Product ratings prediction project

By: LAKSHITA KAIN

DATA SCIENCE INTERN

FLIP ROBO TECHNOLOGIES

PRODUCT RATINGS PREDICTION ANALYSIS

By: LAKSHITA KAIN

DATA SCIENTIST(INTERN)

FLIPROBO TECHNOLOGIES

INTRODUCTION

In an ecommerce-driven world where customers can't physically experience products before purchasing, many consumers turn to online product reviews. For any company that exists in the digital space, online reviews are critically important when it comes to winning business and maintaining a positive reputation.

In today's web-based world, virtually everyone is reading online reviews. In fact, 91% of people read them and 84% trust them as much as they would a personal recommendation. The effects of reviews are measurable, too. The average customer is willing to spend 31% more on a retailer that has excellent reviews.

Negative reviews can carry as much weight as positive ones. One study found that 82% of those who read online reviews specifically seek out negative reviews. Research indicates that users spend five times as long on sites when interacting with negative reviews, with an 85% increase in conversion rate.

Customers like to see lots of reviews. A single review with a few positive words makes up an opinion, but a few dozen that say the same thing make a consensus. The more reviews, the better, and one study found that consumers want to see at least 40 reviews to justify trusting an average star rating. However, a few reviews are still better than no reviews

PROBLEM STATEMENT

Sentiment Analysis is the task of analyzing people’s opinions in textual data (e.g., product reviews, movie reviews, or tweets), and extracting their polarity and viewpoint. The task can be cast as either a binary or a multi-class problem

Identifying user review ratings based off of sentiment analysis techniques is an important topic in machine learning, computer vision, and data science. In this paper we build a model to predict product ratings based off of rating text using a bag-of-words model. The two models tested utilized unigrams and bigrams.

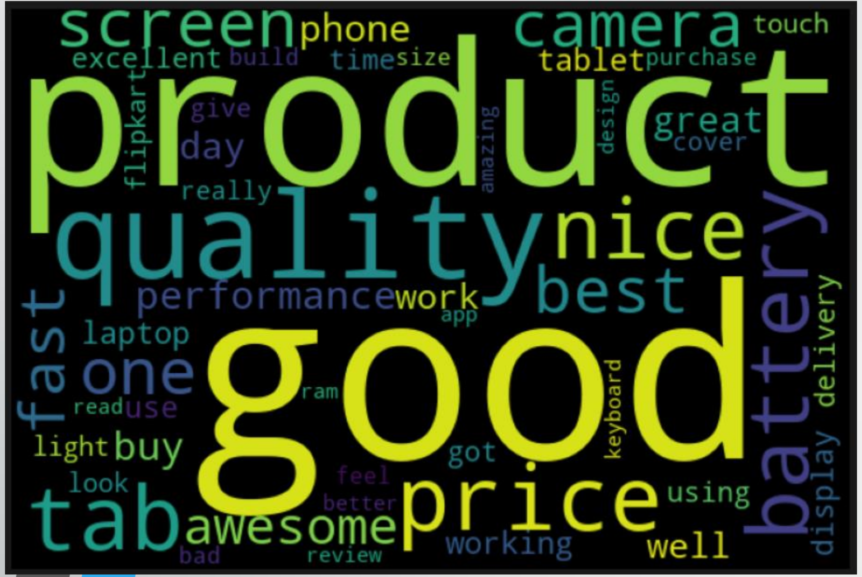
Our dataset consists of 5 labels of Ratings of a product from 1 star to 5 stars (1 being worst rating and 5 being the highest rating given to a product). This project aims to implement various Machine Learning algorithms and deep learning algorithms like Multilayer perceptron(MLP), Long Short Term Memory Networks, Multinomial Naïve Baiyes, Logistic Regression, Random Forest Classifier , Linear SVC and Adaptive Boosting. Data used in this project are online product reviews collected from “flipkart.com”.

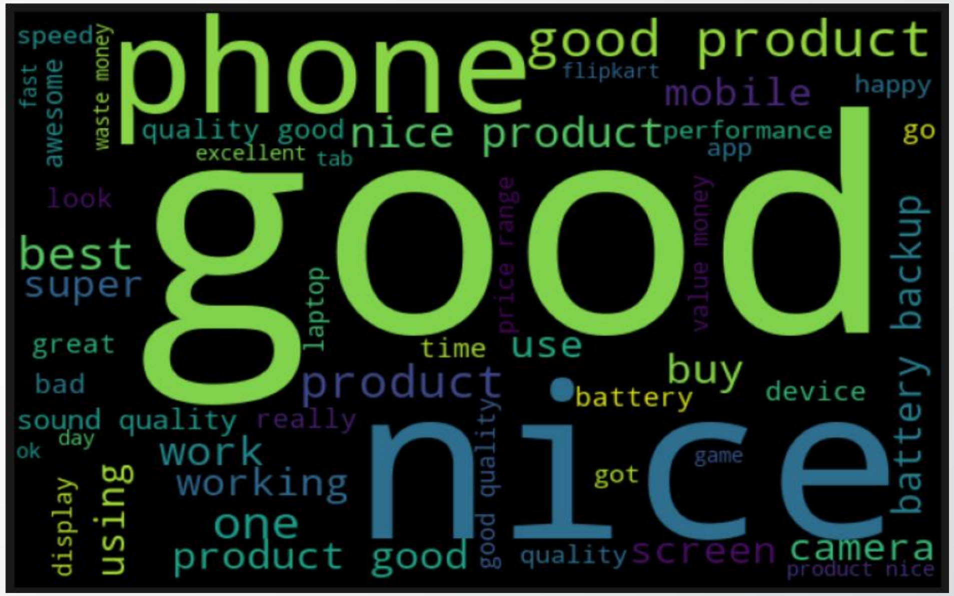
DATA COLLECTION

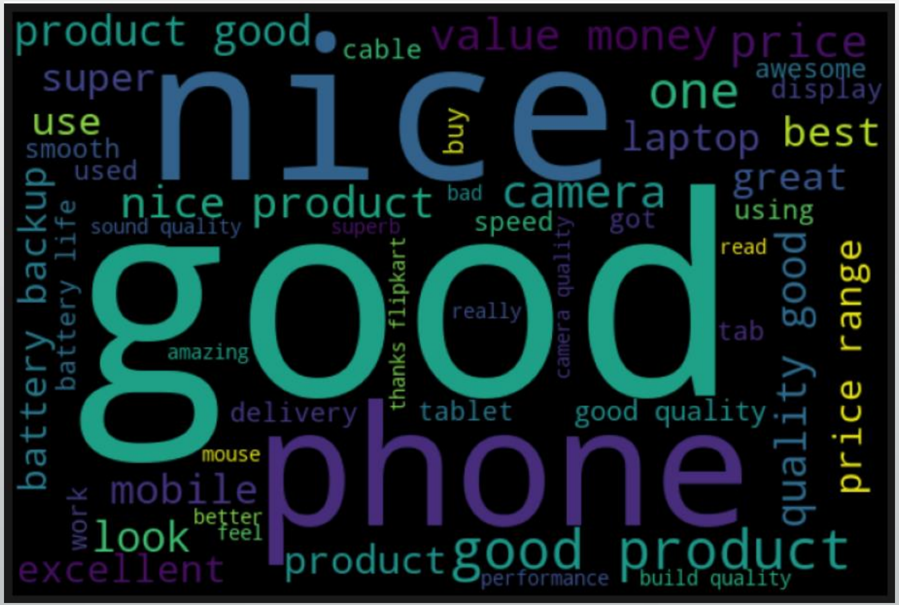
Data has been scrapped using Selenium webdriver with python. Selenium is a powerful tool for controlling web browsers through programs and performing browser automation.

Our data has been retrieved from https://www.flipkart.com which consists of Electronic Product such as Mobile Smartphones, Smartwatches, Laptops, Gaming Accessories, Desktop PC, Speakers, Earphones/Headphones etc. retained information consists of Number of Reviews, Number of Ratings, Review of a buyer, Average Rating of the product.

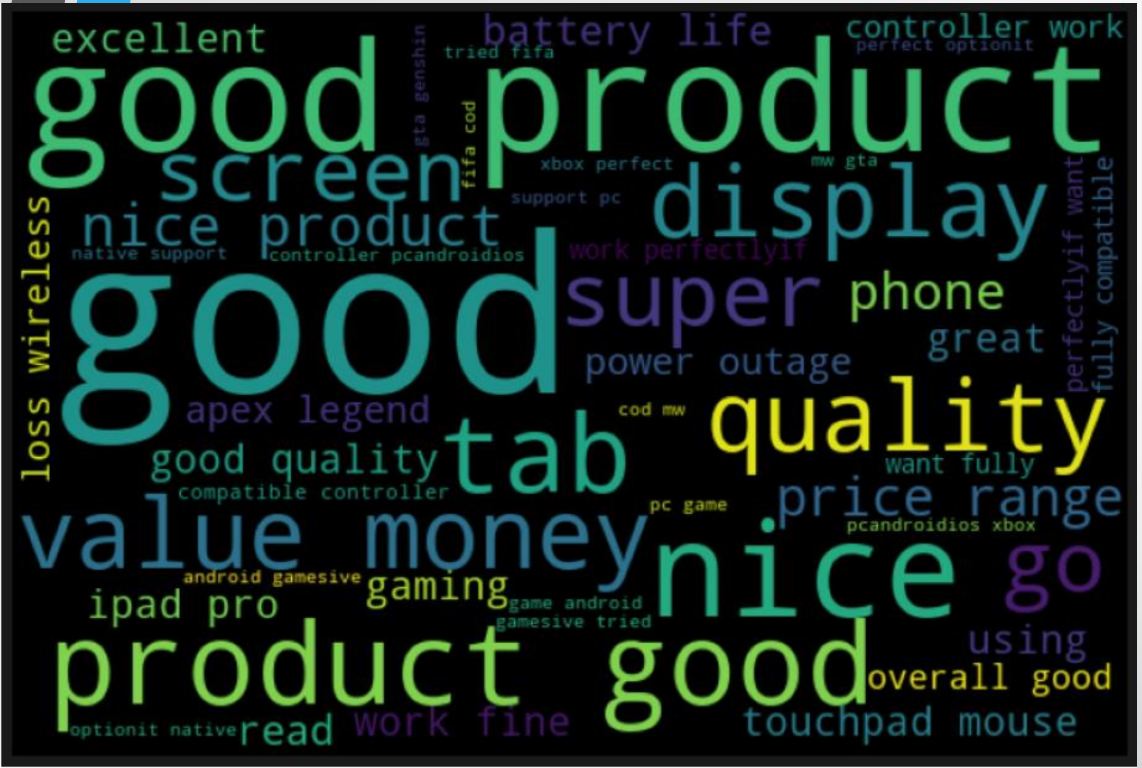
VISUALIZATION USING WORDCLOUD

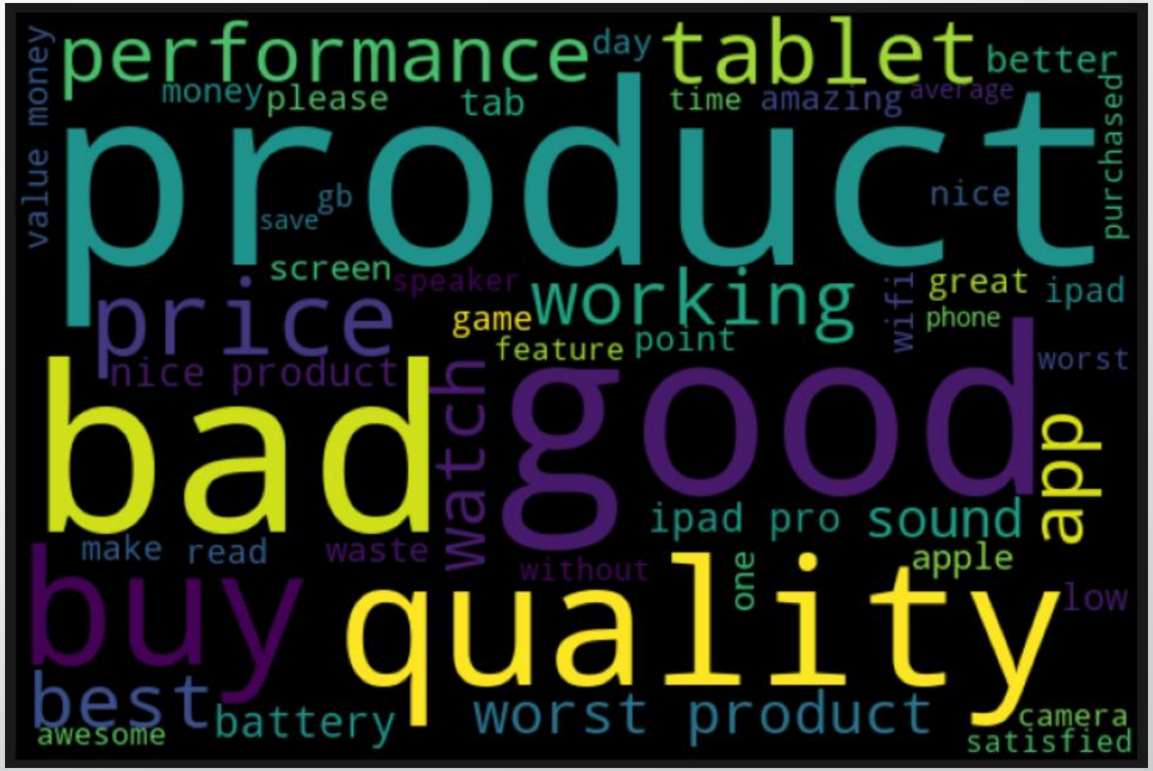




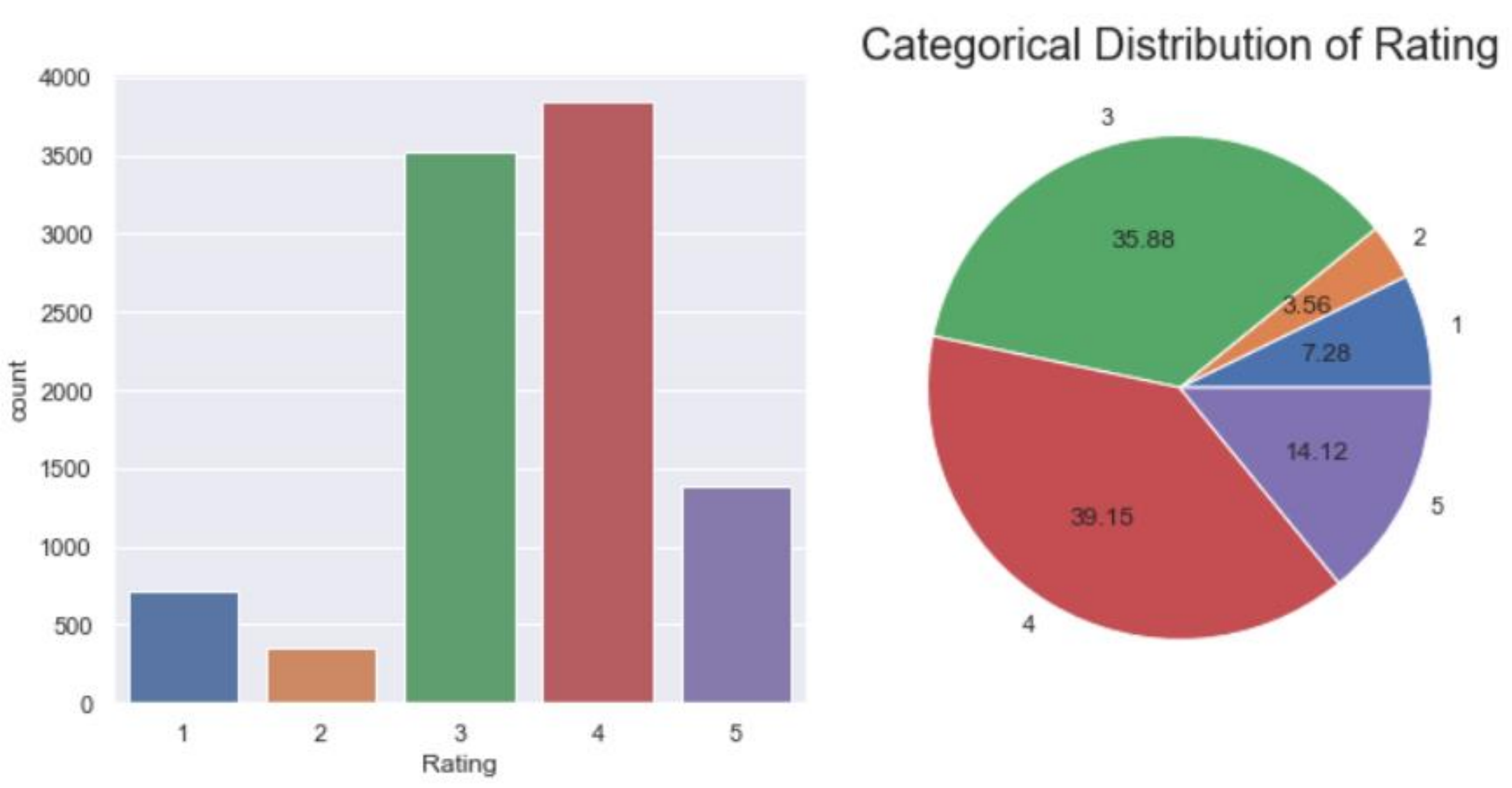


VISUALIZATION USING WORDCLOUD





DISTRIBUTION OF OUR TARGET VARIABLE

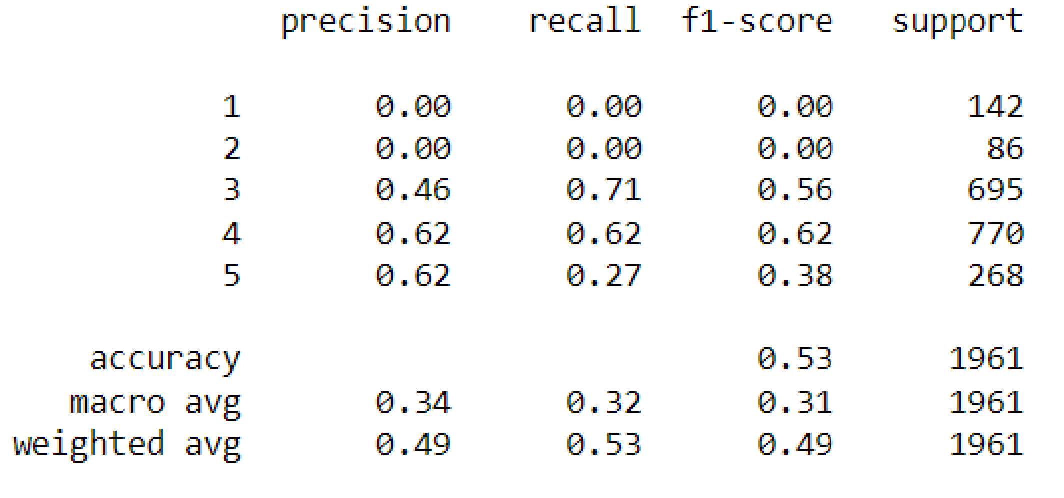


DATA NORMALIZATION

- Lower casing of all the sentences.
- Tokenization : converting sentences into separate word tokens.
- Removal of numeric, special characters, symbols etc.
- Removal of stop words from English (I, they, we, are , they, a , so etc.)
- Lemmatization: reduces the inflected words properly ensuring that the root word belongs to English language.
- TFID vectorization : provide adequate weight to a word in proportion of the impact it has on the meaning of a sentence. The score is a product of 2 independent scores, term frequency(TF) and inverse document frequency (IDF)
- Embedding Padding : In our LSTM model pre-padding is used. “embedded_docs” is the input list of sentences which is one hot encoded and every sentence is made of the same length.

Logistic Regression

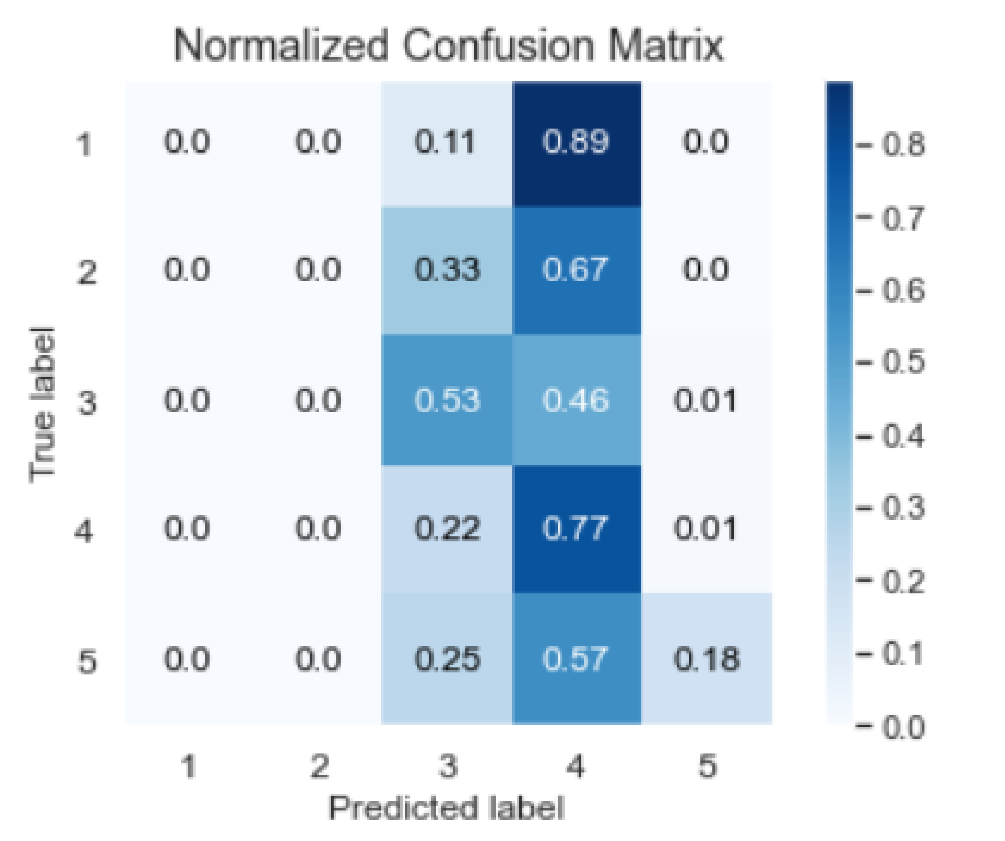
Accuracy Score : 53.08%
Hamming Loss : 0.4691
Cross – Validation Score : 52.839%



```
              precision    recall  f1-score   support

           1       0.00      0.00      0.00       142
           2       0.00      0.00      0.00        86
           3       0.46      0.71      0.56       695
           4       0.62      0.62      0.62       770
           5       0.62      0.27      0.38       268

    accuracy                           0.53      1961
   macro avg       0.34      0.32      0.31      1961
weighted avg       0.49      0.53      0.49      1961
```



Normalized Confusion Matrix

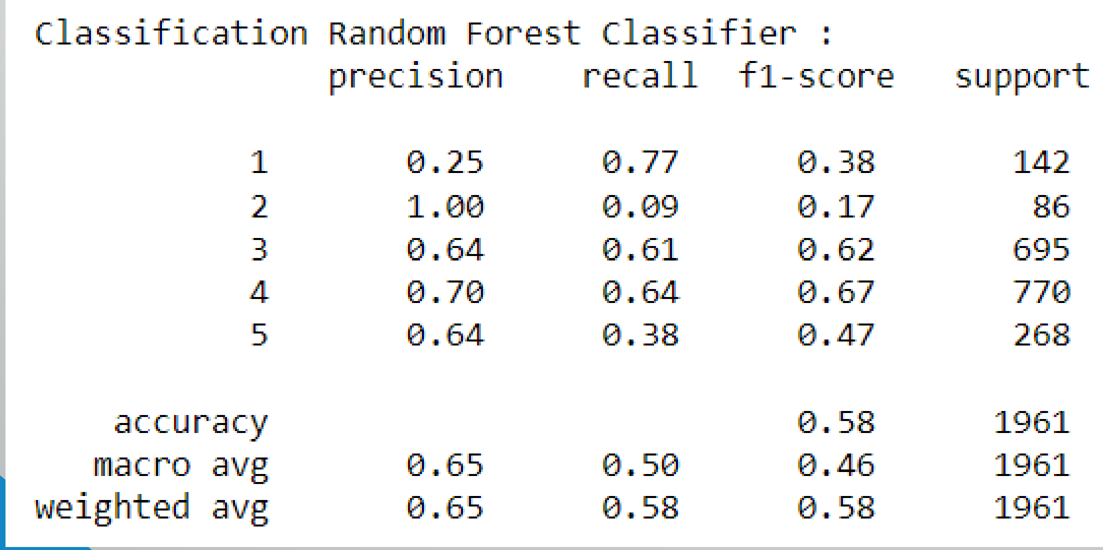
True label \ Predicted label	1	2	3	4	5
1	0.0	0.0	0.11	0.89	0.0
2	0.0	0.0	0.33	0.67	0.0
3	0.0	0.0	0.53	0.46	0.01
4	0.0	0.0	0.22	0.77	0.01
5	0.0	0.0	0.25	0.57	0.18

Random Forest Classifier

Accuracy Score : 58.90%

Hamming Loss : 0.41106

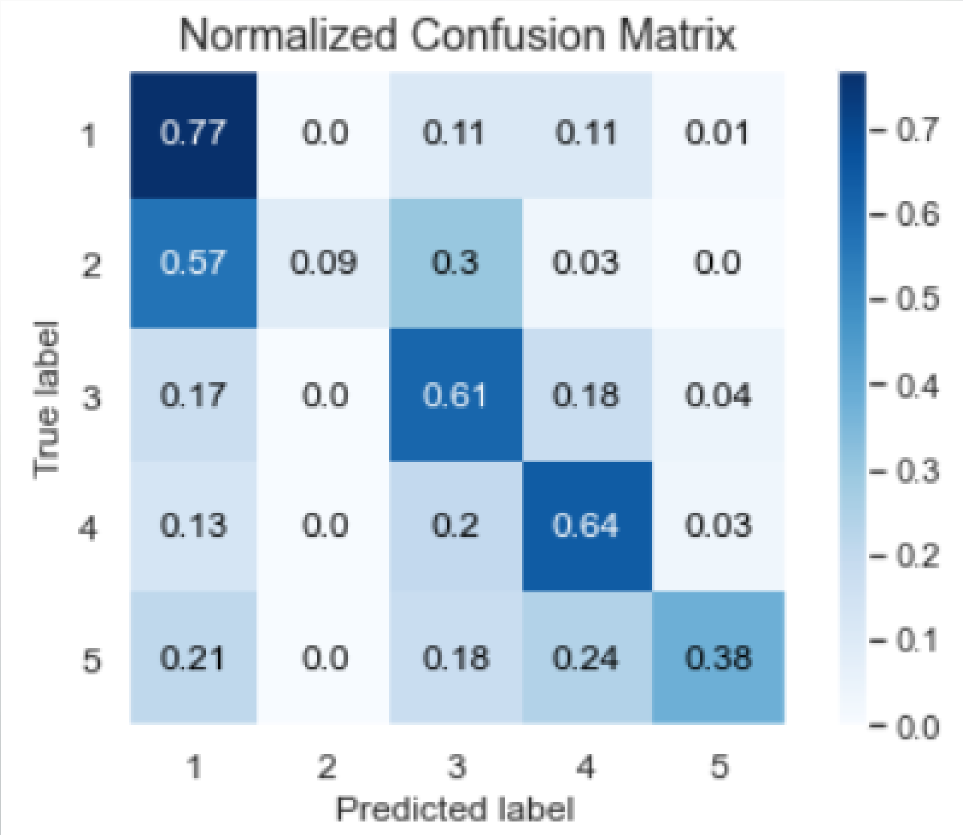
Cross – Validation Score: 56.62%



```
Classification Random Forest Classifier :
              precision    recall  f1-score   support

           1       0.25      0.77      0.38       142
           2       1.00      0.09      0.17        86
           3       0.64      0.61      0.62       695
           4       0.70      0.64      0.67       770
           5       0.64      0.38      0.47       268

    accuracy                           0.58      1961
   macro avg       0.65      0.50      0.46      1961
weighted avg       0.65      0.58      0.58      1961
```



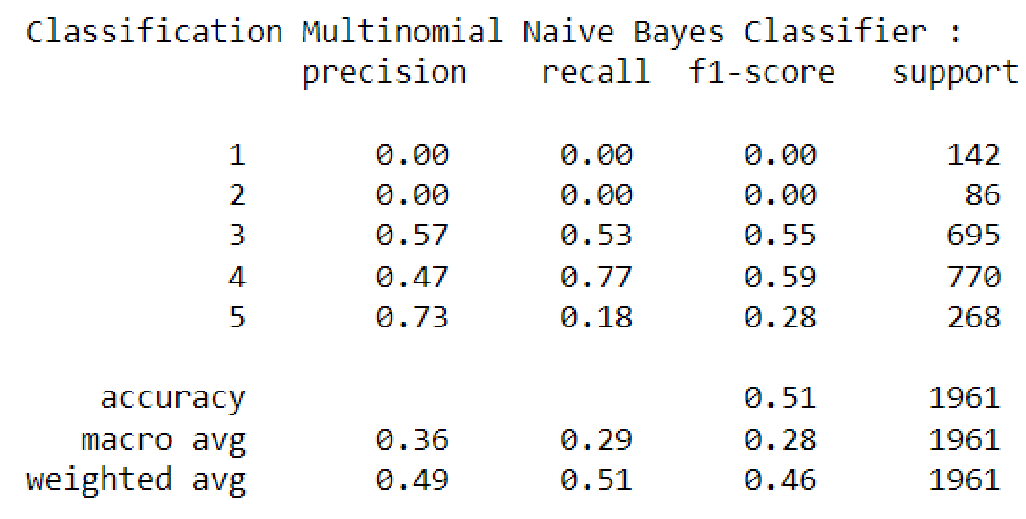
	1	2	3	4	5
1	0.77	0.0	0.11	0.11	0.01
2	0.57	0.09	0.3	0.03	0.0
3	0.17	0.0	0.61	0.18	0.04
4	0.13	0.0	0.2	0.64	0.03
5	0.21	0.0	0.18	0.24	0.38

Multinomial Naive Bayes

Accuracy Score : 52.629%

Hamming Loss : 0.41101

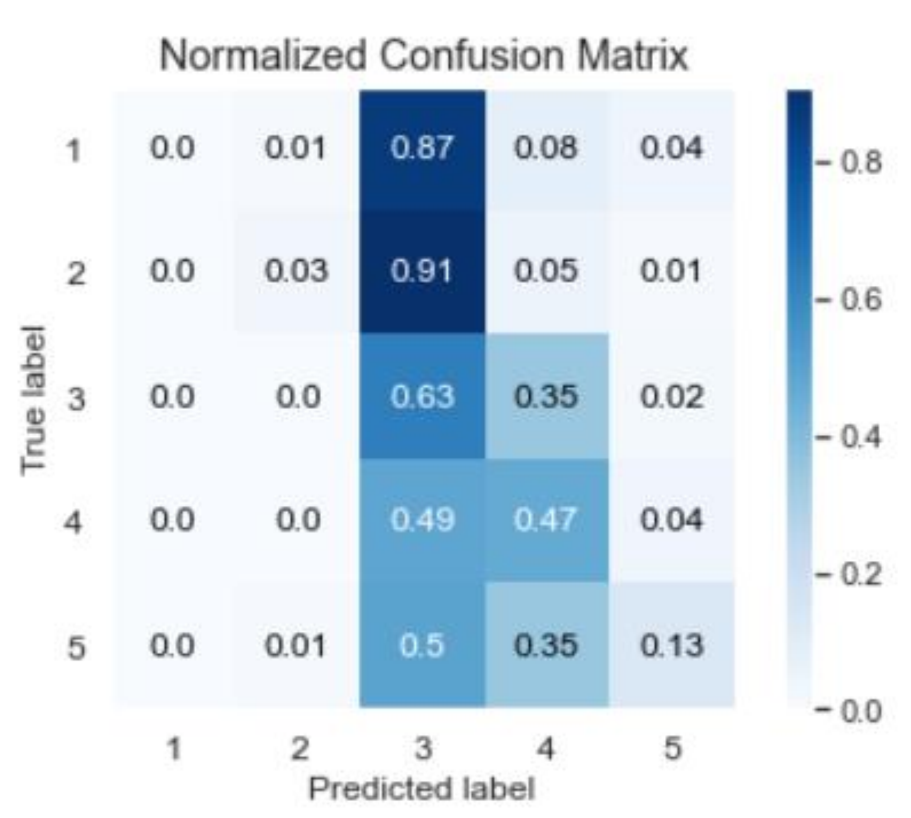
Cross – Validation Score: 49.69%



```
Classification Multinomial Naive Bayes Classifier :
              precision    recall  f1-score   support

           1       0.00      0.00      0.00       142
           2       0.00      0.00      0.00        86
           3       0.57      0.53      0.55       695
           4       0.47      0.77      0.59       770
           5       0.73      0.18      0.28       268

    accuracy                           0.51      1961
   macro avg       0.36      0.29      0.28      1961
weighted avg       0.49      0.51      0.46      1961
```



Normalized Confusion Matrix

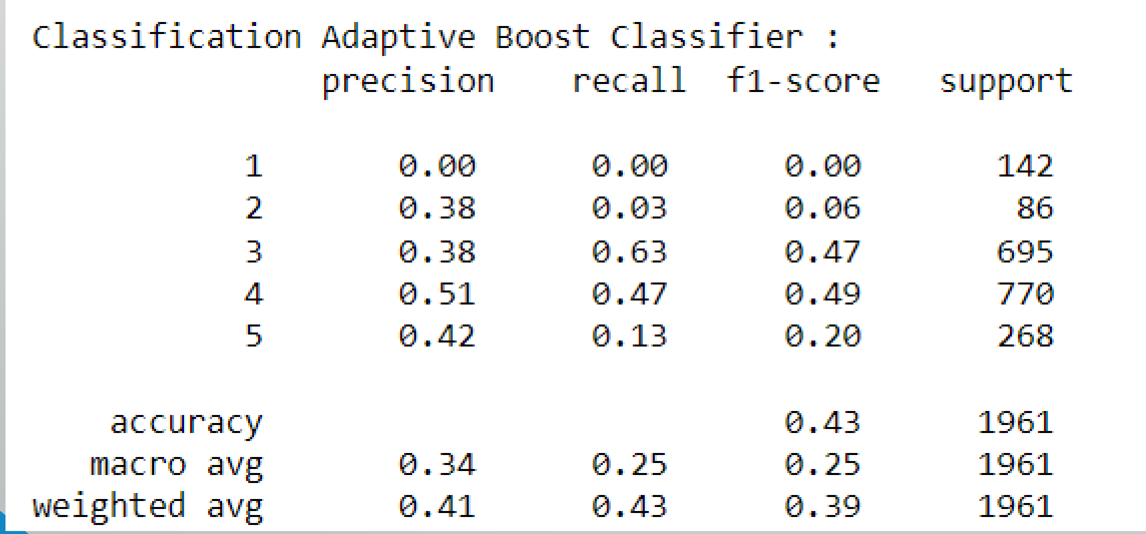
True label \ Predicted label	1	2	3	4	5
1	0.0	0.01	0.87	0.08	0.04
2	0.0	0.03	0.91	0.05	0.01
3	0.0	0.0	0.63	0.35	0.02
4	0.0	0.0	0.49	0.47	0.04
5	0.0	0.01	0.5	0.35	0.13

Adaptive Boosting

Accuracy Score : 48.75%

Hamming Loss : 0.51249

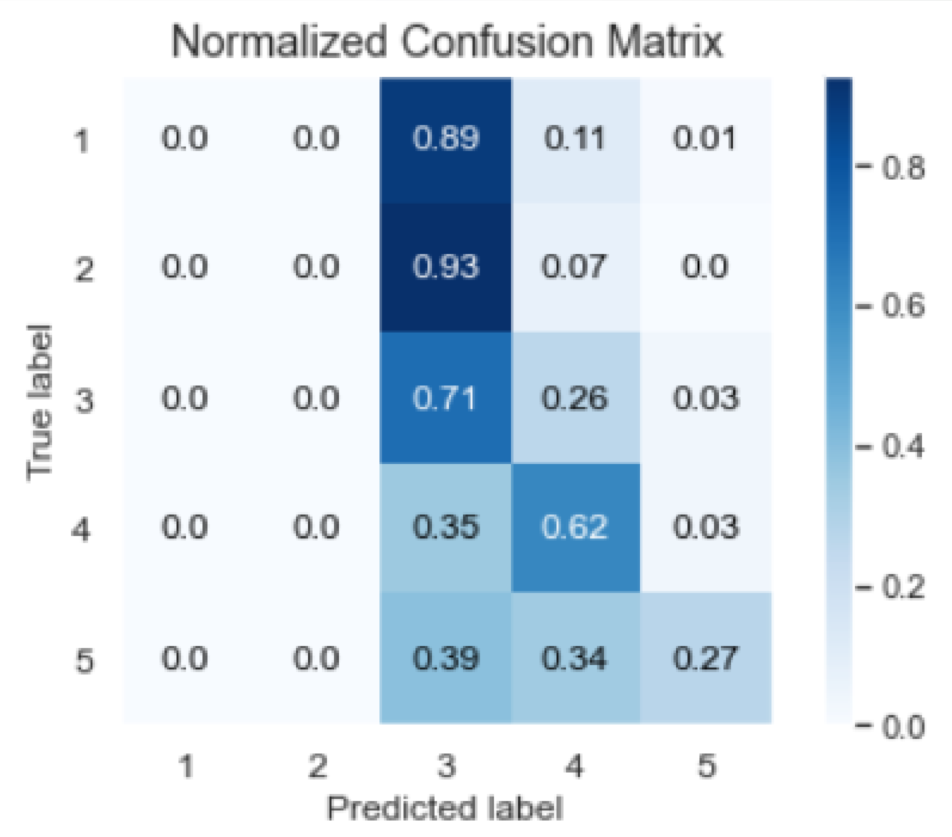
Cross – Validation Score: 42.75%



```
Classification Adaptive Boost Classifier :
              precision    recall  f1-score   support

           1       0.00      0.00      0.00       142
           2       0.38      0.03      0.06        86
           3       0.38      0.63      0.47       695
           4       0.51      0.47      0.49       770
           5       0.42      0.13      0.20       268

    accuracy                           0.43      1961
   macro avg       0.34      0.25      0.25      1961
weighted avg       0.41      0.43      0.39      1961
```



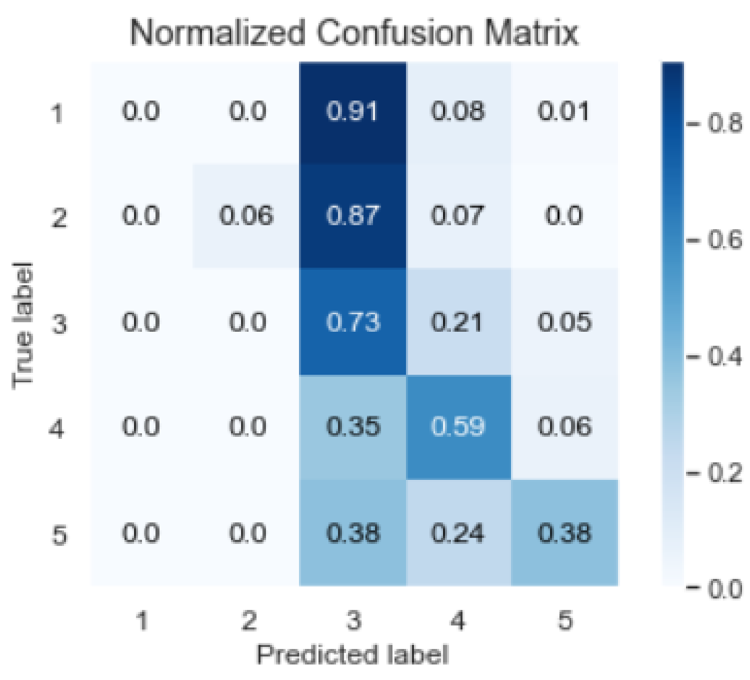
True label \ Predicted label	1	2	3	4	5
1	0.0	0.0	0.89	0.11	0.01
2	0.0	0.0	0.93	0.07	0.0
3	0.0	0.0	0.71	0.26	0.03
4	0.0	0.0	0.35	0.62	0.03
5	0.0	0.0	0.39	0.34	0.27

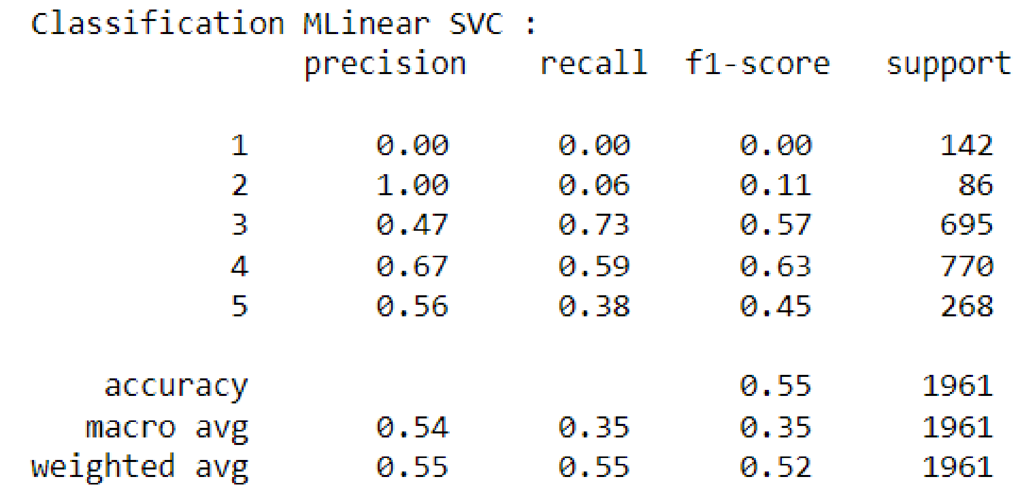
Linear SVC

Accuracy Score : 58.08%

Hamming Loss : 0.41917

Cross – Validation Score: 54.50%





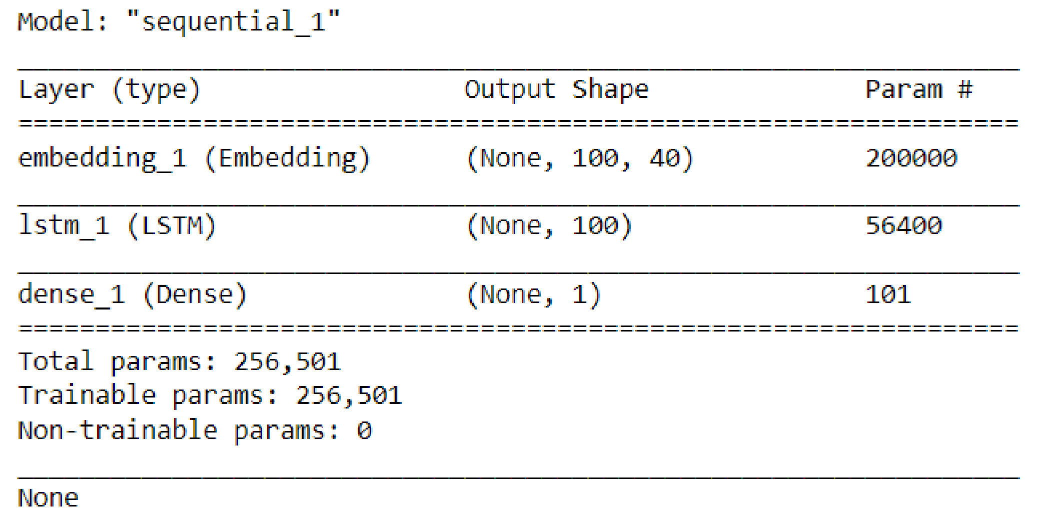
```
Classification MLinear SVC :
              precision    recall  f1-score   support

           1       0.00      0.00      0.00       142
           2       1.00      0.06      0.11        86
           3       0.47      0.73      0.57       695
           4       0.67      0.59      0.63       770
           5       0.56      0.38      0.45       268

    accuracy                           0.55      1961
   macro avg       0.54      0.35      0.35      1961
weighted avg       0.55      0.55      0.52      1961
```

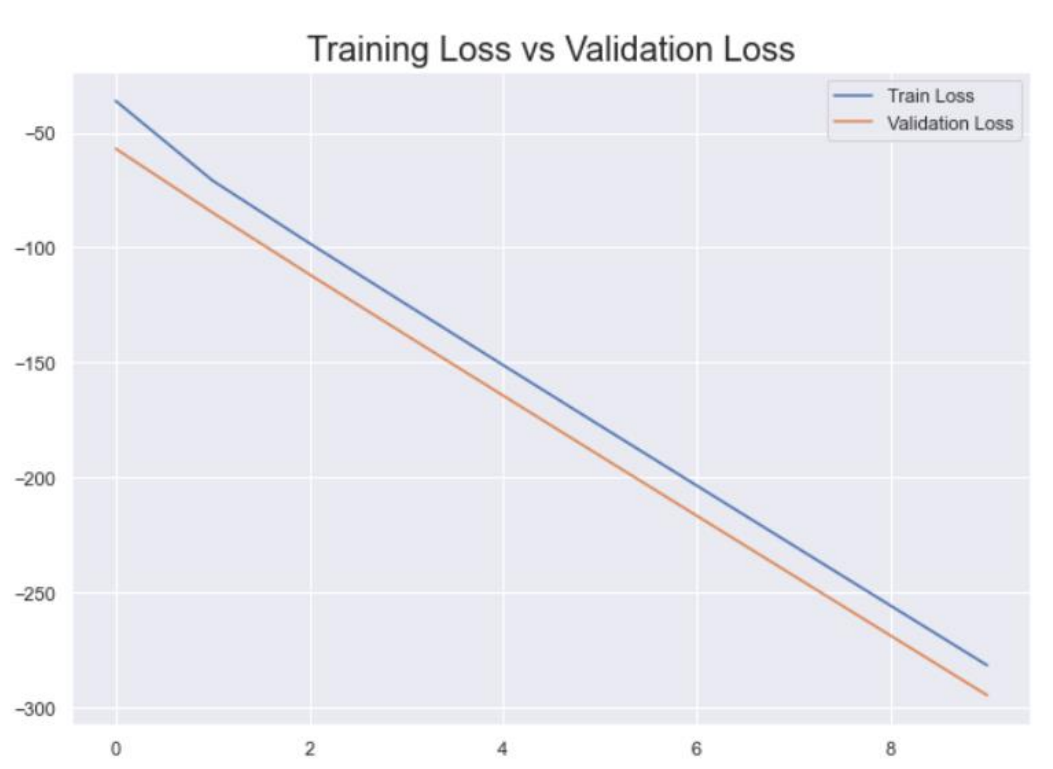
Long Short Term Memory

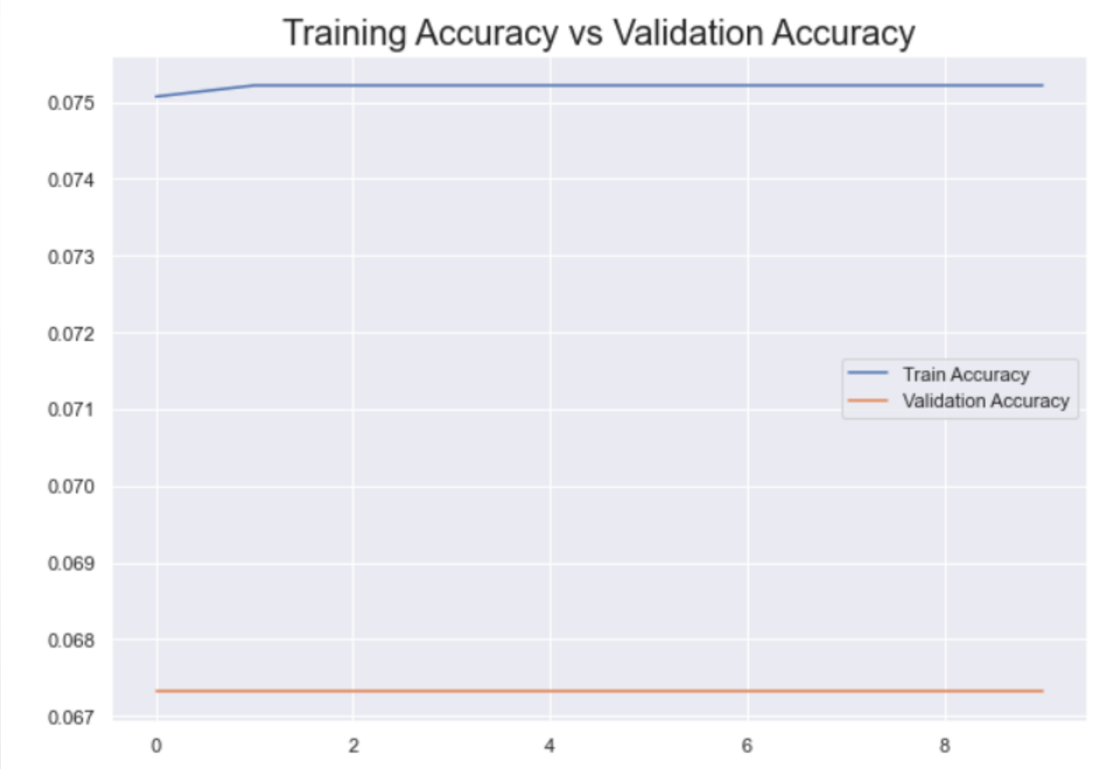
Tradition neural networks suffer from short-term memory. Also, a big drawback is the vanishing gradient problem. (While backpropagation the gradient becomes so small that it tends to 0 and such a neuron is of no use in further processing.) LSTMs efficiently improves performance by memorizing the relevant information that is important and finds the pattern.



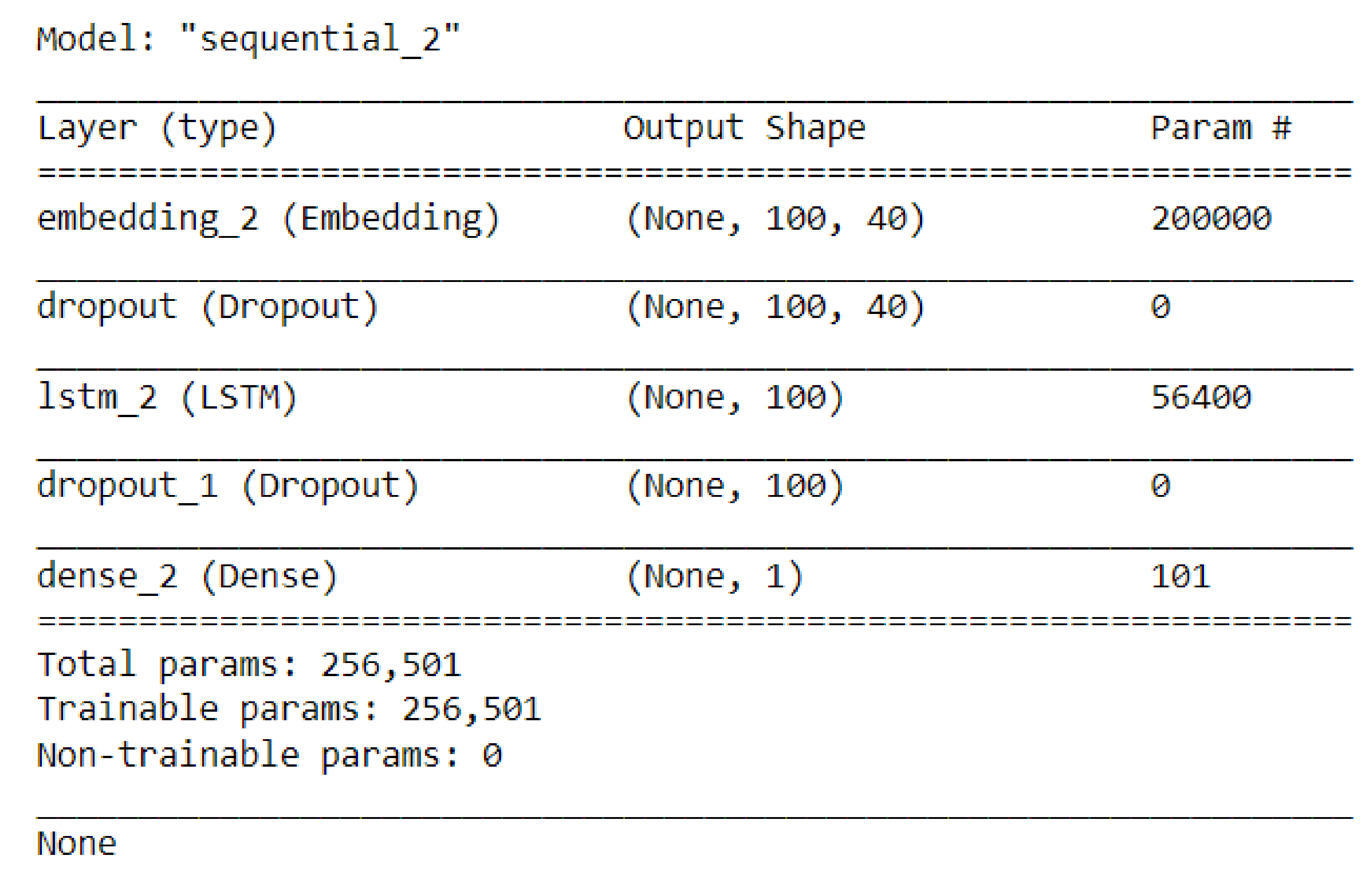
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 100, 40)           200000
_________________________________________________________________
lstm_1 (LSTM)                (None, 100)               56400
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 101
=================================================================
Total params: 256,501
Trainable params: 256,501
Non-trainable params: 0
_________________________________________________________________
None
```

Long Short Term Memory



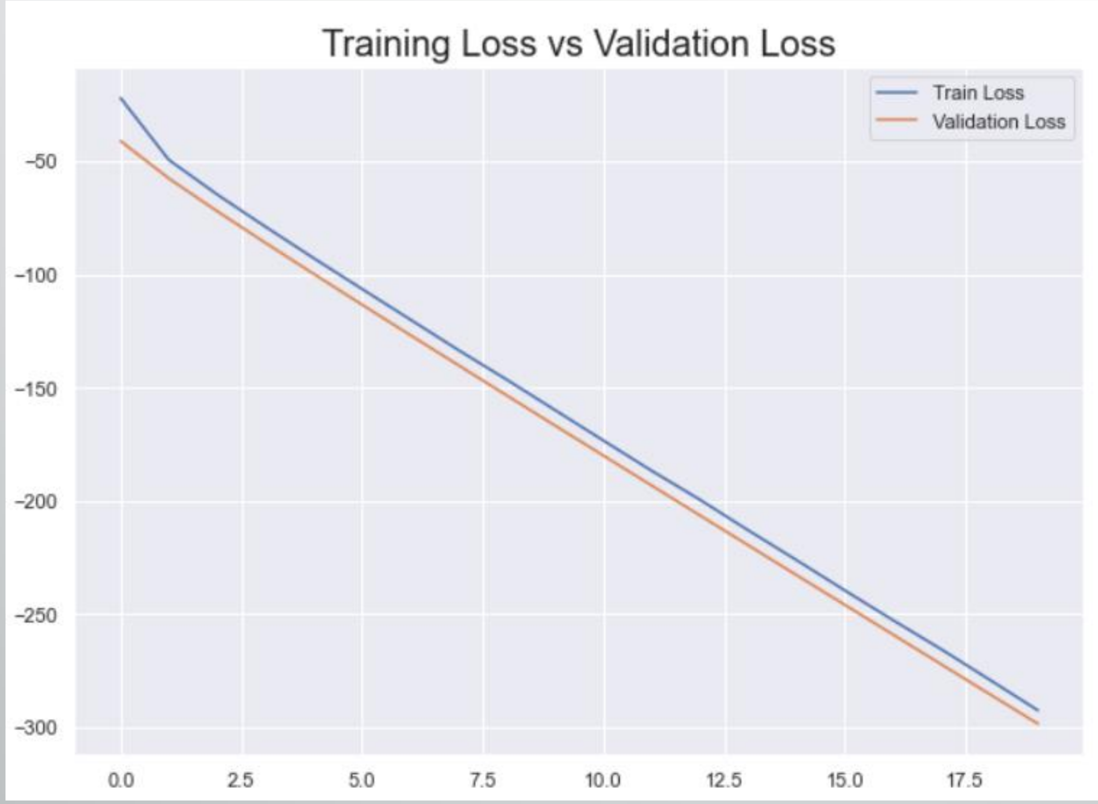


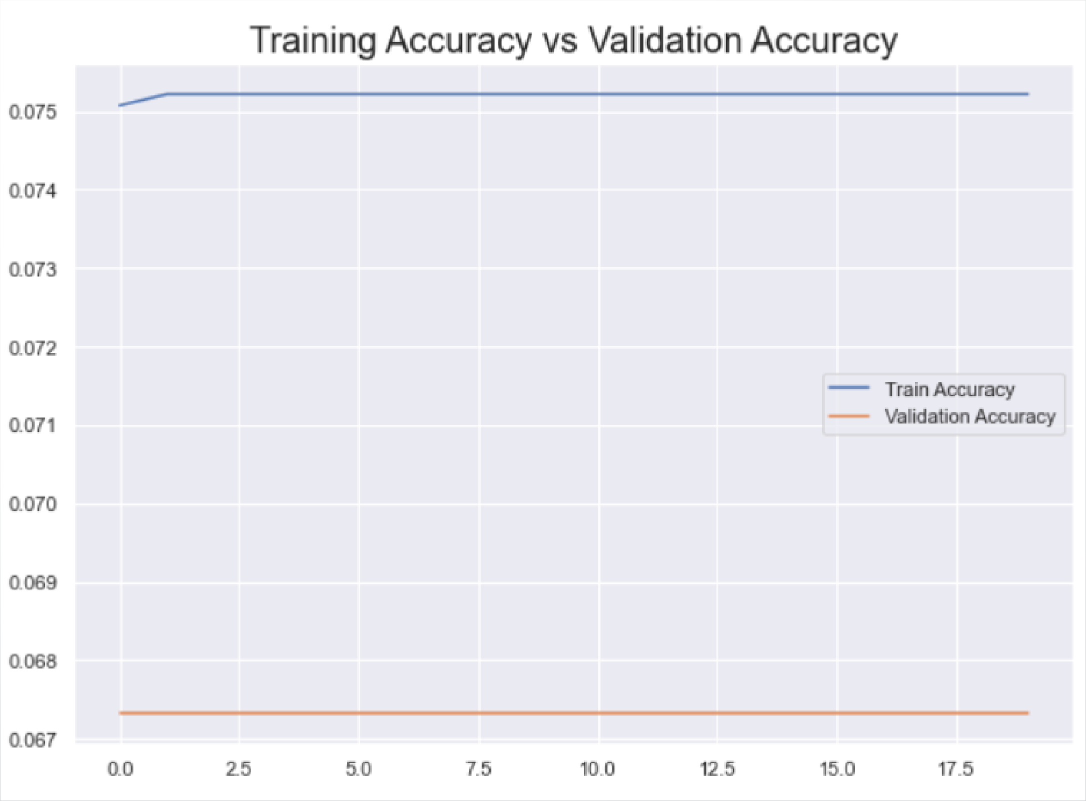
LSTM With Dropout



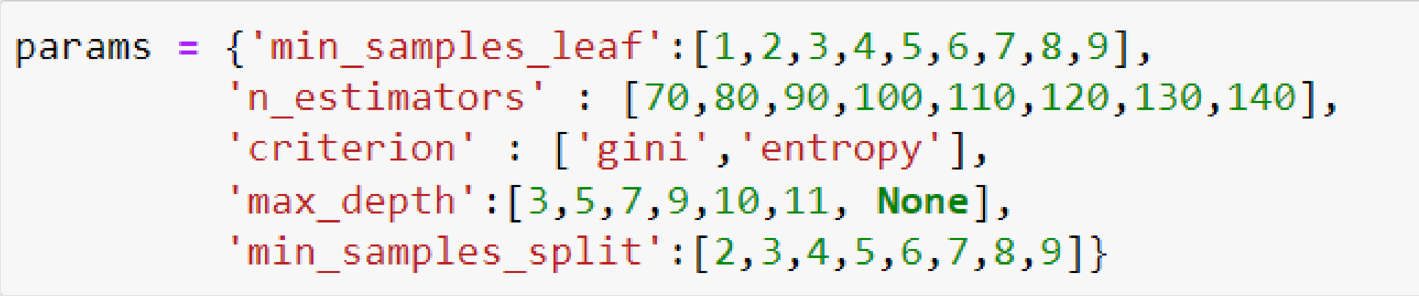
```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_2 (Embedding)      (None, 100, 40)           200000    
_________________________________________________________________
dropout (Dropout)            (None, 100, 40)           0         
_________________________________________________________________
lstm_2 (LSTM)                (None, 100)               56400     
_________________________________________________________________
dropout_1 (Dropout)          (None, 100)               0         
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 101       
=================================================================
Total params: 256,501
Trainable params: 256,501
Non-trainable params: 0
_________________________________________________________________
None
```

LSTM With Dropout

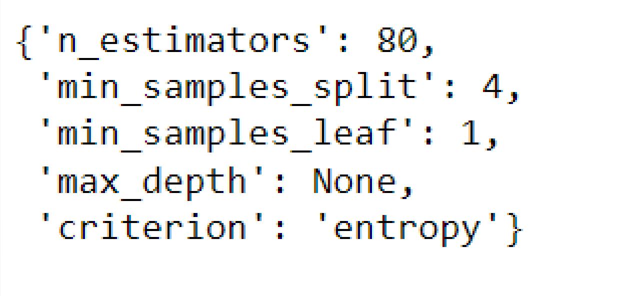




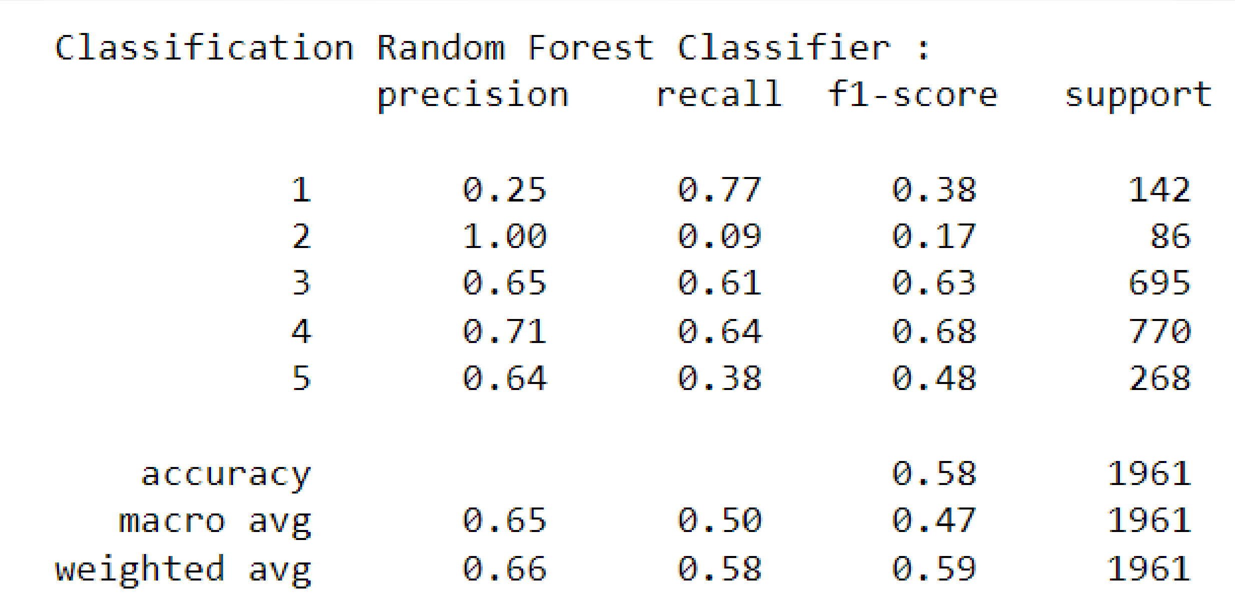
Hyper Parameter Tuning of Random Forest Classifier



```
params = {'min_samples_leaf':[1,2,3,4,5,6,7,8,9],
          'n_estimators' : [70,80,90,100,110,120,130,140],
          'criterion' : ['gini','entropy'],
          'max_depth':[3,5,7,9,10,11, None],
          'min_samples_split':[2,3,4,5,6,7,8,9]}
```



```
{'n_estimators': 80,
 'min_samples_split': 4,
 'min_samples_leaf': 1,
 'max_depth': None,
 'criterion': 'entropy'}
```

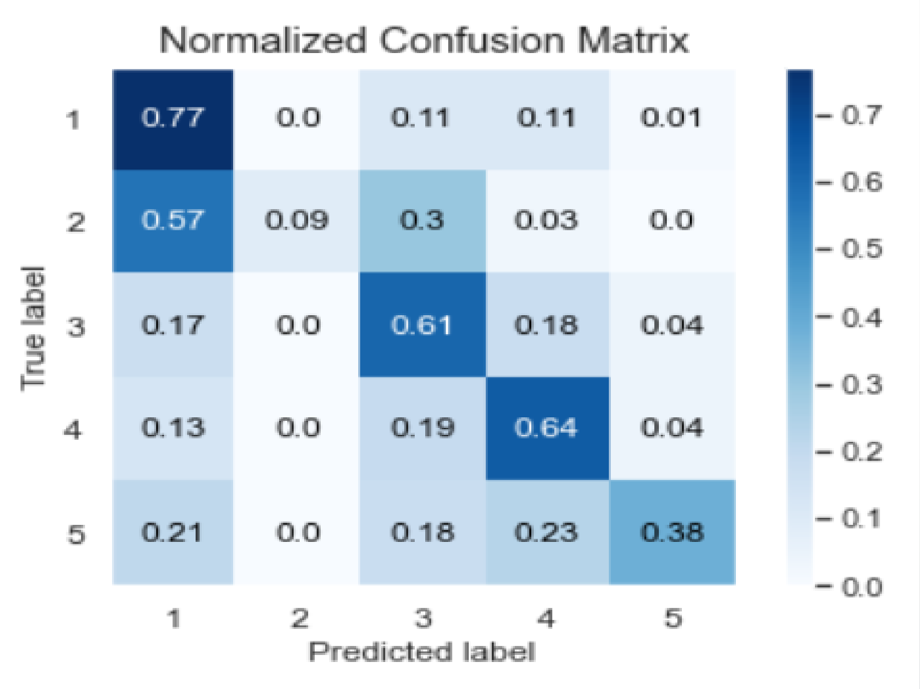


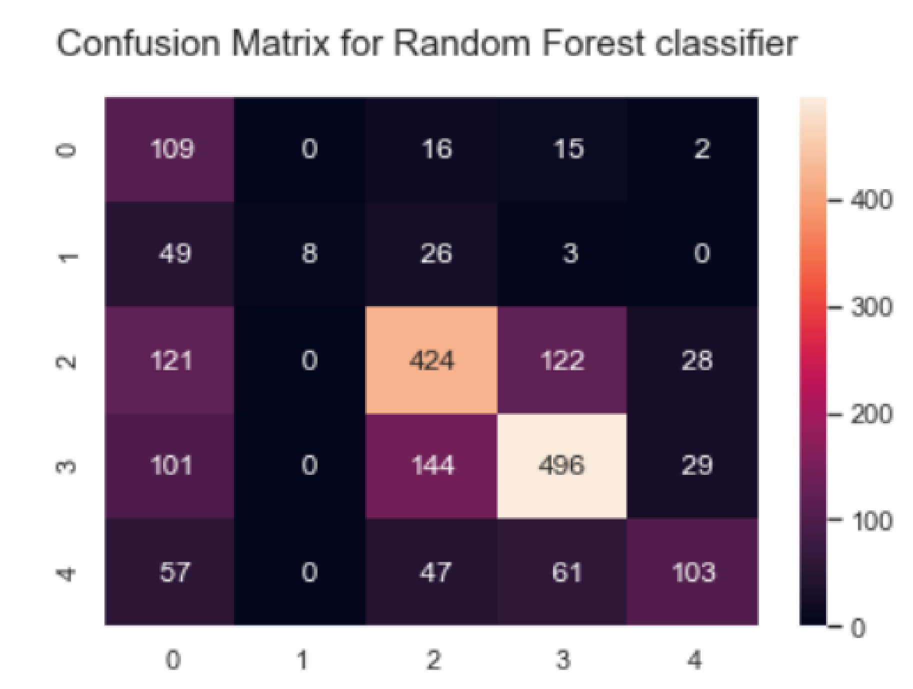
```
Classification Random Forest Classifier :
              precision    recall  f1-score   support

           1       0.25      0.77      0.38       142
           2       1.00      0.09      0.17        86
           3       0.65      0.61      0.63       695
           4       0.71      0.64      0.68       770
           5       0.64      0.38      0.48       268

    accuracy                           0.58      1961
   macro avg       0.65      0.50      0.47      1961
weighted avg       0.66      0.58      0.59      1961
```

Hyper Parameter Tuning of Random Forest Classifier





Accuracy Score : 58.13%

Brand
Reputation
CRM
Quality
NLP Natural Language Processing
Ai

THANK YOU!