Final Project Report

On

Sentiment Analysis of Movie Reviews

ECE 569A - Artificial Intelligence

Instructor- Deepali Arora

Jonaki Medda (V00950300)
Karthiga Thangavelu (V00925048)
Mandeep Kaur (V00971821)

30 JULY, 2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Sentiment analysis is the technique of detecting if a user's expressed opinion is positive or negative. The process of opinion mining is stated to be the mechanism of sentiment analysis which then resembles the speaker's behavior. This is incredibly important in a situation where a recommendation is absolutely necessary. This work focuses sentiment analysis effort in this research on the IMDB movie review database. This project employs feature extraction and ranking to identify the polarity of the movie review and these features are used to train our multilabel classifier to classify the movie review into its correct label. This report intends to analyze the reviews of customers on various movies by implementing three algorithms and evaluating the score values of the Multinomial Naive Bayes, SVM, and Naive Bayes algorithms to illustrate the best text analysis algorithm by analyzing data based on movie reviews. Lastly, a comparison of all algorithms is being done to find the most efficient one and with the stream lit, we are able to create web apps by working on the interactive cycle of coding and it is easy to watch outcomes on the app whether good movie or bad movie.

Keywords: Sentiment Analysis, Machine Learning, Natural Language Processing, Support Vector Machine, Naive Bayes, Multinomial Naive Bayes.

1. INTRODUCTION

Movie reviews are a significant method to measure the performance of a film. Sentiment Analysis is a significant subject in machine learning which aims to extract subjective information from textual reviews. While providing a star rating to a movie informs us regarding the success or failure of a movie quantitatively, a deeper qualitative insight of a movie is achieved by a collection of movie reviews which can tell us if the movie, in general, meets the expectations of the reviewer. Using this, we can find the state of mind of the reviewer while providing the review and understand if the person was “happy”, “sad”, “angry” and so on. In this project, we are aiming to use sentiment analysis on a set of movie reviews given by reviewers and try to understand what their overall reaction to the movie will be, i.e., they like the movie or hate it [1].

1.1 BACKGROUND

Machine learning-based sentiment analysis and lexicon-based sentiment analysis are two types of sentiment analysis. Algorithms for supervised and unsupervised learning are at the heart of the machine learning-based approach. Class labeled training data is used in the supervised learning method.

There are different sources of data from the websites, blogs, and other places that can be held to provide a superior allowing of user opinion, such as the challenge at the intersection of NLP and data mining research. In order for a firm to grow, it is necessary to analyze all of this data. This is where sentimental analysis comes in [2]. Sentiment is a term used to describe the process of determining a person's level of contentment. Sentiment Analysis categories are:

- Knowledge-based Approach
- Applied Math Approach
- Hybrid Approach

Knowledge-based techniques are used to categorize language that is unambiguous, such as pleased, sad, fearful, or bored. Some context bases are unlikely to include any terms that are influenced by related to assign access to use the words that are likely to understand explicit emotions.

Latent linguistics analysis, support vector machines, "bags of words," and linguistics are examples of applied math approaches to machine learning. The grammatical relationships of words are employed to extract the speaker's opinion in context and locate the trait about which the speaker has remarked.

To notice linguistics that is expressed in an exceedingly refined manner, for example, through the analysis of ideas that don't expressly convey relevant data but are implicitly coupled to alternative ideas that do, a hybrid approach maximizes advantages on both machine learning and parts from information illustration like ontologies and linguistics networks to notice linguistics that is expressed in an exceedingly refined manner, e.g., through the analysis of ideas that don't expressly convey relevant data but are implicitly coupled to alternative ideas.

1.2 MOTIVATION

The motivation for pursuing the project is to provide some recommendations to the public to choose which movies to watch and for the movie industry to better judge their contents and know how their films are going to do in the box office. Since the opinions and feedback of the public is a major concern for the people making movies, it is extremely important for them to know how the public is reacting to their creations. There are several platforms available online where people can post their comments and reviews about a particular film that they have viewed. Several websites also provide ratings of those movies to help users for better search results. This encouraged us to do the project for a movie recommendation system using artificial intelligence. Also, our desire to explore the field of machine learning using natural language processing techniques gave us the idea to pursue this project [4][15].

2. RELATED WORK

The work presented in this project considers the previous works that have been performed in the problem domain of sentiment analysis of movie reviews.

In **2021, Vihaan Nama et al. [1]** aimed to conduct sentiment analysis of movie reviews by utilizing the Naive-Bayes algorithm and results compared to that of a Rule-Based Approach using the sentiment dictionary AFINN-111. **Soubraylu Sivakumar et al. [2]** proposed a novel architecture by combining word embedding with long short term memory (LSTM) to extract the semantic relationship between the neighboring words and to extract the key terms from the reviews, a weighted self-attention was applied. The experimental analysis done on the IMDB dataset and the proposed architecture word-embedding self-attention LSTM achieved an 88.67% F1 score, while LSTM and word embedding LSTMbased models resulted in an F1 score of 84.42% and 85.69%, respectively.

In **2020, A. Sheik Abdullah et al. [3]** focused on the analysis of review data by using data clustering and data classification process to determine the aspect based on sentiments using TF, IDF, and SVM. The classification algorithm used is SVM with a linear kernel process. The proposed scheme provided an improved accuracy of about 87.56% determining the positive and negative cases more efficiently for various sorts of recommendation systems which makes the user have good insight for a product review, movie review, and user rating analysis. **Rahul et al. [4]** showed that SVM has been used for classification techniques and accuracy measures as performance metrics. The dataset was yielded from reviews in the form of tweets from Twitter and movie reviews from other social networking sites' opinions.

In 2019, Mamtesh et al. [5] analyzed the reviews of customers on various movies by implementing K Nearest Neighbor, Logistic Regression, and Naive Bayes and concluded that Naïve Bayes be the ultra-efficient and appropriate algorithm in implementing sentiment analysis on movie reviews/brand reviews with 98.91% accuracy as compared to LR (99.34%) and KNN (98.69%) to understand the emotions and behavioral preferences of the consumer so as to provide better customer experience. **Lakshmi Surekha P. et al. [6]** analyzed the implementation of Feature Selection & Classification Algorithms for Movie review sentiment analysis using Machine Learning. **Mais Yasen et al. [7]** evaluated a model on a real-world dataset and compared it to different classifiers. The results revealed that Random Forest outperforms the other classifiers. **H. SWATHI et al. [8]** gained insight on market research analysis on social media and traditional media sources. The resulting analysis was helpful to movie viewers and the movie industry in understanding the perception of reviews. **Beatrice Lopez et al. [9]** implemented various classification models to predict the sentiment of IMDb reviews. k-fold cross-validation was used to evaluate the model by ensuring the consistency of their performance. This paper concluded that the best performing model was the Naive Bayes - Support Vector Machines classier with an accuracy score of 91.880%.

In **2018, V. Uma Ramya et al. [10]** analyzed tweets based on movie reviews using the Multinomial Logistic Regression, Naïve Bayes, and SVM algorithms by comparing precision, recall, and F-measure values to show the best text analysis algorithm. This paper concluded that Multinomial Naïve Bayes with Machine Learning algorithm produced improvised results as precision=1, recall=1, and F-measure=1 when compared to the other classifier algorithms for language processing like Multinomial Logistic Regression (precision0.90, recall=0.86, and F-measure=0.19) and Support Vector Machine (precision=1, recall=0.7, and F-measure=0.8). **Gurshobit Singh Brar et al. [11]** included feature-based opinion mining and supervised machine learning for sentiment analysis of movie reviews. In this paper, the polarity of reviews is determined using nouns, verbs, and adjectives as opinion words, and Reviews of Open Movie Database is used as a source data set and Natural Language Processing Toolkit for Part of Speech Tagging used. The proposed system is a web-based API for sentiment analysis for movie reviews with JSON output to display results on any operating system.

In 2017, Palak Baid et al. [12] analyzed the movie reviews by using Naïve Bayes, K-Nearest Neighbor, and Random Forest with 81.45% accuracy, 78.65% accuracy, and 55.30% accuracy achieved. So, the best results were given by the Naïve Bayes classifier.

In 2016, Tirath Prasad Sahu et al. [13] focussed on the IMDB movie review database and examined the sentiment expressions by using an approach based on structured N-grams. This paper concluded that the proposed approach to sentiment classification supplements the existing movie rating systems with 88.95%. accuracy. **Akshay Amolik et al. [14]** proposed a highly accurate model of sentiment analysis of movie tweets and employed feature vectors and classifiers such as Support vector machine and Naïve Bayes to classify these tweets as positive, negative and neutral to give the sentiment of each tweet.

3. PROBLEM FORMULATION

The project will involve building up a machine learning model by training it with a huge amount of data about the movie reviews collected from various sources of the web and building up an application to see if the model is working properly. The model will analyze the sentiments of the audience from their feedback, that is the dataset collected from the internet, and determine their reactions and feedback. After getting the public's reaction regarding different movies, we can apply that knowledge and use it to determine which ones are popular and are a must-watch and which ones are not [3]. Sentiment analysis is an important topic in NPL (Natural Language Processing). The IMDB Dataset having 50000 rows is considered for this project and in sentiment analysis, as a first step, the dataset is cleaned for better classification. Using preprocessing steps like stop words, removing punctuation, and tokenizing the words task of classification can be sophisticated. In the second step, the necessary feature is extracted as shown in figure 1. By using an appropriate algorithm for classification such as Naive Bayesian, SVM, and Multinomial Naive Bayesian the performance of the classifier can be accurate and performed well [17]. In this project sentiment analysis for movie review, extracting the feature from the words and classifying it into positive and negative based on which the user can select the movie to watch. Here the classifier has to perform well for the complex sentence.

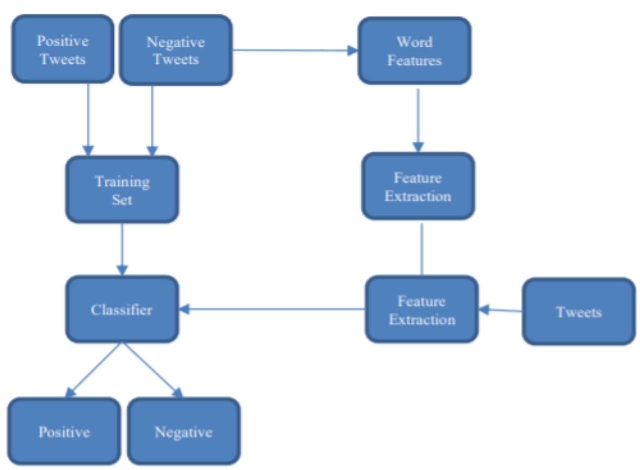


Figure 1. Sentiment Flow Chart

4. METHODOLOGY USED

4.1 PYTHON PACKAGES

Python is an application-oriented and problem-oriented programming language that is used by thousands of people for testing, games with the PyGame library, and system management. It supports imperative, object-oriented, functional, and procedural programming models, as well as a comprehensive standard library [10]. We have used four Python packages in this project are:

- ***NumPy*** is a Python package that provides multi-dimensional array reinforcement. It quickly interfaces with a broad range of databases, as well as with C/C++.
- ***Scikit-learn*** is a free Python machine learning toolkit that includes several classifications, regression, and clustering algorithms. It's compatible with NumPy and Spicy and can interchange and use data with them.
- ***Natural Language Toolkit (NLTK)*** is a library collection. It aids in the creation of representative token programs and the processing of common language using the Python programming language.

4.2 DATASET COLLECTION

The IMDB Dataset having 50000 rows is used and has 3 columns review, label either 0, 1 and 3rd column is sentiment words either positive or negative. For testing 30% dataset was used and 70% dataset was used for training. Moreover, to test the stream lit application we are using a 100 rows dataset collected manually which has 3 columns (reviews, movie name, picture link). The goal is to guess whether a review will be good or negative. To do so, an algorithm is trained with 70% dataset using the reviews and classifications in train data.csv and then used to generate predictions based on the reviews in test data.csv.

4.3 DATA PREPROCESSING

The pre-processing includes the compilation of the dataset before applying any algorithm to it. This is done in order to get rid of any undesirable words or symbols [7]. These words/symbols have no effect on the outcome, however, they can slow down the algorithm's processing. The following are the steps in our dataset pre-processing:

- ***Stopping***: It is a strategy for removing most recurring words using a stop-word list as a starting point in order to minimize the size of a text. Stop words include a, an, the, this, too, for, and others.

- ***Stemming:*** It is the practice of eliminating morphological ends from English words on a regular or frequent basis. It converts words into root words by stemming them. For example, the words hot, hotter, and hottest are all derived from the root word ‘hot.' Furthermore, terms with a frequency of more than 80% of the sample are excluded since they are probable stop-words. Similarly, words with extremely low frequency should be disregarded.
- ***Removing html tags***: It cleans the code and leaves only plain data.
- ***Lemmatization***: It refers to doing things correctly using a vocabulary and morphological study of words, with the goal of removing only inflectional ends and returning the base or dictionary form of a word, known as the lemma.
- ***Text tokenization:*** It is the process of breaking down the text into sentences and phrases using the basic linguistic units of words, numbers, and punctuation [23]. Words in the English language are Usually, white spaces are used to separate them. The division of a string into sentences is known as sentence tokenization. Punctuation, particularly the full stop letter, in English indicates the conclusion of a statement [24]. The full stop, on the other hand, abbreviations can be used with a character, which does not have to be capitalized. A table of abbreviations is employed to avoid this problem. Sentence tokenization was performed using NLTK [20], which has been trained on a large number of sentences. The training comprises the following:
- ***Punctuation***: Punctuation and characters that appear in the text are identified.

4.4 TRAIN TEST SPLITTING

Training data (X train, y train) and test data (X test, y test) make up the entirety of the dataset. After learning from training data, test data will be utilized in the process of calculating the effectiveness of various classifiers. To split up the original dataset, a 70:30 ratio was used. The following lists are generated as a result of this step:

X_train: training features/review

y _train: sentiments/output training (1 for positive, 0 for negative)

X _test: review/features test

y _test: evaluate sentiments and output (1 for positive, 0 for negative)

4.5 FEATURE EXTRACTION

It is an example of a method that uses vectors to represent text. The following are the most common types of word embeddings:

- ***BoW (Bag of Words)***

  It is the most basic type of numerical text representation. A phrase can be represented as a bag of words vector, much like the term itself (a string of numbers). The work started by creating a vocabulary out of all of the unique words from the reviews. Then each of these words is taken and their occurrence in the movie reviews is marked with 1s and 0s. This will give us vectors for reviews [10]. Hence The Bag of Words model simply generates a collection of vectors representing the count of word occurrences in the text (reviews), and BOW vectors are simple to understand.

- ***TFIDF (Term Frequency-Inverse Document Frequency)***

  TF-IDF (Term Frequency Inverse Documentation Frequency) technique uses numerical statistics to indicate the value of a certain term in a text [10]. It demonstrates how the document's words are reflected. Data mining with a weighted approach to IF (Information Retrieval) in the domain of digital libraries, the most recommended systems. Web indexes commonly employ the TF-IDF weighted-plan collection as a focus in evaluating and positioning an archive's relevance in response to a customer inquiry [25]. By dividing stop words into different topics, such as content rundown and characterization, TF-IDF may be utilized effectively for stopwords.

  Term Frequency (TF) is a metric that evaluates how often a specific term appears in a report. Because each report is different in length, it's understandable that a word might appear substantially larger in longer reports than in shorter ones. As a result, the term recurrence is separated from the length of the report (also known as the total number of words in the record) on a regular basis as a method of standardization.

  $TF(t) = (Number\ of\ times\ a\ term\ appears\ in\ a\ report)\ /\ (Total\ number\ of\ terms\ in\ the\ archive)$

  IDF stands for Inverse Document Frequency depicts the imperative of a sentence. When registering TF, it is important to think about all of the words in the same way. In any event, it is recognized that some words, such as "is," "of," and "that," may repeat a little importance. As a result, we must overburden the frequent words while scaling up the unusual ones by recording the following:

$$IDF(t) = log\_e(Total\ number\ of\ archives)\ /\ Number\ of\ reports\ containing\ the\ term\ t)$$

Hence, the TF-IDF model includes information on both the more significant and less important words. TF-IDF outperforms in machine learning models [10]. The implementation of the feature extraction is shown in figure 2.

```
# # bag of words
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(train_data)
X_test_counts = count_vect.transform(test_data)
#
# #tfidf
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
#
# tfidf_transformer_test = TfidfTransformer()
X_test_tfidf = tfidf_transformer.fit_transform(X_test_counts)
#
# # scale standard
sc = StandardScaler(with_mean=False)
sc.fit(X_train_tfidf)
X_train_std = sc.transform(X_train_tfidf)
X_test_std = sc.transform(X_test_tfidf)
```

Figure 2: Feature Extraction Implementation

- ***Standard Scaling***

  In order to compute the tf-idf representation, we computed the inverse document frequencies based on the training data and used these statistics to scale both the training and test data. In scikit-learn, fitting the feature transformer on the training data amounts to collecting the relevant statistics. The fitted transformer can then be applied to the test data.

4.6 CLASSIFICATION USED

To evaluate the proposed model, three classifiers run on the same training and testing datasets [5]. The classifiers could be summarized as mentioned below:

- ***Naive Bayes***: This classifier contains two probabilities: P(class), which is the likelihood that input will create a certain class, and the probability of an input condition (input condition|class) is P(input condition|class). Given the class, each feature has a specific value. Otherwise, the probability is 0 by default. For text-based classification, the Naive Bayes technique is commonly employed. It's a supervised learning approach for classification problems [21], and it's popular when working with large training datasets. For example, Naive Bayes is commonly used in Spam Filtration, Sentimental Analysis, and

Article Classification. Machine learning allows you to quickly create models and make predictions. Machine learning allows you to quickly create models and make predictions. A text classifier can produce good results when given a well-trained dataset.

- ***Support Vector Classifier (SVM)***: A classifier that normalizes nominal features to binary features and handles with missing values. It formalizes all characteristics by default. The output coefficients are calculated with the help of the features in a normalized form. The Support Vector Machine (SVM) is often regarded as the most accurate classifier for voice classification issues [22]. They did it by constructing a hyperplane with the shortest Euclidean distance between the closest trained instances. The hyperplane of the Support Vector Machine is entirely resolved by a small subset of the training data sets that are considered as support vectors. The certified classifier is not available to the remaining training data sets. As a result, the classifier SVMs have been utilized in the majority of disciplines for text categorization.
- ***Multinomial Naive Bayes:*** The multinomial Naive Bayes classifier is good for discrete feature classification (e.g., word counts for text classification). Normally, integer feature counts are required for the multinomial distribution. Fractional counts, such as TF-IDF, may also function in practice. The multinomial Naive Bayes classification technique is commonly used as a starting point for sentiment analysis [5]. The main idea behind the Naive Bayes approach is to use the joint probabilities of words and classes to discover the probability of classes given to texts.

4.7 EVALUATION METRICS

Precision, recall, F-measure, and Accuracy are the four prominent assessment metrics that have been used to assess the effectiveness of the method employed in the study [13]. Before getting into evaluation metrics, and understanding of the confusion matrix is required. The confusion matrix reveals not just a predictive model's performance, but also which classes are properly predicted, and what types of errors are being produced. A confusion matrix is a table that shows how well a classification model performs on a set of test data for which the real values are known.

- ***Precision***: Precision is the percentage of recovered documents that are relevant to the query in the field of information retrieval. The number of positive class predictions that really belong to the positive class is measured by precision [13]:

  $Precision = |(Relevant\ Documents \cap Retrieved\ Documents)\ /\ Retrieved\ Documents|$

- ***Recall:*** The percentage of documents that are relevant to the query that is successfully retrieved is known as recall in information retrieval. The number of positive class predictions made out of all positive examples in the dataset is measured by recall [13]:

  $Recall = |(Relevant\ Documents \cap Retrieved\ Documents)\ /\ Relevant\ Documents|$

- ***F-measure***: The F1 score (also F-score or F-measure) is a measure of a test's accuracy in binary classification statistical analysis [13]. F-Measure generates a single score that accounts for both accuracy and recall issues in a single number:

  $F\ measure = 2 * [(Precision * Recall)\ /\ (Precision + recall)]$

- ***Accuracy:*** The number of real findings (both true positives and true negatives) among the entire number of instances investigated, i.e. true positives, true negatives, false positives, and false negatives., is the accuracy [13]:

  $Accuracy = (TP + TN)\ /\ (TP + TN + FP + FN)$

- ***Confusion Matrix:*** A confusion matrix is a table that shows how well a classification model performs on a set of test data for which the real values are known.

4.8 STREAMLIT

It is a Python framework for creating Machine Learning and Data Science web apps that is open-source. Working on the interactive cycle of coding and watching outcomes in the web app is made simple using Streamlit. In this project, using Streamlit an app has been designed which is easy to use by the public to find the must-watch and average movies without surfing on the browser every time. The required library is installed using *pip install streamlit, scikit-learn.*

5. RESULTS & DISCUSSIONS

Since the dataset has 50000 rows and three columns namely review, label (either 0,1), and sentiment words (either positive or negative). The column represents the emotions in accordance with the labeled classifications The information was gathered from movie reviews, which indicate whether the opinions about the film were good or negative. The sentiment analysis technique identifies stop words and noun phrases. As a consequence, a word cloud is created. With the observed opinion set, the most similar opinion pairs are evaluated and recorded. Three classifiers are implemented in this project report and their accuracies are compared. Firstly, Naive Bayes implemented which gives only 45.6% accuracy which is not acceptable as shown in figure 3.

```
Accuracy is: 0.45614035087719296
The moview was genuinely awesome Positive
```

Figure 3: Naive Bayes Accuracy

Secondly, the SVM classifier performed better with 84.3% accuracy as compared to multinomial naive Bayes and naive Bayes as shown in figure 4.

```
0.8014
Accuracy score 0.801
Precision: 0.7942715145173947
Recall: 0.8120069528011766
F1 score: 0.8013862073755122
[[5948 1573]
 [1406 6073]]
```

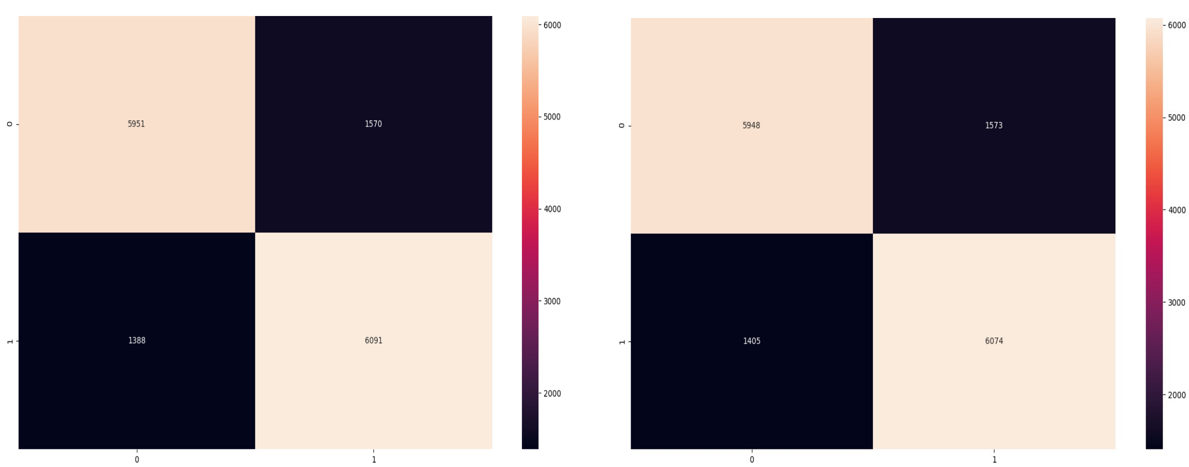
Figure 4: SVM classifier Score Value

Lastly, Multinomial Naive Bayes is implemented and gives 86.4 % accuracy. Also, evaluation metrics are also being evaluated as shown in figure 5.

```
0.8642666666666666
Accuracy score 0.864
Precision: 0.8729614910237083
Recall: 0.8517181441369167
F1 score: 0.8642363907784164
[[6594  927]
 [1109 6370]]
```

Figure 5: Multinomial Naive Bayes Score Value

To elaborate on SVM and Multinomial Naive Bayes performance, a heat map is used as a two-dimensional data visualization method that displays the magnitude of phenomena as color. The data matrix's columns/rows are re-ordered based on the hierarchical clustering result, grouping comparable observations together. In the data matrix, the blocks of ‘high' and ‘low' values are contiguous. Opinion pairs that are seen are evaluated and recorded for both SVM and MNB as shown in figure 6.



a) b)

Figure 6: Heat Map of SVM (a) and Multinomial Naive Bayes (b)

After identifying stop words and noun phrases, a word cloud is created which is a visual representation of positive and negative words. Popular words and phrases are highlighted using cloud creators depending on their frequency and significance. It gives fast and easy visual insights that can lead to more detailed investigations. Figure 7 shows the positive word cloud for SVM and MNB.

a) b)

Figure 7: Positive Word Cloud od SVM (a) and Multinomial Naive Bayes (b)

Besides this, we successfully implemented a Streamlit where one can easily search for must-watch movies as good movies or average movies as bad movies before watching. Here are some Streamlit screenshots of our project as shown in figure 8 and figure 9.

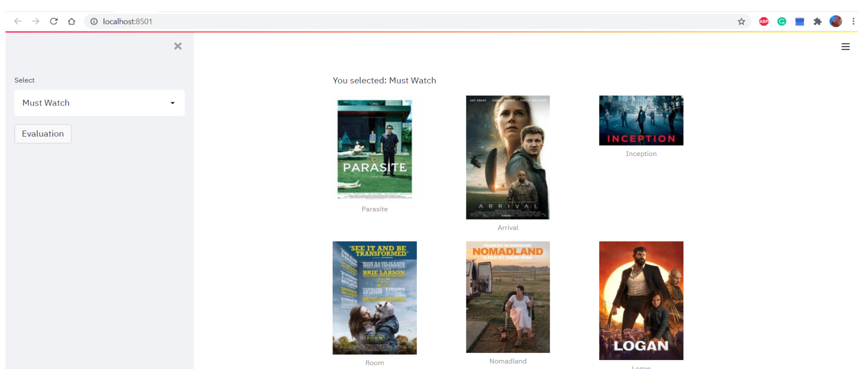


Figure 8: Must Watch Movies on Streamlit

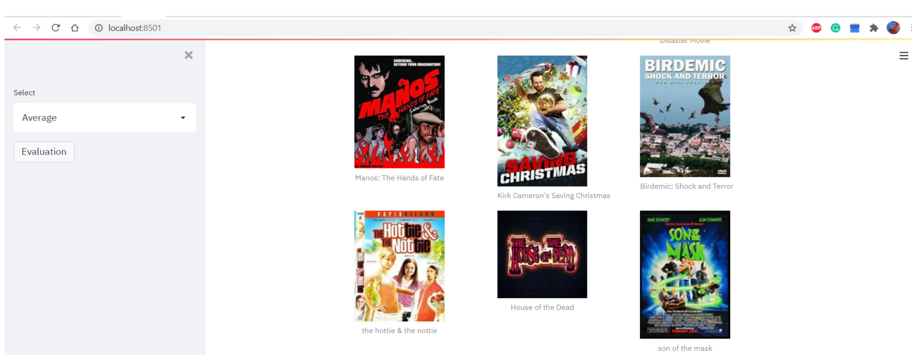


Figure 9: Average Movies on Streamlit

5.1 COMPARISON BETWEEN MODELS

Initially, when three algorithms are implemented, their accuracies are compared and concluded that Multinomial Naive Bayes with 86.4% accuracy and Support Vector Machine with 80.14% accuracy which are better than Naive Bayes and as shown in figure 10.

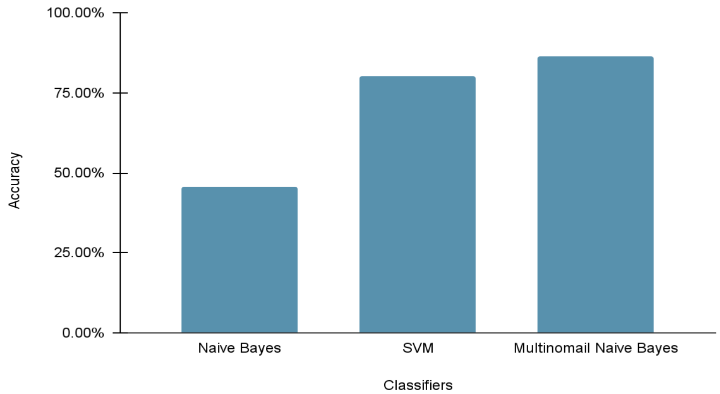


Figure 10: Comparison between three classifiers

Based upon accuracy comparison, we focussed on Multinomial Naive Bayes and Support Vector Machine, then precision, recall, and F-measure for both positive and negative are calculated for these two algorithms as a confusion matrix as shown in table 1.

Table 1: Confusion Matrix Report

Algorithm	Precision	Recall	F1-score
Multinomial Naive Bayes	0.87	0.85	0.86
SVM	0.79	0.81	0.80

From the above results, we can conclude that the Multinomial Naive Bayes algorithm performs better for text analysis or sentiment analysis when compared to SVM.

5.2 CHALLENGES FACED

The major challenge is with the dataset that we took from IMDB because the dataset was not good enough for our project. While running streamlit, we were not able to get the maximum movie names along with reviews. As a result, only 100 movies from another IMDB dataset were manually collected along with movie names, reviews, and poster links. Another major impact of this dataset was on the Naive Bayes algorithm's performance because poor accuracy achieved in Naive Bayes due to preprocessing of the dataset was not adequate, otherwise, it performed well in the research papers that we reviewed. Moreover, the algorithm unfortunately could not perform well against unseen data due to overfitting and results in showing some positive movies in the bad movies so we need to change the test data size and employ cross validation to avoid this.

6. CONCLUSION AND FUTURE WORK

In many disciplines, sentiment analysis has become one of the most significant and developing topics. In all concerns, the domain of identifying positive and negative opinions creates judgment. This project looks at how classification may be used to analyze movie review data. The SVM classification algorithm separates positive and negative views for movie review data with an accuracy rate of 84.3% and Multinomial Naive Bayes gives an accuracy of 86.4%. Therefore, it is concluded that Multinomial Naive Bayes with Machine Learning algorithm produces an improvised result when compared to the other classifier algorithms like SVM and Naive Bayes. As a result, this may be used to categorize more social media data in the future. The method of gathering real-time tweets for categorizing a socially important action and determining good and negative tweets will be the focus of future studies.

References

[1] Vihaan Nama, Vinay Hegde, B. Satish Babu, "Sentiment Analysis of Movie Reviews: A Comparative Study between the Naive-Bayes Classifier and a Rule-based Approach", ICITIIT, IEEE, 2021.

[2] B Pang, L Lee- "Opinion mining and sentiment analysis" Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008) 1–135

[3] A. Sheik Abdullah, K. Akash, J. ShaminThres, S. Selvakumar, "Sentiment Analysis of Movie Reviews Using Support Vector Machine Classifier with Linear Kernel Function", Evolution in Computational Intelligence, Springer, pp 345-354, 2020.

[4] Rahul, Vasundhara Raj, "Online Reviews Over Sentiment Analysis using Machine Learning: A Systematic Review" JOURNAL OF CRITICAL REVIEWS, ISSN- 2394-5125 VOL 7, ISSUE 05, 2020

[5] Mamtesh, Seema Mehla, "Sentiment Analysis of Movie Reviews using Machine Learning Classifiers", International Journal of Computer Applications (0975 – 8887), Volume 182 – No. 50, April 2019.

[6] Lakshmi Surekha P, Jayanthi A, "A Movie Review Sentiment Analysis Using Machine Learning Techniques", International Journal of Recent Scientific Research, Vol. 10, Issue 8, pp. 34492-34497, 2019

[7] Mais Yasen, Sara Tedmori, "Movies Reviews Sentiment Analysis and Classification" IEEE, 2019.

[8] H. Swathi, S. S. Aravinth, V. Nivethitha, T. Saranya, R. Nivethanandhini, "Sentiment Analysis of Movie Review using data Analytics Techniques", IRE Journals | Volume 2 Issue 9 | ISSN: 2456-8880, 2019.

[9] Beatrice Lopez, Minh Anh Nguyen, Xavier Sumba, "IMDb Sentiment Analysis", Mcgill Univ, CA, February 2019.

[10] V.Uma Ramya, K. Thirupathi Rao, "Sentiment Analysis of Movie Review using Machine Learning Techniques", International Journal of Engineering & Technology, Volume 7, pp 676-681, 2018.

[11] Gurshabad Singh Brar, Ankit Sharma, "Sentiment Analysis of Movie Review Using Supervised Machine Learning Techniques", International Journal of Applied Engineering Research, ISSN 0973-4562, Volume 13, Number 16, pp. 12788-12791, 2018.

[12] Palak Baid, Apoorva Gupta, Neelam Chaplot, "Sentiment Analysis of Movie Reviews using Machine Learning Techniques" International Journal of Computer Applications (0975 – 8887), Volume 179 – No.7, December 2017.

[13] Tirath Prasad Sahu, Sanjeev Ahuja, "Sentiment Analysis of Movie Reviews: A study on Feature Selection & Classification Algorithms" IEEE, 2016.

[14] Akshay Amolik, Niketan Jivane, Mahavir Bhandari, Dr.M.Venkatesan, "Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques", International Journal of Engineering and Technology, Vol. 7, No.6, 2016.

[15] Bhumika Gupta, Monika Negi, Kanika Vishwakarma, Goldi Rawat, Priyanka Badhani, "Study of Twitter Sentiment Analysis using Machine Learning Algorithms on Python", International Journal of Computer Applications, Vol.165, 2017.

[16] Mitali Desai, Mayuri A. Mehta, "Techniques for sentiment analysis of Twitter data: A comprehensive survey", 2016 International Conference on Computing Communication and Automation (ICCCA), pp. 149-154,2016.

[17] Mayuri R.Lahamage, "Analysis and Visualization of Total Movie Review System Using Sentiment Analysis" International Journal of Computer Science and Information Technology Research, Vol. 2, Issue 4, pp: (27-33), December 2014.

[18] Ligthart, A., Catal, C. & Tekinerdogan, B. Systematic reviews in sentiment analysis: a tertiary study. Artif Intell Rev (2021). https://doi.org/10.1007/s10462-021-09973-3.

[19] Naresh, A., Venkata Krishna, P. "An efficient approach for sentiment analysis using a machine learning algorithm". Evol. Intel. (2020).

[20] Soubraylu Sivakumar, Ratnavel Rajalakshmi, "Analysis of Sentiment on Movie Reviews Using Word Embedding Self-Attentive LSTM", International Journal of Ambient Computing and Intelligence, Vol 12(2) pp. 33-52, 2021.

[21] Bingwei Liu∗, Erik Blasch†, Yu Chen‡, Dan Shen∗ and Genshe Chen, "Scalable Sentiment Classification for Big Data Analysis Using Na¨ıve Bayes Classifier" Big Data, IEEE International Conference,2103 Electronic ISBN: 978-1-4799-1293-3, December 2013

[22] Shweta Rana, Archana Singh "Comparative analysis of sentiment orientation using SVM and Naïve Bayes techniques", Next Generation Computing Technologies(NGCT), Electronic ISBN: 978-1-5090-3257-0, 2016.

[23] Vinodhini, Chandrasekaran (2012), "Sentiment Analysis and Opinion Mining: A Survey", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, PP 1-11.

[24] Jeffrey Reynar, (1998), "Topic Segmentation: Algorithms and Applications", IRCS Technical Reports Series, Vol. 66, PP 1- 189.

[25] Ramadhan WP, strip Novianty S.T, Casi Setianing S.T., M.T "Sentiment Analysis Using Multinomial Logistic Regression", International Conference on Control, Electronics, Renewable Energy and communication(ICCEREC), 2017