Sentiment Analysis of Movie Reviews using Natural Language Processing with Toxic Comment Detection

Love Vishwas Jeswani, Navya Rajashekhar Sogi, Sushmitha Srinath Kenkare

University of Texas, Arlington, Texas

Abstract

The word ‘Sentiment’ can be described as the meaning of a word or sequence of words that is often associated with an emotion. Analyzing emotions and attitudes behind any form of text is known as Sentiment Analysis, which is also called as Opinion Mining. In recent years, Sentiment Analysis has been the most researched areas in natural language processing. Gauging the opinions of a person or a group of people on certain topics can be very helpful for making business, economic and political decisions. The main source of data for sentiment analysis is social media where users express what they feel about a particular post/picture/movie/restaurant etc. through likes, comments and reviews. A downside to having opportunity to express opinions freely on social media is conversation toxicity where people use rather hurtful/abusive/hateful comments to express their dissatisfaction with a service/ business or disagreement with a post/ person. This can potentially affect the confidence and self-esteem of the person on the receiving end of the negative comments and can lead to people stop seeking others’ help and opinions out of fear of online harassment and abuse.

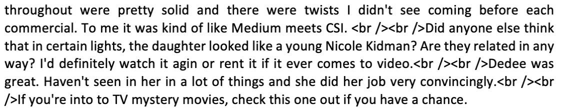
Introduction

In this project we have implemented Sentiment Analysis on IMDb movie review dataset using natural language processing with added functionality of toxic comment detection. The goal of this project is to identify the underlying sentiment of a movie review based on the textual information present and to achieve it we have classified whether a reviewer liked the movie or not. Keyword spotting has been used to analyze data in this project. Although, keyword spotting is considered the most naïve approach, its accessibility and economy make it popular. This approach classifies text by affect categories based on the unambiguous affect words such as happy, sad, afraid or bored. For Data pre-processing, vectorization and normalization techniques - Stemming and Lemmatization have been implemented. Results are evaluated by comparing Accuracy, F1-Scores and confusion matrix obtained by applying four feature extraction methods - *Bag of Words, N-gram, Word Counts, TF-IDF,* across five models - *Logistic Regression, Naïve Bayes, Decision Trees, Linear SVC and Random Forest*. This project aims at exploring multiple classification models, comparing their results and analyzing which model is the best with the best feature extraction technique. The Toxic Comment Detection model predicts if the input text is appropriate or toxic. This is achieved using Logistic Regression Classifier and Accuracy Score is used as the evaluation metric.

Dataset Description

Dataset used for this project is the Large Movie Review Dataset that contains 50,000 IMDb movie reviews. This large dataset has been further split evenly into 25,000 reviews to train and 25,000 reviews to test the classifier. These 25,000 reviews in training and testing datasets are comprised of 12,500 positive and 12,500 negative reviews. A typical review from the dataset might look like this:

Figure 1: A typical movie review



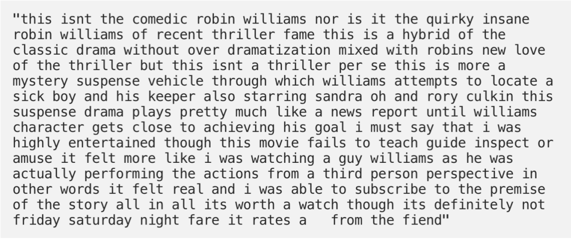
I'm a fan of TV movies in general and this was one of the good ones. The cast performances throughout were pretty solid and there were twists I didn't see coming before each commercial. To me it was kind of like Medium meets CSI. <br /><br />Did anyone else think that in certain lights, the daughter looked like a young Nicole Kidman? Are they related in any way? I'd definitely watch it agin or rent it if it ever comes to video.<br /><br />Dedee was great. Haven't seen in her in a lot of things and she did her job very convincingly.<br /><br />If you're into to TV mystery movies, check this one out if you have a chance.

IMDb lets users rate movies on a scale of 1 to 10. To label these reviews, the data curator has labelled any review that is ≤ 4 stars as negative and any review with ≥ 7 stars as positive. Reviews with 5- or 6-star ratings are omitted.

Project Description

Data Pre-processing. Raw data collected can be messy. So, before we perform any sort of analytics on data, it should be cleaned. In this project, punctuation marks and HTML tags have been removed using regular expressions/ pattern matching approaching. The text is also converted into lower-case for easy processing.

Figure 2: Text after cleaning

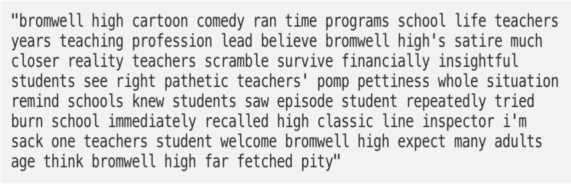


```
"this isnt the comedic robin williams nor is it the quirky insane
robin williams of recent thriller fame this is a hybrid of the
classic drama without over dramatization mixed with robins new love
of the thriller but this isnt a thriller per se this is more a
mystery suspense vehicle through which williams attempts to locate a
sick boy and his keeper also starring sandra oh and rory culkin this
suspense drama plays pretty much like a news report until williams
character gets close to achieving his goal i must say that i was
highly entertained though this movie fails to teach guide inspect or
amuse it felt more like i was watching a guy williams as he was
actually performing the actions from a third person perspective in
other words it felt real and i was able to subscribe to the premise
of the story all in all its worth a watch though its definitely not
friday saturday night fare it rates a   from the fiend"
```

Vectorization. In order to make the data understandable by the machine learning algorithm, each of the reviews must be converted into numeric representation. This process is known as vectorization. This is also implemented to make sure the algorithm works on multiple values at a time instead of working on one value.

Removing Stop Words. Stop words are common words like ‘if’, ‘but’, ‘we’, ‘he’, ‘she’, and ‘they’ which can usually be removed without changing the semantics of a text. By doing so, the performance of the model is improved. We have implemented this using NLTK’s full list of stop words as stop_words = ‘english’. As an alternative, we have also supplied our own list of stop words [‘in’, 'of', 'at', 'a', 'the'].

Figure 3: Text after removing stop words



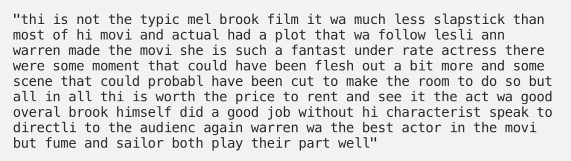
```
"bromwell high cartoon comedy ran time programs school life teachers
years teaching profession lead believe bromwell high's satire much
closer reality teachers scramble survive financially insightful
students see right pathetic teachers' pomp pettiness whole situation
remind schools knew students saw episode student repeatedly tried
burn school immediately recalled high classic line inspector i'm
sack one teachers student welcome bromwell high expect many adults
age think bromwell high far fetched pity"
```

Normalization. NLP model can be further enhanced by converting all the different forms of a given word into one. This process is called as Normalization. Normalization can be done in two ways: Stemming and Lemmatization.

- **Stemming.** Stemming involves removal of affixes (beginning or the end of a word) to extract the base form of the word. For example, consider the words, ‘eating’, ‘eaten’ and ‘eats’. The stem for the above-mentioned words is ‘eat’. This is obtained by removing the last few letters of the words. There are multiple algorithms available to determine how many letters need to be chopped off at the end. But the algorithms do not know the meaning of the word. Search engines use stemming for indexing the words to reduce the size of the index and to increase the word retrieval rate.

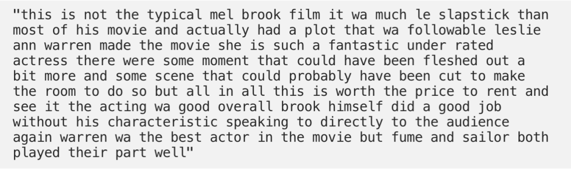
- **Lemmatization.** Lemmatization is similar to stemming. The output of this technique is referred to as a ‘lemma’. But the difference is that the algorithms have the knowledge of meaning of words in lemmatization. We can say that the algorithms refer to a dictionary to understand the meaning of the word before reducing it to a root word or lemma. For example, the algorithm would identify that the word ‘better’ is derived from the lemma ‘good’.

Figure 4: Stemmed Text



```
"thi is not the typic mel brook film it wa much less slapstick than
most of hi movi and actual had a plot that wa follow lesli ann
warren made the movi she is such a fantast under rate actress there
were some moment that could have been flesh out a bit more and some
scene that could probabl have been cut to make the room to do so but
all in all thi is worth the price to rent and see it the act wa good
overal brook himself did a good job without hi characterist speak to
directli to the audienc again warren wa the best actor in the movi
but fume and sailor both play their part well"
```

Figure 5: Lemmatized Text



```
"this is not the typical mel brook film it wa much le slapstick than
most of his movie and actually had a plot that wa followable leslie
ann warren made the movie she is such a fantastic under rated
actress there were some moment that could have been fleshed out a
bit more and some scene that could probably have been cut to make
the room to do so but all in all this is worth the price to rent and
see it the acting wa good overall brook himself did a good job
without his characteristic speaking to directly to the audience
again warren wa the best actor in the movie but fume and sailor both
played their part well"
```

Feature Extraction methods used

Bag of Words. This model calculates the frequency of the word occurrences in a text file. The order of the words does not hold importance in Bag of Words, but the model only cares about what words appear in the text. It can be used on multiple levels by using 500, 5000, 50000 words, and so on.

N-Gram: it is a sequence of N-words in a sentence. N is an integer which stands for the number of words in the sequence. When N = 1, N=2, N=3, it is referred to as uni-gram, bigram and tri-gram respectively. N-gram is used because unlike in bag of words, the order in which the words appear is important. For example, it is a good idea to consider bigrams like "New York" instead of splitting them into individual words like "New" and "York".

Word Count. We can just note the number of times a particular a given word appears in a text file instead of finding if a word appears in the file or not. This gives a model much more predictive power.

Term Frequency, Inverse Document Frequency (TF-IDF). This is the most popular way to represent documents as feature vectors. TF-IDF measures how important a word is with respect to a specific text file or document. Term Frequency measures the counts of each word in a document out of all the words in the same document. Inverse Document Frequency measures how important a word is by considering the frequency of the word in the entire corpus.

Accuracy and F1 Scores of models like Logistic Regression, Naïve Bayes, Decision Tree, Linear SVC and Random Forests are calculated for the above-mentioned feature extraction methods for result evaluation.

Toxic Comment Detection

Main goal is to create a classifier model that predicts if input text is inappropriate (toxic). Logistic Regression Classifier is used in order to achieve this. Evaluation metric for Toxic Comment Detection model is accuracy score.

Main Reference Paper

https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/fa15/003.pdf

The main paper that we have used as a reference, aims at classifying whether a person likes the movie or not based on their review on the movie. Large Movie Review Dataset has been preprocessed using regular expression matching to remove punctuation marks and stop words have been eliminated. Exploratory analysis has been conducted on the dataset by plotting graphs for counting number of words in a review, occurrence of words across reviews and calculating review index. Bag of words, N-gram modeling and TF-IDF Feature Extraction methods have been used. Bag of words has been implemented on different levels like using 5000 words, 50000 words, etc and Unigram, Bigram and Mixed N-gram modeling has been used. Various classifiers like Naive Bayes, KNN, Random Forest, Logistic Regression and SGD Classifiers have been analyzed with the above mentioned feature extraction methods and results have been tabulated. Conclusions based on the above results have been made with suggestions on how to improve performance.

Difference in APPROACH/METHOD between your project and main projects of your references

- Vectorization: Numerical representation of data to make sure algorithm works on multiple values at a time, to ensure increased execution speed and can make the process of computation much faster and optimized.
- Normalization (Lemmatization and Stemming): Converting all forms of a given word into one to enhance NLP model.
- Word count feature extraction method.
- Regularization for better results.
- Decision Tree and Linear SVC Classifiers used for better comparison and analysis.
- Word Count feature extraction added that gives faster results and better accuracy scores.

Difference in ACCURACY/ PERFORMANCE between your project and main projects of your references

In reference paper – Logistic Regression model has the best performance across all feature representations with an average accuracy of 88%.

In our project – Logistic Regression model has the best performance with an average accuracy of 89.9% (~ 90%).

Figure 6: Bag of Words Accuracy scores

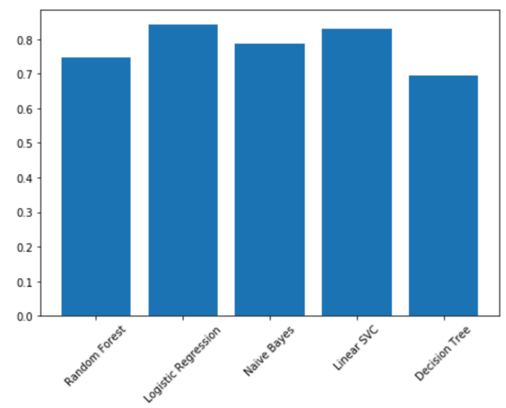


Figure 7: N-gram Accuracy Scores

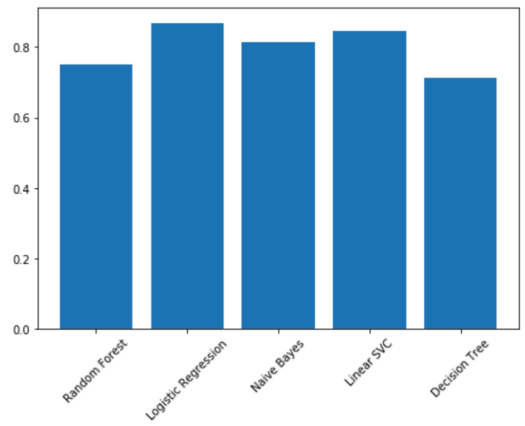


Figure 8: Word Count Accuracy scores

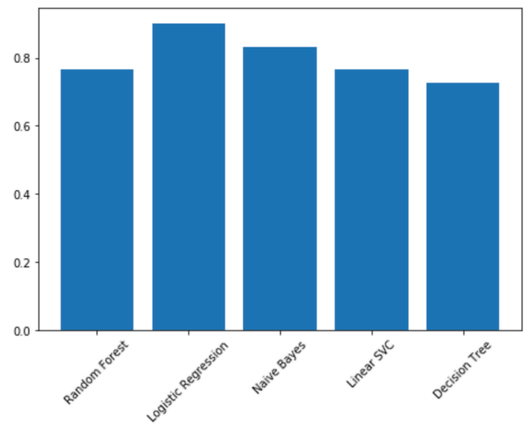


Figure 9: TF-IDF Accuracy scores

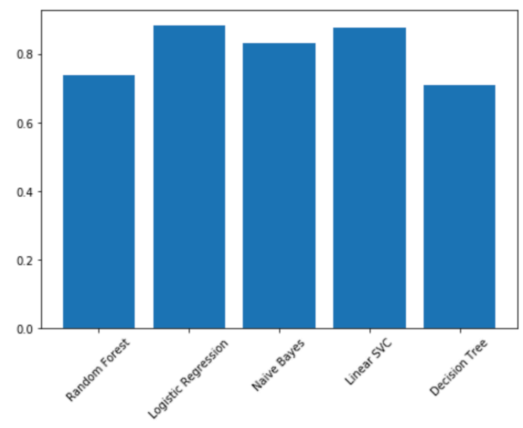


Figure 10: Bag of Words F1 scores

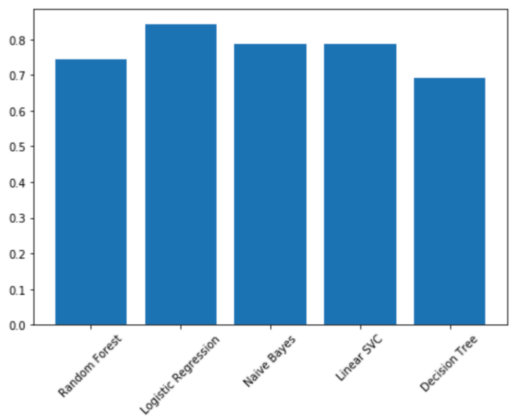


Figure 11: N-gram F1 scores

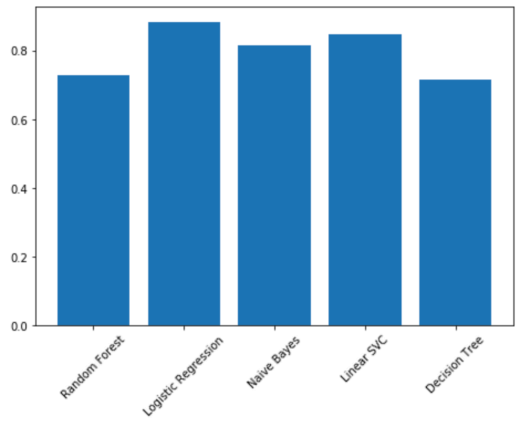


Figure 12: Word Count F1 scores

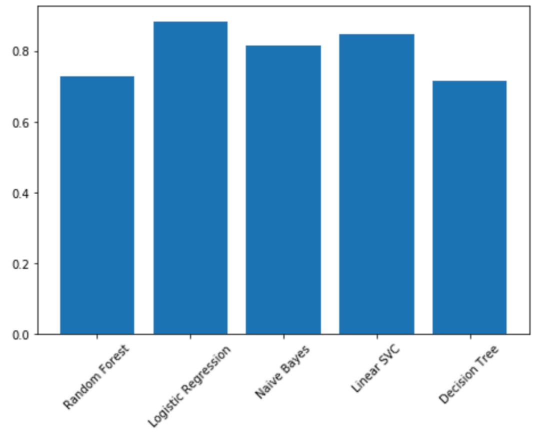
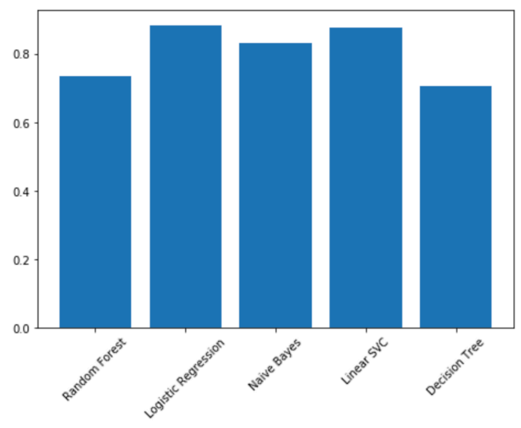


Figure 13: TF-IDF F1 scores



From the above graphs, it can be observed that Logistic Regression and Linear SVC Classifiers perform the best with most of the feature extraction methods except for TF-IDF method in terms of Accuracy scores. Logistic Regression and Linear SVC Classifiers again perform the best in terms for F1 scores across all feature extraction methods.

Also, Naive Bayes classifier also performs considerably well for N-gram and TF-IDF methods, but for optimization, TF-IDF produces faster results.

List of your contributions in the project

- Added Vectorization and Normalization techniques – Stemming and Lemmatization for efficient Data Pre-processing.
- Implemented four feature extraction methods - N-gram, Bag of Words, Word Count and TF-IDF.
- Implemented five classifier models - Linear SVC, Naive Bayes, Random Forest, Logistic Regression and Decision Tree Models.
- Evaluated models using Accuracy score, F1 score and confusion matrix.
- Comparison of all the feature extraction methods and all the classifiers' accuracy scores, F1 scores through plotting graphs and tabulating the results.
- Analyzed the results and concluded which model and feature extraction method are the best and used this analysis for toxic comment classification.
- Added regularization for better results.
- Implemented Toxic Comment Classification where a user can input a review and check if it is positive or negative.

Analysis

What did I do well?

- Successful implemented Toxic Comment classification.
- Added Normalization techniques (Stemming and Lemmatization) and vectorization for efficient data pre-processing.
- Implemented word count feature extraction method.
- Added Regularization for better results.
- User inputs a review – Checks if it is positive or negative.
- Implemented multiple classifiers and feature extraction methods, analyzed the best ones and increased accuracy of the model by 2%.

What could I have done better?

- When the review entered is 'Not Bad', the algorithms classifies it as a negative review. An entire line could be read rather than just reading individual words.
- N-gram modeling, when used on Decision Tree classifier, can be optimized better for quick results.

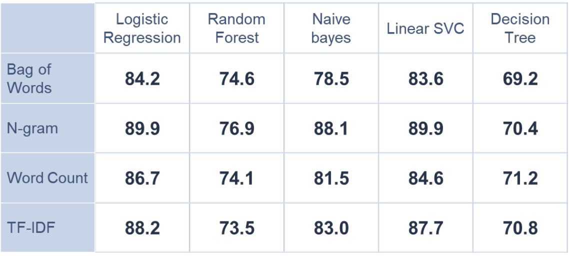
What is left for future work?

- Results from the above mentioned models can be used to create a Recommender System.
- Multi-Classification System – Classify sentiments of reviewers in more binary fashion like 'Happy', 'Bored', 'Afraid', etc. Regression techniques can be used to achieve this.

Conclusion

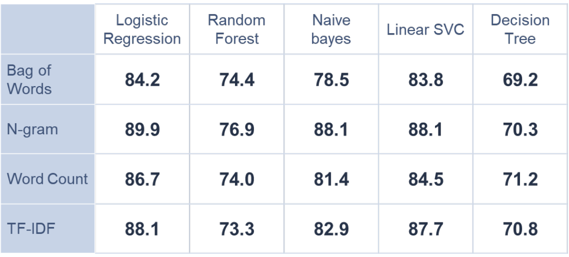
Model Evaluation Results

Table 1: Accuracy Scores (in %)



	Logistic Regression	Random Forest	Naive bayes	Linear SVC	Decision Tree
Bag of Words	84.2	74.6	78.5	83.6	69.2
N-gram	89.9	76.9	88.1	89.9	70.4
Word Count	86.7	74.1	81.5	84.6	71.2
TF-IDF	88.2	73.5	83.0	87.7	70.8

Table 2: F1 Scores (in %)



	Logistic Regression	Random Forest	Naive bayes	Linear SVC	Decision Tree
Bag of Words	84.2	74.4	78.5	83.8	69.2
N-gram	89.9	76.9	88.1	88.1	70.3
Word Count	86.7	74.0	81.4	84.5	71.2
TF-IDF	88.1	73.3	82.9	87.7	70.8

We have successfully implemented Sentiment Analysis using Natural Language Processing with added functionality of Toxic Comment Detection on Large Movie Review Dataset. Logistic Regression Model has better accuracy and F1 scores across all feature extraction representations, hence can be considered the best model. On further analyzing the results, Naive Bayes Classifier along with N-gram/TF-IDF can be used but since N-gram is slow in processing, TF-IDF serves as the next best method for optimized results. Decision Tree classifier has the least scores and is not recommended for this data.

References

1. https://www.researchgate.net/publication/321843804_Sentiment_Analysis_of_Movie_Reviews_using_Machine_Learning_Techniques
2. https://www.andrew.cmu.edu/user/angli2/li2019sentiment.pdf
3. https://www.lexalytics.com/lexablog/sentiment-accuracy-baseline-testing
4. https://towardsdatascience.com/classifying-toxicity-in-online-comment-forums-end-to-end-project-57720af39d0b
5. https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview/description
6. https://realpython.com/sentiment-analysis-python/
7. https://towardsdatascience.com/imdb-reviews-or-8143fe57c825