Sentiment Analysis of Movie Reviews Using Natural Language Processing

Christo Vijay, Shiva Prasad Reddy Katta

University of Texas at Arlington (UTA), Texas, United States  
[christo.vijay@mavs.uta.edu](mailto:christo.vijay@mavs.uta.edu), [shivaprasadredd.katta@mavs.uta.edu](mailto:shivaprasadredd.katta@mavs.uta.edu)

Abstract

Sentiment analysis is a field of study which employs techniques like natural language processing to analyse any text and give its emotion. With the vast advancements in technology, internet has found its way into every corner of the world thus generating ginormous amounts of data every day. For instance, there are millions of user reviews on movies which we can analyse and extract useful insights from them with the help of sentiment analysis. When it comes to finding the polarity of a sentence nothing can beat humans, however, when dealing with huge amounts of data, machines always outperform humans. Hence, it is better to train computers in such a way that their accuracy will come very close to that of humans. This paper uses different machine learning approaches like Multinomial Naïve Bayes, SVM, and KNN for sentiment analysis of movie reviews using natural language processing and aims to increase the accuracy with the help of various techniques.

Introduction

During these digital times, there is a lot of content for us to watch online, and we are always confused about what movie to watch. However, we can overcome this issue by looking at the vast number of reviews and ratings left by other people online. For a human, just by looking at the review, we can detect the sentiment of it. But, for a computer, it's not possible unless we train them.

With the internet being available to many people, the amount of digital foot-print everybody is leaving behind is huge. Instead of piling this data, one has to put this data to good use such that it is beneficial to all parties. Sentiment analysis does exactly this, and due to this very reason, there is a never-before rise in academic interest in this particular field of study in the past few years. Despite so many advancements in computing power and the approaches we employ to extract opinions, there is always room for improvement, and nothing is perfect.

Sentiment analysis is not just limited to finding the polarity of a movie review, but it can do much more. Just throw data it and it will spit out useful insights. For instance, a company might want to know the user feedback for a product. One can conclude whether the product was a hit or a miss by analysing the user reviews and mining the polarity of them. Coming back to the movie reviews, we used different classification methods like Multinomial Naive Bayes, SVM, and KNN on the given large movie dataset and concluded which method best suited the cause. We evaluated the performance of these methods using techniques like confusion matrix, classification report, and plain accuracy.

Dataset Description

The dataset we used for our project is ‘Large Movie Review Dataset v1.0’ which we have obtained from the Stanford University website. The dataset has a total of 50,000 reviews which are divided into training and testing sets. Each set consists of 25,000 reviews and are in-turn divided into positive and negative reviews. Each review is stored in a different text file, and any review with a score less than or equal to 4 is considered as negative and greater than or equal to 7 is positive.

The given data is in the form of text files. Hence, for convenience purposes, we extracted all the reviews from 50,000 text files and created two CSV files, each containing training, and testing data respectively. These datasets now have two attributes, namely, review and score. Any review with a rating less than or equal to 4 is assigned a score of '-1' and '1' is assigned to any review with a rating of greater than or equal to 7. Other than that, the data is already cleaned and is ready for our project.

Project Description

1. Description:

Natural Language Processing (or NLP) is the process of applying Machine Learning methods to text and language to classify the data. NLP also deals with predictive analysis. The main aim of our project is to perform sentiment analysis on the given large movie data set to find out whether the review is positive or negative. The first step is to clean the data by techniques like tokenization, stop words removal, and stemming. The next step is feature extraction where we vectorize the data using models like Bag of Words and Tfidf.

Finally, we apply the appropriate classifier to the vectorized training data and predict the test data. Confusion matrix can be used to measure the accuracy. The project aims at predicting the reviews as positive or negative using classifiers like Multinomial Naïve Bayes, SVM, and KNN. Different classification algorithms give different results for different data sets.

Hence, depending on the data set the classifier should be selected. We will be using the above-mentioned three classifiers and measure their accuracies using the confusion matrix method, and classification report. The one with the highest accuracy will the most suitable classifier.

Our project consists of four phases

a. Data Preparation

b. Feature Extraction

c. Classifiers

d. Evaluation

1. **Data Preparation**

**Tokenization:**

Tokenization is not as hard as it sounds. To keep things simple, it is the process of chopping the given text into small bits also called tokens. However, these tokens might not always be words: they can be numbers, punctuation, etc. The sentences are divided based on boundaries which means the content between the ending point of one word and the beginning of another word. In our case, this boundary is a white space. This process helps in the process of eliminating unwanted characters like punctations which sometimes do not add additional meaning to a sentence.

**Removing unwanted data:**

In NLP, data preparation or data pre-processing simply means converting raw text into a form that a machine can easily understand. Removing unwanted data is a part of this process. In our case, unwanted data can be anything like stop words, non-letter characters, HTML tags, etc. Coming to the stop words, these are the words that do not add much of a meaning to a sentence and can use valuable computing resources if not removed. Words such as 'a', 'an', 'the', etc., are known as stop words and are already available in the NLTK library. We can use appropriate functions from the NLTK library to remove these stop words. Also, we can add our own stop words depending on the data. Similarly, we need to remove other useless data like HTML tags, and also convert whole data into lower case.

**Stemming:**

Stemming is the process of deriving a root word which makes the classification much easier. For instance, words like 'cars', 'cats', 'caresses', etc., can be reduced to 'car', 'cat', 'caress' respectively. However, in some cases stemming might do more bad than good. For instance, 'laziness' is reduced to 'lazi' which doesn't make any sense. Hence, depending on the classifier and the data set we have to decide whether to use stemming or not.

1. **Feature Extraction**

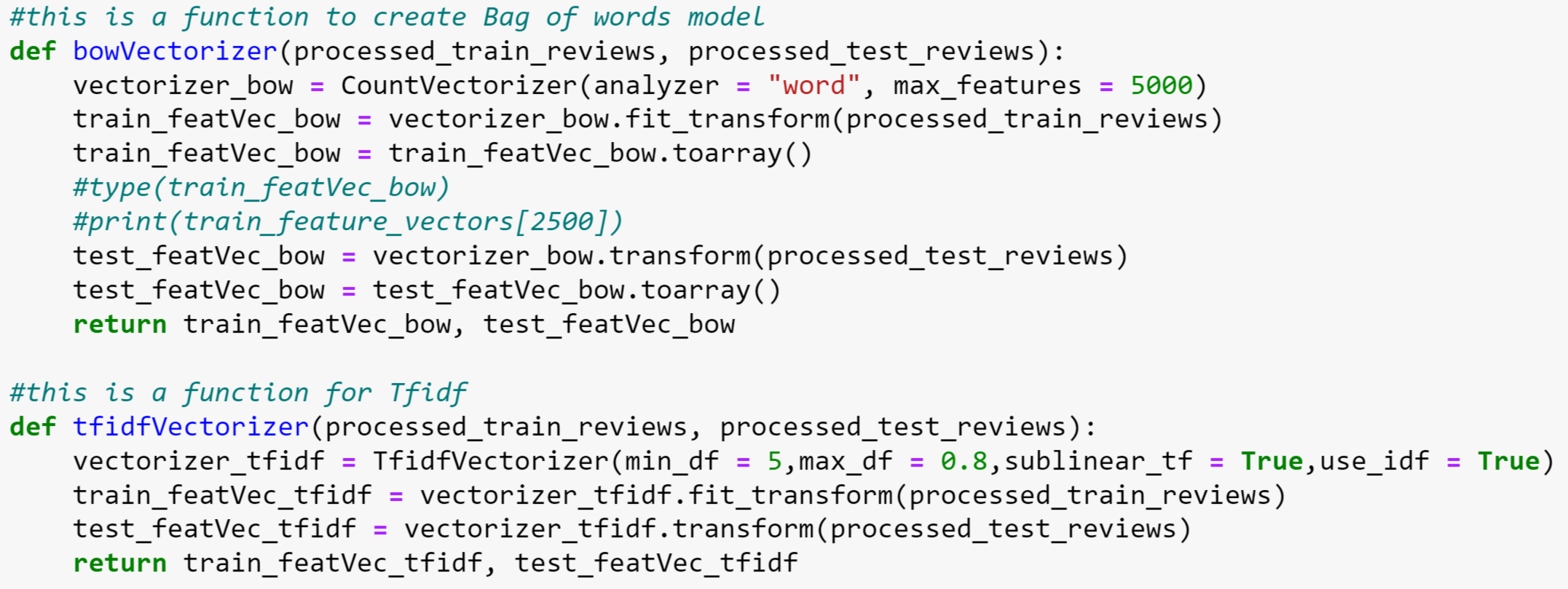
**Bag of words model:**

The bag of words model is a widely used feature extraction method. To keep things simple, it is a way of representing data by just showing whether the word is present or not. For instance, let's assume that a corpus has 20 unique words and 3 documents. Now, each document can be represented as a binary vector of 20 elements where each element can either be '0' or '1'. Here, zero means the word is absent in the document and one means the word is present in the document. In this model, the order of the words is discarded. This kind of model is not suitable for all classifiers, and depending on the classifier, different vectorization techniques must be employed.

**Tfidf:**

Tfidf stands for term frequency and inverse document frequency. This method addresses the problems of counting and frequencies. Term frequency means how frequently the word appears in the current document, and inverse document frequency means how sparse the word is spread across all the other documents in the corpus. This method will make sure that high-frequency words that do not add meaning to the sentence are considered as less important than rare words which have less frequency and add so much meaning to the sentence.

tf-idf(t, d) = tf(t, d) \* log(N/(df + 1))

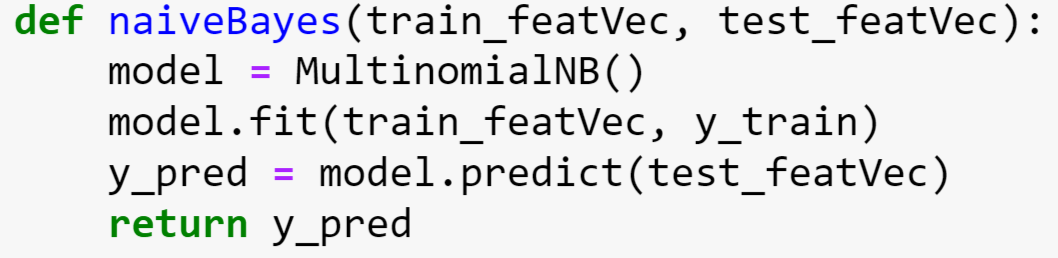


Code Snippet for Feature Extraction

1. **Classifiers**

**Multinomial Naïve-Bayes:**

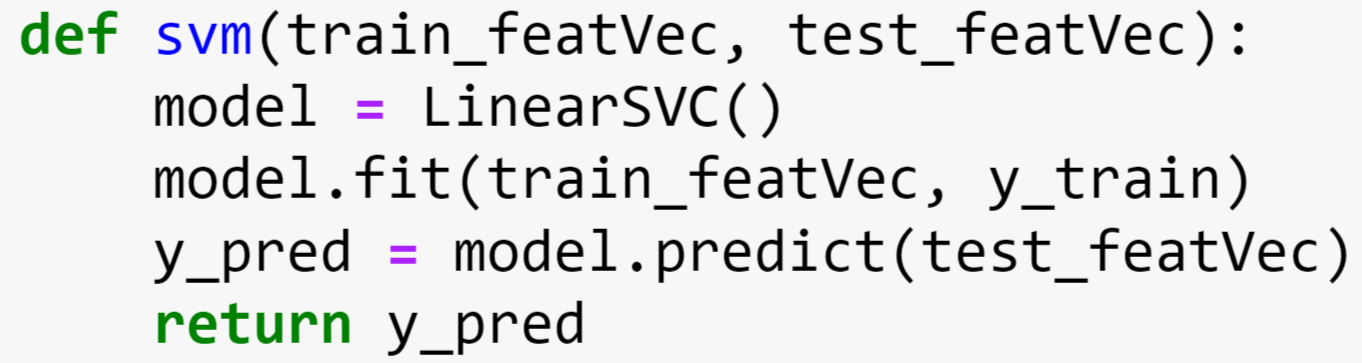
Naive Bayes is a simple machine learning classifier that is computationally inexpensive and is widely used because of its simplicity. It is a probabilistic classifier that assumes that features are independent, and due to this very reason it is called naive. Despite its simplicity, this algorithm is widely employed in complex problems like spam detection, sentiment extraction, etc. Multinomial Naive Bayes is just an extension to Naive Bayes where it is assumed that the conditional probability of a term given that it belongs to class c is equal to the relative frequency of the term present in documents classified as type c.



Naïve Bayes Implementation

**Support Vector Machines:**

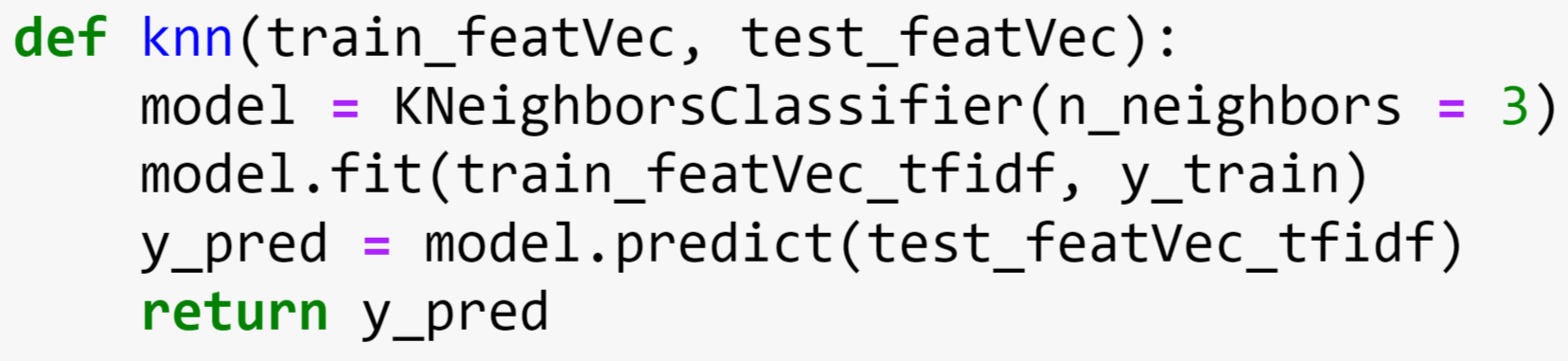
Support Vector Machines (SVM) is a supervised machine learning algorithm that can be used to classify data. Here, we need to find a hyperplane that categorizes the given data into a positive review and a negative review. We applied two different feature extraction methods for the SVM namely the bag-of-words model and Tfidf(Term Frequency and Inverse Document Frequency) statistic. However, we got higher accuracy when we employed Tfidf.



SVM Implementation

**K Nearest Neighbor:**

K Nearest Neighbor algorithm classifies the given data by initially finding the K nearest matches in training data and then the label of the closest matches are used to predict the test data. This is done by using distance functions. Out of all the three classifiers employed, KNN fetched the least accuracy score.



KNN Implementation

1. **Evaluation**

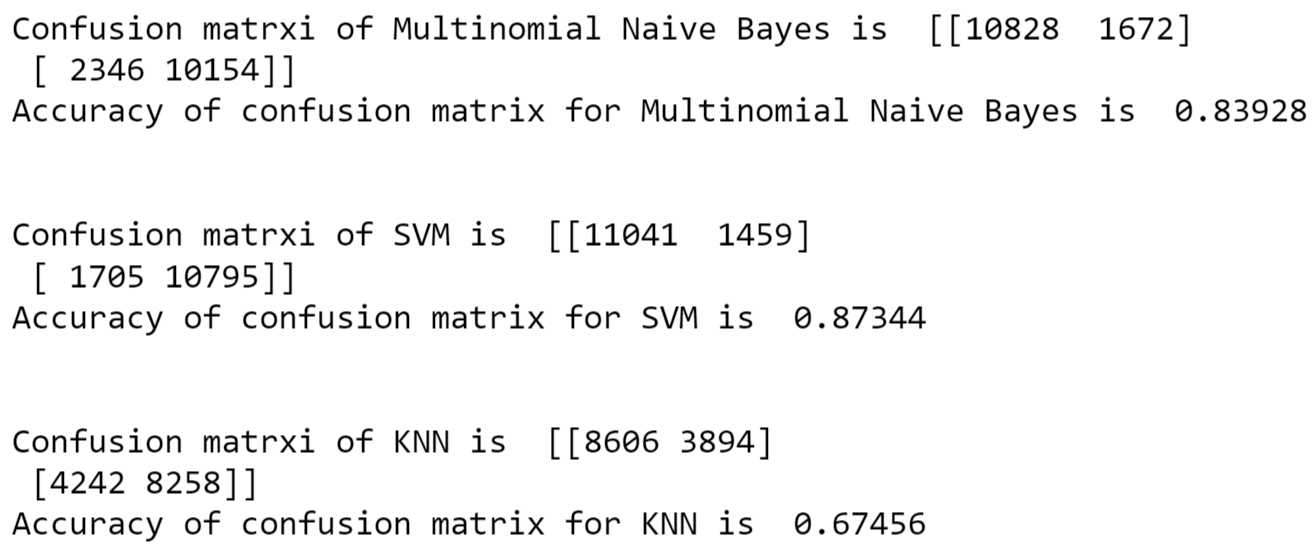
For evaluation, we used three different methods. First one is the simple accuracy method which just means how often the classifier is right. The second method is the confusion matrix method which is based on the information about true positives, true negatives, false positives, and false negatives. The third method is the classification report where the information about precision, recall, f1 score are generated.

**Accuracy:**

This is a simple method that is based on how many times the classifier classified correctly. This method gave the scores of 87% for SVM, 84% for Multinomial Naïve Bayes, and 67% for KNN

**Confusion Matrix:**

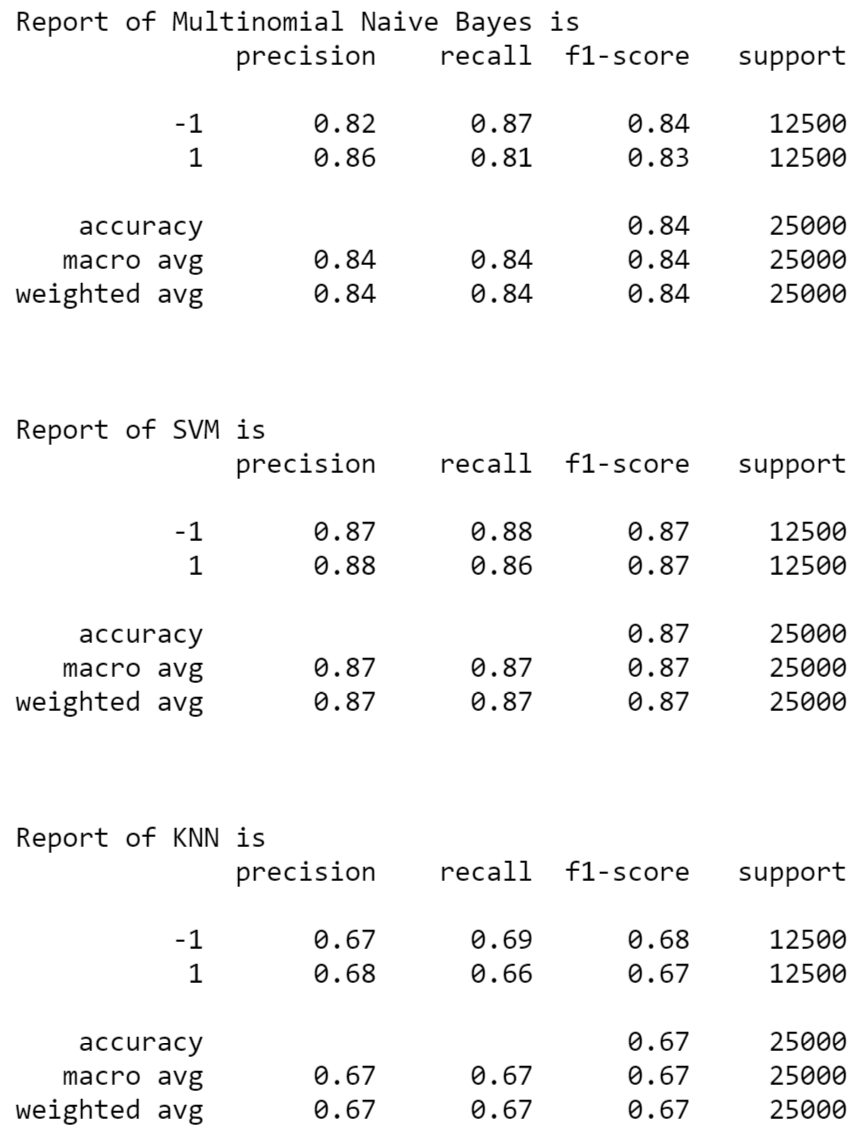
The confusion matrix method fetched the following results.



Confusion Matrix Results

**Classification report:**

A classification report involves precision, recall, and f1 score which give more insights than the plain accuracy. However, for the given data, all the three evaluation methods fetched almost similar results.



Classification Report Result

1. **Main References Description**

In [1] we learned the basic outline of how the sentiment analysis is performed using natural language processing. Also, major techniques like feature extraction, supervised machine learning models are learnt from this reference

In [2] and [3] we came to know about how different classifiers work and what is the underlying theoretical approach in those classifiers.

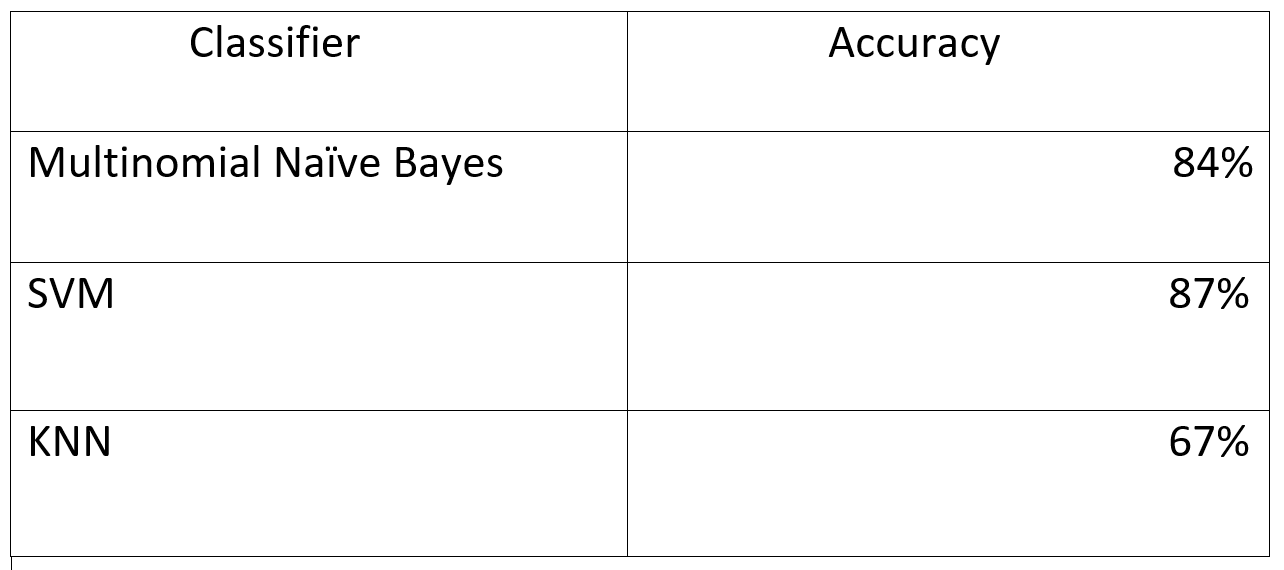
In [4] and [5] we learned how the classifiers are implemented in real world, and how the data is pre-processed.

1. **Difference in Approach between our Project and Projects of Reference:**

In the references we mentioned, they only employed one or two classifiers, and also they were limited to the theoretical part. In our project, we used three different classifiers, and also applied them practically. On the given data set, all three classifiers namely Multinomial Naive Bayes, Support Vector Machines, and K Nearest Neighbor are applied. Different classifiers need different vectorization methods, hence, we have implemented both the Bag-of-Words model and the Tfidf method. We cannot say that one classifier is an optimal one in all the scenarios. Different data sets require different kinds of classifiers to fetch high accuracy. Due to this very reason, we used three classifiers and evaluated them with the help of various performance evaluation methods like simple accuracy, confusion matrix, and classification report.

1. **Difference in Accuracy between our Project and Projects of Reference:**

In most of the references we used, they were mostly limited to the theoretical part. Even in the ones where they approached practically, they used a different data set than ours. In this case, it is hard to compare the accuracy of our models to the ones that are implemented in the references we used. The references we used implemented only a bag of words model and also they used only one of the above classifiers. We added another feature extraction method and checked which method worked best with which feature extraction model. Also, we compared all the accuracies scored by all the three classifiers with each of the feature extraction methods to determine which classifier is better. However, coming to the performance, we tried to incorporate methods from all the references and have done a combined approach where our performance is greatly improved.



Evaluation Results

1. **List of Contributions:**
2. Keen observation of data and carefully pre-processing it using suitable methods like tokenization, and removing stop words using NLTK library.
3. Implementation of TF-IDF.
4. Implementation of Bag-of-Words model.
5. Employing three different classifiers.
6. Evaluation of the three classifiers using various methods like confusion matrix, simple accuracy, and classification report

**Analysis**

1. **What did I do well:**

There is single classifier which suits all kinds of data. Hence, finding an optimal classifier based on the given data set is the most important part of sentiment analysis. We have chosen three different classifiers and evaluated them using various techniques to find out which classifier was an optimal one for the provided data set. By employing some of the techniques which we have learned from our references we were able to conclude that SVM fetched high score among all the three classifiers.

1. **What could I have done Better:**

Though we have implemented most of the techniques we have learned through various sources, we are not satisfied by our accuracy. We still need to employ more complex data pre-processing methods and feature extraction methods to improve our accuracy.

1. **What is Left for Future:**

Improving the accuracy of algorithms seems to be the most important thing. Most of the algorithms need large amounts of training data to classify the test data, and we need to develop complex algorithms which will reduce the amount of training data. Because in all cases large amounts of training data are not available. Also, the computing resources and time required to learn must be reduced which will parallelly improve the performance. Deep learning has also shown great promise in Natural Language Processing tasks. More research should be done in areas like Neural Networks on how to make better use of them to extract sentiment from texts.

**Conclusion**

After implementing all the three methods, we concluded that, on this particular data set SVM classifier performed very well with an accuracy of 87%. More complex data pre-processing methods, and feature extraction methods should be employed to further improve the accuracy of the model. Extensive research should be done in areas like Deep Learning, and Neural Networks to further improve the accuracy in the field of sentiment analysis.

**References**

[1]. Shabina Dhuria and Harmunish Taneja, “A Survey on Sentiment Analysis and Opinion Mining”, International Conference on Computing Sciences, pp. 124-128, Nov. 2013, Elsevier.

[2]. P. Rudy and Th. Mike, “Sentiment analysis: A combined approach” , International Journal of Informatics, 3, pp. 143–157, 2009.

[3]. Jagtap V. S. and Pawar Karishma, “Analysis of different approaches to Sentence-Level Sentiment Classification”, International Journal of Scientific Engineering and Technology, 2(3), pp. 164-170, April 2013.

[4]. Özgür Genç , “The basics of NLP and real time sentiment analysis with open source tools”, Towardsdatascience.

[5]. Mananmongia, “NLP Analysis of Restaurant Reviews”, GeeksforGeeks.