YouTube Comments Sentiment Analysis

EXECUTIVE SUMMARY

In today's dynamic world of YouTube content creation effectively understanding and responding to viewer feedback is challenging and at the same time, it is the key to know more about viewer engagement. Our project directly addresses this challenge by simplifying how content creators process and learn from the vast number of comments their videos receive.

We've built a system that efficiently collects comments from a variety of YouTube videos and analyzes them to determine overall sentiments. The sentiment could be positive, negative, or neutral. This process not only tells what viewers truly think about the content but also pinpoints trends and preferences, providing clear and actionable insights. This means creators can quickly see which parts of their videos are interesting for the audience and which parts might need improvement.

Our tool dramatically reduces the time creators spend scrolling comments which will enable them to focus more on enhancing their content's relevance. The interface of our system is straightforward and easy to use, ensuring that even those with little technical knowledge can benefit from our in-depth analysis with ease.

Hence, our project makes it easier for creators to understand what their viewers think. This tool is key for anyone who really wants to connect with their audience. It helps creators to not just answer back to viewers comments but will also help engage with their viewers which is important if they want to stand out online.

Table of Contents

1. Project Description 2
2. Problem Description 2
3. Data Exploration and Preprocessing 3
   3.1 Dataset Source 4
   3.2 Dataset Description 4
   3.3 Data Cleaning & Processing 4
4. Exploratory Data Analysis 5
5. Modelling 6
   5.1 VADER 6
   5.2 Random Forest 7
   5.3 Naïve Bayes 8
   5.4 SVM 9
6. Model Evaluation 9
7. **Error! Bookmark not defined.** 10

References 11

1. Project Description

Our project aims to reorganize how content creators interact with and understand their viewer's feedback on YouTube. We recognize that comments on videos are rich with insights about the viewer's sentiments and preferences, but manually going through them is hectic due to their large volume. Our objective is to extract meaningful information from these comments and giving a quick grasp of general opinions, sentiments and trends without the need to read each one individually to the creator.

To achieve this, we collected comments from various YouTube videos and parallelly analyze them to detect common words and phrases which will then help identify whether the sentiments expressed by the viewers are primarily positive, negative, or neutral for the video. This analysis will help discover what audience's sentiments while watching the content are.

Hence the approach is to offer content creators a clear and brief overview of viewer sentiments about the videos which will enable them to alter their videos and content more effectively according to audience preferences. Through our insights the creators can easily analyze viewer satisfaction and engagement. Ultimately this project will provide a practical and efficient tool for creators to connect viewer feedback for improved content creation.

2. Problem Description

On YouTube, each video generates thousands of viewer comments which represent a large amount of feedback that could be very huge for content creators to improve their work and engagement with their audience effectively. The vast number and diverse comments can be challenging in many ways.

1. Volume of Data: The complete quantity of comments on popular videos makes it impossible for content creators to manually read through each one.

2. Variability in Quality: Comments on YouTube vary greatly in terms of relevance and constructiveness. Some comments may be very insightful and some doesn't hence it is a complex task and requires more than just a manual review.

3. Time Sensitivity: YouTube content creators operate in a highly dynamic environment where trends change quickly. Manual analysis of comments is not feasible in this fast-paced scenario.

4. Emotional Nuance: Understanding the tone and emotional context of comments is another challenge. Comments can contain sarcasm or jokes that might be misinterpreted without proper contextual analysis.

5. Diverse Audience: YouTube is a global platform which means that viewers come from a wide range of cultural backgrounds this diversity can complicate the understanding of comments.

Our project aims to tackle these challenges by using a systematic approach that makes the analysis of YouTube comments possible by summarizing and categorizing sentiments expressed in the comments. We provide content creators with actionable insights of their audience.

3. Data Exploration and preprocessing

Before comments could be analyzed, the data collected from YouTube had to be organized and cleaned. This process helps us ensure that the data we use for our research is accurate and relevant and consistent. We collected our data using YouTube API over a period of 3 weeks and then went through several steps to prepare it for analysis. The data collected is stored in MongoDB Atlas which is our secondary data source. Our primary data source during analysis is YouTube API. We merge both data source into a data frame before we start preprocessing. This preparation includes organizing the data into a structure as the API data can be inconsistent and not necessarily useful, refining any issues and organizing them in a way that our analytics algorithm can understand.

3.1. Dataset Source

We used the YouTube Data API v3 to fetch data from YouTube, which allowed us to obtain a number of different types of information about YouTube videos and the comments left on them. Our MongoDB stores data in two separate collections namely videos to store data related to a video and another comment collection which stores data detailed to comments like Video ID, comment and likes and replies count on the comment. We specifically viewed content from popular videos at fixed intervals in order to obtain interactions that viewers have made.

3.2. Dataset Description

Dataset is divided into 2 parts, videos collection stores data about the video including Video ID, category of the video, title, tags, likes and views, while Comments collection stores data related to the video and the comments like Video ID, comment, likes count and replies count. The videos collection has 8k documents while comments collection has 100k documents in Mongo. which merges with the YouTube data fetched from API during analysis.

3.3. Data Cleaning and Preprocessing

We took several steps to prepare our data for analysis.

- **Text Cleaning:** Cleaning punctuation and special characters. Lower casing to maintain consistency in text. Removing URLs from the comment, redirecting to irrelevant sites or videos.
- **Tokenization:** We broke down the statement into smaller pieces—first into sentences, then into words. This helps to analyze the structure and content of the text.
- **Text normalization:** We performed Lemmatization to reduce the words to its root form. For e.g., changing 'running' to 'run', to ensure consistency in our analysis
- **Noise removal:** We removed parts of the comments that weren't helpful for analysis, such as emojis and links to websites.

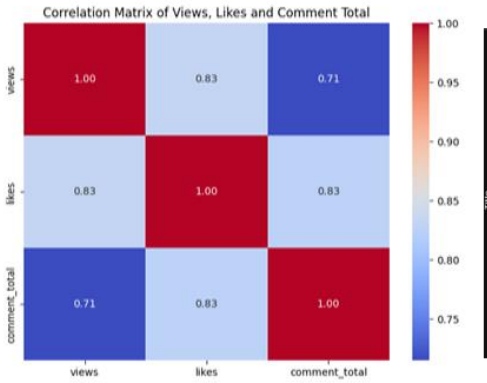
- **Stop words removal:** We removed common words like "the" and "is", which don't add much value when trying to understand them
- **Handling Negations**: This is handled in our Vader analysis for example “not happy” is handled as negative

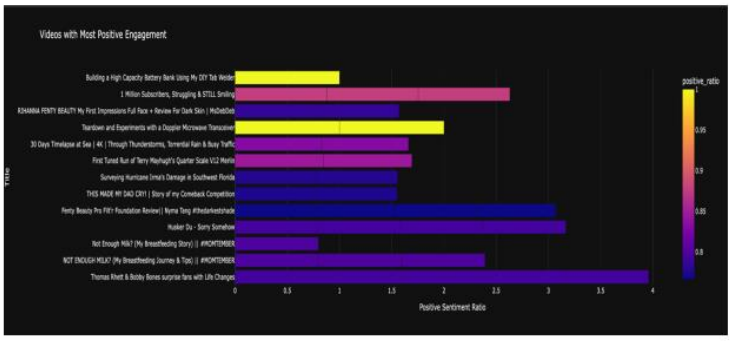
A portion of these comments was also manually labelled as positive, negative, or neutral to help us train our models to recognize these sentiments automatically in new comments.

4. Exploratory Data Analysis

Once we had our records cleaned and ready, we moved on to what’s called exploratory facts evaluation, or EDA for brief. This part of our challenge was all about looking into the facts to discover patterns, trends, or interesting functions that could inform us something valuable about the emotions in YouTube remarks. During this segment, we used easy visualization equipment to assist us see what turned into taking place in the facts:

- **Word Clouds:** These are visual representations that display us the most commonplace phrases discovered inside the remarks. The bigger and bolder a phrase seems, often it becomes referred to through visitors. We made unique phrase clouds for nice, negative, and impartial feedback to see what phrases had been maximum associated with every sentiment.
- **Frequency Analysis:** We counted how often sure phrases seemed across all remarks and inside every class of sentiment. This helped us discover the maximum popular subjects and feelings expressed via the audience.





These visualizations were not only beneficial in the know-how of the overall temper of the feedback but additionally in spotting which topics had been maximum enticing or difficult for viewers. This insight is mainly beneficial for content creators who need to decorate the attraction of their films or deal with viewer concerns.

Our exploratory evaluation was vital as it set the level for extra targeted analysis later on. By understanding the huge traits and commonplace topics, we had been better organized to use extra complicated techniques to degree sentiments correctly.

5. Modelling

The sentiments expressed in the YouTube comments were categorized using VADER and three different ML algorithms, each designed to tackle this problem in a different way.

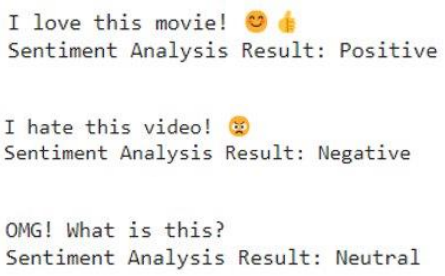
We used VADER's lexicon-based technique, which is best for social media text, to effectively analyse the sentiments in YouTube comments. VADER provides a quick sentiment score, for later on sentiment classification. To improve sentiment prediction accuracy even more, we used three machine learning models. We wanted to compare the sentiment analysis performance of both VADER and machine learning models by preprocessing the YouTube comments, converting them into numerical features using TF-IDF Vectorization, and training the models with labelled sentiment data. Finding the best method for categorizing sentiments in YouTube comments was made easier by using charts or graphs to visualize the sentiment analysis results and provide a clear picture of the sentiment distribution.

5.1. VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER has been designed to detect emotional expressions on social media. It’s very good at understanding things like slang, emojis, and exclamations, something that's common in YouTube comments. For instance, VADER is aware that the symbol ":-)" is positive while the symbol "hate" is bad.

With its pre-built sentiment lexicon, VADER rates each word in a given text according to its intensity and polarity (positive, negative, or neutral). We calculated sentiment scores, or more precisely compound scores, for every remark in the US Comments Data Frame by utilizing VADER. Based on predetermined thresholds, we were able to categorize each comment's emotion as Positive, Neutral, or Negative due to these compound scores.

We evaluated the sentiment classification distribution to determine the proportion of positive remarks. We deduced that the bulk of comments fell into the negative or neutral categories if the percentage of positive comments is zero, indicating a lack of optimism in the dataset. Additionally, to improve analysis, we mapped positive remarks to the names of the associated channels, showing which channels received favourable input. Along with that, VADER also analyzes the emoji present in the comments. It also considers the frequency of emojis. For e.g., if there are 4 happy emoji then that comment is more impactful than a comment with a single happy emoji.



	video_id	Positivity	Positive_Percentage	Channel
0	XpVt6Z1Gjjo	33.17	33.17	Logan Paul Vlogs
1	cLdxuaxaQwc	37.06	37.06	PewDiePie
2	WYYvHb03Eog	34.00	34.00	The Verge
3	sjlHnJvXdQs	37.33	37.33	jacksfilms
4	cMKX2tE5Luk	42.17	42.17	A24

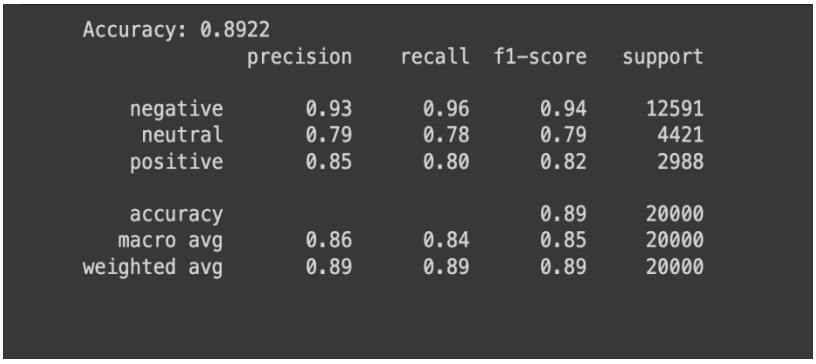
In the above screenshots, we can see that VADER has correctly identified the sentiments behind the comments as positive/ negative/ neutral.

5.2. Random Forest

Random Forest is a classifier which consists of group of decision trees to predict sentiments. Each tree makes prediction based on input text-features from YouTube comments using TF-IDF vector. TF-IDF (Term Frequency - Inverse Document Frequency) vector evaluates the word importance with the comments. Random forest predicts model accuracy for the comments based on sentiment labels (positive, negative, neutral). A new comment is converted into a TF-IDF vector which is passed down each tree in the forest. At each node of the tree, the comment is compared

to the split condition based on a particular feature and a threshold value. Each leaf node in the forest "votes" for a sentiment class (positive, neutral, or negative). The final sentiment prediction for the comment is determined by a majority vote from all the trees in the forest. The class with the most votes wins.

Below is the performance metrics achieved by training the random forest model



Accuracy: 0.8922

	precision	recall	f1-score	support
negative	0.93	0.96	0.94	12591
neutral	0.79	0.78	0.79	4421
positive	0.85	0.80	0.82	2988
accuracy			0.89	20000
macro avg	0.86	0.84	0.85	20000
weighted avg	0.89	0.89	0.89	20000

The model achieved an accuracy of 89%, which shows the percentage of comments where the model's predicted sentiment label matched the actual sentiment label.

A precision of 0.93 for negative sentiment indicates that out of all the comments the model classified as negative, 93% were truly negative. Recall reflects the sentiment labels identified by the model. F1 score shows a balance between precision and recall.

Random Forest leverages the power of multiple decision trees to analyze the features extracted from YouTube comments and predict their sentiment effectively.

5.3. Naïve Bayes

Naive Bayes is an ML Algorithm which is a probabilistic ML classifier. Naive Bayes algorithm uses Bayes theorem to calculate probability of a feature belonging to a particular sentiment class. It's called "naive" because it assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. It calculates the probability of each word in the comment appearing in a specific sentiment class and multiplies those probabilities together. The sentiment class with the highest probability is assigned to the comment. Naive Bayes is particularly efficient for large datasets and it's relatively simple to implement.

Below are the results we got from training the naive bayes algorithm. We've achieved an accuracy of 74%.

```
Accuracy: 0.74175

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.55      0.67      3616
           1       0.83      0.64      0.72      7754
           2       0.67      0.91      0.77      8630

    accuracy                           0.74     20000
   macro avg       0.79      0.70      0.72     20000
weighted avg       0.77      0.74      0.74     20000
```

5.4. Support Vector Classifier (SVC)

Support Vector Classifier (SVC) is an ML algorithm used for both classification and regression. SVC Algorithm works in 3 stages: Feature Extraction, Training and Prediction. Text data of YouTube comments is pre-processed and transformed into feature vectors using TF-IDF Vectorization, Word Embeddings, or Bag-of-Words. The pre-processed feature vectors and corresponding sentiment labels (Positive, Neutral, Negative) are used to train an SVM classifier. SVM finds the hyperplane that separates the feature vectors into different classes. SVM classifier is used to predict the sentiment label for new or unseen YouTube comments based on their feature vectors. The predicted sentiment labels are used to assess the performance of the SVM model.

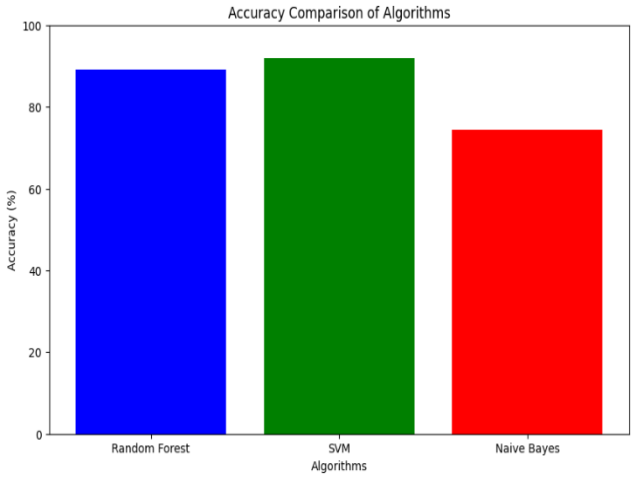
```
Accuracy: 0.91725

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.78      0.83      3616
           1       0.93      0.96      0.94      7754
           2       0.92      0.94      0.93      8630

    accuracy                           0.92     20000
   macro avg       0.91      0.89      0.90     20000
weighted avg       0.92      0.92      0.92     20000
```

6. Model Evaluation



Once our models are created, we need to check how these models are expressed in the YouTube comments. VADER picked up on the smallest details in sentiment. The SVC model gave us clear boundaries between different sentiments. Naive Bayes and Random Forest dealt with big and diverse dataset to give strong and reliable solutions.

7. Summary

Understanding and reacting to comments left by viewers on YouTube videos is a challenge that is taken on by the "YouTube Comments Sentiment Analysis" project. Today's fast-paced digital ecosystem presents artists with the difficult challenge of sorting through a bulk of comments in order to gain insights about the mood and preferences of their audience. Our project simplifies this process by developing a system which collects and analyzes YouTube comments, by classifying comments into positive, negative and neutral.

First, we collect comments from a wide range of YouTube videos. After that, we used a list of preprocessing steps to clean and organize the data. We will first examine our data to identify patterns and relationships in the comments.

We utilized VADER because it provides quick sentiment scores for comments and is the greatest at capturing complex emotional expressions. These machine learning algorithms - Support Vector Classifier (SVC), Naïve Bayes, and Random Forest helped us to accurately predict sentiment labels. to understand the YouTube comments and do perform sentiment analysis on the comments.

Our intuitive interface makes it simple for content creators to understand the thoughts of their target audience and modify their work to improve engagement.

To summarize, the "YouTube Comments Sentiment Analysis" project aims to utilize NLP techniques along with Machine Learning Algorithms to understand the sentiments behind the YouTube video comments. This helps content creators to understand viewers reaction, optimize content and manage brand reputation.

8. References

[1] YouTube Data API - https://developers.google.com/youtube/v3

[2] Python - https://www.python.org/

[3] Pandas Documentation - https://pandas.pydata.org/

[4] NumPy Documentation - https://numpy.org/

[5] NLTK Documentation - http://www.nltk.org/

[6] Scikit-Learn Documentation - https://scikit-learn.org/

[7] GeeksForGeeks - https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/