BAN 693

CAPSTONE PROJECT

Data pipeline & Analytics for YouTube Ad Recommendations

Using Video Categories

Submitted By:

Dileep Lingamallu

MS Business Analytics

elingamallu@horizon.csueastbay.edu

Contents

ABSTRACT

The explosion of online video content has led to an increased demand for tools and techniques to analyze viewer engagement, preferences, likings, and behavior. In this project, we explore the building of a serverless cloud-based data pipeline on AWS to analyze YouTube video statistics and draw key insights. Our pipeline leverages a range of AWS services, including Amazon S3 for data storage, AWS Lambda for serverless computing, Amazon Glue, Crawlers for data cataloging and Amazon Quicksight for data visualization.

Our pipeline enables the ingestion of YouTube data into our pipeline via AWS Command Line Interface, and which is then stored in Amazon S3. From there, the data is transformed using AWS Lambda and AWS Glue Studio and loaded into AWS Athena for analysis. Our pipeline also incorporates Tableau, providing users with an interactive dashboard to explore the insights generated from the data.

Through our implementation, we are able to analyze a range of video analytics metrics, including views, likes, tags, and sentiment. Our pipeline also provides insights into audience demographics and geographic distribution, enabling content creators and marketers to better understand their viewership and tailor their content to meet their needs.

Overall, our data pipeline on AWS provides a scalable and flexible solution for analyzing YouTube video analytics, with the potential for further customization and integration with other data sources.

PROBLEM STATEMENT

This project is inspired by the work of data engineers, who are responsible for making raw data more useful to the enterprise. The role requires a broad set of technical skills alongside the ability to communicate across departments that allow the data engineer to understand what business leaders want to gain from the company's data sets.

In addition to it, the rise of big data has led to an increased demand for scalable and efficient solutions for storing, processing, and analyzing data. As organizations strive to make sense of the vast amounts of data generated by their operations, they require powerful tools and technologies that can enable them to gain valuable insights and make informed decisions.

In response to this need, we have undertaken a capstone project to build a data pipeline on AWS. Our inspiration for this project comes from the recognition that AWS provides a wide range of services and tools that can be leveraged to develop a flexible and robust data pipeline that can handle diverse data types and support a range of analytics applications.

In this project we choose YouTube video dataset. YouTube, a platform which was founded as a free video sharing website in 2005, after spending a year focused on growing its user-base and make its presence felt globally; in 2008 it turned its focus on creating advertising formats that enabled its subscribers to monetize from their content on the site. While this was widely approved by its user community it was the following acquisition of DoubleClick which enabled Google to build an advertising ecosystem on YouTube.

YouTube tools today states that its advertisers currently have a reach to 2.476 billion users on YouTube daily, making it the world's 2nd most 'active' social media platforms. Moreover, tools show that YouTube's active users have grown quickly over the past years. The total number of users that advertisers can reach with ads on YouTube increased by roughly 185 million (-3.4%) in the twelve months leading up to July 2022.

With this project we built a serverless completely cloud-based data pipeline using various ETL, Big Data, Data Warehousing, Data Analytics, Data Mining techniques. The pipeline provides key insights and poses as a recommender model to recommend an appropriate AD for each video according to the most viewed category in a particular region.

DATASET

Our dataset for the project consists of analytical data on a large set of daily trending YouTube videos spanning over several months and several regions saved in several file formats. The dataset majorly consolidated into 2 different file formats.

Firstly, a set of 10 csv files separately for each region: United States (US), Great Britain (GB), Denmark (DE), California (CA), Russia (RU), Mexico (MX), South Korea (KR), Japan (JP), India (IN) and France (FR). Each CSV file lists details regarding the variables title, channel title, category_id, publish time, tags, views, likes and dislikes, description, and comment count for each video in it.

Secondly, a set of 10 JSON files separately for each region listed above: consisting of information related to what categories are associated with each category_id, which again varies from region to region.

METHODOLOGY

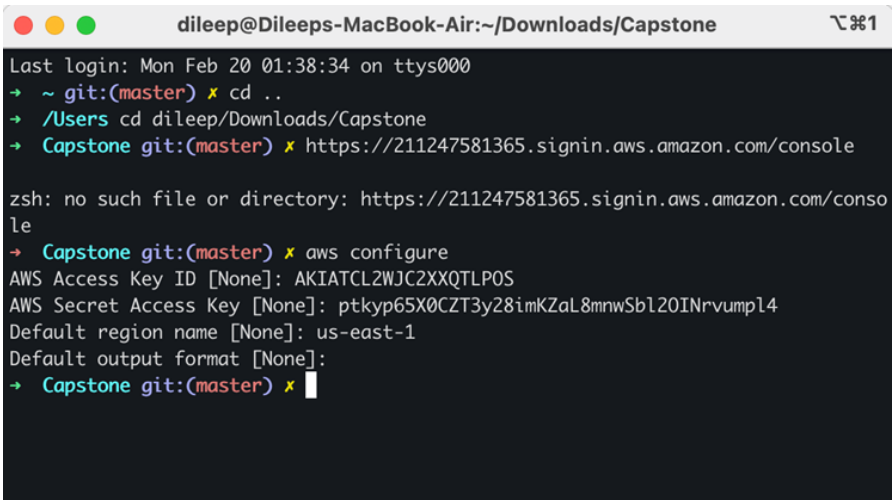
Step 1: Infrastructure Selection

After conducting extensive research on what infrastructure options are available to build our data pipeline, we came up with a few, specifically Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure. On further contemplation, we chose AWS as our infrastructure for the following reasons:

- **Comprehensive suite of services:** AWS offers a comprehensive suite of services for building data pipelines, including data storage, processing, analysis, and machine learning, enabling us to build a solution that could handle the full spectrum of our data pipeline needs.
- **Highly scalable infrastructure:** Given the quantum of the current chosen dataset and its extensibility into the future to incorporate real time streaming of YouTube video statistics through Kafka, there is a possibility of rapid growth in the dataset. AWS infrastructure could easily scale to handle the increasing volume of data. Its auto-scaling and load balancing features made it easy to scale up or down based on demand.
- **Cost-effective pricing:** AWS's pay-as-you-go pricing model was appealing, as it meant you only paid for the resources and services that we used. This helped in keeping infrastructure costs under control and allowing us to allocate resources more efficiently.
- **Strong security:** AWS has a strong security posture, with a wide range of security features such as encryption, access control, and monitoring. AWS's compliance with various industry standards and regulations, such as HIPAA and PCI-DSS, was also reassuring.

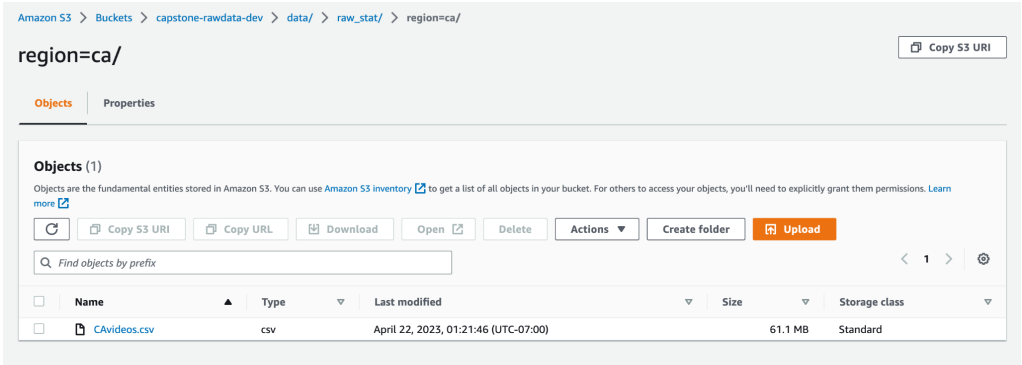
Step 2: Data Ingestion

- Created an Identity Access Management user with less access to the AWS console which is accessible by everyone in the team.
- Once the user was created, created the Access key and Secret Key.
- Configured the IAM user with the Access key and Security Key through AWS CLI to be able to access the AWS console programmatically as well.

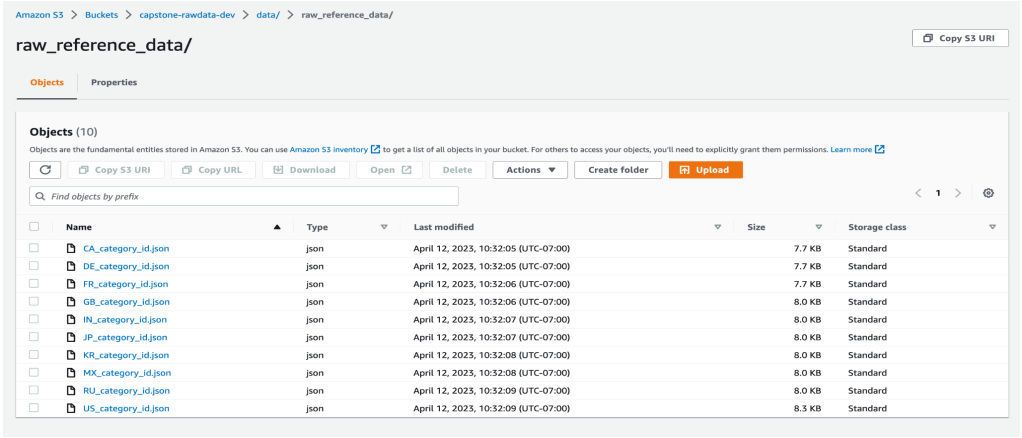


- Downloaded the YouTube video statistics dataset from Kaggle.
- Since the dataset consists of videos from other regions like Korea, Russia etc., and thereby their special characters, we converted the CSV files to UTF-8 encoding as it supports all Unicode characters.
- Created S3 bucket (capstone-rawdata-dev) and saved the 10 CSV files (as raw_stat/region=’region’/) and the 10 JSON files (as raw_reference_data) in the data subfolder within the bucket.

CSV:



JSON:



Step 3: Data Cataloging using AWS Glue Crawler for the 10 CSV files

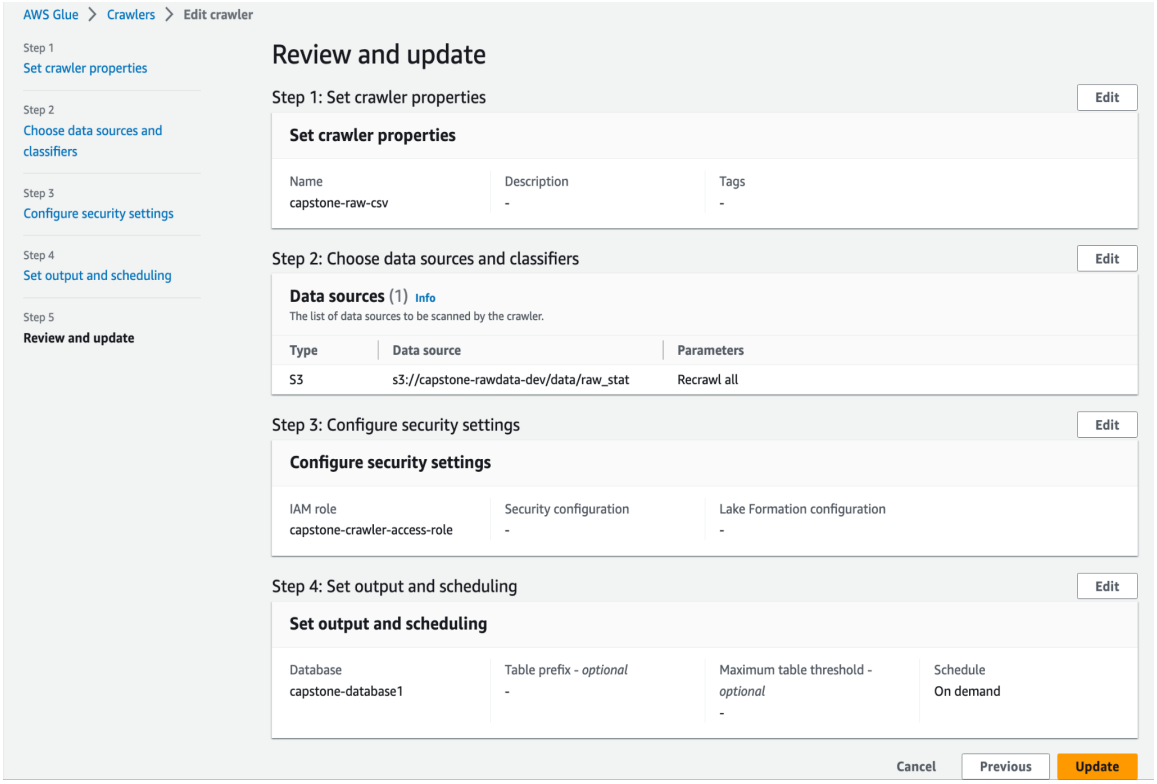
AWS Glue Crawler is a service provided by AWS that is used to automatically discover and catalog metadata about data stored in various data sources. Data cataloging using AWS Glue Crawler is important in a data pipeline for several reasons:

- **Automated discovery and cataloging of data:** The AWS Glue Crawler automates the process of discovering and cataloging metadata about data stored in various data sources. This makes it easier for users to analyze and query data without having to manually create and maintain data catalogs.
- **Improved data quality and accuracy:** Cataloging data using AWS Glue Crawler ensures that data is properly formatted and validated. This reduces errors and improves the overall quality and accuracy of data.
- **Facilitates data integration and interoperability:** AWS Glue Crawler identifies relationships between tables, which helps users create more complex queries that join data from multiple sources. This facilitates data integration and interoperability, making it easier to build data pipelines and automate data processing workflows.
- **Enables faster and more efficient data analysis:** By automatically cataloging data, AWS Glue Crawler eliminates the need for manual data cataloging, saving time and effort. This enables users to analyze and query data more quickly and efficiently, leading to faster and more informed decision-making.

The AWS Glue Crawler can be used to crawl a variety of data sources including Amazon S3, relational databases, and other cloud-based data stores. When the crawler runs, it examines the data in the specified location and creates a table schema that can be used to query the data.

We crawled the CSV files in the S3 bucket by following the below steps:

- Created AWS Glue Crawler (capstone-raw-csv).
- With Datasource as S3 bucket subfolder with the 10 CSV files (s3://capstone-rawdata-dev/data/raw_stat/).
- And the output destination is a newly created database named capstone-database1.



Step 4: Data Cataloging using AWS Glue Crawler and AWS Lambda for JSON files

We crawled the 10 JSON files in the S3 bucket by following the below steps:

- Created AWS Glue Crawler (capstone-Json-crawler).
- With Datasource as S3 bucket subfolder with the 10 JSON files (s3://capstone-rawdata-dev/data/raw_reference_data/).
- And the output destination is a newly created database named capstone-database2.

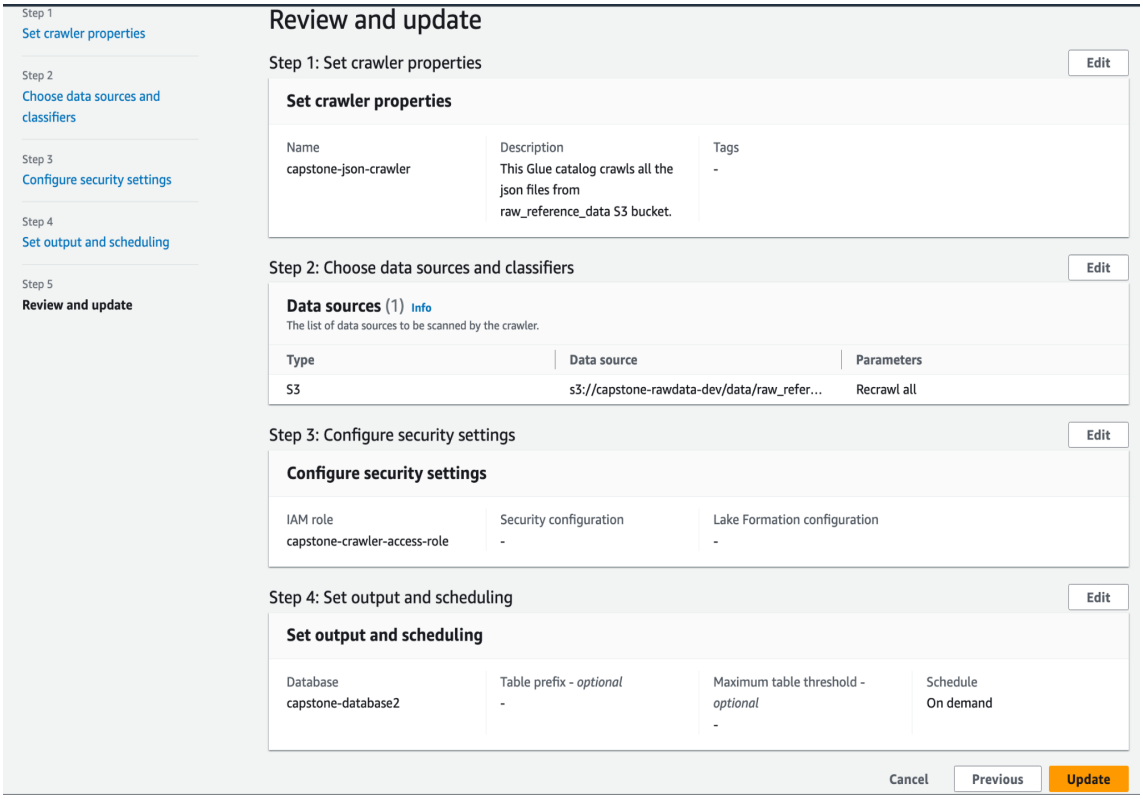
However, on loading the JSON database on AWS Athena, we realized that we were not able to query the database. The JSON files needed some pre-processing because the Glue Catalog was unable to understand the JSON files format. For this purpose, the JSON files had to be flattened first, stored in an S3 bucket and then generate a database for it. This was all achieved by converting the JSON data files to big data file formats using AWS Lambda, thereby, to COLUMN and ROW data in the following steps:

This was achieved in the following steps:

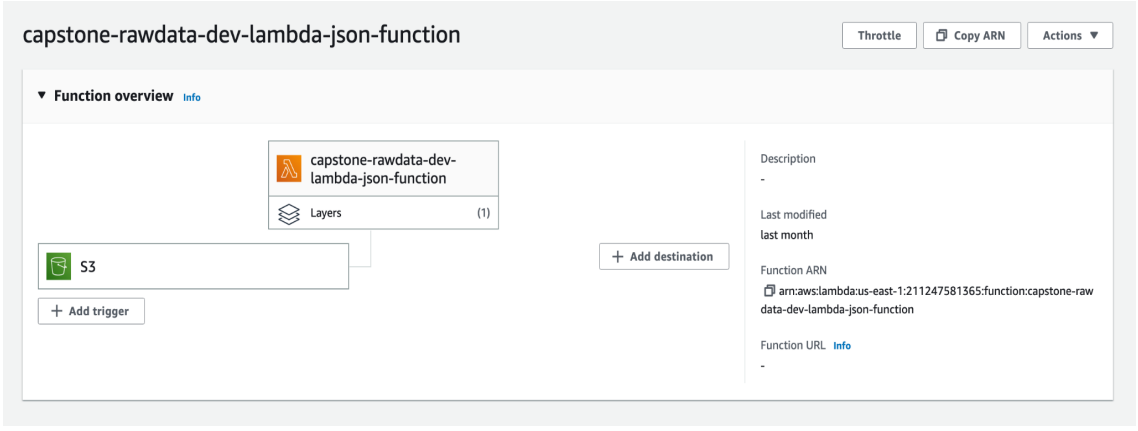
- Created an AWS Lambda function called capstone-rawdata-dev-lambda-json-function.

- The Lambda script read all the raw JSON files stored in the S3 bucket capstone-rawdata-dev, one by one.
- Converted them to Apache Parquet file formats.
- Wrote them back to another S3 bucket created and named, capstone-rawdata-json-cleansed, while also generating a data catalog in the new database created and named, cleansed-json-data.

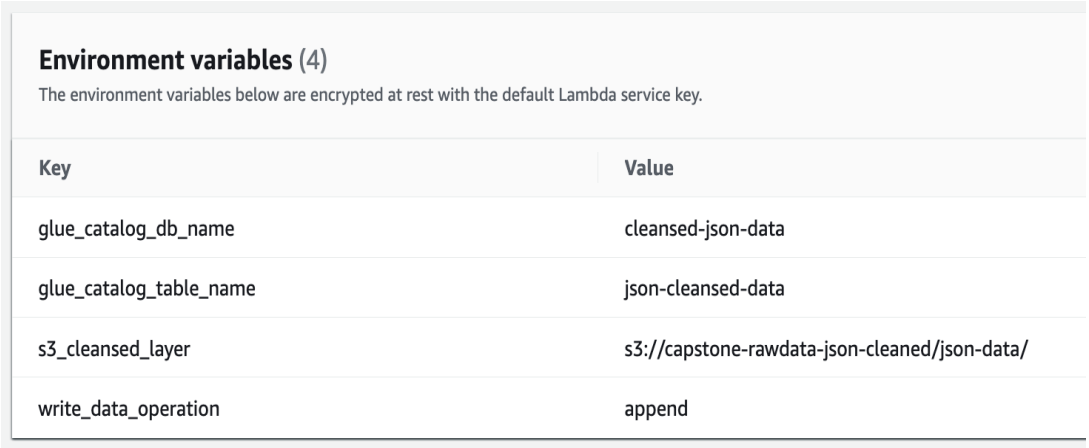
Crawler for JSON Files:



Lambda Function for JSON:



Environment Variables showing the operations run through Lambda:



Environment variables (4)

The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
glue_catalog_db_name	cleansed-json-data
glue_catalog_table_name	json-cleansed-data
s3_cleansed_layer	s3://capstone-rawdata-json-cleaned/json-data/
write_data_operation	append

Step 5: Sentiment Analysis using AWS Comprehend and Glue ETL Job

Today consumers are encouraged to express their satisfaction or frustration with a company or product through social media, blogs, and review platforms. Sentiment analysis can help companies better understand their customers' opinions and needs and make more informed business decisions.

With that in mind, we intended to implement a sentimental analysis for each video in the dataset based on its tags and determine key insights based on the sentiment analysis as well using AWS Comprehend.

AWS Comprehend is a natural language processing (NLP) service provided by AWS. It enables users to analyze and extract meaningful insights from unstructured text data such as emails, social media posts, customer reviews, and more. AWS Comprehend uses advanced machine learning algorithms to identify key phrases, entities, sentiment, and topics within text data, providing users with a deeper understanding of their data by categorizing the reviews as Positive or Negative or Neutral.

While AWS Comprehend is a powerful tool, it still comes with its own limitations. The maximum document size that can be processed through AWS comprehend at a single job for real-time analysis to detect sentiments is only 5KB. Thus, to simply pass through the data as a single AWS comprehend job, we would have to break the dataset into smaller chunks of 5KB. However, given the huge size of dataset we have, it would lead to 96,900 chunks which is not feasible.

In order to overcome the above size limitations, we used a AWS Glue ETL service leveraging the power of Apache Spark in the following ways:

- Firstly, converted the CSV files in the capstone-rawdata-dev S3 bucket to Apache Parquet file format (for faster processing with limited use of system processes) using an AWS ETL job (capstone-cleansed-csv-to-parquet).
- The ETL job reads the CSV data from the database capstone-databse1, converts them to Apache Parquet files in a new S3 bucket created and named capstone-rawdata-csv-cleansed.
- Secondly, scripted an ETL Pyspark script named, sentiment-ETL-US that read the Apache Parquet files and called the Comprehend bulk API for each video row to extract its sentiment.
- The Comprehend bulk API takes up to 25 strings at a time and enables batch processing of the entire dataset.
- The ETL job then ran the AWS Comprehend sentiment analysis based on the video's tags.
- Wrote the output of the ETL job to a new S3 bucket created and named, capstone-sentiment-analysis again in Apache Parquet format.

AWS Comprehend supports the following languages:

While all the various languages in our dataset are covered by AWS Comprehend, our dataset (for regions other than us, ca, gb and de) consists of a combination of english and local language characters. This combination of characters disorients the AWS comprehend tool and does not get processed.

Thus, given the limitations, we limited the AWS Comprehend Sentiment Analysis to the region folders us, ca, gb and de.

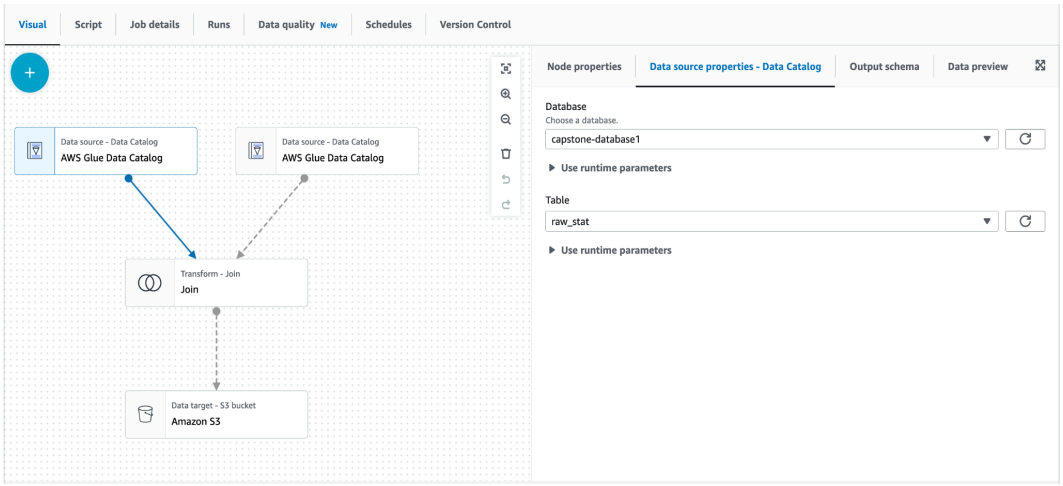
Step 6: Data Integration with AWS Glue Studio

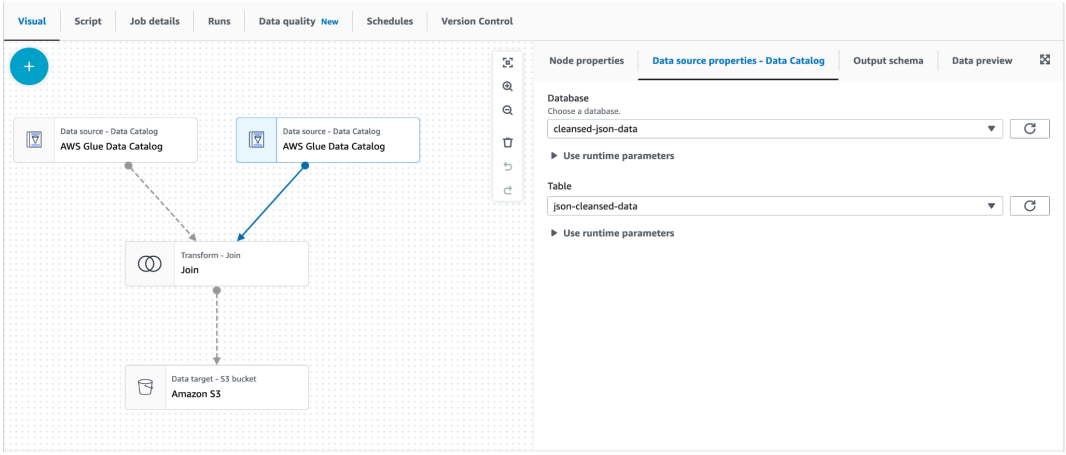
On completing all the steps from 1-5 above, we had all the cleansed CSV files, CSV-Parquet files, JSON-parquet files saved in separate S3 buckets.

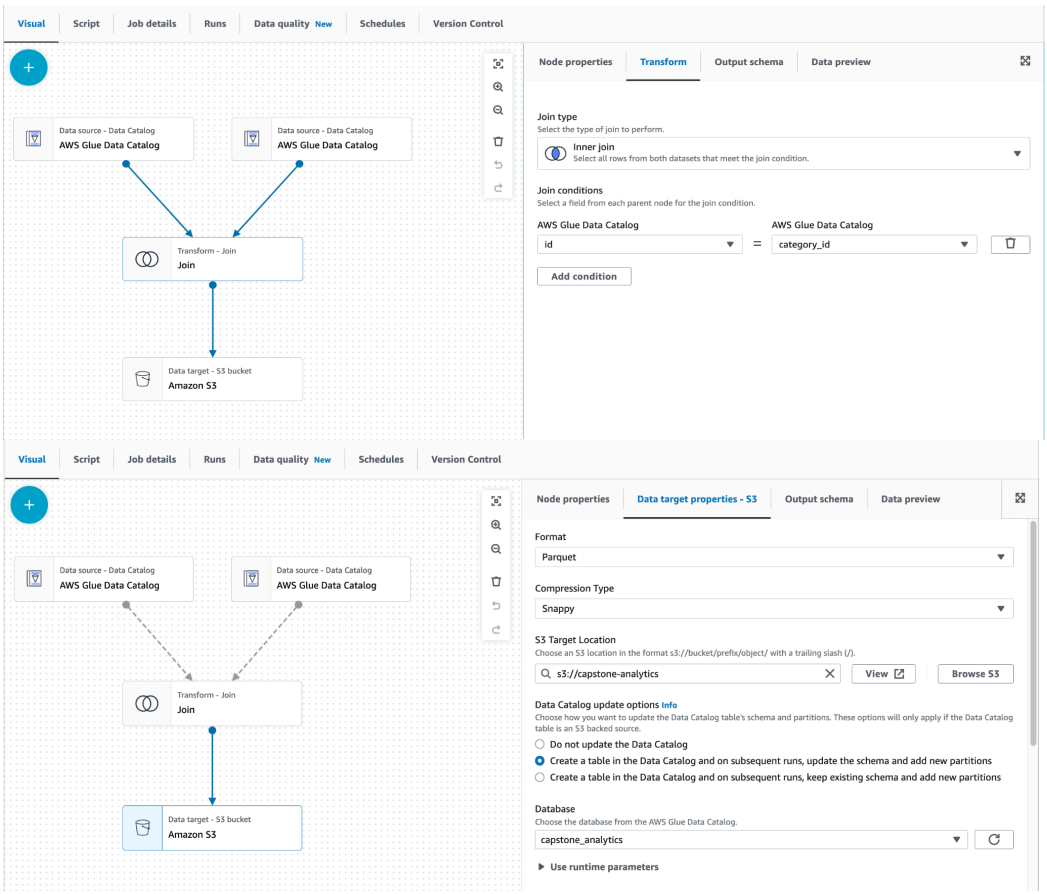
To get a combined output from all the files we loaded the CSV data database and the JSON-parquet data database to AWS Athena, and queried the entire dataset using a INNER JOIN between category_id field in CSV files and the id field in JSON-parquet file.

However, given that we intended to query this combined data multiple times and in multiple ways, the JOIN operation is not very cost-efficient. In order to avoid repetitive JOINs, we integrated the files using AWS Glue Studio by following the below steps:

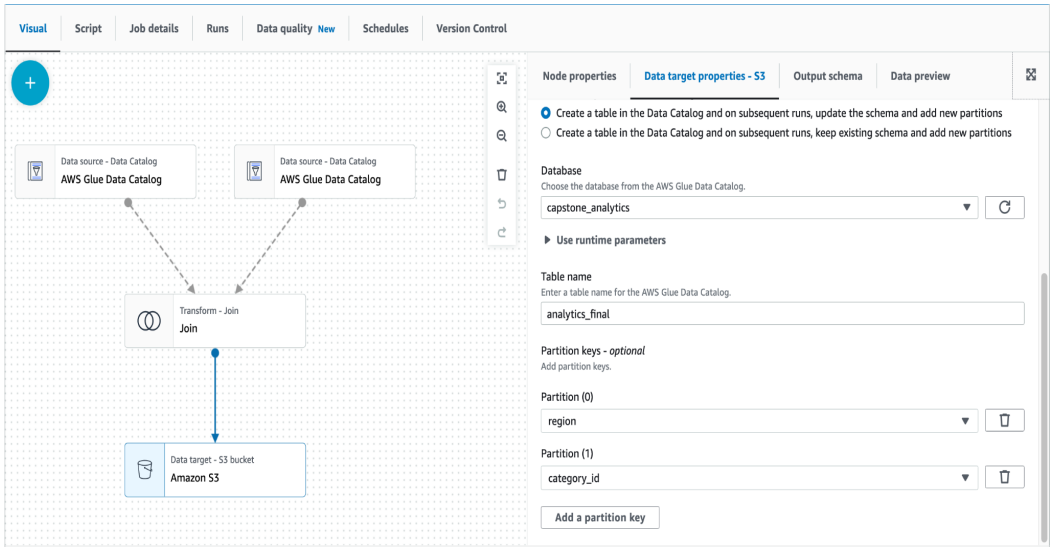
- Created AWS Glue Studio Job named capstone-parquet-analytics
- The Job read data from 2 nodes:
    - CSV data from the database - capstone-database1 (node1)
    - JSON-Parquet data from the database - cleansed-json-data(node2)
- The Job transformed the data from both the nodes with an INNER JOIN operation between category_id field from node1 and id field from node 2.
- The Job finally wrote the integrated final output to a new S3 bucket created and named, capstone-analytics.

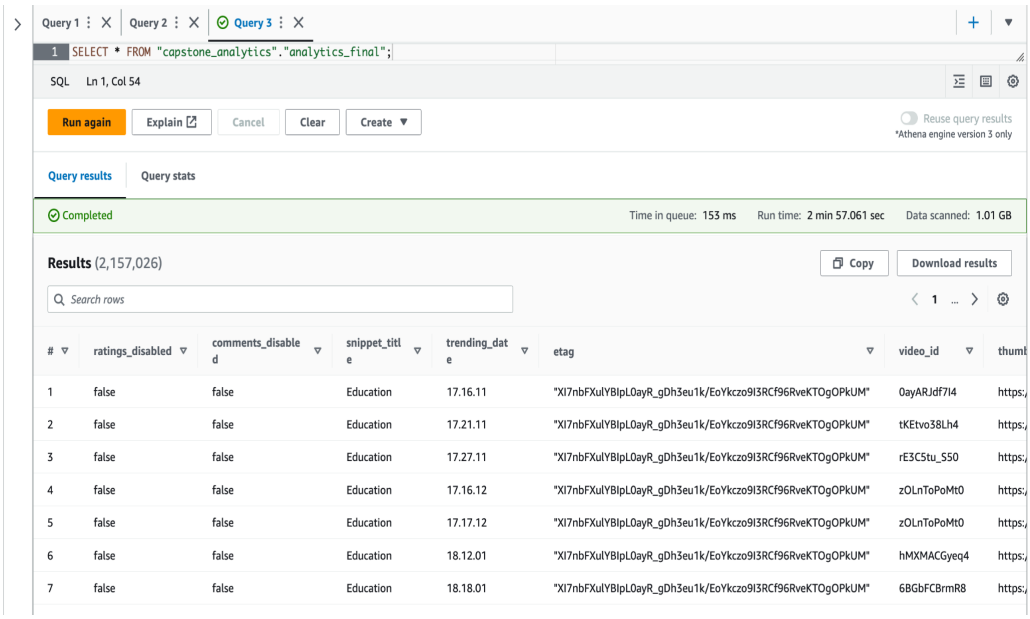





Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Data source - Data Catalog
AWS Glue Data Catalog
Data source - Data Catalog
AWS Glue Data Catalog
Transform - Join
Join
Data target - S3 bucket
Amazon S3
Node properties
Transform
Output schema
Data preview
Join type
Select the type of join to perform.
Inner join
Select all rows from both datasets that meet the join condition.
Join conditions
Select a field from each parent node for the join condition.
AWS Glue Data Catalog
id
=
AWS Glue Data Catalog
category_id
Add condition

Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Data source - Data Catalog
AWS Glue Data Catalog
Data source - Data Catalog
AWS Glue Data Catalog
Transform - Join
Join
Data target - S3 bucket
Amazon S3
Node properties
Data target properties - S3
Output schema
Data preview
Format
Parquet
Compression Type
Snappy
S3 Target Location
Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).
s3://capstone-analytics
View
Browse S3
Data Catalog update options Info
Choose how you want to update the Data Catalog table's schema and partitions. These options will only apply if the Data Catalog table is an S3 backed source.
Do not update the Data Catalog
Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions
Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions
Database
Choose the database from the AWS Glue Data Catalog.
capstone_analytics
Use runtime parameters


Visual
Script
Job details
Runs
Data quality New
Schedules
Version Control
Data source - Data Catalog
AWS Glue Data Catalog
Data source - Data Catalog
AWS Glue Data Catalog
Transform - Join
Join
Data target - S3 bucket
Amazon S3
Node properties
Data target properties - S3
Output schema
Data preview
Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions
Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions
Database
Choose the database from the AWS Glue Data Catalog.
capstone_analytics
Use runtime parameters
Table name
Enter a table name for the AWS Glue Data Catalog.
analytics_final
Partition keys - optional
Add partition keys.
Partition (0)
region
Partition (1)
category_id
Add a partition key



#	ratings_disabled	comments_disabled	snippet_title	trending_date	etag	video_id	thumb
1	false	false	Education	17.16.11	"XI7nbFXulYBIpL0ayR_gDh3eu1k/EoYkczo9I3RCf96RveKTOgOPkUM"	0ayARJdf7I4	https:/
2	false	false	Education	17.21.11	"XI7nbFXulYBIpL0ayR_gDh3eu1k/EoYkczo9I3RCf96RveKTOgOPkUM"	tKEtvo38Lh4	https:/
3	false	false	Education	17.27.11	"XI7nbFXulYBIpL0ayR_gDh3eu1k/EoYkczo9I3RCf96RveKTOgOPkUM"	rE3C5tu_550	https:/
4	false	false	Education	17.16.12	"XI7nbFXulYBIpL0ayR_gDh3eu1k/EoYkczo9I3RCf96RveKTOgOPkUM"	zOLnToPoMt0	https:/
5	false	false	Education	17.17.12	"XI7nbFXulYBIpL0ayR_gDh3eu1k/EoYkczo9I3RCf96RveKTOgOPkUM"	zOLnToPoMt0	https:/
6	false	false	Education	18.12.01	"XI7nbFXulYBIpL0ayR_gDh3eu1k/EoYkczo9I3RCf96RveKTOgOPkUM"	hMXMACGyeq4	https:/
7	false	false	Education	18.18.01	"XI7nbFXulYBIpL0ayR_gDh3eu1k/EoYkczo9I3RCf96RveKTOgOPkUM"	6BGbFCBrmR8	https:/

Step 7: Data Visualization with Tableau

After finally achieving a single integrated dataset from the data pipeline built above, it was time to analyze the video analytics metrics and identify key trends and insights through data visualization.

Data visualization is an essential aspect of any project that involves working with data. It allows us to transform complex information into meaningful and visually engaging representations, enabling easier comprehension, analysis, and decision-making. By leveraging effective data visualization techniques, we can unlock the full potential of our data and communicate insights in a clear and impactful manner.

There are a wide range of data visualization tools available, catering to different skill levels, data types, and project requirements like Tableau, Power BI, Google data Studio, QlikView, AWS Quick sight etc.,

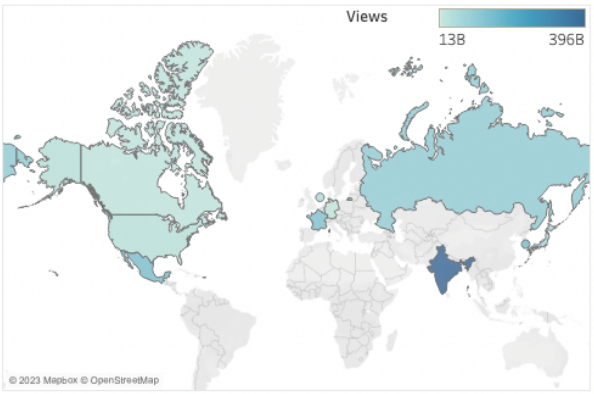
For this project we chose Tableau as our tool to carry out the visualizations for its:

- Intuitive and User-Friendly Interface
- Wide Range of Visualization Options
- Strong Data Connectivity and Integration
- Scalability and Performance
- Active Community and Resources etc.,

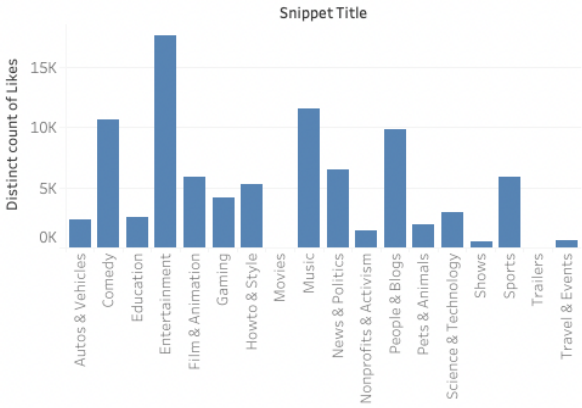
On loading the integrated dataset into tableau, we were able to capture the below trends:

Based on these trends, determining the most effective advertising category for a business use case is possible. For instance, if a company wants to know the optimal category to publish their advertising, the Entertainment category should be the primary choice followed by the Music category. This information can guide them in selecting their target industry accordingly.

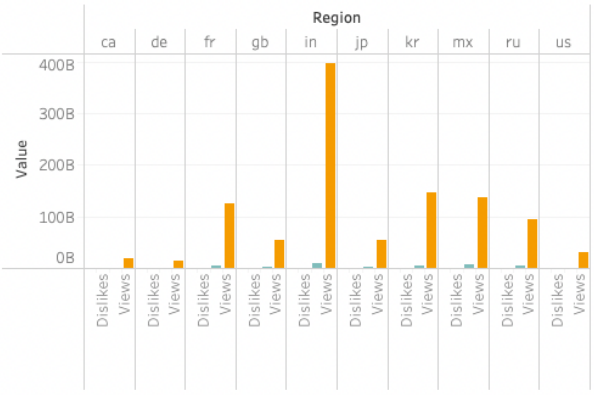
Geospatial Analysis: Video Distribution Across Locations



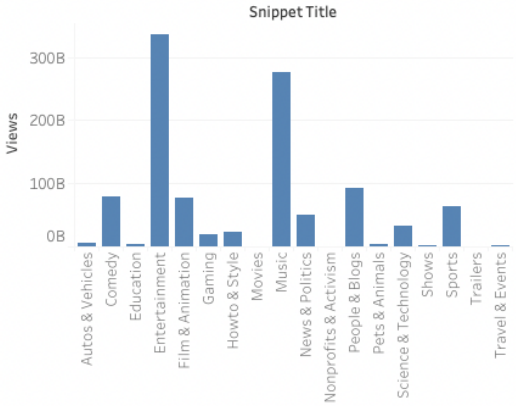
Category-wise Likes Count



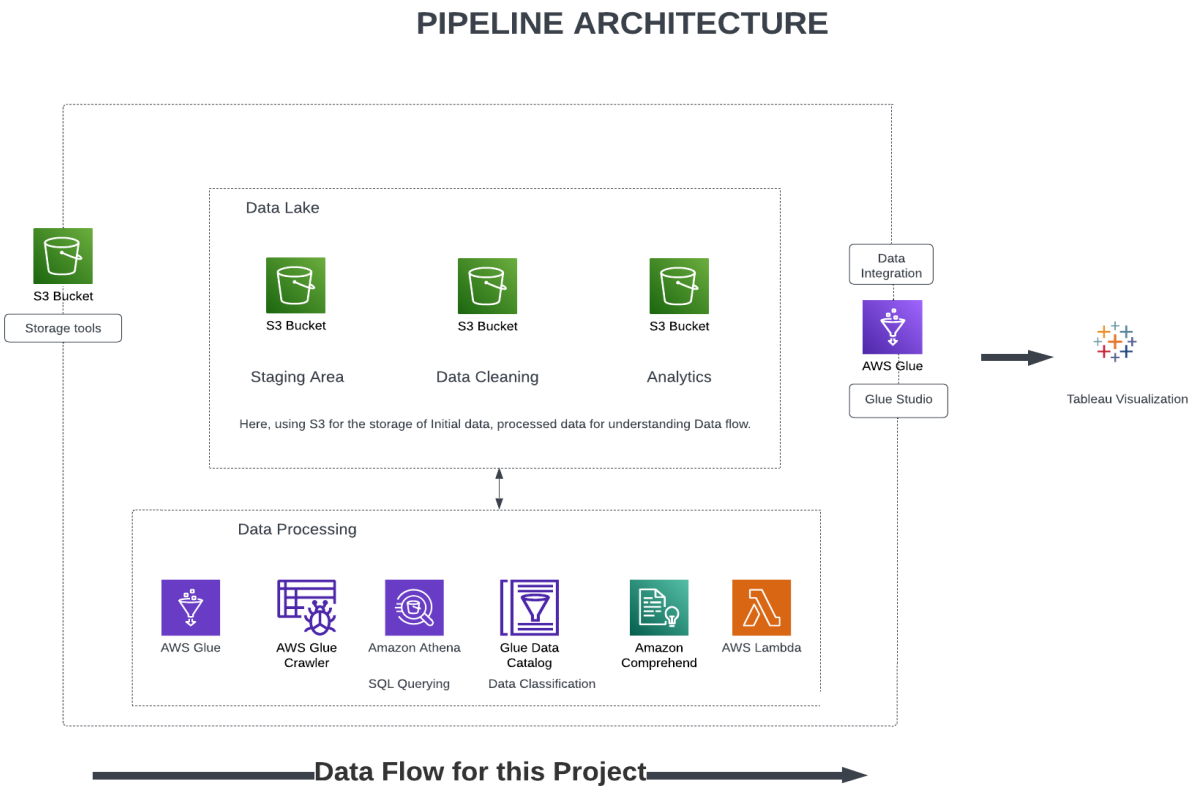
Analyzing Demographic Trends Across Different Countries in the Dataset



Category-wise Views Count



RESULTS


PIPELINE ARCHITECTURE
S3 Bucket
Storage tools
Data Lake
S3 Bucket
S3 Bucket
S3 Bucket
Staging Area
Data Cleaning
Analytics
Here, using S3 for the storage of Initial data, processed data for understanding Data flow.
Data Integration
AWS Glue
Glue Studio
Tableau Visualization
Data Processing
AWS Glue
AWS Glue Crawler
Amazon Athena
SQL Querying
Glue Data Catalog
Data Classification
Amazon Comprehend
AWS Lambda
Data Flow for this Project

FURTHER IMPROVEMENTS

While successfully being able to build a pipeline, the project was also a great learning experience on researching how to understand a business problem, understanding big data infrastructure, data flow and identifying the appropriate tech stack for the problem. It was also a great opportunity to learn about the visualization tools, data pipelining tools available on AWS, writing ETL jobs, scheduling them and running them.

However, we do identify some areas of improvements in the pipeline, which we would like to explore further in the future.

Comprehend at the end of the Data Pipeline

With the help of AWS comprehend and Glue ETL, we were able to perform sentiment analysis on video tags prior to performing data integration. We can possibly ingest and try sentiment analysis with the analytics data at the end of pipeline but only to the regions such as US, CA, GB and DE and not for the regions KR, JP, RU, IN, MX and FR as they consist of a combination of English and local language characters. With the analytics data, we can explicitly perform sentiment analysis for the above-mentioned four regions and query them with the help of region and category filters instead of querying them separately for each region.

Similar Datasets

By utilizing the established procedures or processes for data analysis within AWS, we can also explore analogous datasets such as those from Twitter, IMDB, Elections, Sports, etc. These datasets have already been collected and can be used for conducting sentiment analysis. We may gain insights from the analysis after performing sentiment analysis on these datasets. We can anticipate the future or offer recommendations based on these observations. For instance, if we do sentiment analysis on Twitter data pertaining to a specific brand, we may comprehend how people feel about that brand and provide suggestions on how to enhance its reputation. Similar to this, by analyzing election data, we can predict the outcome of future elections based on public opinion. Businesses, governments, and people trying to make data-driven choices can benefit greatly from the insights gained from sentiment analysis.

Performing sentimental analysis by scraping recent comments on existing dataset

There are multiple ways to do an analysis on daily increasing data. On more way to perform accurate sentiment analysis is to scrap the top/recent 50-100 comments on the videos that

are present in our dataset. With the help of python web scrapping libraries, we can fetch the most recent data to perform sentiment analysis and find out how the viewers perspectives are on different video categories and encourage the advertisers to make ad suggestions.

REFERENCES

- https://www.kaggle.com/datasets/datasnaek/youtube-new
- https://aws.amazon.com/blogs/big-data/harmonize-query-and-visualize-data-from-various-providers-using-aws-glue-amazon-athena-and-amazon-quicksight/
- https://aws.amazon.com/blogs/big-data/build-a-lake-house-architecture-on-aws/
- https://jheck.gitbook.io/hadoop/data_ingestion
- https://www.google.com/books/edition/Deep_Learning_Based_Approaches_for_Senti/tSTMDwAAQBAJ?hl=en&gbpv=0
- https://www.youtube.com/watch?v=KA0QHWm0nWo&t=4s&ab_channel=Intellipaat
- https://www.youtube.com/watch?v=7JUyTqglmNU&ab_channel=AmazonWebServices
- https://www.youtube.com/watch?v=-_DJPRrFQXI&list=PLEiEAq2VkUUKkxDwsS24zFU9kYwma3kdV&ab_channel=Simplilear
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html
- https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html
- https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html
- https://explore.skillbuilder.aws/learn/course/external/view/elearning/8171/getting-started-with-aws-glue
- https://towardsdatascience.com/big-data-file-formats-explained-dfaabe9e8b33
- https://docs.aws.amazon.com/lambda/latest/dg/lambda-foundation.html
- https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html
- https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-awscli.html
- https://aws-sdk-pandas.readthedocs.io/en/stable/install.html
- https://www.unfetteredmarketing.com/post/what-is-a-utf-8-csv-and-why-should-i-care
- https://www.simform.com/blog/how-to-build-a-real-time-social-media-sentiment-analysis-data-pipeline/
- https://medium.com/plumbersofdatascience/building-etl-pipeline-and-performing-sentiment-analysis-on-ipl-2022-twitter-data-f488e038d66a
- https://docs.aws.amazon.com/comprehend/latest/dg/tutorial-reviews-analysis.html
- https://aws.amazon.com/blogs/machine-learning/how-to-scale-sentiment-analysis-using-amazon-comprehend-aws-glue-and-amazon-athena/
- https://www.simplilearn.com/tutorials/tableau-tutorial/what-is-tableau

SUMMARY OF TOOLS MENTIONED

- AWS S3 is a highly scalable and secure object storage service that can store and retrieve any amount of data from anywhere.
- AWS Athena is a serverless, interactive query service that allows you to analyze data stored in S3 using standard SQL.
- AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy to move data among data stores.
- AWS Glue Studio is a visual development environment for creating, testing, and deploying Glue ETL jobs.
- AWS Glue Crawler is a tool that automatically discovers and classifies the data stored in S3, creating a Glue Data Catalog for use with Athena and Glue.
- AWS Lambda is used for serverless execution of code in response to events, without the need to provision or manage servers.
- AWS Comprehend is used for natural language processing (NLP), allowing businesses to gain insights from unstructured text data by extracting information such as sentiment, entities, and key phrases.
- Tableau is a business intelligence and data visualization tool that allows users to connect to various data sources, create interactive dashboards and reports, and share insights with others. It is specially designed for data analysis, visualization and report creation.
- Spark, PySpark - Apache Spark is a fast and general-purpose cluster computing system. PySpark is the Python library for Spark programming. Together, they provide a powerful tool for processing and analyzing large data sets in a distributed computing environment

APPENDIX: DATA DICTIONARY

- Video_id: Identification of video.
- Trending_date: Date on which video was trending.
- Title: Title of the video.
- Channel_title: Channel name on which video was uploaded.
- Category_id: Identification for the category of video. This ranges from 1 - 43
- Publish_time: Time video was published.
- Tags: Tags that were mentioned on the video.
- Views: Number of views obtained by the video.
- Likes: Number of likes obtained by the video.
- Dislikes: Number of dislikes obtained by the video.
- Comment_count: Number of comments on the video.
- Thumbnail_link: Contains the link to the thumbnail on the video.

APPENDIX: ARTICLES

Article : Effects of having and not having Staging area in the Data Pipeline
Article : Pros and Cons of Implementing a Staging Area in Your Workflow
Article : Unleashing the Power of AWS Comprehend: Understanding Text Analytics at Scale