

# TubePulse: Harnessing Youtube Insights Using The Cloud

Abhigna Sowgandhika Vadlamudi, 2268166, [avadlam2@cougarnet.uh.edu](mailto:avadlam2@cougarnet.uh.edu), Geethika Komma, 2263649, [gkomma2@cougarnet.uh.edu](mailto:gkomma2@cougarnet.uh.edu), Kausthubh Reddy Eleti, 2297857, [keleti@cougarnet.uh.edu](mailto:keleti@cougarnet.uh.edu)

**Abstract**—As we delve deeper into technology, the amount of data produced and transferred keeps increasing exponentially. This results in the continuous development of new methodologies to access this urge in data. This project aims to establish a secure and efficient data management and analysis system for structured and semi-structured YouTube video data, focusing on video categories and trending metrics. Key objectives include developing robust data ingestion mechanisms from diverse sources, implementing an Extract, Transform, Load (ETL) system for raw data transformation, establishing a centralized data lake for multi-source data storage, ensuring scalability to accommodate increasing data volumes, leveraging cloud computing capabilities (specifically AWS), and creating a reporting dashboard for insightful analysis.

**Index Terms**—Cloud Computing, Cloud Analytics, Data Management, AWS, Lambda Functions, Athena, Glue, ETL System, Optimization, Big Data, Content Strategy Optimisation, YouTube, Predictive Analysis, User Engagement

## I. INTRODUCTION

The era of digital content creation and consumption demands comprehensive solutions to harness large amounts of data. Our project achieves this by developing a framework to harness YouTube insights using cloud analytics. With the explosive growth of video content on YouTube, content creators and analysts face the challenge of efficiently managing, analyzing, and deriving meaningful insights from vast amounts of structured and semi-structured data. This project was born out of the necessity to provide a secure, streamlined, and analytical approach to handling YouTube video data, focusing on crucial metrics such as video categories and trending indicators.

The project addresses the critical need for content creators to enhance their strategies by understanding the performance of their content. It allows them to analyze user engagement, watch time, and ad performance, enabling informed decisions on content creation and optimization. Additionally, the project caters to advertisers by offering insights into audience reactions and sentiment through comment analysis, facilitating better decision-making regarding ad placements and monetization strategies. The need for a centralized and

organized system for YouTube analytics becomes evident, and TubePulse efficiently fulfills this requirement through its innovative approach.

This project achieves its objectives through a well-defined set of processes and a robust architecture built on AWS cloud services. Leveraging the AWS CLI for data ingestion, AWS Glue crawlers, and Lambda functions for ETL processes, TubePulse ensures the transformation of raw data into structured, clean information. The establishment of a data lake on Amazon S3 ensures centralized and organized storage, enhancing data retrieval and management efficiency. The project's scalability is a key feature, designed to handle high data volumes effectively. TubePulse utilizes Amazon QuickSight to create an intuitive reporting dashboard, offering user-friendly insights into YouTube video analytics. The project not only caters to the current needs of content creators and advertisers but also positions itself for future advancements, with proposed steps including scheduled ETL jobs, enhanced dashboard experiences, integration of machine learning for predictive analysis, and continuous monitoring and optimization for performance and cost efficiency. In essence, TubePulse not only addresses current challenges in YouTube analytics but also anticipates and prepares for the evolving landscape of digital content.

### A. PROJECT SCOPE

The Kaggle dataset, which includes regularly watched YouTube videos from several regions, will be the project's main emphasis. It will include information on a number of different factors, such as the video's title, channel name, publishing date, tags, views, likes, and dislikes, as well as the description, comment count, and categoryid.

The following features will be implemented:

**Data Ingestion Process:** Utilize AWS CLI for uploading raw data to Amazon S3 and create a centralized storage of YouTube video statistics.

**Develop an ETL System:** Implement AWS Glue crawlers for cataloging and Lambda functions for data transformation to result in structured and clean data ready for analysis.

**Create a Data Lake:** Establish a data lake on Amazon S3 for centralized and organized storage enabling efficient data retrieval and management.

**Scalability:** Design a scalable system capable of handling high data volume to ensure the system's ability to grow with expanding data.

**Enable Reporting:** Create a reporting dashboard using Amazon QuickSight for user-friendly insights into YouTube video analytics.

## B. USE CASES

**Content Strategy Optimization:** to analyze the performance of a user's content and develop a strategy to produce more appealing and relevant videos.

**Audience Segmentation:** helps in better understanding of the demographics and preferences of the audience which allows improved segmentation.

**User Engagement Analysis:** to analyze sentiment in comments and understand audience reactions.

**Ad-campaign and Monetization:** allows us to analyze viewer behavior, watch time, ad performance which can lead to better decisions on ad placements.

## II. RELATED WORKS

[1] Z. Cheng, et al. (2008). "Exploring User Behavior in YouTube":

This study delves into the intricate aspects of user behavior within the YouTube platform, offering insights into patterns of content consumption, engagement metrics, and factors influencing user preferences. By analyzing a vast dataset, the research contributes to a deeper understanding of user interactions with video content on YouTube.

[2] L. Golab, et al. (2011). "Issues in Data Stream Management":

This publication addresses challenges and complexities associated with managing data streams, highlighting issues related to real-time processing, scalability, and maintaining data consistency. By examining the unique characteristics of data streams, the work contributes valuable perspectives to the field of data management and analytics in dynamic, fast-paced environments.

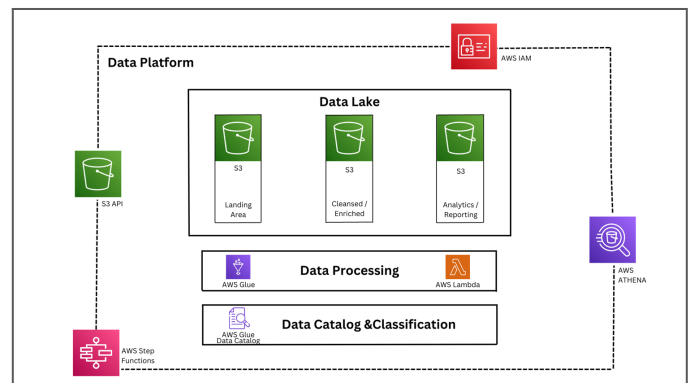
[3] S. Varia (2010). "Amazon Web Services: Overview of Security Processes":

Providing a comprehensive overview, this document outlines the security processes within Amazon Web Services (AWS). Covering aspects such as data protection, identity management, and infrastructure security, it serves as a crucial resource for understanding the foundational security measures implemented by AWS, an essential consideration for projects hosted on cloud platforms.

[4] A. Mayer-Schönberger, et al. (2009). "Big Data: A Revolution That Will Transform How We Live, Work, and

Think": This influential work explores the transformative impact of big data on various aspects of society, including daily life, work dynamics, and cognitive processes. It anticipates the profound societal changes resulting from the growing importance of big data analytics, making a compelling case for the revolutionary nature of the emerging data-driven paradigm.

## III ARCHITECTURE



## A. TOOLS USED

**Amazon S3:** It is a scalable object storage service designed to store and retrieve any amount of data from anywhere on the web. It provides developers with a highly durable and available storage infrastructure at a low cost, enabling them to store and retrieve data effortlessly. S3 is widely used for backup, data archiving, content distribution, and hosting static websites.

**AWS IAM (Identity and Access Management):** AWS IAM is a web service that enables secure access control to AWS resources. It allows you to manage and control access to various AWS services and resources securely. IAM enables the creation and management of AWS users and groups, granting them specific permissions to access AWS resources. This helps organizations implement the principle of least privilege and enhance the security of their AWS environments.

**Amazon QuickSight:** Amazon QuickSight is a cloud-based business intelligence service provided by AWS. It facilitates interactive and collaborative data visualization and analytics. QuickSight allows users to create and share insightful dashboards and reports by connecting to various data sources.

**AWS Glue:** AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for users to prepare and load their data for analysis. Glue automates the time-consuming tasks of data preparation and ETL, making it simpler to discover, catalog, and transform data stored in various sources. It supports both structured and unstructured data, providing a unified data integration platform.

**AWS Lambda:** AWS Lambda is a serverless computing service that allows developers to run code without provisioning or managing servers. Lambda executes code in

response to events, such as changes to data in an Amazon S3 bucket or updates to a DynamoDB table. It is widely used for building scalable and cost-effective applications, as users only pay for the compute time consumed during code execution.

**AWS Athena:** Amazon Athena is an interactive query service that allows users to analyze data stored in Amazon S3 using standard SQL queries. Athena eliminates the need for complex ETL processes by enabling users to query data in its raw, unprocessed form directly. It is a serverless and pay-per-query service, making it easy to analyze large datasets without the need for upfront infrastructure provisioning.

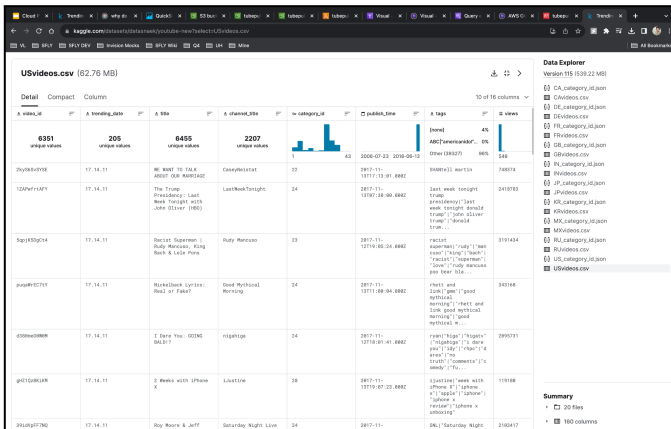
## IV METHODOLOGY AND IMPLEMENTATION

### A. DATA COLLECTION

The first phase of our project involved gathering the dataset from Kaggle, a renowned platform for sharing and discovering datasets.

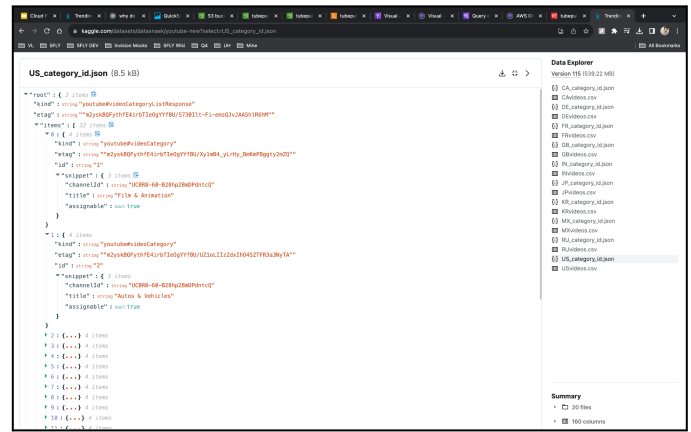
The specific dataset we utilized "Trending YouTube Video Statistics", provides daily records of trending YouTube videos, including information such as video ID, title, channel title, publish time, trending date, views, likes, dislikes, comment count, and trending region.

We utilized this comprehensive dataset as the foundation for our analysis of YouTube trends and user engagement.



The image shows a preview of the 'USVideos.csv' dataset (62.78 MB) in a data explorer interface. The table has 10 columns: video\_id, trending\_date, views, likes, dislikes, comment\_count, trending\_region, publish\_time, channel\_title, and video\_title. The first few rows of data are visible, showing video IDs, trending dates, and various engagement metrics.

Preview of USVideos.csv



The image shows a preview of the 'US\_Category\_id.json' dataset (0.5 KB) in a data explorer interface. The JSON structure contains an array of objects, each representing a video category. Each object includes fields like 'category\_id', 'video\_id', 'channel\_title', 'publish\_time', 'likes', 'dislikes', 'comment\_count', and 'trending\_region'.

Preview of US\_Category\_id.json

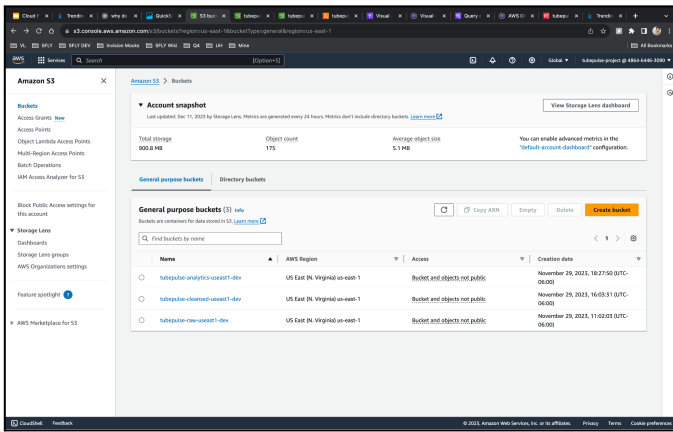
### B. DATA STORAGE

**Raw Data Storage (Raw S3 Bucket):** In the initial phase of our project, we established a Raw S3 bucket dedicated to storing the raw dataset obtained from Kaggle. This untouched, unaltered data serves as our primary source, preserving the original information and ensuring data integrity throughout the analysis process.

**Cleansed Data Storage (Cleansed S3 Bucket):** Upon collecting raw data, we proceeded to process and cleanse it, transforming it into a structured and standardized format. The cleansed dataset was then stored in the Cleansed S3 bucket in Apache Parquet format. This columnar storage format optimizes query performance and reduces storage costs, enhancing the efficiency of subsequent data processing stages.

**Analytics Data Storage (Analytics S3 Bucket):** The Analytics S3 bucket is the final repository for our processed and analyzed data. After the cleansed data undergoes structured changes, the resulting dataset is stored in this bucket. This curated dataset, tailored for analytics, serves as the foundation for generating visualizations, conducting exploratory analyses, and extracting meaningful conclusions. The structured changes applied here ensure that the analytics data is optimized for specific analytical queries and operations.

By structuring our data storage across these three S3 buckets and incorporating Apache Parquet for the cleansed data, we have implemented a systematic approach to data management. This approach not only maintains the integrity of the raw data but also optimizes storage efficiency and query performance, contributing to the overall effectiveness and reliability of our project.



*S3 Data Pool consisting of 3 S3 Buckets*

### C. DATA PROCESSING

**AWS Glue ETL Job for CSV to Parquet Conversion:** In the initial stage of data processing, we employed an AWS Glue ETL Job to transform the regional CSV data into a more efficient and faster-performing Apache Parquet format. This step ensures optimized data storage and facilitates faster query performance. This process involves schema mapping, data type conversion, and other transformations as specified in the ETL script. The resulting cleansed data is stored in the cleansed S3 bucket for further analysis.

```
job = Job(glueContext)
job.init(arguments['JOB_NAME'], arguments)
pushdown = "region in ('ca','gb','us')"

datasource =
glueContext.create_dynamic_frame.from_catalog(
    database = "tubepulse_raw", table_name =
    "raw_statistics", transformation_ctx =
    "datasource", push_down_predicate =
    pushdown)

appliedmapping = ApplyMapping.apply(frame =
datasource, mappings, transformation_ctx =
"appliedmapping")
resolvechoice = ResolveChoice.apply(frame =
appliedmapping, choice = "make_struct",
transformation_ctx = "resolvechoice")
datasink = resolvechoice.toDF().coalesce(1)
df_final_output =
DynamicFrame.fromDF(datasink, glueContext,
"df_final_output")
datasink =
glueContext.write_dynamic_frame.from_options(
    s(frame = df_final_output, connection_type
```

```
= "s3", connection_options = {"path":
"s3://tubepulse-cleansed-useast1-dev/youtub
e/raw_statistics/", "partitionKeys":
["region"]}, format = "parquet",
transformation_ctx = "datasink")

job.commit()
```

*Part of ETL Job Script*

### **AWS Lambda Function for JSON to Parquet Conversion:**

To handle the categorical JSON data, we set up an AWS Lambda Function in conjunction with an event trigger on the raw S3 bucket. Whenever new data is introduced into the raw bucket, the Lambda function automatically initiates the conversion process. The data is then transformed into Apache Parquet columnar format which is then stored in the cleansed S3 bucket.

```
def lambda_handler(event, context):
    bucket =
    event['Records'][0]['s3']['bucket']['name']
    key =
    urllib.parse.unquote_plus(event['Records'][
    0]['s3']['object']['key'],
    encoding='utf-8')

    df_raw =
    wr.s3.read_json('s3://{}/{}'.format(bucket,
    key))

    df_step_1 =
    pd.json_normalize(df_raw['items'])

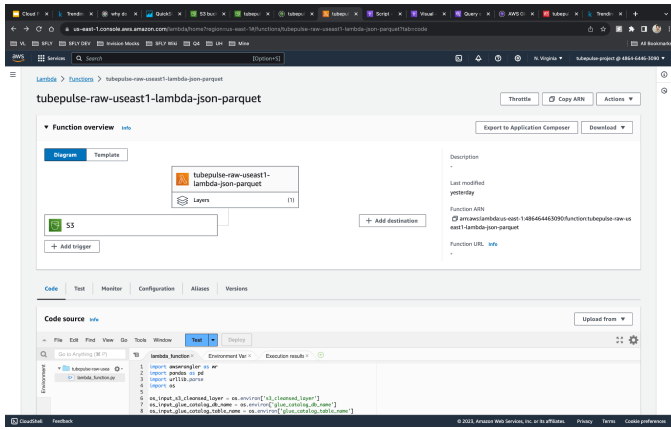
    wr_response = wr.s3.to_parquet(
        df=df_step_1,
        path=os_input_s3_cleansed_layer,
        dataset=True,

    database=os_input_glue_catalog_db_name,

    table=os_input_glue_catalog_table_name,
    mode=os_input_write_data_operation
    )

    return wr_response
```

*Lambda Function Code*



*Lambda Function*

**Benefits of Data Processing Approach:** The chosen approach utilizing AWS Glue ETL Jobs and Lambda Functions offers several advantages. Firstly, the use of the Parquet format optimizes the query performance. Additionally, the automation provided by AWS Lambda ensures a seamless and real-time data processing pipeline, reducing manual intervention. This combination of technologies not only enhances data efficiency but also lays the groundwork for scalable and agile analytics in our project.

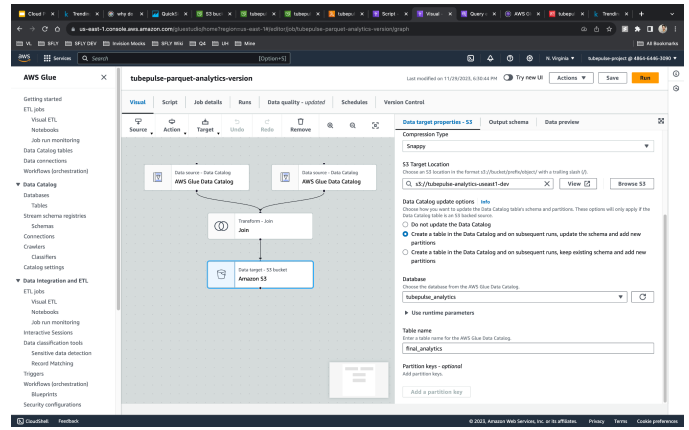
#### D. DATA ANALYSIS

**Data Joining using Athena Queries:** To streamline the data analysis process for our project, we implemented an efficient data joining mechanism using AWS Athena queries. The cleansed data consists of two key components - raw statistics and categorical data. The "tubepulse\_cleaned" database houses these tables, providing a structured foundation for analysis. To correlate the statistics and categorical data, we employed the INNER JOIN operation using AWS Athena queries. The SQL query used is as follows:

```
SELECT * FROM
"tubepulse_cleaned"."raw_statistics" a
INNER JOIN
"tubepulse_cleaned"."cleaned_statistics_ref
erence_data" b ON a.category_id=b.id;
```

This operation connects the two datasets based on the common identifier, 'category\_id,' facilitating a cohesive and integrated dataset for further analysis.

**Automation with ETL Job:** Recognizing the need for efficiency and automation, we implemented an ETL job that automates the INNER JOIN operation and stores the resulting joined output data in the Analytics S3 bucket. By doing so, we ensure that the most up-to-date and analyzed dataset is readily available for our analytics processes.



*ETL Job for Outputting the Joined Data for Analytics*

The integration of ETL Job significantly reduces the time and effort required for manual query execution. With the ETL job in place, the joined output data is seamlessly delivered to the Analytics S3 bucket, creating a streamlined and automated workflow for our data analysis endeavors.

#### E. DATA VISUALIZATION

Data analysis is a crucial aspect of our project, and we harnessed the capabilities of AWS QuickSight for effective data visualization. This powerful tool enabled us to create dynamic and insightful visual dashboards for a comprehensive analysis of the dataset stored in our Analytics DB using Athena.

**Integration with Analytics DB:** AWS QuickSight seamlessly integrates with Athena, allowing us to directly connect to the Analytics DB where our Youtube Analytics Dataset resides.

**Dashboard Creation:** We leveraged the user-friendly interface of AWS QuickSight to design multiple visual dashboards tailored to our analytical needs. These dashboards serve as interactive hubs for exploring various dimensions of YouTube data, offering a user-friendly experience for both technical and non-technical stakeholders.

**Key Visualizations:** Our dashboards feature a range of key visualizations, including bar charts and pie charts, to convey different aspects of the data. We employed these visual elements to illustrate trends, patterns, and correlations within the YouTube dataset, enhancing our ability to extract meaningful insights.

## V. RESULTS AND DISCUSSION

TubePulse, a comprehensive framework has thus successfully enabled harnessing YouTube insights using cloud analytics. The project addresses the growing challenge of managing and analyzing vast amounts of structured and semi-structured YouTube video data. By focusing on key metrics such as video categories and trending indicators, TubePulse provides content creators and advertisers with valuable insights into user engagement, content performance, and ad optimization.

The architecture of TubePulse, built on AWS cloud services, ensures secure and efficient data management. The use of AWS Glue ETL Jobs and Lambda Functions for data processing, along with the implementation of data lakes on Amazon S3, guarantees scalability and performance optimization. The reporting dashboard created with Amazon QuickSight offers an intuitive interface for users to analyze YouTube video analytics effortlessly.

The project's significance lies not only in its current capabilities but also in its preparedness for future advancements. With proposed enhancements such as scheduled ETL jobs, improved dashboard experiences, integration of machine learning for predictive analysis, and continuous monitoring and optimization for performance and cost efficiency, this application can be made to adapt to the evolving landscape of digital content.

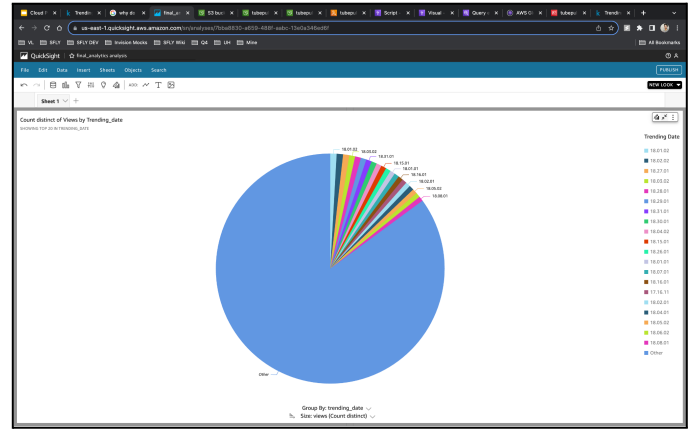
### Few Visual Dashboards by AWS QuickSight:

Attached below are a few visual representations of the dashboards developed using AWS Quicksight.

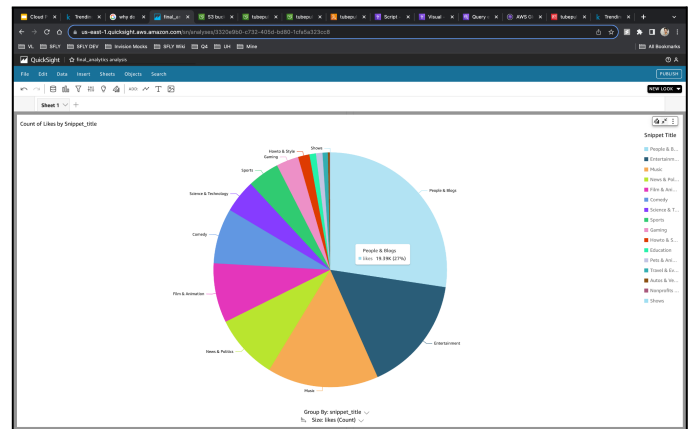
The first figure represents the number of likes a video receives according to the users in a particular region.

The second figure displays a pie chart of the views received based on the date the videos were trending.

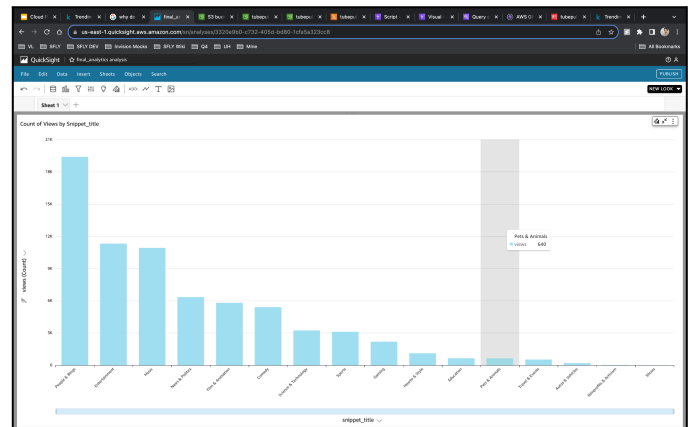
The third figure visualizes the number of likes by the title while the last figure displays the views by the title.



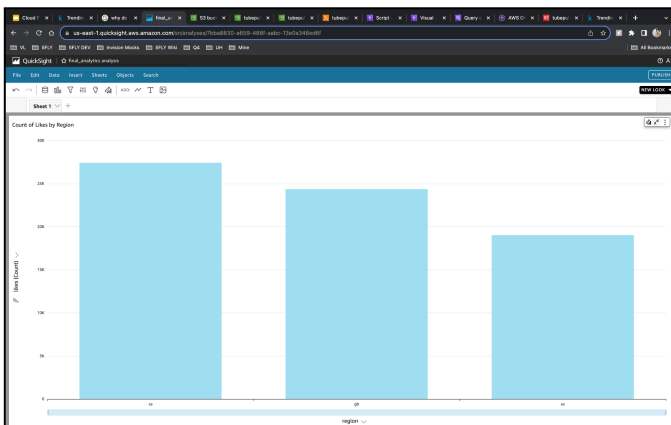
*Count of Views by Trending Date*



*Count of Likes by Snippet\_title*



*Count of Views by Snippet\_title*



*Count of Likes by Region*



## VI. CONCLUSION AND FUTURE WORK

**Scheduled ETL Jobs:** Implementing scheduled ETL jobs will automate the data processing pipeline, allowing for regular updates and real-time insights into YouTube trends.

**Enhanced Dashboard Experiences:** Improving the reporting dashboard in terms of interactivity, visualization options, and user customization will enhance the overall user experience and analytical capabilities.

**Machine Learning Integration:** Incorporating machine learning algorithms for predictive analysis can provide content creators and advertisers with valuable insights into future trends and user behavior, further optimizing content strategies and ad placements.

**Continuous Monitoring and Optimization:** Implementing a robust monitoring system to track system performance, identify bottlenecks, and optimize resource utilization will ensure TubePulse's efficiency and reliability over time.

**Cost Efficiency Optimization:** Continuously evaluating and optimizing the cost efficiency of the AWS services used in TubePulse will contribute to long-term sustainability and affordability.

In conclusion, TubePulse not only addresses the current challenges in YouTube analytics but also lays the foundation for continuous improvement and adaptation to the dynamic digital content landscape. The future work outlined above aims to further enhance this application's capabilities and ensure its relevance in the evolving world of online video content.