A

Minor Project Report

On

VIDEO DATA ANALYTICS PROCESS AND ANALYZE META DATA

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY
IN

COMPUTER SCIENCE AND ENGINEERING(DATASCIENCE)

Submitted
By

M. HARSHINI	**(228R1A67G5)**
DEVI SREE RATHOD	**(228R1A67E0)**
ASHWINI MARATI	**(228R1A67G7)**
D. SHRAVANI	**(228R1A67D9)**

Under the Esteemed guidance of

Mr. R. Srikanth

Assistant Professor, Department of CSE(Data Science)

Department of CSE(Data Science)

CMR ENGINEERING COLLEGE
(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

(Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

(2024-2025)

CMR ENGINEERING COLLEGE

(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

(Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

CERTIFICATE

This is to certify that the project entitled **" Video data analytics process and analyze meta data "** is a bonafide work carried out by

M. HARSHINI	**(228R1A67G5)**
DEVI SREE RATHOD	**(228R1A67E0)**
ASHWINI MARATI	**(228R1A67G7)**
D. SHRAVANI	**(228R1A67D9)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **CSE (DATA SCIENCE)** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	Mini Project Coordinator	Head of the Department
Mr. R. Srikanth	**Mr. E. Laxman**	**Dr. M. LAXMAIAH**
Assistant Professor	Assistant Professor	Professor & HOD
Department of CSE(DS)	Department of CSE(DS),	Department of CES(DS)
CMREC, Hyderabad	CMREC, Hyderabad	CMREC, Hyderabad

DECLARATION

This is to certify that the work reported in the present project entitled **" VIDEO DATA ANALYTICS PROCESS AND ANALYZE META DATA"**is a record of bonafide work done by us in the Department of CSE(data science), CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

M. HARSHINI (228R1A67G5)

DEVI SREE RATHOD (228R1A67E0)

ASHWINI MARATI (228R1A67G7)

D. SHRAVANI (228R1A67D9)

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal ,**Dr. M. Laxmaiah**, Professsor&HOD, **Department of CSE(Data science), CMR Engineering College** for their constant support.

We are extremely thankful to **Mr. R. Srikanth,** Assistant Professor, Internal Guide, Department of CSE(Data science), for his constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literature referred in this Project.

We express our thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, We are very much thankful to our parents who guided us for every step.

M. HARSHINI (228R1A67G5)
DEVI SREE RATHOD (228R1A67E0)
ASHWINI MARATI (228R1A67G7)
D. SHRAVANI (228R1A67D9)

CONTENTS

ABSTRACT

Video Data Analytics is changing the way media companies and marketers understand how viewers interact with video content. In today's digital world, where millions of videos are uploaded and watched every day, it's important to know which videos are successful and why. Video Data Analytics uses Python tools and smart algorithms to study video metadata like watch time, likes, shares, and video descriptions. This helps companies discover patterns in viewer behavior and make better decisions about the kind of content they create.

One of the biggest advantages of Video Data Analytics is that it saves time and effort. Instead of going through each video manually, the system automatically analyzes large amounts of data and delivers fast insights. This is especially useful for platforms with thousands of videos. It also helps improve content quality by showing exactly which parts of a video viewers enjoy the most and where they tend to lose interest. With this information, creators can make videos that are more engaging and appealing to their audience.

Keywords:

Video Data Analytics,Viewer behavior,Video metadata,Engagement metrics,Content strategy,Python tools,Smart algorithms.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Introduction

In today's digital era, where video content dominates online platforms, Video Data Analytics has emerged as a critical tool for media companies and marketers aiming to understand viewer behavior and improve content performance. With billions of videos being uploaded, streamed, and shared across platforms like YouTube, Netflix, and social media, analyzing video data has become essential for creating effective content strategies, enhancing viewer engagement, and staying competitive in the fast-paced digital landscape. Traditional methods of manual content evaluation are no longer sufficient to handle the sheer volume and complexity of video data, making automated analytic s solutions increasingly indispensable.

Video Data Analytics systems leverage Python-based libraries and advanced algorithms to automatically extract, process, and analyze video metadata, such as timestamps, viewer engagement metrics (likes, shares, comments, watch time), and content descriptors (tags, categories, transcripts). This enables organizations to identify meaningful patterns and trends across large video datasets quickly and efficiently. The insights generated help decision-makers fine-tune their content based on real viewer behavior, ensuring that the right type of content reaches the right audience at the right time. By automating these analytical tasks, Video Data Analytics not only saves time but also improves the accuracy and depth of insights, which are crucial for developing data-driven strategies in competitive digital environments.

Furthermore, Video Data Analytics plays a pivotal role in optimizing the overall content lifecycle, from planning and production to distribution and performance monitoring. It enables creators and marketers to tailor their content for different demographics and platforms by uncovering granular viewer preferences and behaviors. Centralizing video data insights into a single platform allows for better collaboration among creative, technical, and marketing teams, ensuring consistent and informed decision-making. In addition to improving internal workflows, analytics systems also contribute to a better viewer experience by enabling more relevant, engaging, and personalized video content.

1.2 Project Objectives

In today's content-driven digital world, **Video Data Analytics** plays a vital role in helping media companies and marketers understand how viewers interact with videos across various platforms. The main objective of this project is to extract meaningful insights from video metadata to improve content strategies, enhance viewer engagement, and support more personalized content delivery. This system aims to automate the analysis of video performance using advanced Python libraries and algorithms, allowing organizations to make faster, more accurate, and data-driven decisions.

1.3 Purpose of the Project

The primary purpose of implementing a Video Data Analytics system is to optimize content creation and distribution strategies by analyzing key metrics such as watch time, engagement, and content descriptors. This system is designed to identify patterns in viewer behavior, reduce the time spent on manual analysis, and enable targeted content recommendations. By streamlining the process of analyzing video performance, this solution aims to enhance the quality of content, improve audience targeting, and support better collaboration among creative, technical, and marketing teams.

1.4 Existing System with Disadvantages

In many organizations, video performance evaluation is still done manually or with limited tools, which can be **time-consuming, inconsistent, and inefficient**. Without automated analytics, understanding viewer engagement across a large volume of video content becomes difficult, often resulting in missed opportunities for improving content effectiveness and reaching the right audience. The manual approach also limits the ability to identify real-time trends and respond quickly to changing viewer preferences.

Disadvantages

Manual video analysis is slow and does not scale well with large volumes of content.

Limited insight into detailed viewer behavior such as drop-off points and most-watched segments

1.5 Proposed System with Features

The proposed Video Data Analytics system is designed to overcome the limitations of traditional manual video performance analysis by introducing automation, pattern recognition, and viewer engagement tracking. This system uses Python-based tools and libraries to automatically extract and analyze metadata such as watch time, viewer drop-off points, content type, and timestamps from video files and platforms. The goal is to provide real-time, data-driven insights that help media teams and marketers improve video quality, content strategy, and audience engagement.

Key features of the system include automated metadata extraction, trend detection, engagement scoring, and interactive visualizations of performance metrics. These insights allow stakeholders to identify what type of content performs best, which parts of a video keep viewers engaged, and where viewers tend to lose interest. The system also integrates with video hosting and streaming platforms for seamless data flow and enables organizations to make quick decisions based on viewer behavior analytics.

Additionally, the system supports personalized recommendations for content creators by highlighting preferred topics, durations, and engagement tactics. It also includes reporting capabilities that allow users to generate detailed performance summaries, making it easier to measure ROI and adapt strategies over time. Overall, the system transforms raw video data into actionable insights, promoting better content planning, increased viewer satisfaction, and more efficient marketing campaigns.

1.6 Input And Output

DesignInput Design

The input design for a **Video Data Analysis System** begins with the collection of video content and associated metadata. The system accepts input in the form of video files (such as MP4, AVI, or MKV formats) and utilizes tools like **FFmpeg** and **OpenCV** to extract metadata including timestamps, frame rate, duration, resolution, audio track info, and more. Additionally, user engagement metrics (e.g., views, likes, shares, drop-off time) are gathered from integrated platforms like YouTube, Vimeo, or social media analytics APIs.

This input data is then structured and cleaned using **Python libraries such as Pandas**, ensuring consistency, removing noise, and preparing it for meaningful analysis. The design supports batch video uploads, making it scalable for organizations with large video libraries. Structured inputs such as CSV files containing engagement stats, or logs from streaming services, can also be integrated for a comprehensive analysis pipeline.

Objectives:

The system should accept video files in multiple formats (e.g., MP4, AVI, MKV).

Automatically extract video metadata (duration, resolution, codec, frame rate, etc.) using FFmpeg/OpenCV.

Ingest structured data like CSV/JSON files for user behavior metrics (views, watch time, drop-off rates).

Normalize data from various sources to a unified format for analytics.

Include validation checks for corrupted files or missing metadata entries to ensure data integrity.

Output Design

The output design of the Video Data Analysis System is structured to provide actionable insights into viewer behavior and video performance. The system generates visual and textual reports that summarize key engagement metrics such as average view duration, peak engagement time, and viewer drop-off points. These insights are presented using visualization libraries like Matplotlib, Seaborn, or interactive dashboards in Streamlit, allowing media teams to make data-driven decisions.

The output includes trend analysis across different content categories, viewer demographics, and publishing timelines. Advanced analytics outputs like content recommendation insights, best-performing video length, and optimal posting time are also generated. These help marketing and content teams fine-tune their strategies.

Interactive reports allow users to filter data by video type, audience location, and platform. Additionally, the system can export the analytics in formats like PDF, Excel, or as dynamic web dashboards, making reporting and sharing with stakeholders easy and efficient.

Output Objectives:

Generate visual dashboards showing engagement trends and audience behavior.

Provide summary reports of key metadata and performance metrics per video.

Output predictive insights such as recommended content duration or publish time.

Deliver customizable exports (PDF, CSV, web dashboard) for presentation and review.

Support real-time or scheduled report generation for continuous monitoring.

2. LITERATURE SURVEY

1. Jagad et al., "A Study on Video Analytics and Their Performance Analysis for Various Object Detection Algorithms," 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET), Arad, Romania, 2022, pp. 1095-1100, doi: 10.1109/GlobConET53749.2022.9872506.

Video analytics has gained immense significance from the industry and academics as it provides simultaneous monitoring, analysis, and swift response to unforeseen circumstances. The ability to automatically identify temporal and spatial occurrences in videos and generate important insights is a key element of video analytics. The prime functionality of Video analytics includes Object Detection, Anomaly Detection, and Video annotation. Video analytics is extremely beneficial in the security domain, and video surveillance, an application of video analytics, is used to identify improper behavior or anomalies. Anomaly detection is important in cybersecurity intrusion detection, system health monitoring, surveillance detection in sensor networks, and law enforcement. Analysis of videos can be achieved using Object Detection algorithms like You Only Look Once (YOLO), Region Proposals algorithms (R-CNN, Fast R-CNN and Faster R-CNN), or Single Shot MultiBox Detector (SSD). A comprehensive introduction, network architecture, and internal mechanisms of these algorithms are presented in this paper. Furthermore, a comparative performance analysis of these models was conducted by evaluating the models based on results obtained from the implementation and studying previous research works. We further discuss some of the challenges encountered in video analytics and a set of possible future opportunities and new perspectives on addressing them.

2. Khatri, T. Choudhury, T. P. Singh and M. Shamoon, "Video Analytics Based Identification and Tracking in Smart Spaces," *2019 International Conference on contemporary Computing and Informatics (IC3I)*, Singapore, 2019, pp. 261-267, doi: 10.1109/IC3I46837.2019.9055614.

Nowadays, video cameras have become a necessary part of everyday life. These cameras can be found at various places for surveillance, health care, emergencies and disasters. In modern technology, the availability of cheaper and faster communication technologies and efficient algorithms has led to a large-scale deployment of camera networks. Now, Video Content Analysis also referred as Intelligent Video Analytics is used to detect,

track and analyze the movement and behaviors of objects and humans.

This topic finds various applications because ability to recognize, track and take out information about human activity is very much required. The more technical uses of video content analytics are in face recognition, license plate recognition, tracking a moving object, searching for an object, identifying the geographic location of something, detecting suspicious activities etc. This paper proposes to show and review various algorithms and steps for the applications of the video content analysis that are available in the present date and will also highlight the problems that still exist to find correct results.

3.Xia *et al*., "Video Visualization and Visual Analytics: A Task-Based and Application- Driven Investigation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 11, pp. 11316-11339, Nov. 2024, doi: 10.1109/TCSVT.2024.3423402.

Video data refers to digital information in the form of a series of frames or images representing continuous motion captured by a video recording device. In various domains such as security, sports, education, and entertainment, a significant amount of video data is generated and stored daily. However, analyzing these videos manually is challenging due to their intrinsic characteristics, including large-scale, redundancy, contextual dependencies, and multimodality. Consequently, researchers have extensively explored visualization techniques to address these complexities. In this investigation, we review the state-of-the-art techniques in video visualization and visual analysis. Initially, we provide an overview of the design space for video visualization and visual analysis techniques. Subsequently, we organize and classify these techniques based on visual analysis tasks and application scenarios, providing detailed descriptions within each category. Drawing upon a comprehensive review of existing research, we provide a critical evaluation and propose potential opportunities for future research. Additionally, we have developed a web-based survey browser for convenient exploration of our created classification framework and the associated scholarly articles (https://zjutvis.github.io/VOVideo/).

4. **Suvonvorn, "A video analysis framework for surveillance system," *2008 IEEE 10th Workshop on Multimedia Signal Processing*, Cairns, QLD, Australia, 2008, pp. 867-871, doi: 10.1109/MMSP.2008.4665195**.

An on-line video processing for surveillance system is a very challenging problem.

The computational complexity of video analysis algorithms and the massive amount of data to be analyzed must be considered under real-time constraints. Moreover it needs to satisfy different criteria of application domain, such as, scalability, re-configurability, and quality of service. In this paper we propose a flexible/efficient video analysis framework for surveillance system which is a component-based architecture. The video acquisition, re-configurable video analysis, and video storage are some of the basic components. The component execution and inter-components synchronization are designed for supporting the multi-cores and multi-processors architecture with multi-threading implementation on .NET Framework. Experimental results on real-time motion tracking are presented with discussion.

Nowadays, there are video platforms emerging and data virtualization and prediction on those platforms draw little attention. Related research on it is beneficial for both content creators and advertisers as it helps decision-making. In this study, a video platform in China called Bilibili was selected. This paper acquired the dataset from a number of top-rated video uploaders in their field from a data share website and preprocess them to combine them for each month. For each frame of the data, the number of likes, shares, comments, archives, and bullet charts was used to predict the number of views. In the analysis, this study mainly used random forest and sklearn to achieve the prediction. 368 data points were utilized, and in each data point the amount of favorite, share, save and play are combined to make the prediction. The prediction that is made by using random forest is much more realistic than that made by using sklearn because some of the statistics in the sklearn graph reach negative, which is not realistic in the real situation. According

3. SOFTWARE REQUREIMENTS ANALYSIS

3.1 Problem Statement

The primary problem addressed by the **Video Data Analysis System** is the lack of actionable insights and strategic understanding derived from video content and user engagement data. In media, marketing, and educational contexts, vast volumes of video content are generated without sufficient analysis of how audiences interact with them. Traditional video evaluation methods rely on manual observation or basic platform analytics, which are time-consuming, lack depth, and often miss key patterns like viewer drop-off points or high engagement segments.

3.2 Modules and Their Functionalities

1. Metadata Extraction Module

Utilizes tools like **FFmpeg** and **OpenCV** to extract metadata from uploaded videos, including resolution, duration, frame rate, bit rate, codec, and audio properties.

Automates timestamp generation for further scene or activity tracking.

2. Data Preprocessing Module

Cleans, structures, and organizes the raw metadata and user engagement logs using **Pandas**.

Supports normalization and standardization for cross-platform comparisons.

Integrates structured logs from platforms like YouTube Analytics, Vimeo, or custom LMS dashboards.

3. Analytics Engine

Applies descriptive and predictive analytics on user interaction data.

Identifies patterns such as drop-off points, peak engagement, retention curves, and watch-time trends.

Supports content classification and tagging for deeper insight.

4. Visualization & Reporting Module

Generates interactive visualizations using Matplotlib, Plotly, and dashboards with Streamlit.

3.3 Non-Functional Requirements

Scalability: The system must handle large volumes of video files and concurrent user access, with efficient processing and storage mechanisms to ensure performance at scale.

Reliability: It should provide consistent availability, accurate results, and reliable analytics even under load. Backup and recovery systems must prevent loss of processed data or analytics history.

Performance: Real-time or near-real-time metadata extraction and analysis must be supported. Large video datasets should not cause significant delays in report generation.

Usability: The UI/UX should be intuitive for both technical and non-technical users, providing visual tools and dashboards that simplify exploration of data insights.

Security: All uploaded video files and engagement data must be securely stored with appropriate encryption. Access controls should be enforced based on user roles.

Compatibility: The system should work seamlessly across major browsers and operating systems. It should be accessible via desktop and mobile devices.

3.4 Feasibility Study

The feasibility study for implementing a Video Data Analysis System evaluates the practicality and benefits of the project, considering technical, operational, and financial aspects. From a **Technical Feasibility** perspective, the system can be developed using existing technologies such as Python-based libraries (e.g., OpenCV, FFmpeg) for video processing, and data science tools (e.g., Pandas, NumPy, and machine learning frameworks) for insight extraction and predictive analysis. Cloud storage and computing platforms also support scalability and real-time processing. The necessary infrastructure, software tools, and technical expertise are readily available, making the technical implementation viable.

In terms of **Operational Feasibility**, the system addresses key challenges in managing and extracting value from large volumes of video content. It automates metadata extraction, analyzes viewer engagement patterns, and supports decision-making in content creation and marketing strategies. The system can be seamlessly integrated into existing media workflows and content management systems, offering intuitive dashboards and visualization tools for end-users such as content analysts and digital marketers.

4 SOFTWARE AND HARDWARE REQUIREMENTS

4.1 Software Requirements

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraintsand user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

Operating system : Windows 10

Coding Language : python

Anaconda platform : python coding platform

4.2 Hardware Requirements

Minimum hardware requirements are very dependent on the particular software being developed bya given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

System : Intel Core i7 .

Hard Disk : 256 GB SSD.

Monitor : 15'' LED

Input Devices

Keyboard, Mouse Ram: 4 GB

5. SOFTWARE DESIGN

5.1 System Architecture

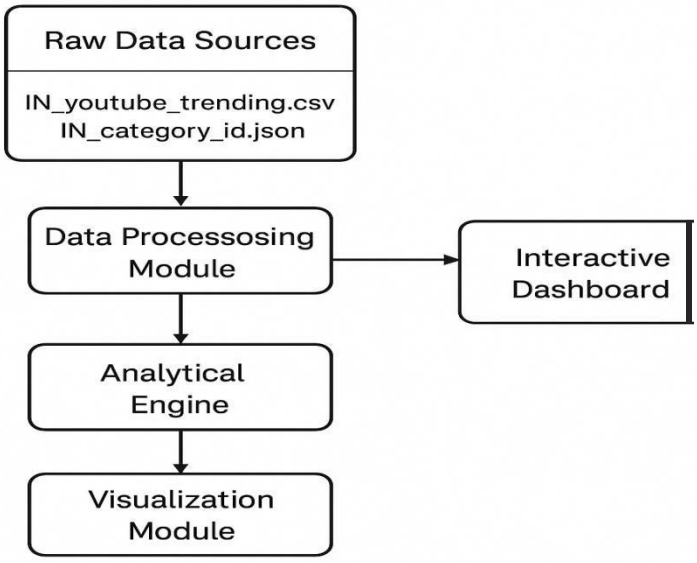

Figure:5.1 System Architecture

Traditional content and media performance evaluation methods rely heavily on manual review of video files and scattered analytics from multiple platforms. This approach is inefficient, time-consuming, and often leads to missed insights due to the lack of centralized data analysis. The Video Data Analysis System is designed to automate the process of extracting metadata from video content, analyze viewer engagement patterns, and provide meaningful insights for media planning and marketing strategies.

The architecture follows a modular design, ensuring scalability, maintainability, and ease of integration with future components like real-time analysis or streaming platform APIs.

5.2 Dataflow Diagram

The Data Flow Diagram (DFD), often referred to as a bubble chart, is a graphical representation that illustrates how data flows through a system, including how the data is input, processed, stored, and output. In the context of a Video Data Analytics system, the DFD visualizes how video metadata and user engagement inputs are transformed into valuable insights.

The DFD serves as a powerful modeling tool to break down the system into its logical components. These components typically include the input sources (such as video platforms), processes (like metadata extraction and trend analysis), data stores (like analytics databases), and external entities (such as content strategists or marketing teams).

It helps in understanding how data moves through the analytics pipeline—from video ingestion to metadata extraction, engagement analysis, pattern detection, and finally to insight generation for decision-makers.

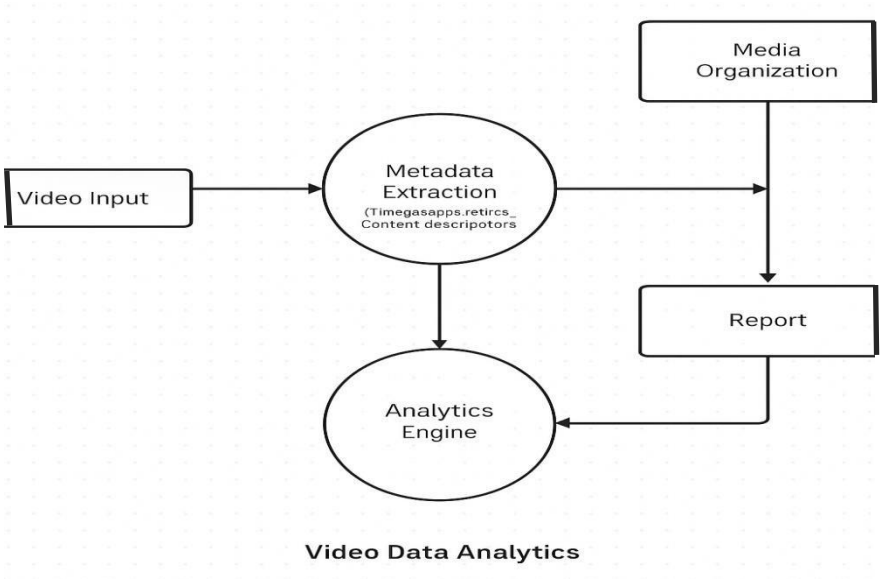

Figure:5.2 Dataflow Diagram

5.3 UML Diagrams

UML is a standard language for specifying, visualizing, constructing, and documenting the artifactsof software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.
There are several types of UML diagrams and each one of them serves a different purpose regardless of whether it is being designed before the implementation or after (as part of documentation). UMLhas a direct relation with object-oriented analysis and design. After some standardization, UML hasbecome an OMG standard. The two broadest categories that encompass all other types are:

- Behavioral UML diagram and
- Structural UML diagram.

As the name suggests, some UML diagrams try to analyses and depict the structure of a system or process, whereas other describe the behavior of the system, its actors, and its building components.

Goals: The Primary goals in the design of the UML are as follows:
Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

Provide a formal basis for understanding the modeling language.

Encourage the growth tools market.

Integrate best practices.

The different types are as follows:

- Sequence diagram
- Use case Diagram
- Activity diagram
- Class diagram
- Collaboration diagram

Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

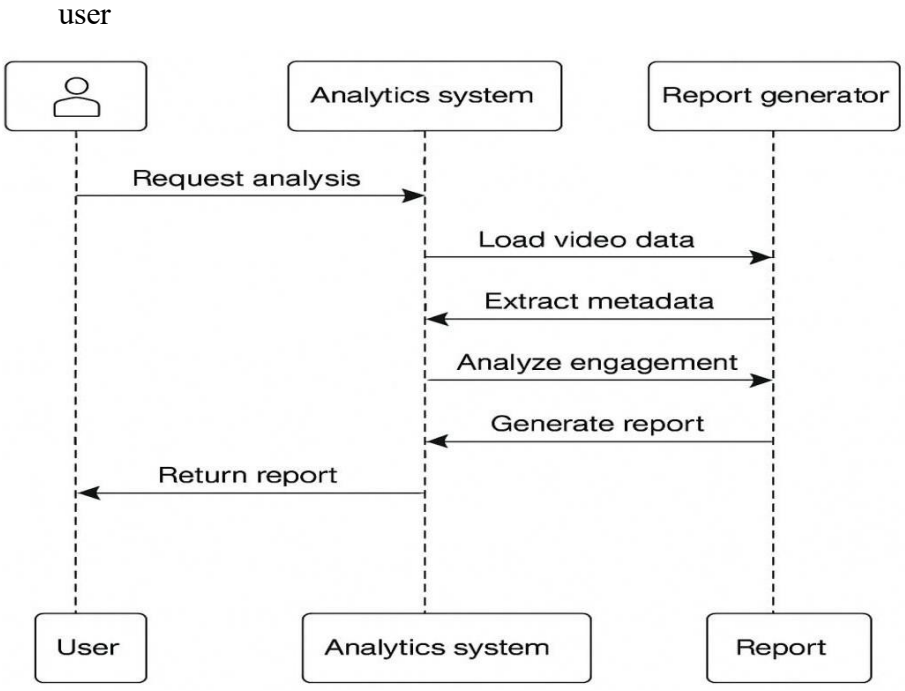

Figure 5.3.1 Sequence Diagram

List of actions

User:

The user needs to press any of the given three options (i.e., **Prediction Data**, **Prediction Skills**, or **Analyze Video Metrics**) from the interface. Based on the selected option, the user will provide input such as video metadata, engagement statistics, or content category.

System:

The system receives the input and processes it using the trained machine learning model. It evaluates the data to detect suspicious patterns, analyze trends, or assess content performance.

Result:

Based on the processed input, the system provides the output. For example, it may determine whether the content is **Fraudulent or Not Fraudulent**, or provide insights into **viewer engagement**, **content quality**, or **trend alignment**.

Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that showsthe relationship between the user and the different use case in which the user is involved. A use casediagram is used to structure of the behavior thing in a model. The use cases are represented by eithercircles or ellipses.

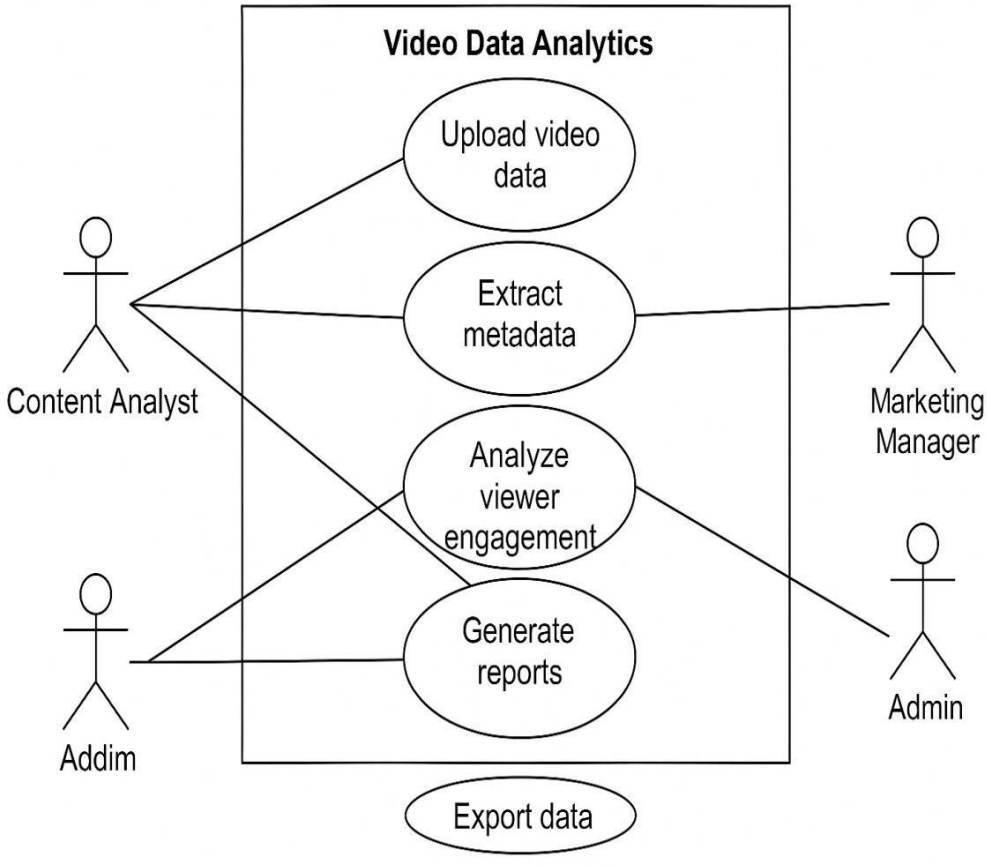

Figure 5.3.2 Use Case Diagram

Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of staticstructure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

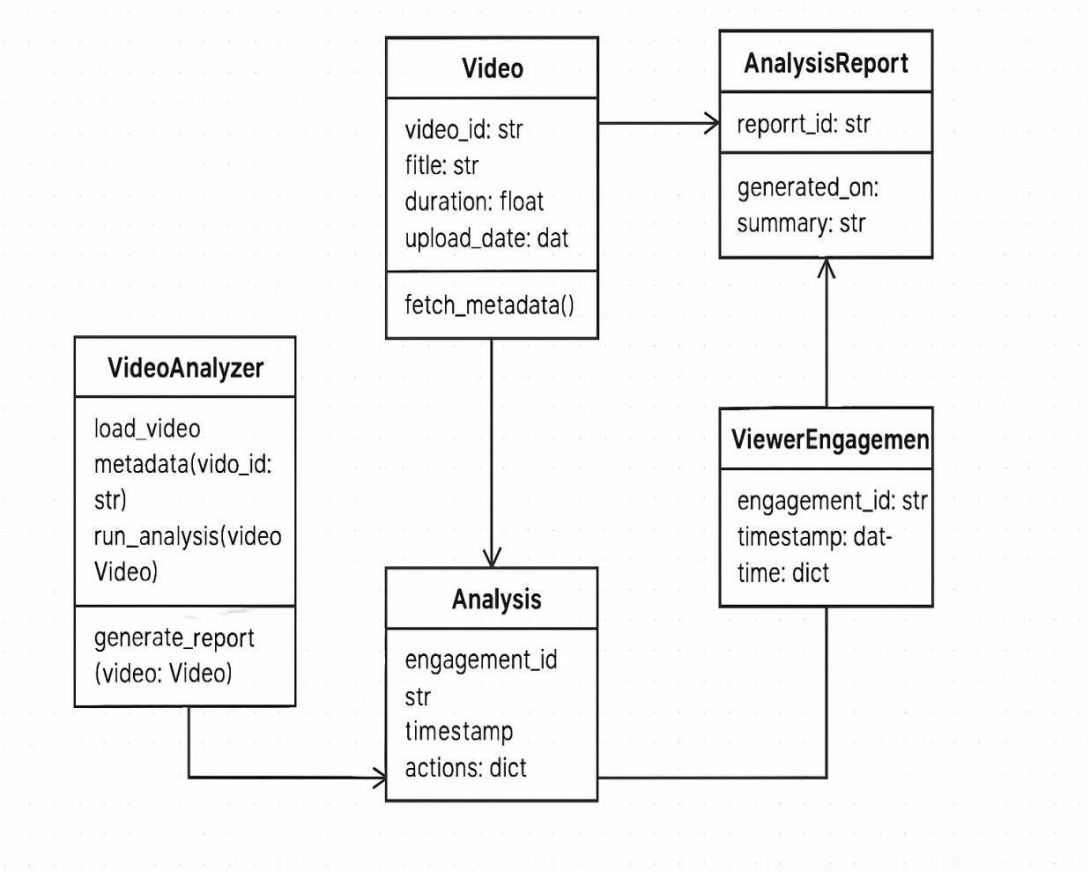

Figure 5.3.4 Class Diagram

6. CODING

6.1 Source code

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
from scipy import stats
from sklearn import preprocessing
from sklearn.preprocessing import scale
import plotly.express as px
warnings.filterwarnings('ignore')
sns.set(style="whitegrid")
categories = pd.read_json("IN_category_id.json")
categories
categories['items'][1]
category_dict = {}
for i in categories['items']: category_dict[i['id']]
   = i['snippet']['title']
category_dict
df = pd.read_csv("IN_youtube_trending_data.csv")
df.head()
df.describe() df.info()
df[df.video_id=="Iot0eF6EoNA"]
df.video_id.nunique()
df.channelTitle.nunique()
df.drop(["channelId","thumbnail_link"], inplace=True, axis=1)
df_maxViews = df.sort_values('view_count', ascending=False).drop_duplicates(['video_id'])
```

```
df_maxViews.sample(5)
df_minViews = df.sort_values('view_count', ascending=True).drop_duplicates(['video_id'])
df_minViews.sample(5)
categories['items'][10]['snippet']['title']
def fetch_video_category(value):
    return categories['items'][value]['snippet']['title']
    #add the categories to the various dataframes:
df_maxViews["Video_Category"]=df_maxViews.categoryId.apply(fetch_video_category)
df_minViews["Video_Category"]=df_minViews.categoryId.apply(fetch_video_category)
df["Video_Category"]=df.categoryId.apply(fetch_video_category)
df.sample(5)
df.Video_Category.value_counts()
df['publishedAt'] = pd.to_datetime(df['publishedAt'])
df['trending_date'] = pd.to_datetime(df['trending_date'])
df.dtypes
df.isnull().sum()
df['channelTitle'] = df['channelTitle'].fillna("Other") df['description']
= df['description'].fillna('No description provided') df.isnull().sum()
df = df.drop_duplicates('title',keep='last')
import pandas as pd
# ... (Your previous code for loading data, defining fetch_video_category, etc.) ...
# Make sure 'Video_Category' is created in df before creating df_maxViews
def fetch_video_category(value):
   try:
        return categories['items'][value]['snippet']['title']
   except IndexError:
       return 'Unknown'
df["Video_Category"] = df.categoryId.apply(fetch_video_category)
df_maxViews = df.sort_values('view_count', ascending=False).drop_duplicates(['video_id'])
```

```
                # Now, proceed with the groupby operations:
df_Views_Analysis =
df_maxViews.groupby('Video_Category')["view_count"].sum().reset_index()

df_Like = df_maxViews.groupby('Video_Category')["likes"].sum().reset_index()

df_Dislike = df_maxViews.groupby('Video_Category')["dislikes"].sum().reset_index()

df_Comments =
df_maxViews.groupby('Video_Category')["comment_count"].sum().reset_index()

df_Views_Analysis["Likes"] = df_Like.likes df_Views_Analysis["Dislikes"]

= df_Dislike.dislikes pd.set_option("display.max_rows", None,

"display.max_columns", None) print(df_Views_Analysis.to_string())

channel_group_df = df.select_dtypes(include=['number']).groupby(by =
df['channelTitle']).sum()

channel_group_df = df.select_dtypes(include=['number']).groupby(by =
df['channelTitle']).sum()

channel_group_df[channel_group_df['view_count'] == channel_group_df['view_count'].max()]

"""Channel With The Most View Count"""

plt.figure(figsize=(20, 18))

plt.subplot(2, 2, 1)

var_list = ['view_count', 'likes', 'dislikes', 'comment_count']

color_palette = ['orange', 'red', 'green', 'blue', 'purple']  # Define your color list

for i in range(0, 4):

    plt.subplot(2, 2, i + 1)

    x = channel_group_df[var_list[i]].nlargest(5).index

    y = channel_group_df[var_list[i]].nlargest(5)

    sns.barplot(x=x, y=y, palette=color_palette)  # Use the color list

    plt.xticks(rotation=45, ha='right')

    plt.title(f'Top 5 Channels by {var_list[i]}')

plt.tight_layout()

plt.show()

"""Most Watched Category"""

category_group_df = df.drop(columns=['publishedAt', 'trending_date']).groupby(by =
df['Video_Category']).sum()
```

```
category_group_df

category_group_df[category_group_df['view_count'] ==
category_group_df['view_count'].max()]

#Plotting the 5 categories with largest view count, likes, dislikes, comment_count}

plt.figure(figsize = (26,20))

plt.subplot(2,2,1)

var_list = ['view_count','likes','dislikes','comment_count']

color_palette = ['orange', 'red', 'green', 'blue', 'purple', 'pink']

for i in range(0,4):

    plt.subplot(2,2,i+1)

    x = category_group_df[var_list[i]].nlargest(6).index

    y = category_group_df[var_list[i]].nlargest(6)

    sns.barplot(x = x, y = y, palette=color_palette)

    plt.xticks(rotation=45, ha='right')

    plt.title(f'Top 6 Categories by {var_list[i]}')

plt.tight_layout()

plt.show()

"""Least Watched Category""

import matplotlib.pyplot as plt

plt.figure(figsize=(26, 18))

var_list = ['view_count', 'likes', 'dislikes', 'comment_count']

for i in range(0, 4):

    plt.subplot(2, 2, i + 1)

    x = category_group_df[var_list[i]].nsmallest(5).index

    y = category_group_df[var_list[i]].nsmallest(5)

                # Create the pie chart

    plt.pie(y, labels=x, autopct='%1.1f%%', startangle=90)

    plt.title(f'Bottom 5 Categories by {var_list[i]}')

plt.tight_layout() plt.show()

df_Views_Analysis["Response_Percentage"] = round((( df_Views_Analysis.Likes +
```

```
df_Views_Analysis.Dislikes ) /df_Views_Analysis.view_count)* 100,2)
df_Views_Analysis
"""
Public Response vs Type of Videos"""
plt.figure(figsize=(20, 10))
plt.title("Public Response vs Type of Videos")
plt.xticks(rotation=90)
# Display the Dashboard
display(dashboard_layout)
plt.figure(figsize=(20, 10))
plt.title("Public Response vs Type of Videos")
plt.xticks(rotation=90)
# Define your color palette
color_palette = 'viridis'  # Example palette, you can choose others
sns.barplot(x=df_Views_Analysis.Video_Category,
        y=df_Views_Analysis.Response_Percentage,
        palette=color_palette)  # Add the palette argument
plt.ylabel("Public Response (in percentage)")
plt.show()
df_maxViews[df_maxViews.Video_Category.isin(["Sports",  "Classics"])].sample(5)
df_Views_Analysis["LikesPercentage"] =
df_Views_Analysis.Likes*100/df_Views_Analysis.view_count
df_Views_Analysis["DislikesPercentage"] =
df_Views_Analysis.Dislikes*100/df_Views_Analysis.view_count
df_Views_Analysis["Comments"] = df_Comments["comment_count"]
df_Views_Analysis["CommentsPercentage"] =
df_Views_Analysis.Comments*100/df_Views_Analysis.view_count
# Set the width and height of the figure
plt.figure(figsize=(20,10))
df_Views_Analysis.plot(x="Video_Category", y=["LikesPercentage", "DislikesPercentage",
"CommentsPercentage"], kind="bar",figsize=(20,10))
"""CommentsPercentage vs Type of Videos"""
```

```
plt.figure(figsize=(20, 10))
plt.title("CommentsPercentage vs Type of Videos")
plt.xticks(rotation=90)
# Create the line plot with a thicker line
sns.lineplot(x=df_Views_Analysis.Video_Category,
             y=df_Views_Analysis.CommentsPercentage,
             linewidth=5)  # Set linewidth to 5 for a thicker line

plt.ylabel("Comments")
plt.show()
"""LikesPercentage vs Type of Videos"""
plt.figure(figsize=(20, 10))
plt.xticks(rotation=90)
plt.title("LikesPercentage vs Type of Videos")
# Use a different color palette
sns.barplot(x=df_Views_Analysis.Video_Category,
            y=df_Views_Analysis.LikesPercentage,
            palette="viridis")  # Example palette, you can choose others
plt.ylabel("Likes")
plt.show()
```

7. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

7.1. Types Of Tests

Unit tesing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application

Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Inputs: contain all required fields (company Name, role, required Skills, and resume) with different scenarios to cover cases like missing company name, missing role, missing resume, and various skill checks.

Output: provide meaningful feedback according to the input scenario: confirmation of suitability or identification of missing fields or skills.

Functions:

checkResume(),document.getElementById(),split(), trim(), toLowerCase().

Integration testing

Integration tests are designed to test integrated software components to determine if they actually runas one program. Testing is event driven and is more concerned with the basic outcome of screens orfields. Integration tests demonstrate that although the components were individually satisfaction, asshown by successfully unit testing, the combination of components is correct and consistent.

Functional testing is centered on the following items:

Organization and preparation of functional tests is focused on requirements, key functions, or specialtest cases. In addition, systematic coverage pertaining to identify Business process flows; data fields,predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure orlanguage of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated softwarecomponents on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. componentsin a software system or – one step up – software applications at the company level – interact withouterror.

Test Results**:** All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results**:** All the test cases mentioned above passed successfully. No defects encountered.

7.2 Test Cases:

S.no	Test Case	Excepted Result	Result	Remarks (IFFails)
1.	Data Loading	Video data CSV and JSON category data should load correctly	Pass	Executed successfully
2.	Data Cleaning	Handle missing values, drop duplicates, remove unused columns	Pass	Cleaning steps applied correctly
3.	Category Mapping	Categories should be mapped from JSON file to dataframe	Pass	Category names mapped successfully
4.	Max/Min View Extraction	Extract videos with max and min views per video ID	Pass	Achieved using sorting and drop_duplicates()
5.	Video Category Distribution	Count and visualize videos per category	Pass	Distribution and counts computed
6.	Time Conversion	Convert publishedAt and trending_date to datetime	Pass	Datetime conversion successful
7.	Group by View Count& Engagement Metrics	Group and sum views, likes, dislikes, and comments by category	Pass	Metrics grouped and merged into final dataframe
8.	Response Percentage Calculation	Calculate (likes + dislikes) / views	Pass	Values added to df_Views_Analysis
9.	Visualization - Top Channels	Show bar charts of top 5 channels by metrics	Pass	All 4 charts displayed with proper labels
10.	Visualization - Top Categories	Show top 6 categories by metrics	Pass	Bar plots rendered correctly

Table 7.2 Test cases

8. RESULT

	video_id	title	publishedAt	channelId	channelTitle	categoryId	trending_date	tags	view_count	likes	dislikes	comment_count	thumbnail_link	comments_disabled	ratings_disabled
0	Iot0eF6EoNA	Sadak 2 | Official Trailer | Sanjay | Pooja | ...	2020-08-12T04:31:41Z	UCGqvJPRcv7aVFun-eTsatcA	FoxStarHindi	24	2020-08-12T00:00:00Z	sadak|sadak 2|mahesh bhatt|vishesh films|pooja...	9885899	224925	3979409	350210	https://i.ytimg.com/vi/Iot0eF6EoNA/default.jpg	False	False
1	x-KbnJ9fvJc	Kya Baat Aa : Karan Aujla (Official Video) Tan...	2020-08-11T09:00:11Z	UCm9SZAI03Rev9sFwloCdz1g	Rehaan Records	10	2020-08-12T00:00:00Z	[None]	11308046	655450	33242	405146	https://i.ytimg.com/vi/x-KbnJ9fvJc/default.jpg	False	False
2	KX06ksuS6Xo	Diljit Dosanjh: CLASH (Official) Music Video | ...	2020-08-11T07:30:02Z	UCZRdNleCgW-BGUJf-bbjzQg	Diljit Dosanjh	10	2020-08-12T00:00:00Z	clash diljit dosanjh|diljit dosanjh|diljit dos...	9140911	296533	6179	30058	https://i.ytimg.com/vi/KX06ksuS6Xo/default.jpg	False	False
3	UsMRgnTchY	Dil Ko Maine Di Kasam Video | Amaal M Ft.Ariji...	2020-08-10T05:30:49Z	UCq-Fj5jknLsUf-MWSy4_brA	T-Series	10	2020-08-12T00:00:00Z	hindi songs|2020 hindi songs|2020 new songs|t-...	23564512	743931	84162	136942	https://i.ytimg.com/vi/UsMRgnTchY/default.jpg	False	False
4	WNSEXJJhKTU	Baarish (Official Video) Payal Dev,Stebin Ben ...	2020-08-11T05:30:13Z	UCye6Oz0mg46S362LwARGVcA	VYRLOriginals	10	2020-08-12T00:00:00Z	VYRL Original|Mohsin Khan|Shivangi Joshi|Payal...	6783649	268817	8798	22984	https://i.ytimg.com/vi/WNSEXJJhKTU/default.jpg	False	False

Figure 8.1: data loading

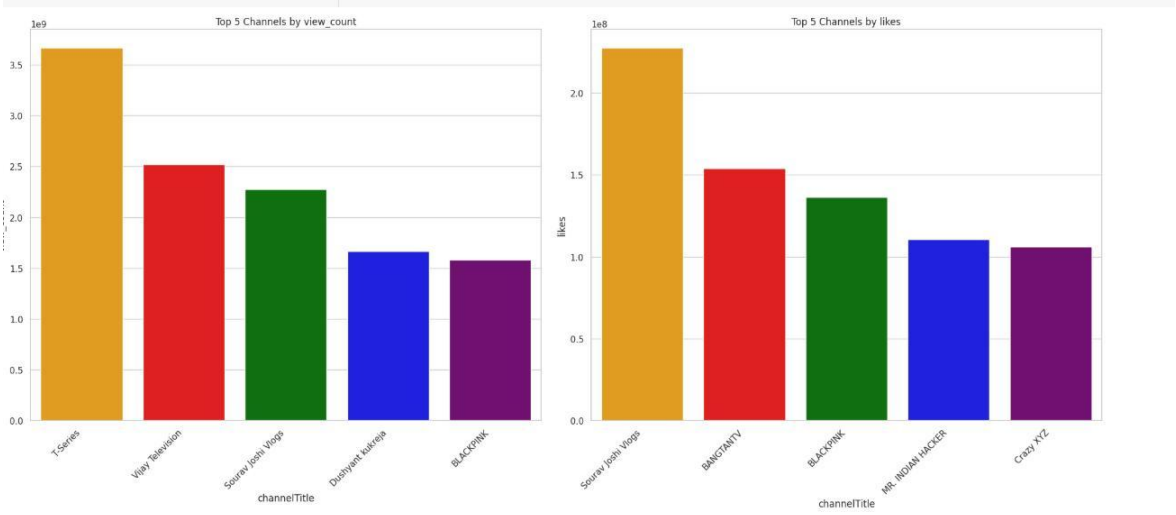

Figure 8.2: view ,likes count

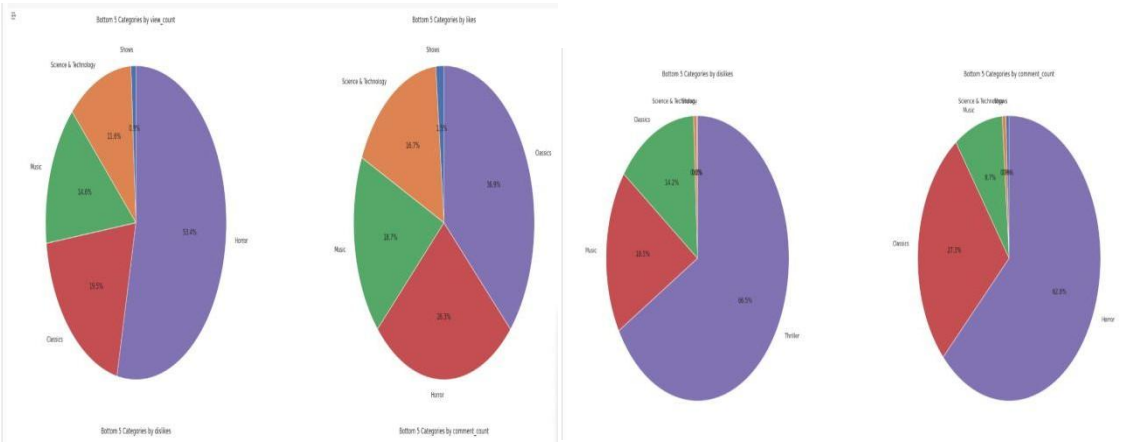

Figure 8.3: categories Maping

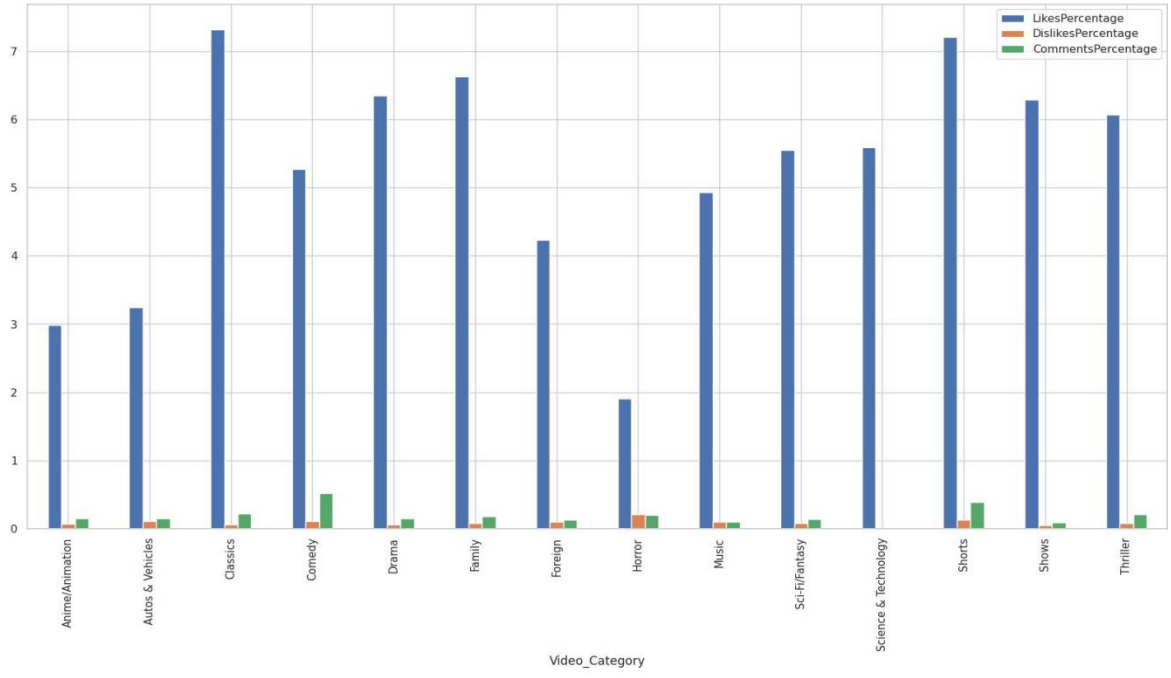

Figure 8.4: Max/Min View Extraction

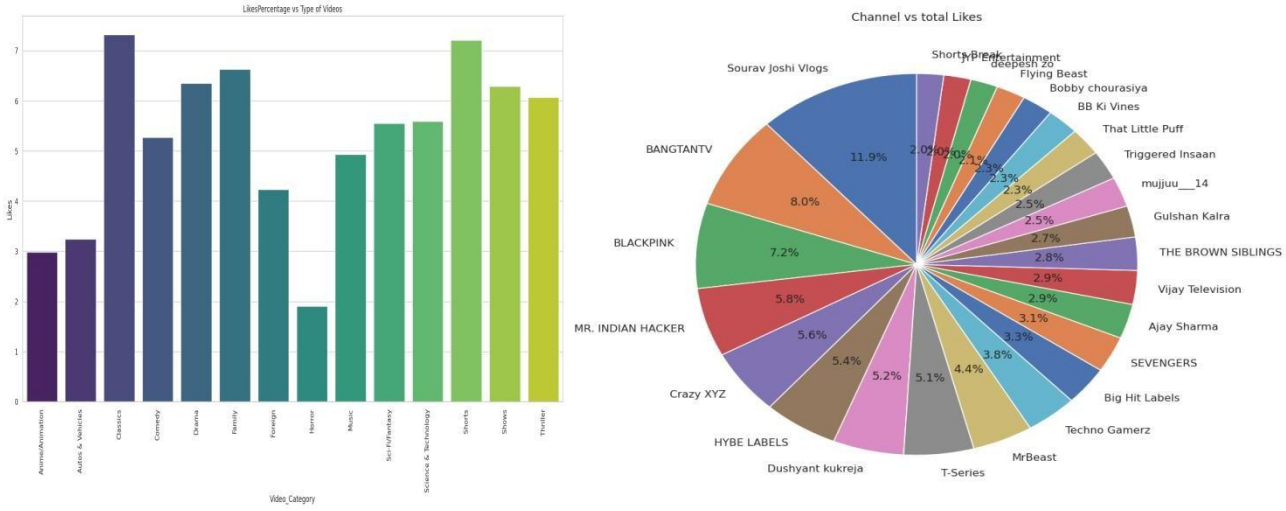

Figure 8.5: Video Category Distribution

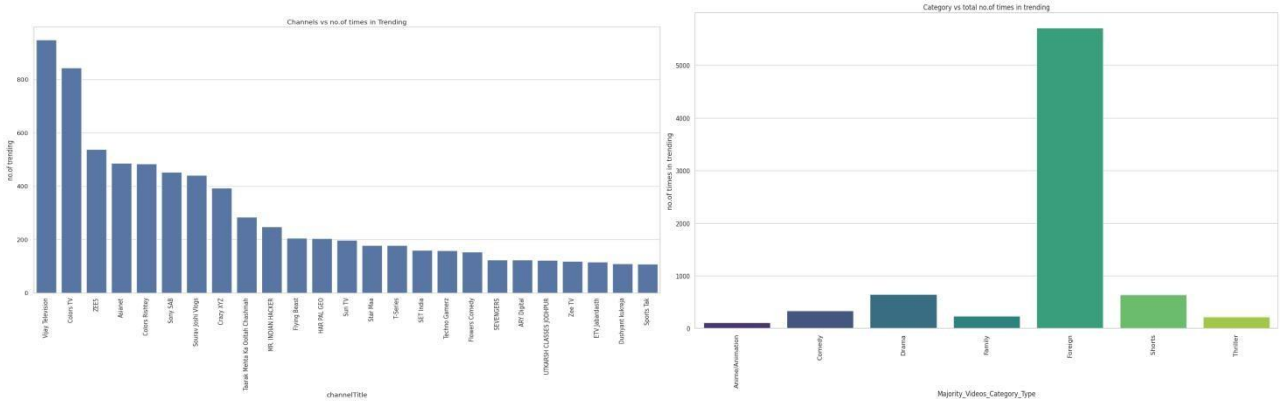

Figure 8.6: Time Conversion

9. CONCLUSION

The Video Data Analytics project provides a comprehensive understanding of trending YouTube content by analyzing video metadata such as view count, likes, dislikes, comments, category, and publication date. By preprocessing and enriching the dataset with category labels and removing inconsistencies, the project set a strong foundation for meaningful visualizations and insights.

Through statistical grouping and graphical analysis, the project identified which video categories gain the most attention. Categories like *Music*, *Entertainment*, and *News & Politics* were found to dominate in terms of both viewership and public engagement. These insights help content creators and marketers understand the types of content that tend to trend on the platform.

Channel-level analysis revealed that a limited number of channels repeatedly produce trending content, attracting high views and engagement. Metrics such as total likes, comments, and the number of days a video remained trending were analyzed to assess channel influence and consistency.

Public engagement was measured using calculated metrics like Likes Percentage, Dislikes Percentage, and Comments Percentage. These revealed that certain categories—such as *Comedy* and *Gaming*—encourage more audience interaction, even if their total views are comparatively lower. This distinction is critical for identifying high-engagement content versus high-reach content.

The project also featured an interactive dashboard using `ipywidgets` and `Plotly`, enabling users to dynamically explore the most viewed channels and trending categories. Although some dashboard functionalities (like additional plot buttons) remain incomplete, the framework is well-positioned for extension into a real-time analytics tool.

In conclusion, this project highlights the power of data analytics in decoding video trends and viewer behavior on YouTube. The findings can guide content strategy, improve audience targeting, and ultimately optimize visibility and performance in digital video platforms.

10. FUTURE ENHANCEMENTS

The Video Data Analytics project can be significantly expanded with several meaningful enhancements. Integrating real-time data using the YouTube Data API would allow the system to monitor trending videos as they appear, keeping the analysis current and dynamic. Incorporating sentiment analysis through natural language processing on user comments would provide deeper insight into public opinion beyond simple metrics like likes or dislikes. Predictive modeling using machine learning could be introduced to forecast the likelihood of a video trending based on early indicators such as initial views, likes, upload timing, and category.

Enhancing the dashboard with full interactivity using tools like Streamlit or Dash would offer a more intuitive and user-friendly experience, allowing users to filter data by category, region, date, or engagement level. Incorporating geographic and demographic information could unlock region-specific or audience-specific insights, helping creators tailor their content more effectively. Tracking the lifecycle of trending videos—such as how many days they remain trending—and correlating that with features like title length, posting time, or thumbnail quality could provide valuable strategic guidance.

Expanding the dataset to include video data from multiple countries would allow comparative analysis of global and regional trends. Further, the system could support automated report generation, creating weekly or monthly summaries of key metrics and trends in downloadable PDF or HTML formats. This would be particularly useful for analysts, marketers, or content strategists who need regular updates.

Lastly, the project could be extended to cover additional platforms such as TikTok, Instagram Reels, or Facebook Watch. This would make the system more comprehensive and relevant in a multi-platform digital ecosystem. Together, these enhancements would transform the project into a powerful and intelligent video analytics and strategy tool.

REFERENCES

[1] Jagad et al., "A Study on Video Analytics and Their Performance Analysis for Various Object Detection Algorithms," 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET),Arad,Romania,2022,pp.1095-1100, doi:10.1109/GlobConET53749.2022.9872506.

[2] Khatri, T. Choudhury, T. P. Singh and M. Shamoon, "Video Analytics Based Identification and Tracking in Smart Spaces," *2019 International Conference on contemporary Computing and Informatics (IC3I)*, Singapore, 2019, pp. 261-267, doi: 10.1109/IC3I46837.2019.9055614. .

[3] Xia *et al*., "Video Visualization and Visual Analytics: A Task-Based and Application-Driven Investigation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 11, pp. 11316-11339, Nov. 2024, doi: 10.1109/TCSVT.2024.3423402.

[4] D. Tran et al., "Learning Spatiotemporal Features with 3D Convolutional Networks," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[5] M. Jain, J. C. van Gemert, and C. G. M. Snoek, "What Do 15,000 Object Categories Tell Us About Classifying and Localizing Actions?" in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 46–55.

[6] F. Yu, W. Wang, and Y. Wang, "Predicting YouTube Video Popularity Based on Video Metadata," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, no. 10, pp. 205–216, Oct. 2016.

[7] A. Basharat, A. Gritai, and M. Shah, "Learning Object Motion Patterns for Anomaly Detection and Improved Object Detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[8] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 568–576.

[9] M. Gygli, H. Grabner, and L. Van Gool, "Video Summarization by Learning Submodular Mixtures of Objectives," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3090–3098.

[10] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 3551–3558.