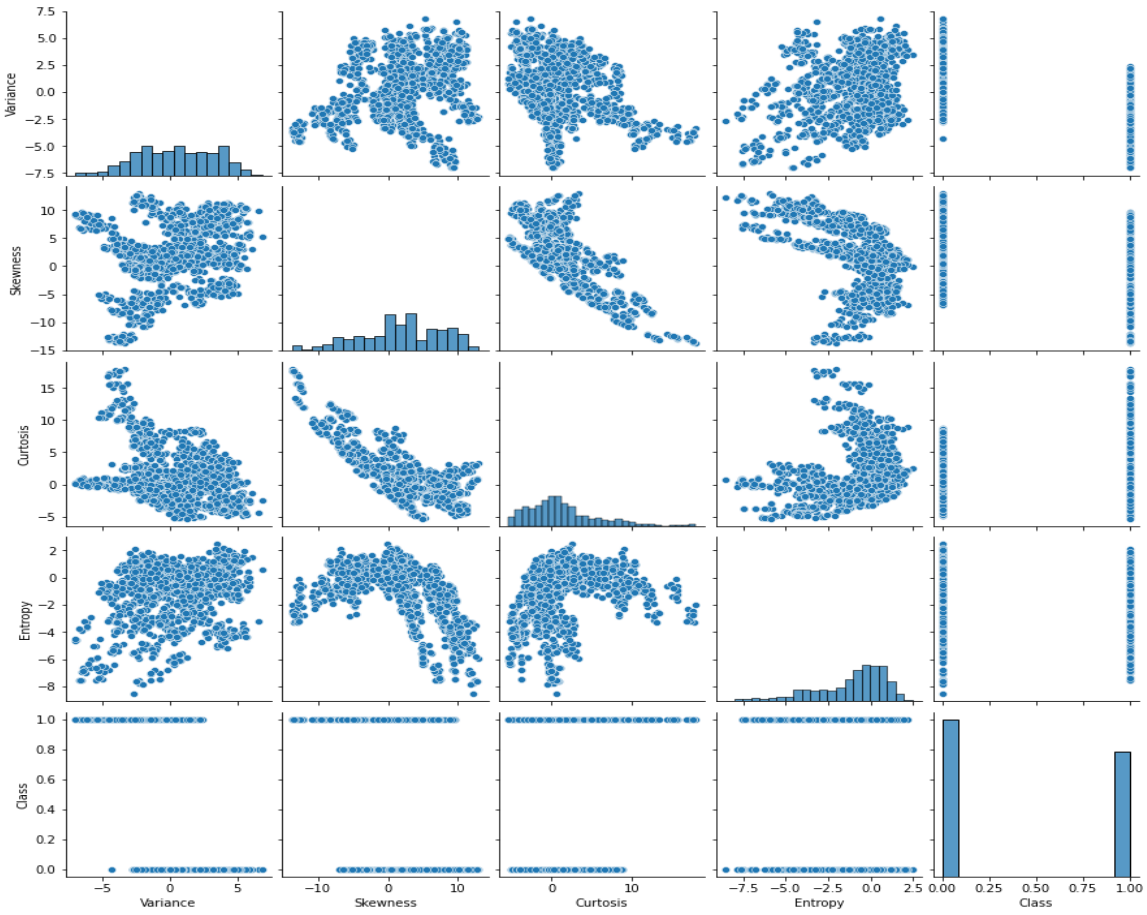
Attribute Information:

1.variance of Wavelet Transformed image (continuous)
2. skewness of Wavelet Transformed image (continuous)
3. curtosis of Wavelet Transformed image (continuous)
4. entropy of image (continuous)
5. class (integer)

2.2. Pre-processing

Initially the dataset heading were not there so we named the columns with proper attribute names and also scaled the features using Standard Scaler for standardization for better results.

We also visualized our dataset:



Our dataset contains total 1372 data.

In this question, we need to divide the dataset into training and testing set in such a way that both contains 50 % of 0 label data and 1 label data.

So, We made a dataframe containing 0 and 1 label data separately and we splitted them using train_test_split function and later concatenated 50% of 1 label data and 50% 0 label data to make training and testing set. Code snippet is shown below:

```
from sklearn.model_selection import train_test_split
x1_train,x1_test,y1_train,y1_test=train_test_split(x1,y1,test_size=1/2,
random_state=1)
x0_train,x0_test,y0_train,y0_test=train_test_split(x0,y0,test_size=1/2,
random_state=1)
x_train=np.concatenate((x1_train,x0_train),axis=0)
y_train=np.concatenate((y1_train,y0_train),axis=0)
x_test=np.concatenate((x1_test,x0_test),axis=0)
y_test=np.concatenate((y1_test,y0_test),axis=0)
print(x_train.shape)
print(x_test.shape)
```

Which resulted in training set and testing set of shape (686,4) each.

```
(686, 4)
(686, 4)
```

2.2.1. For Classes with Equiprobablity

Here we set the class_prior_=[0.5,0.5] for the classifier as the part of the question tells us that the classes are equiprobable.

2.2.2. For Classes having prior of genuine class 0.9 and forged 0.1.

Here we set the class_prior_=[0.1,0.9] for the classifier.

After performing these preprocessing steps for each part of question, we trained our model on these datasets and predicted on x_test to evaluate our Classifier's performance.

2.3. Classifier Used

The Classifier used is Naïve Bayes. In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution. A Gaussian distribution is also called Normal distribution. Conditional probability is given by:-

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

2.4. Performance measures

We have used accuracy, precision , recall , f1-score as our performance measures . We have evaluated the confusion matrix . The other measure which we have used is ROC curve .
A ROC curve is a curve which is used to evaluate the performance of algorithms with false-positive rates on the x-axis and true positive rates being plotted on the y-axis. AUC (Area under ROC curve) is considered as the most appropriate measure to evaluate ROC Curves.

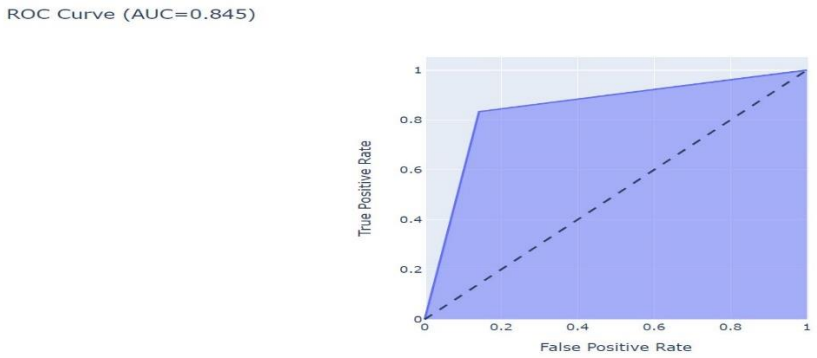
2.5. Results and Analysis

2.5.1. For Equiprobable Classes

The Confusion Matrix, Classification Report, and Accuracy Score generated for this part :

```
Confusion Matrix:
[[331  50]
 [ 55 250]]

Classification Report
              precision    recall  f1-score   support

           0       0.86      0.87      0.86       381
           1       0.83      0.82      0.83       305

    accuracy                           0.85       686
   macro avg       0.85      0.84      0.84       686
weighted avg       0.85      0.85      0.85       686

Accuracy:
0.8469387755102041
```

We have also generated ROC Curve and AUC score for this part came to be 0.845



We can analyse from the result of performance measures that we took into consideration for this problem that our Bayesian Model performed quite well classifying between forged and genuine notes.

2.5.2. For Classes where prior are 0.1 and 0.9

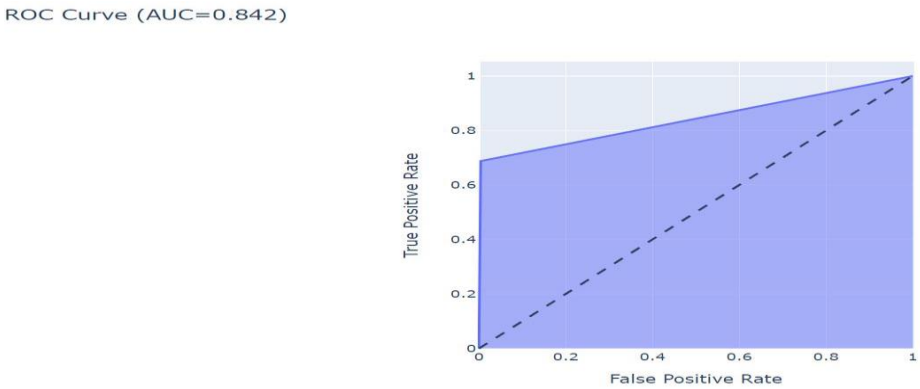
The Confusion Matrix, Classification Report, and Accuracy Score generated for this part :

```
Confusion Matrix:
[[243 138]
 [  1 304]]

Classification Report
              precision    recall  f1-score   support
```

```
           0       1.00      0.64      0.78       381
           1       0.69      1.00      0.81       305

    accuracy                           0.80       686
   macro avg       0.84      0.82      0.80       686
weighted avg       0.86      0.80      0.79       686
```

```
Accuracy:
0.7973760932944607
```

We have also generated ROC Curve and AUC score for this part came to be 0.842



We can analyse from the result of performance measures that we took into consideration for this problem that our Bayesian Model performed quite well classifying between forged and genuine notes.

2.6. Result

The Naive Bayes Classifier performs well in the detection of counterfeit notes. Upon changing the priors for the genuine Notes class to 0.9 and the forged Notes to 0.1, we see a very steep dip in the Precision of the Fake Notes as we have altered the priors for the classes and provided a low value of prior to Fake Notes. Change of priors makes the model classify to genuine Notes more often thus resulting in reduction in the number of actual fake classifications. We can also see the result using precision shown in the classification report of each.