International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Comparing Machine Learning Models for YouTube Movie Trailer Comments: An Approach for Accuracy and Overall Sentiment Prediction

Aryan Nilakhe
Computer Science and Engineering, Symbiosis Institute of Technology,
Symbiosis International (Deemed University) (SIU), Pune, India
aryan.nilakhe.btech2020@sitpune.edu.in

Aryan Gupta
Computer Science and Engineering, Symbiosis Institute of Technology,
Symbiosis International (Deemed University) (SIU), Pune, India
aryan.gupta.btech2020@sitpune.edu.in

Aryan Jadhav
Computer Science and Engineering, Symbiosis Institute of Technology,
Symbiosis International (Deemed University) (SIU), Pune, India
aryan.jadhav.btech2020@sitpune.edu.in

Tanmay Bholane
Computer Science and Engineering, Symbiosis Institute of Technology,
Symbiosis International (Deemed University) (SIU), Pune, India
tanmay.bholane.btech2020@sitpune.edu.in

Rupali Gangarde *
Computer Science and Engineering, Symbiosis Institute of Technology,
Symbiosis International (Deemed University) (SIU), Pune, India
rupali.gangarde@sitpune.edu.in

Shubhangi Deokar
Computer Science and Engineering, Symbiosis Institute of Technology,
Symbiosis International (Deemed University) (SIU), Pune, India
shubhangi.deokar@sitpune.edu.in

Abstract—This study compares multiple Machine Learning (ML) models for analyzing the sentiment of YouTube comments on movie trailers. The aim of this study is to determine which Machine Learning (ML) model can best accurately predict the overall sentiment of YouTube comments. We compiled a dataset of YouTube comments on a well-known movie trailer and labeled them based on their sentiment using a tokenizer. We then evaluated the performance of different ML models such as Naive Bayes, Support Vector Machine, k-Nearest Neighbors, Random Forest, and Bagging. Our findings show that the Naive Bayes model achieved the highest accuracy for sentiment analysis and provided the most accurate prediction for the overall sentiment of the comments.

Keywords- Classification Report ,Sentiment Analysis, Movie Sentiment Prediction, Natural Language Toolkit, Naive Bayes.

I. INTRODUCTION

Sentiment analysis is an application of machine learning that uses Natural Language Processing more commonly known as NLP which includes determining the sentiment of a text. Because of the vast volume of user-generated material, social media platforms have emerged as a major source for sentiment analysis in recent years. As the second most visited websites and the most popular video-sharing platforms, YouTube has a massive number of user comments on its films, including movie trailers. Opinions can be expressed over what are, for example, products, services, individuals, organizations, or an event. [1] The categorization of good and bad content becomes critical for YouTube users to judge how important the video released is based on user opinion. [2] An example of the relevance of sentiment analysis in this age is the analysis of comments on YouTube about the 2020 US Presidential Election, using SentiWordNet, which revealed a positive reception of Joe Biden's presidency. [3]. In recent literature, various studies have explored sentiment analysis in the context of YouTube videos and user comments, focusing on techniques such as machine learning algorithms for classification [4]. The influence of video duration on user engagement has also been studied, revealing a positive correlation between the amount of negative sentiment in video comments and viewing duration [4].

We examine the accuracy of multiple machine learning (ML) models for sentiment analysis of YouTube video comments on trailers for movies in this work. The ML models

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

evaluated in this study include [5] Naive Bayes, k-Nearest Neighbors (k-NN), Random Forest, Support Vector Machine (SVM), and Bagging. The objective of this study is to find the most accurate and suitable model for predicting the overall sentiment of YouTube comments.

To conduct our study, we collected a dataset of comments on popular movie trailers from YouTube. We labeled the comments based on their sentiment, i.e., positive, negative, or neutral using a tokenizer and then used this labeled dataset to train and test the performance of different ML models.

In our project, we use NLTK (Natural Language Processing Toolkit) to predict the success of a movie based on sentiment analysis of YouTube comments posted on the movie trailer. This helps in providing a relatively accurate representation of how a movie will succeed as opposed to raw numerical data from past releases as such a prediction is more inclined towards sentiment data than regression analysis of previous data from the same producer.

A. Problem Statement

Our problem statement includes comparing the accuracies of various ML models on sentiment analysis of YouTube comments on movie trailers [6][7] and using the best one to produce the overall sentiment and give a better understanding to the filmmakers as to what they can expect from their or someone's movie with respect to public opinion. Our tool will help them analyze whether a movie will succeed or fail based on public opinions in the form of comments.

B. Objectives

Our Objectives are-

1. Extract YouTube comments from movie trailers.
2. Use the NLTK library in python to do text analysis.
3. Perform sentiment analysis on the given comments using the VADER library in NLTK.
4. Provide the user with a GUI and visualization for the analysis with a web-app developed using HTML, CSS, JS and Flask.

C. Project Purpose

Our project provides an accurate analysis of YouTube video comments based on sentiment analysis after comparing the accuracies of multiple ML models and using the highest accuracy model, such that it becomes easy for an individual - say a reviewer or a filmmaker, to predict whether a movie will succeed or not completely based on the public reception of said movie from comments on the YouTube video of the trailer.

Negative and Positive [8][10] keywords are weighted using our dataset and provided to the user as graphs. This helps in the representation of the basic sentiments of the public for the said movie. Machine learning algorithms are the powerful algorithms that result in accuracy. [11]

II. LITERATURE REVIEW

TABLE I. LITERATURE REVIEW TABLE

Sr. No	**Research Paper**		
	Title	***Methodology***	***Research Gap***
[1]	Sentiment Analysis of School Zoning System On Youtube Social Media Using The K-Nearest Neighbor With Levenshtein Distance Algorithm	This study employed a two-situation technique for sentiment analysis on college zoning policies. Scenario I used the K-Means algorithm, whilst Scenario II combined K-Means with Levenshtein Distance for word normalization. Data preprocessing protected case folding, tokenization, Levenshtein normalization, stopword elimination, and stemming. The K-Means set of rules turned into used to categorize sentiments, and guide labeling changed into performed for evaluation. Accuracy become measured the usage of the confusion matrix.lexicon-based unsupervised learning. Various algorithms are employed, and R programming and packages like TwitteR are used for analysis.	The research paper lacked a detailed exploration of alternative sentiment analysis strategies past K-Means and Levenshtein Distance, leaving a studies gap in the investigation of doubtlessly more advanced and accurate algorithms for sentiment classification inside the context of college zoning rules.
[2]	Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes – Support Vector Machine (NBSVM) Classifier	The study used an extensive methodology that combined techniques from sentiment analysis, text mining, and natural language processing. The study thoroughly processed and analysed Indonesian text data using a hybrid approach of Naive Bayes and Support Vector Machine, producing superior sentiment classification findings.	The research report contains gaps in its training data selection criteria and justification for stop words and root word sources. It lacks insights into method limits, future research fields, and fails to compare sentiment analysis methodologies comprehensively. Furthermore, practical ramifications and real-world applications necessitate a more in-depth examination.
[3]	YouTube Sentiment Analysis on US Elections 2020	This paper utilizes ensemble machine learning, combining semantic classification, prior polarity scores, and n-grams features. Models include SVM, Logistic Regression, and Random Forest. The prior polarity score of a tweet is calculated using PMI	While the paper doesn't consistently outperform single base classifiers and doesn't guarantee better performance than the best classifier in the ensemble, it introduces valuable focus on detecting

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Sr. No	Research Paper		
	Title	***Methodology***	***Research Gap***
		between words and positive/negative categories.	implicit semantic meaning in text. Future research aims to enhance performance on neutral tweets with enriched datasets and additional features.
[4]	Sentiment Analysis for Youtube Videos with User Comments: Review	Using NLP and ML, sentiment analysis of YouTube comments is part of the research technique. Simple SA, Complex SA, and Advanced SA are the three levels. Using ML, Simple SA divides comments into two to three kinds (such as good and negative). Advanced SA employs Lexicon-based techniques and expands to include additional classes (happy, sad). Normalisation, stemming, tokenization, and stop word removal are all components of preprocessing. Next, classification using ML or polarity-based lexicon approaches is performed.	There are gaps in the literature, such as the scant attention paid to languages outside English and Indonesian and the very limited span of the sorts of YouTube comments examined. Furthermore, the research does not address difficulties with dealing with sarcasm or conflicting emotions that are frequent in online debates. Future studies could look into these areas and more sophisticated methods for sentiment analysis on social media sites like YouTube.
[6]	Will Sentiments in Comments Influence Online Video Popularity?	To evaluate online video popularity, the study used sentiment analysis and evaluation theory. After reviewing 307 videos and 60,000 comments, eight YouTube channels were chosen. The study presented a Video Popularity Index based on Video Visibility and Sentiment.	The study falls short of linking sentiment analysis results with specific video material and fails to investigate the influence of user demographics on sentiment. Furthermore, it ignores the effect of video quality on viewer sentiment, limiting thorough insights into the dynamics of online video popularity.

III. METHODOLOGY

D. Procedure

1. Data Preprocessing and Transformation

1.1 Data Cleaning

The dataset undergoes an initial cleaning phase. Redundant columns, specifically 'Unnamed: 0', 'Likes', 'Time', 'user', 'UserLink', which do not contribute to sentiment analysis, are systematically removed to streamline the dataset.

1.2 Sentiment Derivation

The VADER sentiment analysis tool from the Natural Language Toolkit library is employed to evaluate the inherent sentiment within each comment. VADER, being lexicon and rule-based, computes four scores for each comment: positive, negative, neutral, and compound. The compound score, a normalized aggregation of the three primary scores, is then used to classify each comment as Positive, Negative, or Neutral based on predefined thresholds.

1.3 Textual Data Transformation

The comments undergo rigorous textual transformation for ease of machine processing. Initial transformations encompass converting text to lowercase and stripping off new line characters, punctuations, special characters, references, and hashtags. Subsequent advanced transformations involve tokenization (splitting text into individual words or tokens), removal of stop words (common words such as 'and', 'the', etc., that don't hold significant meaning), and lemmatization, where words are reduced to their base or root form using WordNetLemmatizer from the Natural Language Toolkit library.

1.4 Label Encoding

To facilitate machine processing, sentiment labels, which are categorical in nature (Positive, Negative, Neutral), are transformed into a numerical format using the LabelEncoder utility from scikit-learn.

1.5 Dataset Balancing

Imbalanced datasets can bias the predictive modeling, favoring the majority class. To prevent this, the script utilizes the resample method from sklearn.utils to up-sample the minority classes, ensuring each sentiment class has an equal number of instances.

2. Feature Engineering

Post data cleaning and transformation, features are extracted from the cleaned comments using the CountVectorizer method from scikit-learn. This method converts the comments into a bag-of-words representation, transforming textual data into a numerical matrix where each row represents a comment and each column represents the frequency of a specific word in that comment.

E. Architecture Diagram

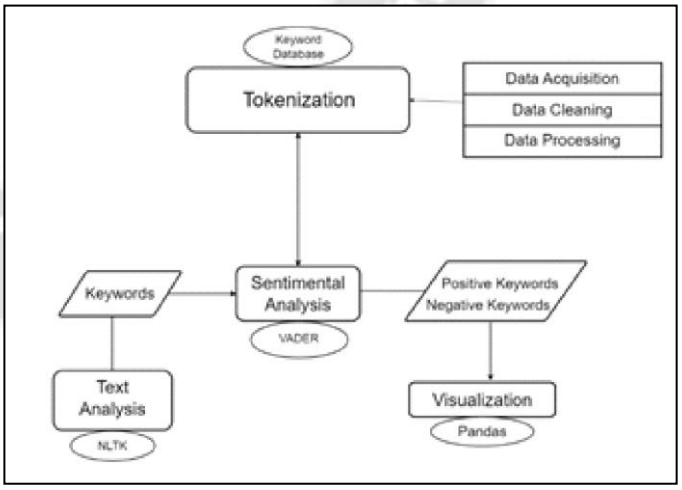


Figure 1. Back End Server-Side Architecture

In the Back End as shown in Figure 1, the comments data acquired from scraping is cleaned and pre-processed. It is then tokenized using VADER library which is part of NLTK (Natural Language Toolkit). The tokenizer tokenizes positive

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

and negative keywords which are the used for text analysis to predict sentiment for the given comment as positive, neutral, or negative.

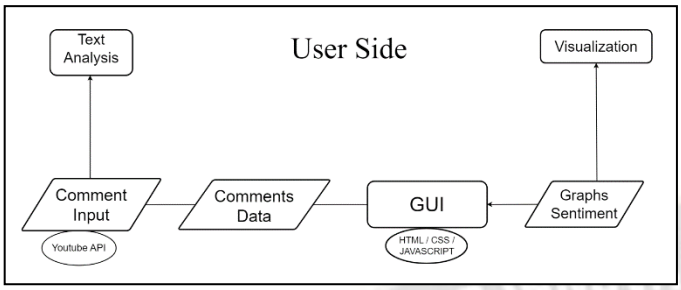


Figure 2. User-Side Architecture

On the user end as seen in Figure 2, a GUI is provided with the help of JavaScript, which scrapes comments data and gives output after analysis in the form of visualization as well as a sentiment score.

IV. Discussions And Results

F. Naive Bayes

Naive Bayes [12] is a popular probability-based algorithm. It is based on the Bayes theorem, which says that it is possible to determine whether a hypothesis is true or false (for instance, if a text is positive or negative) by taking into account the probabilities of its words and other properties. Naive Bayes is useful for sentiment analysis as it can accurately classify text as positive, negative, or neutral based on the recurrence of certain words in the given text.

For example, if the word "nice" appears frequently in a text, the algorithm can assign a value for the text as positive. Similarly, if the word "bad" appears frequently, the algorithm will classify the text as negative.

The Naive Bayes Classifier is based on Bayes' theorem

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \qquad (1)$$

Where:

- P(Y|X) is Posterior probability of class Y given the predictor(s) X.
- P(X|Y) is the likelihood which is the probability of the predictor(s) given class
- P(Y) is prior probability of the class
- P(X) is the prior probability of predictor(s).

One of the major merits of Naive Bayes in sentiment analysis is its capability for handling magnanimous amounts of data and is highly computationally efficient. It can also perform well with limited training data, making it a popular choice for sentiment analysis tasks including our site. Another advantage is that it can handle noisy data well, which is important in sentiment analysis because people often use sarcasm, irony, and other forms of figurative language to express their opinions. Naive Bayes can still classify such texts accurately based on the frequencies of certain words.

Naive Bayes is a simple and effective algorithm for sentiment analysis, which is why in our testing it has the highest accuracy and is used primarily for our sentiment analysis.

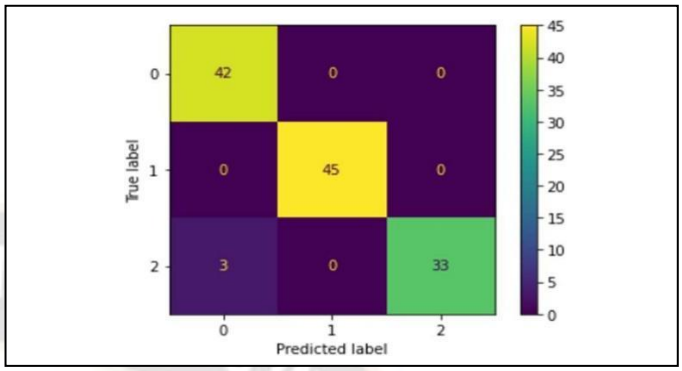


Figure 3. Naive Bayes Confusion Matrix

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
import numpy as np

clf= classifier
vectorizer = CountVectorizer()

X_vect = vectorizer.fit_transform(corpus).toarray()
y = final_data.iloc[:, -1].values
print(len(y))
# Train the classifier
clf = MultinomialNB()
clf.fit(X_vect, y)
```

Figure 4. Naive Bayes Code

The confusion matrix for Naive Bayes in Figure 3 reveals that the model effectively distinguishes between neutral and positive sentiments, achieving perfect accuracy in these categories. However, it encounters challenges in correctly identifying negative sentiments, as it erroneously classifies three instances as negative when they are, in fact, positive. This discrepancy may possibly be because of slight class imbalance as number of samples having positive labels were only 35 compared to 42 and 45 of negative and neutral respectively.

G. Support Vector Machine (SVM)

The SVM algorithm finds the optimum hyperplane for classifying the data using several criteria. [8][10]

SVM is especially helpful for resolving challenging classification issues where the data cannot be separated linearly. The approach locates the hyperplane that optimally separates the data by mapping it into a higher dimensional space where it is linearly separable.

The SVM Classifier finds the hyperplane that best divides the dataset into classes. The mathematical criterion is:

$$y(x) = \omega^T \phi(x) + b \qquad (2)$$

Where:

- ω is the normal vector to the hyperplane
- $\phi(x)$ is the transformed feature space
- b is the bias

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Text analysis activities including text categorization, sentiment analysis, and text clustering may all be accomplished using SVM. The SVM algorithm may be used to categorize text into several groups, such as spam or non-spam emails or various subjects for news articles.

SVM can be used in sentiment analysis to classify text into positive, negative, or neutral emotions. By evaluating the frequency of words or phrases in the text, it discovers patterns that are suggestive of a given mindset.

SVM is particularly effective in text analysis because it can handle high-dimensional data, such as the large number of features that are typically present in text data. Additionally, SVM is a relatively simple algorithm that can be trained with a small amount of data. However, in our case, since our data doesn't have many dimensions, SVM isn't as useful as something like Naive Bayes.

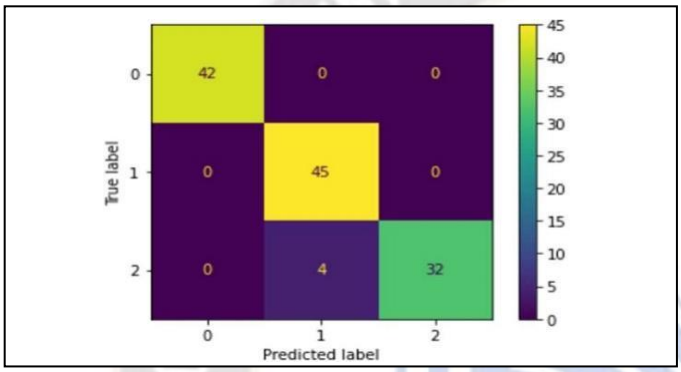


Figure 5. SVM Confusion Matrix

Figure 6. SVM Code

The confusion matrix for the SVM model in Figure 5 demonstrates a different performance pattern compared to Naive Bayes. SVM effectively classifies all negative and positive sentiments, showcasing strong accuracy in these categories. However, it encounters difficulties in distinguishing between neutral and positive sentiments, misclassifying four instances as neutral when they indeed belong to the positive class. Interestingly, unlike Naive Bayes, SVM does not exhibit any misclassifications for the negative class. This may give evidence to the fact that SVM may excel in handling binary classification problems where class separation is more distinct,

```
In [188]: from sklearn import svm
          from sklearn.svm import SVC
          from sklearn.model_selection import train_test_split
          svm = SVC(probability=True)
          svm.fit(X_train, y_train)
          y_pred = svm.predict(X_test)
          svm_score = accuracy_score(y_test, y_pred)
          print('svm accuracy',svm_score)

          svm accuracy 0.967479674796748
```

whereas Naive Bayes might be more sensitive to class imbalance.

H. Random Forest

Random Forest [12] is the most popular for classification tasks, such as sentiment analysis. It works by building a lot of decision trees during training and combining their results to make a prediction.

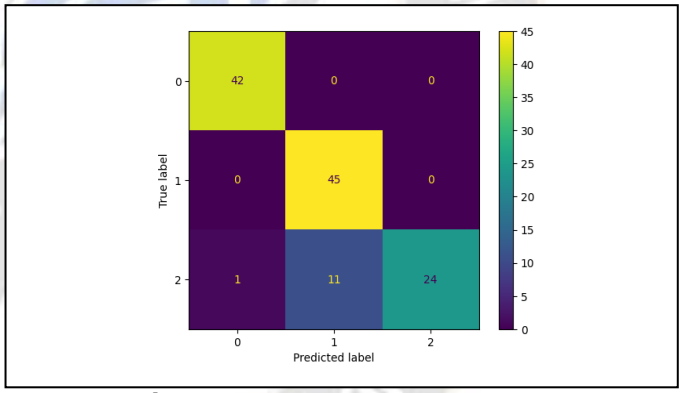
$$\text{Entropy}(S) = -p_{+}\log_2(p_{+}) - p_{-}\log_2(p_{-}) \quad (3)$$

Where:
S is a set of instances
p_+ and p_- are the proportions of positive and negative instances in S, respectively.

In sentiment analysis, the algorithm first pre-processes the data by removing stop-words and converting the text into number values. Then it generates decision trees that are trained on a random subset of the dataset's features and observations (a process known as slicing). During prediction, the algorithm integrates all of the trees' predictions to create a final choice.

It is also useful in sentiment analysis since it can handle large datasets with many features and still maintain pretty good accuracy. It also doesn't have overfitting issues, which occur when a model learns to perform well on the training data but fails to predict new data.

Random Forest can predict emotion by identifying the most essential characteristics. This information may be utilized to better understand the components that lead to a certain sentiment and to improve the model's accuracy during



sentiment analysis.

Figure 7. Random Forest Confusion Matrix

Figure 8. Random Forest Code

In the confusion matrix for the Random Forest model in Figure 7, it displays a generally medium performance across the three sentiment classes. Random Forest effectively classifies 42 samples as negative and accurately identifies 45 samples as neutral. However, it encounters challenges in correctly

```
: from sklearn.ensemble import RandomForestClassifier
  model = RandomForestClassifier(random_state=40)
  model.fit(X_train, y_train)
  y_pred_train = model.predict(X_train)
  y_pred = model.predict(X_test)
  score = accuracy_score(y_test, y_pred)
  print(accuracy_score(y_train, y_pred_train))
  print('random forest accuracy',score)

  1.0
  random forest accuracy 0.9024390243902439
```

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

distinguishing positive sentiments, where it misclassifies 1 instance as negative and 11 instances as neutral when they are indeed positive. Therefore, it must not be the most appropriate model to use in this case.

I. k-Nearest Neighbors (k-NN)

k-Nearest Neighbors [13] is a popular machine learning algorithm that is used for classification and regression. It is used by finding the k nearest data points of a given point and using its labels to conduct prediction of the label of the point.

k-NN doesn't have a specific formula as it works on the principal of distance metrics. However, the commonly used distance metric is the Euclidean distance:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (4)$$

Where:

- *d(x ,y)* is the distance between the points *x* and *y.*
- *n* is the number of dimensions.

It is also useful for solving problems where data is non-linear or does not have a clear decision boundary. The algorithm works well with both numerical and categorical data and can handle noisy data as well.

k-NN could be useful for sentiment analysis, but it is not the most used algorithm for this task. k-NN is used in sentiment analysis to categorize text into distinct sentiment categories such as positive, negative, or neutral. The program analyses the frequency of words or phrases in the text to uncover patterns that suggest a specific attitude. One of the limitations of k-NN for sentiment analysis is that it can be computationally expensive, especially when dealing with large amounts of text data.

Additionally, k-NN requires a large amount of training data to perform well, which can be a challenge for sentiment analysis tasks where labeled data is not easily available. Overall, while k-NN can be useful for sentiment analysis, there are other techniques, such as Naive Bayes and SVM, which are more commonly used for this task and have shown better performance in many applications including ours.

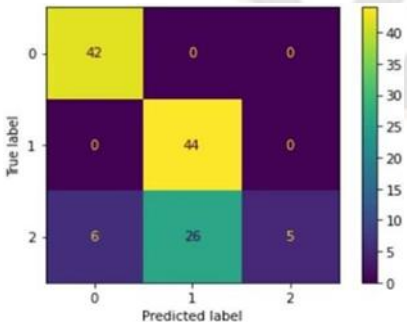


Figure 9. k-NN Confusion Matrix

```
In [190]: from sklearn.neighbors import KNeighborsClassifier
          from sklearn.neighbors import NearestNeighbors
          neigh = KNeighborsClassifier(n_neighbors=3)
          neigh.fit(X_train, y_train)
          y_pred = neigh.predict(X_test)
          knn_score = accuracy_score(y_test, y_pred)
          print('knn accuracy',knn_score)

          knn accuracy 0.7560975609756098
```

Figure 10. k-NN Code

V. COMPARISON OF ALL ALGORITHMS

Notably, the Naive Bayes model exhibited remarkable prowess, attaining an exemplary accuracy score of 0.9756 on the test set. However, a trifling number of misclassifications emerged, primarily occurring when the true label denoted positivity (2), yet the model erroneously predicted negativity (0). In a similar vein, the SVM model exhibited commendable performance, yielding an accuracy of 0.9675 on the test set. Albeit minor, the misclassifications were limited to instances where the true label suggested positivity (2), only to be predicted as neutrality (1).

In contrast, the k-NN model struggled to maintain the same level of accuracy, registering a score of 0.7561 on the test set. Notably, this model faced challenges when confronted with instances belonging to the "2" class (positive), consequently leading to misclassifications. Conversely, the Random Forest Classifier showcased a moderate performance, recording an accuracy score of 0.9024.

In pursuit of further augmenting the accuracy score, innovative ensemble techniques such as Bagging and Voting were meticulously explored. However, these methods failed to surpass the exceptional accuracy score attained by Naive Bayes operating autonomously.

Proposed research paper's thorough investigation highlights the comparative effectiveness of many machine learning techniques for sentiment analysis in the domain of YouTube movie trailer comments. It is evident that Naive Bayes and SVM emerged as frontrunners, with Naive Bayes exhibiting a slight edge in accuracy.

TABLE II. COMPARISON OF ALL ALGORITHMS

Sr. No	**Parameters**	
	Name of Algorithm	***Accuracy***
1	Naive Bayes	97.56%
2	SVM	96.75%
3	Random Forest	90.24%
4	k-Nearest Neighbors	75.61%

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

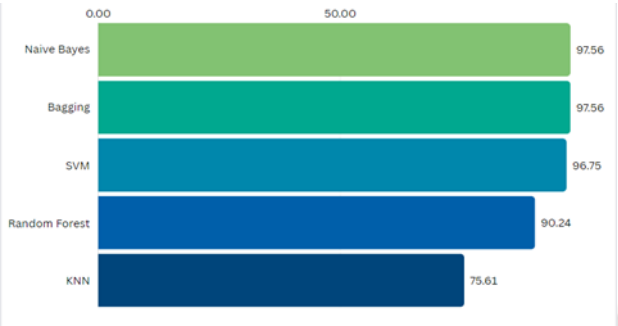


Figure 11. Horizontal Bar Chart comparing all Algorithms

In Figure 11, the chart shows that Naive Bayes and Bagging classifiers perform exceptionally well, both achieving an accuracy of 97.56%. SVM follows closely with 96.75% accuracy. However, Random Forest lags behind at 90.24%, while k-NN trails with the lowest accuracy at 75.61%. Despite attempting an ensemble approach with a Voting Classifier, results remain consistent with Naive Bayes and Bagging, suggesting their effectiveness in this sentiment analysis task.

VI. TOOLS

J. Python

Python was used as the primary programming language for our research, along with the Pandas library for data handling and analysis. Pandas is a powerful Python library that has various data structures like Pandas data frames used here and tools for data manipulation and analysis. With Pandas, we were able to pre-process and analyze our dataset efficiently, which helped us to extract meaningful insights from the data.

K. Natural Language Toolkit (NLTK)

We used the Natural Language Toolkit (NLTK) package for text analysis, which is a widely used open-source Python Natural Language Processing (NLP) library and can be installed with just a few steps. It is a framework for developing text analysis programs that may be found in the Python library [14]. It provides a wide range of tools and algorithms for various NLP tasks, including sentiment analysis. By adopting an NLP-based approach to analyze user comments, as demonstrated in related studies [15], we can uncover sentiments that shed light on video quality and user engagement, enabling improved strategies for enhancing video performance. We utilized the NLTK library to perform various preprocessing tasks like tokenization, stemming, and stop word removal, which helped us to obtain clean and meaningful text data before passing it to the algorithm.

L. VADER

VADER (Valence Aware Dictionary and sentiment Reasoned), uses machine learning for lexicon and rule-based algorithms and is widely used to define and analyze sentiments posted on social media. Its full open-source nature is governed by the MIT License.

M. Classification Algorithms

Naive Bayes Algorithm will be our primary classification algorithm to provide classification of keywords as well as sentiments. It will be used to define weightage for each keyword and hence provide an accurate analysis.

Other algorithms that we have used for comparison include SVM, k-NN (k-Nearest Neighbors), Random Forest, Bagging. Studies suggest LSTM deep learning algorithms can also be implemented using Softmax activation and Adam optimization. [16]

N. HTML, CSS, JS, Flask

We have employed Flask along with HTML, CSS, and Chart JS as the front-end tools to display graphs and results. Flask is a Python web framework for creating small-scale online applications. The application's user interface was built with HTML and CSS, and the results were visualized with ChartJS. With the help of the aforementioned technologies, we were able to construct the desired web application that displayed our sentiment analysis results in an understandable manner for the average user.

VII. CONCLUSION AND FUTURE SCOPE

To conclude, the aim of this study was to assess the effectiveness of several machine learning techniques for sentiment analysis of movie trailer comments on YouTube, as well as to evaluate their accuracy and overall sentiment prediction. Among the models evaluated, Naive Bayes exhibited the highest accuracy of 97.56% with SVM following closely with an accuracy of 96.75% however SVM won't be able to categorize the neutral sentiments. Other algorithms that were implemented include SVM, k-NN, and Random Forest.

The future scope of this project includes adding a web scraper to fetch comments of videos for which the URL is provided and then perform sentiment analysis on the same.

REFERENCES

[1] N. Anggraini and M. J. Tursina, "Sentiment Analysis of School Zoning System On Youtube Social Media Using The K-Nearest Neighbor With Levenshtein Distance Algorithm," 2019 7th International Conference on Cyber and IT Service Management (CITSM), Jakarta, Indonesia, 2019, pp. 1-4, doi: 10.1109/CITSM47753.2019.8965407.

[2] A. N. Muhammad, S. Bukhori and P. Pandunata, "Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes – Support Vector Machine (NBSVM) Classifier," 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), Jember, Indonesia, 2019, pp. 199-205, doi: 10.1109/ICOMITEE.2019.8920923.

[3] S. Singh and G. Sikka, "YouTube Sentiment Analysis on US Elections 2020," 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), Jalandhar, India, 2021, pp. 250-254, doi: 10.1109/ICSCCC51823.2021.9478128.

[4] R. F. Alhujaili and W. M. S. Yafooz, "Sentiment Analysis for Youtube Videos with User Comments: Review," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 814-820, doi: 10.1109/ICAIS50930.2021.9396049.

[5] S. Sharma, A. Pandey, V. Kumar, D. Ohdar, A. R. Pillai and M. Mahajan, "Recent Trends in Sentiment Analysis using Different Machine Learning based Models: A Short Review," 2023 2nd International Conference on Applied Artificial

Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 474-481, doi: 10.1109/ICAAIC56838.2023.10140954.

[6] W. -L. Chang, "Will Sentiments in Comments Influence Online Video Popularity," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 3644-3646, doi: 10.1109/BigData.2018.8621938.

[7] H. Timani, P. Shah and M. Joshi, "Predicting Success of a Movie from Youtube Trailer Comments using Sentiment Analysis," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2019, pp. 584-586.

[8] R. Pradhan, "Extracting Sentiments from YouTube Comments," 2021 Sixth International Conference on Image Information Processing (ICIIP), Shimla, India, 2021, pp. 1-4, doi: 10.1109/ICIIP53038.2021.9702561.

[9] H. A. Jannah and D. Hermawan, "Analysis of Indonesian Society's Perceptions of the COVID-19 Vaccine in Youtube Comments Using Machine Learning Algorithms," 2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia, 2022, pp. 72-77, doi: 10.1109/AiDAS56890.2022.9918796.

[10] M. Alkaff, A. Rizky Baskara and Y. Hendro Wicaksono, "Sentiment Analysis of Indonesian Movie Trailer on YouTube Using Delta TF-IDF and SVM," 2020 Fifth International Conference on Informatics and Computing (ICIC), Gorontalo, Indonesia, 2020, pp. 1-5, doi: 10.1109/ICIC50835.2020.9288579.

[11] V. Uma Ramya and K. Thirupathi Rao, "Sentiment analysis of movie review using machine learning techniques," Int. J. Eng. Technol., vol. 7, no. 16, pp. 676–681, 2018, doi: 10.14419/ijet.v7i2.7.10921

[12] M. Aufar, R. Andreswari and D. Pramesti, "Sentiment Analysis on Youtube Social Media Using Decision Tree and Random Forest Algorithm: A Case Study," 2020 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 2020, pp. 1-7, doi: 10.1109/ICoDSA50139.2020.9213078.

[13] I. Irawaty, R. Andreswari and D. Pramesti, "Development of Youtube Sentiment Analysis Application using K-Nearest Neighbors (Nokia Case Study)," 2020 3rd International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2020, pp. 39-44, doi: 10.1109/ICOIACT50329.2020.9332151.

[14] I. Irawaty, R. Andreswari and D. Pramesti, "Development of Youtube Sentiment Analysis Application using K-Nearest Neighbors (Nokia Case Study)," 2020 3rd International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2020, pp. 39-44, doi: 10.1109/ICOIACT50329.2020.9332151.

[15] H. Bhuiyan, J. Ara, R. Bardhan and M. R. Islam, "Retrieving YouTube video by sentiment analysis on user comment," 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuching, Malaysia, 2017, pp. 474-478, doi: 10.1109/ICSIPA.2017.8120658.

[16] A. M. Putri, D. A. P. Basya, M. T. Ardiyanto and I. Sarathan, "Sentiment Analysis of YouTube Video Comments with the Topic of Starlink Mission Using Long Short Term Memory," 2021 International Conference on Artificial Intelligence and Big Data Analytics, Bandung, Indonesia, 2021, pp. 28-32, doi: 10.1109/ICAIBDA53487.2021.9689754.