YOUTUBE DATA ANALYSIS

Abstract

This report outlines a comprehensive data engineering project designed to analyze YouTube channel performance metrics within the data science and analytics domain. Utilizing Google Cloud Platform's robust cloud computing capabilities, this project harnesses the power of YouTube's API to ingest real-time data on various prominent data science channels. The data pipeline automates the collection, storage, and processing of data points such as subscriber counts, view statistics, video engagement, and content trends through a series of GCP services including Cloud Functions, Pub/Sub, Cloud Storage, DataProc with PySpark, and BigQuery. The processed data is then visualized using Looker Studio to uncover key patterns and strategies that enhance content creation practices. This report details the project's architecture, methodology, and the insights derived from the data analysis, aiming to provide a scalable solution for continuous YouTube channel monitoring and analytics.

Table of contents:

Chapter 1

Introduction

I. Application Domain:

The realm of YouTube content creation has evolved into a dynamic platform where data-driven insights significantly influence success. This project is situated within the application domain of data science and analytics, a rapidly growing field on YouTube. By analyzing data from popular channels such as Corey Schafer, Sentdex, and others, the project aims to uncover the nuances that contribute to channel growth and viewer engagement. Through systematic data collection and analysis, we seek to understand how content strategies affect various performance metrics like subscriber and view counts. This domain-specific study serves as a cornerstone for creators aiming to optimize their content to meet the evolving preferences and trends of their audience.

II. Detailed Problem Examination on Application Level:

In the digital age, YouTube channels dedicated to data science and analytics face intense competition to capture and maintain viewer interest. The challenge lies in understanding what metrics most significantly impact a channel's growth and how these can be optimized to enhance audience engagement. Channels must navigate the complexities of content frequency, diversity, and relevance to retain a growing subscriber base. Additionally, there is a need to analyze viewer interaction through likes, comments, and shares to gauge content effectiveness. These issues complicate content strategy, necessitating a thorough examination of performance data to identify trends and opportunities for improvement in a crowded market.

III. Comprehensive Benefits of an Application-Level Solution

Implementing a data-driven solution at the application level offers significant benefits to YouTube content managers and strategists in the field of data science. Firstly, it provides a systematic approach to quantifying engagement metrics such as view counts, like counts, and comment counts. By analyzing these metrics, content managers can identify standout content features that consistently engage viewers, helping to tailor future videos to audience preferences. Secondly, this solution empowers strategists to discern patterns in viewer behavior related to video release timing. Understanding how different publishing schedules affect video performance allows for the optimization of release strategies to maximize viewer engagement and interaction. Additionally, such an application-level solution facilitates the aggregation and comparison of data across multiple channels, enabling a broader understanding of competitive positioning and content strategy efficacy in the data science community. This, in turn, supports more informed decision-making and strategic planning aimed at enhancing the channel’s reach and impact in the digital content landscape.

IV. Technical Challenges and Analysis:

Building a data engineering pipeline to process the provided dataset presents significant technical challenges, particularly related to data quality and handling multilingual content. The primary task involves comprehensive data cleansing to resolve encoding issues and standardize the dataset for subsequent analysis.

The pipeline is constructed on a Spark cluster, utilizing PySpark to leverage Python's powerful data processing libraries alongside Spark's distributed computing capabilities. This combination effectively manages the complexities and scale of the dataset without the need for natural language processing techniques.

Integrating with Google Cloud Platform (GCP) enhances the pipeline's scalability and flexibility. GCP's storage and compute services offer a robust infrastructure for large-scale data handling. Additionally, incorporating Google BigQuery into the pipeline provides powerful data warehousing capabilities, facilitating efficient storage, retrieval, and analysis of large datasets. BigQuery's SQL-like interface and rapid analytics engine enable sophisticated data queries and aggregations, which are essential for deriving insights from multilingual feedback data.

V. Proposed Technical Solution in Depth:

1. Data Source: The architecture begins with YouTube as an external data source. YouTube datasets are accessed via an API, which suggests that this solution intends to programmatically pull data from YouTube for processing.

2. Cloud Function: The Cloud Function executes a script that makes an API call to the YouTube API, retrieving data such as channel statistics, video details, and engagement metrics.

3. Cloud Pub/Sub: A Cloud Pub/Sub topic is set to publish a message every 5 minutes. This message triggers the Cloud Function to start the data ingestion process.

3. Cloud Storage Bucket: The data is then stored in a Google Cloud Storage Bucket, which is a scalable and secure object storage service. This would be used for storing large amounts of unstructured data that the pipeline processes.

4. Data Processing:

- Dataproc Cluster: The data is processed using a Dataproc Cluster, which is GCP's managed Hadoop and Spark service. This would be responsible for running large-scale data processing jobs, likely transforming the raw data into a more usable format or computing aggregations.

- Spark Jobs: Specifically, Spark jobs within the Dataproc cluster are used for fast, in-memory data processing. This could include ETL (Extract, Transform, Load) tasks, batch processing, and machine learning model training.

5. BigQuery Tables: After processing, the data is stored in BigQuery tables. BigQuery is a fully managed, serverless data warehouse that enables scalable analysis over petabytes of data. It is highly optimized for SQL queries.

6. Visualization:

- Looker Studio: For visualization and business intelligence, the processed data is pulled into Looker Studio (formerly known as Google Data Studio). Looker Studio connects to BigQuery to create interactive reports and dashboards.

Chapter 2

Related work

This literature review provides a contextual backdrop for the analytical exploration of YouTube data analytics. By synthesizing and analyzing existing research, we aim to uncover trends, methodologies, and key findings that have shaped our understanding of video analytics and audience engagement on digital platforms.

1. **Title**: "Mining YouTube Metadata for Predicting Video Popularity" [1]
    - Investigates the impact of metadata elements (titles, descriptions, tags) on the popularity and viewer engagement of YouTube videos.
    - Utilizes various machine learning models to predict outcomes based on metadata, demonstrating the potential for predictive analytics in content strategy.
    - Suggests that well-crafted metadata significantly enhances discoverability and engagement, emphasizing the strategic aspect of content management.
    - Provides case studies showing different metadata strategies and their effectiveness in different content on YouTube.

2. **Title**: "Big Data and Cloud Computing: Current State and Future Opportunities"[2]
    - Reviews the evolution of big data technologies and their convergence with cloud computing platforms, specifically within the media sector.
    - Discusses how cloud solutions like Google Cloud Platform and Apache Spark offer scalability and flexibility, essential for handling large datasets typical in media analytics.
    - Explores the economic and technological advantages of cloud computing in data processing, particularly the reduction of infrastructure costs and the improvement of data accessibility.
    - Highlights the challenges of data security and privacy in cloud environments, an important consideration for projects dealing with sensitive or personal data.
    - Proposes future research directions and technologies that could further enhance big data analytics in media, such as improved machine learning algorithms and more robust data integration techniques.

By examining these studies, the review not only emphasizes the relevance of advanced computational techniques in analyzing media content but also illustrates the practical challenges and solutions encountered in the field. These insights directly inform the methodology and expectations of our project, ensuring that our approach is grounded in proven scientific principles and industry practices.

Chapter 3

Data Set

The YouTube dataset consists of various data points collected from a selection of popular data science and analytics-focused YouTube channels. The data is obtained through the YouTube API and encompasses a wide range of metrics related to video performance and channel engagement. This dataset is crucial for analyzing trends, understanding audience engagement, and optimizing content strategy.

```
RangeIndex: 2019 entries, 0 to 2018
Data columns (total 19 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   video_id        2019 non-null   object
 1   channelTitle    2019 non-null   object
 2   title           2019 non-null   object
 3   description     1975 non-null   object
 4   tags            1731 non-null   object
 5   publishedAt     2019 non-null   object
 6   viewCount       2019 non-null   int64
 7   likeCount       2018 non-null   float64
 8   favouriteCount  0 non-null      float64
 9   commentCount    2018 non-null   float64
 10  duration        2019 non-null   object
 11  definition      2019 non-null   object
 12  caption         2019 non-null   bool
 13  publishDayName  2019 non-null   object
 14  durationSecs    2019 non-null   object
 15  tagsCount       2019 non-null   int64
 16  likeRatio       2014 non-null   float64
 17  commentRatio    2013 non-null   float64
 18  titleLength     2019 non-null   int64
dtypes: bool(1), float64(5), int64(3), object(10)
```

The data for this project is sourced from several prominent YouTube channels focused on data science and analytics.

These channels include:

Corey Schafer, Sentdex, Krish Naik, DatascienceDoJo, Luke Barousse, Ken Jee, Alex the Analyst, Tina Huang, Data School, Codebasics

Channel Data Overview:

These channels were selected based on their significant influence and large follower base within the data science community. The data extracted from these channels includes various metrics such as:

- **Channel Statistics**: Details such as channel name, subscriber count, view count, total number of videos, and the uploads playlist ID.
- **Video IDs from Uploads Playlist**: Video IDs retrieved using the playlist ID to gather video-specific data.
- **Video Details**: Information for each video including title, description, tags, published date, view count, like count, comment count, and duration.
- **Video Comments**: Top-level comments for each video to analyze user engagement and feedback.

API Update Uncertainty:

One of the challenges faced in this project is the uncertainty surrounding the exact timing of updates to the YouTube API data. YouTube's API does not provide explicit details on when data points such as view counts, likes, comments, and other metrics are updated. This lack of precise update timing means that:

- **Data Latency**: There can be a delay between when an event occurs (e.g., a video gets more views or likes) and when this data is reflected in the API responses.
- **Inconsistent Data Freshness**: The freshness of the data cannot be guaranteed, making it essential to account for potential discrepancies and lag in real-time analysis.
- **Scheduling Challenges**: Without knowing the exact update schedule, triggering data fetches at optimal intervals becomes challenging. This uncertainty necessitates a balance between too frequent data pulls (which may waste resources) and too infrequent pulls (which may miss timely updates).

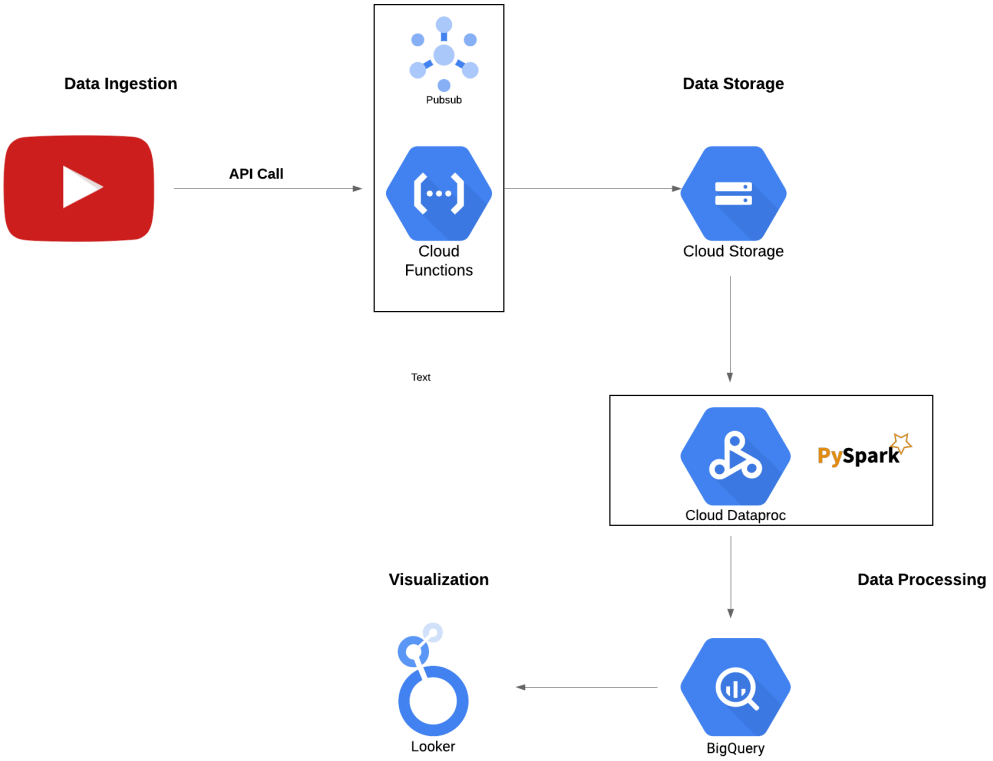
Chapter 4

Solution

Project Architecture:

In the contemporary data-centric landscape, the cornerstone of deriving actionable insights from raw data lies in the efficient execution of Extract, Transform, and Load (ETL) processes.

This section delineates the architecture underpinning our ETL system deployed on the Google Cloud Platform (GCP).



- **Why Data Ingestion using YouTube API [1]:** The YouTube API is a powerful tool for programmatically retrieving data from YouTube. It provides access to a wide range of data, including channel statistics, video details, and user engagement metrics. By leveraging the API, we can automate the collection of real-time data from various YouTube channels.
    - The API allows us to fetch comprehensive data about channels, videos, and user interactions.
    - It supports various endpoints for retrieving different types of data, making it highly versatile.

    - Using the API ensures that we get the most up-to-date information directly from YouTube's servers.
    - It simplifies the process of data extraction, reducing manual efforts and potential errors.
    - The API provides structured data, which is easy to process and analyze.

- **Why Use Google Cloud Functions [2]:** Cloud Functions provide a serverless environment to execute code in response to events, such as HTTP requests or messages from Cloud Pub/Sub.
    - Cloud Functions automatically scale, ensuring they handle varying workloads without manual intervention.
    - They allow for the execution of lightweight, single-purpose functions, simplifying the development process.
    - Within the project, Cloud Functions are used to automate the extraction of data from the YouTube API, ensuring timely data retrieval.

- **Why Use Google Cloud Pub/Sub [3]:** Cloud Pub/Sub is a messaging service that allows for asynchronous communication between different components of a cloud application.
    - Enables the decoupling of services, making the system more flexible and scalable.
    - Supports reliable message delivery and processing, ensuring that no data is lost.
    - In the project, Pub/Sub triggers Cloud Functions every 5 minutes to fetch and process new YouTube data.

- **Why Data Storage in Google Cloud Storage [4]:** Cloud Storage is a service for storing objects in Google Cloud. An object is an immutable piece of data consisting of a file of any format. You store objects in containers called buckets.
    - Buckets offer durability, availability, and scalability, handling vast amounts of data with ease.
    - Managed folders within buckets allow for expanded access to groups of objects with shared name prefixes.
    - Buckets have globally unique names, ensuring easy access from anywhere in the world.
    - Data uploaded to Cloud Storage is highly secure, with multiple layers of redundancy.
    - Within the project, as discussed in the ingestion layer, the data is directly uploaded to these buckets.

- **Why Data Processing in DataProc Cluster [5]:** Dataproc is a managed Spark and Hadoop service that lets you take advantage of open-source data tools for batch processing, querying, streaming, and machine learning.
    - Dataproc automation helps you create clusters quickly, manage them easily, and save money by turning clusters off when not needed.

- Easily integrates with other GCP services like Cloud Storage, BigQuery, Cloud Logging, and Monitoring.
- Fast to create and cost-effective for processing large datasets.
- The master node in a Dataproc cluster runs essential services like Resource Manager, JobHistoryServer, and HDFS Name Node.
- The PySpark script reads files stored in Cloud Storage, performs necessary transformations, and processes data efficiently.

- **Why Processed Data in BigQuery [6]:** BigQuery is a fully managed enterprise data warehouse that helps you manage and analyze your data.
  - BigQuery's serverless architecture lets you use SQL queries to answer your organization's biggest questions with zero infrastructure management.
  - Scalable, distributed analysis engine lets you query terabytes in seconds and petabytes in minutes.
  - Automatically encrypts all data before it is written to disk, ensuring security.
  - Stores data in tables with defined schemas and data types, making it easy to manage structured data.
  - In this project, processed data from PySpark jobs are stored in BigQuery tables, enabling sophisticated data analysis.

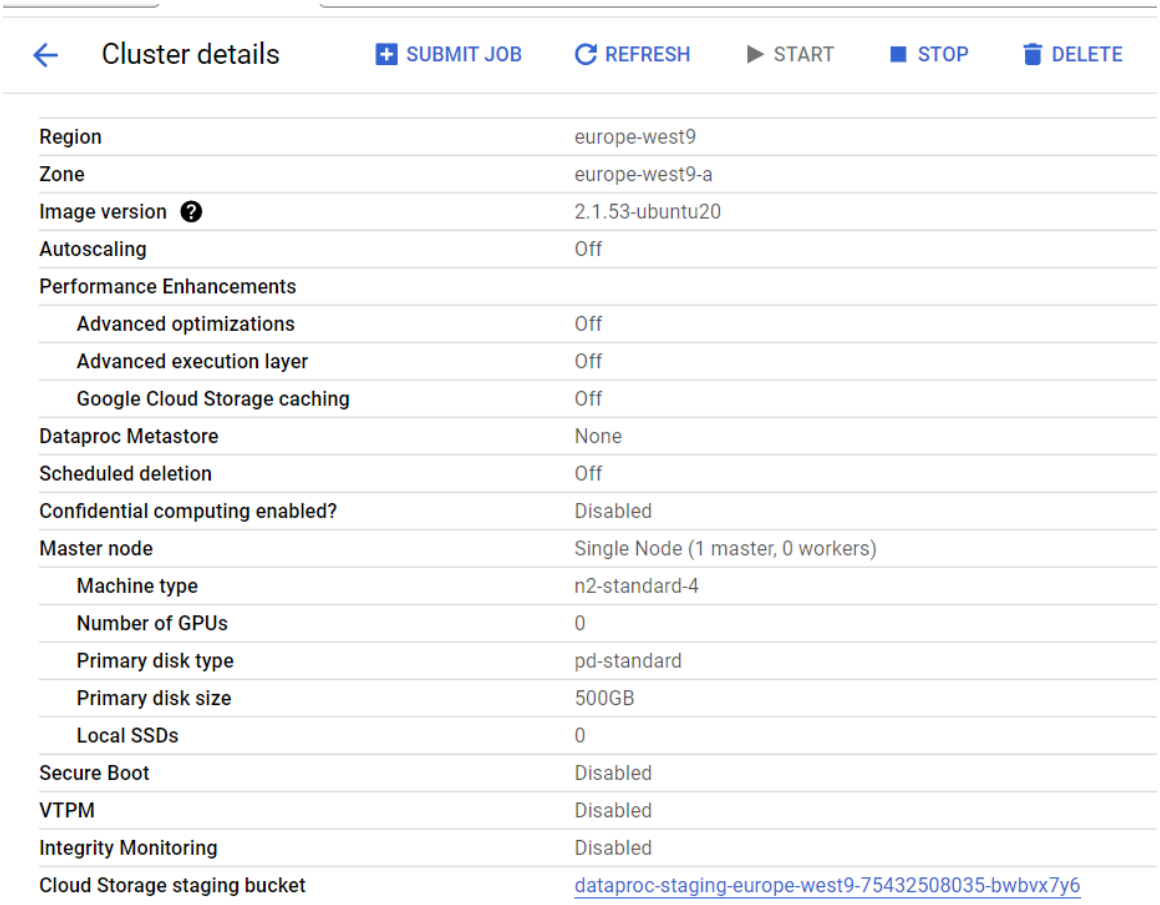
- **Why Visualization using Looker Studio [7]:** Looker Studio is a no-cost self-service business intelligence platform that lets you build and consume data visualizations, dashboards, and reports.
  - Allows connection to BigQuery data for creating detailed visualizations.
  - Enables the creation of interactive dashboards that can be shared with stakeholders.
  - Supports various types of visualizations to represent data insights effectively.
  - User-friendly interface makes it easy to build and customize reports.
  - Enhances data storytelling, helping to communicate insights derived from the YouTube data analysis effectively.

<u>Dataproc Cluster Setup:</u>

Dataproc is a managed Spark and Hadoop service that lets you take advantage of open-source data tools for batch processing, querying, streaming, and machine learning.

1. Cluster setup steps and configuration:
2. We created a cluster instance with a compute engine.
3. We specified a location near Germany "Europe-west9".
4. Cluster type: Single Node (1 Master, 0 Workers).
5. Machine type: n2-standard-4 (4 vCPUs, 16 GB memory), disk size - 500GB.
6. Versioning of the services: Ubuntu 20.04, Hadoop 2.10, Spark 2.4.

7. No autoscaling.
8. We opted for external components like Anaconda and Jupyter Notebook.



Limitations:

1. **Single Node Configuration**: The cluster consists of only one master node without any worker nodes. This configuration severely limits parallel processing capabilities and may not efficiently handle large-scale data processing tasks.
2. **No Autoscaling**: Autoscaling is disabled, which means the cluster cannot automatically adjust resources based on workload demands. This could lead to performance issues during peak processing times or underutilization during low-demand periods.
3. **Fixed Resources**: The machine type is fixed at n2-standard-4 with 4 vCPUs and 16 GB of memory. While adequate for small to moderate workloads, it may not be sufficient for more extensive data processing requirements or high concurrency tasks.
4. **Limited Durability and Redundancy**: As the setup includes only one node, there is no redundancy. In case of a node failure, the entire processing would halt, risking data loss or requiring significant recovery time.

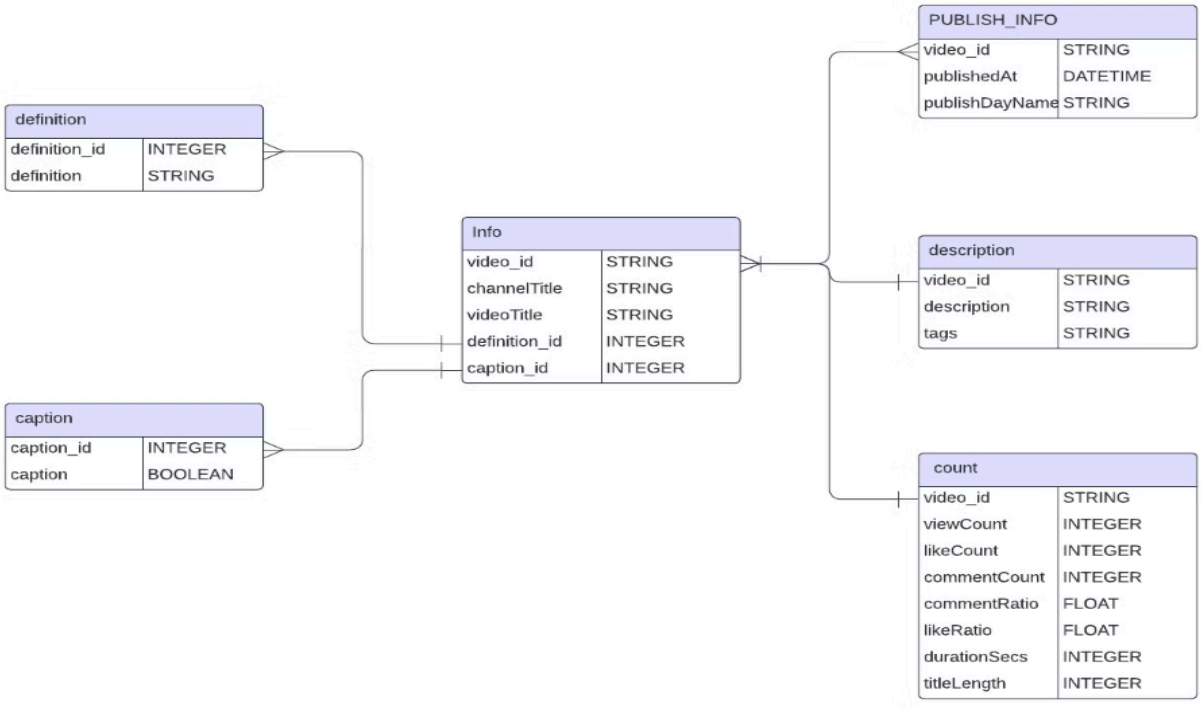
5. **Manual Management**: Without autoscaling and advanced optimizations, the cluster requires more manual management and monitoring to ensure performance and reliability, increasing operational overhead.
6. **Lack of Specialized Hardware**: The setup does not include GPUs or local SSDs, which could accelerate certain types of computations, such as machine learning tasks or high I/O operations.

PySpark Process Overview:

1. Established a connection to the Google Cloud Storage (GCS) bucket to retrieve the stored CSV file.
2. Initialized the Spark session and defined a schema to read the CSV file into a Spark DataFrame.
3. Imputed null values with appropriate placeholders, such as 0 and N/A.
4. Removed unnecessary columns from the DataFrame.
5. Converted the publishedAt column to Datetime format.
6. Normalized the DataFrame and split it into six separate DataFrames for different data segments.
7. Created schemas for each of the six tables and loaded the data into BigQuery.

This streamlined approach ensures efficient data processing and accurate data loading into BigQuery for further analysis.

Database Schema:



Chapter 5

Summary and Outlook

Pyspark (Use-cases and Visualization)

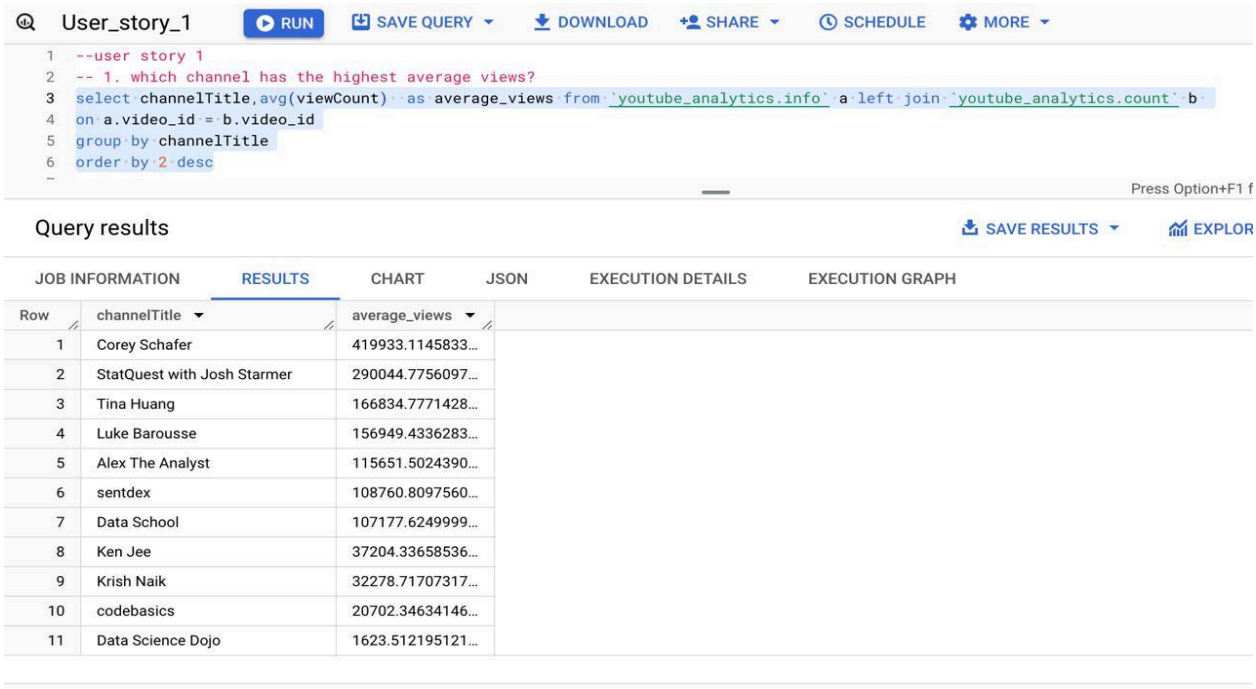
User Story 1:

Understanding Engagement Metrics Across Data Science Channels:

As a Data Science Content Manager, I want to analyze the engagement metrics (viewCount, likeCount, commentCount) of data science-related YouTube channels, so that I can identify the most engaging content and understand what type of videos resonate most with our audience.

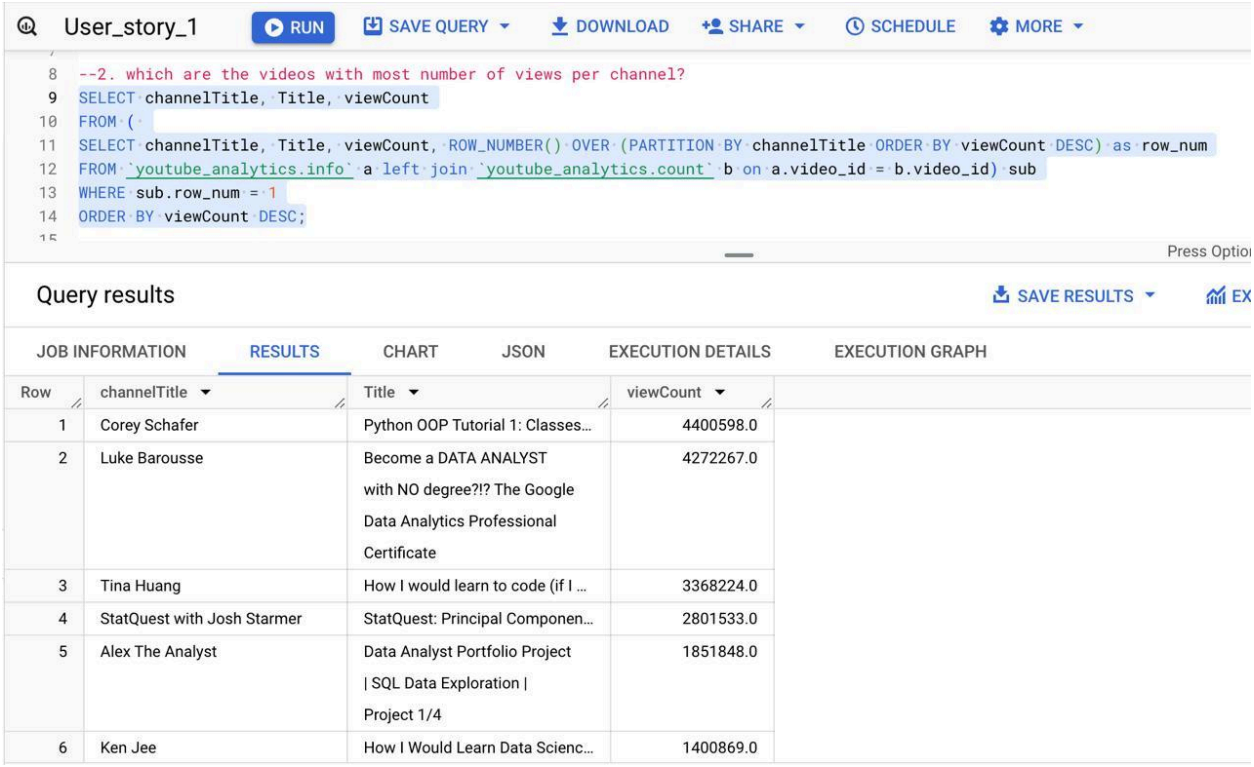
Business Query:

1. Which YouTube channel has the highest average views? (Total Views/Number of Videos)



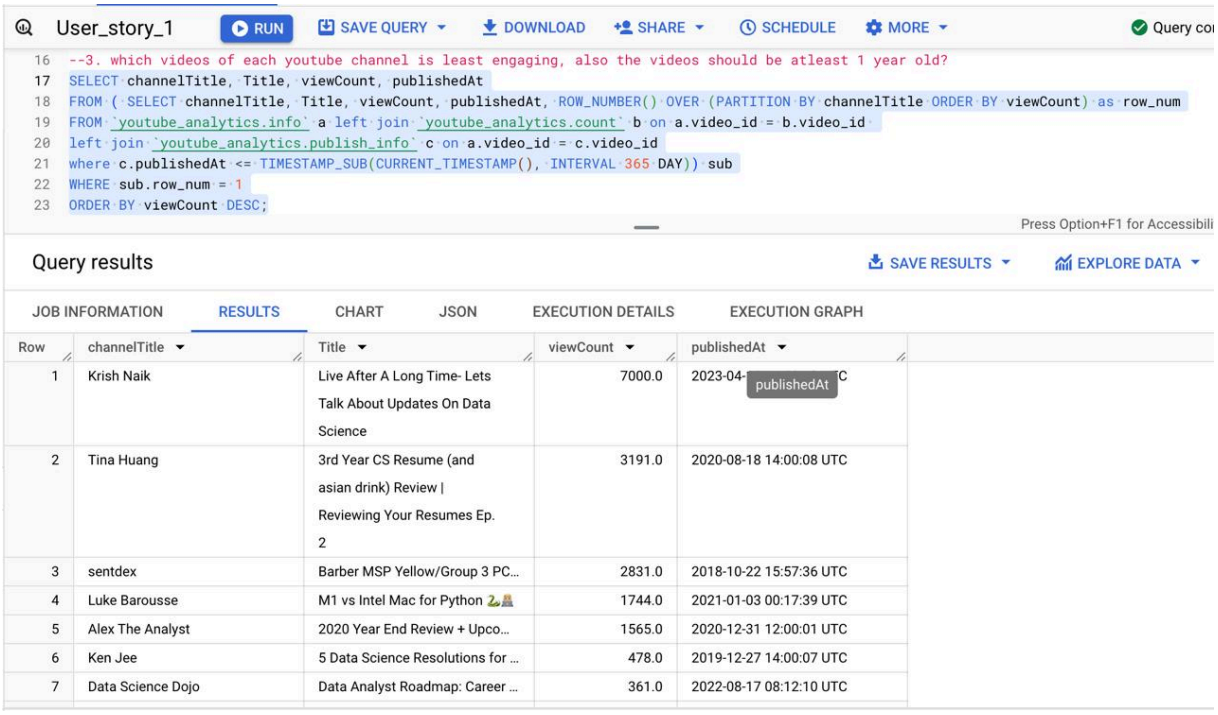
Row	channelTitle	average_views
1	Corey Schafer	419933.1145833...
2	StatQuest with Josh Starmer	290044.7756097...
3	Tina Huang	166834.7771428...
4	Luke Barousse	156949.4336283...
5	Alex The Analyst	115651.5024390...
6	sentdex	108760.8097560...
7	Data School	107177.6249999...
8	Ken Jee	37204.33658536...
9	Krish Naik	32278.71707317...
10	codebasics	20702.34634146...
11	Data Science Dojo	1623.512195121...

2. What kind of videos have the most number of views per channel?



Row	channelTitle	Title	viewCount
1	Corey Schafer	Python OOP Tutorial 1: Classes...	4400598.0
2	Luke Barousse	Become a DATA ANALYST with NO degree?!? The Google Data Analytics Professional Certificate	4272267.0
3	Tina Huang	How I would learn to code (if I ...	3368224.0
4	StatQuest with Josh Starmer	StatQuest: Principal Componen...	2801533.0
5	Alex The Analyst	Data Analyst Portfolio Project | SQL Data Exploration | Project 1/4	1851848.0
6	Ken Jee	How I Would Learn Data Scienc...	1400869.0

3. Which videos of each youtube channel is the least engaging with the videos being atleast 1 year old?



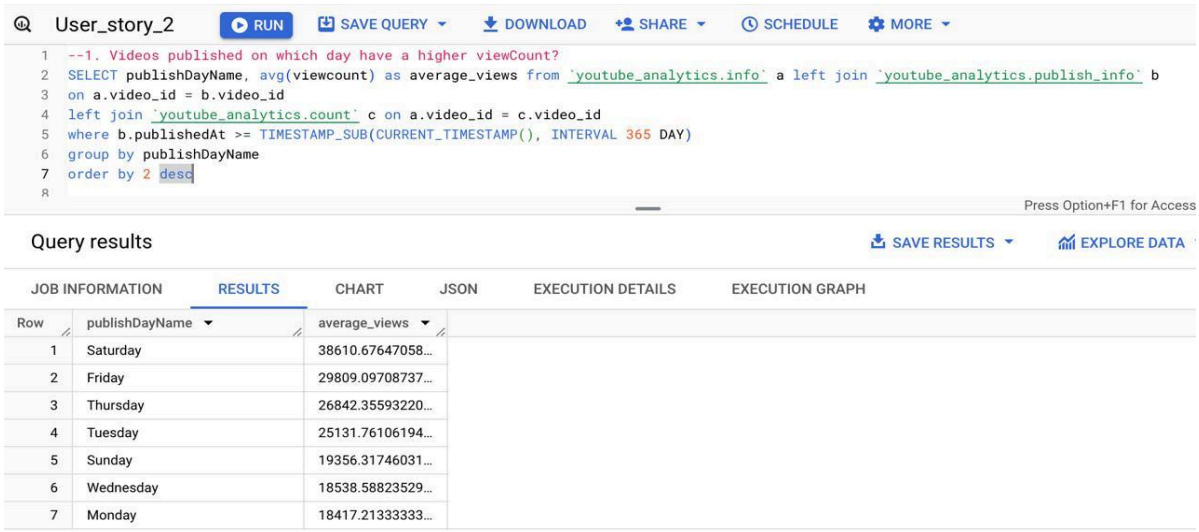
Row	channelTitle	Title	viewCount	publishedAt
1	Krish Naik	Live After A Long Time- Lets Talk About Updates On Data Science	7000.0	2023-04-
2	Tina Huang	3rd Year CS Resume (and asian drink) Review | Reviewing Your Resumes Ep. 2	3191.0	2020-08-18 14:00:08 UTC
3	sentdex	Barber MSP Yellow/Group 3 PC...	2831.0	2018-10-22 15:57:36 UTC
4	Luke Barousse	M1 vs Intel Mac for Python	1744.0	2021-01-03 00:17:39 UTC
5	Alex The Analyst	2020 Year End Review + Upco...	1565.0	2020-12-31 12:00:01 UTC
6	Ken Jee	5 Data Science Resolutions for ...	478.0	2019-12-27 14:00:07 UTC
7	Data Science Dojo	Data Analyst Roadmap: Career ...	361.0	2022-08-17 08:12:10 UTC

User Story 2:

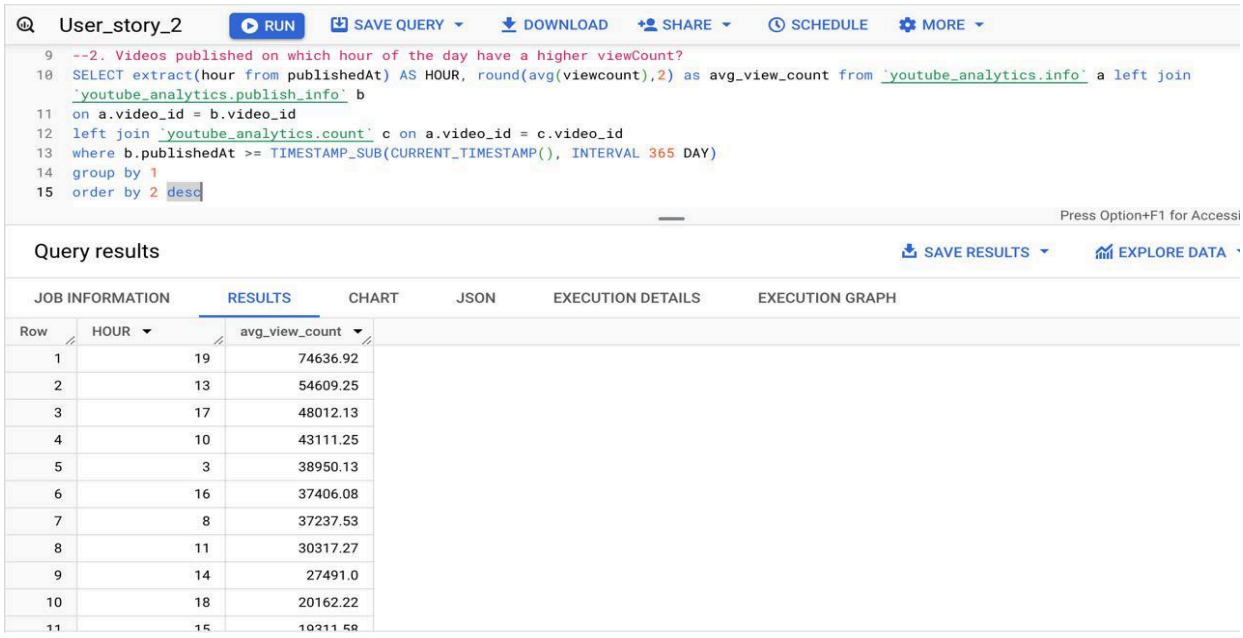
As a Data Science Content Strategist, I want to analyze the impact of publishing day and time on the engagement of videos, so that I can optimize the release schedule to maximize views and interactions.

Business Query:

1. Videos published on which day have a higher viewCount?



2. Videos published on which hour of the day have a higher viewCount?



3. Which are the favorite days for youtubers to publish the videos?

User_story_2 RUN SAVE QUERY DOWNLOAD SHARE SCHEDULE MORE

```sql
--3. which are the favourite days for youtubers to post the videos?
WITH CombinedData AS (SELECT a.channelTitle, b.publishDayName
FROM `youtube_analytics.info` a
JOIN `youtube_analytics.publish_info` b ON a.video_id = b.video_id
JOIN `youtube_analytics.count` c ON a.video_id = c.video_id),
DayFrequency AS (SELECT channelTitle, publishDayName, COUNT(*) AS video_count FROM CombinedData
GROUP BY channelTitle, publishDayName),
MaxDayFrequency AS (SELECT channelTitle, publishDayName, video_count, RANK() OVER (PARTITION BY channelTitle ORDER BY video_count DESC)
as rank FROM DayFrequency)
SELECT channelTitle, publishDayName, video_count
FROM MaxDayFrequency
WHERE rank = 1
ORDER BY channelTitle;
```

Press Option+F1 for Accessibility O

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	channelTitle	publishDayName	video_count
1	Alex The Analyst	Tuesday	141
2	Corey Schafer	Wednesday	49
3	Data School	Tuesday	30
4	Data Science Dojo	Thursday	87
5	Ken Jee	Friday	90
6	Krish Naik	Thursday	32
7	Krish Naik	Friday	32

Visualization: [Looker Dashboard]