Coal City University Journal of Science

ISSN: 2734-3758(Print), 2734-3766 (Online)

A publication of faculty of Natural and Applied Sciences, Coal City University, Enugu Nigeria.

CCU Journal of Science
Vol. 3, Issue 1, July, 2023

ISSN: 2734-3758(Print), 2734-3766 (Online)
ttps://ccujos.com

A MACHINE LEARNING CLASSIFICATION MODEL FOR MOVIE REVIEWS USING N-GRAM FEATURES SELECTION

[1]Ademola Abiodun Omilabu.,[2]Adedeji Oladimeji Adebare.,[3]Olayinka Olufunmilayo Olusanya.,[4]Omotayo Joseph Adeyemi
[1,2,3,4]Department of Computer Science, Tai Solarin University of Education, Ijagun, Ogun State, Nigeria.
omilabuaa@tasued.edu.ng, adebaredj06@gmail.com, olusanya_oo@tasued.edu.ng, tayo_009@hotmail.com.

Corresponding Author: Adebare, A.O. (adebaredj06@gmail.com**)**

Abstract

This study addresses the escalating reliance of consumers on online platforms for informed decision-making, particularly in the realm of movie reviews. Leveraging a dataset of 50,000 movie reviews from Kaggle's IMDb dataset, the research focuses on developing a sentiment classification model by amalgamating N-Gram features extraction and diverse machine learning algorithms. The exploration emerges from identified limitations in prior methodologies, including modest dataset sizes, rule-based approaches, and a predominant reliance on TF-IDF feature extraction. Employing Python for Exploratory Data Analysis (EDA), the study encompasses essential preprocessing tasks such as stemming, lowercase conversion, stop-word removal, and tokenization. N-Gram feature selection takes center stage, aimed at encapsulating the nuanced contextual relationships between words. The suite of machine learning algorithms engaged Linear Support Classifier, Logistic Regression, Decision Trees, Bernoulli Naïve Bayes, and Multinomial Naïve Bayes. Comparative analyses reveal that the N-Gram methodology, notably in conjunction with the Linear Support Vector Classifier, outshines the traditional TF-IDF approach, displaying superior accuracy and yielding more balanced confusion matrices. While Multinomial Naive Bayes and Logistic Regression exhibit effectiveness, Decision Trees present limitations in precision. In conclusion, the research underscores the supremacy of the N-Gram approach, particularly in tandem with the Linear Support Vector Classifier, as a robust framework for sentiment analysis in movie reviews. These insights hold practical implications for businesses relying on user-generated content for informed decision-making, emphasizing the necessity of incorporating N-Gram feature extraction in sentiment analysis models.

Keywords: *Sentiment Analysis, N-Gram Features, Machine Learning, Movie Reviews, IMDb Dataset, Exploratory Data Analysis.*

1. Introduction

With the proliferation of online review sites, micro-blogging services, social networks, and discussion forums, customers have become increasingly reliant on online resources to assist them in their purchasing decisions. These review sites allow existing customers to provide impartial feedback about products and services they have used, enabling potential consumers to make more informed purchasing choices. Recent surveys indicate that a vast majority of consumers rely heavily on online reviews and user-generated content to aid their purchasing decisions. Specifically, 95% of shoppers consult online reviews before making a purchase (Spiegel Research Centre, 2017), while 97% consider such reviews a significant information source for purchasing decisions (Fan and Fuel, 2016). Furthermore, a substantial 73% of online adults actively engage with social networks like Facebook, LinkedIn, and Google Plus (M. Duggan and A. Smith, 2013). Across various online platforms, including social media, e-commerce sites, and forums, a wealth of user opinions, thoughts, and sentiments exist. Consequently, there is a pressing need to automate the process of text sentiment analysis. Sentiment analysis has proven beneficial for numerous natural language processing (NLP) tasks, such as question-answering systems and information extraction (Pang & Lee, 2008). The primary objective of sentiment analysis is to determine whether a given text conveys a positive, negative, or neutral sentiment. It has diverse applications, including social media monitoring, customer feedback analysis, brand reputation management, and market research. By analyzing customer opinions, preferences, and behaviors through sentiment analysis, businesses can gain valuable insights to enhance their products, services, and marketing strategies.

Machine learning, a subset of artificial intelligence, enables computers to acquire knowledge from historical datasets, and it has been widely used in the past few years in analyzing and extracting the polarity of sentiments from reviews made available online on different domains. Lu and Wu (2019) developed support vector machine classification for sentiment analysis of film reviews using a sentiment dictionary. The SVM showed higher accuracy of sentiment accuracy than the basic sentiment dictionary. However, the research was limited to a few-word corpus sentiment dictionary and the use of Bag of Word (BoW) which failed to capture semantic meaning among words. Mohsin Ahmed and RabeeaJaber (2020) established and applied four machine learning models, namely Naïve Bayes, K-Nearest Neighbor (KNN), J48, and Logistic Regression on Movies Review. However, the dataset used was limited in size, no detailed descriptive analysis of the data was applied, and TF-IDF used as features extraction failed to generate context among words. This research will focus on the application of machine learning models on sentiment analysis of the (IMDb) Internet Movie Database dataset's movie reviews using N-Gram features extraction. The objective of this research endeavor is to formulate a classification model for sentiment analysis of movie reviews based on relevant information using a combination of N-Gram features extraction and Machine Learning Algorithms.The subsequent sections of this paper are organized as follows: Section 2 presents a review of relevant prior work about the current study. Section 3 delineates the materials and methodological approaches employed in this research. Section 4 showcases the

obtained results and provides a discussion of their implications. Finally, Section 5 concludes the study by summarizing the key findings and contributions.

2. LITERATURE REVIEW

Lu &Wu (2019) presented a method that utilizes sentiment dictionaries and SVM classification technology for sentiment analysis of film review texts. They constructed four sentiment dictionaries from various sources of sentiment words and employed SVM classification to categorize the features as either positive or negative sentiment. However, their study had limitations as it focused solely on film reviews in the Chinese language. Furthermore, the proposed method solely relied on sentiment dictionaries for feature extraction, which may not capture the full range of sentiment expressed in the text. In a separate study, Mohsin Ahmed and RabeeaJaber (2020) investigated the application of machine learning techniques for sentiment analysis, specifically in the context of movie reviews. They employed four different machine learning techniques (naïve Bayes, KNN, j48, and logistic regression) to classify the sentiment of each review as either positive or negative. TF-IDF was used for feature extraction to identify relevant features in the sentiments. However, the TF-IDF feature extraction method employed did not retain contextual information.

Cahyanti et al. (2020) conducted a study focusing on sentiment analysis of movie reviews. They employed Support Vector Machine (SVM) and feature extraction classification to develop a method. The proposed approach was juxtaposed with other machine learning techniques, including Naïve Bayes (NB), Random Forest (RF), and K-nearest Neighbor (KNN) in terms of F1 score, recall, precision, and accuracy. The findings revealed that the proposed method achieved an accuracy rate of 87.5%, surpassing the accuracy rates of KNN (80%), RF (82.5%), and NB (85%). The researchers also experimented with Information Gain selection features but discovered that using a higher threshold value could hinder SVM's ability to build a classification model. The study was limited to 2000 movie reviews from IMDB, and the selected features did not establish contextual relationships among the text. Additionally, the performance of K-nearest neighbor (KNN) with Information Gain feature selection in sentiment analysis was compared to other machine learning methods such asRandom Forest,Support Vector Machine, and Naïve Bayes, using the Polarity v2.0 dataset from the Cornell movie review dataset. The results indicated that KNN with information gain feature selection achieved the highest performance, with an accuracy of 96.8% compared to the other proposed methods. However, the study only employed geometric-type feature selection, which failed to capture the semantic meaning of the text (Daeli&Adiwijaya, 2020; Zhao et al, 2022, Edeh et al, 2020).

Khan et al. (2020) researched summarizing online movie reviews using machine learning techniques. They employed Support Vector Machines and Naïve Bayes for the classification and ranking of review sentences. The authors evaluated the effectiveness of their approach by comparing it to benchmark summarization methods. However, their research was limited to extractive summarization of online movie reviews and relied on the bag-of-words (BoW) technique for feature extraction, which inaccurately captured the semantic meaning of words

and phrases. Maulana et al. (2020) Investigated to enhance the accuracy of sentiment analysis in movie reviews using information gained with support vector machines. The experimental results demonstrated that their proposed method, which utilized a support vector machine based on information gain, achieved higher accuracy in classifying movie reviews as positive or negative compared to other existing methods. Specifically, their proposed method attained an accuracy of 86.62%, which was 0.166% higher than the accuracy obtained using a standard support vector machine algorithm. However, the study was limited to a small dataset, and the information gain feature extraction exhibited better performance with larger datasets.

Mitra (2020) conducted a study exploring the application of Natural Language Processing (NLP) in extracting emotional states from text data, specifically focusing on movie review datasets. The study employed Naïve Bayes, Support Vector Machine (SVM), and Maximum Entropy models with a lexicon-based approach for sentiment analysis. However, the specific feature extraction method used in the study was not mentioned. In recent research, movie reviews were analyzed using the Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction, along with optimized machine learning techniques. The study employed four methodologies in machine learning: support vector machines (SVM), random forest (RF),decision trees (DT), and gradient boosting classifier (GBC) for sentiment analysis. Additionally, a deep learning model was utilized to compare its performance with conventional machine learning algorithms. The efficacy of four feature extraction methodologies—TF-IDF, Bag of Words (BoW), Global Vectors (GloVe) for word embeddings, and Word2Vec—was assessed for sentiment analysis. The findings of the study, demonstrated that the proposed methodology, which involved classifying movie reviews using TF-IDF and optimized machine learning algorithms, achieved a high level of accuracy in categorizing movie reviews into positive and negative sentiments. However, the study was limited by the use of TF-IDF for feature extraction, as it failed to capture contextual information among words (Naeem et al., 2022, Edeh et al, 2021).

Based on the aforementioned related work summary, it can be concluded that most sentiment analyses of movie reviews utilize vectorized datasets generated from TF-IDF. However, TF-IDF does not retain contextual information or establish connections between words. Therefore, this research aims to address this limitation.

3. MATERIALS& METHOD

This section delineates the materials and methodologies employed to develop the sentiment analysis classification model. It encompasses the data collection process, natural language processing techniques applied to the dataset, feature extraction methods utilized, machine learning algorithms employed, simulation techniques, and performance evaluation metrics.

3.1 Data identification and collection

The dataset utilized in this study was sourced from the Kaggle online repository, accessible at the following URL:https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-

movie-reviews. The dataset encompassed reviews about various movies from the Internet Movie Database (IMDb), constituting a binary sentiment classification task with each review labeled as either positive or negative. The dataset, acquired from the source, was stored in a comma-separated variable (.csv) file format. Subsequently, to facilitate seamless access to the simulation environment employed for conducting numerous analyses, the dataset was uploaded to Google Drive.

3.2 Method of Data Preprocessing

The dataset gathered for this investigation was structured, yet the movie reviews it contained were unstructured, and presented in natural language. To rectify this, natural language processing techniques were employed to transform the unstructured dataset into a structured format. Python natural language tool toolkit (NLTK) was applied for converting text to lowercase, stop word removal, stemming, and tokenizing the reviews into individual words

3.3 N-Gram Feature Selection

Feature selection is a crucial step in developing an effective sentiment classification model for movie reviews. In this study, N-Gram feature extraction is employed to capture the contextual information and relationship between words in the movie reviews. N-gram models create a representation of the text by considering sequences of N words as features. The selection of the appropriate N-Gram size is determined through experimentation and evaluation of the model's performance. The N-gram feature selection process typically involves using statistical measures to evaluate the relevance and importance of N-gram features for a particular task, such as sentiment analysis. While there isn't a specific mathematical formula that universally applies to N-Gram feature selection, several common approaches utilize statistical measures like chi-square, mutual information, or feature importance scores.

3.4 Machine learning algorithms adopted

Once the N-Gram features are selected, a classification model is formulated using machine learning algorithms such as Multinomial Naïve Bayes algorithms, Bernoulli Naïve Bayes algorithms,Logistic Regression algorithms, Linear Support Vector Classifier (SVC), and Ensemble machine learning algorithms such as Decision Tree algorithms are considered for this purpose. For each algorithm considered in this study, the requisite libraries were imported into the Python environment utilized for developing the sentiment classification model. Subsequently, classification models were constructed employing these algorithms, utilizing the datasets and incorporating N-gram feature selection. The specific algorithms adopted in this study are as follows:

a. Multinomial Naïve Bayes algorithms

The Multinomial Naïve Bayes algorithm is a specialized variant of the Naïve Bayes classifier, tailored to handle discrete features with multiple categories. It is commonly used for text classification tasks, such as sentiment analysis, spam detection, and document categorization. The scikit-learn library in Python provides an implementation of the Multinomial Naïve

Bayes algorithm ("sklearn.naive_bayes.MultinomialNB"). You can use it by importing the class, fitting it on your training data, and using it for predictions on new data.

b. Bernoulli Naïve Bayes algorithms

The Bernoulli Naïve Bayes algorithm is a specific adaptation of the Naïve Bayes classifier engineered to handle binary features. It is commonly used for text classification tasks where the presence or absence of words or features is considered. The scikit-learn library in Python provides an implementation of the Bernoulli Naïve Bayes algorithm ("sklearn.naive_bayes.BernoulliNB"). You can use it by importing the class, fitting it on your training data, and using it for predictions on new data.

c. Logistic Regression algorithms

Logistic Regression, widely utilized for binary classification tasks, establishes a correlation between a group of independent variables and a binary dependent variable. It calculates the likelihood of the dependent variable belonging to a specific class. The scikit-learn library in Python provides an implementation of logistic regression ("sklearn.linear_model.LogisticRegression"). You can use it by importing the class, fitting it on your training data, and using it for predictions on new data.

d. Linear Support Vector Classifier

Linear Support Vector Classifier, also known as Linear SVC, is a classification algorithm that uses support vector machines (SVM) to classify data into different classes. It is particularly effective for binary classification problems.

e. Decision Trees Algorithms

Decision Trees represent versatile and extensively employed supervised machine learning methodologies suitable for classification and regression assignments. They construct a decision-based model by dividing the feature space according to input features, forming a tree-like structure. In this structure, internal nodes denote features or attributes, branches represent decision rules, and leaf nodes signify outcomes or class labels.

3.5 Method of Model Simulation

The simulation of the predictive model adopted in this study was conducted using the Python programming language within the Google Colaboratory (CoLab), a Python JupyterNotebook environment created by Google for Gmail users. The requisite libraries for performing the analysis were imported into the notebook, including pandas for data manipulation and storage in data frames, NumPy for array manipulation to feed the machine learning algorithms, seaborn and matplotlib for data visualization, and the natural language toolkit (NLTK) for natural language processing (NLP) tasks on the review data necessary for sentiment analysis.
Additionally, the scikit-learn (sklearn) packages were imported to provide libraries for NLP tasks, machine learning algorithms, and model evaluation via the model selection and metrics modules, respectively. The dataset was imported into Google Drive and divided into training and testing subsets, with 70% allocated for model training and 30% for evaluating the

predictive models' performance. The validation of the predictive models using the testing dataset was carried out employing various performance evaluation metrics.

3.6 Method of Model Evaluation

To evaluate the performance of the simulated predictive models for sentiment classification of reviews, the sklearn.metrics package was employed, comprising the confusion matrix and classification_report libraries. The confusion matrix library was utilized to interpret the validation results of the predictive model using the testing dataset, presenting both correct and incorrect classifications made by the model. Figure 1 depicts the confusion matrix used to interpret the correct and incorrect classifications made by the predictive model on the testing dataset. This 2x2 matrix displays the number of records in the actual dataset (sum of horizontal cells) and the number of records classified by the model (sum of vertical columns). In the confusion matrix, true positives (TP) represent positive sentiment records correctly classified, true negatives (TN) are negative sentiments correctly classified, false positives (FP) are positive sentiments incorrectly classified, and false negatives (FN) are negative sentiments incorrectly classified. TP and TN denote correct classifications, while FP and FN represent incorrect classifications. Consequently, TP+FN represents the total actual positive sentiments, FP+TN the total actual negative sentiments, TP+FP the total predicted positive sentiments, and FN+TN the total predicted negative sentiments.

X	Y
TP	**FP**
FN	**TN**

Figure 1: Simulation outcome's confusion matrix

The classification report assessed the predictive model's performance by calculating various performance evaluation metrics derived from the confusion matrix. The performance evaluation metrics employed to validate the predictive model using the test dataset are outlined below.

1) **Accuracy** – This indicates the percentage of accurate predictions generated by the predictive model, with values expressed in a percentage format. A higher percentage value signifies better model performance.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \qquad (1)$$

2) **True Positive (TP) rate or recall** – This metric represents the percentage of actual sentiment records accurately classified by the predictive model. A value closer to 1 indicates better model performance, showcasing its capability to correctly differentiate between negative and positive sentiments. This metric was calculated individually for each class, followed by the determination of the average value.

$$TP_{negative\ sentiments} = \frac{TP}{TP + FN} \quad (2a)$$

$$TP_{positive\ sentiments} = \frac{TN}{TN + FP} \quad (2b)$$

3) **Precision** – This metric denotes the percentage of predicted sentiment records accurately predicted by the predictive model. A value nearer to 1 indicates superior model performance, showcasing its capability to accurately predict sentiment classes. This metric was computed for each class individually, followed by the determination of the average value.

$$Precision_{negative\ sentiments} = \frac{TP}{TP + FP} \quad (3a)$$

$$Precision_{positive\ sentiments} = \frac{TN}{FN + TN} \quad (3b)$$

4) **F1-score** – This metric was calculated as the harmonic mean of precision and recall, providing a consolidated measure of both metrics. A value approaching 1 signifies superior predictive model performance. This metric was computed individually for each class, followed by the determination of the average value.

$$F1 - score_{negative\ sentiments} = 2 \cdot \left(\frac{Precision_{negative} \times Recall_{negative}}{Precision_{negative} + Recall_{negative}} \right) \quad (4a)$$

$$F1 - score_{positive\ sentiments} = 2 \cdot \left(\frac{Precision_{positive} \times Recall_{positive}}{Precision_{positive} + Recall_{positive}} \right) \quad (4b)$$

4.0 RESULT AND DISCUSSION

This section outlines the outcomes of the tasks undertaken in this study to construct a sentiment classification model for movie reviews sourced from Kaggle. The first part of the results includes the dataset description and visualization of its contents. Following that, the results of the natural language tasks performed on the dataset, along with the utilization of N-Gram feature selections to convert the preprocessed review data into vectors, are presented. The section concludes with the simulation results of the sentiment classification model for movie reviews, including the evaluation of its performance and a comparison with the TF-IDF feature selection model.

4.1 Results of Data Description

The data collection process revealed that the original dataset comprises information from 50,000 review records, consisting of two features. These features, named "Review" and "Sentiment," were found to be of object data type, as shown in Figure 2. To visualize the distribution of sentiments within the "MoviesReview" dataset, a Python command utilizing the seaborn library was used: "sns.countplot(x='sentiment', data=MoviesReview) plt.title("Sentiment distribution")". This count plot effectively presents the frequency of different sentiments in movie reviews, as depicted in Figure 2.

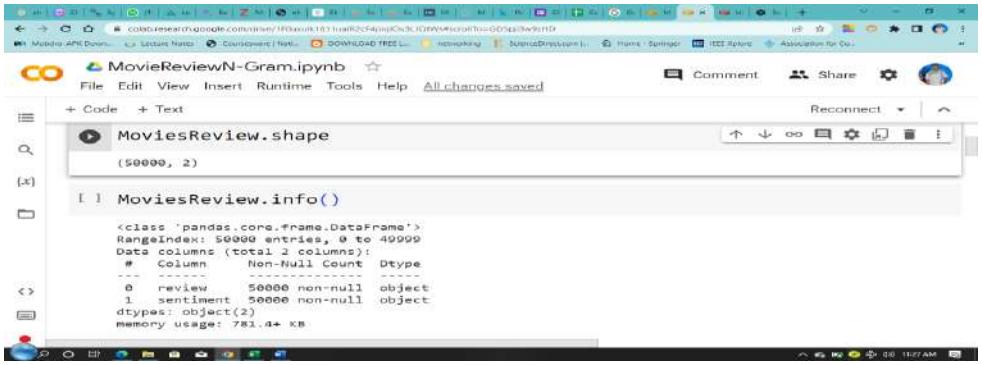

Figure 2: - Movies Review Dataset Information

Figure 3: - Frequency Distribution of Movies Reviews Sentiment

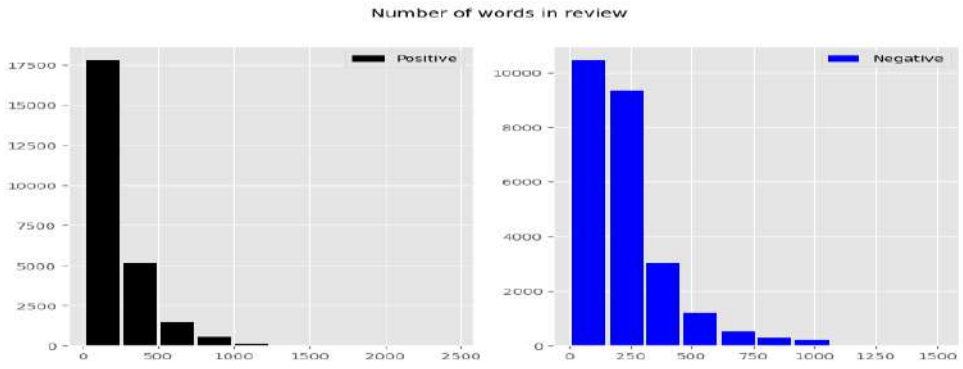

Figure 4: - Histogram of Word Counts in Positive and Negative Reviews

Figure 4 displays two subplots, each presenting a histogram depicting the word counts in positive and negative reviews extracted from the 'MoviesReview' dataset. The first subplot (indexed as 0) is generated by the Python command "ax[0].hist(MoviesReview[MoviesReview['sentiment']== 'positive']['word count'], label='Positive', color='Black', rwidth=0.9)". It uses the 'word count' column from the 'MoviesReview' dataset, specifically for reviews with a positive sentiment. This histogram represents the distribution of word counts in positive reviews. The label='Positive' sets the histogram's label, while the color='Black' parameter determines the color of the histogram bars, which are set to black. The rwidth=0.9 parameter controls the width of the bars relative

to the bin width.Similarly, the Python command "ax[1].hist(MoviesReview[MoviesReview['sentiment'] =='negative']['word count'], label='Negative', color='Blue', rwidth=0.9)" generates the histogram in the second subplot (indexed as 1). It uses the 'word count' column from the 'Movies Review' dataset, specifically for reviews with a negative sentiment. This histogram illustrates the distribution of word counts in negative reviews. The label='Negative' sets the histogram's label, and the color='Blue' parameter determines the color of the histogram bars, which are set to blue.

Figure 5, it showcases a word cloud that visually represents the most frequently occurring words in positive reviews. The size of each word in the cloud corresponds to its frequency in positive movie reviews. This word cloud offers a quick and visually engaging way to identify the prominent words used in positive reviews. Figure 6 presents a bar chart visualization that provides insights into the common words found in positive movie reviews. It facilitates easy comparison of word frequencies and allows for the identification of the most prevalent terms.

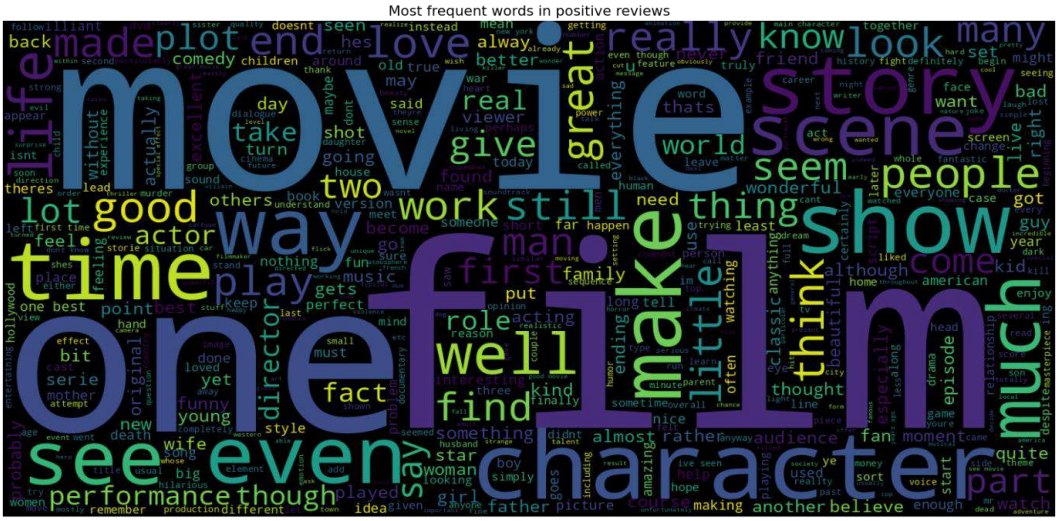

Figure 5: Most frequent words in Positive Review for Movies Reviews

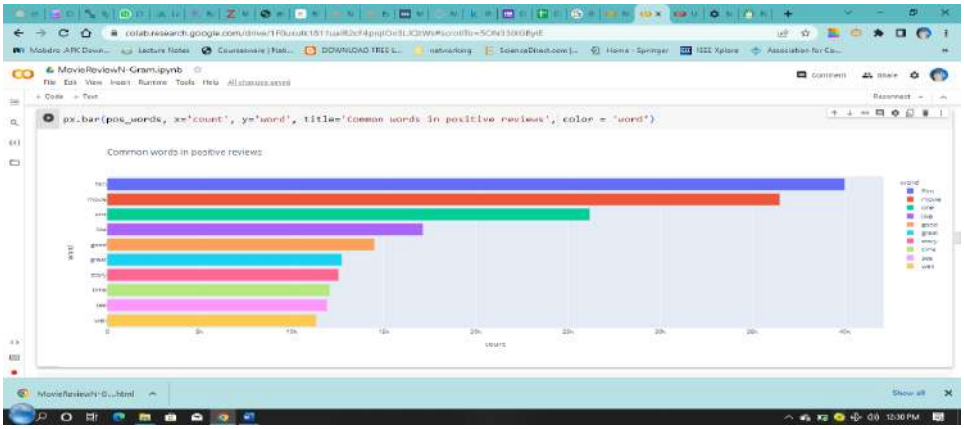

Figure 6: Bar Chart showing common positive word in Movies Review

Figure 7 presents a word cloud that provides a visual representation of the most frequently occurring words in negative reviews. The size of each word in the cloud reflects its frequency in the negative reviews. This word cloud serves as a visual aid to identify the prominent words commonly found in negative reviews. Figure 8, it depicts a resulting bar chart that showcases the frequency or count of each word in the negative reviews. Each word is

represented by a bar, and the length of the bar corresponds to its count. The chart utilizes different colors to distinguish between various words. This visualization facilitates the comparison of word frequencies and aids in the identification of the most prevalent words associated with negative sentiment.

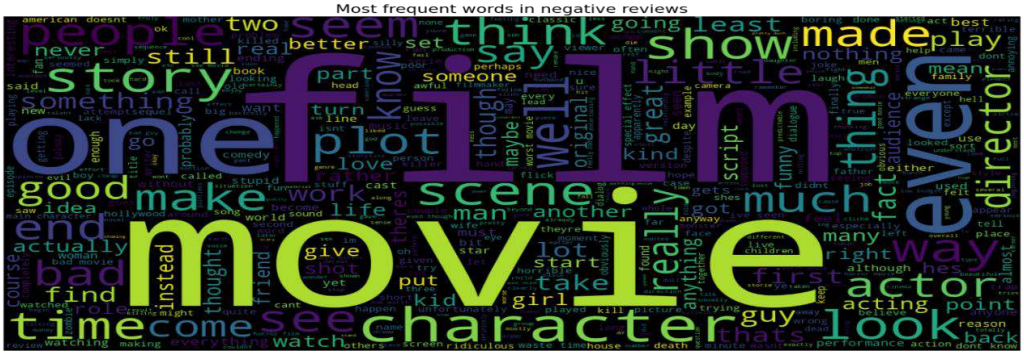

Figure 7: Most frequent words in Negative Review for Movies Reviews

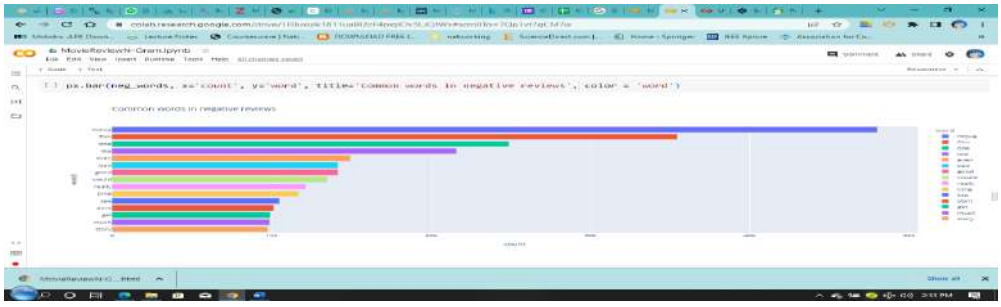

Figure 8: Bar Chart showing common positive word in Movies Review

4.2 Results of Data Preprocessing

The preprocessing of the review portion of the dataset, contributed by the reviewers, was conducted using the Natural Language Toolkit (NLTK). To accomplish this, a function named "data processing" was developed, incorporating a series of text processing techniques. The Python code snippet depicted in Figure 9 illustrates the implementation of these techniques. The "data processing" function encompasses various preprocessing steps, including the conversion of text to lowercase, elimination of specific strings, URLs, Twitter mentions, hashtags, non-alphanumeric characters, and punctuation. Subsequently, the text is tokenized, and stop words are removed. The function ultimately returns the processed text, ready for further analysis and modeling.

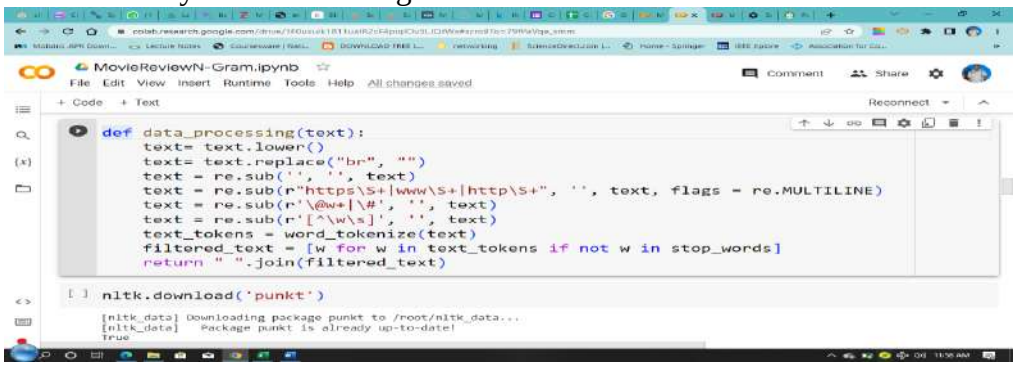

Figure 9: Preprocessing Dataset Python code

4.3 Results of Simulation and Validation of Classification Model

In this study, the dataset underwent division into two distinct segments: the training dataset and the testing dataset. The training dataset was utilized to construct the sentiment analysis classification model, whereas the testing dataset was employed for model performance validation. As outlined in the simulation procedure, the data allocation for training and testing was set at 70% and 30%, respectively. Figure 9 presents the Python code snippet depicting the command used for dataset splitting. The training dataset played a pivotal role in model development, while the testing dataset facilitated performance assessment. Evaluation of model performance was carried out based on the performance evaluation metrics identified within this study.

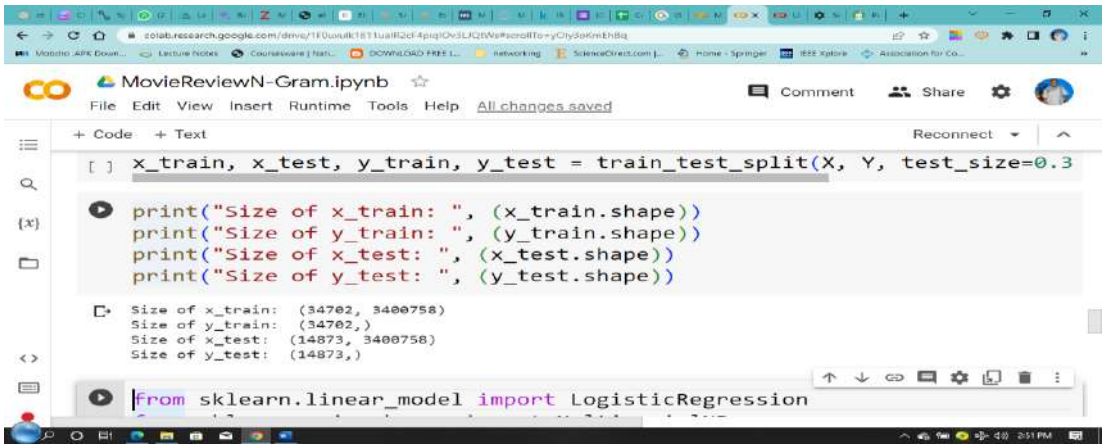

Figure 9: Splitting of Dataset into Training and Testing

a. Multinomial Naïve Bayes Algorithms

Figure 10 and 11 present the confusion matrix and classification report, respectively, for two different algorithms: Multinomial Naive Bayes with N-Gram feature selection and Multinomial Naive Bayes with TF-IDF feature selection. The confusion matrix indicates that the N-Gram approach exhibits slightly fewer false positives 804 and false negatives 830 compared to the TF-IDF approach, which has 896 false positives and 1058 false negatives. Therefore, the N-Gram approach demonstrates a more balanced distribution of errors between these two types.

Table 1 provides a comprehensive summary of the simulation and evaluation results for the classification model utilizing the Multinomial Naive Bayes algorithms. According to the findings, the N-Gram approach demonstrates slightly superior overall accuracy and a more balanced confusion matrix with fewer false positives and false negatives. However, both approaches achieve comparable precision, recall, and F1-scores.

Table 1: Evaluation of Multinomial Naïve Bayes with N-Gram & TF-IDF

Features Selection	Accuracy (%)	Precision			TP rate/Recall			F1-score		
		Negative	Positive	Average	Negative	Positive	Average	Negative	Positive	Average
N-Gram	89.01%	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
TF-IDF	86.86%	0.86	0.88	0.87	0.88	0.86	0.87	0.87	0.87	087

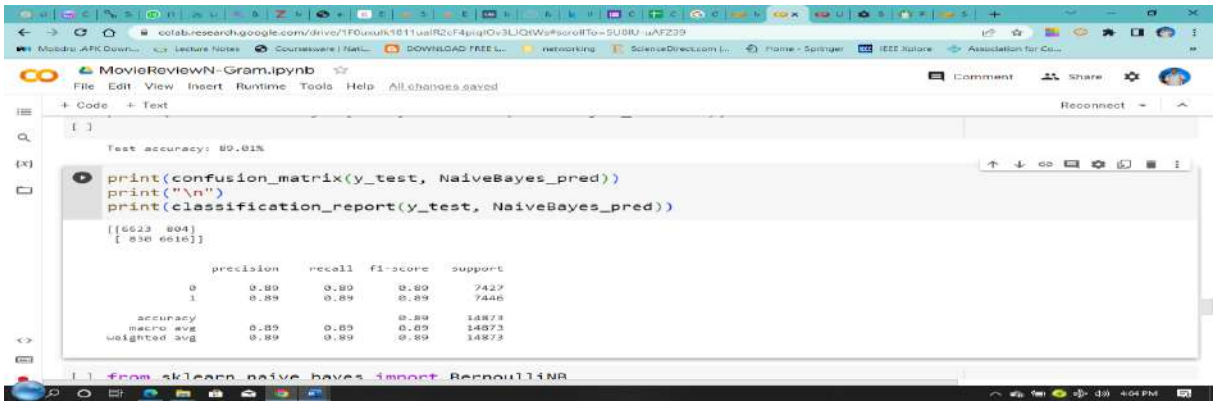

Figure 10: Confusion Matrix using N-Gram with Multimonial

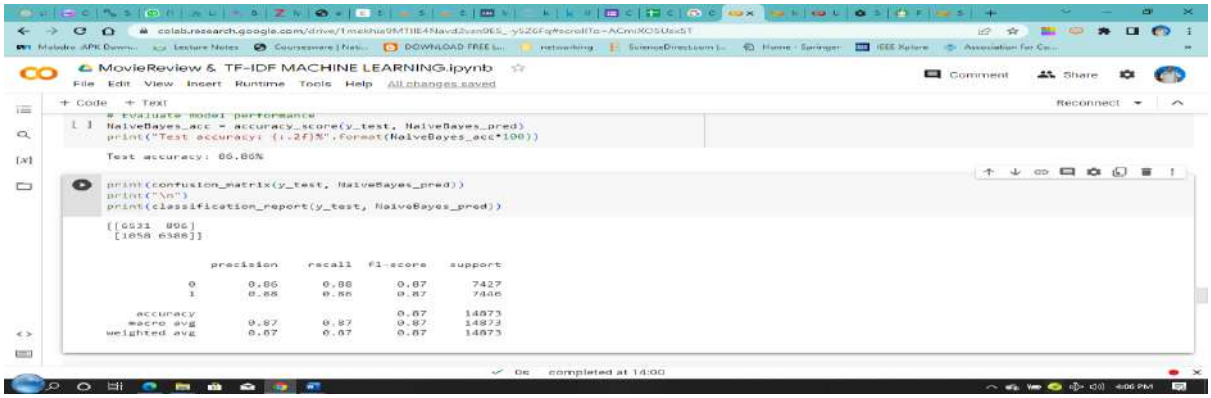

Figure 11: Confusion Matrix using Tf-Idf with Multimonial

a. Bernoulli Naïve Bayes Algorithms

Table 2 displays the outcomes of the simulation and evaluation of the classification model utilizing the Bernoulli Naive Bayes algorithms. The findings reveal that the N-Gram approach with Bernoulli Naive Bayes exhibits slightly superior overall accuracy, achieving 87.92%, compared to the Tf-Idf approach with Bernoulli Naive Bayes, which attains an accuracy of 87.46%. Figure 12 and 13 illustrate the confusion matrices for both the N-Gram and Tf-Idf approaches with Bernoulli Naive Bayes. The results from both approaches depict similar patterns in the confusion matrices, with a slightly higher number of false positives and false negatives observed in the Tf-Idf approach when compared to the N-Gram approach. Consequently, the N-Gram approach with Bernoulli Naive Bayes demonstrates a more favorable balance between these two types of errors.

Table 2: Evaluation of Bernoulli Naïve Bayes with N-Gram & TF-IDF

Features Selection	Accuracy (%)	Precision			TP rate/Recall			F1-score		
		Negative	Positive	Average	Negative	Positive	Average	Negative	Positive	Average
N-Gram	87.92%	0.85	0.92	0.88	0.92	0.83	0.88	0.88	0.87	0.88
TF-IDF	85.46%	0.83	0.88	0.86	0.89	0.82	0.85	0.86	0.85	085

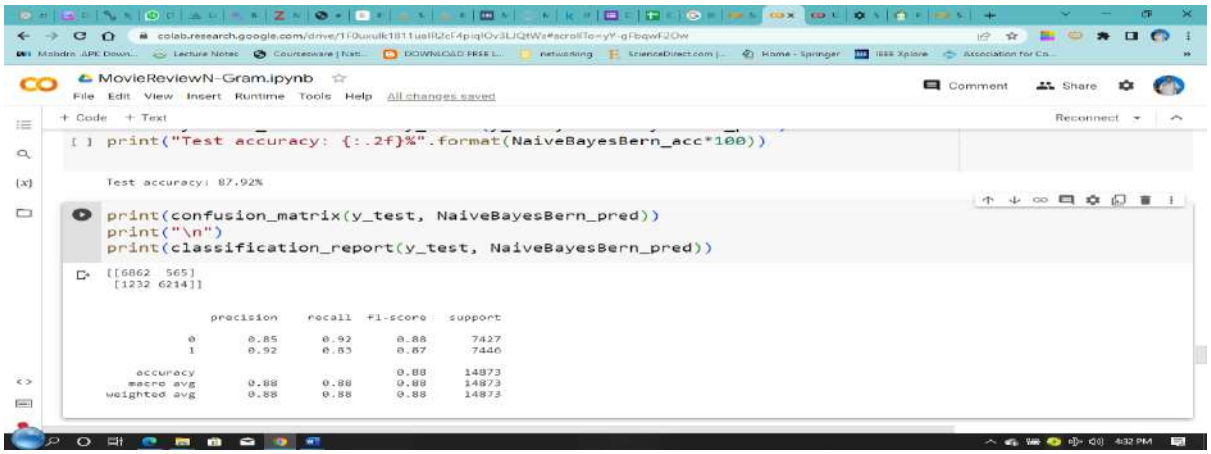

Figure 12: Confusion Matrix Results for N-Gram with Bernoulli

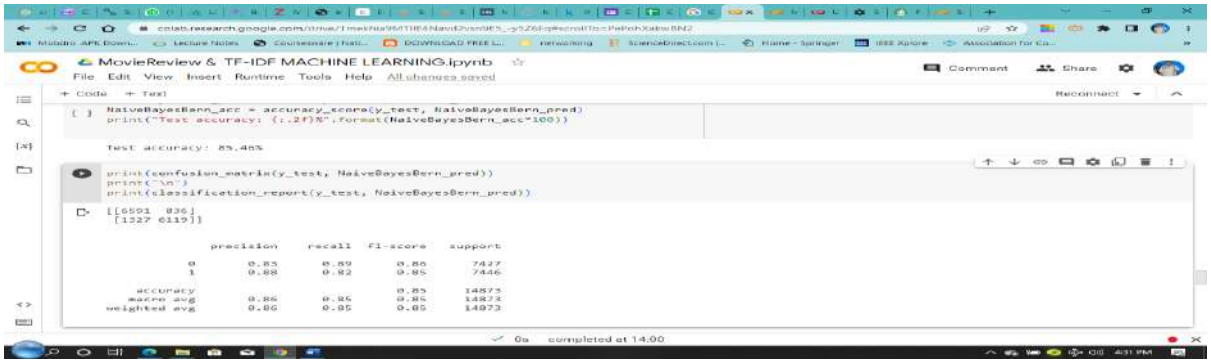

Figure 13: Confusion Matrix of TF-IDF with Bernoulli

a. Logistic Regression algorithms

Table 3 presents the outcomes of the simulation and evaluation of the classification model employing the Logistic Regression algorithms. Both the N-Gram and TF-IDF approaches with Logistic Regression yield comparable accuracy rates, with 88.71% and 89.46% respectively. Figure 14 and 15 display the corresponding confusion matrices. The results reveal similar patterns in both approaches, with a slightly higher number of false positives and false negatives observed in the N-Gram approach compared to the TF-IDF approach. However, the disparities between the two approaches are relatively small.

Table 3: Evaluation of Logistic Regression with N-Gram & TF-IDF

Features Selection	Accuracy (%)	Precision			TP rate/Recall			F1-score		
		Negative	Positive	Average	Negative	Positive	Average	Negative	Positive	Average
N-Gram	88.71%	0.90	0.87	0.89	0.87	0.91	0.89	0.88	0.89	0.89
TF-IDF	89.46%	0.91	0.88	0.89	0.88	0.91	0.89	0.89	0.90	0.89

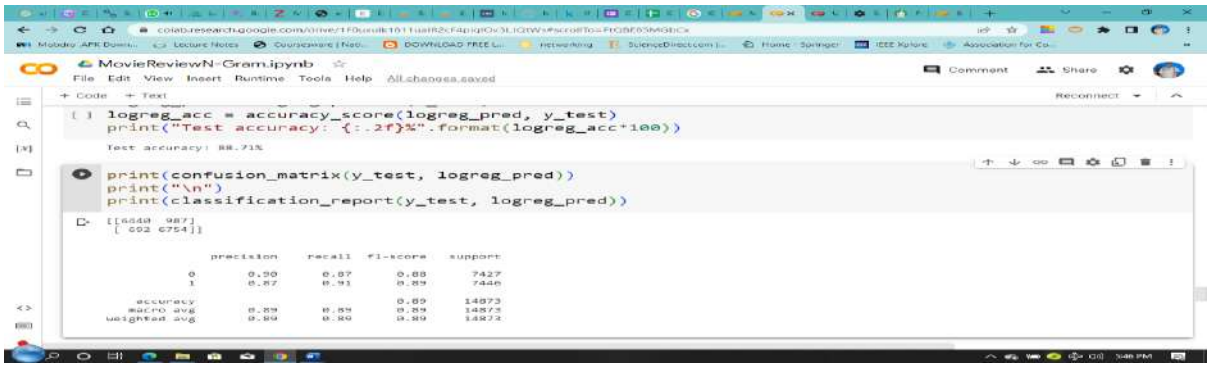

Figure 14: Confusion Matrix on N-Gram with Logistic Regression

Figure 15: Confusion Matrix for Tf-Idf with Logistic Regression

a. Linear Support Vector Classifier

Figure 16 and 17 depict the confusion matrix and classification report of the Linear Support Vector Classifier algorithms, considering both N-Gram and TF-IDF features selections. The confusion matrix reveals that the N-Gram approach exhibits marginally fewer false positives (792) and false negatives (610) compared to the TF-IDF approach, which recorded 845 false positives and 699 false negatives. These results indicate that the N-Gram approach achieves a slightly better balance in terms of classification errors. Table 4 provides an overview of the simulation and evaluation outcomes for the classification model employing the Linear Support Vector algorithms. The N-Gram approach with Linear SVC attains an accuracy of 90.57%, while the TF-IDF approach with Linear SVC achieves an accuracy of 89.62%. These findings suggest that the N-Gram approach outperforms the TF-IDF approach in terms of overall accuracy.

Table 4: Evaluation of Linear Support Vector Classifier with N-Gram & TF-IDF

Features Selection	Accuracy (%)	Precision			TP rate/Recall			F1-score		
		Negative	Positive	Average	Negative	Positive	Average	Negative	Positive	Average
N-Gram	90.57%	0.92	0.90	0.91	0.89	0.92	0.91	0.90	0.91	0.91
TF-IDF	89.62%	0.90	0.89	0.90	0.89	0.91	0.90	0.90	0.90	0.90

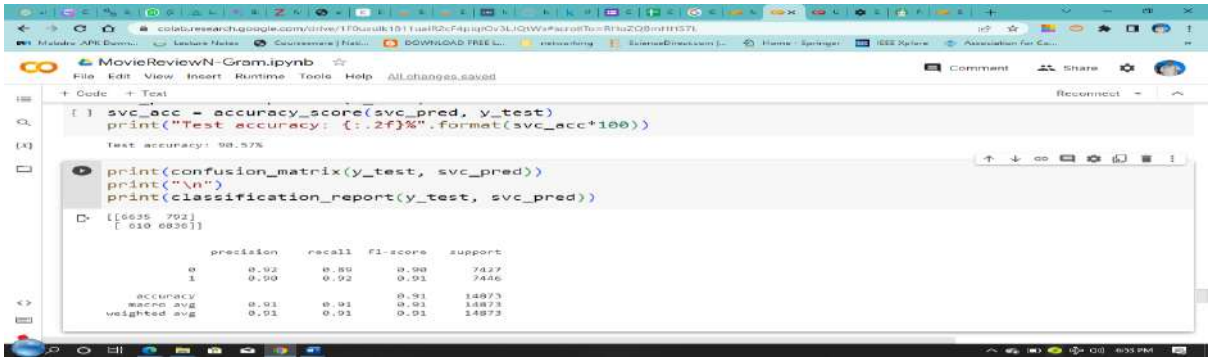

Figure 16: Confusion Matrix for N-Gram with Linear Support Vector Classifier

Figure 17: Confusion Matrix for Tf-Idf with Linear Support Vector Classifier

a. Decision Trees (DT) Algorithms

Table 5 presents the results of the simulation and evaluation of the classification model utilizing the Decision Tree algorithms. The obtained results indicate a small difference in accuracy between the N-Gram and TF-IDF approaches, with accuracies of 68.65% and 68.78% respectively. Additionally, the confusion reports generated for both approaches exhibit similar patterns. The N-Gram approach has a higher number of false positives compared to the TF-IDF approach. Furthermore, the N-Gram approach achieves a significantly smaller number of true positives 3660 compared to the TF-IDF approach 6582, as shown in figure 18 and 19. These findings suggest that the Decision Tree algorithms struggles to accurately classifying certain reviews in movie reviews.

Table 5: Evaluation of Decision Tree Algorithms with N-Gram & TF-IDF

Features Selection	Accuracy (%)	Precision			TP rate/Recall			F1-score		
		Negative	Positive	Average	Negative	Positive	Average	Negative	Positive	Average
N-Gram	68.65%	0.80	0.63	0.72	0.49	0.88	0.69	0.61	0.74	0.67
TF-IDF	68.78%	0.80	0.64	0.72	0.50	0.87	0.69	0.62	0.74	0.68

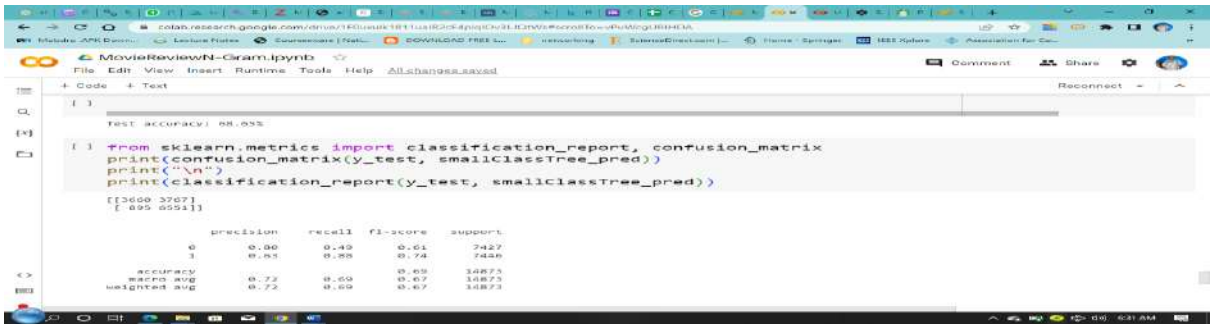

Figure 18: Confusion Matrix for N-Gram with Decision Tree

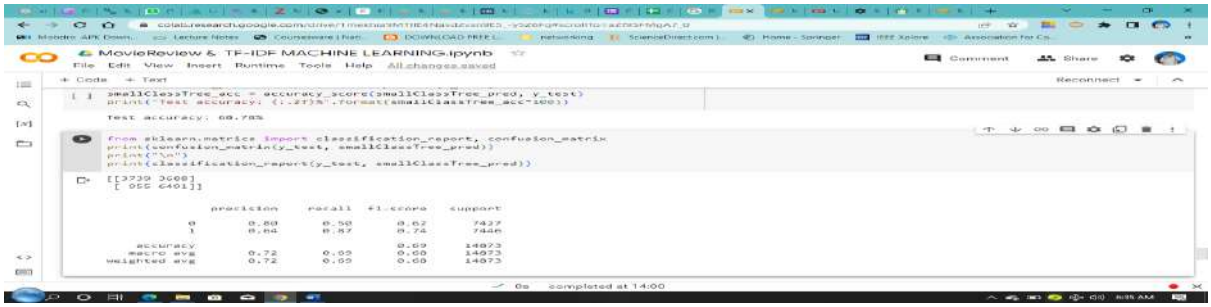

Figure 19: Confusion Matrix for N-Gram with Decision Tree

In summary, the N-Gram approach generally demonstrates slightly superior performance in terms of accuracy and balanced confusion matrices, while the TF-IDF approaches perform comparably but may have slightly higher false positives and false negatives. On the other hand, Decision Tree algorithms exhibit limitations in accurately classifying certain movie reviews.

5.0 CONCLUSION AND RECOMMENDATION

The research findings suggest that the N-Gram approach, specifically when combined with the Linear Support Vector Classifier algorithm, proves to be effective in sentiment classification of movie reviews. This approach demonstrates higher accuracy and yields a more balanced confusion matrix when compared to the TF-IDF approach. The findings also underscore the efficacy of Multinomial Naive Bayes and Logistic Regression algorithms in sentiment classification. However, it is worth noting that the Decision Trees algorithm exhibits limitations in accurately classifying movie reviews. Therefore, the N-Gram approach with Linear Support Vector Classifier emerges as a recommended choice for sentiment analysis in the context of movie reviews. However, future research can focus on exploring ensemble methods, which involve combining multiple classification algorithms, to enhance the accuracy and performance of sentiment classification models for movie reviews. Ensemble methods can leverage the strengths of different algorithms and improve overall prediction accuracy.

REFERENCES

Cahyanti, F. E., Adiwijaya, & Faraby, S. A. (2020). On The Feature Extraction For Sentiment Analysis of Movie Reviews Based on SVM. *2020 8th International Conference on Information and Communication Technology (ICoICT)*, 1–5. https://doi.org/10.1109/ICoICT49345.2020.9166397

Daeli, N. O. F., & Adiwijaya, A. (2020). Sentiment Analysis on Movie Reviews using Information Gain and K-Nearest Neighbor. *Journal of Data Science and Its Applications*, 1-7 Pages. https://doi.org/10.34818/JDSA.2020.3.22

Edeh, M. O., Ugorji, C. C., Nduanya, U. I., Onyewuchi, C., Ohwo , S. O., & Ikedilo, O. E. (2021). Prospects and Limitations of Machine Learning in Computer Science Education. Benin Journal of Educational Studies, 27(1), 48–62. Retrieved from http://beninjes.com/index.php/bjes/article/view/70

Edeh, M.O., Nwafor, C.E., Nnaji, A.D., Fyneface, G.A., Obiekwe, C.P. and Omachi, D. (2020). The Impact of Inquiry-Based Teaching Approach on Computer Science Learning. *EBSU Science Journal*, 1(1), 61–70.

Fan and Fuel,2016. (n.d.). *No online customer reviews means BIG problem in 217*. http://fanandfuel.com/no-online-customer-reviews-means-big-problems-2017/

Khan, A., Gul, M. A., Uddin, M. I., Ali Shah, S. A., Ahmad, S., Al Firdausi, M. D., & Zaindin, M. (2020). Summarizing Online Movie Reviews: A Machine Learning Approach to Big Data Analytics. *Scientific Programming*, *2020*, 1–13. https://doi.org/10.1155/2020/5812715

Lu, K., & Wu, J. (2019). Sentiment Analysis of Film Review Texts Based on Sentiment Dictionary and SVM. *Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence*, 73–77. https://doi.org/10.1145/3319921.3319966

Maulana, R., Rahayuningsih, P. A., Irmayani, W., Saputra, D., & Jayanti, W. E. (2020). Improved Accuracy of Sentiment Analysis Movie Review Using Support Vector Machine Based Information Gain. *Journal of Physics: Conference Series*, *1641*(1), 012060. https://doi.org/10.1088/1742-6596/1641/1/012060

M. Duggan and A. Smith, 2013. (n.d.). *Social media update 2013. Pew Internet & American Life Project Tracking surveys, December 2013*. https://www.pewresearch.org/internet/2013/12/30/social-media-update-2013/

Mitra, A. (2020). Sentiment Analysis Using Machine Learning Approaches (Lexicon based on movie review dataset). *Journal of Ubiquitous Computing and Communication Technologies*, *2*(3), 145–152. https://doi.org/10.36548/jucct.2020.3.004

Mohsin Ahmed, H., & Rabeea Jaber, H. (2020). Sentiment Analysis for Movie Reviews Based on Four Machine Learning Techniques. *Diyala Journal For Pure Science*, *16*(1), 69–87. https://doi.org/10.24237/djps.16.01.514B

Naeem, M. Z., Rustam, F., Mehmood, A., Mui-zzud-din, Ashraf, I., & Choi, G. S. (2022). Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning algorithms. *PeerJ Computer Science*, *8*, e914. https://doi.org/10.7717/peerj-cs.914

Pang, B., & Lee, L., 2008. (n.d.). *Opinion mining and sentiment analysis*.

Spiegel Research Centre, 2017. (n.d.). *Data-Driven Insights on How Retailers can maximize the Value of thier Engagement with customers Through Online Reviews*. https://spiegel.medill.northwestern.edu/_pdf/Spiegel_Online%20Review_ebook_Jun2017_FINAL.pdf.

Zhao, Y; Gupta, RK and Onyema, EM. "Robot visual navigation estimation and target localization based on neural network" *Paladyn, Journal of Behavioral Robotics*, vol. 13, no. 1, 2022, pp. 76-83. https://doi.org/10.1515/pjbr-2022-0005