**Name: Achyuth S.S**

**Registration Number: INBT09157**

**Month: June**

**Year: 2024**

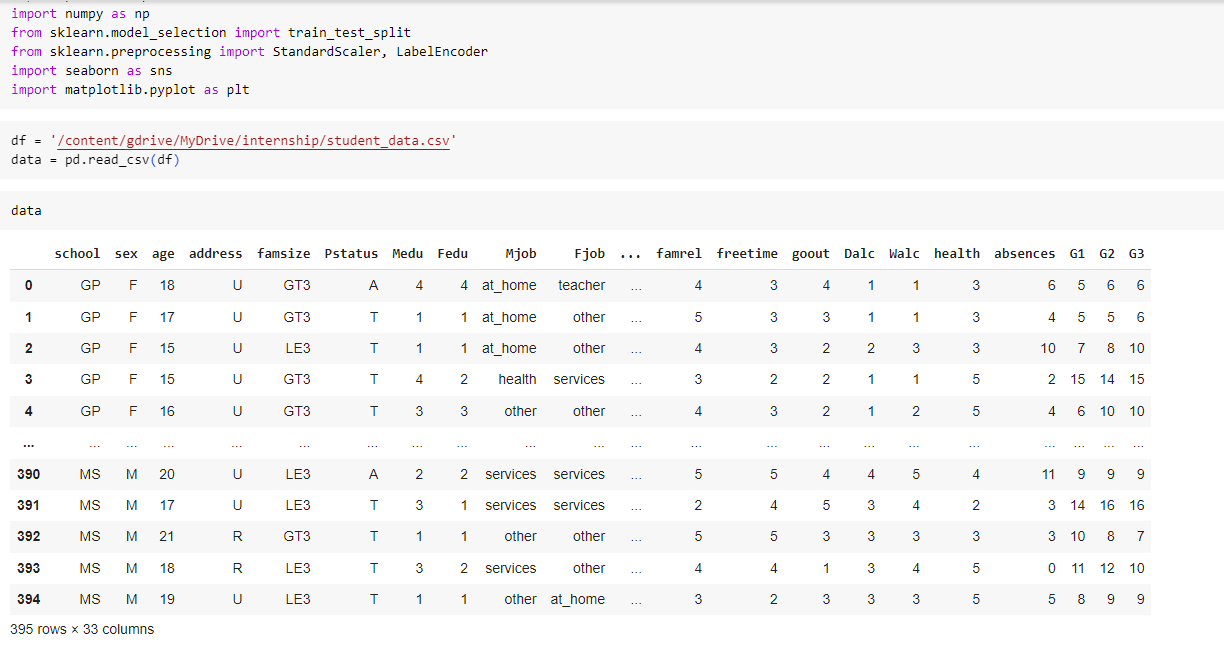
**GitHub Repository -** <https://github.com/achyu2003/Achyuth_INBT09157_June2024>

# Micro-assessment Task-1: Application of Supervised and Unsupervised Machine learning models on the student performance Dataset

Firsts step is mounting Google drive so that we get the dataset

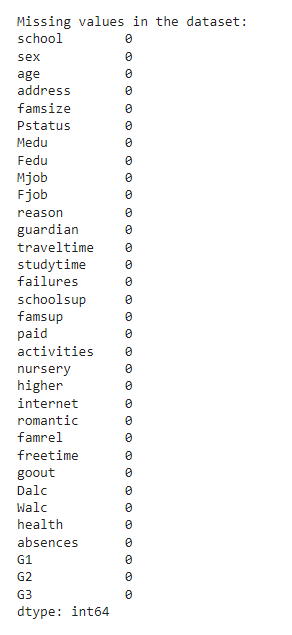
Step 2-

Exploratory data analysis



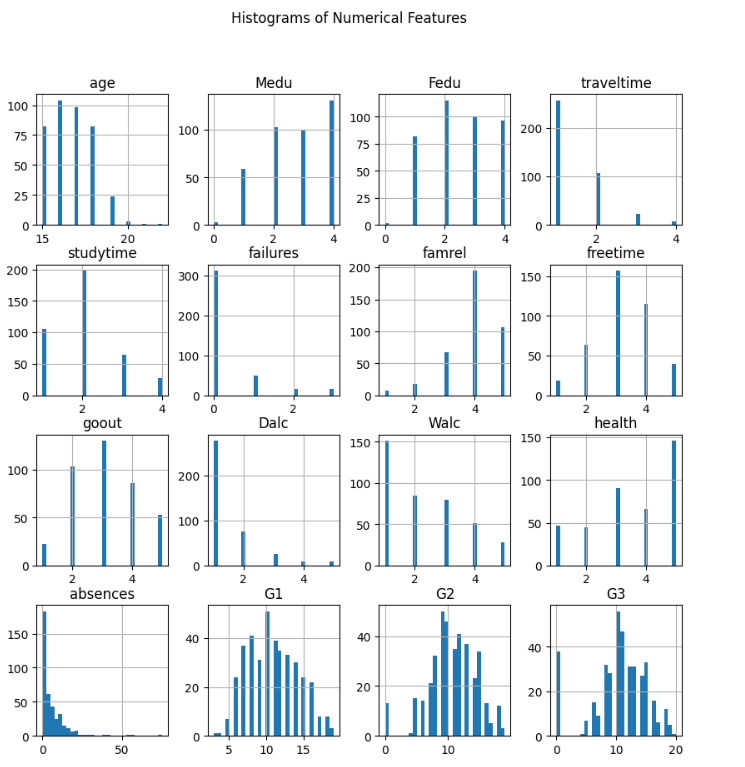
Step 3-

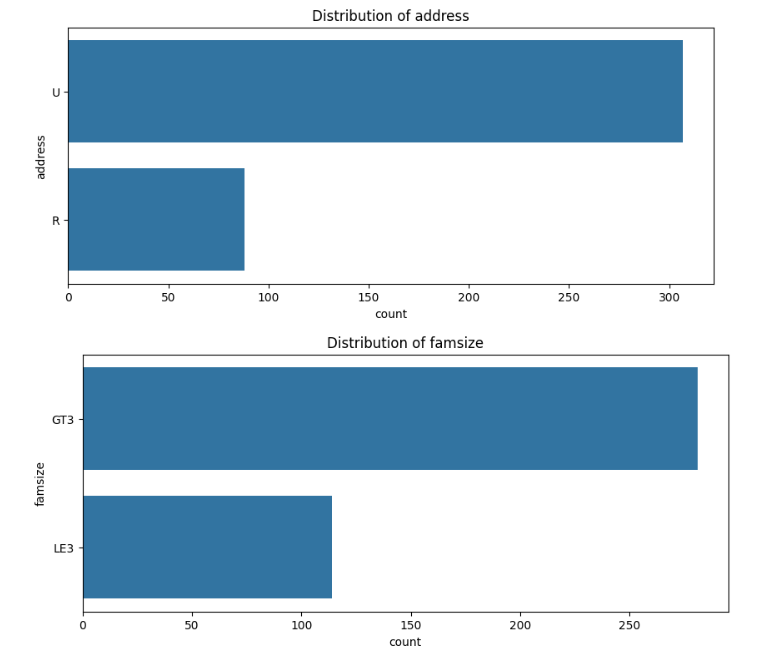
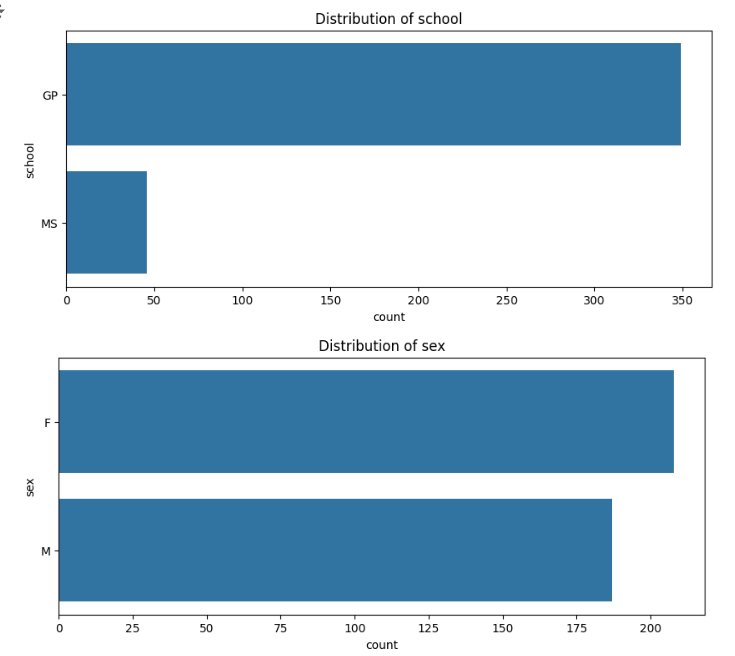
We then check for missing values in the dataset

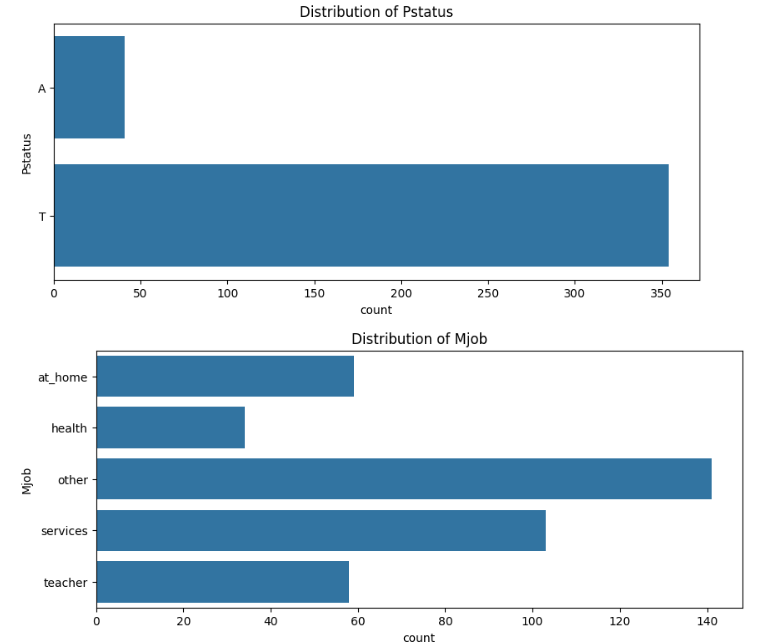


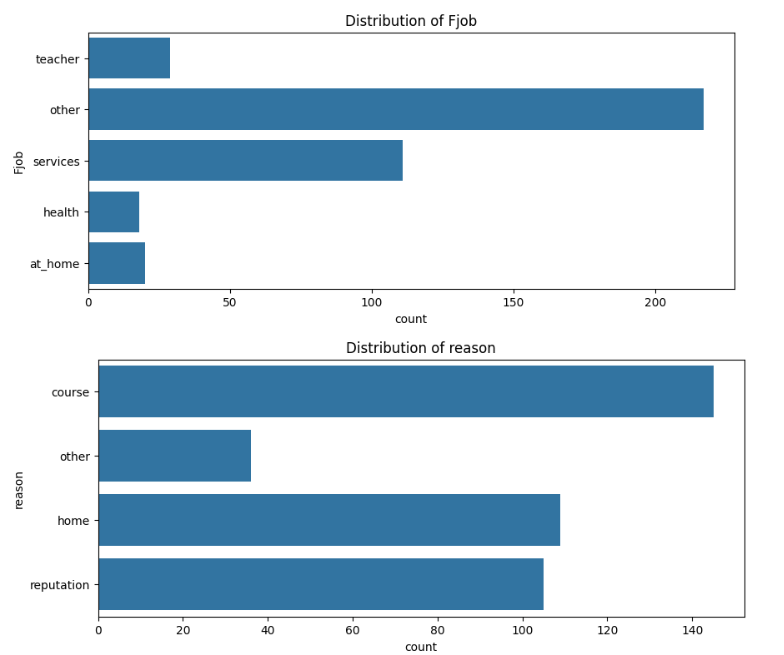
Step 4-

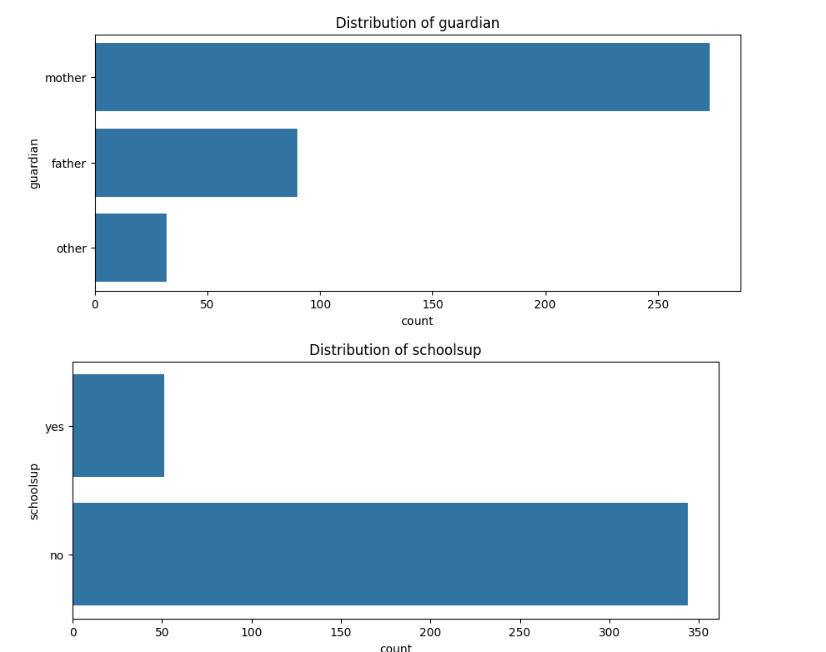
We then check the distribution of Categorical and numerical features in the dataset

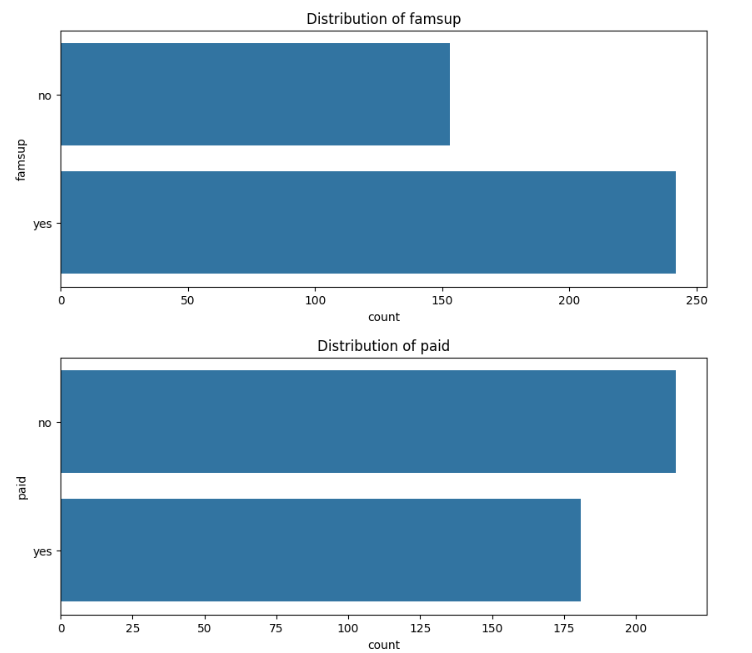


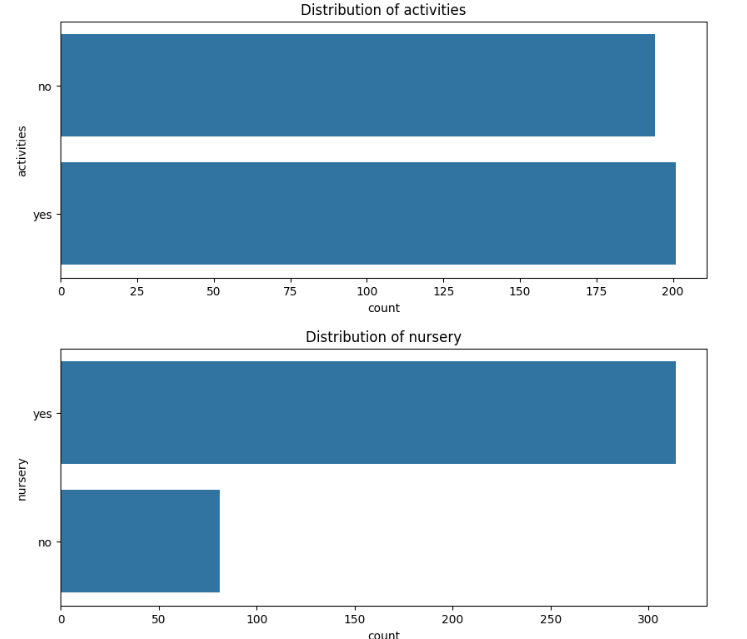


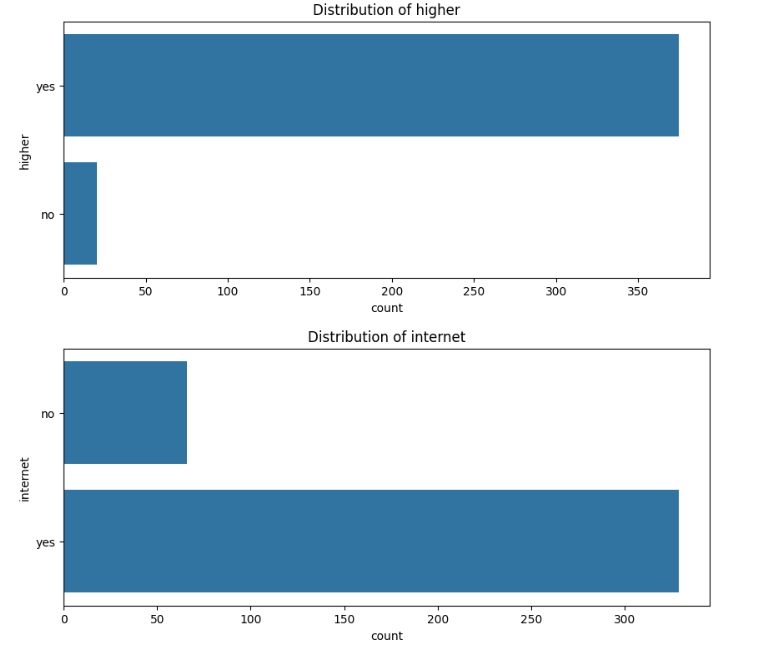


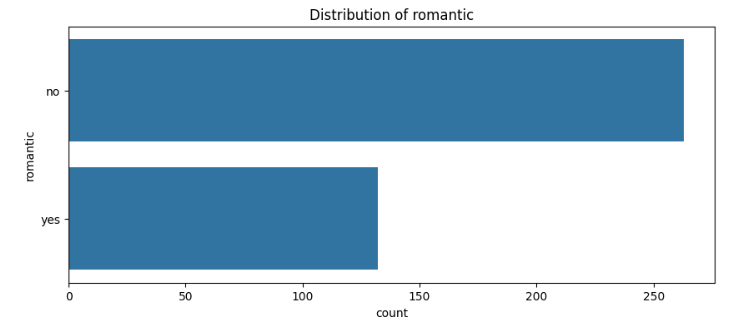






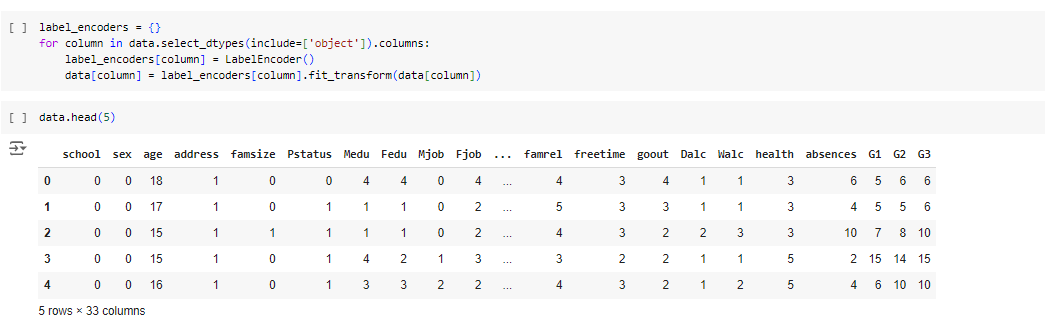






Step 5-

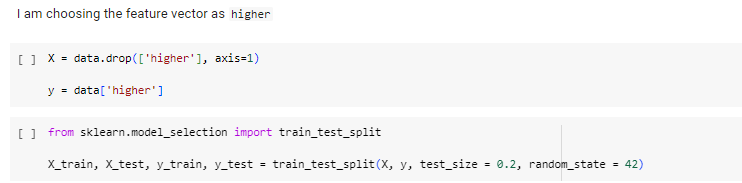
Apply label Encoding techniques on categorical features



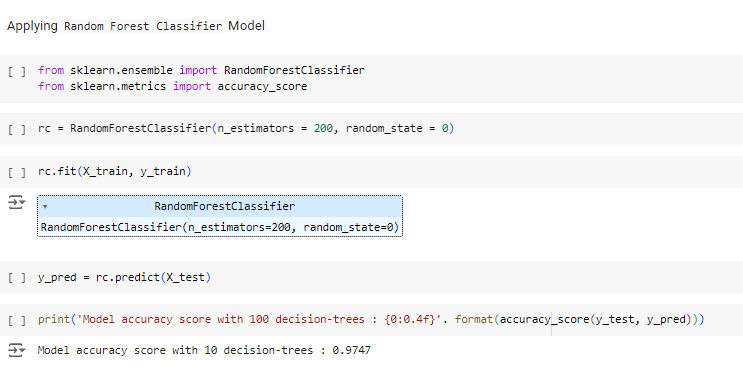
Step 6-

Classification task – To Predict whether a student will go to higher studies or not using Random Forest Classifier

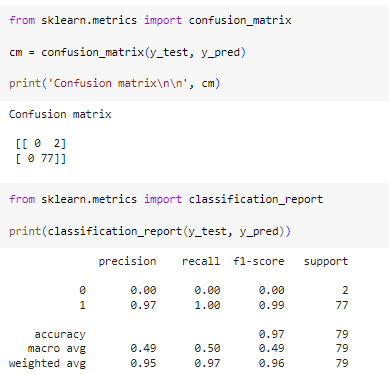
Choosing feature vector as “higher”, we perform the further analysis



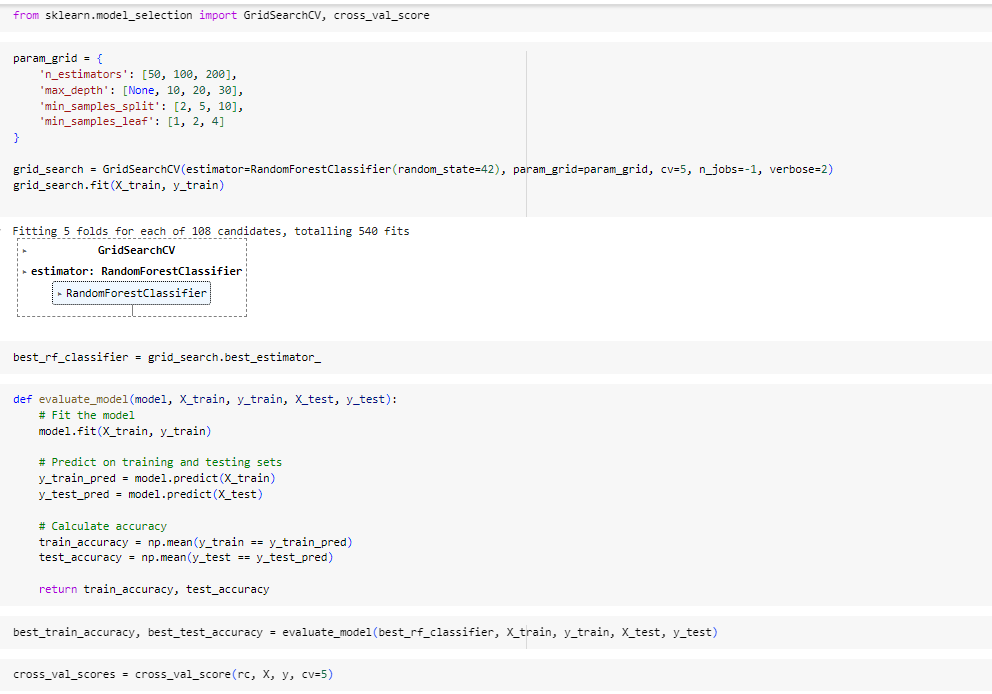
Then, we apply **Random Forest Classifier** to get desired output –



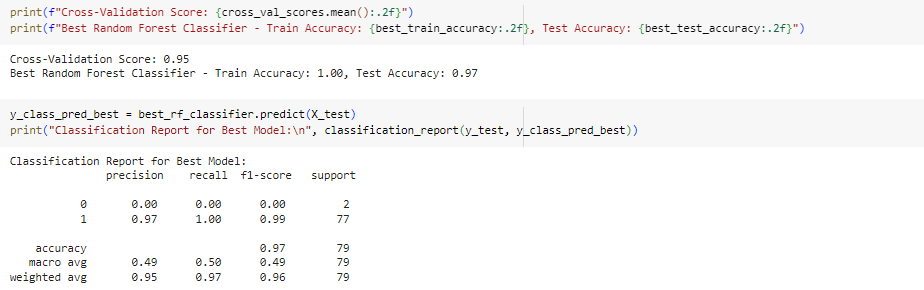
Then, we plot the Confusion Matrix and the Classification report



In order to perform optimization, we applied GridSearchCV method of optimization –



And then we also apply Cross Validation Techniques to overcome overfitting

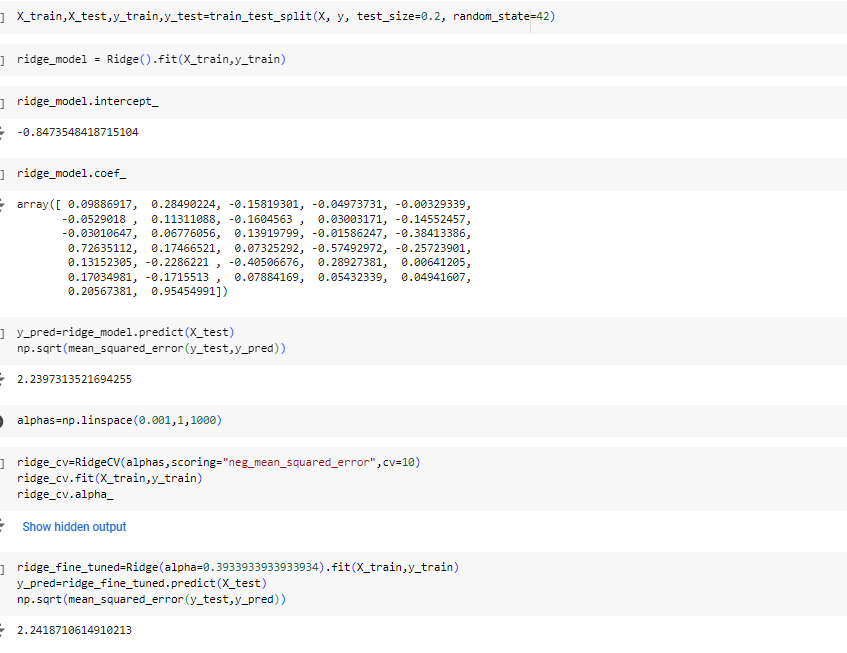


Thus, the Test accuracy is 97% as above.

Step 7- Regression task to Predict final grade G3

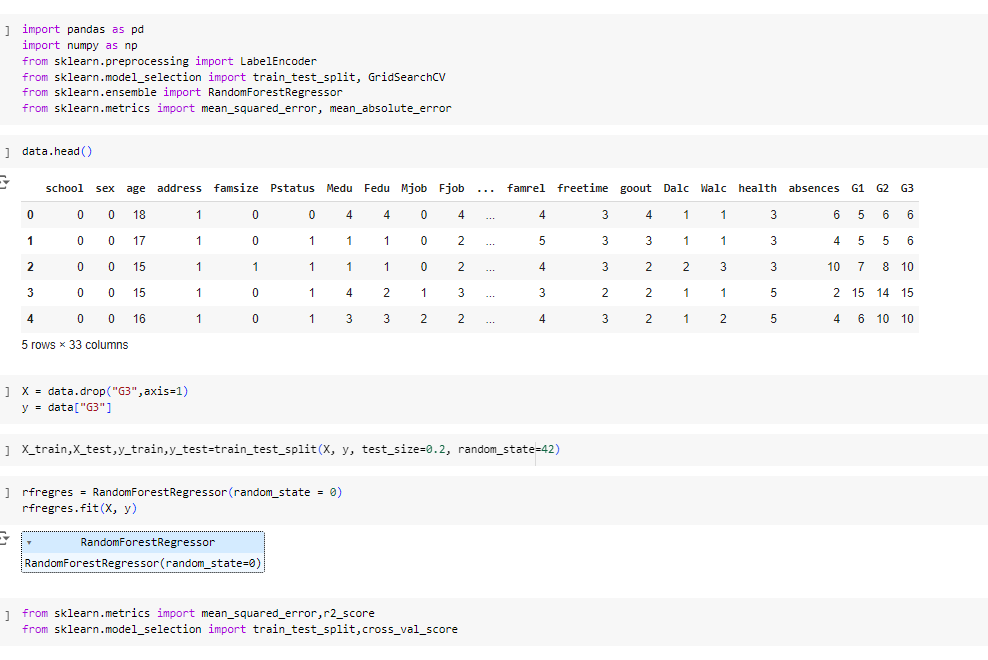
Step 8- Ridge regression

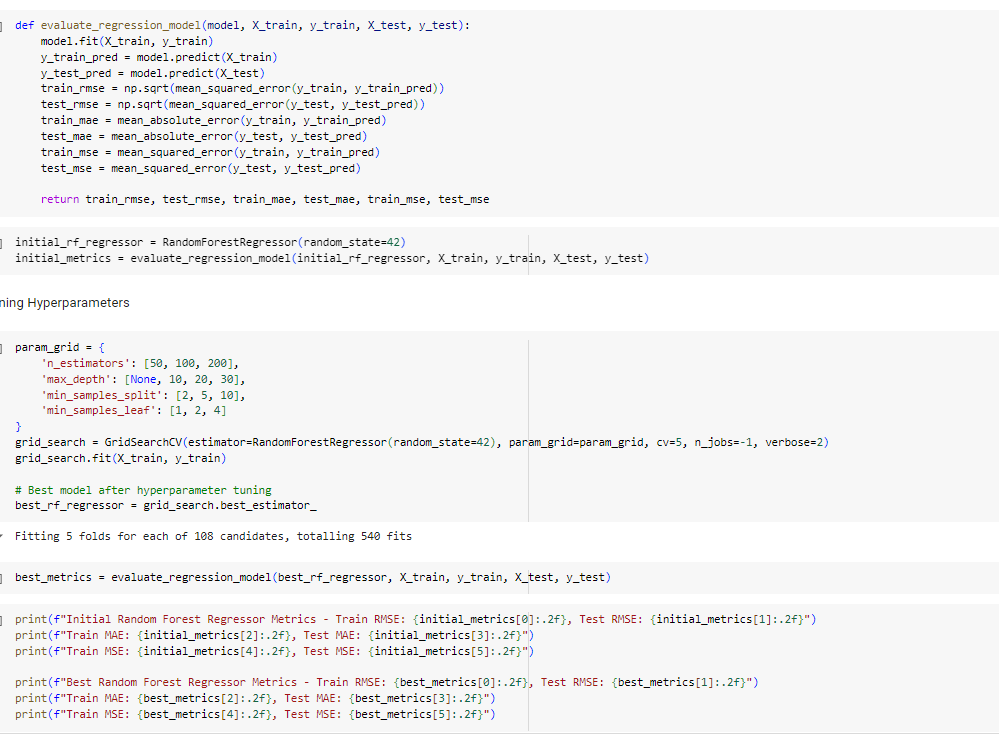




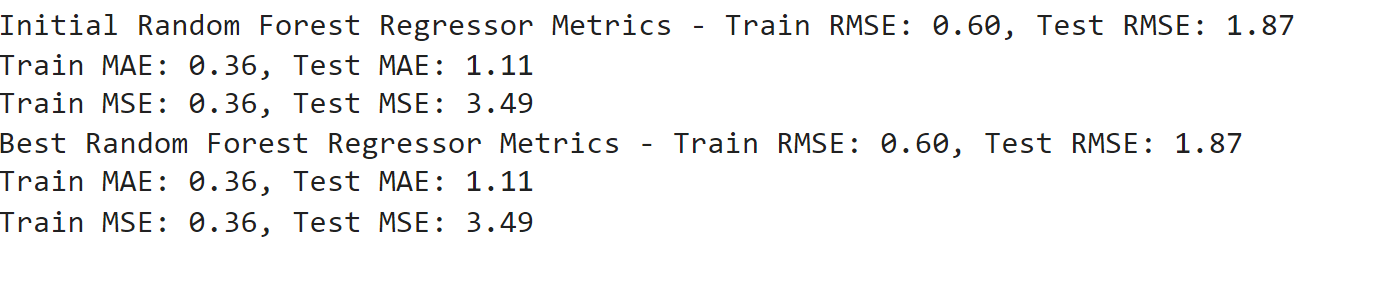
The results are as shown above

Step 8- Random Forest regressor

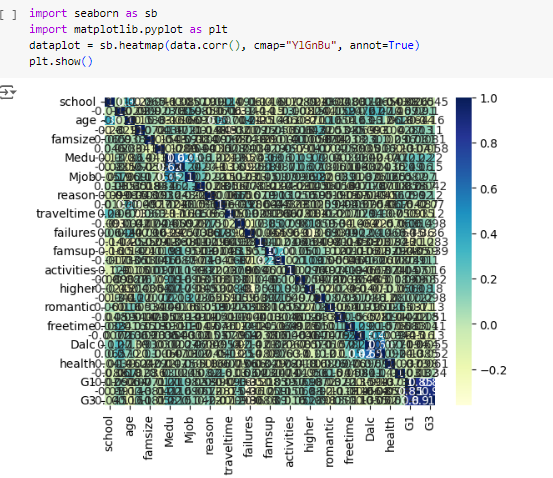




We fine tune the hyperparameters and overcome overfitting and achieve the results as below – for Train, Test MSE, MAE –

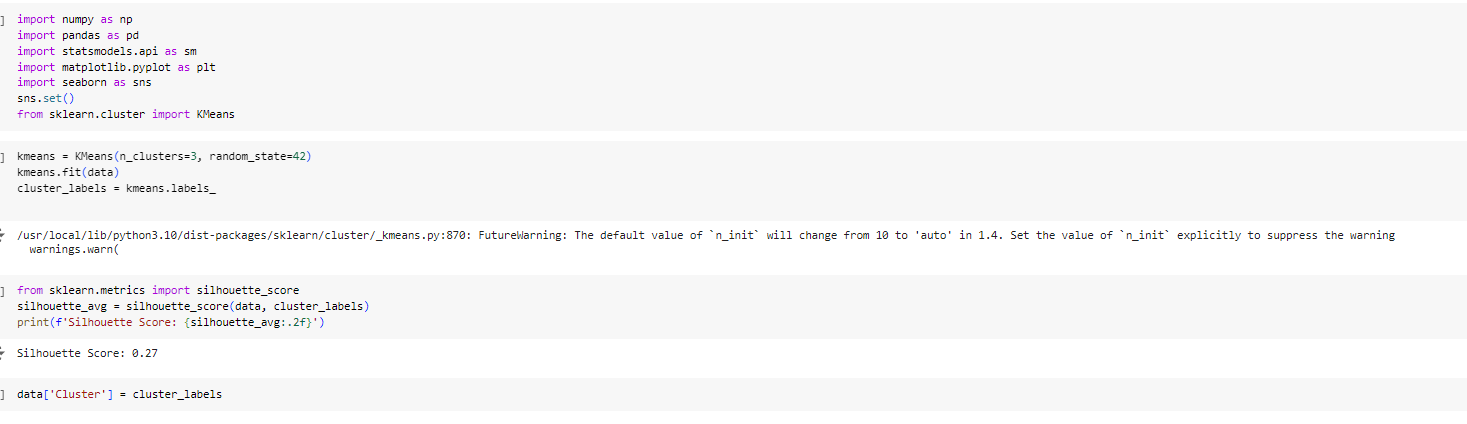


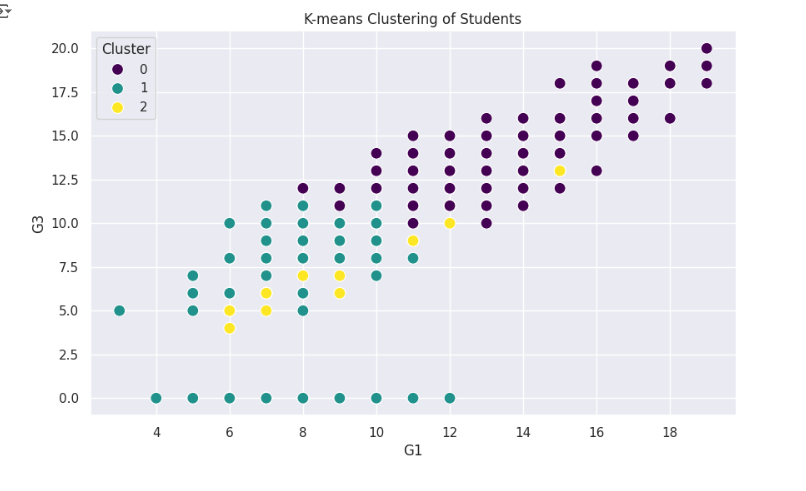
Step 9- K means clustering to identify data clusters



Correlation matrix

We use Silhouette Score to compute The optimal number of clusters and then subsequently plot the Clusters graph –

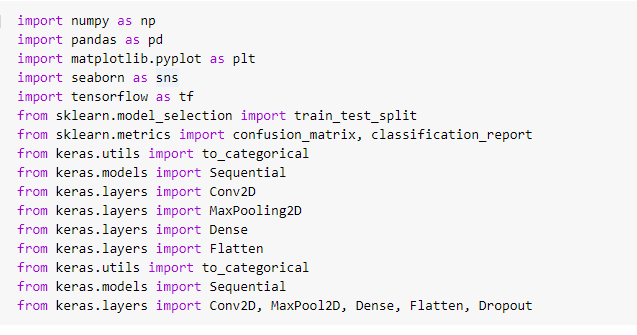




# Micro-assessment Task-2: Utilizing MNIST Dataset on a Convolutional Neural Network Model

1. CNN Model –

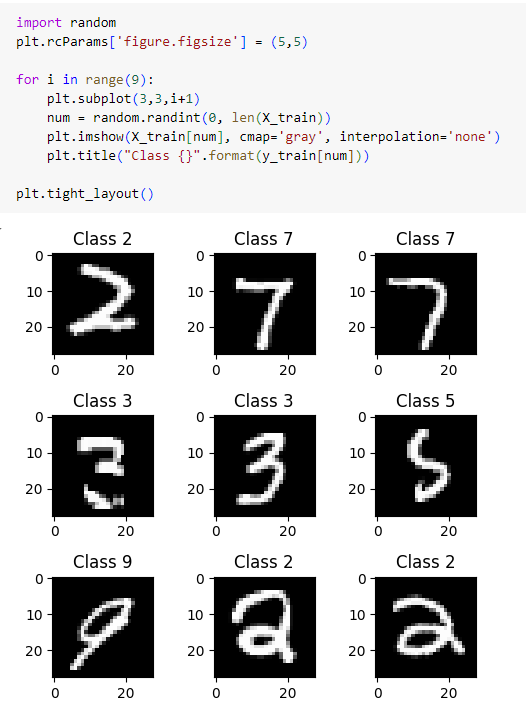
Firstly, we import all required libraries



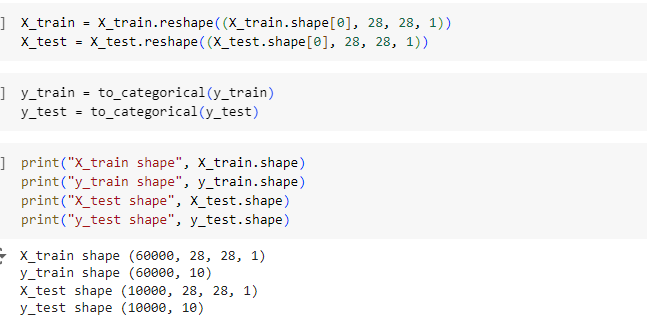
Next, we import MNIST Dataset and then check the split of train and test as follows –

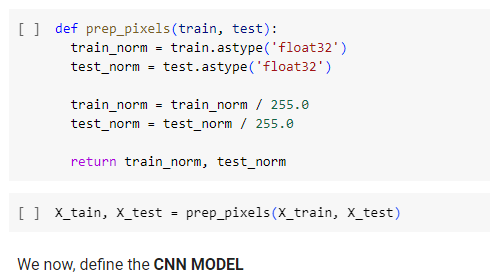


Next, we print some random samples to examine the data

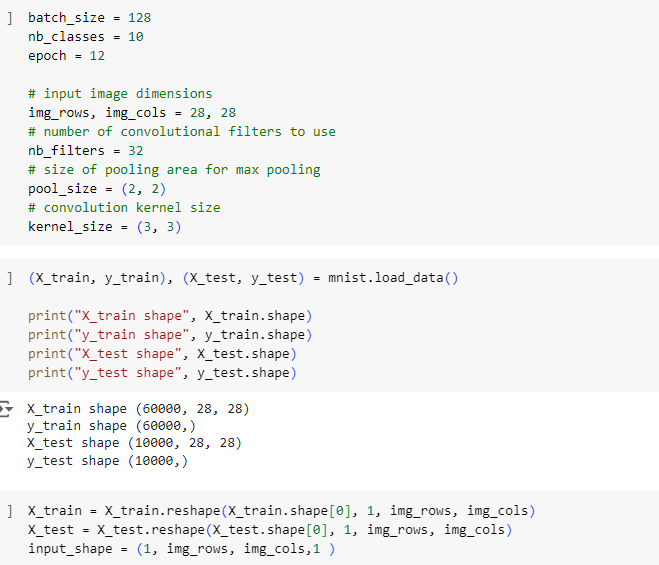


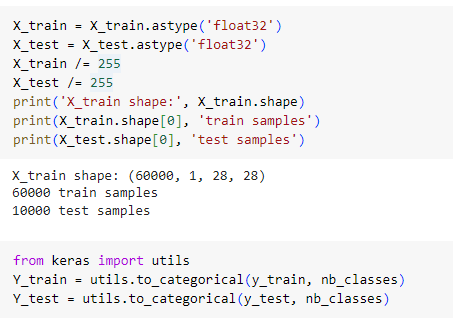
Then, we reshape images so as to fit for the CNN Model.





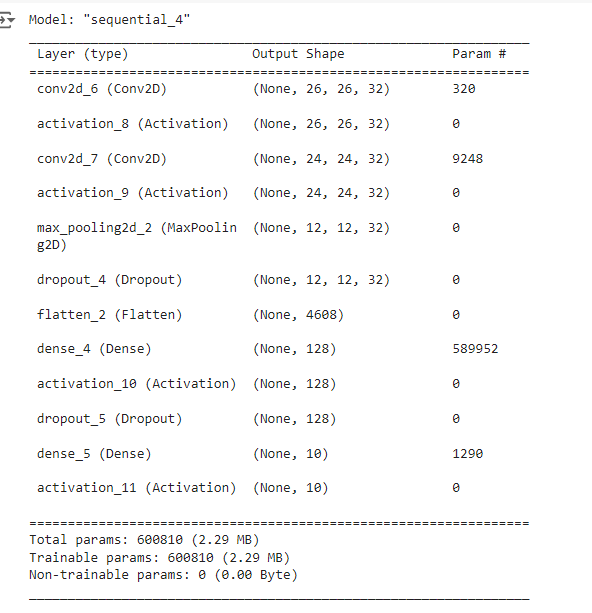
Now, defining the CNN model and its required parameters as follows



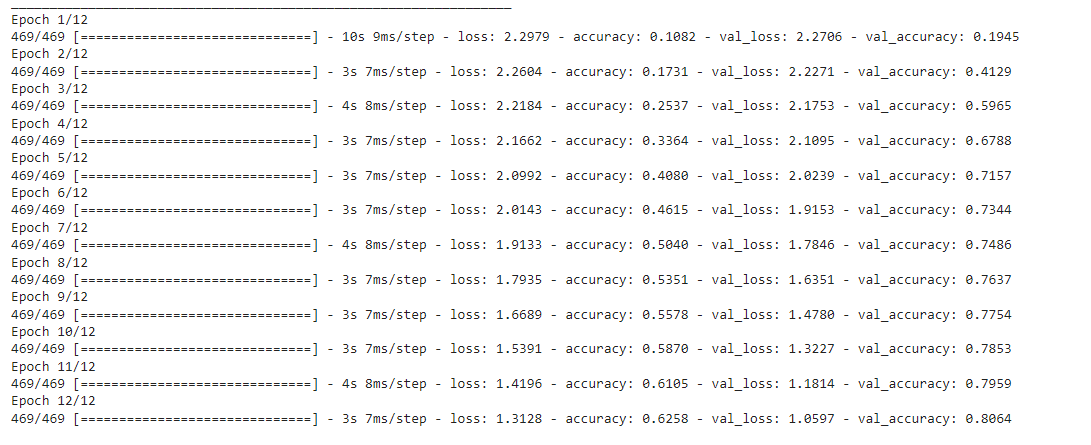


The CNN model is defined as below –

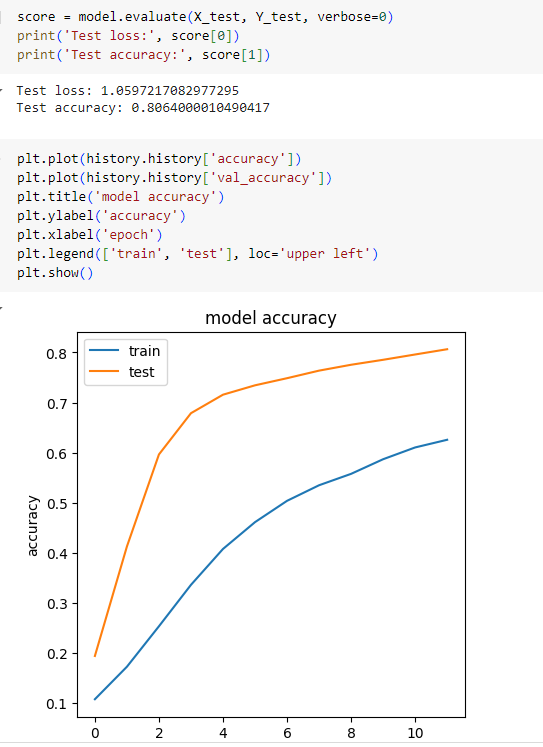




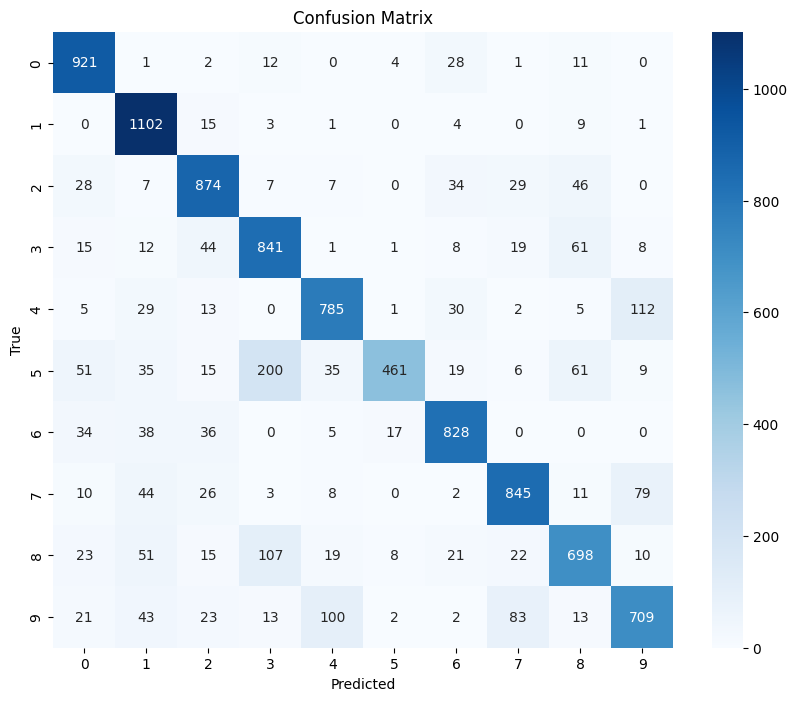
Then, we train the model over 12 epochs –

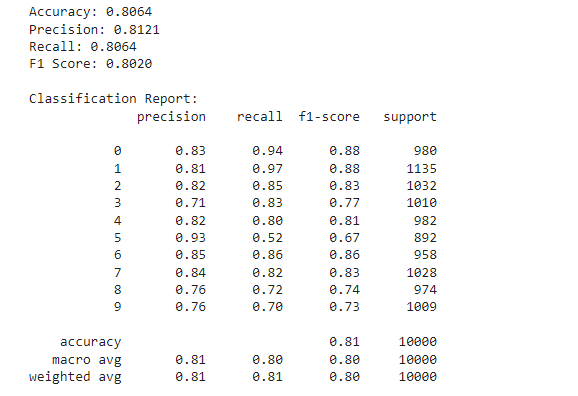


I got a test accuracy of 80.47%



The confusion matrix is as below –



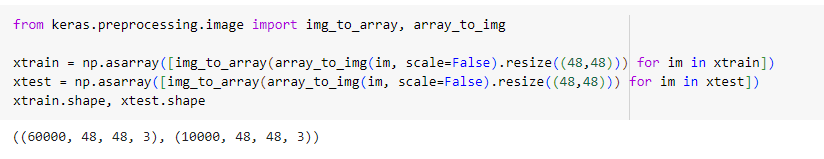


1. VGG16 Architecture

We firstly import all libraries and load the MNIST dataset –



Next, we pre process it, so as to fit in the Model





We then prepare train, test and validation sets as above.

The Final VGG 16 model is as below –

last\_layer=model\_vgg16.output

flatten=layers.Flatten()(last\_layer)

dense1=layers.Dense(100,activation='relu')(flatten)

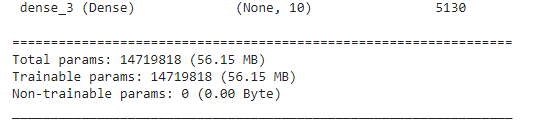
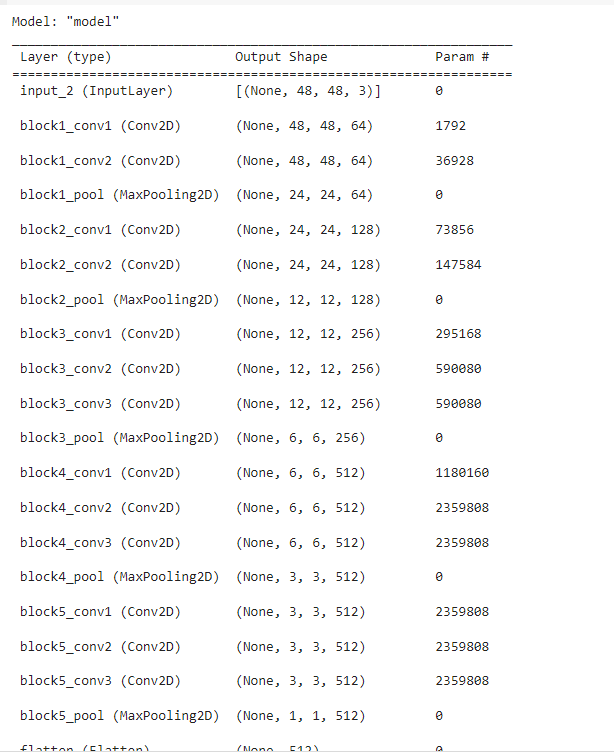
dense1=layers.Dense(100,activation='relu')(flatten)

dense1=layers.Dense(100,activation='relu')(flatten)

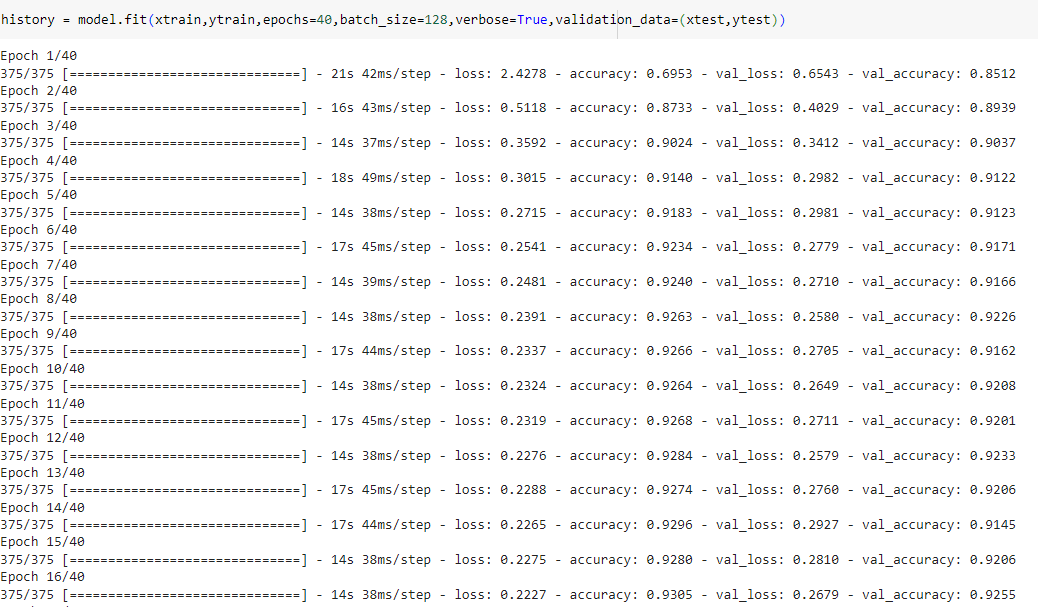
output\_layer=layers.Dense(10,activation='softmax')(flatten)

model=models.Model(inputs=input\_layer,outputs=output\_layer)

model.summary()



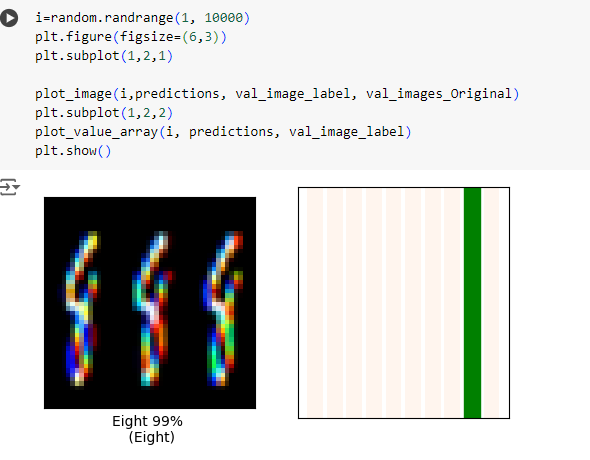
We then compile the model and train it over 40 epochs –



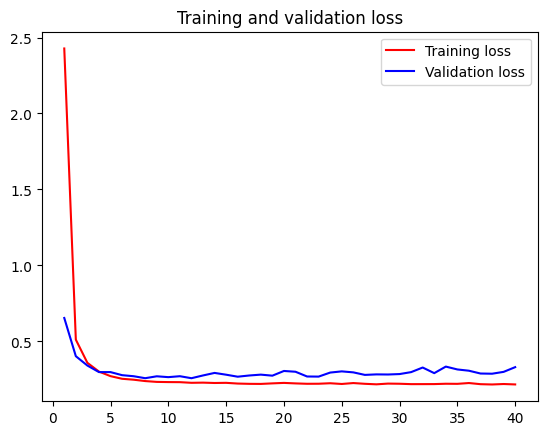
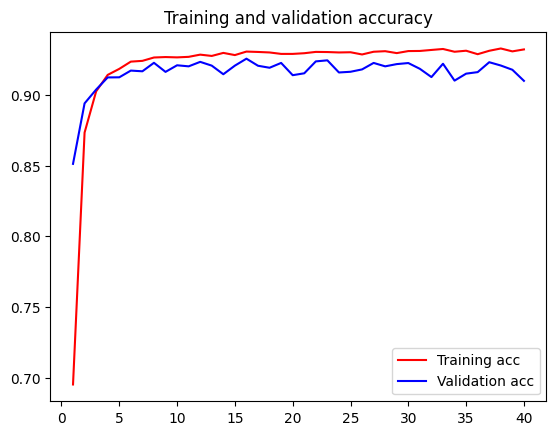
We then, define the class names



We then make predictions

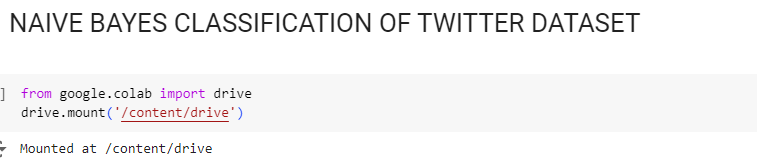


The accuracy plots -

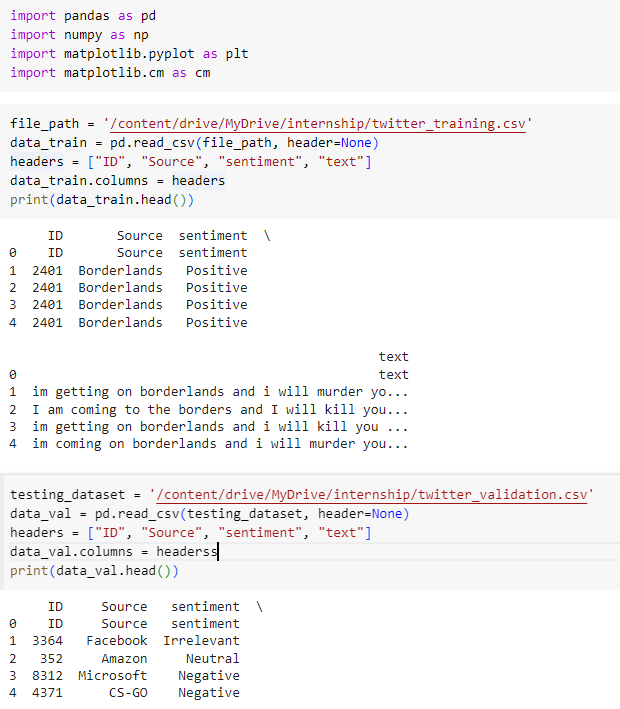


# Naïve Bayes Classification of twitter tweets

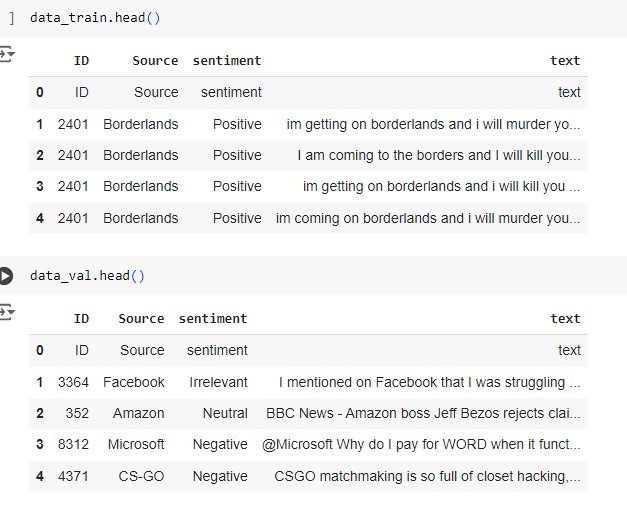
We first mount Google drive, where Dataset is present



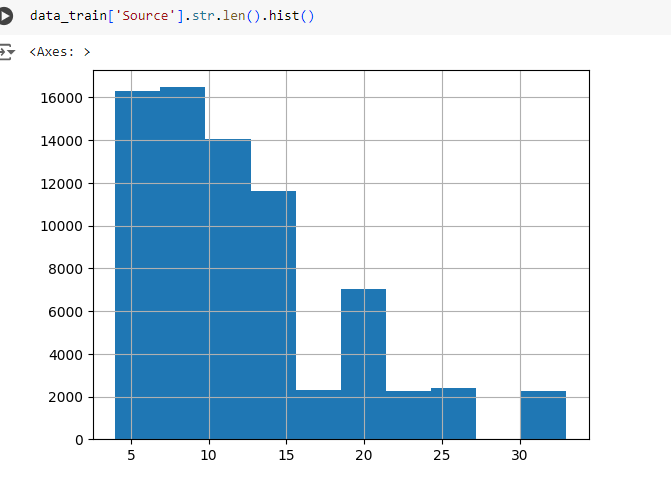
We import libraries and analyze data



After some pre processing like renaming column and text, final data is as follows –



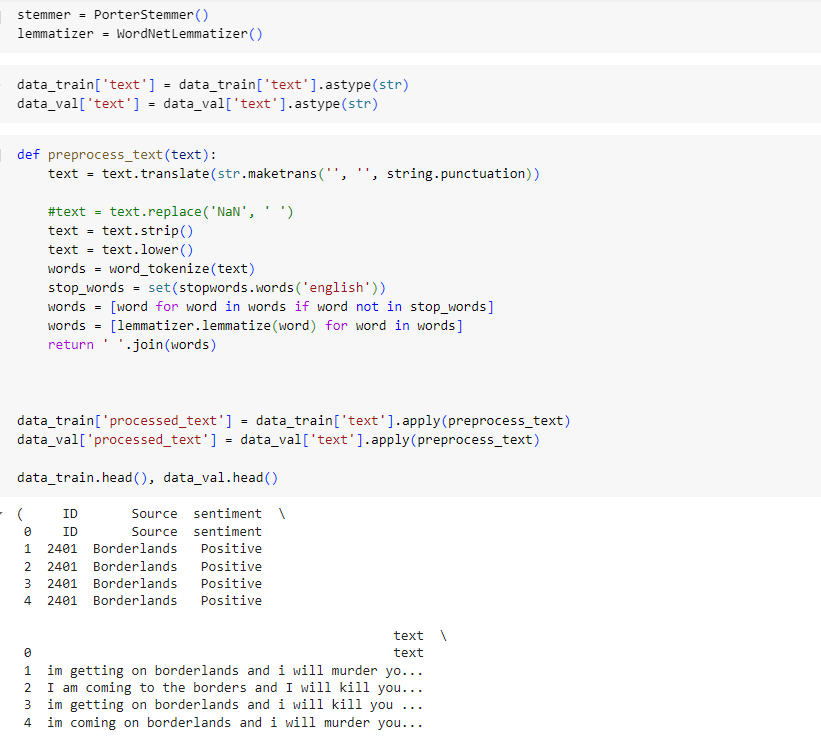
Histogram plot of various sources in dataset –

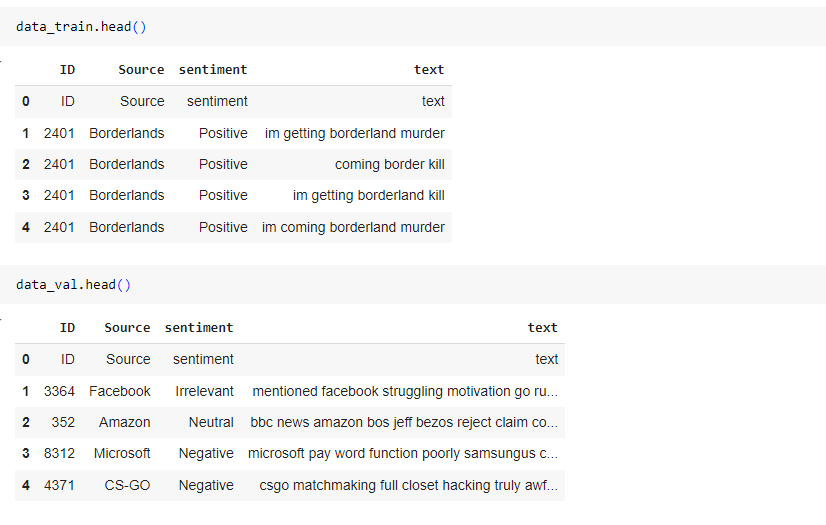


We then import nltk library to identify stopwords.



We then use PorterStemmer and WordLemmatizer to stem and lemmatize the text data, which means stemming removes last few characters from the words and lemmatizing converts word to base form





Thus, final data is as above.

Concatenate the data into one for easy Exploratory data analysis



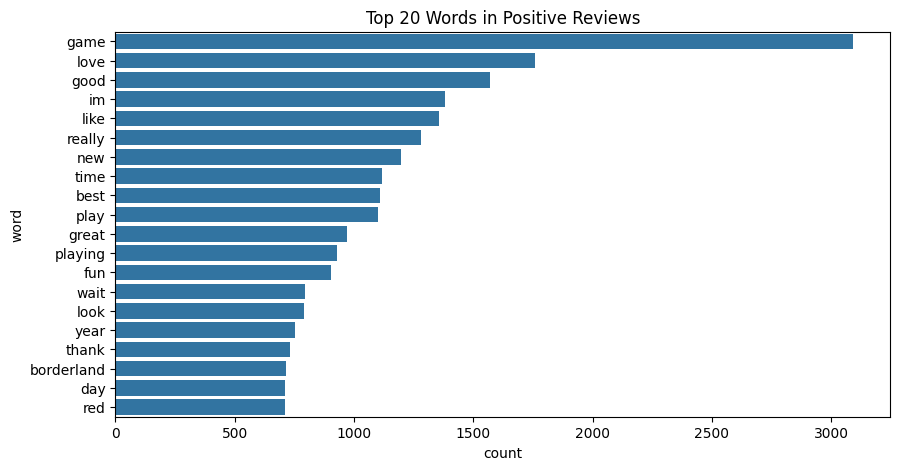


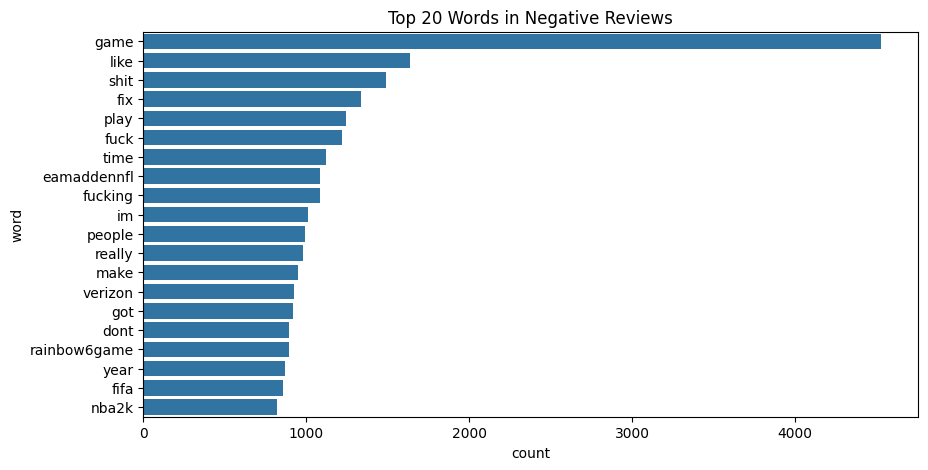


The word clouds of positive and negative words are as above.

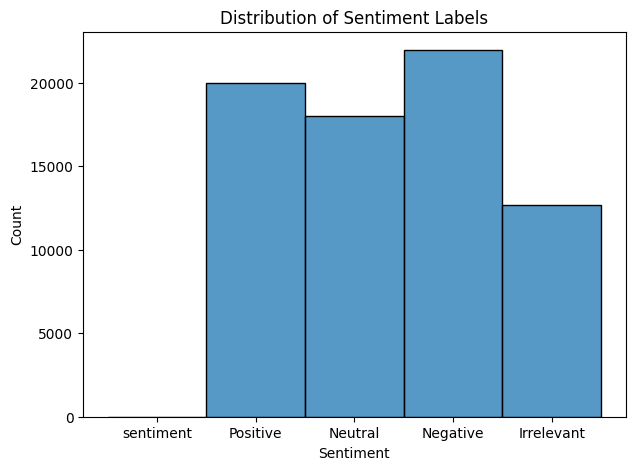
The bar plot of top 20 words is plotted –

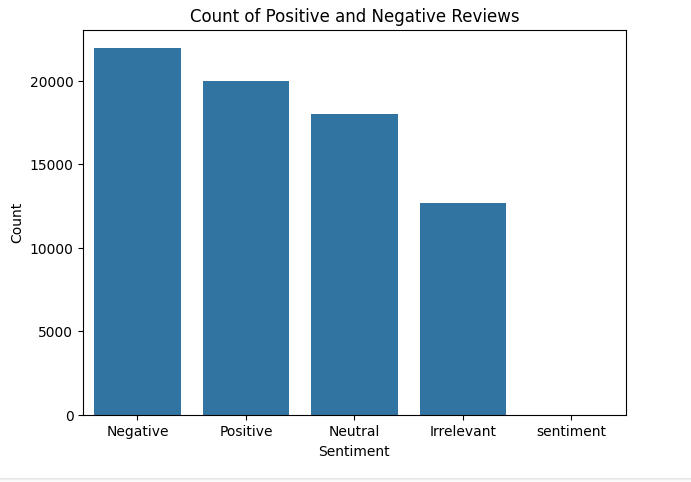






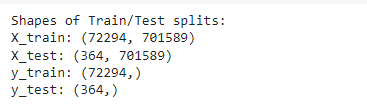
Sentiment distribution Histogram is then plotted





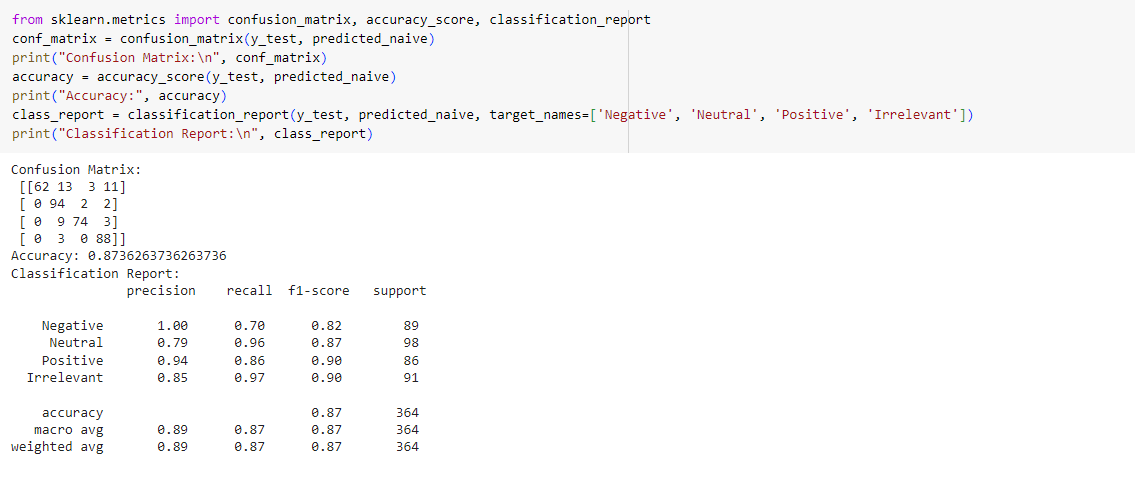
We then tokenize the words that is the text part of the dataset and apply TF-IDF vectorizer as below





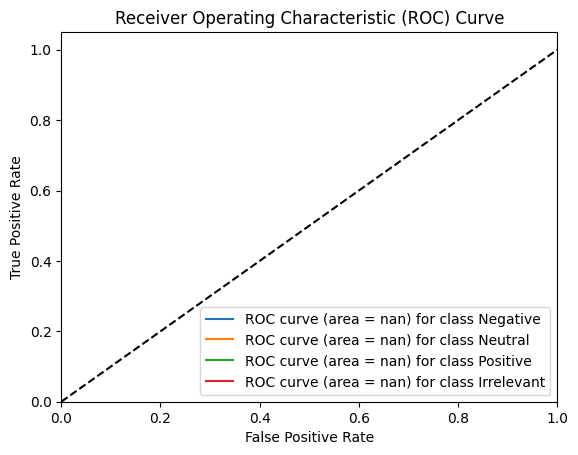
We then import the NaiveBayes MultinomialNB Model –

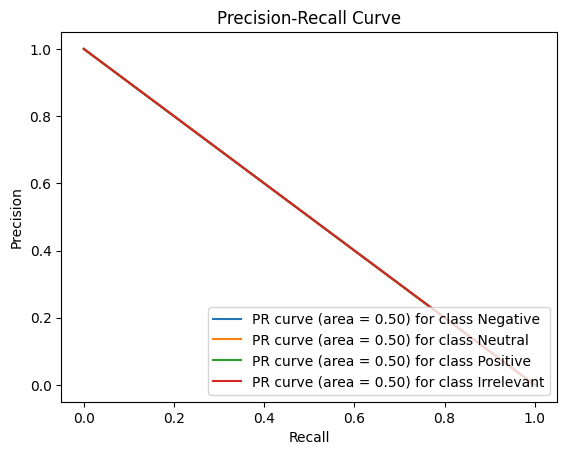




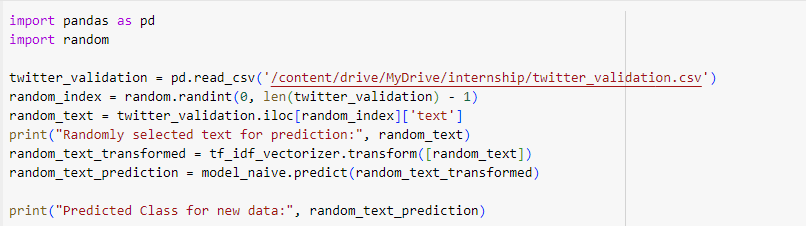
An accuracy of 87.36% is obtained.

Plotting the ROC and Precision-recall Curve as below –





A prediction sample is as shown below –



Randomly selected text for prediction: @NBA2K you guys gotta get Chris smoove as 2k manager. He commentates real basketball, has influence and has the most basketball knowledge when it comes to playing and understanding. Go after him he won’t reach out for a shite job, make the offer make sense

Predicted Class for new data: ['Neutral']