AI Course Project Report

for

Comparative analysis of different deep learning models for twitter sentiment analysis

Prepared by:

**Syed Salman Shah**

**Muhammad Asim Khan**

**Muhammad Adil Shahid**

2nd August 2021

[1. Abstract 1](#_Toc73905114)

[2. Introduction 1](#_Toc73905115)

[3. Data-preprocessing 1](#_Toc73905116)

[4. Feature Engineering 1](#_Toc73905117)

[5. Classification Algorithms 2](#_Toc73905118)

[6. Comparison & Performance Evaluation 5](#_Toc73905119)

[7. Conclusion 5](#_Toc73905120)

# Abstract

This project aims to develop classifiers and compare three different models for the classification of various tweets into either positive or negative sentiments. The dataset contains an equal distribution of 5000 positive and 5000 negative sentiments for the models to be trained on. Initially we preprocessed the data heavily by primarily using the NLTK library (Natural language tool kit) and then built three models which we trained and tested the data set on.

# Introduction

Twitter is heavily used across the globe by many devoted users who wish to share their experiences and opinions online; whether it be on politics, sports, gaming, lifestyle etc. However more increasingly than ever before more and more individuals are falling victim to online bullying and harassment which has led to serious mental issues being developed. Hence, we decided to build an Artificial Intelligent model which would detect certain negative comments and would censor/delete them to avoid unnecessary harm.

# Data-preprocessing

Data-preprocessing is the technique of extracting and cleaning data to make it suitable for training Machine Learning models so it transforms raw data into understandable and readable format. Firstly, various libraries are used such as NLTK, Pandas, NumPy, Sklearn, Matploblib, Tensorflow etc. The data is extracted from the train and test file where the data from each file is cleaned to the point where certain English key words remain which allows the model to associate certain key words to the sentiment during the training phase.

Tokenization splits a word into many different piece based on white spaces

We then cleaned all the punctuation by using string.punctuation function

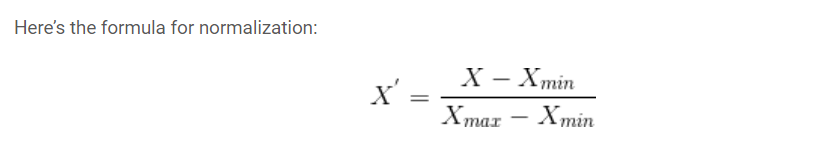
Next, we used regular expressions to delete repeating characters, @ signs, URLs etc

Once all the data is cleaned, we are ready for using it to train the model.

# Feature Engineering

Feature engineering is used to prepare proper input dataset and to improve the performance of machine learning models. Machine learning algorithms like ANN along with RandomForests and RNN need to be scaled before they are fed to the algorithm.

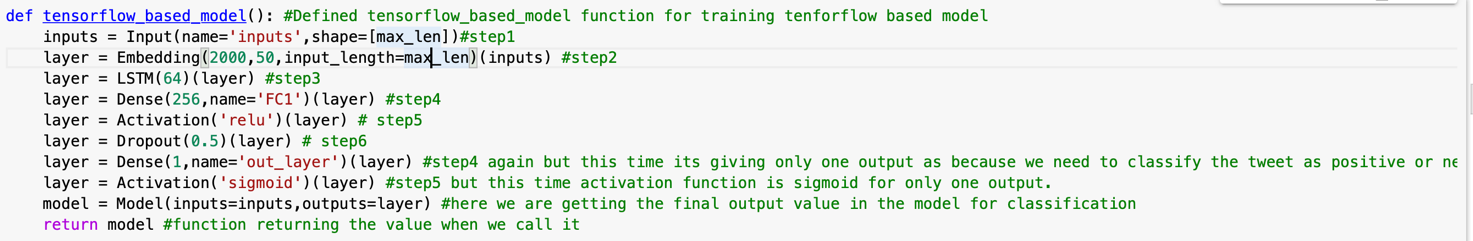
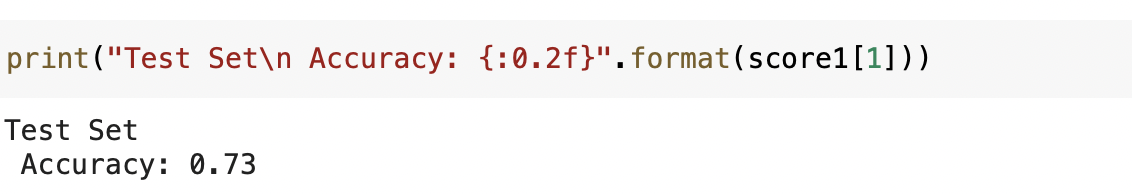
We used the Keras library, sequence.pad\_sequences that transforms each individual tweet into a 2d numpy array, any tweets which aren’t able to be translated into an array are padded to ensure to ensure they are also converted into arrays. These arrays are fed into the models for training purposes.

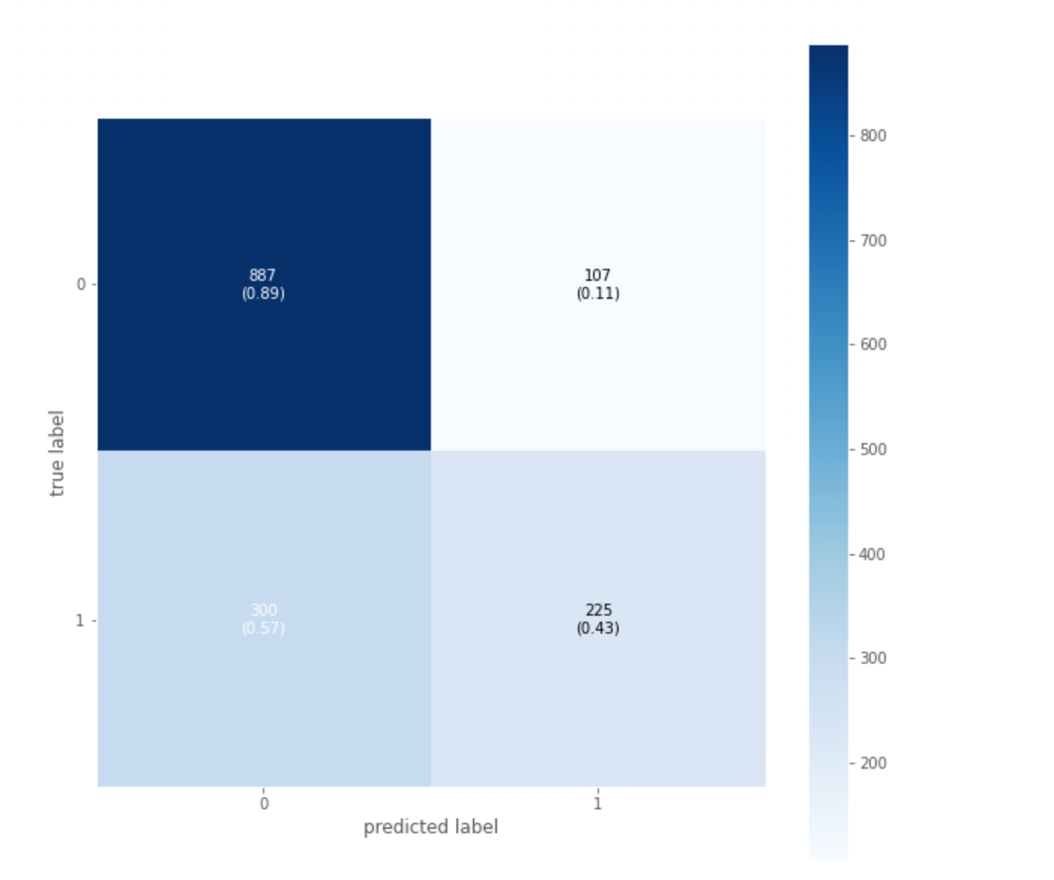


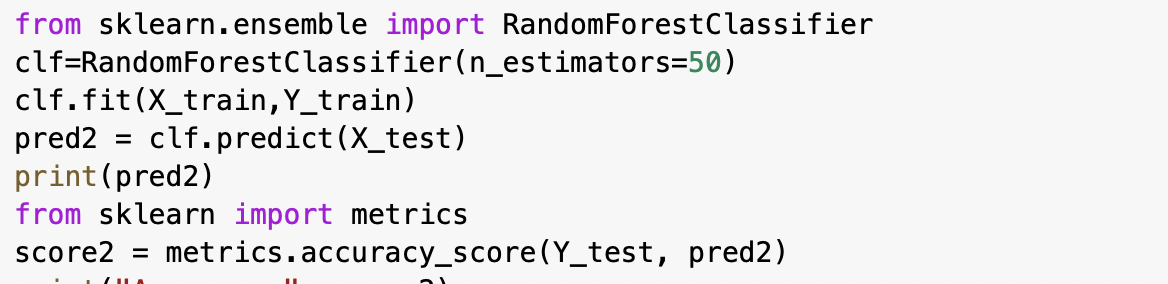
# Classification Algorithms

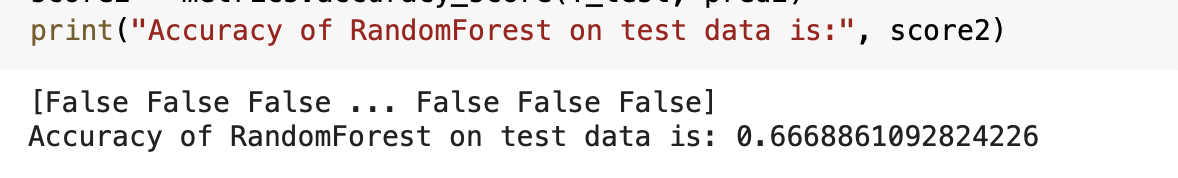
In order to classify the data into positive or negative sentiment, three classifiers have been used separately and then compared with each other. The models are namely LSTM (RNN), ANN and RandomForest.

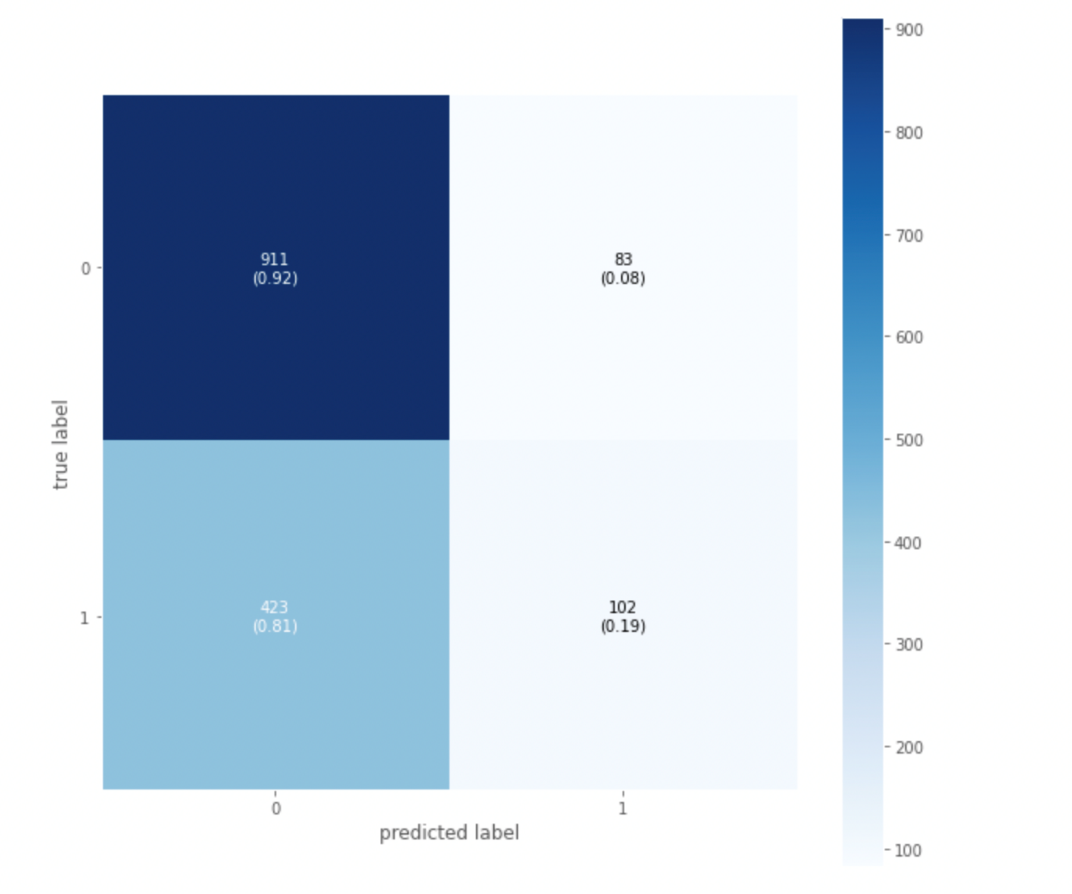
These models are then compared by plotting a graph of accuracy against each model using the matplot library.

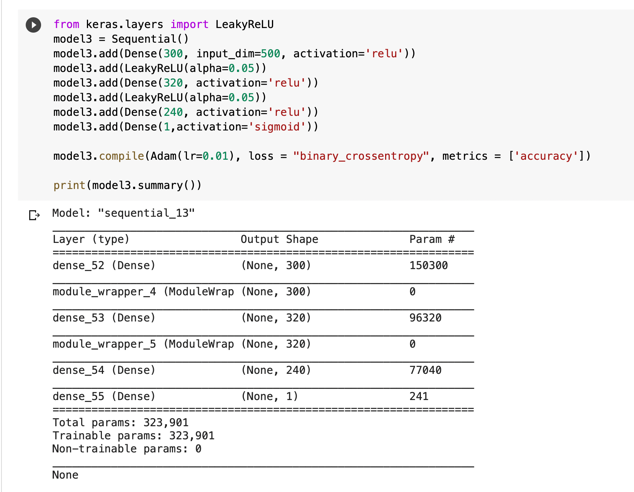
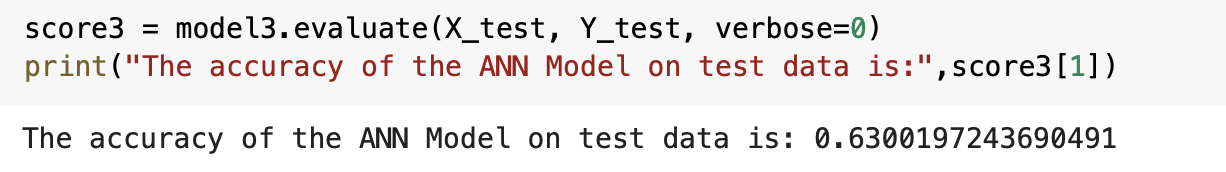
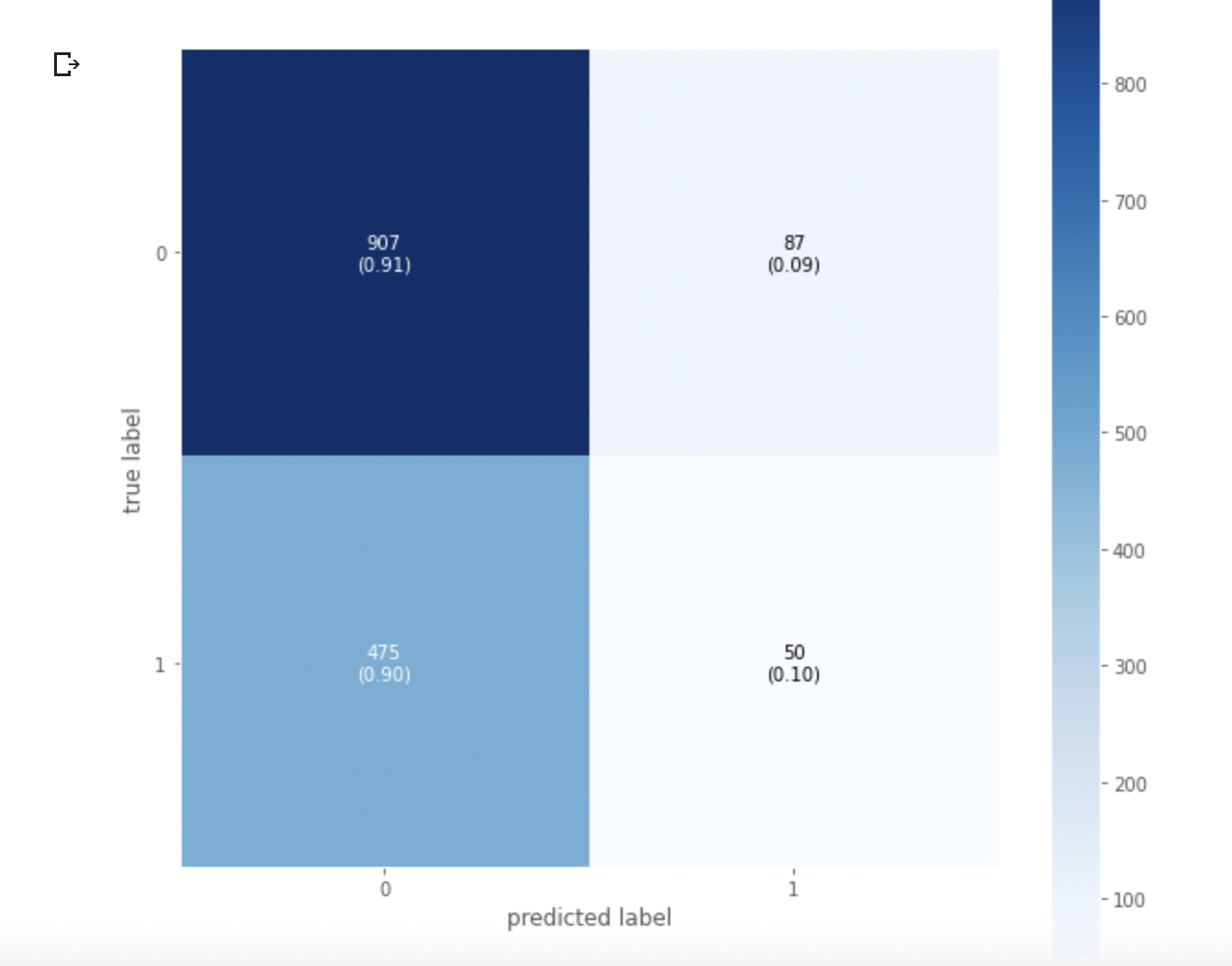
**First Model (RNN):**

****

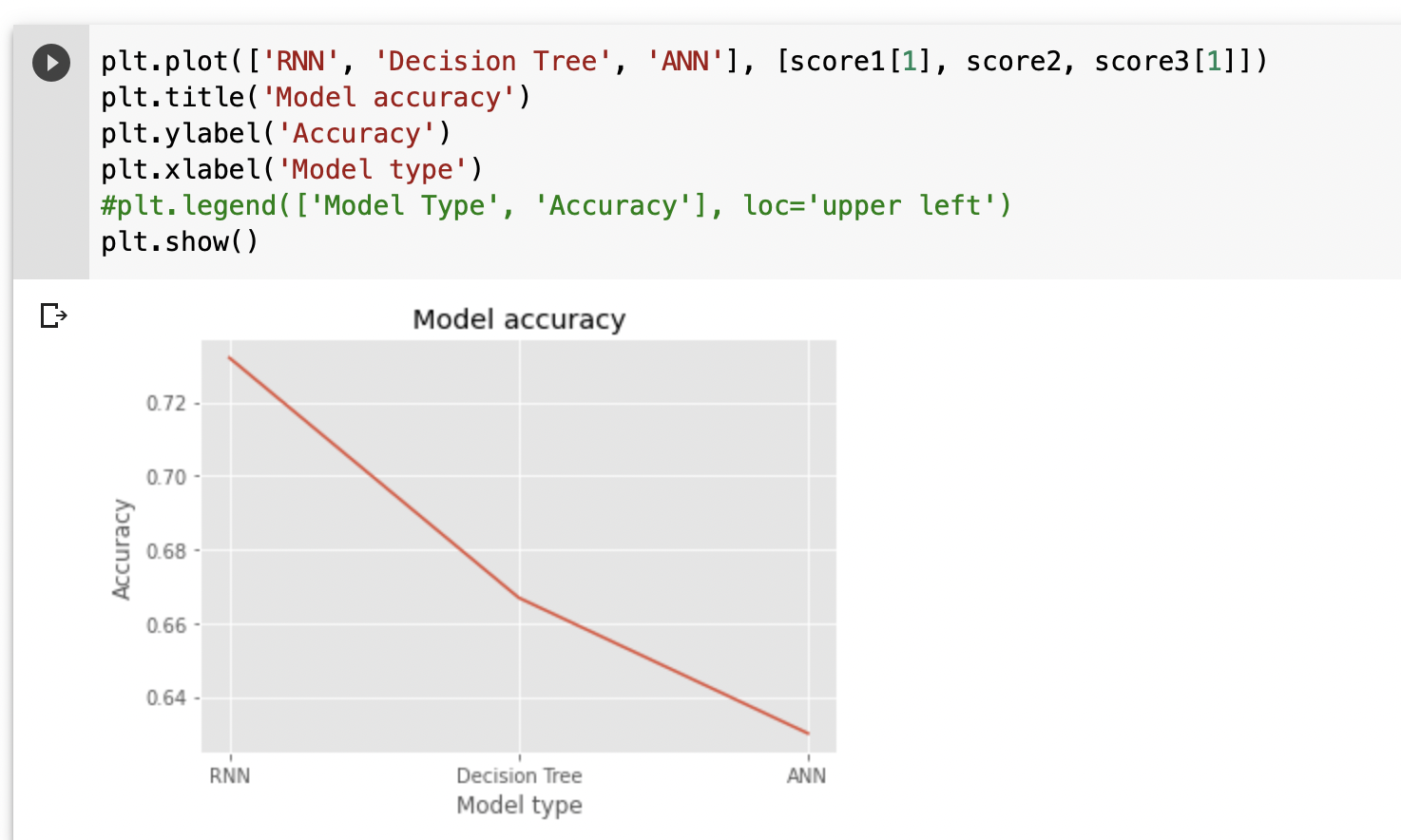
**Second Model (RandomForest):**

****

****

**Third Model (ANN):**

# Comparison & Performance Evaluation

****The graph computes accuracy by comparing the original label with the output of each respective model.

As we can see that the RNN model performed the best with the highest accuracy rate classifying each tweet into the right sentiment more than 7 times out of 10. While the decision tree’s accuracy was around the 67% mark and the ANN being the least accurate with a 63% correct classification rate.

# Conclusion

In conclusion we initially extracted the data from csv file retrieved the relevant information by combining the files. Preprocessed the data, performed feature engineering, built several models trained them and compared their accuracy. We concluded that the best model to use for sentiment analysis is RNN as it gave a high accuracy as compared to the others.