Hate Speech Analysis on Roman Urdu Dataset

### Dyass Khalid, Syed Muhammad Arham Noman [20100004@lums.edu.pk](mailto:20100004@lums.edu.pk) , 20100059@lums.edu.pk

May 20, 2020

**ABSTRACT**

**Hate speech detection has been a growing area of interest as the user base of social media websites has grown. Human content moderators can only do so much, hence automatic detection has become crucial. This paper explores the topic with a focus on Roman Urdu texts as these have not been explored.**

# Introduction:

Due to the recent advancement in the field of ML particularly due to advent of deep learning networks, Natural Language Processing (NLP) has seen an increase in research papers. These advances have led to the adoption of NLP in automated speech recognition, question-answering systems, and more sophisticated NLP systems such as Siri and Alexa. However, recent applications such as exact voice copying with the voice generating hate-speech without even the people whose voice is being copied knowing is a big concern. To alleviate such accidents, there is a growing interest in hate-speech classification.

Classifying hate speech accurately poses new challenges for the NLP community. First, due to multi-lingual nature of social media platforms, it has become difficult for the moderators to identify hate-speech. Furthermore, hate-speech for one community may not be hate-speech for another. Therefore, it is important to design hate speech recognition systems that are specifically trained to detect hate speech in a language.

# Motivation:

Classifying hate speech in a multi-lingual society where 57% of the population can converse in English is a challenge as the rest of population speaks Urdu, the National language of a country of 212.2 million people. Hate speech classification can be a challenge in this scenario manually as the moderator might not know both languages. Furthermore, the system is significantly flawed, firstly human moderators are not consistent in marking content as hateful or inappropriate, human bias can creep into moderation and result in racism or xenophobia deciding what content stays online. Secondly, human moderators cannot work on this job for long because of the negative effects on their mental health due to exposure to graphic content. Therefore, it is in the interest of online platforms to make the transition towards automatic detection of hateful and inappropriate content.

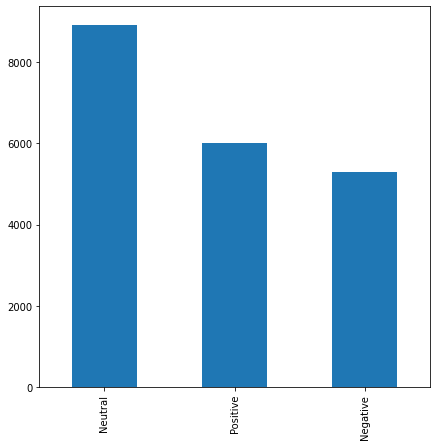
# Related Work:

[1] suggest that unigram features do not perform well in classification tasks and feature extraction criteria can be made more robust by using feature extraction techniques. [2] suggest that ensemble learning is the way forward in the multi-lingual classification problem whereas [3] improves the existing models using the pre-trained embeddings which are robust vector representations.[3] also suggests to use complex features. [4] [5] also suggest to use deep learning networks to these tasks. We aim not only apply these techniques to Roman Urdu dataset for sentiment classification task but also try hybrid model approaches.

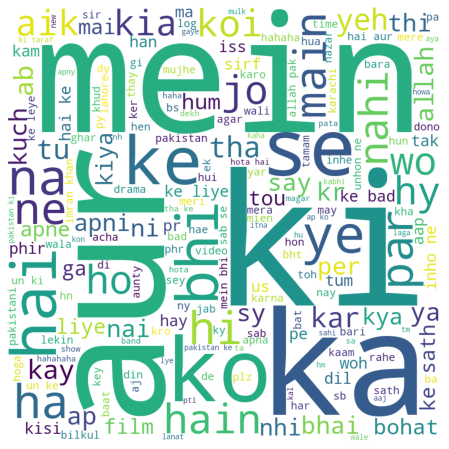
# Methodology:

# Dataset:

The dataset is taken from Kaggle and consist of 20,228 tweets with its sentiment and contain the following count of Positive, Negative and Neutral tweets:



The word cloud of our dataset after removal of stop-words is the following:

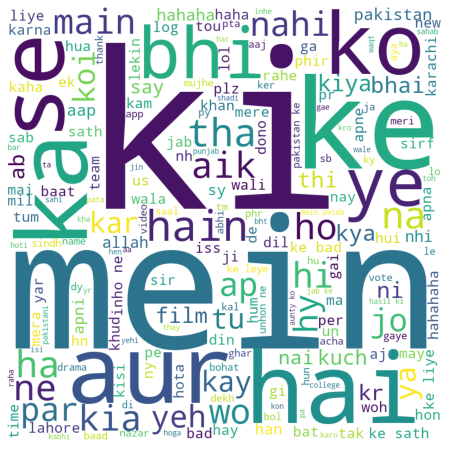


The word-cloud for positive tweets is as follows:



As you can see words like ‘Allah’, ’Imran Khan’ are also present.

The word cloud for Neutral Tweets is as follows:



As you can see that words like ‘Team’ and ‘Pakistan’ are in the word cloud prompting to think that Pakistan Cricket team might have lost the match but the overall sentiment about

it is neutral.

The word cloud for the negative tweets is as follows:



As you can see that words like ‘Lanat’ and ‘Nawaz Shareef’ are in negative word cloud as people on twitter tend to go against them regularly.

# Pre-Processing Method:

Standard Approach: The most common pre-processing strategy used in Roman Urdu is lowercasing.

Our Approach: We decided not to do only lowercasing but using an initial stop-word list removed words with edit distance of 1 from the corpus which gave accuracy less than the baseline model, so we decided to follow the standard approach.

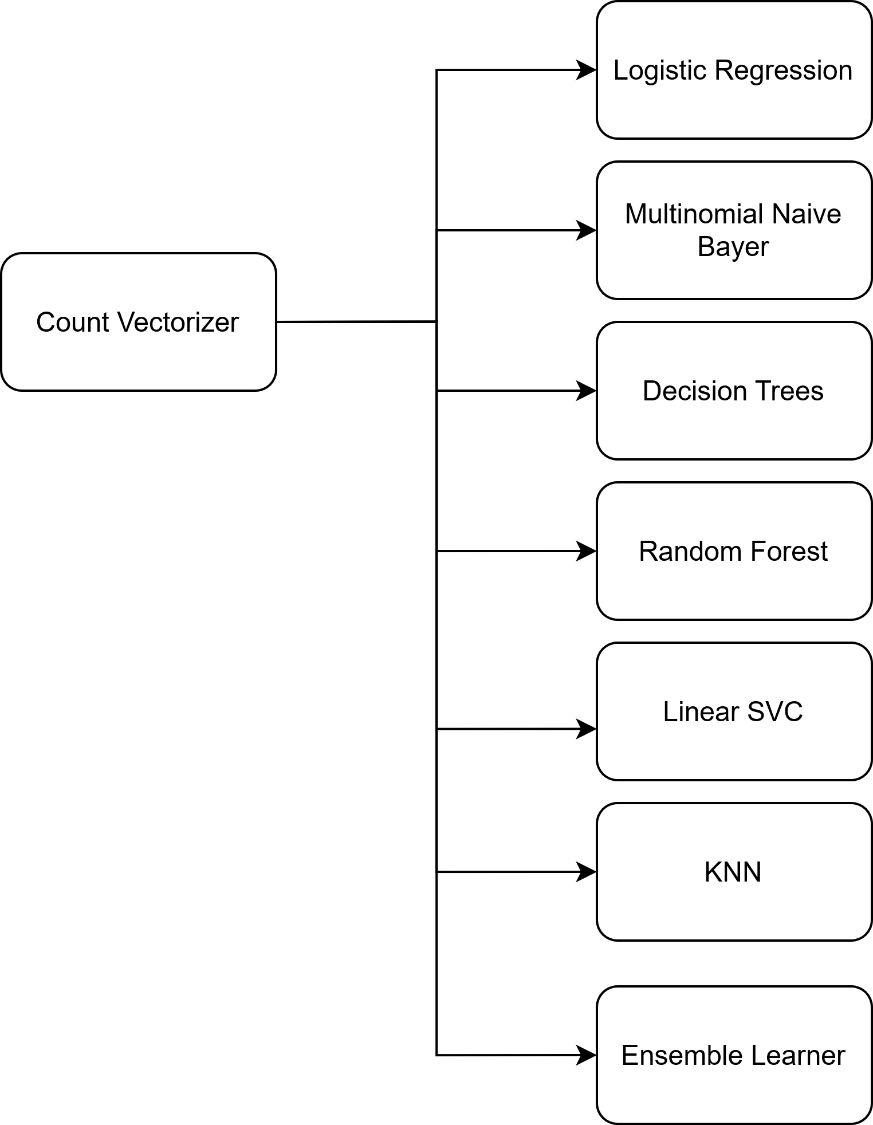
# 

# Experiments:

We experimented the following machine learning classifier using Count Vectorizer:

1. Logistic Regression
2. Multinomial Naïve Bayes
3. Linear SVC
4. Decision Trees
5. Random Forest
6. KNN
7. Voting Ensemble classifier with classifiers mentioned above.

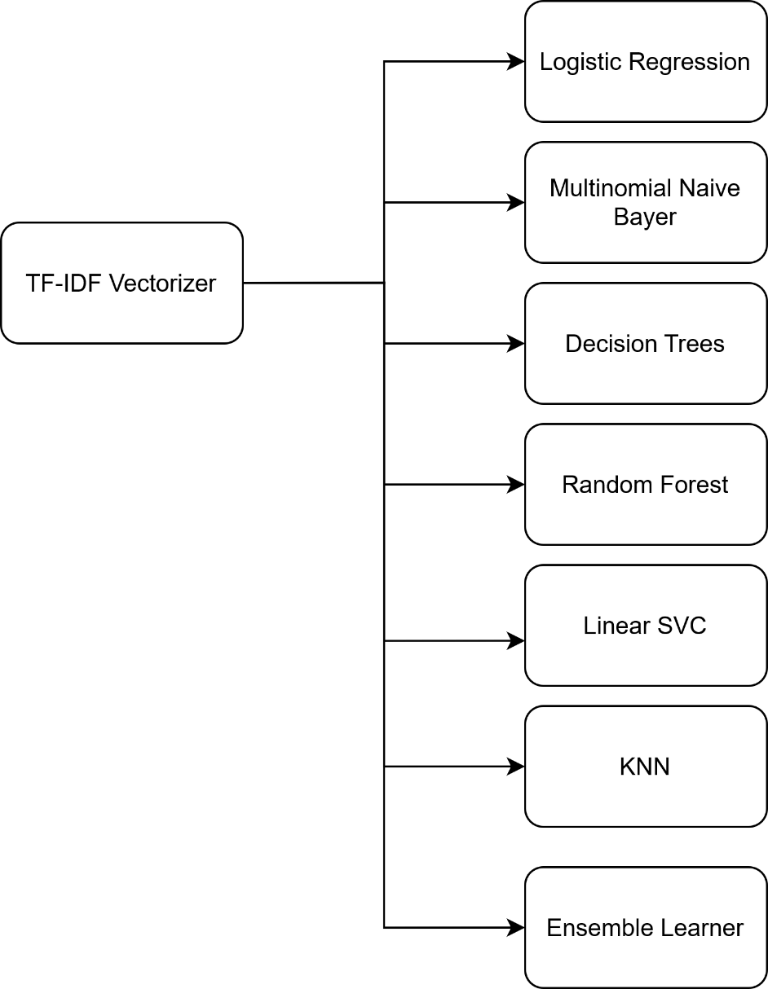
This is shown in the following figure:



We trained the following machine learning classifiers using TF-IDF Vectorizer:

1. Logistic Regression
2. Multinomial Naïve Bayes
3. Linear SVC
4. Decision Trees
5. Random Forest
6. KNN

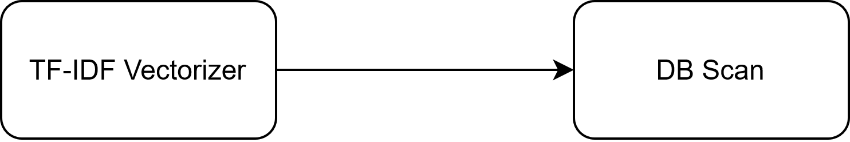
This is shown in the following figure:



For Un-supervised learning we tried the following approach:

1. DB Scan

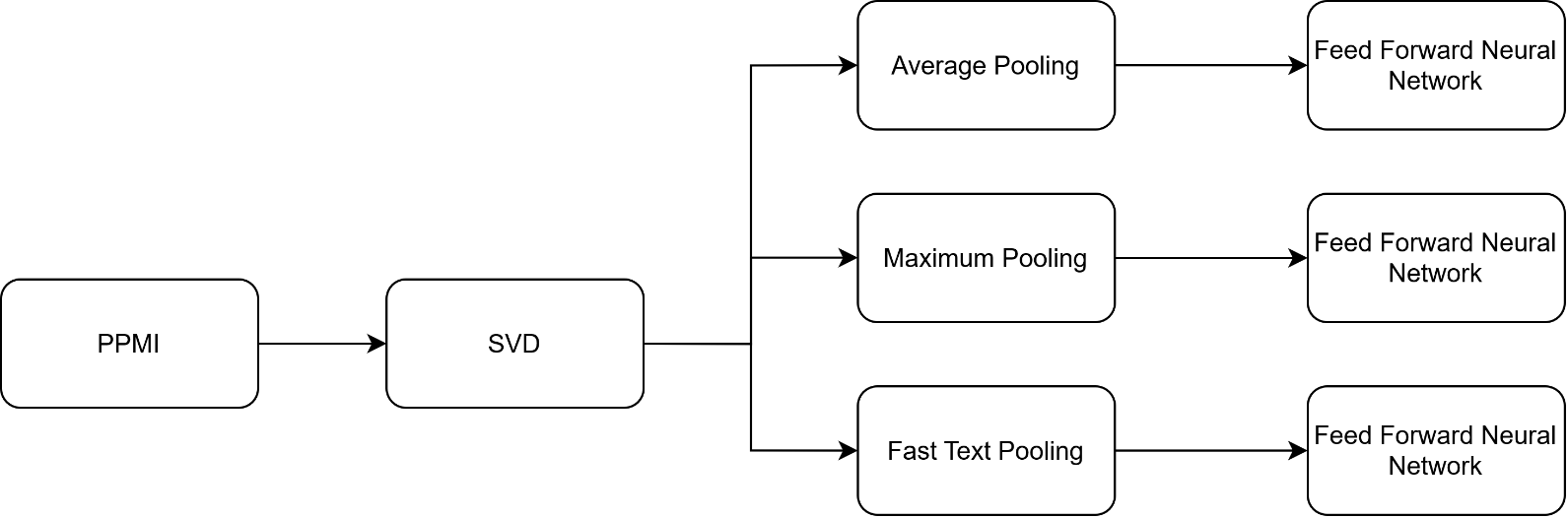
This is shown in the following:



We tried the following three feed-forward neural network approaches with the following pooling techniques using word 2 vector models:

1. Average Pooling
2. Max Pooling
3. Fast text normalization pooling

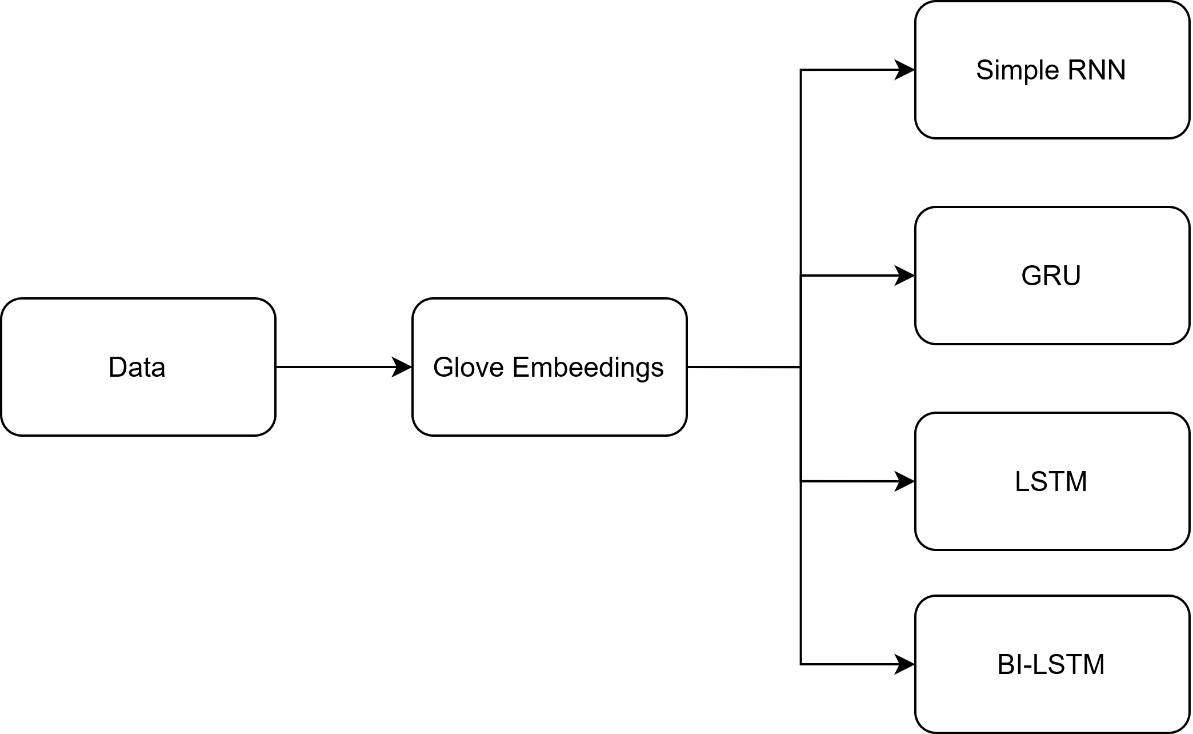
This is shown in the following figure:



We trained the recurrent neural network approaches using glove embeddings:

1. Simple RNN
2. LSTM
3. GRU
4. Bi-LSTM

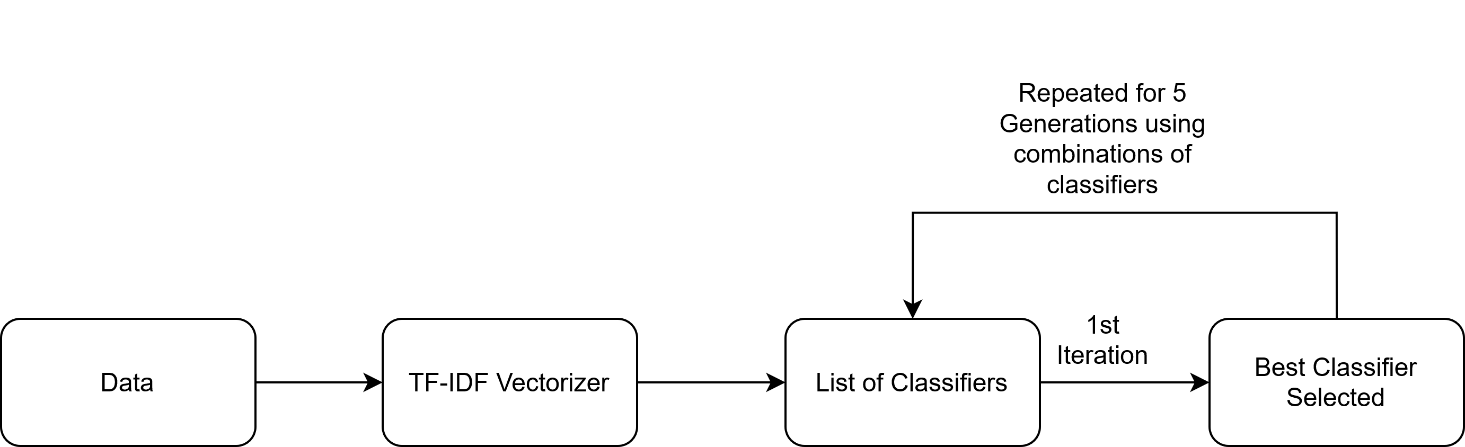
This is shown in the following figure:



We also tested the results in RNN with different loss functions which are discussed.

We also employed techniques from the domain of evolutionary computing by testing our machine learning classifiers for 5 generations which results in **120** models being tested and evaluated in the pipeline. The results are in the submitted folders with best pipelines saved separately.

This is shown in the following figure:



1. **Results:**

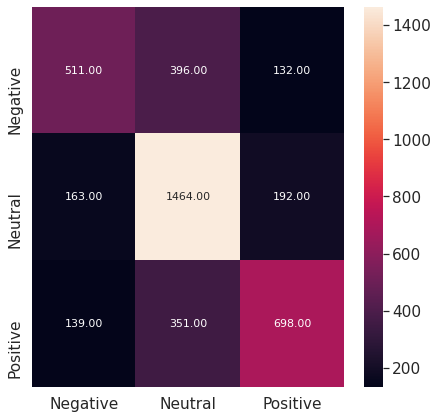
The results are as follows:

## Logistic Regression:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 95.24 |
| Testing Data | 66.06 |

The following confusion matrix is generated:



The following classification reports is obtained:

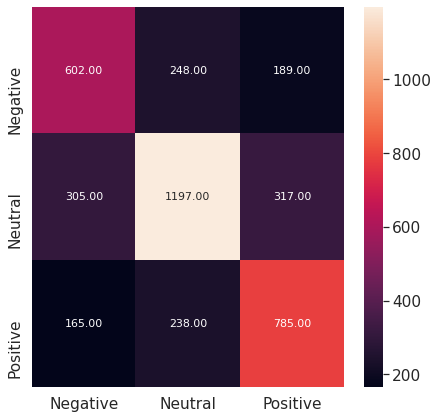
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.63 | 0.49 | 0.55 | 1039 |
| Neutral | 0.66 | 0.80 | 0.73 | 1819 |
| Positive | 0.68 | 0.59 | 0.63 | 1188 |
| Accuracy |  |  | 0.66 | 4046 |
| Macro Accuracy | 0.66 | 0.63 | 0.64 | 4046 |
| Weighted Accuracy | 0.66 | 0.66 | 0.65 | 4046 |

## Multinomial Naïve Bayes:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 83.85 |
| Testing Data | 63.86 |

The following confusion matrix is generated:



The following classification reports is obtained:

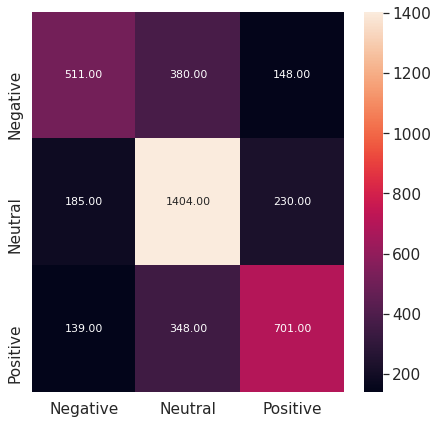
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.56 | 0.58 | 0.57 | 1039 |
| Neutral | 0.71 | 0.66 | 0.68 | 1819 |
| Positive | 0.61 | 0.66 | 0.63 | 1188 |
| Accuracy |  |  | 0.64 | 4046 |
| Macro Accuracy | 0.63 | 0.63 | 0.63 | 4046 |
| Weighted Accuracy | 0.64 | 0.64 | 0.64 | 4046 |

## Linear SVC:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 98.82 |
| Testing Data | 64.65 |

The following confusion matrix is generated:



The following classification reports is obtained:

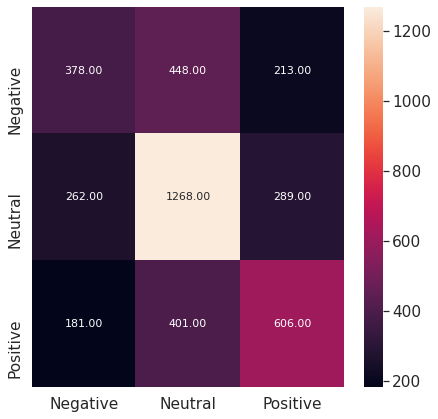
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.61 | 0.49 | 0.55 | 1039 |
| Neutral | 0.65 | 0.77 | 0.71 | 1819 |
| Positive | 0.65 | 0.59 | 0.63 | 1188 |
| Accuracy |  |  | 0.65 | 4046 |
| Macro Accuracy | 0.64 | 0.62 | 0.62 | 4046 |
| Weighted Accuracy | 0.64 | 0.65 | 0.64 | 4046 |

## Decision Trees:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 99.80 |
| Testing Data | 55.65 |

The following confusion matrix is generated:



The following classification reports is obtained:

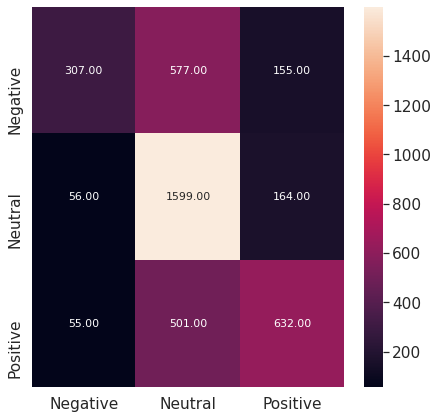
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.46 | 0.36 | 0.41 | 1039 |
| Neutral | 0.60 | 0.70 | 0.64 | 1819 |
| Positive | 0.55 | 0.51 | 0.53 | 1188 |
| Accuracy |  |  | 0.56 | 4046 |
| Macro Accuracy | 0.54 | 0.52 | 0.53 | 4046 |
| Weighted Accuracy | 0.55 | 0.56 | 0.55 | 4046 |

## Random Forest:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 99.80 |
| Testing Data | 62.72 |

The following confusion matrix is generated:



The following classification reports is obtained:

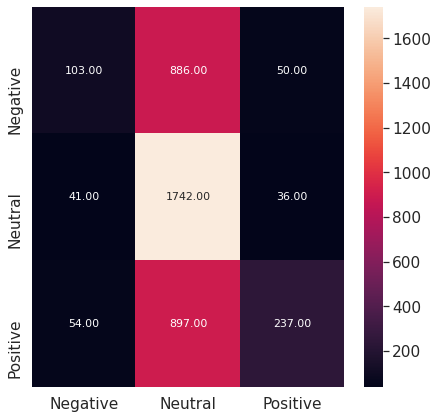
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.73 | 0.30 | 0.42 | 1039 |
| Neutral | 0.60 | 0.88 | 0.71 | 1819 |
| Positive | 0.66 | 0.53 | 0.59 | 1188 |
| Accuracy |  |  | 0.66 | 4046 |
| Macro Accuracy | 0.67 | 0.57 | 0.57 | 4046 |
| Weighted Accuracy | 0.65 | 0.63 | 0.60 | 4046 |

## KNN:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 56.38 |
| Testing Data | 51.45 |

The following confusion matrix is generated:



The following classification reports is obtained:

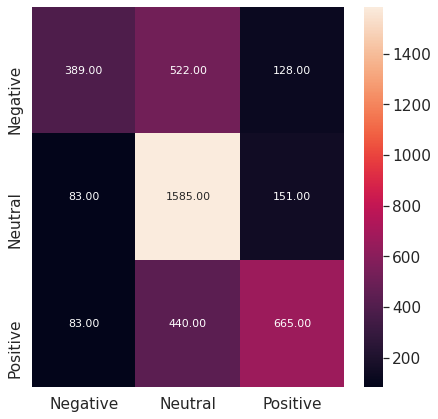
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.52 | 0.10 | 0.17 | 1039 |
| Neutral | 0.49 | 0.96 | 0.65 | 1819 |
| Positive | 0.73 | 0.20 | 0.31 | 1188 |
| Accuracy |  |  | 0.51 | 4046 |
| Macro Accuracy | 0.58 | 0.42 | 0.38 | 4046 |
| Weighted Accuracy | 0.57 | 0.51 | 0.43 | 4046 |

## 6.7 Ensemble Classifier:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 98.96 |
| Testing Data | 65.22 |

The following confusion matrix is generated:



The following classification reports is obtained:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.70 | 0.37 | 0.49 | 1039 |
| Neutral | 0.62 | 0.87 | 0.73 | 1819 |
| Positive | 0.70 | 0.56 | 0.62 | 1188 |
| Accuracy |  |  | 0.65 | 4046 |
| Macro Accuracy | 0.68 | 0.60 | 0.61 | 4046 |
| Weighted Accuracy | 0.67 | 0.65 | 0.63 | 4046 |

## 7 Discussion

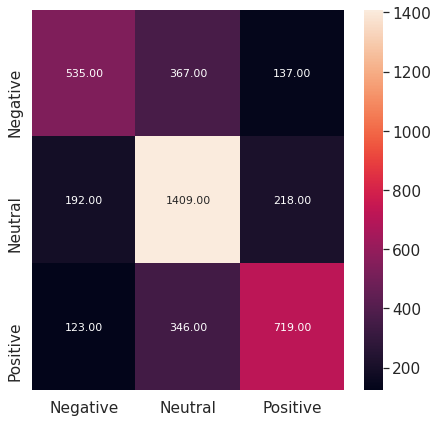
We also performed experiments using unigram, bigram, and unigram plus bigram. The results of unigram language modeling out. The results of only the unigram modelling using count vectorizer are discussed above. Clearly logistic regression has out-performed other classifiers as it tries to learn a non-linear decision boundary from the start rather than estimating linear boundaries to learn a non-linear boundary.

## Logistic Regression Using TF-IDF Vectorizer:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 83.50 |
| Testing Data | 65.81 |

The following confusion matrix is generated:



The following classification reports is obtained:

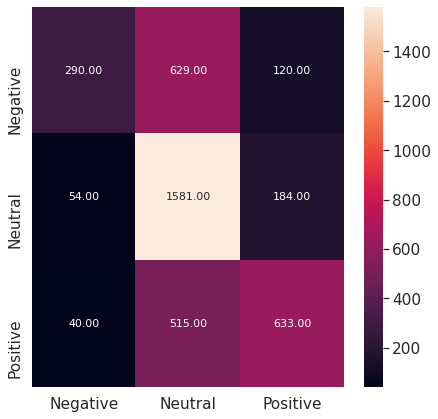
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.63 | 0.51 | 0.57 | 1039 |
| Neutral | 0.66 | 0.77 | 0.72 | 1819 |
| Positive | 0.6 | 0.61 | 0.64 | 1188 |
| Accuracy |  |  | 0.66 | 4046 |
| Macro Accuracy | 0.65 | 0.63 | 0.64 | 4046 |
| Weighted Accuracy | 0.66 | 0.66 | 0.65 | 4046 |

## Multi-Nomial Naïve Bayes Using TF-IDF Vectorizer:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 79.66 |
| Testing Data | 61.88 |

The following confusion matrix is generated



The following classification reports is obtained:

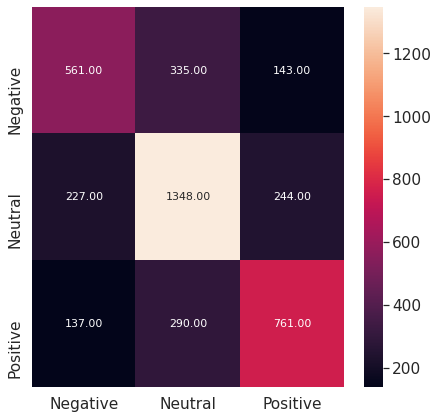
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.76 | 0.28 | 0.41 | 1039 |
| Neutral | 0.58 | 0.87 | 0.70 | 1819 |
| Positive | 0.68 | 0.53 | 0.60 | 1188 |
| Accuracy |  |  | 0.62 | 4046 |
| Macro Accuracy | 0.67 | 0.56 | 0.57 | 4046 |
| Weighted Accuracy | 0.65 | 0.62 | 0.59 | 4046 |

## Linear SVC Using TF-IDF Vectorizer:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 96.79 |
| Testing Data | 65.99 |

The following confusion matrix is generated:



The following classification reports is obtained:

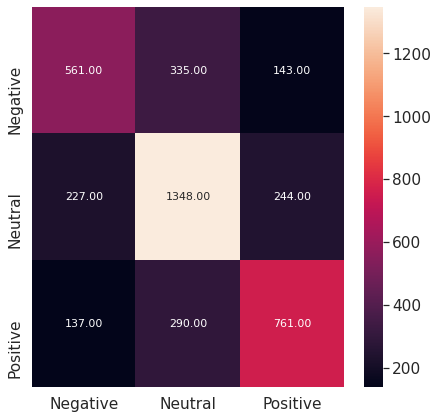
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.61 | 0.54 | 0.57 | 1039 |
| Neutral | 0.68 | 0.74 | 0.71 | 1819 |
| Positive | 0.66 | 0.64 | 0.65 | 1188 |
| Accuracy |  |  | 0.66 | 4046 |
| Macro Accuracy | 0.65 | 0.64 | 0.64 | 4046 |
| Weighted Accuracy | 0.66 | 0.66 | 0.66 | 4046 |

## Linear SVC Using TF-IDF Vectorizer:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 96.79 |
| Testing Data | 65.99 |

The following confusion matrix is generated:



The following classification reports is obtained:

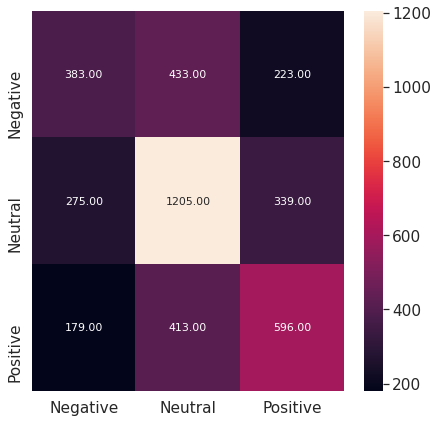
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.61 | 0.54 | 0.57 | 1039 |
| Neutral | 0.68 | 0.74 | 0.71 | 1819 |
| Positive | 0.66 | 0.64 | 0.65 | 1188 |
| Accuracy |  |  | 0.66 | 4046 |
| Macro Accuracy | 0.65 | 0.64 | 0.64 | 4046 |
| Weighted Accuracy | 0.66 | 0.66 | 0.66 | 4046 |

## Decision Trees Using TF-IDF Vectorizer:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 99.80 |
| Testing Data | 53.97 |

The following confusion matrix is generated:



The following classification reports is obtained:

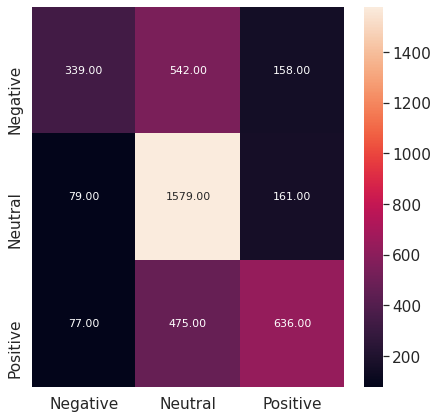
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.46 | 0.37 | 0.41 | 1039 |
| Neutral | 0.59 | 0.66 | 0.62 | 1819 |
| Positive | 0.51 | 0.50 | 0.51 | 1188 |
| Accuracy |  |  | 0.54 | 4046 |
| Macro Accuracy | 0.52 | 0.51 | 0.51 | 4046 |
| Weighted Accuracy | 0.53 | 0.54 | 0.53 | 4046 |

## Random Forest Using TF-IDF Vectorizer:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 99.80 |
| Testing Data | 63.12 |

The following confusion matrix is generated:



The following classification reports is obtained:

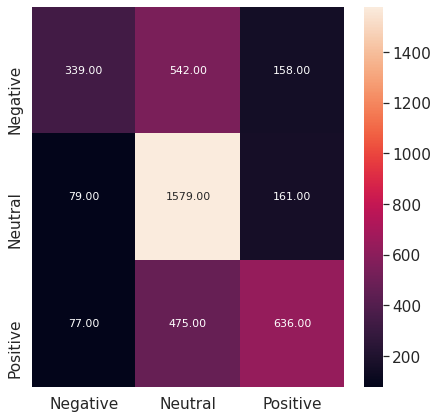
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.68 | 0.33 | 0.44 | 1039 |
| Neutral | 0.61 | 0.87 | 0.72 | 1819 |
| Positive | 0.65 | 0.58 | 0.58 | 1188 |
| Accuracy |  |  | 0.63 | 4046 |
| Macro Accuracy | 0.65 | 0.58 | 0.58 | 4046 |
| Weighted Accuracy | 0.64 | 0.63 | 0.61 | 4046 |

## KNN Using TF-IDF Vectorizer:

We obtained the following accuracy on training and testing data as shown in the table below:

|  |  |
| --- | --- |
| Data | Mean Accuracy |
| Training Data | 47.23 |
| Testing Data | 48.02 |

The following confusion matrix is generated:



The following classification reports is obtained:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Precision | Recall | F1-Score | Support |
| Negative | 0.68 | 0.33 | 0.44 | 1039 |
| Neutral | 0.61 | 0.87 | 0.72 | 1819 |
| Positive | 0.65 | 0.58 | 0.58 | 1188 |
| Accuracy |  |  | 0.63 | 4046 |
| Macro Accuracy | 0.65 | 0.58 | 0.58 | 4046 |
| Weighted Accuracy | 0.64 | 0.63 | 0.61 | 4046 |

We also performed experiments using unigram, bigram and unigram plus bigram. The results of unigram language modeling out. The results of only the unigram modelling using TF-IDF vectorizer are discussed above. Clearly linear SVC has outperformed other classifiers as it tries to learn a linear decision boundary in the n-dimensional plane.

## Genetic Algorithm:

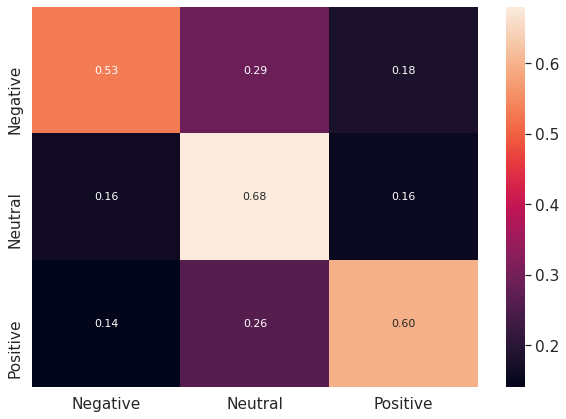
The detailed results of this algorithm can be found in the csv file that accompanies this paper, in the interest of saving space only key results are written here in the following table.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

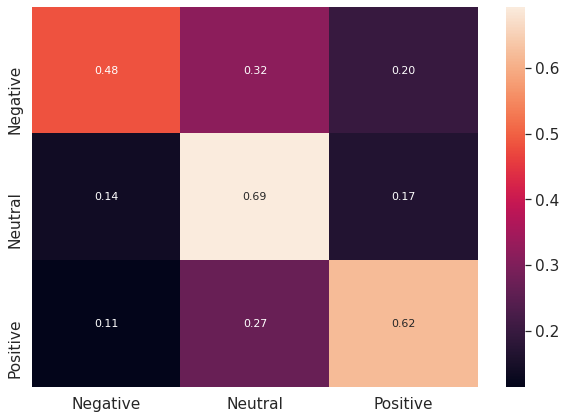
## :

* 1. **RNN:**

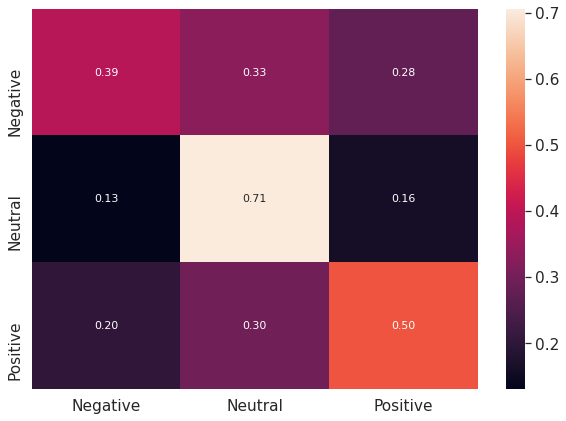
We trained four recurrent neural networks using LSTM, GRU, Simple RNN and Bi LSTM. The GRU model outperforms the rest of the models with 66.67% accuracy. The most possible explanations for this are that GRU is better at capturing dependencies but is simpler than LSTM and BI-LSTM. The confusion matrices of GRU is as follows:



The confusion matrix for LSTM is as follows:

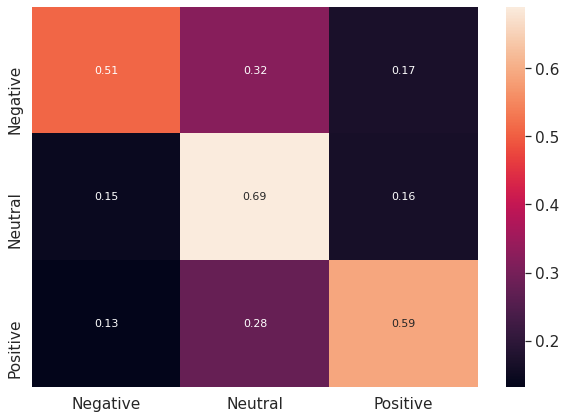


The confusion matrix for simple RNN is as follows:



Clearly it performed worse as it is not better at capturing long-term dependencies and suffers from the exploding gradient problem.

The confusion matrix for Bi-LSTM is as follows:



## 8 Future Work:

The learning ability of machine learning models depends on the quality of data available mainly. In future, we want to extract our own hate-speech data set using the twitter API. Moreover, we want to try more ensemble learning techniques such as AdaMax and AdaBoost. Furthermore, we want to train multiple inputs and single output neural network for classification.

# Bibliography

|  |  |
| --- | --- |
| [1] | William Warner and Julia Hirschberg. “Detecting Hate Speech on the World Wide Web”. In: Proceedings of the Second Workshop on Language in Social Media. LSM ’12. Montreal, Canada: Association for Computational Linguistics, 2012, pp. 19–26. |
| [2] | Michele Corazza et al. “A Multilingual Evaluation for Online Hate Speech Detection”. In: ACM Trans. Internet Technol. 20.2 (Mar. 2020). |
| [3] | Wafa Alorainy et al. “ “The Enemy Among Us”: Detecting Cyber Hate Speech with Threats-Based Othering Language Embeddings”. In: ACM Trans. Web 13.3 (July 2019). |
| [4] | Mainack Mondal, Leandro Araújo Silva, and Fabrıcio Benevenuto. “A Measurement Study of Hate Speech in Social Media”. In: Proceedings of the 28th ACM Conference on Hypertext and Social Media. HT ’17. Prague, Czech Republic: Association for Computing Machinery, 2017, pp. 85–94. |
| [5] | Alexandra Schofield and Thomas Davidson. “Identifying Hate Speech in Social Media”. In: XRDS 24.2 (Dec. 2017), pp. 56–59 |