Big Data

Youtube Exploratory Data Analysis Using Pyspark and Kibana Dashboard

Project Report

Team Introduction

- Anshul Agrawal (ava8249)
- Nikhil Mane (nm3765)
- Rahul Ramaswamy (rr4059)

Motivation

YouTube is a free video sharing website that makes it easy to watch online videos. You can even create and upload your own videos to share with others. It is a community-based space to find and contribute answers to technical challenges and is one of the most popular websites in the world.

Being a popular website, many interesting insights can be drawn from YouTube data which can be further used to perform different analysis. The analysis of this data can help us answer some questions about the viewer's video preference

Problem Statement

Perform Exploratory Data Analysis on Youtube data for countries like India, United States of America, etc. on Pyspark and produce visualizations on Kibana dashboard. We analyze the data to obtain insights on country level, few of which are-

- Channel with most views, likes, dislikes, comments
- Distribution of videos across categories
- Top channels and its statistics (mean/max/sum) for views, likes, dislikes, comments, etc.
- Most watched/uploaded categories of videos

Data Acquisition

The data used for this project is collected from the available Youtube data API. This API provides access to data like the view count, likes, dislikes, category of videos, title and description of videos etc. We cleaned this data of redundant and null values for analysis.

Data Attributes

- Video_id
- Title
- PublishedAt
- ChannelId
- ChannelTitle
- CategoryId
- Trending_date

- Tags
- View_count
- Likes
- Dislikes
- Comment_count
- Thumbnail_link
- Comments_disabled
- Ratings_disabled

Data Preparation

We need to explore data before analyzing it. First, we performed data cleaning. Youtube provides their data in a rather clean format, so our data cleaning involves only removing unnecessary symbols, and columns, retrieving corresponding categories from ID, grouping data, and handling data types.

1. **Preprocessing of data**
   Unnecessary columns like Description, comments disabled etc, are not really useful for us. So we get rid of them.

2. **Retrieving the category name from ID**
   The data has a category ID field which is an integer. There is a json file available to read the corresponding category name. We need to replace the category ID with its corresponding name by reading the json file.

3. **Datatype of columns**
   The data provided by the stack overflow data archive had columns like 'ViewCount' as strings. Such columns were converted to Integer datatypes to make further computations possible. Converted published date/time to the right format which can be pushed to ES.

4. **Grouping of data**
   We group the videos based on their categories and country for both India and USA to get insights on the viewing behavior in the 2 countries.

High level design overview

Building A Data Analysis Module

Using Pyspark and Elasticsearch, we have built a data analysis module that gives interesting analysis and an interactive dashboard to view and understand custom data trends and visualizations.

1. Ingesting the acquired data.

The data that is obtained from the youtube api. This data is converted into CSV format and read into Pyspark for further processing.

2. Processing the data using Pyspark.

The appropriate preprocessing is performed on the ingested data before it is loaded into a Pyspark data frame. The data is ready to be given to Elastic search after it is fully processed and available in Pyspark.

3. Indexing data into Elasticsearch

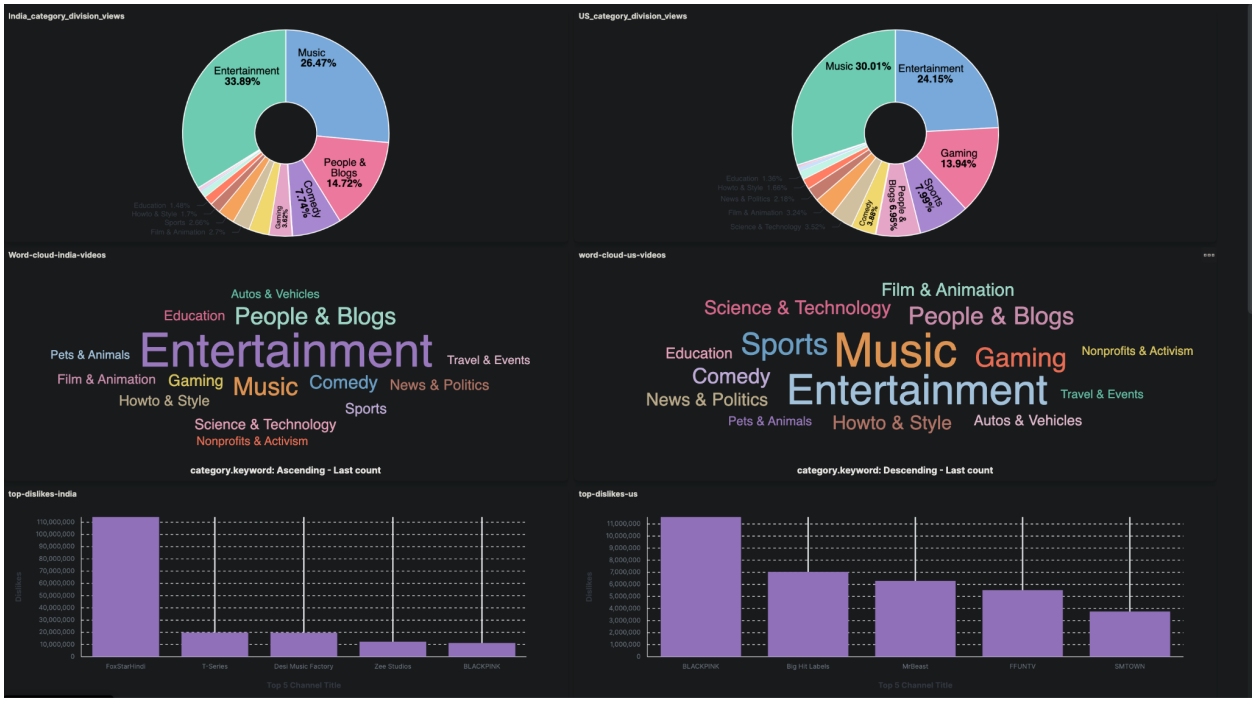
After processing, the data is added to the Elasticsearch index. The Elastic ES-Hadoop library, an open-source, standalone, self-contained, tiny library that enables Hadoop operations to connect with Elasticsearch, is used to integrate Elasticsearch and Pyspark. In the form of an RDD (Resilient Distributed Dataset) (or Pair RDD, to be precise), Elasticsearch-Hadoop offers

direct interaction between Elasticsearch and Apache Spark and can read data from Elasticsearch.

4. Visualizing the data in Kibana

Finally, we use Kibana to query data from the Elasticsearch index and create an interactive dashboard to visualize the data and show metrics like categories, top liked channels, top disliked channels, etc.

Final Output



Why Big Data?

- Youtube videos get uploaded/deleted and viewed/liked/disliked every second.
- All this data needs to be pre processed, analyzed and presented in a visually appealing manner to the user.
- Kafka streaming helps us fetch updated data in real time.
- Spark helps process this data in-memory, parallelly thereby speeding up the process.
- ES Hadoop helps us push this analyzed data to Elastic cloud.
- Elastic Cloud helps in quickly searching for the required data, which is sent to Kibana for visualizations.

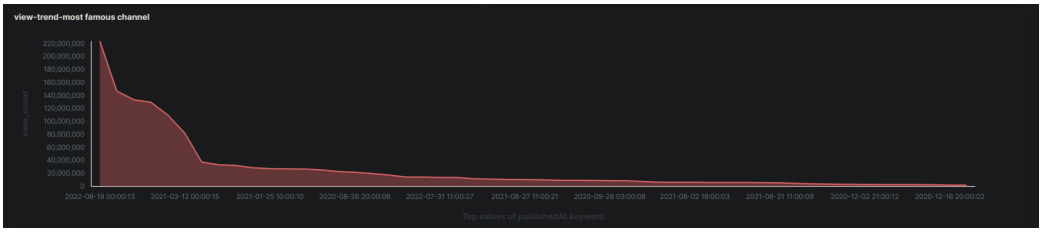
What tools?

- PySpark
- ElasticSearch Hadoop
- Elastic Cloud
- Kibana

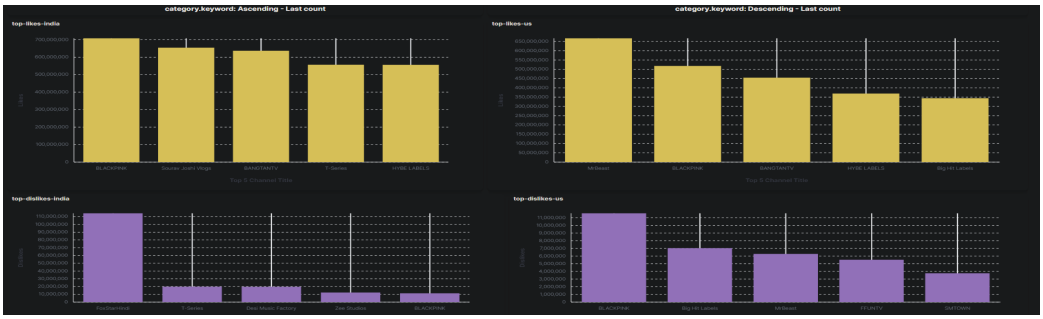
Insights / Reports

- We compare the video consumption between US and India

  - Entertainment is a more popular category in India while Music is more popular in the USA



  - The most popular channel overall (Blackpink) is on a decline and its viewership is drastically reducing.



  - The most liked video in India , Blackpink is also the most disliked video in the USA.

Lessons learned

- How to read ever-updated data on a real time basis.
- How to process and condense huge amounts of data quickly using spark.

- Use ES Hadoop to push data to ElasticSearch
- Utilize Elastic Search for quick computations and presentation on Kibana Dashboards.

Future Work

- Data for more countries can be added
- Paid YouTube API can be used for realtime limitless feed of data
- The data can be fetched in real time from the Youtube API using Kafka and sent to a topic from where Spark can read the same as an input.