Spandan Das

Riverus Assignment Report
Text Classification

The Task

The dataset consists of various comments/reviews, which insinuate either a positive or negative sentiment. Using this training data, a model is to be built to predict the sentiment based on the input text fed to the model.

A **Negative Sentiment** is denoted by **0**

A **Positive Sentiment** is denoted by **1**

Approach

Data Analysis:

1. First, the dataset is loaded onto a dataframe in Python. Once loaded, length distribution is analyzed for the two target variables, i.e., 0 an 1. A histogram depicting the same is plotted.
2. It is observed that in the corpus, negative sentiments (0) are generally expressed through longer sentences as compared to positive sentiments (1).
3. Next, word frequency distribution analysis is done to determine and derive insights about how the words are distributed and to determine the family of words used to convey the sentiments.
4. It is observed that occurrences of words such "*as", "I", "the", "it*" are high owing to the fact that they are pretty common in language usage. This tells us that some words may need to be removed from the corpus before building the model.
5. A few symbols like the double quote sign are also used frequently and add noise to the data.

Preprocessing:

1. Now that analysis is done, it is time to preprocess the data. This is done by utilizing the NLTK an regex library in Python.
2. First, punctuation marks and symbols such as, (}{“,=+-\ *; : ’ / .)!,* etc. are removed. Next, URLs are also taken care of.
3. Words commonly used in the English language such as pronouns, determiners, etc. are removed using stopwords.
4. Next, in order to make the training data denser and to reduce the size of the input data, that is, words, in this case, stemming is performed. It converts words to their respective stems.
5. Once the preprocessing is over, frequency analysis is performed one last time to check how our preprocessing has affected the dataset.

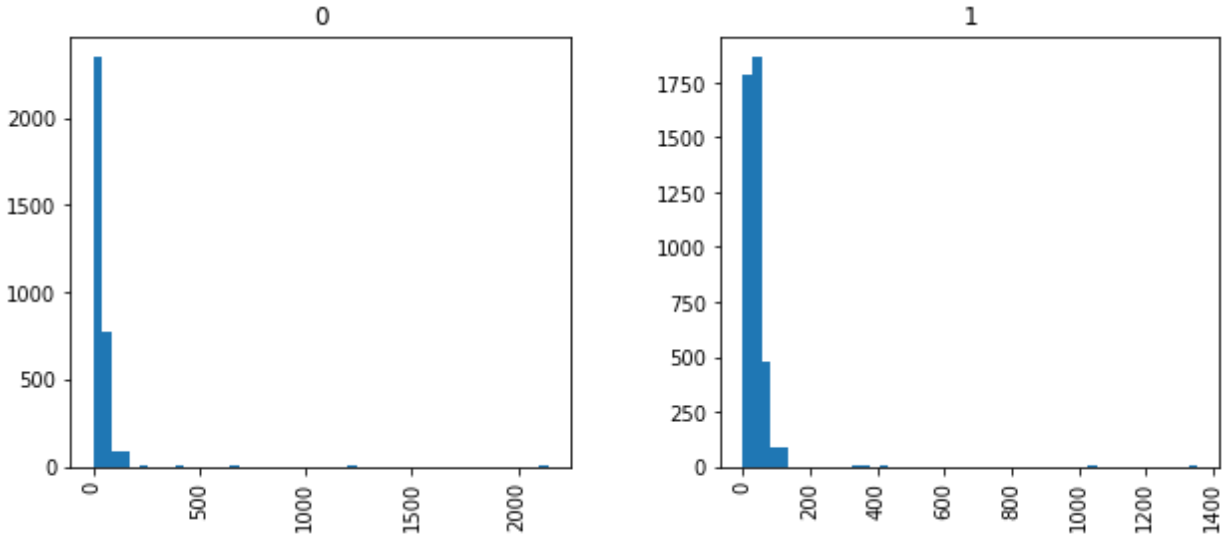

Figure: Length Distribution Histogram for the Two Target Variables

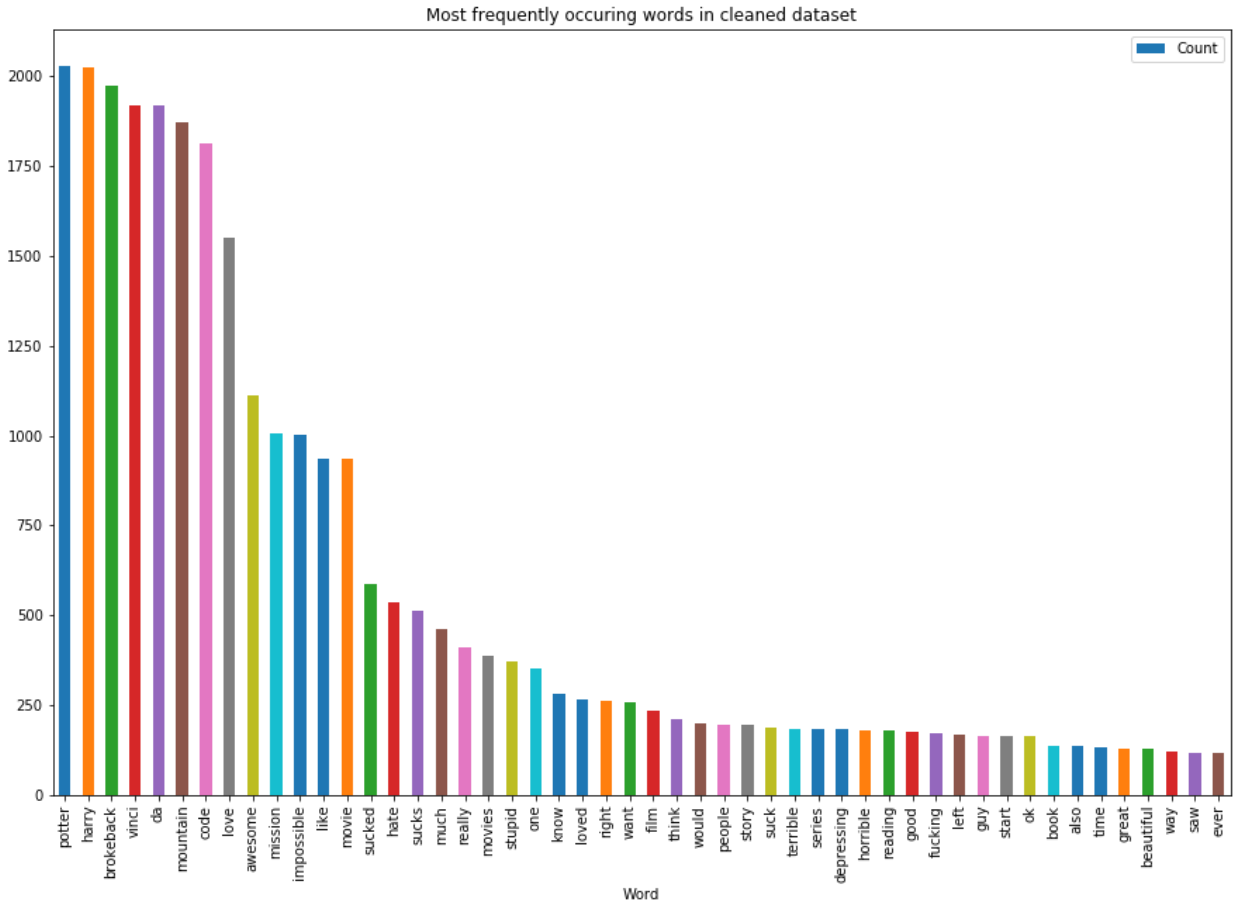

Figure: Top 50 Most Frequently Occuring Words

Building the Model:

1. Since the corpus consists of words which express the sentiments strongly, a Bag of Words approach did not seem enough.
2. To decode the sentiment expressed by the sentences better, TF-IDF with an n-gram range of 1-2 was used. For this, Tfidf Vectorizer was used. Tfidf vectorizer vectorizes the sentences and computes the term and inverse frequencies of the words. It helps in determining the important features in the corpus by assigning weights to the terms.
3. Once this is computed, the dataset is then split into training and validation data.

4. The training data is fed into a Logistic Regression and a Random Forest Classifier model.
5. The classification reports for both the models are then generated and the confusion matrix is created for both the model predictions.
6. It is observed that the performance of both the models, in this case, is comparable. Logistic Regression in some cases does perform slightly better than Random Forest.
7. The model is saved and used to predict the data present in the testing file and the predictions are then saved.

Classification Report and Confusion Matrix Obtained from the Models

	precision	recall	f1-score	support
0	0.99	0.94	0.96	680
1	0.95	0.99	0.97	844
micro avg	0.97	0.97	0.97	1524
macro avg	0.97	0.96	0.97	1524
weighted avg	0.97	0.97	0.97	1524

Figure: Classification Report of Logistic Regression

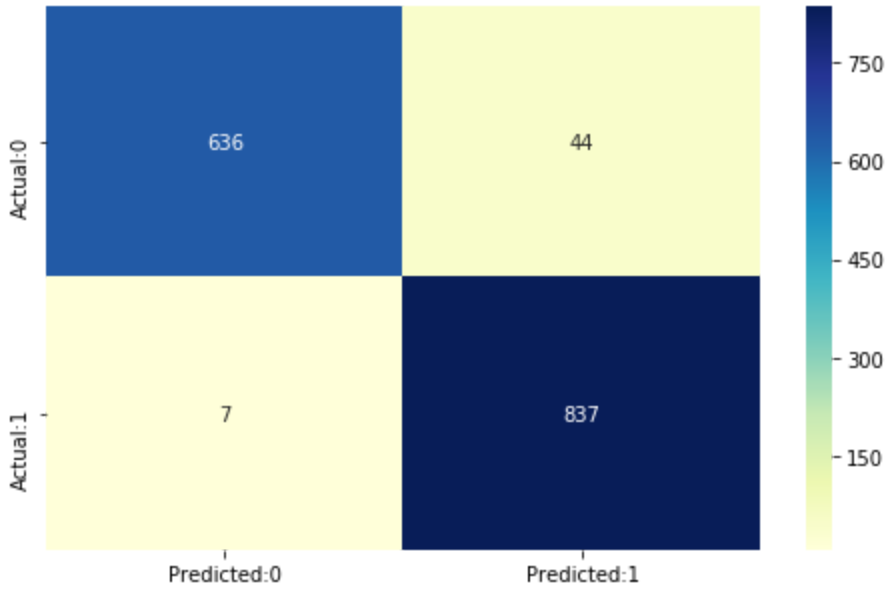

Figure: Confusion Matrix obtained from Logistic Regression Model

	precision	recall	f1-score	support
0	0.99	0.92	0.96	680
1	0.94	1.00	0.97	844
micro avg	0.96	0.96	0.96	1524
macro avg	0.97	0.96	0.96	1524
weighted avg	0.96	0.96	0.96	1524

Figure: Classification Report of Random Forest Classifier

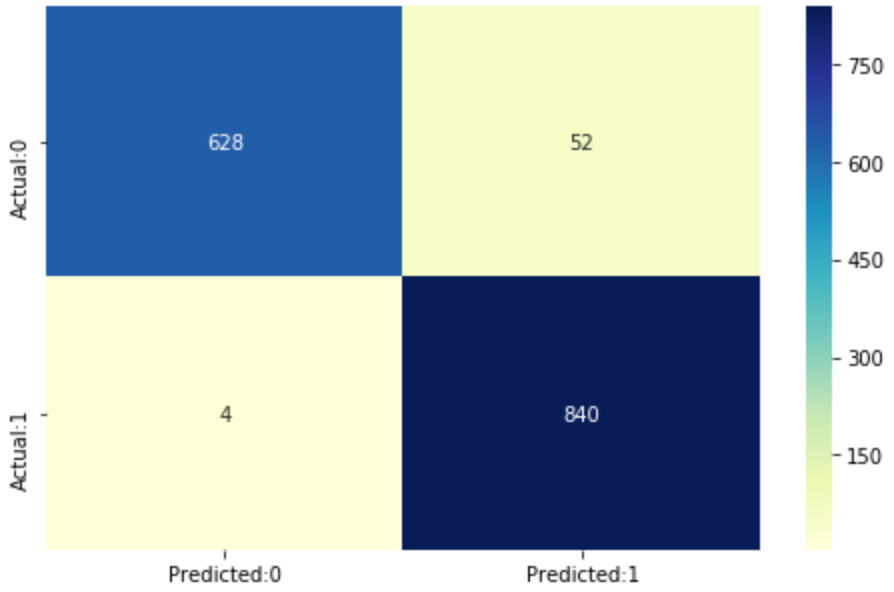

Figure: Confusion Matrix obtained from Random Forest Classifier Model