Project Report

On

YouTube Comments Sentiment Analysis



Submitted in partial fulfilment for the award of

Post Graduate Diploma in Big Data Analytics (PG-DBDA)

From Know-IT(Pune)

Guided by:

Mr. Anay Tamhankar Sir

Mr. Prasad Deshmukh Sir

Submitted By:

Aniket Adwankar (230343025002)

Vinay Gadade (230343025013)

Yashodhan Jagtap (230343025021)

Shubham Khumbhar (230343025029)

CERTIFICATE

TO WHOMSOEVER IT MAY CONCERN

This is to certify that

Aniket Adwankar (230343025002)

Vinay Gadade (230343025013)

Yashodhan Jagtap (230343025021)

Shubham Khumbhar (230343025029)

**Have successfully completed their project on**

YouTube Comment Sentiment Analysis

Under the guidance of

Mr. Anay Tamhankar Sir

Mr. Prasad Deshmukh Sir

**ACKNOWLEDGEMENT**

This project “You Tube Comment Sentiment Analysis” was a great learning experience for us and we are submitting this work to CDAC Know-IT (Pune).

We all are very glad to mention the name of Anay Tamhankar Sir and Prasad Deshmukh Sir for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We are highly grateful to Mr. Vaibhav Inamdar Manager (Know-IT), C-DAC, for his guidance and support whenever necessary while doing this course Post Graduate Diploma in Big Data Analytics (PG-DBDA) through C-DAC ACTS, Pune.

Our most heartfelt thanks goes to Mrs. Bakul Joshi (Course Coordinator-DBDA) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC Know-IT, Pune.

**From:**

Aniket Adwankar (230343025002)

Vinay Gadade (230343025013)

Yashodhan Jagtap (230343025021)

Shubham Khumbhar (230343025029)

**TABLE OF CONTENTS**

**1.ABSTRACT**

**2. INTRODUCTION**

**3. SYSTEM REQUIREMENTS**

2.1 Software Requirements

**4. FUNCTIONAL REQUIREMENTS**

**5. SYSTEM ARCHITECTURE**

**6. METHODOLOGY**

**7. MACHINE LEARNING ALGORITHMS**

**8. DATA VISUALIZATION AND REPRESENTATION**

**9. CONCLUSION AND FUTURE SCOPE**

**10. REFERANCE**

1. **Abstract**

With the exponential growth of digital content on platforms like YouTube, understanding public sentiments and feedback has become crucial for content creators, artists, and viewers alike. The "YouTube Comment Sentiment Analysis" project addresses this need by employing advanced data analytics techniques to analyse and interpret the sentiments expressed in YouTube comments.

This project leverages a range of technologies, including Apache Kafka for real-time data streaming, Apache Spark for data processing, MongoDB for data storage, and Python for data analysis. The primary objective is to predict and classify sentiments within YouTube comments as positive, negative, or neutral. Natural Language Processing (NLP) techniques are employed to process and analyze the textual content of comments.

The findings of this analysis provide valuable insights into viewer sentiments regarding specific songs or content posted on YouTube. Content creators and artists can benefit from this analysis by gaining a deeper understanding of how their work is received by the audience. Moreover, viewers can discover content aligned with their preferences, enhancing their overall YouTube experience.

The project also explores the challenges encountered during sentiment analysis and discusses potential future enhancements, including advanced sentiment analysis techniques and sentiment-based content recommendations.

In conclusion, the "YouTube Comment Sentiment Analysis" project serves as a valuable tool for both content creators and viewers, offering a data-driven approach to understanding sentiments and enhancing the YouTube music ecosystem.

1. **Introduction**

In the digital age, online platforms have emerged as powerful mediums for content dissemination, entertainment, and interaction. Among these platforms, YouTube stands as a global hub for video content, where creators share their work with a diverse and vast audience. With billions of users engaging with YouTube content daily, the platform has become a dynamic space for expressing opinions, providing feedback, and sharing sentiments.

The significance of YouTube comments extends beyond mere textual exchanges; they offer a window into the collective sentiment of viewers toward the content they engage with. Understanding the sentiment expressed within these comments is of paramount importance, not only for content creators seeking to gauge the reception of their work but also for viewers seeking content aligned with their preferences.

The "YouTube Comment Sentiment Analysis" project is designed to address this need. By harnessing the power of data analytics, natural language processing (NLP), and cutting-edge technologies, this project delves into the intricate world of YouTube comments to decipher sentiments. Its primary objective is to predict and classify sentiments within these comments, categorizing them as positive, negative, or neutral.

The project draws upon a toolkit of technologies, including Apache Kafka for real-time data streaming, Apache Spark for efficient data processing, MongoDB for robust data storage, and Python for advanced sentiment analysis. This combination of tools facilitates the collection, processing, and analysis of vast amounts of textual data in near-real-time.

As we embark on this journey through the YouTube comment landscape, we explore not only the methodologies employed but also the challenges encountered in sentiment analysis. This project represents an opportunity to bridge the gap between creators and their audience, offering valuable insights that can inform content creation and curation.

Furthermore, it showcases the potential for data-driven enhancements to the YouTube experience, such as personalized content recommendations based on sentiment preferences.

**Dataset:**

**YouTube API Data**: You can use the YouTube Data API to collect comments from specific videos or channels. The comments often come with user ratings, and you can use these ratings as labels for sentiment analysis.

**3. SYSTEM REQUIREMENTS**

Software Requirements:

∉ Python 3

∉ Kafka

∉ Apache Spark

∉ Tableau

∉ MongoDB

∉ Google Colab and Jupiter

∉ OS – Windows

4. FUNCTIONAL REQUIREMENTS

**(1) Python 3:**

Python is a general purpose and high level programming language. It is use for developing desktop GUI applications, websites and web applications. Python allows to focus on core functionality of the application by taking care of common programming tasks. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

**(2)** **Apache Spark:**

What is Spark: Apache Spark is an open-source distributed computing system designed for processing large volumes of data. Key Features: Spark provides a number of key features that make it well-suited for processing big data, including in-memory processing, support for various data sources and formats, fault tolerance, and scalability. Spark also provides a range of APIs, including SQL, streaming, machine learning, and graph processing, making it a versatile platform for a wide range of use cases.

**(3) Tableau:**

Data visualization is the graphical representation of information and data. It helps create interactive elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Tableau is widely used for Business Intelligence but is not limited to it. It helps create interactive graphs and charts in the form of dashboards and worksheets to gain business insights. All of this is made possible with gestures as simple as drag and drop.

**(4) Spark:**

1. **Data Processing**:
   * Apache Spark should be capable of processing large-scale datasets efficiently.
2. **Batch and Real-Time Processing**:
   * Support for both batch processing (e.g., processing historical data) and real-time stream processing.
3. **Cluster Computing**:
   * Ability to distribute computation across a cluster of machines for parallel processing.
4. **Fault Tolerance**:
   * Robust fault tolerance mechanisms to handle node failures without data loss.
5. **APIs**:
   * Provide a variety of APIs (e.g., Scala, Python, Java) for data processing and machine learning.

**5) MongoDB**

is a popular and widely used NoSQL (non-relational) database management system that is designed for storing, retrieving, and managing large volumes of unstructured or semi-structured data. Here are some basic details and information about MongoDB:

NoSQL Database: MongoDB is categorized as a NoSQL database because it does not rely on the traditional relational database model. Instead, it uses a document-oriented data model.

Document-Oriented: MongoDB stores data in flexible, JSON-like documents called BSON (Binary JSON). These documents can contain various data types and nested structures, making it suitable for handling complex and dynamic data.

Schema-less: Unlike traditional relational databases, MongoDB is schema-less, which means that documents within a collection can have different structures. This flexibility is well-suited for applications where data structures may evolve over time.

Collections: In MongoDB, data is organized into collections, which are analogous to tables in relational databases. Each collection contains a set of documents, and documents within a collection do not need to have the same fields or structure.

Scalability: MongoDB is designed for horizontal scalability, allowing you to distribute data across multiple servers or clusters. This scalability makes it suitable for handling large-scale applications and big data.

Query Language: MongoDB provides a powerful and flexible query language for retrieving and manipulating data. Queries are expressed using a JavaScript-like syntax and can be used to filter, sort, and aggregate data.

Indexes: MongoDB supports the creation of indexes on fields within collections to improve query performance. Indexes can significantly speed up data retrieval operations.

Aggregation Framework: MongoDB includes a versatile aggregation framework that allows you to perform complex data transformations and aggregations within the database. This is useful for analytics and reporting tasks.

High Availability: MongoDB offers features like replica sets and automated failover to ensure high availability and data redundancy. This means that even if one server fails, data remains accessible from other replicas.

Security: MongoDB provides robust security features, including authentication, authorization, and encryption, to protect data and ensure secure access.

Community and Enterprise Editions: MongoDB is available in both a free and open-source Community Edition and a commercial Enterprise Edition with additional features and support.

Popular Use Cases: MongoDB is commonly used in a wide range of applications, including content management systems, e-commerce platforms, real-time analytics, mobile app backends, and IoT (Internet of Things) applications.

Official Drivers: MongoDB offers official drivers and libraries for various programming languages, making it easy to integrate with your application stack.

MongoDB's flexibility, scalability, and ease of use have contributed to its popularity in the development community. It is particularly well-suited for projects that require handling large volumes of semi-structured or unstructured data with evolving schemas.

Data Cleaning Process:

**a. Text Cleaning:**

-Remove special characters, symbols, and HTML tags from comment text.

-Handle punctuation marks appropriately.

-Convert text to lowercase for consistency.

-Remove extra whitespaces.

**b. Tokenization and Stop word Removal:**

-Tokenize comment text into individual words or tokens.

- Remove common stop words that do not carry meaningful sentiment information.

**c. Handling Missing Data:**

-Address missing values if they exist. Options include imputation or removing rows with missing values, depending on the impact on your analysis.

**d. Data Export:**

- Export the cleaned and transformed data to a suitable format for analysis, such as CSV or a database.

**5. SYSTEM ARCHITECTURE**

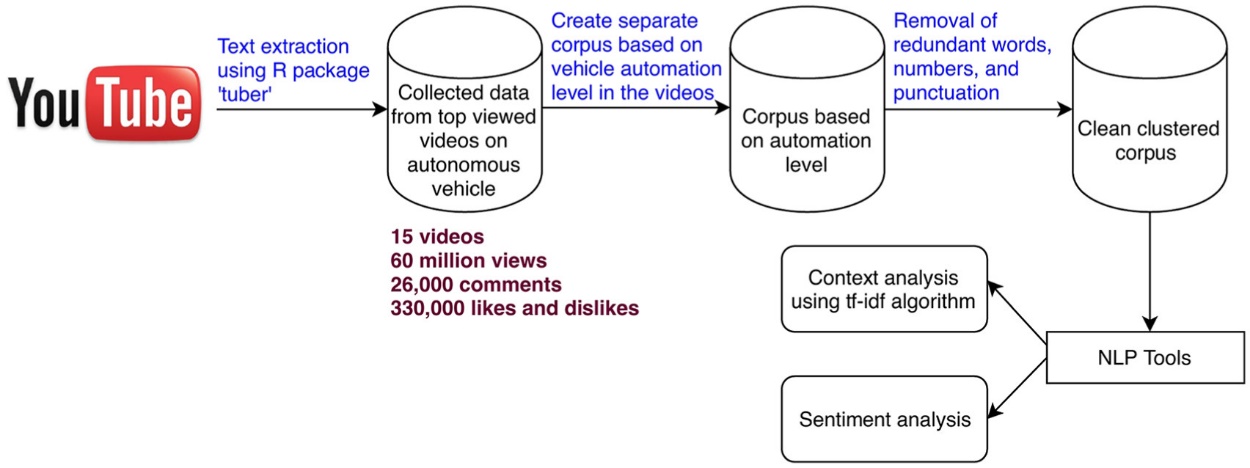


Fig.5.1 YouTube API flow chart

6. **METHODOLOGY**

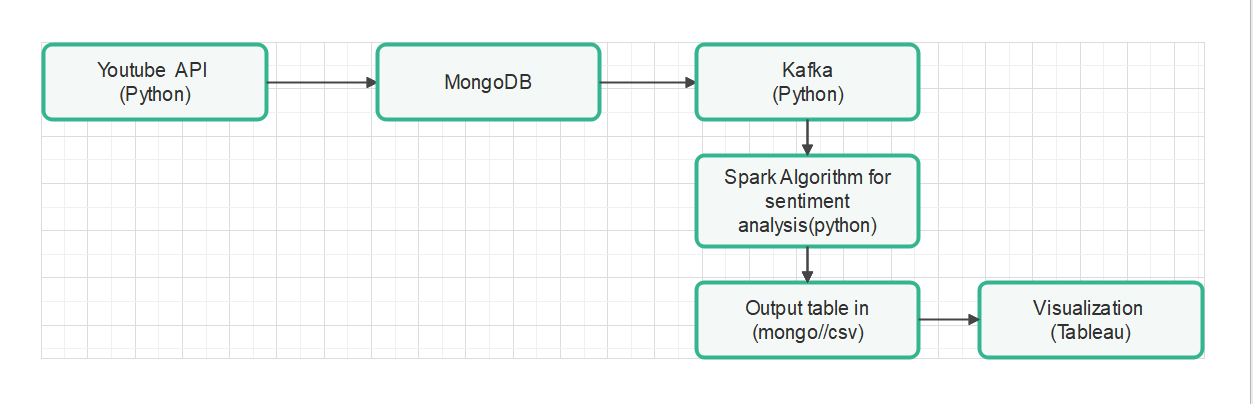


Fig. 6.1 Execution Diagram.

1. **Machine Learning Algorithm**

**Natural Language Processing (NLP)** is a branch of artificial intelligence (AI) that focuses on the interaction between computers and human language. It encompasses the development of algorithms and models that enable computers to understand, interpret, and generate human language. Here are some basic aspects of NLP:

1. Text Processing
2. Language Understanding
3. Sentiment Analysis
4. Machine Translation
5. Speech Recognition
6. Information Retrieval
7. Text Generation
8. Text Summarization
9. **VADER (Valence Aware Dictionary and sentiment Reasoner)** is a lexicon and rule-based sentiment analysis tool designed for analysing and categorizing the sentiment or emotional tone of text data. It was specifically developed for social media text, product reviews, and other forms of text data found on the internet. Here are some basic characteristics and information about VADER:
10. **Lexicon-Based Approach:** VADER relies on a pre-built lexicon (a dictionary or list of words) that is associated with sentiment scores. Each word in the lexicon is assigned a polarity score that indicates how positive or negative it is.
11. **Valence Scores:** VADER lexicon includes valence scores for individual words, which represent the strength and direction of sentiment (positive, negative, or neutral). These scores are typically on a scale from -4 to +4.
12. **Rule-Based Analysis:** In addition to word-level scores, VADER uses a set of grammatical and syntactical rules to analyse the sentiment of sentences and text as a whole. This helps capture nuances in sentiment that may not be evident at the word level.
13. **Emphasis on Context:** VADER is designed to consider the context of words and phrases in a sentence. For example, it can recognize negations (e.g., "not good") and intensifiers (e.g., "very good") to provide more accurate sentiment analysis.
14. **Sentiment Categories:** VADER typically categorizes text sentiment into three main categories:
    * Positive Sentiment
    * Negative Sentiment
    * Neutral Sentiment
15. **Compound Score:** VADER also provides a compound score, which is a single number that represents the overall sentiment of a piece of text. This score is often used to classify text as positive, negative, or neutral based on a chosen threshold.
16. **Real-Time Analysis:** VADER is capable of providing sentiment analysis in real-time, making it suitable for applications such as social media monitoring and customer feedback analysis.
17. **Python Implementation:** VADER is commonly used in Python through the **vaderSentiment** library, making it accessible and easy to integrate into Python-based projects.
18. **Strengths and Limitations:** VADER is known for its simplicity and speed, but it may not perform as accurately as more complex machine learning models on certain tasks. Its performance can vary depending on the domain and context of the text being analyzed.
19. **Open Source:** VADER is an open-source tool and is freely available for use in various natural language processing (NLP) applications.

VADER is a valuable tool for quick and relatively straightforward sentiment analysis tasks. However, for more complex NLP tasks or domain-specific applications, machine learning-based approaches and custom models may be preferred.

Top of Form

1. **DATA VISUALIZATION AND REPRESENTATION**

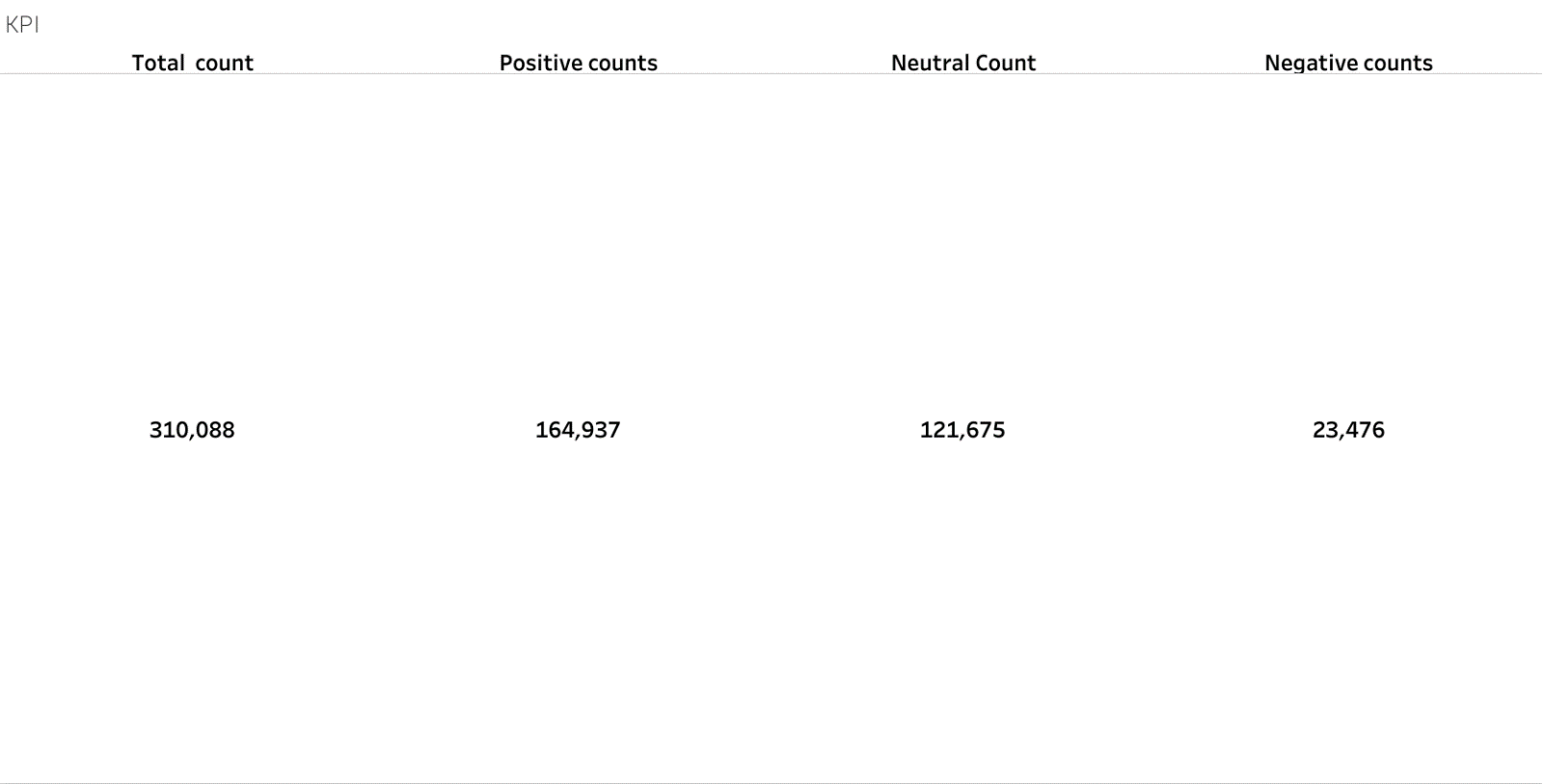
****

Fig.8.1 KPI visualization

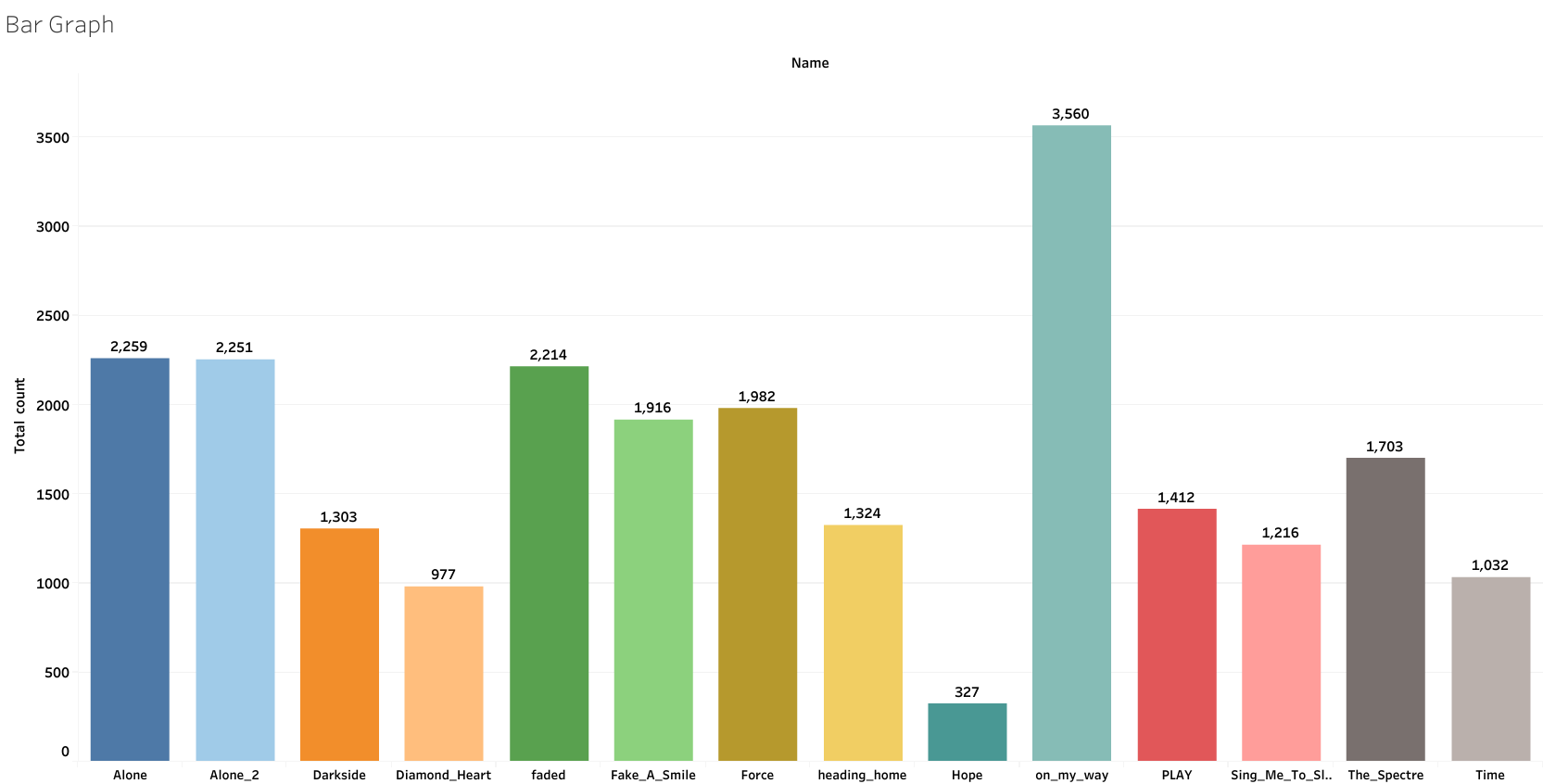
****

Fig.8.2 Bar chart

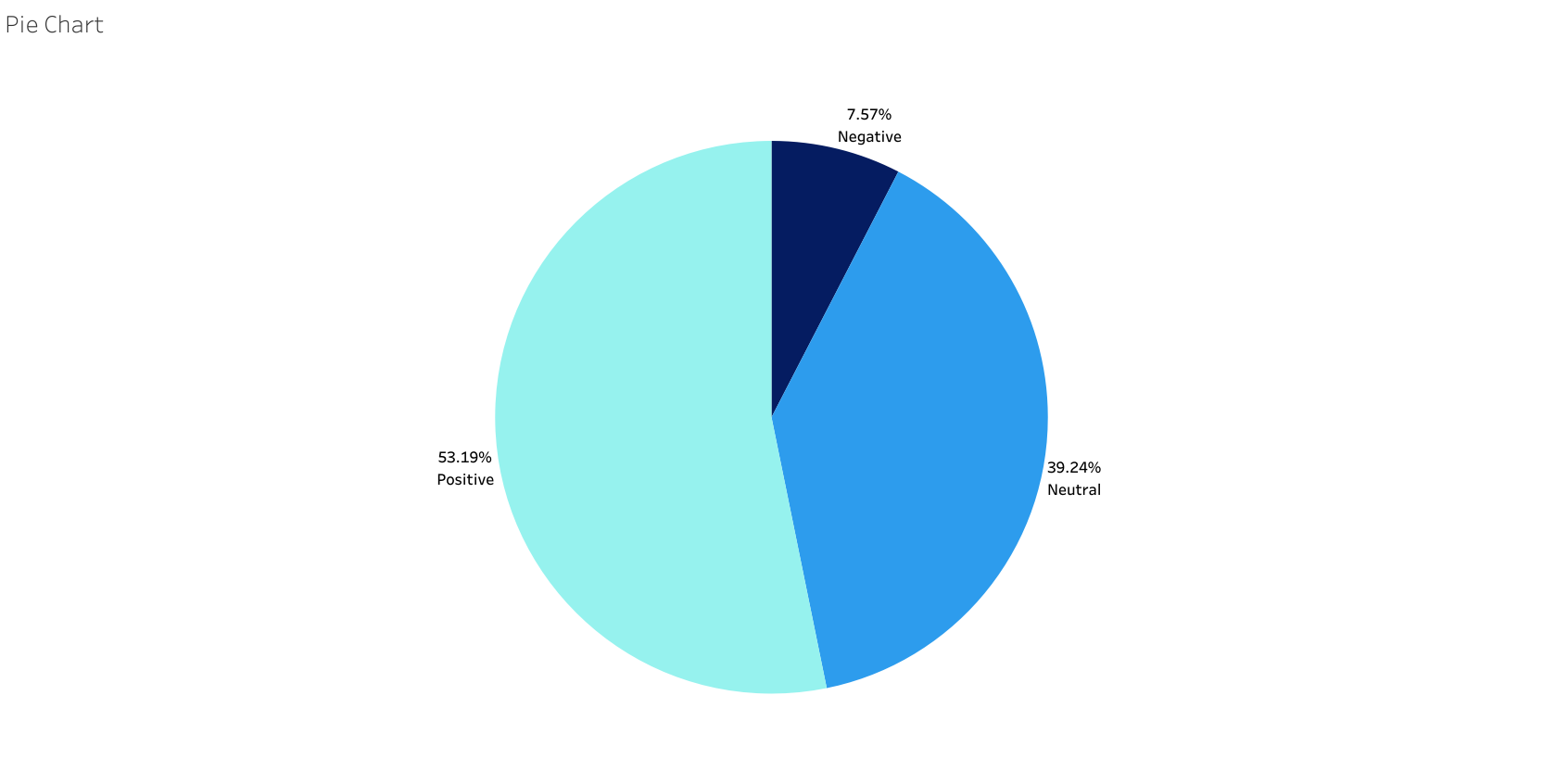


Fig.8.3 Pie chart

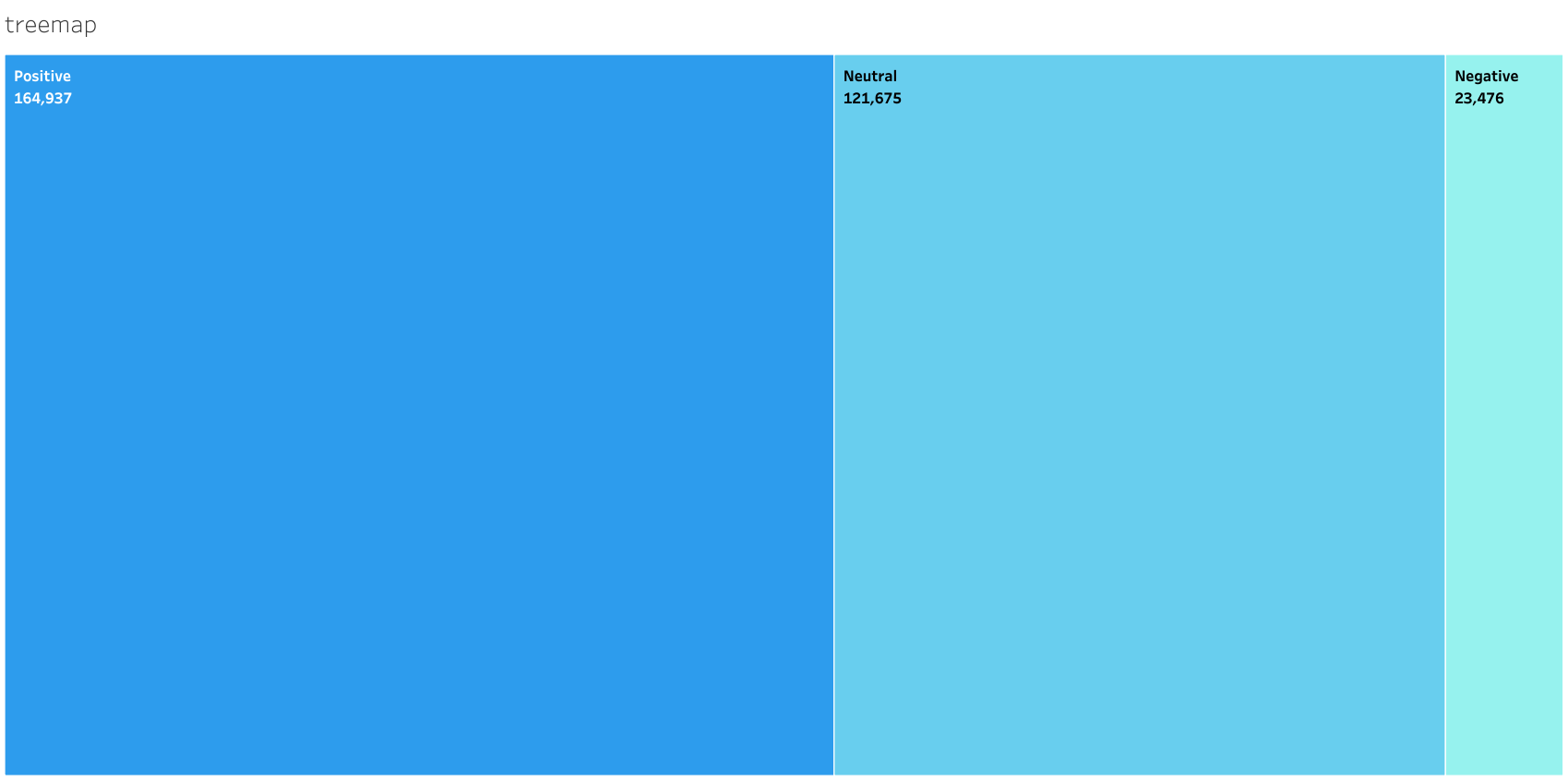


Fig.8.4 Tree Map

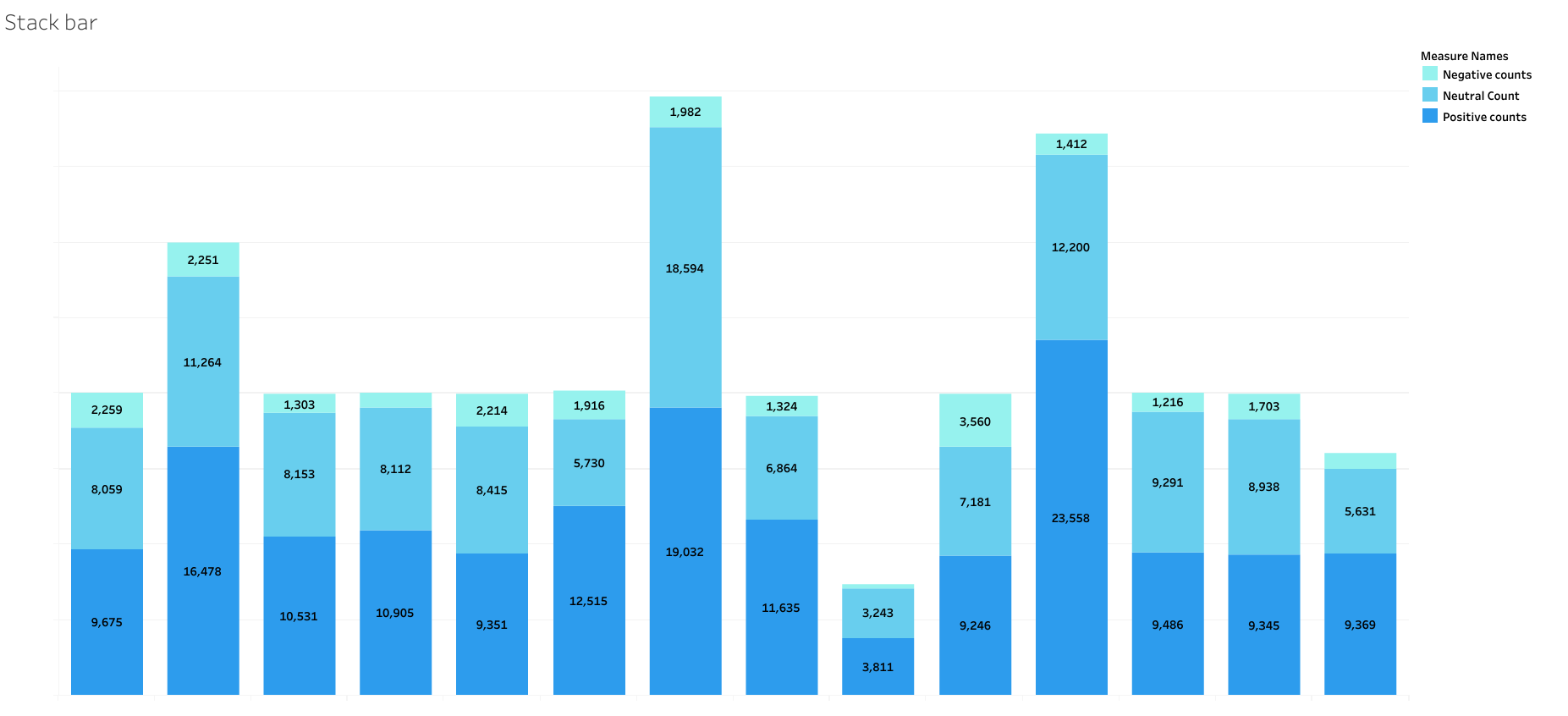


Fig.8.5 Stacked Bar Graph

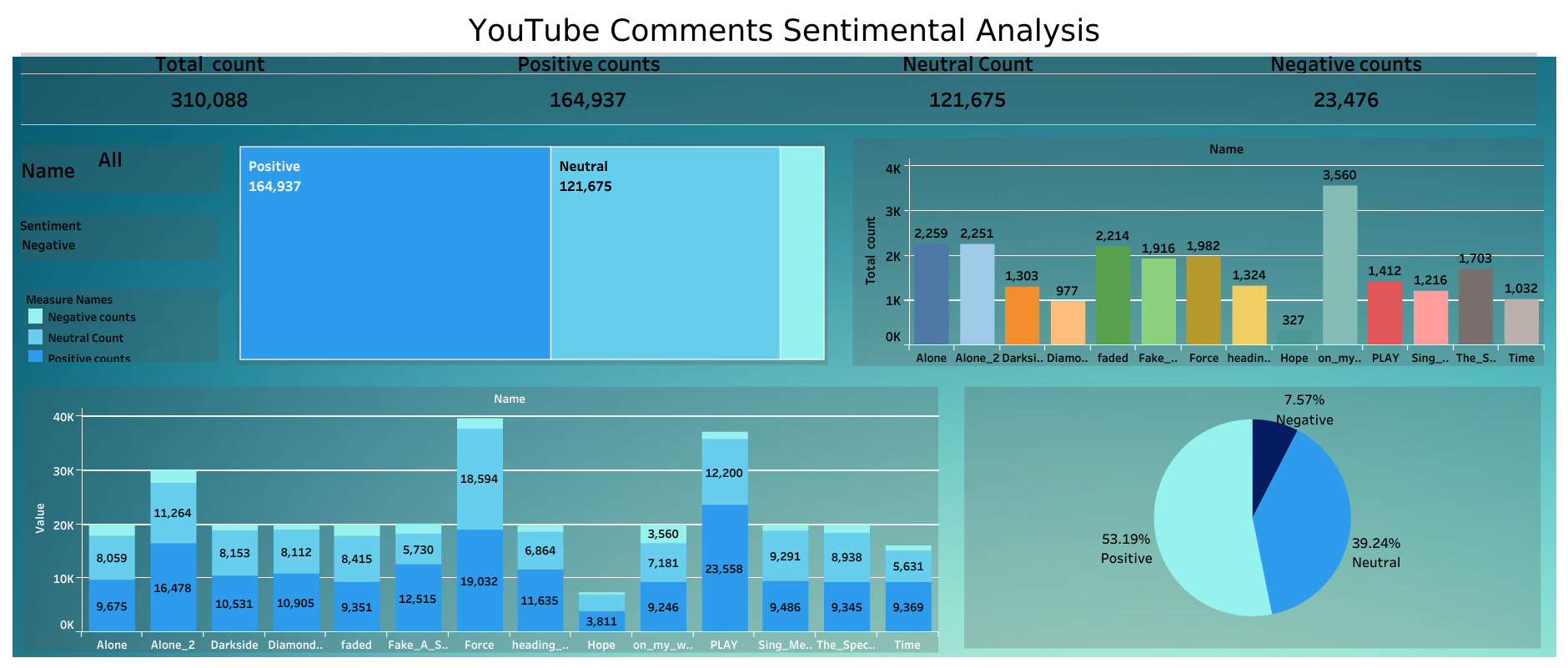


Fig.8.6 Dashboard

1. **Conclusion & Future Scope**

**Conclusion:**

YouTube comment sentiment analysis is a valuable tool for understanding and harnessing the power of user-generated content on the platform. Through sentiment analysis, you can gain insights into how viewers perceive and interact with videos, songs, or any other content. Here are some key conclusions from this project:

1. **Sentiment Insights:** The analysis has provided valuable insights into the sentiment of YouTube comments related to songs. By categorizing comments as positive, negative, or neutral, we can gauge the overall sentiment surrounding a particular video.
2. **User Engagement:** Sentiment analysis can help identify the level of user engagement and whether viewers like or dislike a song. Positive sentiment may indicate a song's popularity, while negative sentiment can highlight areas for improvement.
3. **Content Improvement:** By analysing negative comments, content creators and artists can identify areas where they can enhance their work or address viewer concerns.
4. **Audience Feedback:** Sentiment analysis enables content creators to better understand their audience's preferences and tailor their content accordingly.
5. **Data-Driven Decisions:** Insights from sentiment analysis can inform data-driven decisions for content promotion, marketing strategies, and artist collaborations.

**Future Scope:**

The field of YouTube comment sentiment analysis offers numerous opportunities for further research, development, and application. Here are some potential future directions:

1. **Multimodal Analysis:** Incorporate other types of data, such as video views, likes, dislikes, and user demographics, for a more comprehensive analysis of viewer sentiment.

2.**Real-Time Analysis:** Develop real-time sentiment analysis systems that can process and respond to comments as they are posted on YouTube.

3.**Improved Accuracy:** Enhance sentiment analysis models using state-of-the-art NLP techniques, including deep learning and transformer-based models.

4.**Topic Modelling:** Extend analysis to identify common topics or themes in comments, providing deeper insights into viewer interests.

5.**User Sentiment Profiles:** Develop profiles of user sentiment over time to track changes in audience sentiment and preferences.

6.**Content Recommendations:** Use sentiment analysis to improve content recommendations on YouTube, helping users discover videos aligned with their sentiments and interests.

7.**Language Support:** Extend sentiment analysis to multiple languages to cater to a global audience.

8.**Ethical Considerations**: Continue to address ethical considerations in sentiment analysis, including bias mitigation and privacy protection.

9.**Content Creator Tools**: Create tools and dashboards that content creators can use to monitor sentiment and engage with their audience effectively.

10.**Business Insights:** Apply sentiment analysis to provide business insights for record labels, marketing agencies, and advertisers to make informed decisions.

YouTube comment sentiment analysis is a dynamic field with the potential to transform how content creators and platforms interact with their audiences. As the volume of user-generated content continues to grow, sentiment analysis will play a crucial role in extracting valuable insights and enhancing the user experience on YouTube and other social media platforms.

**10. References**

1. YouTube Api :-<https://developers.google.com/youtube/v3/docs/>
2. NLTK :-<https://www.nltk.org/>
3. About VADER :-<https://www.analyticsvidhya.com/blog/2021/06/vader-for-sentiment-analysis/>
4. VADER :-<https://www.nltk.org/_modules/nltk/sentiment/vader.html>
5. Kafka Documentation :- https://kafka.apache.org/24/documentation.html
6. Spark Documentation : <https://spark.apache.org/documentation.html>
7. Pandas Data frame :-https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html
8. PyMOngo :- <https://pymongo.readthedocs.io/en/stable/index.html>