**General Overview:**

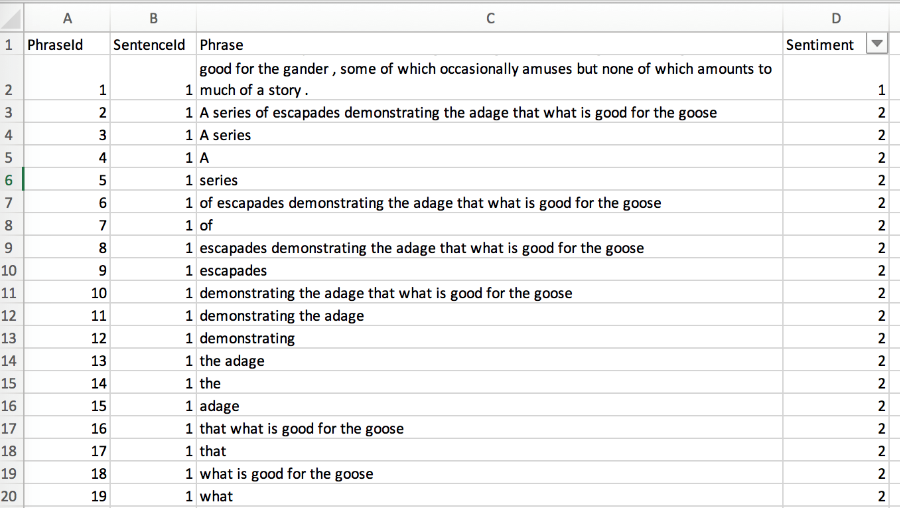
Quinn E Knudsen & Sathish Rajendiran

NLP Final Project NLP

For this final project, we chose to analyze the Kaggle Movie Review Dataset. Our objective was to maximize the predictive power of a model aimed at classifying a review as either positive or negative. A variety of models and preprocessing techniques were experimented with, however, the best combination as measured by accuracy was a tensor flow deep learning model. Accuracy scores ranged from approximately 60-80%.

We used 5- and 10-fold cross validation to minimize the risk of variability in our model results and to strengthen the validity of our findings.  For some models, however, the sheer runtime prohibited proper cross validation.

**Dataset:**



**Data types:**

PhraseId int64

SentenceId int64

Phrase object

Sentiment int64

**Null Values:**

PhraseId 0

SentenceId 0

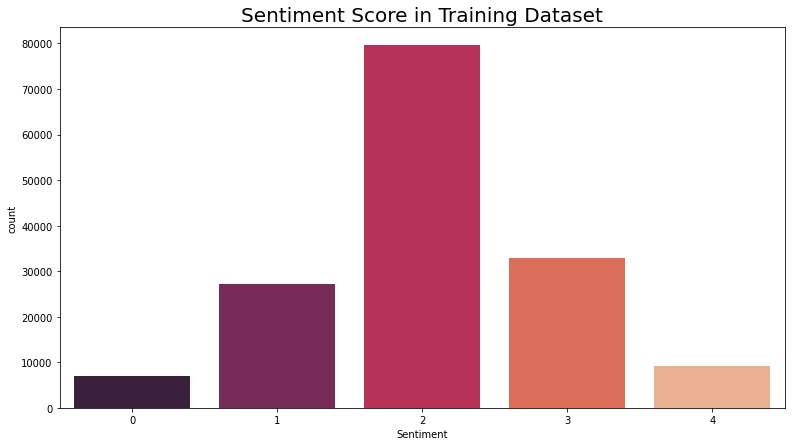
Phrase 0

Sentiment 0

**Step 1: Text Processing**

1. **Exploratory Data Analysis**

The first stage of this analysis included exploring the dataset to get a feel for the data we would be working with. This included summarizing the variables in the dataset along with the variable types.  As the dependent variable in this analysis is “Sentiment”, additional focus was paid here by visualizing the count of sentiment scores which revealed a normal distribution with the majority of the reviews receiving a ‘2’ or neutral rating (figure 1 below).



Number of Postive Sentiment: 42133

Number of Neutral Sentiment: 79582

Number of Negative Sentiment: 34345

Percentage of positive Sentiment 27.0%

Percentage of neutral Sentiment 50.99%

Percentage of negative Sentiment 22.01%

1. **Tokenize the dataset**

The dataset was tokenized, separating the written text from the numerical variables to data cleaning.

1. **Stop words were removed**

The NLTK stop words were removed to reduce some of the noise that might be found within the dataset.

1. **Custom stop words**

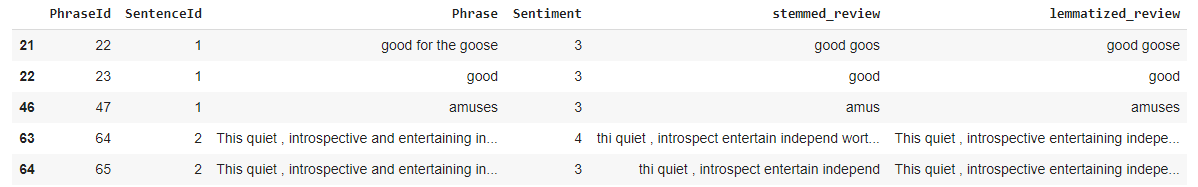
Custom stop words were added to this specific dataset and included: ['could','would','might','must','need','sha','wo','y',"'s","'d","'ll","'t","'m","'re","'ve", "n't",'wa','gon','na']

1. **Stemming**

A stemmed variable was created by stemming the original review and adding the stemmed data to the original dataset to experiment with later.

1. **Lemmatization**

Similar to stemming, a lemmatized variable was also added to experiment with during classification.

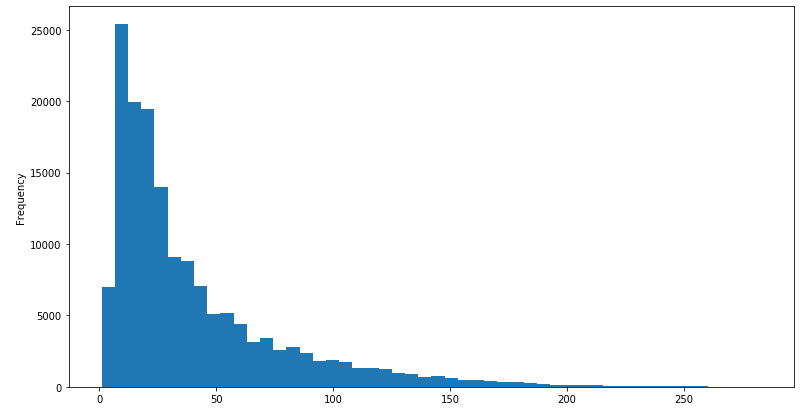


1. **Remove alphanumeric**

All alphanumeric noise was removed from the dataset.

**8. Count the number of characters per review**

For each review, a total count of characters was added as a feature. For example, the sentence “A series of escapades demonstrating the adage that what is good for the goose” contains 77 characters and was scored as 77.



The average review had 40 characters.

count 156060.000000

mean 40.217224

std 38.154130

min 1.000000

25% 14.000000

50% 26.000000

75% 53.000000

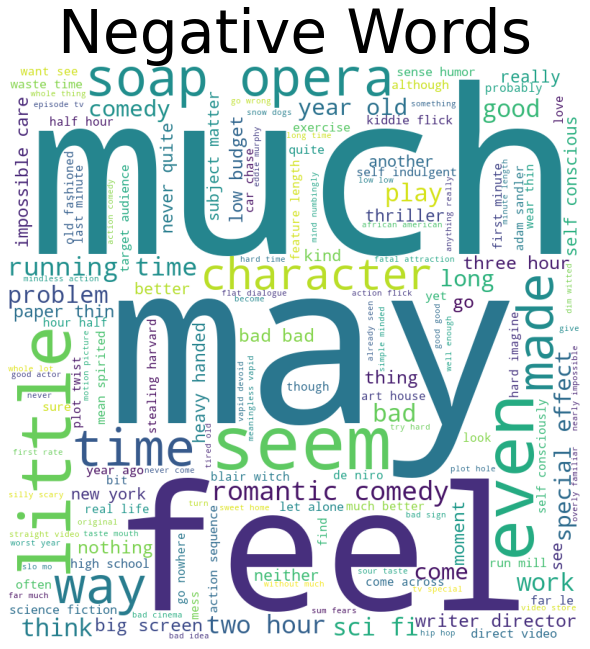
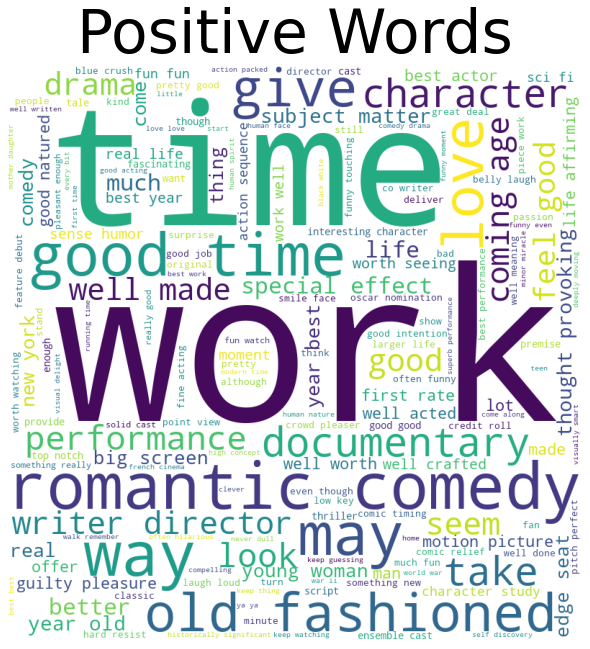
max 283.000000

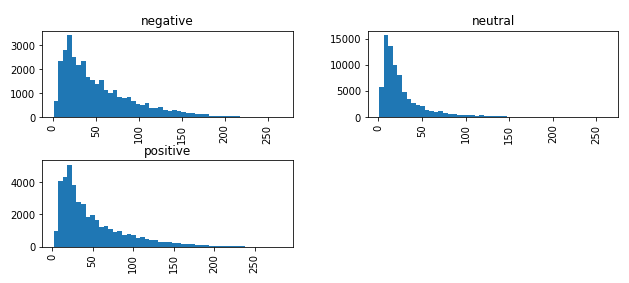
**9. Separate into positive and negative and review word cloud**

The dataset was further explored by segmenting the words into positive and negative and comparing the word clouds of each. Additional stop words were added here:

["movie","film","one","ha","story","make","doe","lrb","rrb"]

Romantic comedy, old fashioned, documentary, drama, comedy, good time, love, ensemble cast, well crafted, special effect, solid cast, oscar nomination, first rate, well mad were some standouts in the positive words.  On the negative words, soap opera, self-conscious, low budget, sci fi, wate time, special effect, paper thin, heavy handed, write director, thriller, romantic comedy, never quite were used. Although some clear language differences emerge, it is evident that some words appear in both such as romantic comedy, so additional context around the sentence is needed outside of the bigram. It does appear, however, that romantic comedy is much more central to the positive word cloud than the negative.



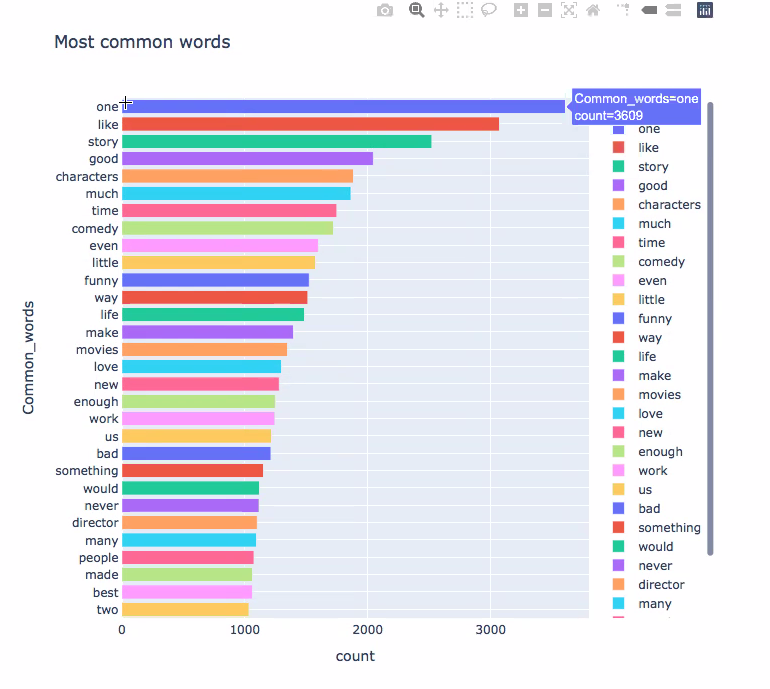


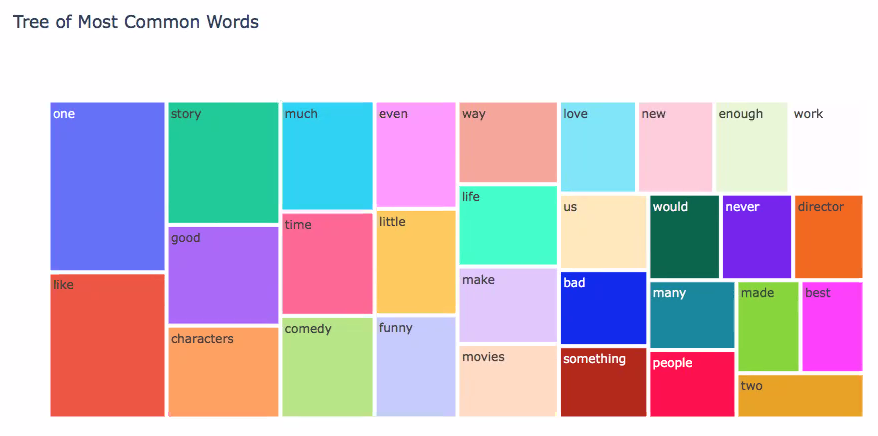
Positive and negative reviews had slightly longer character counts than neutral with skews extending more often on the longer end.

1. **Review top n-grams**

The top 30 words by frequency are displayed here after removing the noise of stopwords and punctuations.







**Step 3: Experiments**

The preprocessed data was then split into train test (80/20) and a variety of models were used to predict the sentiment.

In addition, General findings included

little disparity between stemmed and lemmatized data and the apparent importance of response length (displayed with best cross validated model in the

random forest).

Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Neutral category had the best

Precision on both NLTK Classifier and external Classifier. Mainly because of the more observations (~51%) and is obvious from the Support values.

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations. Recall % is consistent across all categories. Which is good news!

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is

not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works

best if false positives and false negatives have similar cost. In our case, F1 score is 0.76 on the random forest model. F1 Score = 2*(Recall* Precision) /

(Recall + Precision)

Turning the problem into a binary classification significantly improved the accuracy as one might expect. This classification removed neutral responses and

considered just whether or not the note was positive or negative. In a practical application, this could be a useful starting place.

The tensor flow models were highly accurate but were more of an experimental black box approach. The training time limited our ability to cross validate

the results as well. A general finding was that the cross validation curbed our enthusiasm on accuracy.

In fact, the cross validated models showed were more in line with the Kaggle submissions. Our submissions ranged from 55-60% on Kaggle.

**Summary**

Overall, We were able to perform exploratory analysis, preprocessing techniques, tokenization using Wordtokenizer, removing stop words, Stemming and Lemmatization techniques and finally exploring several machine learning classifiers including NLTK's native classifiers and external classifiers to predict the sentiment on the movie reviews dataset.

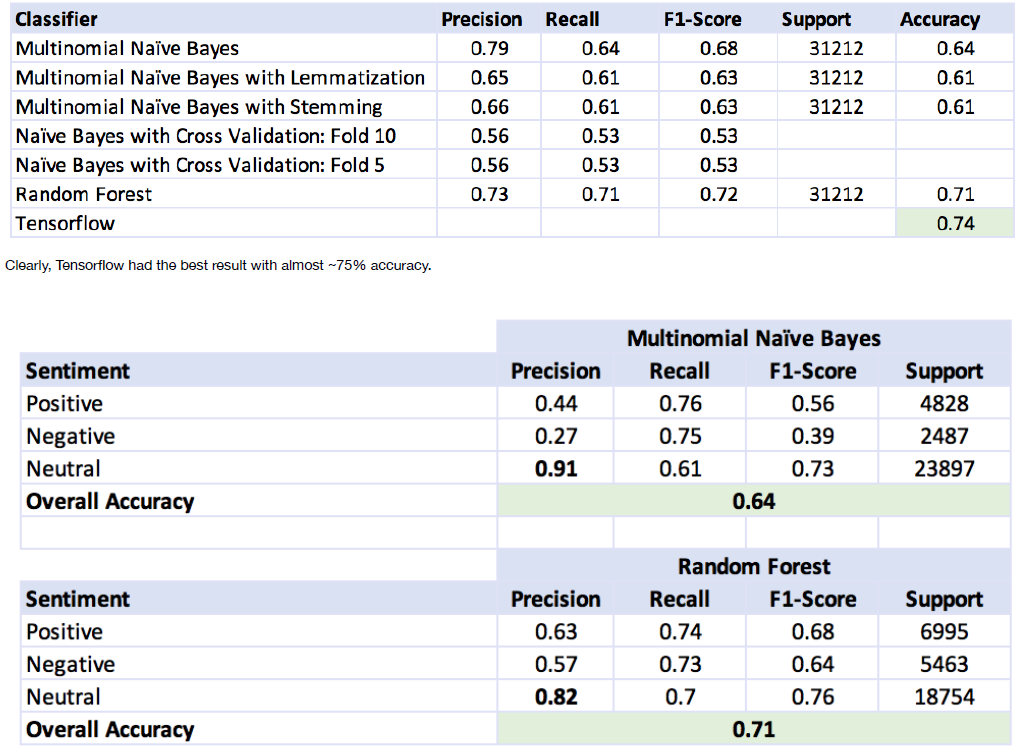
Training set had 156060 observations with 1 dependent and 4 independent variables.We mainly focused on Phrase(actual reviews) and Sentiment (rating from 0 to 4). Sentiment score is further grouped into 3 major categories as below

* Positive (3,4) - consisting of 27% overall reviews
* Negative (0,1) - consisting of 22% overall reviews
* Neutral (2) - consisting of 51% overall reviews Yes, 50% of the data has been tagged as neutral.

There were total ~200 + Stop words including NLTK's and custom words, alphanumeric characters excluded to improve the quality of the data.

Several visualization techniques were used including Barplots, Tree graphs, Wordclouds and other histograms to understand the data spread better.

The preprocessed data was then split into train:test (80/20) and a variety of models were used to predict the sentiment. Please find the overall comparison of the results.



**Conclusion**

Finally, We're able to classify the movie reviews dataset as postive, negative and neutral categories. Overall process involving, data collection, pre-processing,

feature engineering, and experimentation using native NLTK libraries and external classifiers using Python.

In addition, We're able to achive an accuracy of 74% using tensorflow (deep learning), 71% with Random Forest and 64% with Multinomial Naïve Bayes model.

This is in the range of our expectation given the data is more skewed towards neutral category.

Precision and Recall% comparatively better with the dataset given, having highest precesion on neutral category with 91% using Multinomial Naïve Bayes. F1

score is comparatively higher in the case of Random Forest with 76%.

We have also learnt that Stemming and Lemmatization didnt much contribute to greater accuracy. Suggesting it may not be an optimal technique to apply in

this case. Cross validation findings did not meet the expectation, however it increased our curiousity to explore further.

As conclusion, With the dataset given - our expected accuracy in the range 60 to 80% is achieved. However, with better data quality and balanced distribution

and with more training, compute and time - we can choose a better model to improve the accuracy greater than 90%. Overall, an exciting topic to explore the

fundamentals of NLP techniques and their effectiveness in realtime applications such as Sentiment analysis.

**Next Steps**

This data is limited in that the training was based upon responses to movie review data and may not generalize well outside of the domain. We also noticed

some significant overfitting when the model was cross validated or applied to the Kaggle test data.

Future work could include more features such as negation, POS tagging, further cleaning the dataset and additional models could be applied. SVM and

Tensorflow were also taking more compute and processing time.

Future work could include more features such as negation, POS tagging, further cleaning the dataset and additional models could be applied.

**Appendix Models**

**Multinomial Naive Bayes**

*Stemmed:*

*precision    recall  f1-score   support*

*0       0.37      0.49      0.42      2135*

*1       0.43      0.54      0.48      8644*

*2       0.82      0.70      0.75     37228*

*3       0.49      0.55      0.52     11897*

*4       0.35      0.52      0.42      2520*

*accuracy                           0.63     62424*

*macro avg       0.49      0.56      0.52     62424*

*weighted avg       0.67      0.63      0.64     62424*

*----------------------------------------------------------------------------------------------------*

*[[ 1045   784   266    35     5]*

*[ 1123  4656  2349   455    61]*

*[  553  4806 25895  5337   637]*

*[   84   583  2978  6490  1762]*

*[    8    44   256   903  1309]]*

*----------------------------------------------------------------------------------------------------*

*0.6572351980007689*

*0.6310874022811739*

**5-fold Cross validation:**

Accuracy: 0.56 (+/- 0.02)

*Lemmatized:*

*precision    recall  f1-score   support*

*0       0.41      0.48      0.44      2371*

*1       0.45      0.54      0.49      9085*

*2       0.80      0.71      0.75     35900*

*3       0.51      0.55      0.53     12265*

*4       0.38      0.52      0.44      2803*

*accuracy                           0.64     62424*

*macro avg       0.51      0.56      0.53     62424*

*weighted avg       0.66      0.64      0.65     62424*

*----------------------------------------------------------------------------------------------------*

*[[ 1147   850   313    56     5]*

*[ 1102  4888  2578   456    61]*

*[  480  4508 25466  4902   544]*

*[   76   585  3092  6798  1714]*

*[    8    42   295  1008  1450]]*

*----------------------------------------------------------------------------------------------------*

*0.6684087850826606*

*0.636758298090478*

**5-fold Cross validation:**

Accuracy: 0.56 (+/- 0.02)

**Tensor Flow**

*Stemmed:*

Epoch 1/7

1951/1951 [==============================] - 391s 200ms/step - loss: 0.8964 - accuracy: 0.6376

Epoch 2/7

1951/1951 [==============================] - 381s 195ms/step - loss: 0.7280 - accuracy: 0.7001

Epoch 3/7

1951/1951 [==============================] - 383s 196ms/step - loss: 0.6573 - accuracy: 0.7248

Epoch 4/7

1951/1951 [==============================] - 380s 195ms/step - loss: 0.6097 - accuracy: 0.7421

Epoch 5/7

1951/1951 [==============================] - 379s 194ms/step - loss: 0.5742 - accuracy: 0.7542

Epoch 6/7

1951/1951 [==============================] - 382s 196ms/step - loss: 0.5458 - accuracy: 0.7634

Epoch 7/7

1951/1951 [==============================] - 384s 197ms/step - loss: 0.5215 - accuracy: 0.7724

<tensorflow.python.keras.callbacks.History at 0x7fa97e1c5898>

**Random Forest**

*Stemmed:*

precision recall f1-score support

negative 0.57 0.73 0.64 5463

neutral 0.82 0.70 0.76 18754

positive 0.63 0.74 0.68 6995

accuracy 0.71 31212

macro avg 0.67 0.72 0.69 31212

weighted avg 0.73 0.71 0.72 31212

----------------------------------------------------------------------------------------------------

[[ 3966 1317 180]

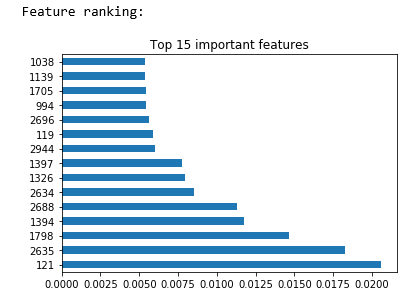
[ 2739 13123 2892]

[ 252 1542 5201]]

----------------------------------------------------------------------------------------------------

0.9059095860566448

0.7141484044598232



5-fold cross validation: Accuracy: 0.53 (+/- 0.02)

Turn the problem into a binary classification problem and greater accuracy is achieved.

**Multinomial Naive Bayes**

*Stemmed:*

  precision    recall  f1-score   support

           0       0.84      0.86      0.85     13470

           1       0.89      0.87      0.88     17122

    accuracy                           0.87     30592

   macro avg       0.86      0.87      0.87     30592

weighted avg       0.87      0.87      0.87     30592

----------------------------------------------------------------------------------------------------

[[11579  1891]

 [ 2178 14944]]

----------------------------------------------------------------------------------------------------

0.8771125494426465

0.866991370292887

**5-fold Cross validation:**

Accuracy: 0.79 (+/- 0.01)

*Lemmatized:*

     precision    recall  f1-score   support

           0       0.85      0.86      0.86     13520

           1       0.89      0.88      0.88     17072

    accuracy                           0.87     30592

   macro avg       0.87      0.87      0.87     30592

weighted avg       0.87      0.87      0.87     30592

----------------------------------------------------------------------------------------------------

[[11682  1838]

 [ 2075 14997]]

----------------------------------------------------------------------------------------------------

0.8842878624432022

0.8720907426778243

**5-fold Cross validation:**

Accuracy: 0.78 (+/- 0.01)

**Random Forest**

*Lemmatized:*

precision recall f1-score support

0 0.86 0.89 0.88 6532

1 0.92 0.89 0.90 8764

micro avg 0.89 0.89 0.89 15296

macro avg 0.89 0.89 0.89 15296

weighted avg 0.89 0.89 0.89 15296

----------------------------------------------------------------------------------------------------

[[5827 705]

[ 958 7806]]

----------------------------------------------------------------------------------------------------

0.9823314046615017

0.8912787656903766

**5-fold Cross validation:**

Accuracy: 0.75 (+/- 0.02)

**Logistic Regression**

*Lemmatized:*

precision recall f1-score support

0 0.85 0.88 0.87 6556

1 0.91 0.88 0.90 8740

micro avg 0.88 0.88 0.88 15296

macro avg 0.88 0.88 0.88 15296

weighted avg 0.88 0.88 0.88 15296

----------------------------------------------------------------------------------------------------

[[5775 781]

[1010 7730]]

----------------------------------------------------------------------------------------------------

0.9217089993789023

0.8829105648535565

**5-fold Cross validation:**

Accuracy: 0.79 (+/- 0.00)

**Tensor Flow**

*Stemmed:*

Epoch 1/7

956/956 [==============================] - 187s 195ms/step - loss: 0.3733 - accuracy: 0.8293

Epoch 2/7

956/956 [==============================] - 187s 196ms/step - loss: 0.1989 - accuracy: 0.9235

Epoch 3/7

956/956 [==============================] - 187s 196ms/step - loss: 0.1561 - accuracy: 0.9405

Epoch 4/7

956/956 [==============================] - 188s 196ms/step - loss: 0.1332 - accuracy: 0.9487

Epoch 5/7

956/956 [==============================] - 187s 196ms/step - loss: 0.1183 - accuracy: 0.9535

Epoch 6/7

956/956 [==============================] - 187s 196ms/step - loss: 0.1052 - accuracy: 0.9586

Epoch 7/7

956/956 [==============================] - 187s 195ms/step - loss: 0.0943 - accuracy: 0.9624

<tensorflow.python.keras.callbacks.History at 0x7fa97e250748>