YOUTUBE COMMENT SENTIMENT ANALYSIS

Contributors
Rishika Rachel Manda (U01863142)
Yakshita Rakholiya (U01875270)
Bansari Sorathiya(U01874134)
Piyush Gupta (U01863856)
Huzefa Sadikot(U0185644)

DATA MINING (CS - 619)
Prof. Liesh Miraj
Pace University
New York
March 2023

Table Of Contents

1 Introduction **2**

2 Prior Work **3**

3 Process Flow **4**

4 All About Data **4**
4.1 Calling the API 5
4.2 More about the API 5
4.3 API Methods Used 5
4.3.1 Search 5
4.3.2 Comment threads 5
4.3.3 Video Statistics 6
4.4 Where does the data go? 6

5 Sentiment Analysis **6**
5.1 What is it? 6
5.2 Preprocessing 7
5.2.1 Tokenization 7
5.2.2 Stopwords 7
5.2.3 Stemming 7
5.3 Models and Methodologies Used 7
5.3.1 Vader 7
5.3.2 Afinn 9
5.3.3 NRC Lexicon 9
5.4 Results and Findings 10
5.4.1 Which is the Best? 13

6 Prediction Models **13**
6.1 What we Aim? 13
6.2 Data Transformation 13
6.3 Models and Methodologies Used 13
6.3.1 Linear Regression 13
6.3.2 Polynomial Regression 15
6.3.3 Long Short Term Memory 13
6.4 Results and Findings 14
6.4.1 Linear Regression 14
6.4.2 Polynomial Regression 14
6.4.3 Long Short-Term Memory 14

7 Additional Analysis **14**
7.1 Models and Features Used 14
7.1.1 Models 14
7.1.2 Feature Selection 14
7.2 Findings 15

8 Conclusion **15**

9 Future Scope **15**

Bibliography **16**

Abstract

This project aims to develop a machine learning model that can perform sentiment analysis on YouTube comments using natural language processing techniques. By analyzing the sentiment of comments, insights can be gained on how viewers perceive a particular video or channel, which can be useful for content creators and businesses.

1 Introduction

With over 2.3 billion monthly active users, YouTube is the second-largest social media platform in the world. Every day, users upload over 500 hours of video content to YouTube, making it a treasure trove of information. Analyzing the sentiment of YouTube comments is an effective way to understand how people feel about a particular video or channel. By doing so, we can get insights into how viewers perceive the content and what changes can be made to improve the content.

The sentiment analysis of YouTube comments can be challenging due to the vast amount of data that needs to be analyzed. Manual analysis is time-consuming and may not be accurate. Additionally, it can be difficult to determine the sentiment of comments that use sarcasm or irony. Therefore, we need an automated system that can efficiently and accurately analyze the sentiment of YouTube comments.

The objective of this project is to develop a machine learning model that can analyze the sentiment of YouTube comments. The model will be trained on a dataset of YouTube comments and will use natural language processing (NLP) techniques to classify the comments into positive, negative, or neutral sentiments. The model will then be used to analyze the sentiment of comments on a given YouTube channel, providing insights into how viewers feel about the content.

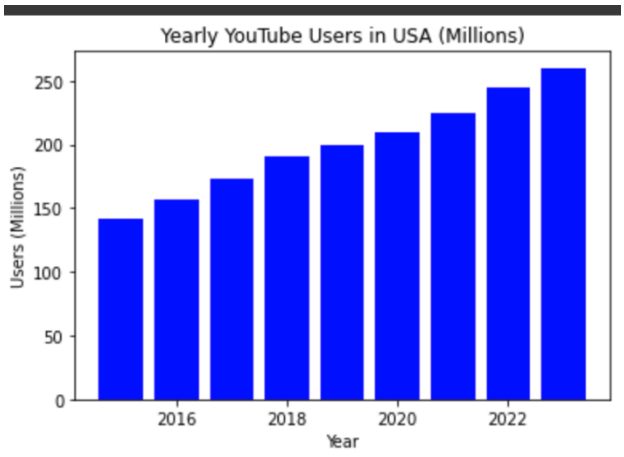

Figure 1: Yearly YouTube Users in USA

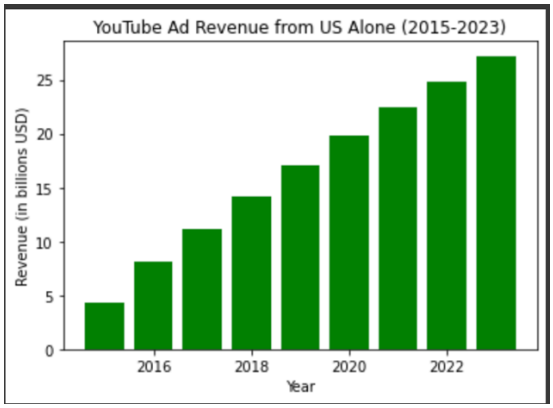

Figure 2: YouTube Ad revenue from US alone

2 Prior Work

It seems that you are citing various sources that you used to understand different techniques for sentiment analysis and time series prediction. For sentiment analysis, you referenced the paper "Survey on mining subjective data on the web" [14] as a starting point, and then went on to reference other sources such as a webpage on using VADER for sentiment analysis [12], a research paper on the ANEW word list for sentiment analysis [11], and a paper on crowdsourcing a word-emotion association lexicon [10] for understanding how the NRC Lexicon works.

For time series prediction, you referred to an article on implementing linear and polynomial regression from scratch [13] and a webpage on time series analysis, visualization, and forecasting with LSTM [9] to understand how LSTM can be used for analysis and forecasting channel sentiment trends.

It's great that you have researched and found relevant sources to help you understand these techniques better. Proper citation is important in academic work, so it's good that you have provided the necessary references for your sources.

3 Process Flow

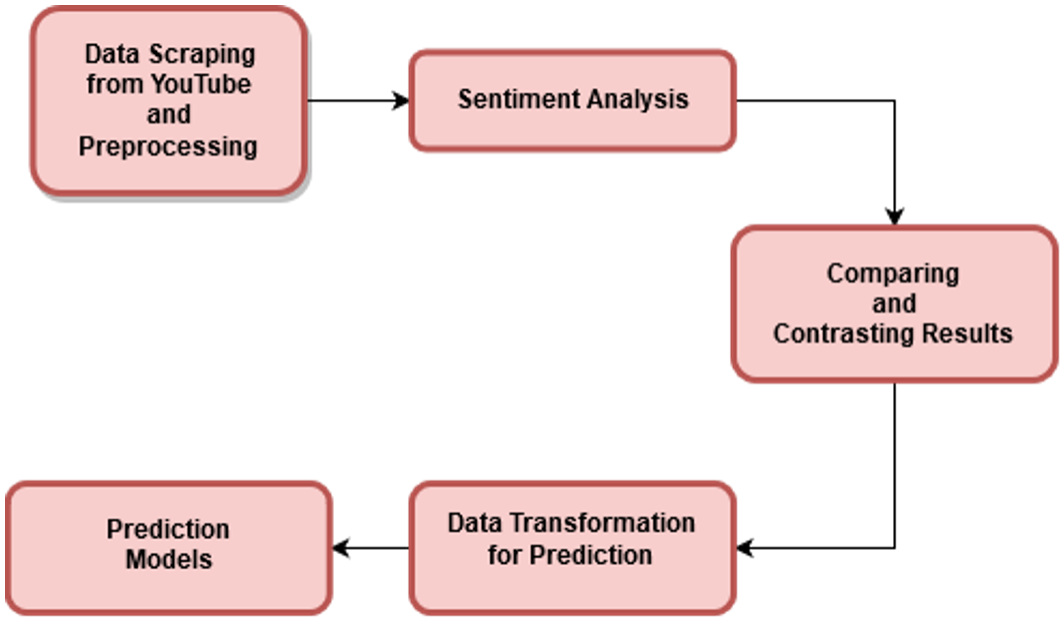

Figure 3: Project Process Flow

4 All About Data

The YouTube Data API enables you to integrate YouTube's functionalities into your own website or application and provides methods to manage resources such as inserting, updating, or deleting them [2]. OAuth authentication is used to verify users during new sessions and link their accounts to manage daily limits and access privileges.

4.1 Calling the API

To request data from the YouTube Data API, certain requirements must be met:

1. Each request should include an API key or OAuth 2.0 token, which can be obtained from the developer console.
2. An authorization token is mandatory for update, delete, or insert requests, as well as any request that accesses the user's private data.
3. The API uses the OAuth 2.0 authentication protocol.

4.2 More about the API

The YouTube Data API Version 3 has specific limits on the number of requests you can make per day, which are listed in the Google API Console where you obtain your API Key. The default quota for projects that enabled the API before April 20, 2016, is 50 million/day, while new projects have a limit of 10,000 units per day. Each API operation has a different cost in terms of units, for example, a simple read operation costs approximately 1 unit, a write operation costs around 50 units, and video upload costs around 1600 units. If you exceed your daily quota, Google will stop returning results until your quota is reset. You can request more than 1 million requests per day, but you will have to pay for the extra requests.

4.3 API Methods Used

4.3.1 Search

One of the API methods used is called "Search". Its purpose is to retrieve the channel ID from a YouTube display name provided by the user. The results are sorted by relevance and the most relevant search query result is selected. After obtaining the channel ID, another search is performed to retrieve the videos of that channel. The videos are retrieved in chronological order and the number of videos is limited to the configuration file settings. Once the videos are obtained, the comments for each video are needed. This method returns a collection of search results that match the specified API request parameters. By default, the search result set includes matching video, channel, and playlist resources, but it's possible to configure queries to retrieve only a specific type of resource [2].

4.3.2 Comment threads

This API method is used to extract comments for each video in the selected channel's video list. The API returns a maximum of 100 comments per call, and we repeat the process to obtain the desired number of comments by fetching the next page. The number of comments fetched per video can be modified in a constants file with the application configuration settings. We extract the original comment text and the date it was last updated from each returned item and create a new JSON file for both sentiment analysis and predictions.

4.3.3 Video Statistics

This API is used to obtain the latest statistics for each video ID passed as a parameter. This enables us to obtain data such as the number of comments, likes, dislikes, and views for each video, which is useful in identifying potential correlations between sentiment analysis and the video's like and dislike counts.

4.3.4 Where does the data go?

We save all the data we collect in a JSON file for each YouTube channel. This approach helps us to persist the data and avoid scraping the comments every time, which could result in exceeding the daily limit. Also, it allows us to compare different models and algorithms easily. If we reach the limit, we can resume scraping the next day where we left off and add the new data to the same JSON file. This simplifies data management, manipulation, and updating. Moreover, we can read the data into a DataFrame using pandas' readjson method.We end up creating 3 JSON files per channel

1. Video list – Contains a list of videos that we fetched initially
2. Comment Scores – Contains the score per comment that the different models give us
3. Stats – This file contains a per video stat list of all the videos in addition to the overall average sentiment for each video.

5 Sentiment Analysis

5.1 What is it?

Sentiment analysis is a technique used to determine whether a piece of writing has a positive, negative, or neutral sentiment. It combines natural language processing and machine learning methods to assign scores to different parts of a sentence or phrase. This technique is widely used by data analysts in large companies for various purposes such as understanding public opinion, conducting market research, monitoring brand and product reputation, and analyzing customer experiences. The process of sentiment analysis involves breaking down the text into different parts, identifying sentiment-bearing phrases, assigning sentiment scores to each part, and potentially combining scores for a more comprehensive analysis.Break each text document down into its component parts (sentences, phrases, tokens and parts of speech)

- Identify each sentiment-bearing phrase and component

- Assign a sentiment score to each phrase and component (-1 to +1 or -5 to +5 in some algorithms)
- Optional: Combine scores for multi-layered sentiment analysis

Sentiment analysis has multiple applications across various industries. It enables businesses to gauge how customers feel about their products, brands, or services based on online feedback and conversations. Sentiment analysis models can also detect unexpected situations and alert companies to take prompt action. However, the process of assigning sentiment scores to text can be subjective and influenced by personal experiences and beliefs. Centralized sentiment analysis systems can help companies improve the accuracy of their analysis and gain unbiased insights by using the same criteria to evaluate all data.

5.2 Preprocessing

Before feeding text data to sentiment analysis models, preprocessing techniques are applied to clean and structure the unstructured text. These techniques help in reducing noise from high dimensional features and obtaining more accurate information from the text in a low dimensional space.

5.2.1 Tokenization

Tokenization is the method of breaking down a string of characters into smaller units called tokens. This process also involves removing unnecessary characters such as punctuation marks and special characters. Emoticons are also excluded in some sentiment analysis algorithms as they can affect the accuracy of the analysis.

5.2.2 Stopwords

Stopwords refer to the words that are frequently used in the text but do not add any significant meaning to the context. These words are eliminated from the text prior to sentiment analysis as they do not contribute to the overall understanding of the text.

5.2.3 Stemming

Stemming involves reducing words to their base form, regardless of whether the resulting stem is a valid word or not in the language. This is done by removing inflections from words. The NLTK Python library offers a comprehensive set of tools for performing stemming, tokenization, and stop word removal on text data.

5.3 Models and Methodologies Used

5.3.1 Vader

VADER, which stands for Valence Aware Dictionary and Sentiment Reasoner, is a tool for sentiment analysis that is specifically designed to analyze sentiments expressed in

social media. It uses a combination of sentiment lexicon, which is a list of lexical features such as words that are labeled according to their semantic orientation as either positive or negative.

VADER has been successful in analyzing sentiment in social media text such as YouTube comments, Tweets, and Facebook posts, even in cases where the text is unstructured or contains slang, emoticons, or punctuation.

VADER's success is attributed to its ability to judge the positivity or negativity of sentiment in such text. The lexicon used by VADER rates positive words more positively and negative words more negatively.

Word	Sentiment rating
tragedy	-3.4
rejoiced	2.0
insane	-1.7
disaster	-3.1
great	3.1

VADER obtains most of its ratings from Amazon's Mechanical Turk, a cost-effective and quick method. It evaluates each word in a sentence by checking if it is present in the lexicon. For instance, in the sentence "The food is nice and the atmosphere is good," the words "good" and "nice" are found in the lexicon with ratings of 1.9 and 1.8, respectively. VADER uses these word ratings to generate four sentiment metrics, which categorize each sentence as positive, negative, or neutral. In the example given, the sentence is rated as 55% neutral, 45% positive, and 0% negative. The fourth metric is the compound score, which represents the total of all the ratings (1.9 and 1.8 in this instance) and is normalized to a score between -1 and 1. The sentence in the example has a compound rating of 0.69, which indicates a strongly positive sentiment.

Word	Sentiment rating
Positive	0.45
Negative	0.55
Neutral	0.00
Compound	0.69

5.3.2 Afinn

AFINN is a sentiment lexicon that rates English words for valence, with values ranging from -5 (negative) to +5 (positive).

YouTube comment sentiment analysis using AFINN involves tokenizing comments, scoring each word using the AFINN lexicon, and then calculating the sentiment score for the comment by summing up the valence values of the words.

The sentiment score is then normalized to a range between -1 and 1, with -1 being the most negative sentiment and 1 being the most positive sentiment.

The sentiment scores can be interpreted to understand the sentiment of the YouTube comments, with scores close to 1 indicating positive sentiment, scores close to -1 indicating negative sentiment, and scores close to 0 indicating neutral sentiment.

YouTube comment sentiment analysis using AFINN can provide valuable insights into how viewers feel about a particular video or topic.

5.3.3 NRC Lexicon

The NRC Emotion Lexicon is a list of English words rated for their association with eight basic emotions: anger, fear, anticipation, trust, surprise, sadness, joy, and disgust.

YouTube comment sentiment analysis using the NRC lexicon involves tokenizing comments, scoring each word using the NRC Emotion Lexicon, and then calculating the sentiment of the comment by summing up the number of words associated with each of the eight emotions, as well as positive and negative sentiment.

The sentiment scores can be normalized to a range between -1 and 1, where -1 is the most negative sentiment and 1 is the most positive sentiment.

The sentiment scores can be interpreted to understand the sentiment of the YouTube comments, with a high score for joy indicating positive sentiment, and high scores for anger or disgust indicating negative sentiment.

Using the NRC Emotion Lexicon to conduct sentiment analysis on YouTube comments can provide a more nuanced understanding of the emotional content of the comments, but it requires more computational resources compared to AFINN, and the lexicon may not accurately capture the sentiment of all comments.

5.3.4 Which Model is Better

Afinn: Afinn is a lexicon-based sentiment analysis tool that uses a list of words with positive and negative sentiments to analyze the sentiment of a text. It assigns a score to each word in the text based on the sentiment it conveys, and then calculates the sentiment score for the entire text based on the sum of these scores. Afinn is relatively simple and easy to use, but it may not perform as well as other tools in some cases.

Vedar: Vedar is a rule-based sentiment analysis tool that uses a set of rules to analyze the sentiment of a text. The rules are based on patterns and relationships between words, and they are designed to capture more complex aspects of sentiment than simple positive and negative word lists. Vedar is more complex than Afinn and may require more expertise to use effectively, but it can be more accurate in some cases.

NRC Lexicon: The NRC Lexicon is a lexicon-based sentiment analysis tool that uses a list of words with eight different emotions (anger, fear, anticipation, trust, surprise, sadness,

joy, and disgust) to analyze the sentiment of a text. It assigns a score to each emotion for each word in the text, and then calculates the sentiment score for the entire text based on the sum of these scores. The NRC Lexicon is more complex than Afinn but may be more accurate in capturing the nuanced emotions in a text.

Overall, the choice of sentiment analysis tool will depend on the specific needs and goals of the project. Afinn may be a good choice for simple sentiment analysis, while Vedar and the NRC Lexicon may be better suited for more complex sentiment analysis tasks.

Why *VADER* outperforms?

Developing lexicons is a time-consuming and costly process, which means that they are rarely updated to include new slang and expressions.

VADER's sentiment analysis algorithm relies on several key factors to accurately predict the sentiment of a text, including:

- Punctuation: The use of exclamation marks can increase the intensity of the sentiment without changing its meaning.
- Capitalization: Using uppercase letters to emphasize sentiment-relevant words can also increase the sentiment's intensity.
- Degree modifiers: These intensifiers impact sentiment intensity by either increasing or decreasing it.
- Conjunctions: Words like "but" can signal a shift in sentiment polarity, with the sentiment following the conjunction becoming dominant.
- Preceding Tri-gram: Examining the three words preceding a sentiment-laden feature can help identify cases where negation flips the polarity of the text.

5.1 Results and Findings

We conducted our project on several YouTube channels with varying genres and data sizes. To analyze the results, we collected around 500,000 comments from three channels, namely Dude Perfect, I Hate Everything, and Unbox Therapy. We then compared the sentiment analysis outcomes produced by Vader for these channels to examine our findings.

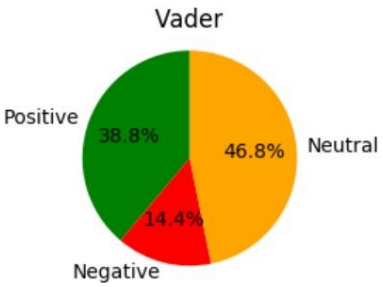

Figure 4: Dude Perfect

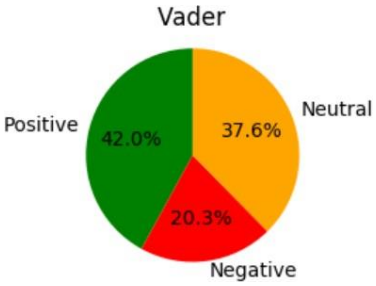

Figure 5: I hate everything

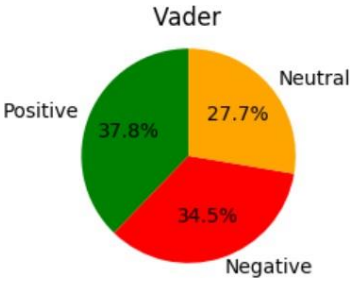

Figure 6: Unbox Therapy

The pie charts above show that Unbox Therapy has the highest number of positive comments, while Dude Perfect and I Hate Everything have similar numbers of positive comments. However, I Hate Everything has a significantly higher percentage of negative comments, around 35%, compared to 14% and 20% for Dude Perfect and Unbox Therapy, respectively. One of the reasons for selecting these particular channels is that their video content is very distinct, which is evident from the word clouds displaying the most frequently used words in the comments on these channels.

WordCloud

6. Prediction Models

6.1 What we Aim?

The objective of our project is to predict the future sentiment of a YouTube channel in order to assist channel owners in modifying their content to meet the viewers' expectations. Time-series data is essential for this prediction. We plan to use this data to train different machine learning and neural network algorithms. To create the required dataset, we carried out a data transformation, which will be explained in detail in the subsequent section.

6.2 Data Transformation

Transformation to Time-Series dataset

The comments and their corresponding sentiment analysis scores have been grouped by dates to form a time-series dataset. This dataset will be used for making predictions about future trends using both simple Machine Learning algorithms and Neural Networks.

6.3 Models and Methodologies Used

6.3.1 Linear Regression

Linear Regression is a fundamental Machine Learning Algorithm that establishes the relationship between independent variables and dependent variables. The equation of linear regression, represented as

$$Y = \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3 + .. + \delta_n x_n + E$$

(

1) consists of the dependent variable Y, independent variables x1, x2, x3...xn, weights δ1, δ2, δ3...δn and an unobserved random error E. As the equation has a degree of 1, it always plots a straight line. However, in certain scenarios such as stock market predictions, linear regression may not be effective, which is why we improve this condition by increasing the degree of the polynomial.

6.3.1 Polynomial Regression

Polynomial regression is a regression model where the equation's degree is greater than 1, in contrast to linear regression. It can fit a nonlinear relationship between x and the corresponding conditional value of y, denoted by E(y|x), which is used to describe nonlinear phenomena. Although it is a nonlinear model, polynomial regression is linear as an estimation problem, as the regression function E(y|x) is

linear in the unknown parameters that are estimated from the time series data. Therefore, polynomial regression is considered a special case of multiple linear regression.

In most cases of polynomial regression, we model the dependent variable y as an nth degree polynomial of the independent variable x, which leads to the basic equation for the polynomial regression model:

$$Y = a_0 + a_1x + a_2x^2 + a_3x^3 + .. + a_nx^n + E \quad (2)$$

From the estimation point of view, this equation is also considered linear because the regression function is linear with respect to the unknown variables a1, a2, ..., an. We also consider x1, x2, ..., xn as independent variables in this type of regression.

6.3.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) is an advanced type of Artificial Neural Network that is recurrent in nature. Unlike Recurrent Neural Networks (RNNs) that are used for short-term dependencies, LSTM is designed to handle long-term dependencies. This makes it a popular choice for predicting stock market trends in various research projects, including the one described in this paper. LSTM is particularly useful for stock market prediction as it requires analyzing large amounts of data and depends on the long-term history of the company. LSTM calculates errors by using RNNs that have a long-term memory, which contributes to its high accuracy rates. To better understand LSTM, it has a remembering cell, an input gate, an output gate, and a forget gate. The cell is responsible for remembering long-term information, while the gates regulate the flow of information in and out of the cell.

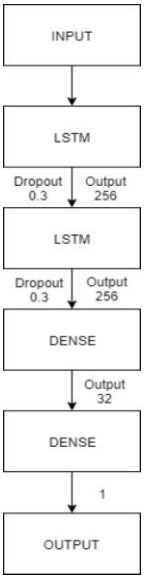

Figure XVI: Illustration on Long Short Term

7.1 Models and Features Used

7.1.1 Models

1. MLP Regressor - MLP Regressor is an artificial neural network that has at least three layers: input, hidden, and output. The nodes in the network, except for the input nodes, are neurons that use a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. This type of network is useful for solving complex problems like fitness approximation in research.
2. Keras - Keras is a deep learning framework that provides two built-in model types: sequential models and functional API models. The sequential class of models is used to create linear layers. The algorithm's performance was improved by using 10 rounds of K-fold cross-validation. Additionally, custom models with their own forward-pass logic can be created.

7.1.2 Feature Selection

We are investigating the relationship between the sentiment of comments and the likes/dislikes a video receives. To do this, we collected the statistics of every video and performed sentiment analysis using three different models. We combined the results of these models with other important video features like view count, comment count, and upload date. Then, for each sentiment analysis model, we combined the analysis with the video stats.

We used the three prediction models mentioned earlier to predict the ratio of likes/dislikes using the comment count, view count, and sentiment of the comments. We split the data into train and test sets, with 70% used for training and the remaining for testing the models. The results were interesting and will be discussed in the next section.

7.2 Findings

Taking the sentiment score output by each of the 3 algorithms we used in addition with the video view count and comment count, we tried to predict the like/dislike ratio of the video. Below were the results we observed.

Sentiment Analysis Algorithm	Neural Network	RMSE
Vader	MLP Regressor	44.31
Vader	Keras Sequential Regressor	24
Afinn	MLP Regressor	43.90
Afinn	Keras Sequential Regressor	20
NRC	MLP Regressor	42.13
NRC	Keras Sequential Regressor	18

8 Conclusion

The project aims to create a tool for YouTube channel owners, content creators, and marketers to gauge viewer sentiment
The sentiment analysis system will be trained on a large and diverse dataset of YouTube comments, including sarcasm or irony

The project will provide a user-friendly web application for real-time sentiment analysis

The application will provide a visual representation of the sentiment of comments for identifying trends and patterns

The sentiment analysis project has the potential to revolutionize YouTube channel management and optimization

Valuable insights into viewer sentiment can help improve engagement and retention, increase subscriber base, and grow businesses or brands.

9 Future Scope

In this study, we utilized the YouTube API v3 to extract data from YouTube. However, the API has its limitations and is not sufficient for analyzing sentiments or predicting the future response of a YouTube channel. For this purpose, the YouTube Analytics API can be used, which allows us to retrieve historical data for a particular channel, such as subscriber count, like/dislike count, and average playback time of videos.

By training machine learning models based on past trends, we can predict the future response of the channel in terms of subscriber gains or an increase in popularity. This information can be useful for channel owners to gauge the perception of their content by viewers and identify future trends. By combining average playback time with comments, likes, and dislikes, we can also attempt to predict the monetary gains a channel may receive.

Bibliography

[1] 40 YouTube stats and facts to power your 2020 marketing strategy : https:// sproutsocial.com/insights/youtube-stats/.

[2] YouTube Data API Overview — Google Developers: https://developers.google. com/youtube/v3/getting-started.

[3] Sentiment Analysis Explained: https://www.lexalytics.com/technology/ sentiment-analysis.

[4] Sentiment analysis of reviews: Text Pre-processing : https://medium.com/ @annabiancajones/sentiment-analysis-of-reviews-text-pre-processing-6359343784fb.

[5] Using VADER to handle sentiment analysis with social media text : http://t-redactyl.io/blog/2017/04/ using-vader-to-handle-sentiment-analysis-with-social-media-text.html.

[6] Multilayer perceptron : https://en.m.wikipedia.org/wiki/Multilayer_ perceptron.

[7] About Keras Models : https://keras.rstudio.com/articles/about_keras_ models.html.

[8] Bart Jongejan and Hercules Dalianis. Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 145–153. Association for Computational Linguistics, 2009.

[9] Susan Li. Time series analysis, visualization forecasting with lstm, May 2019.

[10] Parul Pandey. Simplifying sentiment analysis using vader in python (on social media text), Nov 2019.

[11] Chris Tam. Implementing linear and polynomial regression from scratch, Apr 2020.

[12] Mikalai Tsytsarau and Themis Palpanas. Survey on mining subjective data on the web.

[13] Data Mining and Knowledge Discovery, 24(3):478–514, 2012.