Sentiment Analysis of IMDB Movie Reviews

Abhimanyu Singh
asing134@binghamton.edu
SUNY Binghamton

Chaitanya Kulkarni
ckulkar2@binghamton.edu
SUNY Binghamton

Necati A. Ayan
nayan1@binghamton.edu
SUNY Binghamton

Abstract - **From the last 10 years, popularity of social media has increased at an alarming rate. Everyone is utilizing technology at higher rates than earlier. People are now sharing their emotions and opinions on social media sites allowing others to know what they think about a particular thing. Many companies are utilizing the data from various websites to generate meaningful information out of it which can be further used for business purposes. Huge textual data is available on sites like Amazon, IMDB, Rotten Tomatoes on movies and analyzing such massive data manually is a tedious task. So, to speed up the process, programmers use certain techniques to extract out public opinion. One of which is using sentiment analysis. Sentiment analysis is a submodule of opinion mining where the analysis focuses on the extraction of text and opinions of the people on a particular topic. We are making use of IMDB reviews on movies to predict how the users have rated the movies and predict the movies that have a positive or negative review. We proposed a model that includes different sentiment analysis methods which will help us to extract useful information from the data and predict which is the most suitable classifier for this particular domain by looking at accuracy. Models like Naïve Bayes, Support Vector Machine and Logistic regressions. Due to the lack of strong grammatical formats in movie reviews which is an informal jargon we also take into account the N-Grams and count vectorizer approach. Tokenization is used to transfer the input string into a word vector, stemming is used for extracting the root of the words, while feature selection fetches the essential word and lastly classification is used to classify the movie as positive or negative.**

Keywords—**Sentiment Analysis, N-Grams, Count vectorizer, Tokenization, stemming, Naïve Bayes, Logistic Regression, Support Vector Machine, accuracy.**

1. INTRODUCTION

Movies are the most convenient ways to the people for entertainment. But only a few movies are successful and are rated high. There are many ratings websites that will help the movie fanatics to decide which movie they should watch and which they should not. Websites like IMDB, Rotten tomatoes, etc. are the leading ones amongst those. The rating on these websites determine the success of the movie by giving it a score out of 10 based on the stars given by the viewers. But, there isn’t any method that can provide the prediction based on movie reviews. So, to determine the success of the movie based on reviews, sentiment analysis comes into picture.

Sentiment analysis is the interpretation and classification of emotions within text data using text analysis techniques. Sentiment analysis allows businesses to identify customer sentiment toward products, brands or services in online conversations and feedback. Sentiment analysis models focus on polarity (positive, negative, neutral) but also on feelings and emotions (angry, happy, sad, etc), and even on intentions (e.g. interested v. not interested). Sentiment analysis has become a hot topic and many big companies are investing their resources to predict the results for their businesses.

The working principle of sentiment analysis includes tokenization, word filtering, stemming and classifications. In tokenization, text needs to be segmented into units such as words/ numbers or punctuations. Next step stemming which is the process of removing prefixes and affixes to convert a particular word into its stem. After preprocessing, we analyze the dataset by performing classification using Naïve Bayes, Support Vector Machine and Logistic Regression. Here, we determine the best model based on accuracy. Hence, We analyze and study the features that affect the scores of our review text and finally classify the movie as positive or negative.

2. RELATED WORK

1. Unggul Widodo Wijayanto, Riyanarto Sarno: This paper focuses on supervised methods. To improve the quality authors have also utilised CHI2 and stop words. Models like K-folds, cross validation to get results. The authors conclude that context-based stop words enrich the number of stop words that removes bias features. [3].

2. Sourav Mehra, Tanupriya Choudhary: In this paper authors have implemented SVM and Naïve Bayes and comparison is done between by observing the accuracies of the model They have taken data of IMDB movie reviews which possess of 25000 each for positive and negative provided by the Cornell University The authors concluded by stating that SVM has better accuracy over Naïve Bayes 87.33%. [5].

3. DATA DESCRIPTION

We have gathered data from [7] which includes a dataset that has 50000 reviews from IMDB which is equally divided into 25000 for training and testing. There are only 30 reviews per movie as reviews for the same movie tend to have correlated ratings. Furthermore, the train and test sets contain a disjoint set of movies so memorizing a particular movie terms and their associated labels would have no significance. A negative review is given a score of <=4 out of 10 while a positive one holds a score of >=7 and a neutral review has scores from >4 and <7.

4. PROPOSED FRAMEWORK

The proposed framework for our model includes data cleaning, data pre-processing, applying classifiers on the data and finally comparing the results from the different classification models we used.

A. Data Pre-Processing

In order to improve the performance of our model we have done some operations on the data that we have collected. We have removed the unnecessary noise from the data which will help in classification of our model. It includes the following procedure:

- **Removing HTML tags:** The dataset has some unnecessary html tags which might affect the accuracy of our model. Hence, we have used regex to remove the tags.

```
def rmvhtmltags(text):
    remreg = re.compile('<.*?>')
    cleartext = re.sub(remreg, '', text)
    return text

def remove_urls (vTEXT):
    vTEXT = re.sub(r'(https|http)?:\/\/(\w|
\.|\/|\?|\=|\&|\%)*\b', '', vTEXT, flags=re
.MULTILINE)
    return(vTEXT)
```

- **Lemmatization:** It is a process of converting the given word to its root word. The main objective of lemmatization is to get proper morphological meaning of a word by referring it to the dictionary which is incorporated in the library. We have used wordnet and porter stemmer lemmatization.

```
def lemmatize_words(text):
    lemmatized_words = [lemmatizer.lemmatiz
e(word, 'v') for word in text.split()]
    return('  '.join(lemmatized_words))
```

- **Removing Stop words and special characters:** The data consist of stop words like “a”, “the”, “this”, “that”, etc. These words mostly appear in a lot of reviews and are unimportant.

```
def rmvspclcharacter(text):
    clearspcl = re.sub(r'[^A-Za-z0-9\s.]',
r'', str(text).lower())
    clearspcl = re.sub(r'\n', r' ', text)

    clearspcl = " ".join([word for word in
text.split() if word not in stopWords])
    return text
```

- **Text Tokenization:** Tokenization is the process of tokenizing or splitting a string, text into a list of tokens. One can think of a token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph. For tokenization we have used NLTK library. It consists of many languages like German, English, Spanish, French, etc. trained with NLTK. In NLTK word tokenization is a wrapper function that utilizes treebank tokenization and splits the punctuations other than periods.

B. Features Extraction

Feature extraction is the process of converting a word into a matrix form. We have used the following approaches for feature extraction:

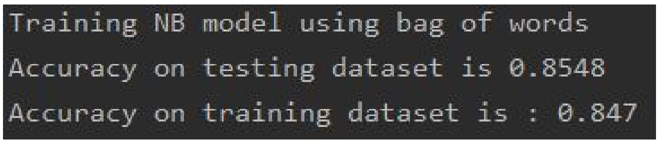
- **Bag of Words Approach:** The bag-of-words (BOW) model is a representation that turns arbitrary text into fixed-length vectors by counting how many times each word appears. This process is often referred to as vectorization. We have used following method for vectorization:
  - **Count Vectorizer:** The process includes a blend of tokenizing a collection of documents from the datasets and then building a set of vocabulary for those words. The result of this is a length of vocabulary words and an integer value assigned for words according to how many times they appear. Words that do not occur may possess zeros as value and are defined as sparse.

C. Classification Models

In our experiment we have made use of Naïve Bayes, Logistic Regression, and Support Vector Machine. We have trained our model on the above classifiers to predict the movie polarity as positive or negative.

Naïve Bayes: It is a classification algorithm, primarily used for text classification involving high dimensional training data sets. Example spam filtering, sentiment analysis etc. This algorithm learns the probability of an object with certain features belonging to a particular class. It is a probabilistic classifier. This algorithm is called Naive Bayes because it makes a naive assumption that occurence of certain features is independent of each other which in reality is not the case.

P(A/B) = P(B/A) P(A) / P(B)
A is called the proposition and B is called the evidence. P(A) is called prior probability of proposition and P(B) is called prior probability of evidence. P(A/B) is called the posterior. P(B/A) is called the likelihood.
P(A/B) = Probability of occurrence of event A, given event B has already occured
P(A) = Probability of event A
P(B) = Probability of event B
P(B/A) = Probability of occurrence of event B, given event A has already occured
In Naive Bayes with count Vectorizer we get an accuracy of 85.48%.



```
Training NB model using bag of words
Accuracy on testing dataset is 0.8548
Accuracy on training dataset is : 0.847
```

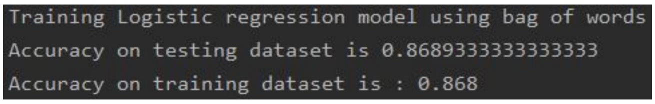
- **Logistic Regression:** Logistic regression is quite similar to linear regression but here, instead of fitting a line to our data we try to fit ‘S’ shaped logistic function(Sigmoid Function). Although it’s name contains regression, on the contrary it is used for classification purposes. Logistic regrssions’s capability to classify data using continuous and discrete measurements makes it a popular machine learning algorithm. Logistic regression uses something called maximum likelihood to fit data. It can be used to classify samples and can use different kinds of data to classify samples. It can also be used to assess what variables are useful for classifying samples.

Sigmoid function

$$Y= e^{(b0 + b1 * x)} / (1 + e^{(b0 + b1 *x)})$$

Here b0 is the bias and b1 is the coefficient for the value x and y.

In logistic Regression with Count Vectorizer we get an accuracy of 86.89 %

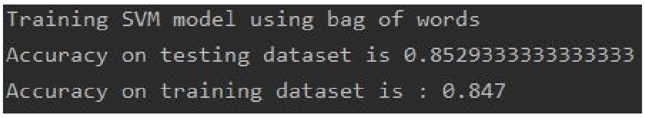


```
Training Logistic regression model using bag of words
Accuracy on testing dataset is 0.8689333333333333
Accuracy on training dataset is : 0.868
```

- **Support Vector Machine:** SVM is a regression and classification algorithm. It constructs a hyperplane or set of hyperplanes in infinite dimensional space to do the classifications. It looks at the extremes of the data set and draws a decision boundary

(Hyperplane). SVM is known for its good performance. It finds the distance between the two given observations which is then followed by search for a decision boundary in order to get distance between the closest members of the separate class. SVM are robust in case of overfitting of the model. Here, we have used SVM from the scikit-learn library. The support vector machines in scikit-learn support both dense and sparse sample vectors as input.

In SVM, with Countvectorizer where we get an accuracy score of 85.29%.



```
Training SVM model using bag of words
Accuracy on testing dataset is 0.8529333333333333
Accuracy on training dataset is : 0.847
```

5. RESULTS EVALUATION

We have compared the results of the classification models based on accuracy.

Accuracy: It is simply the ratio of correctly predicted observations to the total number of observations. We can say that the Higher the accuracy, the better the model. The accuracy is given by

Accuracy = TP + TN / TP + TN + FP + FN.

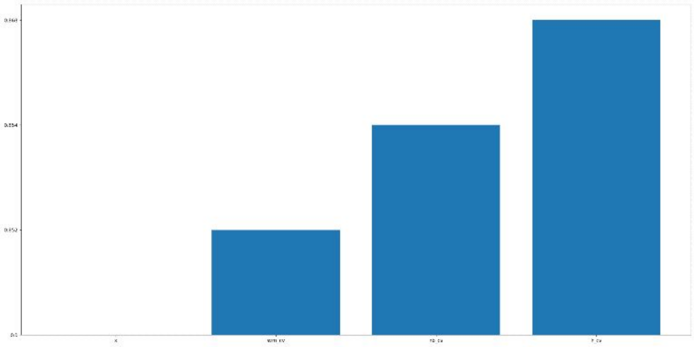
The analysis of accuracies of all the models is given below:

nb_cv = Naïve Bayes with Count Vectorizer = 85.48

svm_cv = SVM with Count Vectorizer = 85.29

lr_cv = Logistic Regression with Count Vectorizer = 86.89

Accuracy graph –



6. CONCLUSION

The main motive behind this project was to construct a sentiment analysis model that will help us to get a better understanding of movie reviews that we have collected, We compared the results of the 3 classifiers - Naive Bayes, Logistic Regression and Support Vector Machine (SVM). For Evaluation, we observed the accuracy provided by each model. By evaluating the models, we found out that Logistic Regression gives us the highest accuracy score of 86.89%.

7. REFERENCES

[1]. MaisYasen, Sara Tedmori. “Movies Reviews Sentiment Analysis and Classification”. IEEE Jordon International Joint Conference on Electrical Engineering and Information Technology (JEEIT). 978-1-5386-7942-5.

[2]. Tirath Prasad Sahu, Sanjeev Ahuja. “Sentiment Analysis of movie reviews: A study on feature selection and classification algorithms”. International Conference on Microelectronics, Computing, and Communication (MicroCom).978-1-4673-6621-2.

[3]. Wijayanto, Unggul and Sarno, Ritanarto. “An Experimental Study of Supervised Sentiment Analysis Using Gaussian Naïve Bayes”. 476-481.10.1109/ISEMANTIC.2018.8549788.

[4]. Tejaswini M. Untawale, G. Choudhari. “Implementation of Sentiment Classification of Movie Reviews by Supervised Machine Learning Approaches”. 978-1-5386-7808-4.

[5]. Sourav Mehra, Tanupriya Choudhary. “Sentiment Analysis of User Entered Text”. International Conference of Computational Techniques, Electronics and Mechanical Systems (CTEMS). 978-1-5386-7709-4.

[6]. Nisha Rathee, Nikita Joshi, Jaspreet Kaur. “Sentiment Analysis Using Machine Learning Techniques on Python”. 978-1-5386-2842-3 “https://ieeexplore.ieee.org/document/8663224”.

[7].https://www.researchgate.net/profile/Raouf_Ganda/publication/318975052_Deep_learning_for_sentence_classification/links/59cd37a30f7e9b454f9f9181/Deep-learning-for-sentence-classification.pdf

[8]. https://www.aclweb.org/anthology/P12-3020.pdf

[9].https://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2857/325