Big Data Engineering

YouTube Trending Video Dataset Analysis Using Data Lakehouse Approach with Snowflake

Master of Data Science and Innovation
University of Technology Sydney

Date: 04-09-2022

1. Project Overview

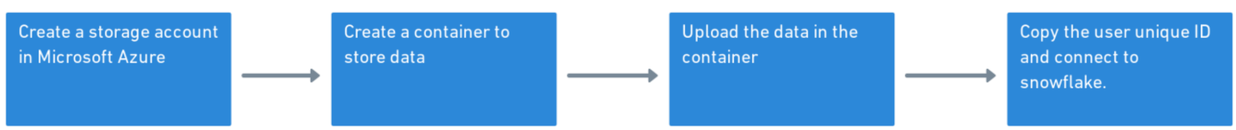
A data lakehouse is a data management architecture that combines the benefits of a traditional data warehouse and a data lake. It is a more cost-effective data storage unit. It implements several improvements in the data architectures, processing and metadata management that end users can efficiently use in machine learning and BI applications.

In this project, the data is the youtube trending dataset. This a continuous data by nature and will be analysed with the help of Microsoft Azure, the cloud storage and Snowflake, An SaaS(software as a service) that provides a platform for a data warehouse, data lake, data lakehouse, and sharing real-time/shared data.

2. Setup

Here, the entire project works on the connection of Microsoft Azure and Snowflake. As a result, several steps must be completed beforehand to start working with the data.
In Microsoft Azure,

1. Creating a storage account.
2. Creating a container.
3. Uploading the data (all the CSV and JSON).
4. Copy the unique id to connect to snowflake.



Flowchart 1: Setup Steps in Microsoft Azure.

In the snowflake,

1. Creating a database.
2. Creating a storage integration with Azure.
3. Providing permission to snowflake to access the data.
4. Creating a stage to store the data.

Flowchart 2: Setup steps in Snowflake.

3. Dataset Exploration

There are two type o data in this analysis process. One is trending, and another is category data. Both datasets have data from 10 countries like the USA, India, Korea etc.

In the youtube trending data, there are 13 columns. The data dictionary is shown below.

Column Name	Description
Video_ID	A unique id of the video
Title	The title of the video.
Publishedat	The time when the video was uploaded
ChannelId	The auto generated id of a channel.
ChannelTitle	The name of the channel where the video was uploaded.
CategoryId	Numeric representation of the category of the video.
Trending_date	The day when the particular video was trending.
View_count	The number of views of the video.
Likes	The number of like impressions pressed on the video.
Dislikes	The numbers of dislike impression pressed on the video.
Comment_count	The number of comments on the video.
Comments_disabled	Boolean Value. If the comment section is disabled for users or not.
Country	Name of the country where the video has been published.

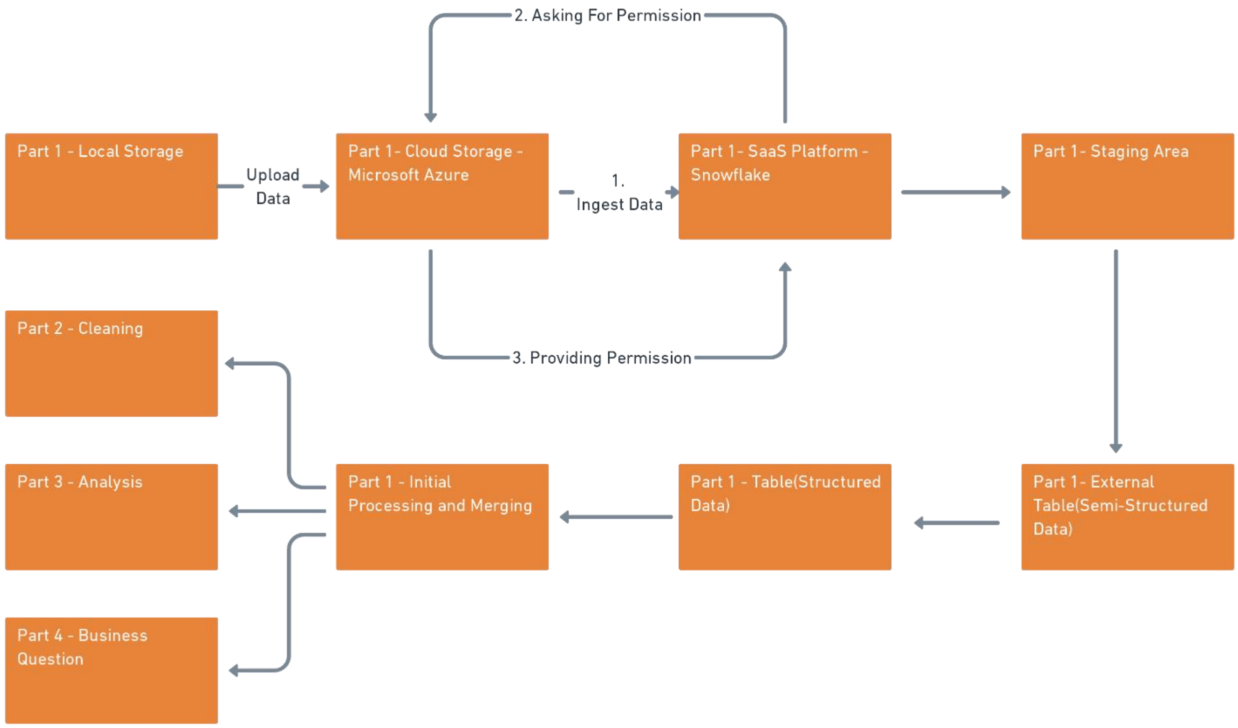
Table 1: Data Dictionary of youtube trending data.

The type of JSON files was nested in nature. It contains information about the categories. Numerous pieces of information were put in the JSON files. However, we are interested in only three data attributes for the analysis.

Column Name	Description
Country	The country where video was uploaded
CategoryId	Numeric representation of the category
Category_title	The title of the category.

Table 2: Data Dictionary of youtube category data.

4. Workflow Diagram of this Analysis



Flowchart 3: Workflow Diagram

The entire workflow of this process consists of several parts. At first, the data must be downloaded and unzipped to be uploaded in the desired format.

Second, upload the data in Microsoft Azure to access it from a remote location or SaaS platform.

Third, ingest the data into SaaS (Snowflake).

After that, create a staging area, create tables, clean, and analyse. Finally, answer the business questions through the existing data in hand.

5. Part 1: Data Ingestion

5.1 Download the Compressed dataset: In this part, using the google drive link, two datasets were downloaded and unzipped in the local machine.

5.2 Upload the dataset in Azure: There were 11 JSON files containing the category data and 11 CSV files containing trending data. All the data was uploaded in Azure container inside a storage account.

5.3 **Ingest the data as external table**: In this process, in the Snowflake, at first, a database was created and used where the entire process can base. A storage integration has been made here with the details, including the type of stage the storage will use, the storage provider, a unique tenant id to identify the user in Azure, and the URL that provides the details of the storage account and container.

Then the required permission was provided to Snowflake from Azure to enable the SaaS platform to access the datasets.

After that, an external stage has been created to define where the storage integration should take place.

For importing all the CSV files into an external table, creating a file format was required beforehand. This step includes a few phases, such as

- defining the type of data to import using that file format.
- the delimiter.
- the number of columns to be discarded to avoid column names from being counted as an observation.
- and how to operate the null values.

For the JSON files, no such step was required.

Finally, upon calling all the CSV files at once and all the JSON files together in another command, two different external tables were created to store the data from Azure.

5.4 **Transfer data from external tables into tables:** In this part, tables were generated to store data in a structured manner. Previously, the external tables contained semi-structured data. The data types and column names were assigned to every column to complete this process.

For the table_youtube_trending, the column names and data types were:

Column in External Table	Name of the column in table	Data type defined
C1	Video_id	varchar
C2	Title	varchar
C3	Publishedat	string
C4	Channelid	varchar
C5	Channeltitle	varchar
C6	Categoryid	int
C7	Trending_date	date
C8	View_count	int
C9	Likes	int

C10	Dislikes	int
C11	Comment_count	int
C12	Comment_disabled	boolean

Table 3: Data types assigned in table_youtube_trending.

For the table_youtube_category, two columns were considered from the external table. Since the data was imported from JSON file, it had built-in key: value relationship. That is why, instead of using column notation as c1/cN, the KEY value will be used to denote the attribute.

Column in External Table	Name of the column in table	Data type defined
Id	Categoryid	int
title	Category_title	varchar

Table 4: Data types assigned in table_youtube_category

In both the tables, a new column was created called "country". It was parsed from the file names in the metadata.

For the table_youtube_category, a challenge was to parse the values from the nested JSON files. The lateral flatten function has been used in a hierarchical form to solve this issue. As the nested file goes into a new leaf, a new lateral flatten is created to locate the key and value. For example, l0(mentioning level 0), l1(level 1), l2(level2).

5.5 Create a final table called "table_youtube_final": In the final table, a new column was incorporated. It was generated using the UUID_STRING() function. This function defines a unique random number to the table so it can be used as a primary key. The tables were merged using left join on the country and categoryid columns; thus, they neither lost any record nor incorporated duplicate values.

6. Part 2: Data Cleaning

6.1 In this step, the categoryid was not considered. However, the duplicate value in the category_title was identified. "Comedy" category had duplicate value in the table_youtube_category.

6.2 In this step, the category_title that only appeared in one country was identified. That was "Nonprofits & Activism". It only appeared in the US.

6.3 In this step, the categoryid of the missing category_title was determined. The categoryid was 29.

6.4 In this step, firstly the category_title of the categoryid 29 was identified which was later used inside the nested query and then the category_title was used to impute all the missing values in the column that had 29 as categoryid. Total 3162 observation was imputed in this process.

6.5 In this step, it was found that, “Kala Official Teaser | Tovino Thomas | Rohith V S | Juvis Productions | Adventure Company” video has no channeltitle.

6.6 In this step, total of 14,619 values were deleted from the table as it had “#NAME?” as a value in the video_id column.

6.7 With the help of CTE and row_number() function, in this step, a new table was created will all the duplicate values. It has total 37,842 observations.

6.8 In this step, total 37,842 observations were deleted from the table_youtube_final. The reference was taken from table_duplicate_values.

6.9 In this step, a total observation of 1,123,017 have been identified in the table_youtube_final.

7. Part 3: Data Analysis

7.1 In this part, for each country in the sports category in the trending_date “2021-10-17”, top 3 videos have been identified. This process was completed using CTE, Rank() function with a partition by country and it was ordered by view_count. All the video that has rank from 1 to 3 was shown in the result.

7.2 In this part, for each country, the count of video_id was identified that has the word “BTS” in the title. It was obtained by contains() function in the where clause and the answers were ordered by the count of the distinct video_id in descending order.

7.3 Here, the most viewed video in each country in a particular date and its like_ratio is identified. It was observed with the help of a CTE. In the CTE dense_rank() was made to find out the most viewed video. This rank was made based on the country and year_month of the video. It was ordered by view_count in descending order.
In the select statement later, All the video having a rank of 1 was brought as outcome. Here, the like ratio was calculated up to 2 decimal points.

7.4 To answer this question, two new tables were incorporated into the database.

- One that finds out the country, category_title and total number of distinct videos per category(total_category_video).
- One that finds out the number of total videos for each country.

In the next part, these two tables have been merged using a LEFT JOIN on the country column. In addition, a new column has been generated in this part that contains the percentage of every category in the total videos in each country. This value was calculated up to 2 decimal points as well. The aim of this query is to find out the number of most distinct videos(video_id) in each country and the percentage of out of the total distinct number of videos of that country.

7.5 Colors TV has produced the most distinct videos and the number of total videos in this channel is 805.

8. Part 4: Business Question

If I were to launch a new YouTube channel tomorrow, I would try to choose the "People and Blogs" category.

To defend my answer, we can look at the category_title and their view_count. I am discarding the likes and dislikes from the list of considerations because, as a user, I never press the impression button, even if I like/dislike the video. Considering many users like me, it can be misleading if we consider the count of likes and dislikes. The table below shows that the number of views is high in the **Gaming** and **People and Blogs** category.

	CATEGORY_TITLE	...	CATEGORY_VIEW_COUNT
1	Music		739,900,339,874
2	Entertainment		519,486,675,387
3	Gaming		195,817,817,788
4	People & Blogs		187,418,325,929
5	Sports		133,651,789,176
6	Comedy		108,493,038,540
7	Science & Technology		61,917,522,815
8	Film & Animation		52,847,803,701
9	Howto & Style		34,743,662,369
10	News & Politics		34,670,377,110
11	Education		28,278,127,210
12	Autos & Vehicles		15,972,078,815
13	Pets & Animals		5,699,093,234
14	Travel & Events		5,670,394,522
15	Nonprofits & Activism		2,770,972,607

Figure 1: Category title and view counts.

If we look at the channels with the highest view counts, we can see **Gaming** and **Sports** primarily**.** If we think for a while, these gaming channels are fairly elder, and the chances of being trendy shortly with a gaming channel are relatively low.

Not picking Sports is because people worldwide are not fans of a single sport. For example, people in Europe are fond of football (Soccer), and people in the USA are fond of American football. In contrast, the people of South Asia are primarily fond of cricket. So, the point of this discussion is that the point of interest varies worldwide. As a result, Sports is not a feasible choice for a new youtube channel.

	CHANNELTITLE	CATEGORY_TITLE	CHANNEL_VIEW_COUNT
1	Brawl Stars	Gaming	12,719,779,741
2	MrBeast Gaming	Gaming	11,901,057,921
3	Apple	Science & Technology	10,217,465,810
4	Dude Perfect	Sports	9,051,719,185
5	Clash of Clans	Gaming	7,724,331,628
6	FORMULA 1	Sports	6,306,436,460
7	cricket.com.au	Sports	5,926,564,716
8	NFL	Sports	5,834,332,005
9	League of Legends	Gaming	5,683,746,083
10	Mark Rober	Science & Technology	5,444,701,165
11	YouTube	Education	5,320,893,963
12	Kimberly Loaiza	People & Blogs	5,063,118,146
13	Dream	Gaming	4,943,881,956
14	Apex Legends	Gaming	4,726,874,625
15	Serie A	Sports	4,591,069,401
16	SSundee	Gaming	4,133,205,338
17	The Tonight Show Starring Jimmy Fallon	Comedy	4,083,381,512
18	Bella Poarch	People & Blogs	3,988,673,396
19	Tsuriki Show	Comedy	3,972,601,513
20	AnthonySenpai	Gaming	3,692,200,878

Figure 2: Channel id, Category Title and View count in the channel.

Now, looking at the table below, we can see the count of the category being trending in a day. It depicts that **People & Blogs** has the highest count in being trendy. With the help of the youtube algorithm, people have been suggested the type of video he usually browses. If I create a new channel tomorrow and start uploading videos on **People and Blogs,** I think it will come forward within a brief period.

	CATEGORY_TITLE	...	TREND_COUNT
1	People & Blogs		134,130
2	Gaming		122,123
3	Sports		112,916
4	Comedy		67,079
5	News & Politics		45,475
6	Howto & Style		37,454
7	Film & Animation		35,796
8	Science & Technology		29,542
9	Autos & Vehicles		25,136
10	Education		21,584
11	Travel & Events		6,972
12	Pets & Animals		6,949
13	Nonprofits & Activism		3,122

Figure 3: Count of being trending for each Category.

The second question concerns whether this decision applies to all the counties; there is an explanation.
We have narrowed down the choice to **People & Blogs** and **Gaming** from the last part**.** Looking at the figure below, we see the popularity of **People & Blogs** and **Gaming.**

	COUNTRY	CATEGORY_TITLE	OUNTRYWISE_CATEGORY_VIEWS	POPULARITY
1	DE	People & Blogs	19,436,702,702	1
2	US	People & Blogs	19,552,738,339	3
3	RU	People & Blogs	8,012,856,672	1
4	BR	People & Blogs	13,300,888,918	1
5	CA	People & Blogs	27,766,335,927	2
6	GB	People & Blogs	22,886,749,012	2
7	IN	People & Blogs	29,447,709,750	1
8	FR	People & Blogs	3,606,448,013	4
9	KR	People & Blogs	13,515,625,109	1
10	MX	People & Blogs	19,965,612,883	2
11	JP	People & Blogs	9,926,658,604	1

Figure 4: Popularity of People& Blogs in all the countries.

	COUNTRY	CATEGORY_TITLE	COUNTRYWISE_CATEGORY_VIEWS	POPULARITY
1	DE	Gaming	16,690,061,893	2
2	BR	Gaming	13,212,566,710	2
3	CA	Gaming	36,677,437,270	1
4	JP	Gaming	9,455,565,338	2
5	KR	Gaming	4,795,341,429	5
6	MX	Gaming	22,537,854,653	1
7	US	Gaming	38,792,528,568	1
8	RU	Gaming	4,204,169,047	4
9	FR	Gaming	7,726,931,818	2
10	GB	Gaming	31,878,388,652	1
11	IN	Gaming	9,846,972,410	4

Figure 4: Popularity of Gaming in all the countries.

If we compare these two figure we see the ranking comparison in popularity in People & Blogs and Gaming.

Rank	People & Blogs Score	Gaming Score
1	6	4
2	3	4
3	1	0
4	1	2
5	0	1

It is evident from this table that People & Blogs is more famous than Gaming.

So, If I were to create a youtube channel tomorrow, I will go for the category "**People & Blogs**".

9. Issues faced in this analysis

9.1 An issue was to extract values from the JSON files. Being NESTED in nature, it was not feasible to extract the information with a single **Lateral Flatten.** As a result, I needed to create a hierarchy in the lateral flatten. For the root level, l0 or lateral flatten(value) was helpful. However, Each time the value was inside another nested key, another new level of Lateral flatten was created to access the value.

For example,

Level 0 – l0 – lateral flatten(value) as l

Level 1 – l1 – lateral flatten(l0.value) as l1

Level 2 – l2 – lateral flatten(l1.value) as l2.

In this hierarchical manner, the issue was resolved.