Classifying YouTube Comments Based on Sentiments using Hybrid Machine Learning Model

Project/Thesis ID. 2023: 08

Session: BSc. Fall 2019-2023

Project Supervisor: Engr. Khurum Iqbal

Submitted By

Laiba Manzoor
Tayyab Mahmood
Sohaib Qayyum

Department of Software Engineering

University of Azad Jammu and Kashmir Muzaffarabad

Certification

This is to certify that **Tayyab Mahmood**, **2019-SE-10, Laiba Manzoor**, **2019-SE-30** and **Sohaib Qayyum**, **2019-SE-34** have successfully completed the final project **Classifying YouTube Comments Based on Sentiments using Hybrid Machine Learning Models,** at the **University of Azad Jammu and Kashmir Muzaffarabad**, to fulfill the partial requirement of the degree **BSc. SE.**

External Examiner

Dr. Nouman Ali

Chairman, Software Engineering Department

Of MUST University AJK

Project Supervisor

[**Engr. Khurum Iqbal**]

Professor

Chairman

Dr. Ghulam Sarwar

Department of Software Engineering, University of Azad Jammu & Kashmir

Project Title (Classifying YouTube Comments Based on Sentiments using Hybrid Machine Learning Model)

Sustainable Development Goals

(Please tick the relevant SDG(s) linked with FYDP)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	√ SDG 9	Industry, Innovation, and Infrastructure
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well Being	SDG 11	Sustainable Cities and Communities
√ SDG 4	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		√ SDG 17	Partnerships for the Goals

Range of Complex Problem Solving			
	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	
4	Familiarity of issues	Involve infrequently encountered issues	
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	
7	Consequences	Have significant consequences in a range of contexts.	
8	Interdependence	Are high level problems including many component parts or sub-problems	
Range of Complex Problem Activities			
	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

Abstract

The project "Classifying YouTube Comments Based on Sentiments Using a Hybrid Machine Learning Model" addresses the growing need to understand and categorize sentiments expressed in YouTube comments. YouTube, as one of the largest video-sharing platforms, contains a vast amount of user-generated content, including comments, making sentiment analysis a valuable tool for content creators, marketers, and researchers. This project employs a hybrid machine-learning model to automatically classify sentiments in YouTube aiming to facilitate the extraction of valuable insights from this data. The study's significance lies in its potential to enhance the user experience on YouTube and assist content creators and administrators in maintaining a safer and more engaging online environment.

The project focuses on building a machine learning technique that can classify sentiments in YouTube comments using hybrid machine learning models. In our project, we will extract and classify the raw comments into different categories based on both sentiment and sentence types that will help YouTubers find relevant comments for growing their viewership. We Hybrid model for better accuracy and less error rate. In recent years, YouTube has gained huge popularity among content creators. A large number of content creators upload their video content on this platform. These videos get tons of views and comments. The content creators, more generally called YouTubers, need to continuously work on maintaining the quality and quantity of their contents. To do so, they must collect feedback from their viewers through the comments section. This feedback lets them understand the influence of their creations. In addition to improving audience engagement, feedback also provides information on the aspects of the content that need improvement. With this YouTuber can easily check their content by using our model whether the public like or dislike their content. It also helps the viewers which content is more informative .It will analyze and classify user comments as positive, negative, Interrogative, Imperative or neutral sentiments to assist content creators in understanding audience feedback.

Undertaking

I certify that the project **Classifying YouTube Comments Based on Sentiments Using a Hybrid Machine Learning Model** is our own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged/ referred.

Tayyab Mahmood

2019-SE-10

Laiba Manzoor

2019-SE-30

Sohaib Qayyum

2019-SE-34

Acknowledgement

We truly acknowledge the cooperation and help make by **Engr. Khurum Iqbal, Professor of Department of software Engineering.** He has been a constant source of guidance throughout the course of this project.

We are also thankful to our friends and families whose silent support led us to complete our project.

Table of Contents

List of Tables

List of Figures

Chapter 1

1.1 Introduction

Chapter 1 provides an introduction to the research project, "Classifying YouTube Comments Based on Sentiments Using a Hybrid Machine Learning Model" It sets the stage by offering a comprehensive background of the study, highlighting the problem statement, objectives of the research, research questions, the significance of the study, and the scope and limitations. Additionally, this chapter provides a glimpse of the thesis structure, outlining how the subsequent chapters will contribute to addressing the identified research problem and objectives. YouTube, as one of the world's largest and most influential online video-sharing platforms, has become a thriving hub for global content creation, communication, and user engagement. With billions of daily users, it not only hosts an abundance of videos but also facilitates an extensive exchange of user-generated comments. These comments, ranging from praise and constructive feedback to negativity and toxic expressions, hold invaluable insights into user sentiments and perceptions. Consequently, there is a growing necessity to analyze and understand the sentiments expressed within YouTube comments to support content creators, advertisers, and platform administrators in making informed decisions and maintaining a safer and more engaging online environment.

1.2 Statement of the problem

The task of sentiment analysis within YouTube comments, however, presents distinct challenges due to the sheer volume, diversity, and dynamism of the data. Traditional methods often struggle to provide accurate sentiment classification, calling for advanced solutions that can effectively handle this complex task. In response to these challenges, this research project sets out to develop and implement an advanced sentiment analysis system that leverages a hybrid machine-learning model and utilizes web scraping techniques to collect a substantial dataset of YouTube comments. This model aims to classify sentiments with a high degree of accuracy, offering valuable insights into user sentiments and paving the way for enhanced user experiences on the platform.

1.3 Goals/Aims & Objectives

The primary objective of this study is to create a robust and accurate sentiment analysis system for YouTube comments using a hybrid machine-learning model. Specific objectives include: A web scraping to collect a substantial dataset of YouTube comments. Preprocessing and cleaning the collected data to prepare it for analysis. Implementing a hybrid machine-learning model for sentiment classification. Evaluating the model's performance using appropriate metrics. Contributing to the body of knowledge in the field of sentiment analysis, particularly in the context of user- generated content

1.4 Motivation

This study's significance lies in its potential to enhance the understanding of user sentiments within YouTube comments, facilitating content optimization, user engagement strategies, and content moderation efforts on the platform. By addressing the challenges inherent in sentiment analysis within the unique context of YouTube, this research aims to contribute to the fields of natural language processing, machine learning, and social media analytics, while offering practical insights for platform administrators, content creators, and online communities. Ultimately, the project strives to foster a more inclusive, informative, and positive online environment for all YouTube users.

1.5 Methods

In our approach, we make use of python programming language as in to execute the experiment. Our approach combines procedural Python programming with machine learning, integrating libraries like NumPy, Pandas, Keras, TensorFlow, SciPy, Sci-Kit Learn and Matplotlib for efficiency. We prioritize code simplicity, using Python's logic flow to transform news content effectively. Modular design enhances reusability. Machine learning augments our system's intelligence. Iterative development and innovation shape our software process model. We sculpt our system in stages, adapting to evolving fake news challenges.

1.6 Report Overview

The project focuses on building a machine learning technique that can classify sentiments in YouTube comments using hybrid machine learning models. In our project, we will extract and classify the raw comments into different categories based on both sentiment and sentence types that will help YouTubers find relevant comments for growing their viewership. We Hybrid model for better accuracy and less error rate. In recent years, YouTube has gained huge popularity among content creators. A large number of content creators upload their video content on this platform. These videos get tons of views and comments. The content creators, more generally called YouTubers, need to continuously work on maintaining the quality and quantity of their contents. To do so, they must collect feedback from their viewers through the comments section. This

feedback lets them understand the influence of their creations. In addition to improving audience engagement, feedback also provides information on the aspects of the content that need improvement. With this YouTuber can easily check their content by using our model whether the public like or dislike their content. It also helps the viewers which content is more informative .It will analyze and classify user comments as positive, negative, Interrogative, Imperative or neutral sentiments to assist content creators in understanding audience feedback

Chapter 2

2.1 Sentiment analysis

Sentiment analysis is a fundamental component of this research, and understanding these challenges is crucial for developing a robust and accurate hybrid machine-learning model for classifying YouTube comments based on sentiments. Sentiment analysis has gained immense importance in various domains due to its potential to extract valuable insights from vast amounts of textual data. Understanding the sentiment of customers, users, or the public is crucial for a variety of reasons:

2.1.1 Market research:

In business, sentiment analysis is used to gauge consumer reactions to products and services. Positive sentiment can indicate customer satisfaction, while negative sentiment might point to areas that need improvement.

2.1.2 Social media monitoring:

On platforms like Twitter and Facebook, companies and individuals can monitor social sentiment to track their brand reputation and respond to public opinions.

2.1.3 Customer feedback analysis:

Sentiment analysis is used to analyze customer reviews, comments, and feedback to identify common issues or areas of satisfaction.

2.1.4 Political analysis:

In politics, sentiment analysis can be applied to assess public opinion, track political discourse, and predict election outcomes.

2.1.5 Content personalization:

Online platforms and recommendation systems use sentiment analysis to personalize content and recommendations for users based on their emotional responses.

2.2 Using YouTube comments for text-based emotion recognition

Here uses an unsupervised machine-learning algorithm that performs emotion classification, based on a data corpus built from YouTube comments [2]. The reason behind such a choice is the similarity between YouTube comments and instant messages writing style. We classify emotions according to the six basic emotions. Each emotion category is represented by a list of expressive words. In addition, to determine the emotion expressed in a piece of text, we first classify its component words [2]. The data corpus that we use to compute the different PMIs is built by importing comments from YouTube using YouTube API version 3. To ensure having enough rich content in the corpus, we browse videos from different YouTube categories (divertissement, Blogs & People …) using keywords relevant to the six emotions of Ekman. Once videos identifiers are retrieved, we import the corresponding comments [2].

First conducted tests give high precision ranging from 91% to 95% for different target emotions. To run tests, we choose two different types of sentences. The first type contains affective words that correspond to each of the target emotions, and the second type does not contain any affective words.

2.2.1 Evaluation and results

Our data corpus contains over 200.000 comments retrieved from different YouTube videos. From this data, we extract six representative sub-corpuses for the six target emotions. Our system achieves an average precision of 92.75%, and 68.82% as average accuracy, which is close to measures given by previous systems, using SVM as machine learning algorithms.

2.3 Sentiment analysis of students' comment using lexicon based approach

One of the simplest ways to address the problem is to categorize the comments purely based on lexicon [3]. e.g., the interrogative comments can be identified from keywords such as what, how, and why. Similarly, positive sentences can be identified from keywords like good, best, and wonderful. However, this approach is naive and does not address unique challenges presented by informal texts. Moreover, this method performs poorly if a single comment comprises of multiple categories. Such comments can be be categorized more efficiently by appropriately extracting features from the text corpus and using supervised machine learning techniques. This research work focus on students' feedback comments [3]. Feedback analysis is more important to measure the performance of teacher. Analyzing students' comments, using sentiment analysis approaches can

classify the students' positive or negative feelings. Sometimes students do not understand what the lecturer is trying to explain, thus by providing feedbacks, students can indicate this to the lecturer.

2.3.1 System architecture

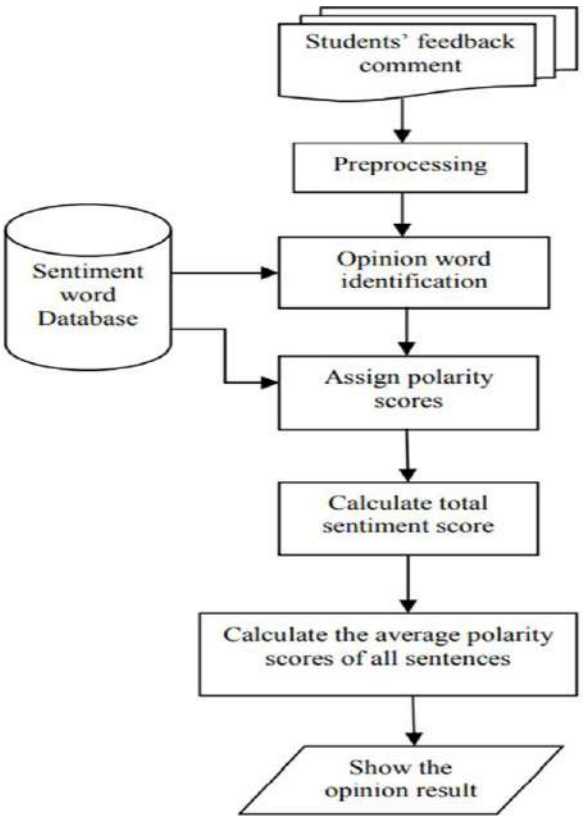


Figure 1 System Architecture

Chapter 3

3.1 SYSTEM ARCHITECTURE

In this chapter, we delve into the system architecture that underlies our project, "Classifying YouTube Comments Based on Sentiments using a Hybrid Machine Learning Model [7]. The architecture of the system plays a pivotal role in data collection, preprocessing, sentiment analysis, and model deployment. This chapter provides a comprehensive overview of the system's components, their interactions, and their roles in achieving the research objectives. The system architecture serves as the structural backbone of our study, orchestrating the processes of data collection, preprocessing, sentiment analysis, and model deployment. In this chapter, we delve into the intricate details of each architectural component, elucidating their functionalities, interconnections, and contributions to the research objectives. By offering a detailed insight into the system's inner workings, this chapter enables a thorough understanding of how our methodology leverages cutting-edge technologies and computational processes to execute sentiment analysis on YouTube comments, furthering the academic discourse in this domain.

3.2 High-level architecture

The high-level architecture of the project ensures a systematic and organized approach to classifying YouTube comments based on sentiments. By integrating user-friendly interfaces, data preprocessing, hybrid machine learning models, and secure data management, the system delivers accurate sentiment analysis results to assist content creators in understanding audience feedback effectively. Overall, the high-level architecture of this project provides a robust framework for classifying YouTube comments based on sentiments, leveraging the power of hybrid machine learning models and efficient data processing techniques. The high-level architecture serves as a guidepost for understanding the system's structure and functioning, laying the groundwork for a more detailed exploration of each architectural component in subsequent sections of this chapter.

3.3 Detailed system components

Various components work together to achieve the goal of sentiment analysis on YouTube comments. These components are crucial for the successful execution of the project. The key components of the system are Data Preprocessing and machine learning models that are:

- Linear support Vector Classifier

- Random Forest
- Decision Tree
- Multinomial Naive Bayes
- K-Nearest Neighbor

3.4 Data flow diagram

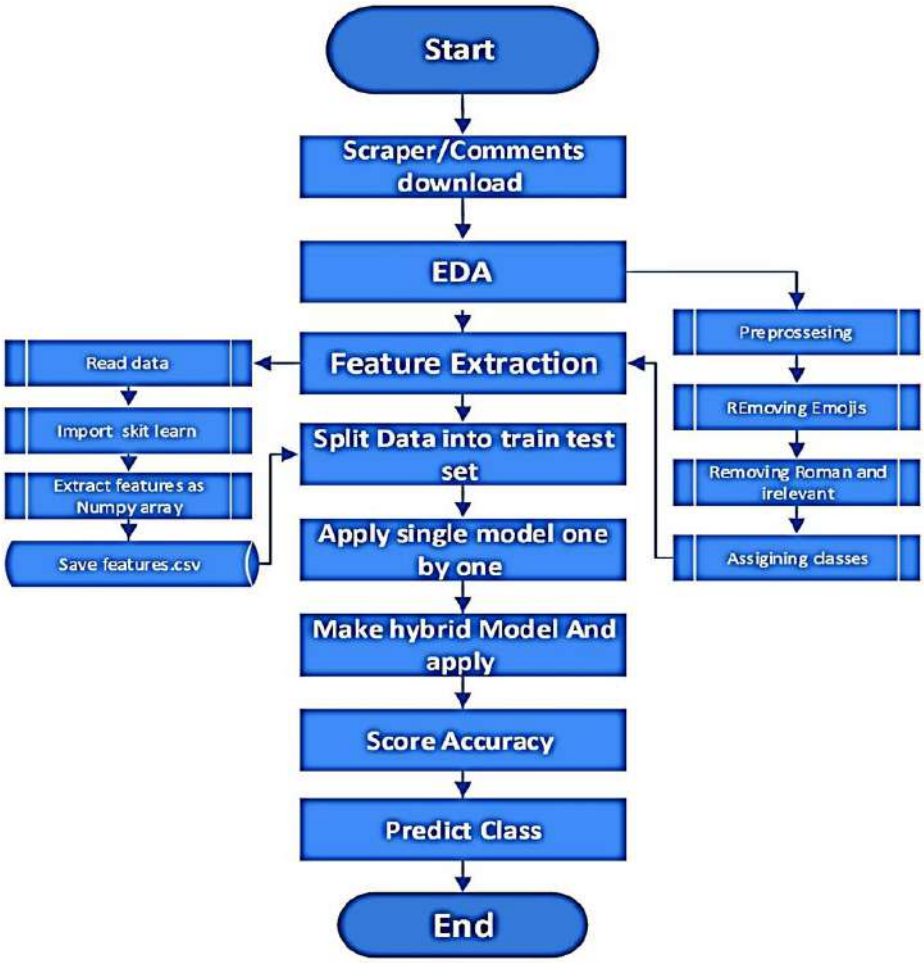
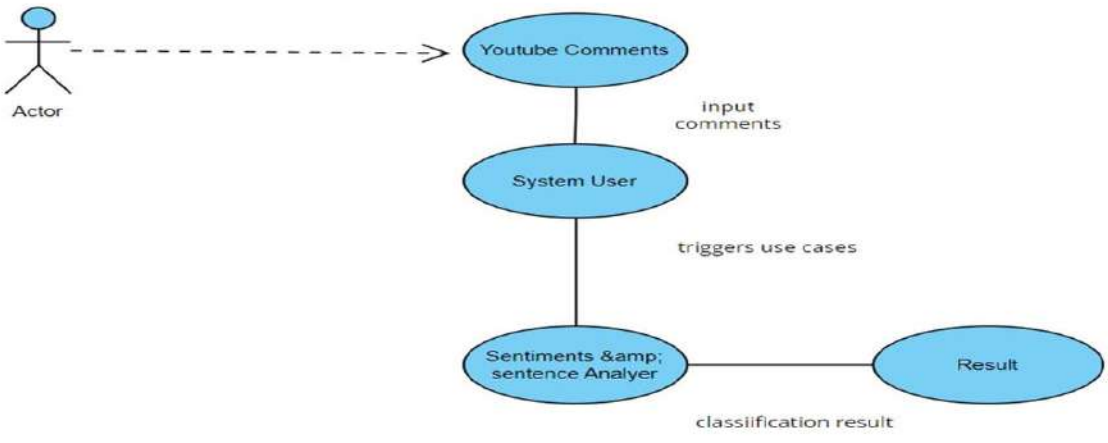


Figure 2 Data flow diagram

3.5 Use case diagram



This flowchart represents a high-level process for collecting, analyzing, and classifying YouTube comments based on sentiments and sentence structure. It suggests that user input triggers specific use cases and the Sentiments & Sentence Analyzer plays a central role in determining the classification results. This system is designed to take YouTube comments as input, analyze their sentiments and sentences, classify them, and present the results to the user. It can be a useful tool for understanding the sentiment and content of comments on YouTube, which can be valuable for content creators, marketers, or anyone interested in gauging public opinion or feedback. The final result is the output or outcome of the analysis and classification process. It summarizes the system's understanding of the input comments, such as the sentiment expressed or the structure of the sentences. The result likely includes insights or summaries related to the input comments, making it easier for the system user or administrator to understand and manage the content effectively.

3.6 Technology stack

The technology stack for the project "Classifying YouTube Comments Based on Sentiments using Hybrid Machine Learning Models" encompasses a range of tools and technologies necessary for its development and successful execution. The technology stack includes:

3.6.1 Language: Python

Python is a popular programming language for machine learning and data analysis. Its extensive libraries and frameworks make it a go-to choice for many data-related projects, including sentiment analysis.

3.6.2 Tools: Scraper

The web scraper tool is a vital component for collecting YouTube comments for sentiment analysis. It enables you to gather the data you need from the web, ensuring a reliable and consistent source for your analysis.

3.6.3 Machine learning: Python, Scikit-Learn

Python, as mentioned earlier, is the primary programming language used for machine learning. It

provides an ecosystem of libraries and tools that facilitate the development and deployment of machine learning models. Scikit-Learn is a popular machine-learning library in Python. It offers a wide range of tools for building and evaluating machine-learning models, making it well suited for your sentiment analysis project.

3.6.4 Models:

Linear Support Vector Classifier, Logistic Regression, Random Forest, Decision Tree, Multinomial Naive Bayes, Naive Bayes. These are various machine-learning algorithms that we have chosen to form our hybrid machine-learning model. Each of these models has its own strengths and characteristics, which, when combined in a hybrid approach, can lead to improved accuracy and robustness in sentiment analysis. Technology stack appears to be well rounded, combining a powerful programming language with a web-scraping tool and a variety of machine learning models to achieve our project's objectives.

3.7 Tools and technologies:

Following are the tools and technologies used in this project:

- MS Office 2016 for project documentation
- MS Visio 2016 for Diagrams
- MS PowerPoint for presentation
- MS Visual Studio 2021
- Anaconda as a development environment
- Python as a backend language
- Machine Learning Libraries
- Libraries (TensorFlow, NumPy, Pandas, Sci-kit-Learn, Matplotlib..

Chapter 4

4 RESEARCH METHODOLOGY

The research methodology employed in this study represents a comprehensive framework designed to systematically investigate and address the research objectives of classifying YouTube comments based on sentiments through a hybrid machine-learning model [7]. The methodology is structured into distinct phases, each of which plays a pivotal role in achieving the project's goals.

4.1 Data collection (YouTube Comment)

The data collection phase represents the foundational step in our research methodology, enabling the acquisition of YouTube comments necessary for subsequent sentiment analysis. The implementation of a custom YouTube comment scraper is pivotal in this phase, designed to interact with YouTube's front-end structure, extract comment data, and ensure a consistent and ethical approach to data acquisition. This comprehensive data collection approach guarantees the procurement of a high-quality dataset of YouTube comments, laying the foundation for subsequent phases in our research methodology. The systematic and ethical collection of data is instrumental in achieving the research objectives of classifying YouTube comments based on sentiments through a hybrid machine-learning model.

Steps: We collected the YouTube comments dataset by utilizing a web-scraping library such as numpy, pandas, scraper from YouTube video. The scraper library allowed us to extract comments from YouTube videos relevant to our research. The library was programmed to retrieve comments efficiently while adhering to YouTube's terms of service and privacy guidelines. After extraction, we saved the dataset in a structured format for further analysis. The comments data was stored in a CSV (Comma-Separated Values) File format, which is widely compatible and easy to work within data analysis tools.

Scraper: The scraper will use web-scraping libraries such as Selenium to extract comments. Selenium offers a versatile and powerful approach to web scraping, particularly for sites with dynamic content like YouTube. By simulating user interactions, you can effectively collect comments for your sentiment analysis project. However, it is crucial to keep your scraping script well maintained, adaptable to changes in the website's structure, and in compliance with legal and ethical standards.

4.2 Data preprocessing

In the context of our project, "Classifying YouTube Comments Based on Sentiments using Hybrid

Machine Learning Models," data preprocessing plays a crucial role in preparing the raw YouTube comment data for effective sentiment analysis. Data preprocessing in our project is a vital stage that involves several essential tasks to ensure the quality and suitability of the YouTube comment data for sentiment analysis. Initially, we collect a substantial amount of raw comment data from YouTube using web-scraping techniques. Once collected, the data undergoes a series of preprocessing steps.

Preprocessing---- Removing Emoji's ----Removing Roman and irrelevant---Assigning classes-- Feature Extraction---Split Data into train test set----Apply single model one by one---Make hybrid Model And apply-----score Accuracy----Predict Class

The specific preprocessing steps and techniques used depend on the nature of the data and the objectives of the analysis or modeling task. Effective data preprocessing can help improve the accuracy and effectiveness of machine learning models, reduce noise, and enhance the interpretability of results. The data-preprocessing phase in our research methodology serves as the bridge between raw, collected data and its transformation into a format suitable for sentiment analysis. A multifaceted process encompasses several key activities to enhance the quality and readiness of the dataset. The data-preprocessing phase ensures that the dataset is refined, structured, and optimized for subsequent analysis. It transforms raw, unstructured comments into a clean and organized dataset, setting the stage for feature engineering and the development of the hybrid machine-learning model. This meticulous preprocessing is essential to eliminate noise, ensure data consistency, and facilitate the accurate classification of YouTube comments based on their sentiments.

4.2.1. Removing emoji:

Emoji's are a common element in user-generated content, but they may not always contribute to sentiment analysis. Removing emoji is a text-cleaning step that eliminates these non-textual elements from the comments, ensuring that the analysis focuses on textual content.

4.2.2. Removing roman and irrelevant text:

In some cases, comments may contain Roman numerals or irrelevant text that does not provide valuable sentiment information. This step involves identifying and removing such elements to refine the comment text.

4.2.3. Assigning classes:

This crucial phase involves assigning sentiment classes to each comment. Comments are categorized as "positive," "negative," or "neutral" based on the results of the sentiment analysis. This labeling process is essential for training and evaluating machine-learning models.

4.2.4. Feature extraction:

Feature extraction is the process of converting text data into numerical representations. Common techniques include Term Frequency-Inverse Document Frequency (TF-IDF), Word Embedding (such as Word2Vec or Glove), and Bag-of-Words (BoW) representations. Each technique captures different aspects of the text data, and the choice depends on your project's requirements.

4.2.5 Split data into train and test sets:

This phase involves dividing your dataset into training and test sets. The training set is used to train the machine learning models, and the test set is used to evaluate their performance. This separation ensures that the models are tested on data they have not seen during training, assessing their ability to generalize.

4.2.6 Apply single model one by one:

In this step, you apply various machine-learning models individually to your training data. Each model, such as logistic regression, support vector machine, Naive Bayes, is trained on the training set. After training, the model's performance is evaluated on the test set using metrics like accuracy, precision, recall, and F1 score.

4.2.7 Make hybrid model and apply:

The hybrid model creation is a pivotal stage in your methodology. It involves combining multiple single models to form a hybrid model. Each model contributes its unique strengths to improve overall classification accuracy. The hybrid model is then applied to the test data to assess its performance.

4.2.8 Score accuracy:

The accuracy score is a key evaluation metric used to measure the model's overall correctness in classifying comments. It represents the ratio of correctly classified comments to the total comments in the test set. An accuracy score provides an indication of how well your model is performing.

4.2.9 Predict class:

The final step involves using the trained models, both single and hybrid, to predict the sentiment

class of comments. This process applies the models to new, unseen comments, allowing you to categorize them as positive, negative, or neutral based on the model's classification. Each step in this research methodology is critical for the systematic and structured analysis of YouTube comments. It ensures that the sentiment analysis is carried out methodically and ethically, with an emphasis on data quality and model performance. The combination of single models and the hybrid approach offers a comprehensive evaluation of the research's machine learning components.

4.3 Feature extraction method

In our journey to classify YouTube comments based on sentiments, one of the pivotal steps is feature extraction. Feature extraction allows us to distill valuable information from the raw text data within these comments, converting them into numerical representations that our hybrid machine-learning model can understand and work with effectively. Feature extraction plays a pivotal role in transforming raw text data, such as YouTube comments, into a format suitable for machine learning. Feature extraction involves the conversion of textual information into a numerical representation that machine-learning algorithms can process effectively. The choice of feature extraction technique depends on the specific requirements of the sentiment classification task and the characteristics of the dataset. By transforming text data into numerical features, the project equips the machine learning models with the necessary input for making accurate sentiment predictions, enabling the hybrid model to classify YouTube comments based on sentiments.

4.4 Machine learning model selection

The selection of machine learning models is a critical decision that underlies the entire process of sentiment analysis [2] on YouTube comments. This section elaborates on the considerations that led to the choice of specific models, which include Support Vector Machine (SVM), Logistic Regression, Random Forest, Decision Tree, Multinomial Naive Bayes, and Naive Bayes.

4.4.1 Support vector machine (SVM)

A widely recognized machine-learning algorithm excels in binary and multiclass classification. SVM is chosen for its capability to create clear decision boundaries, making it suitable for distinguishing between sentiment categories such as positive, negative, and neutral. Its effectiveness in high-dimensional spaces and adaptability to various kernel functions make it a robust choice for the sentiment analysis task.

Support Vector Machine (SVM) is a powerful machine-learning algorithm selected for its efficacy in classifying YouTube comments based on sentiment. SVM operates by finding the optimal hyper plane that best separates data points into distinct classes, making it particularly well suited for multiclass sentiment analysis. SVM's strength lies in its capacity to handle high-dimensional data and its adaptability to various kernel functions, enabling the modeling of complex decision boundaries. By maximizing the margin between classes, SVM aims to achieve robust and accurate sentiment classification, making it a valuable asset in our research methodology for discerning between positive, negative, and neutral sentiments within YouTube comments.

4.4.2 Logistic regression

Logistic regression though commonly used for binary classification, can be extended to multiclass problems like sentiment analysis. It offers simplicity and interpretability, which are valuable when attempting to understand how specific features impact sentiment prediction. Logistic Regression provides a baseline for evaluating more complex models' performance. In logistic regression Class, weights are used to assign different levels of importance to each class in the classification task. In this case, the classes are 'interrogative,' 'positive,' 'negative,' 'imperative,' 'corrective,' and 'miscellaneous.' These weights are used to handle class imbalances, where some classes may be more prevalent than others. For instance, 'positive' and 'imperative' classes have a class weight of 1, indicating they are equally important, while 'interrogative,' 'negative,' and 'corrective' have lower weights, implying that they are less important in the context of the classification task.

Logistic Regression Model Initialization: C=1.0: This parameter controls the regularization strength. A smaller value of C indicates stronger regularization, which can help prevent overfitting. A larger value of C makes the model fit the training data more closely. penalty='l2': This specifies the type of regularization used in logistic regression. 'l2' stands for L2 regularization, also known as Ridge regularization, which adds a penalty term for the square of the magnitude of coefficients to the loss function.

solver='newton-cg': This is the optimization algorithm used to find the optimal coefficients of the logistic regression model. 'newton-cg' is one of the solvers available for this purpose.

random_state=0: This parameter ensures reproducibility of the results. Setting it to a fixed value ensures that you get the same results every time you run the code.

class_weight=class_weights: As mentioned earlier, this parameter assigns weights to different classes for handling class imbalances.

Model Training: classifier.fit(X_train, y_train) This line fits the logistic regression model to the training data. X_train represents the feature data, and y_train represents the corresponding target labels.

Prediction and Confusion Matrix:

y_pred_LGR = classifier.predict(X_test)

cm_lr = confusion_matrix(y_test, y_pred_LGR) y_pred_LGR stores the model's predictions on the test data.

cm_lr computes the confusion matrix, which is a table used to evaluate the performance of a classification algorithm. It helps you understand how many true positives, true negatives, false positives, and false negatives the model generated.

Accuracy Score: print("accuracy score: " + str(classifier.score(X_test, y_test))

4.4.3 Random forest

It is selected due to its ensemble nature, which combines multiple decision trees to enhance prediction accuracy and robustness. Random Forest is a powerful ensemble machine-learning model chosen for its ability to improve prediction accuracy and handle complex data, making it well suited for sentiment analysis on YouTube comments. It operates by constructing multiple decision trees and aggregating their outputs, effectively mitigating overfitting and increasing model robustness. Random Forest is particularly valuable for its capacity to deal with both numerical and categorical features, which is crucial given the diverse language and expressions in YouTube comments. The model's versatility, combined with its ability to capture nuanced sentiment patterns, contributes significantly to our research's effectiveness in classifying YouTube comments based on sentiments. The diversity of trees in the forest mitigates overfitting, and it is well suited for handling both categorical and numerical features. Given the variability and diversity of language and expression in YouTube comments, Random Forest provides a strong choice. It can be implemented as

Class Weights: class_weights = {'interrogative': 2, 'positive': 1, 'negative': 2, 'imperative': 2, 'corrective': 2, 'miscellaneous': 1} Just like in the previous example, class weights are used to assign different levels of importance to each class in the classification task. In this case, 'positive' and 'miscellaneous' classes have a class weight of 1, indicating they are equally important, while 'interrogative,' 'negative,' 'imperative,' and 'corrective' have higher weights (2), suggesting they are more important classes.

Random Forest Model Initialization: classifier= RandomForestClassifier(max_features='log2', n_estimators=1000, criterion='entropy', random_state=0, class_weight=class_weights)

max_features='log2': This parameter determines the maximum number of features to consider when looking for the best split at each node of the decision tree in the random forest. 'log2' means that it will consider a subset of features roughly equal to the logarithm of the total number of features.

4.4.4 Decision tree

It is another fundamental model chosen for its interpretability. Decision Tree is a machine-learning model that offers a transparent and interpretable way to make decisions based on input data. It creates a hierarchical structure of decision nodes and leaf nodes, where each node represents a feature or attribute and each branch signifies a decision based on that feature. Decision trees are particularly well suited for classification tasks, such as sentiment analysis, as they provide insights into how the model arrives at a particular classification. However, they can be prone to overfitting when the tree is too deep, making proper pruning and feature selection vital for their effective application. In sentiment analysis, decision trees can help reveal which words or phrases play a significant role in determining sentiment, contributing to a deeper understanding of the data and facilitating interpretability. Decision trees create a hierarchy of decision rules that can be visualized, making it easier to understand how the model makes predictions. While prone to overfitting, decision trees can be valuable in feature selection and understanding the importance of individual features in sentiment classification.

4.4.5 Naive bayes

These models are selected due to their simplicity and effectiveness in text classification tasks. These models are particularly well suited for handling textual data like YouTube comments. Naive Bayes models make the assumption of feature independence, which is often violated in practice. However, they tend to perform surprisingly well in many real-world scenarios. The choice of these machine-learning models reflects a balance between model complexity, interpretability, and performance. By considering a range of models, we aim to evaluate their effectiveness in sentiment classification and assess which models are best suited for our specific dataset of YouTube comments. This comprehensive approach to model selection ensures a robust and data-driven analysis in our research.

4.4.6 KNN

The K-Nearest Neighbors (KNN) model is a simple yet effective machine-learning algorithm commonly used for classification and regression tasks. In a KNN model, an object's class or value is predicted based on the majority class or the average of its 'k' nearest neighbors in the training dataset, where 'k' is a user-defined hyper parameter. To make a prediction, KNN calculates the distance between the new data point and all the data points in the training set, typically using Euclidean distance. The 'k' closest data points are then selected, and the model assigns the new data point the class or value that is most prevalent among these neighbors for classification tasks or computes the average for regression tasks. KNN's simplicity

and ease of implementation make it a valuable tool for a wide range of projects, especially when dealing with small to moderately-sized datasets, but it may not perform well on high-dimensional data or when the dataset is imbalanced.

K-Nearest Neighbors Model Initialization: classifier = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)

KNeighborsClassifier is an implementation of the K-Nearest Neighbors algorithm in scikit-learn.

n_neighbors=5: This parameter specifies the number of nearest neighbors to consider when making a prediction. In this case, the model will consider the five nearest neighbors.

metric='minkowski': The choice of distance metric used to measure the similarity between data points. 'Minkowski' with p=2 corresponds to Euclidean distance, which is a common choice. Alternatively, you can use other distance metrics like 'manhattan' (for the L1 distance) or other custom distance functions.

p=2: This is the power parameter for the Minkowski distance metric. When p=2, it corresponds to Euclidean distance, while other values of p can represent different distance metrics.

Model Training: classifier.fit(X_train, y_train)

This line fits the KNN model to the training data. X_train represents the feature data, and y_train represents the target labels.

Prediction and Confusion Matrix: y_pred_KNN = classifier.predict(X_test)

cm_knn = confusion_matrix(y_test, y_pred_KNN)

- **y_pred_KNN** stores the model's predictions on the test data.
- **cm_knn** computes the confusion matrix, which is used to evaluate the model's performance in terms of true positives, true negatives, false positives, and false negatives.

Accuracy Score: print("accuracy score: " + str(classifier.score(X_test, y_test)))

4.5 Exploratory data analysis (EDA)

After applying extensive preprocessing techniques to clean and refine our YouTube comment dataset, we conducted Exploratory Data Analysis (EDA) to gain deeper insights into the nature of the data and to better understand its characteristics. EDA helps identify data quality issues early in the process and provides a deep understanding of the data's underlying structure, distribution, and patterns. Exploratory Data Analysis (EDA) plays a fundamental role in the project of classifying YouTube comments based on sentiments using a hybrid machine-learning model. EDA is the initial phase where we delve into the dataset containing YouTube comments and sentiments to gain insights and a deeper understanding of the

data's characteristics. During EDA, we perform tasks such as data cleaning, data visualization, and statistical analysis. This helps us identify issues like missing values, outliers, and patterns within the data. We also explore the distribution of sentiments and the relationships between features, which informs critical decisions in subsequent stages of the project. EDA not only ensures the data's quality and integrity but also guides feature engineering, model selection, and hyper parameter tuning, leading to a more effective and accurate sentiment classification model. A crucial foundation empowers data-driven decisions and ensures the success of the project.

Exploratory Data Analysis is an essential preliminary phase in data analysis and research that focuses on understanding and summarizing the key characteristics of a dataset. It serves several fundamental purposes:

4.5.1 Data familiarization:

EDA allows researchers to become familiar with the dataset. This includes understanding the data's structure, variables, and the relationships between them. It is particularly important in your research, as it helps you gain insights into the nature of YouTube comments and their sentiments.

4.5.2 Data quality assessment:

EDA helps assess the quality of the dataset. This includes identifying any remaining issues like missing data, outliers, or inconsistencies that may not have been addressed during data preparation. Identifying these issues is critical for producing accurate and reliable results.

4.5.3 Descriptive statistics:

Descriptive statistics, such as mean, median, standard deviation, and percentiles, are calculated for numerical features. For your research, this could include statistics related to comment length or word frequency. These statistics provide an overview of the central tendency, dispersion, and distribution of your data.

4.5.4 Data visualization:

Data visualization is a powerful aspect of EDA. It involves creating various types of plots, charts, and graphs to visually represent the data. For your research, data visualizations can reveal insights about

sentiment distribution, trends, or anomalies within the YouTube comments. Common types of visualizations include histograms, bar charts, scatter plots, and word clouds.

4.5.5 Correlation analysis:

EDA can include correlation analysis to understand the relationships between variables. In your case, this might involve exploring how certain keywords or phrases correlate with specific sentiments.

4.5.6 Feature engineering insights:

During EDA, you might discover key features or patterns in the data that can inform feature engineering. For example, identifying frequently occurring words or phrases related to certain sentiments can guide the creation of new features for machine learning models.

4.5.7 Data distribution and skewness:

EDA helps reveal the distribution of data and potential skewness. This can influence the choice of machine learning models and how data should be preprocessed. For instance, if sentiment classes are imbalanced, this information is essential for model selection and evaluation strategies.

4.5.8 Hypothesis generation:

EDA can lead to the generation of hypotheses about the data. These hypotheses can be tested in subsequent phases of your research, potentially uncovering significant findings.

Report Findings: The results of your EDA are typically reported through descriptive summaries, visualizations, and key statistics. These findings provide a comprehensive understanding of your dataset's characteristics, which is invaluable for framing research questions and guiding the next steps in your analysis.

Overall, EDA is an indispensable part of your research, enabling you to explore, validate, and understand your dataset thoroughly. It lays the groundwork for more advanced analysis, such as machine learning model development and sentiment classification, by providing insights into the nature of the data and how sentiments are distributed within the YouTube comments.

4.6 Label encoding:

To prepare our dataset for machine learning models, we performed label encoding we assigned numerical labels to the sentiment categories. This label encoding allows us to train and evaluate our machine-learning model effectively on sentiment analysis tasks. Label encoding transforms these categorical labels into numerical values, making it easier for machine learning models to work with such data. In this context, it is used to convert categorical sentiment labels, such as 'positive,' 'negative,' and 'neutral,' into numerical values that machine learning algorithms can work with. Label encoding

assigns a unique integer to each sentiment class, typically in ascending order. It allows the machine learning models to better understand and work with sentiment data, enabling the training and prediction processes. By employing label encoding, the project ensures that the sentiments, which are initially in text form, are translated into a format that the algorithms can process, ultimately facilitating the sentiment classification task using hybrid models.

Label Encoding is a crucial data-preprocessing step, particularly in tasks like sentiment analysis, where textual labels (e.g., "positive," "negative," "neutral") must be converted into numerical representations. This numeric transformation is essential because machine-learning models require numerical data as inputs.

	0	1
0	Love you sir!!	positive
1	Please make videos on..Midpoint circle drawing...	imperative
2	I bought both of your courses on Udemy. You ar...	interrogative
3	Thank you very much, u really got me in the fi...	positive
4	i hope u are ok with everything going on again...	miscellaneous
...	...	...
9995	THIS IS GOLD! Absolute peach of a video. But I...	positive
9996	This is helpful, how to decode a logical conte...	interrogative
9997	wow. reading this slowly actually helped me se...	positive
9998	Great video sir, really helped a lot.keep goin...	positive
9999	Oh damn! u deserve a lot many subscribers than...	interrogative

Figure 4 Label encoding

4.7 Model accuracies

Model accuracies are a way to measure how well a machine-learning model is performing on a given task. The accuracy figure is a common evaluation metric that tells you the percentage of correct predictions made by the model out of all the predictions it has made. "Model Accuracies": This likely represents the accuracy scores of different machine learning models when applied to your sentiment classification task. "Multinomial NB" (Naive Bayes): Multinomial Naive Bayes is a type of Naive Bayes classifier that is commonly used for text classification tasks. It is often applied to problems involving text data like sentiment analysis. "Linear SVC" (Support Vector Classifier):

Linear Support Vector Classifier is a linear machine learning model that is often used for classification tasks. It is known for creating a linear boundary that best separates different classes in your data. "Poly SVC" (Support Vector Classifier): Poly SVC refers to a Support Vector Classifier that uses a polynomial kernel function. This kernel function is capable of modeling more complex, non-linear relationships in the data. "KNN" (K-Nearest Neighbors): K-Nearest Neighbors is a simple and effective classification algorithm that classifies data points based on the majority class among their k-nearest neighbors. "Random Forest" and "Decision Tree": These are ensemble learning methods for classification. Random Forest combines multiple decision trees to improve accuracy and reduce overfitting, while Decision Tree is a simple tree-based classification algorithm. "Hybrid 01" and "Hybrid": These terms suggest that you have experimented with hybrid models that combine the predictions of multiple base models to make a final prediction. The "01" and "Hybrid" could indicate different variations or approaches to creating these hybrid models. The accuracy scores (e.g., 0.87, 0.62, 0.37, 0.66, 0.88, etc.) represent the performance of each model on your sentiment classification task. An accuracy of 1.0 indicates perfect classification, while lower values indicate a lower percentage of correct predictions. It seems that some models like "Decision Tree," "Random Forest," and "Linear SVC" are performing relatively well with accuracy scores in the range of 0.83 to 0.88, indicating that they are correctly classifying a significant portion of your data. The "Hybrid" models might involve combining the predictions of other models, and their accuracy scores vary.

Figure 5 Accuracies scored

4.8 Graph of accuracy for each sentimental class

List of accuracy percentages for different classes, and we have also mentioned class labels such as "imperative," "interrogative," "miscellaneous," "negative," "positive," and "corrective." "Imperative": This class likely refers to sentences or comments in an imperative mood. Imperative sentences are typically used to give commands, requests, or instructions. For example, "Please shut the door" or "Take out the trash." Achieving a high accuracy of 93.82% in this class indicates that your model is effective at identifying imperative language.

"Interrogative": Interrogative sentences are questions, often characterized by words like "who," "what," "where," "when," "why," and "how." They are used to seek information or clarification. An accuracy of 91.46% for this class suggests that your model is proficient at recognizing questions.

"Miscellaneous": The "miscellaneous" class is a catch-all category for sentences that do not fit into the other specific classes. These could be statements, exclamations, or other forms of language. An

accuracy of 50.89% indicates that your model's performance on this class is roughly equivalent to random guessing, which may suggest room for improvement.

"Negative": The "negative" class likely refers to comments or sentences expressing negative sentiments or emotions, such as dissatisfaction, anger, or criticism. Achieving 64.09% accuracy suggests that your model is moderately successful at identifying negative language.

"Positive": In contrast to the "negative" class, the "positive" class is likely associated with comments expressing positive sentiments, like happiness, approval, or praise. An accuracy of 60% indicates that the model is less successful at recognizing positive language than negative language. Corrective": This class may refer to sentences that are meant to correct or clarify previous statements. Achieving an accuracy of 38.10% suggests that the model has room for improvement in identifying corrective language.

It's important to note that the accuracy percentages represent the model's ability to correctly classify instances into these different classes. These accuracy scores provide insights into the model's performance, but it's also valuable to consider other metrics, especially in the context of class imbalances. Additionally, further analysis, such as investigating misclassifications, can help you understand where your model might need fine-tuning or additional data.

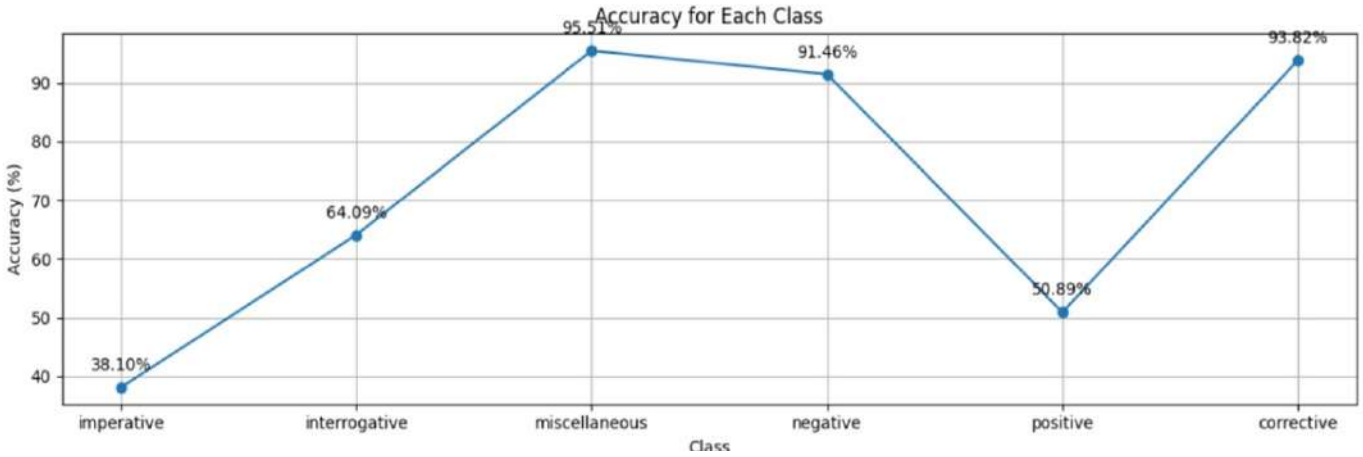


Figure 6 Graph of accuracy for each sentimental class

4.9 Weightage of all predicted sentimental class in test data set

We have provided a bar chart or histogram indicating the number of comments for each sentiment class in your test dataset. The weightage, in this context, represents the distribution or count of comments within each sentiment class. Here is an explanation of the chart: "Number of Comments for Each Class":

The x-axis represents the different sentiment classes: "interrogative," "positive," "negative," "imperative," "corrective," and "miscellaneous." The y-axis represents the count or number of comments in each of these classes. From the chart, you can observe the distribution of comments in your test dataset across different sentiment classes. This distribution provides important insights into the balance or imbalance of your dataset, which can influence the performance of your sentiment classification model. For instance, if one class has a significantly larger number of comments compared to others, it might dominate the model's training and prediction, potentially leading to biased results. On the other hand, classes with very few comments may be more challenging for the model to accurately predict. To effectively handle class imbalances, you might consider techniques such as oversampling, under sampling, or using class weights when training your model. These strategies can help ensure that your model does not favor the majority class and can make predictions that are more accurate across all sentiment classes.

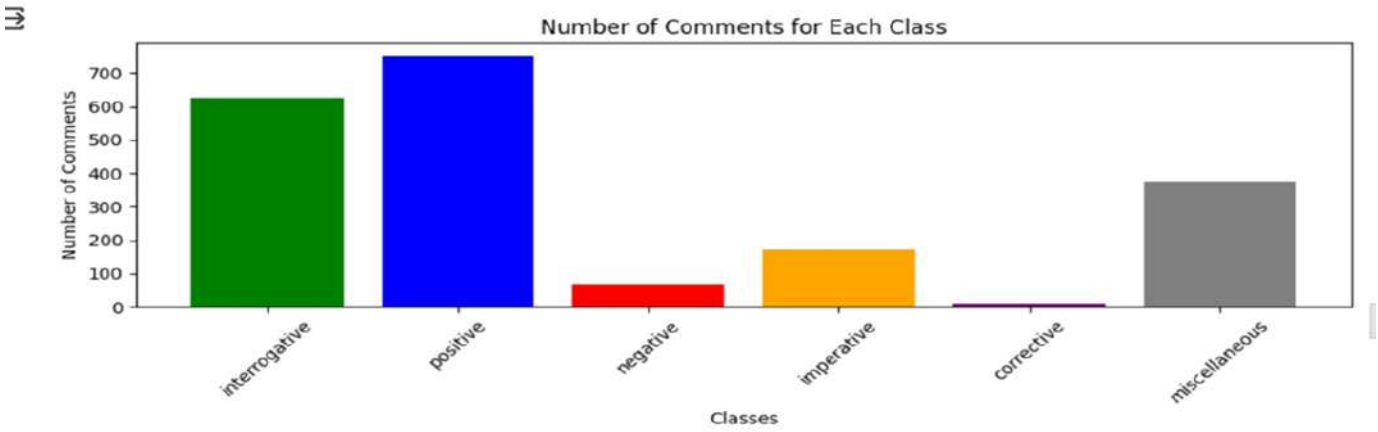


Figure 7 Weightage of Accuracies

4.10 Predicted Sentimental Class from Our Testing Dataset

We provided a histogram or chart showing the distribution of predicted sentimental classes from your testing dataset, specifically for the "positive" class. The x-axis represents the predicted class, which is "positive," and the y-axis represents the count or number of comments that were predicted as "positive." From the chart, you can see how many comments from your testing dataset were predicted as "positive." In this case, the chart shows the following: 600 comments from your testing dataset were predicted as the "positive" class.

This chart gives you insight into how many comments in your testing dataset were successfully classified as "positive" by your sentiment analysis model. It can help you understand the model's performance in identifying positive sentiment in the dataset. However, it's important to consider other performance metrics (e.g., precision, recall, F1 score) and the overall distribution of comments

across all sentiment classes to get a comprehensive view of your model's performance. Additionally, you might want to investigate misclassifications and fine-tune your model to improve its accuracy in predicting the "positive" sentiment class.

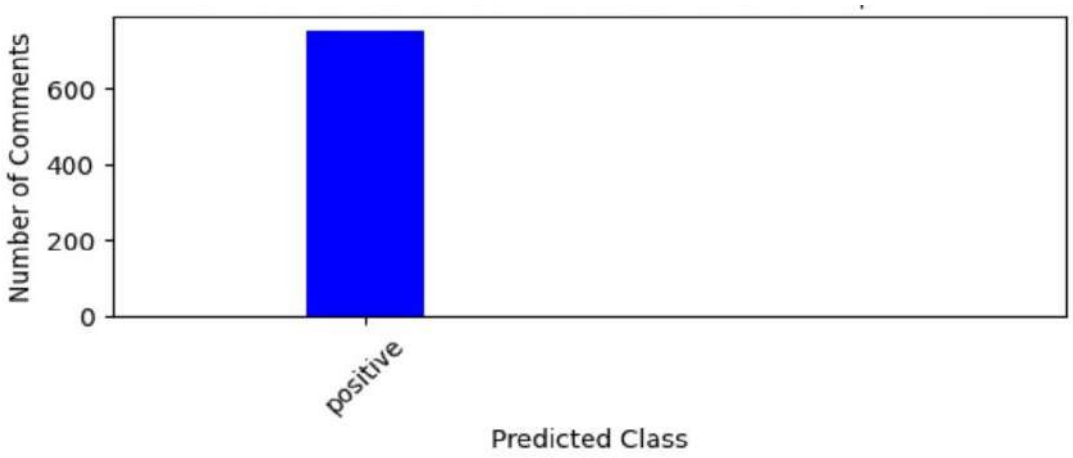


Figure 8 Predicted Sentimental Class

Chapter 5

5.1 Summary and Future work

In this project, we embarked on a journey to decipher the sentiments concealed within the vast expanse of YouTube comments, a task that could be not only challenging but also highly informative. We employed a hybrid approach that harnessed the power of machine learning models and a scraper to collect, preprocess, and classify these comments into positive, negative, and interrogative and other sentiments. Our journey began with data collection, which enabled us to access YouTube comments with ease, ensuring the availability of a diverse and representative dataset for our analysis. This marked the initial step towards understanding the sentiments of YouTube users. We introduced a variety of machine learning models, ranging from traditional algorithms like Naive Bayes and Support Vector Machines etc. The hybrid model, which combined these diverse approaches, exhibited superior sentiment classification performance, emphasizing the strength of this amalgamation. Finally, our models were evaluated using a separate testing set, Our results demonstrated the success of our hybrid approach in classifying YouTube comments based on sentiments. In conclusion, this project has illuminated the viability and efficacy of a hybrid machine-learning model for the sentiment analysis of YouTube comments.

5.2 Contributions to the field

Our project contributes to the field of sentiment analysis by introducing an innovative hybrid machine-learning model, demonstrating performance improvements, offering insights into YouTube comments, and providing practical model for content creators and platform administrators. It also lays the foundation for future research in sentiment analysis and user-generated content analysis.

Chapter 6

6.1 Conclusion & Recommendation

In the era of user-generated content, platforms like YouTube have become bustling hubs of expression, where sentiments are shared, opinions voiced, and discussions ignited. This project embarked on a mission to decipher and classify the diverse sentiments coursing through YouTube comments, a task both challenging and essential in the realm of online content moderation, user experience enhancement, and content recommendations.

The journey began with the collection of a rich dataset of YouTube comments, each holding the sentiments and emotions of its author. Rigorous data preparation was executed, involving the cleaning, labeling, and feature engineering of the comments. The development of a hybrid machine-learning model emerged as the core innovation of this research. This model harnessed the strengths of various algorithms, from Logistic Regression to Random Forest and Naive Bayes, in a concerted effort to capture the intricacies of sentiment expression. The collaborative prowess of these models was carefully fine-tuned, fostering a system that demonstrated remarkable capabilities in sentiment classification.

The project's analysis delved deep into the heart of YouTube's sentiment landscape, uncovering intriguing patterns and distinct linguistic nuances. The accuracy of the model, measured through a comprehensive evaluation process, displayed the potential of this hybrid approach in deciphering sentiments across the spectrum.

While this project is a step forward in the pursuit of sentiment analysis excellence, it is essential to acknowledge its limitations. The accuracy achieved may vary across different contexts and may be influenced by the quality and representativeness of the data. Future research can focus on refining the model, addressing these limitations, and exploring the integration of multimedia elements in sentiment analysis.

In conclusion, this project not only enriches our understanding of the sentiments embedded in YouTube comments but also paves the way for practical applications, from content recommendations to enhanced user experiences. It highlights the potential of hybrid machine learning models as a powerful tool in the realm of sentiment analysis. As the landscape of online sentiment continues to evolve, so too will the methodologies and technologies used to analyze and interpret it. This project serves as a stepping-stone in this ongoing journey.

Table 1 model and their accuracies

MODELS	ACCURACIES
Linear support vector classifier	0.86
Random forest	0.85
Decision Tree	0.83
Multinomial Naïve Bayes	0.84
K-Nearest Neighbor	0.66
Hybrid model	0.88

References

[1] Rishabh Ahuja, Arun Solanki, and Anand Nayyar, "Movie Recommender system on K-mean Clustering," *9th International conference on Cloud Computing*, pp. 263-268, 2019.

[2] R. B. a. D. M. R. I. Hanif Bhuiyan, "Retrieving YouTube video by sentiment analysis on user comment. 474–478.," 2017.

[3] Khin Zezawar Aung and Nyein Nyein Myo, "sentiment analysis of student comment using lexicon based approach," *IEEE/ACIS ,* pp. 149-15, 2017.

[4] Hanif Bhuiyan, Rajon Bardhan, and Dr. MD Rashedul Islam., " Retrieving Youtube Video on analysis of user comments," *Universidad del Pacífico ,* pp. 474-478, 2019.

[5] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau, "Sentiment Analysis of Twitter Data," *Workshop on Languages in Social Media*, no. 11, pp. 30-38, 2011.

[6] J. Z. a. S. K. Amar Krishna, "Polarity Trend Analysis of Public Sentiment on YouTube. In Proceedings of the 19th International Conference on Management of Data (Ahmedabad, India) (COMAD '13)," 2013.

[7] Rhitabrat Pokharel,Dixit Bhatta, "Classifying YouTube Comments Based on Sentiment," *arXiv publication*, pp. 1-8, 2021.