
# Mini Project Report

***on***

# "YOUTUBE DATA ANALYSIS"

**Submitted by**

| | | |
|---|---|---|
| Atharv Nikam | 1032180297 | PA10 |
| Ketaki Patil | 1032180421 | PA17 |
| Saisrijan Gupta | 1032180626 | PA27 |
| Anjali Nair | 1032180751 | PA31 |
| Shivraj Patil | 1032181709 | PA51 |

**Under the Guidance of**

**Prof. Vasundhara Ghate**

**At**

**School of Computer Engineering and Technology**

# ABSTRACT

There is a tremendous growth and popularity of YouTube. It has the potential to touch billions of lives globally as the no. of YouTube users is growing day by day. YouTube, owned by Google, a video streaming site which has billions of users and 400 hours of videos are being uploaded every minute. Almost billions of videos are watched on YouTube every single day, generating a mammoth amount of data daily. Since YouTube data is generally in unstructured form, there is an increased demand to store, process and analyze such real time Big Data. YouTubers can analyze their own channel performance with YouTube Analytics. But one can not analyze other channels. The proposed system uses the MapReduce framework of Hadoop for processing and analyzing real time YouTube datasets. It will help in discovering how competitors are performing on YouTube. One can easily identify what content works best on YouTube. These analytical data can be represented in demographic form which can be used by individuals and organizations for making immediate actionable decisions so as to gain competitive advantages.

# CONTENTS

# LIST OF FIGURES

# 1. INTRODUCTION

YouTube, owned by Google, is the platform where people can upload, watch and share videos with others. They can also give their feedback accordingly through comments. YouTube has attracted users across the globe such as advertisers, various media, politicians and other interest groups. 400 hours of videos are uploaded on YouTube every minute. Billions of videos are watched on YouTube every day. This indicates that YouTube has the great potential to reach out to many people. Thus generating massive amounts of data daily. Such a mammoth amount of data is usually in unstructured form. Proposed system presents analytics of YouTube datasets in demographic form using the Apache Hadoop MapReduce framework. It will help its users make targeted real time and informed decisions.

## 1.1 MOTIVATION

YouTube is one of the most popular and engaging social media tools for uploading, viewing videos and an amazing platform that reveals the users response through comments for published videos, number of likes, dislikes, number of subscribers for a particular channel. YouTube collects a wide variety of traditional data points including View Counts, Likes, and Comments. The analysis of the above listed data points makes a very interesting data source to extract implicit knowledge about users, videos, categories and community interests. In Today's world as the 4 V's of Big data (Volume, Variety, Velocity & Veracity) are very rapidly increasing it has become a must to come up with meaningful insights in order to project and derive meanings from the data to help organizations grow business rapidly. To process this large amount of data, a high-level parallel and distributed system such as Hadoop is necessary. The data obtained from YouTube can be used for analysis and various visualizations. The information generated by this analysis can be used by multiple organizations to make key decisions about their services, targets to improve the business value.

# 2. PROBLEM STATEMENT

## 2.1 PROBLEM DEFINITION

"YouTube has over a billion users and every day people watch hundreds of millions of hours on YouTube and generate billions of views". Every day, people across the world are uploading 1.2 million videos to YouTube, or over 100 hours per minute and this number is ever increasing. To analyze and understand the activity occurring on such a massive scale, a relational SQL database is not enough. Such kind of data is well suited to a massively parallel and distributed system like Hadoop.

## 2.2 OBJECTIVES

The main objective of this project is to focus on how data generated from YouTube can be mined and utilized by different companies to make targeted, real time and informed decisions about their product that can increase their market share.

# 3.TOOLS

## 3.1 Hadoop

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from a single server to thousands of machines, each offering local computation and storage.

### 3.1.1 Modules of Hadoop

1. **HDFS**: Hadoop Distributed File System. Google published its paper GFS and on the basis of that HDFS was developed. It states that the files will be broken into blocks and stored in nodes over the distributed architecture.
2. **Yarn:** Yet another Resource Negotiator is used for job scheduling and managing the cluster.
3. **Map Reduce:** This is a framework which helps Java programs to do the parallel computation on data using key value pairs. The Map task takes input data and converts it into a data set which can be computed in Key value pairs. The output of Map task is consumed by reduce task and then the output of reducer gives the desired result.
4. **Hadoop Common**: These Java libraries are used to start Hadoop and are used by other Hadoop modules.

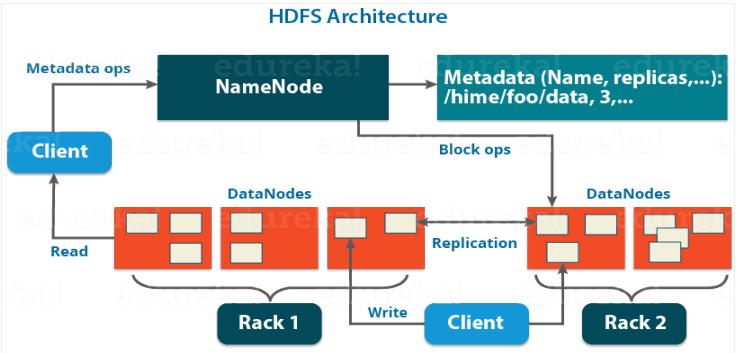



Fig 1.:HDFS Architecture

## 3.2 Map Reduce

The MapReduce is a paradigm which has two phases, the mapper phase, and the reducer phase. In the Mapper, the input is given in the form of a key-value pair. The output of the Mapper is fed to the reducer as input. The reducer runs only after the Mapper is over. The reducer too takes input in key-value format, and the output of the reducer is the final output.

### 3.2.1Steps in Map Reduce

1. The map takes data in the form of pairs and returns a list of <key, value> pairs. The keys will not be unique in this case.
2. Using the output of Map, sort and shuffle are applied by the Hadoop architecture. This sort and shuffle acts on these lists of <key, value> pairs and sends out unique keys and a list of values associated with this unique key <key, list(values)>.
3. An output of sort and shuffle sent to the reducer phase. The reducer performs a defined function on a list of values for unique keys, and Final output <key, value> will be stored/displayed.

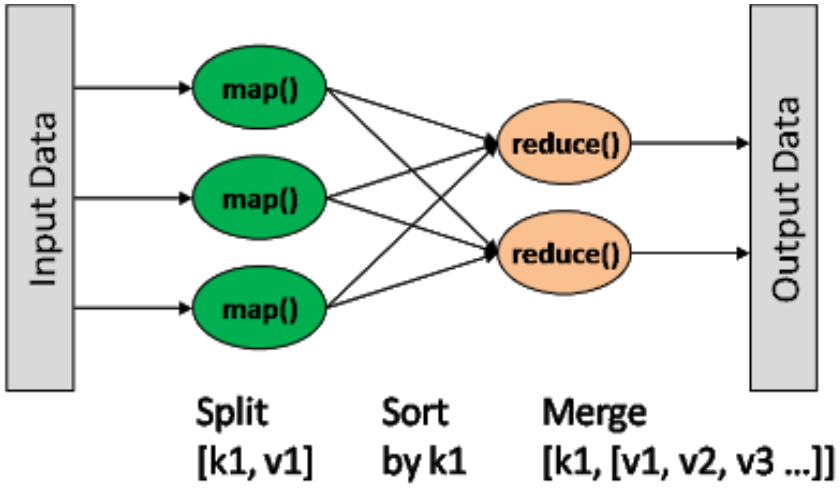



Fig 2.: Map Reduce

## 3.3 Node js

Node.js is a cross-platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications. It is open source and free to use.Many of the basic modules of Node.js are written in JavaScript. Node.js is mostly used to run real-time server applications.

## Features of Node.js

Following is a list of some important features of Node.js that makes it the first choice of software architects.

1. **Extremely fast:** Node.js is built on Google Chrome's V8 JavaScript Engine, so its library is very fast in code execution.
2. **I/O is Asynchronous and Event Driven:** All APIs of Node.js library are asynchronous i.e. non-blocking. So a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.
3. **Single threaded:** Node.js follows a single threaded model with event looping.
4. **Highly Scalable:** Node.js is highly scalable because the event mechanism helps the server to respond in a non-blocking way.
5. **No buffering:** Node.js cuts down the overall processing time while uploading audio and video files. Node.js applications never buffer any data. These applications simply output the data in chunks.
6. **Open source:** Node.js has an open source community which has produced many excellent modules to add additional capabilities to Node.js applications.

## 3.4 Cassandra

Apache Cassandra is an open source, distributed and decentralized/distributed storage system (database), for managing very large amounts of structured data spread out across the world. It provides highly available service with no single point of failure.

### Features of Cassandra

Cassandra has become so popular because of its outstanding technical features. Given below are some of the features of Cassandra:

1. **Elastic scalability** − Cassandra is highly scalable; it allows to add more hardware to accommodate more customers and more data as per requirement.
2. **Always on architecture** − Cassandra has no single point of failure and it is continuously available for business-critical applications that cannot afford a failure.
3. **Fast linear-scale performance** − Cassandra is linearly scalable, i.e., it increases your throughput as you increase the number of nodes in the cluster. Therefore it maintains a quick response time.
4. **Flexible data storage** − Cassandra accommodates all possible data formats including: structured, semi-structured, and unstructured. It can dynamically accommodate changes to your data structures according to your needs.
5. **Easy data distribution** − Cassandra provides the flexibility to distribute data where you need by replicating data across multiple data centers.
6. **Transaction support** − Cassandra supports properties like Atomicity, Consistency, Isolation, and Durability (ACID).
7. **Fast writes** − Cassandra was designed to run on cheap commodity hardware. It performs blazingly fast writes and can store hundreds of terabytes of data, without sacrificing the read efficiency.

### 3.4.1 Data Model of Cassandra

1. **Cluster**

   Cassandra database is distributed over several machines that operate together. The outermost container is known as the Cluster. For failure handling, every node contains a replica, and in case of a failure, the replica takes charge. Cassandra arranges the nodes in a cluster, in a ring format, and assigns data to them.

2. **Keyspaces**

   Cassandra data model consists of keyspaces at the highest level. Keyspaces are the containers of data, similar to the schema or database in a relational database. Typically, keyspaces contain many tables.

3. **Column Family**

   A column family is a container for an ordered collection of rows. Each row, in turn, is an ordered collection of columns.

4. **Table**

   Within the keyspaces, the tables are defined. Tables are also referred to as Column Families in the earlier versions of Cassandra. Tables contain a set of columns and a primary key, and they store data in a set of rows.

### 3.4.2 Cassendra Query Language

Cassandra Query Language (CQL) is used to access Cassandra through its nodes. CQL treats the database (Keyspace) as a container of tables. Programmers use cqlsh: a prompt to work with CQL or separate application language drivers.The client can approach any of the nodes for their read-write operations. That node (coordinator) plays a proxy between the client and the nodes holding the data.

## 3.5 Power BI

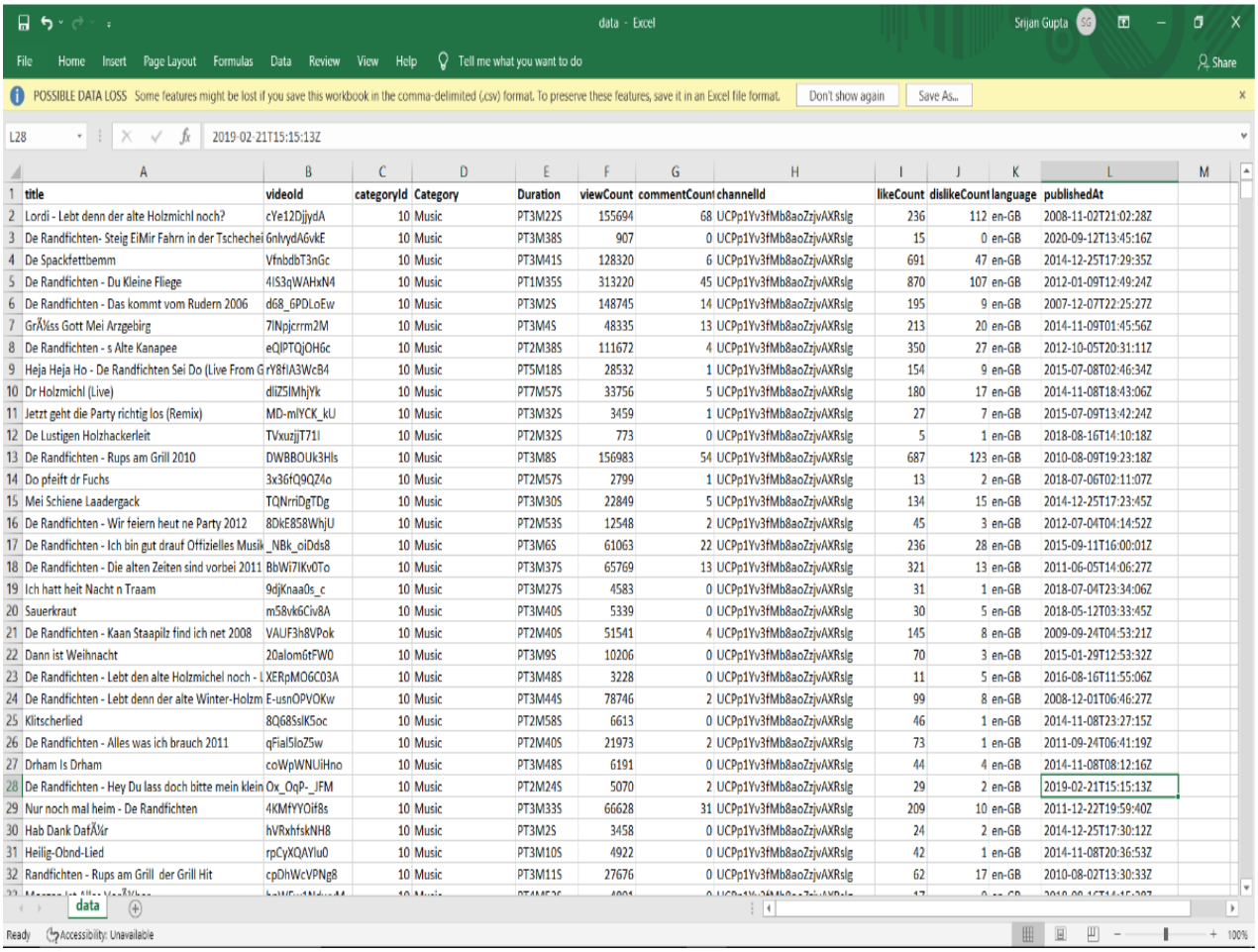
Microsoft Power BI is a business intelligence platform that provides nontechnical business users with tools for aggregating, analyzing, visualizing and sharing data. Power BI's user interface is fairly intuitive for users familiar with Excel and its deep integration with other Microsoft products makes it a very versatile self-service tool that requires little upfront training.

Microsoft Power BI is used to find insights within data. Power BI can help connect disparate data sets, transform and clean the data into a data model and create charts or graphs to provide visuals of the data. All of this can be shared with other Power BI users.

The data models created from Power BI can be used in several ways for organizations, including telling stories through charts and data visualizations and examining "what if" scenarios within the data. Power BI reports can also answer questions in real time and help with forecasting to make sure departments meet business metrics.


Fig 3.: Features of Power BI

# 4. DATASET



data - Excel

Srijan Gupta

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do Share

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

L28 2019-02-21T15:15:13Z

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | title | videoId | categoryId | Category | Duration | viewCount | commentCount | channelId | likeCount | dislikeCount | language | publishedAt | |
| 2 | Lordi - Lebt denn der alte Holzmichl noch? | cYe12DjjydA | 10 | Music | PT3M22S | 155694 | 68 | UCPp1Yv3fMb8aoZzjvAXRsIg | 236 | 112 | en-GB | 2008-11-02T21:02:28Z | |
| 3 | De Randfichten- Steig EiMir Fahrn in der Tschechei | 6nhvydA6vkE | 10 | Music | PT3M38S | 907 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 15 | 0 | en-GB | 2020-09-12T13:45:16Z | |
| 4 | De Spackfettbemm | VfnbdbT3nGc | 10 | Music | PT3M41S | 128320 | 6 | UCPp1Yv3fMb8aoZzjvAXRsIg | 691 | 47 | en-GB | 2014-12-25T17:29:35Z | |
| 5 | De Randfichten - Du Kleine Fliege | 4IS3qWAHxN4 | 10 | Music | PT1M35S | 313220 | 45 | UCPp1Yv3fMb8aoZzjvAXRsIg | 870 | 107 | en-GB | 2012-01-09T12:49:24Z | |
| 6 | De Randfichten - Das kommt vom Rudern 2006 | d68_6PDLoEw | 10 | Music | PT3M2S | 148745 | 14 | UCPp1Yv3fMb8aoZzjvAXRsIg | 195 | 9 | en-GB | 2007-12-07T22:25:27Z | |
| 7 | GrÃ¼ss Gott Mei Arzgebirg | 7INpjcrrm2M | 10 | Music | PT3M4S | 48335 | 13 | UCPp1Yv3fMb8aoZzjvAXRsIg | 213 | 20 | en-GB | 2014-11-09T01:45:56Z | |
| 8 | De Randfichten - s Alte Kanapee | eQlPTQjOH6c | 10 | Music | PT2M38S | 111672 | 4 | UCPp1Yv3fMb8aoZzjvAXRsIg | 350 | 27 | en-GB | 2012-10-05T20:31:11Z | |
| 9 | Heja Heja Ho - De Randfichten Sei Do (Live From G | rY8fiA3WcB4 | 10 | Music | PT5M18S | 28532 | 1 | UCPp1Yv3fMb8aoZzjvAXRsIg | 154 | 9 | en-GB | 2015-07-08T02:46:34Z | |
| 10 | Dr Holzmichl (Live) | diiZSIMhjYk | 10 | Music | PT7M57S | 33756 | 5 | UCPp1Yv3fMb8aoZzjvAXRsIg | 180 | 17 | en-GB | 2014-11-08T18:43:06Z | |
| 11 | Jetzt geht die Party richtig los (Remix) | MD-mlYCK_kU | 10 | Music | PT3M32S | 3459 | 1 | UCPp1Yv3fMb8aoZzjvAXRsIg | 27 | 7 | en-GB | 2015-07-09T13:42:24Z | |
| 12 | De Lustigen Holzhackerleit | TVxuzjjT71I | 10 | Music | PT2M32S | 773 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 5 | 1 | en-GB | 2018-08-16T14:10:18Z | |
| 13 | De Randfichten - Rups am Grill 2010 | DWBBOUk3Hls | 10 | Music | PT3M8S | 156983 | 54 | UCPp1Yv3fMb8aoZzjvAXRsIg | 687 | 123 | en-GB | 2010-08-09T19:23:18Z | |
| 14 | Do pfeift dr Fuchs | 3x36fQ9QZ4o | 10 | Music | PT2M57S | 2799 | 1 | UCPp1Yv3fMb8aoZzjvAXRsIg | 13 | 2 | en-GB | 2018-07-06T02:11:07Z | |
| 15 | Mei Schiene Laadergack | TQNrriDgTDg | 10 | Music | PT3M30S | 22849 | 5 | UCPp1Yv3fMb8aoZzjvAXRsIg | 134 | 15 | en-GB | 2014-12-25T17:23:45Z | |
| 16 | De Randfichten - Wir feiern heut ne Party 2012 | 8DkE8S8WhjU | 10 | Music | PT2M53S | 12548 | 2 | UCPp1Yv3fMb8aoZzjvAXRsIg | 45 | 3 | en-GB | 2012-07-04T04:14:52Z | |
| 17 | De Randfichten - Ich bin gut drauf Offizielles Musik | _NBk_oiDds8 | 10 | Music | PT3M6S | 61063 | 22 | UCPp1Yv3fMb8aoZzjvAXRsIg | 236 | 28 | en-GB | 2015-09-11T16:00:01Z | |
| 18 | De Randfichten - Die alten Zeiten sind vorbei 2011 | BbWi7IKv0To | 10 | Music | PT3M37S | 65769 | 13 | UCPp1Yv3fMb8aoZzjvAXRsIg | 321 | 13 | en-GB | 2011-06-05T14:06:27Z | |
| 19 | Ich hatt heit Nacht n Traam | 9djKnaa0s_c | 10 | Music | PT3M27S | 4583 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 31 | 1 | en-GB | 2018-07-04T23:34:06Z | |
| 20 | Sauerkraut | m58vk6Civ8A | 10 | Music | PT3M40S | 5339 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 30 | 5 | en-GB | 2018-05-12T03:33:45Z | |
| 21 | De Randfichten - Kaan Staapilz find ich net 2008 | VAlJF3h8VPok | 10 | Music | PT2M40S | 51541 | 4 | UCPp1Yv3fMb8aoZzjvAXRsIg | 145 | 8 | en-GB | 2009-09-24T04:53:21Z | |
| 22 | Dann ist Weihnacht | 20alom6tFW0 | 10 | Music | PT3M9S | 10206 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 70 | 3 | en-GB | 2015-01-29T12:53:32Z | |
| 23 | De Randfichten - Lebt den alte Holzmichel noch - L | XERpMO6C03A | 10 | Music | PT3M48S | 3228 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 11 | 5 | en-GB | 2016-08-16T11:55:06Z | |
| 24 | De Randfichten - Lebt denn der alte Winter-Holzm | E-usnOPVOKw | 10 | Music | PT3M44S | 78746 | 2 | UCPp1Yv3fMb8aoZzjvAXRsIg | 99 | 8 | en-GB | 2008-12-01T06:46:27Z | |
| 25 | Klitscherlied | 8Qj68SsIK5oc | 10 | Music | PT2M58S | 6613 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 46 | 1 | en-GB | 2014-11-08T23:27:15Z | |
| 26 | De Randfichten - Alles was ich brauch 2011 | qFial5loZ5w | 10 | Music | PT2M40S | 21973 | 2 | UCPp1Yv3fMb8aoZzjvAXRsIg | 73 | 1 | en-GB | 2011-09-24T06:41:19Z | |
| 27 | Drham Is Drham | coWpWNUiHno | 10 | Music | PT3M48S | 6191 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 44 | 4 | en-GB | 2014-11-08T08:12:16Z | |
| 28 | De Randfichten - Hey Du lass doch bitte mein klein | Ox_OqP-_JFM | 10 | Music | PT2M24S | 5070 | 2 | UCPp1Yv3fMb8aoZzjvAXRsIg | 29 | 2 | en-GB | 2019-02-21T15:15:13Z | |
| 29 | Nur noch mal heim - De Randfichten | 4KMfYYOif8s | 10 | Music | PT3M33S | 66628 | 31 | UCPp1Yv3fMb8aoZzjvAXRsIg | 209 | 10 | en-GB | 2011-12-22T19:59:40Z | |
| 30 | Hab Dank DafÃ¼r | hVRxhfskNH8 | 10 | Music | PT3M2S | 3458 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 24 | 2 | en-GB | 2014-12-25T17:30:12Z | |
| 31 | Heilig-Obnd-Lied | rpCyXQAYlu0 | 10 | Music | PT3M10S | 4922 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 42 | 1 | en-GB | 2014-11-08T20:36:53Z | |
| 32 | Randfichten - Rups am Grill der Grill Hit | cpDhWcVPNg8 | 10 | Music | PT3M11S | 27676 | 0 | UCPp1Yv3fMb8aoZzjvAXRsIg | 62 | 17 | en-GB | 2010-08-02T13:30:33Z | |

data

Ready Accessibility: Unavailable 100%

Fig.4:CSV Dataset

# 5. SYSTEM ARCHITECTURE

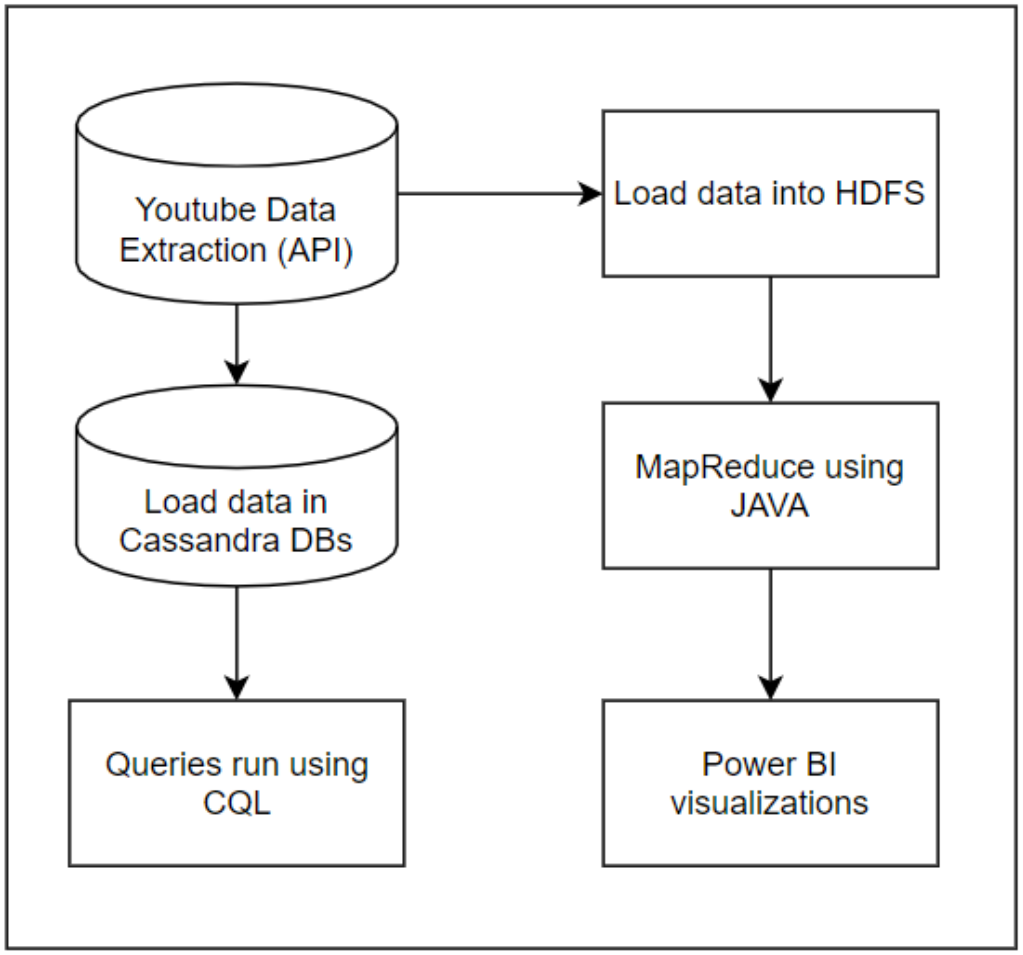



Fig.5: System Architecture for Youtube Data Analysis

# 6. OUTPUT ANALYSIS

## 6.1 Fetch Data using Node JS

- **Selection of duration from front-end UI:**

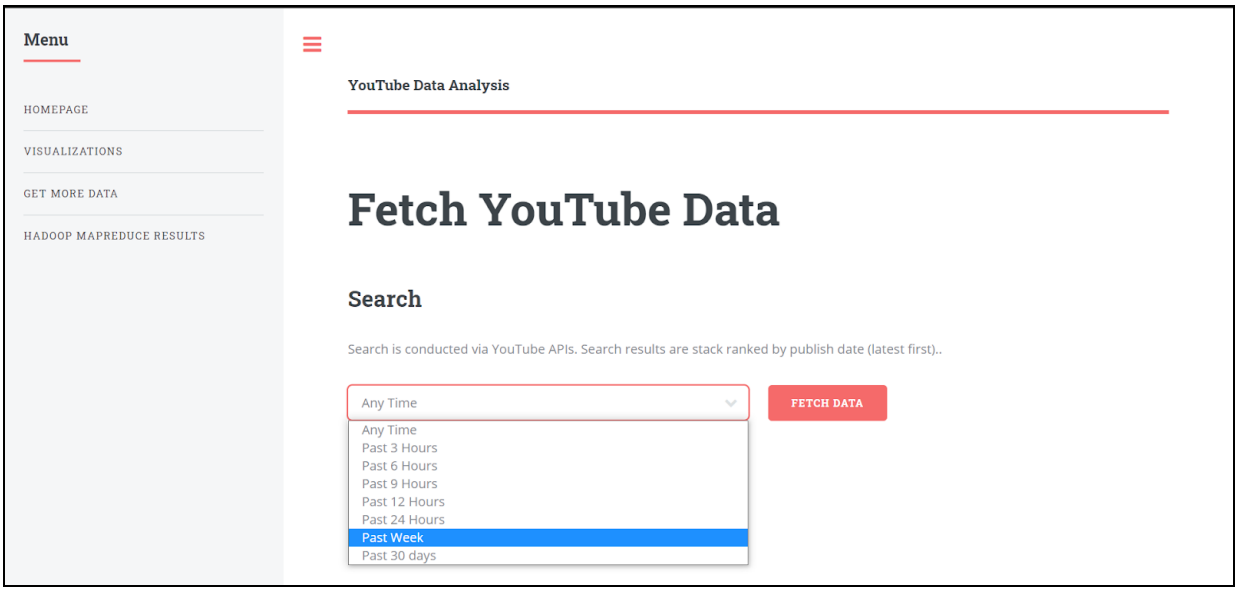



Fig.6:Front-End UI

- **Command prompt output:**

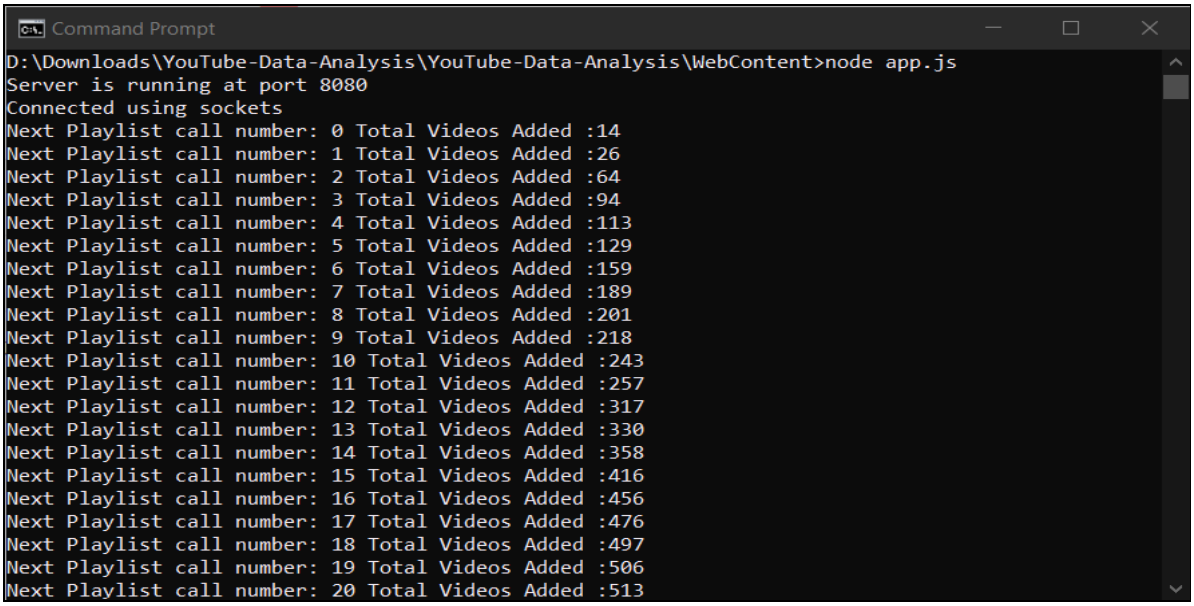



Fig.7:Data Extraction using Node js

## 6.2 Hadoop MapReduce

- **Category**

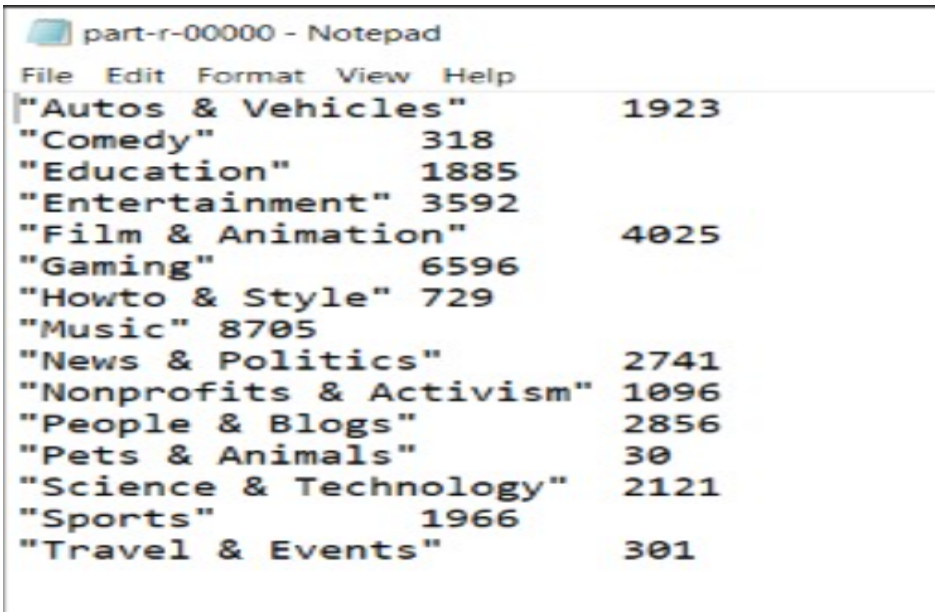



Fig.8:Categories-wise Videos

- **Uploaders**

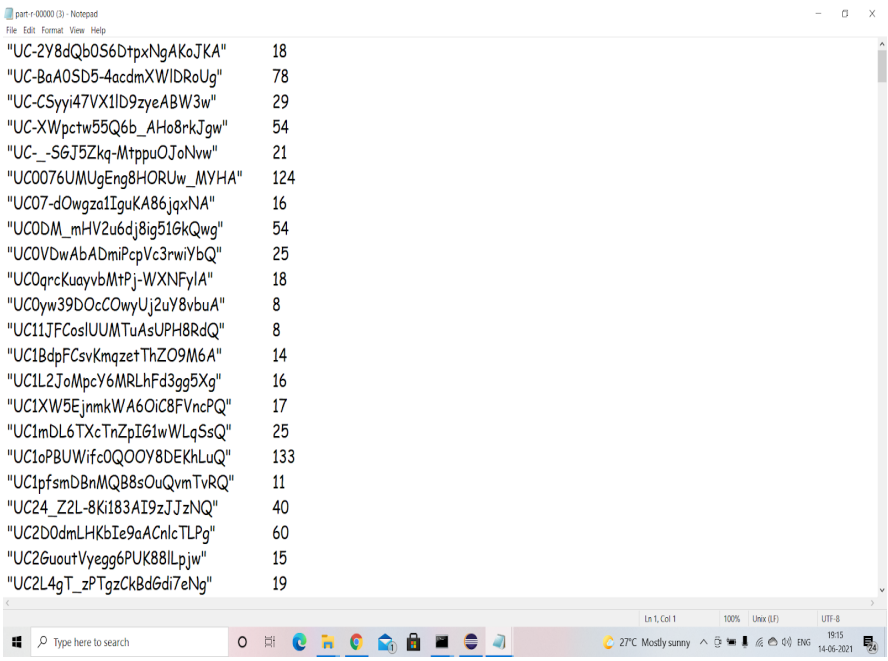



Fig.9:Videos Uploaded by Channel

- **Views**

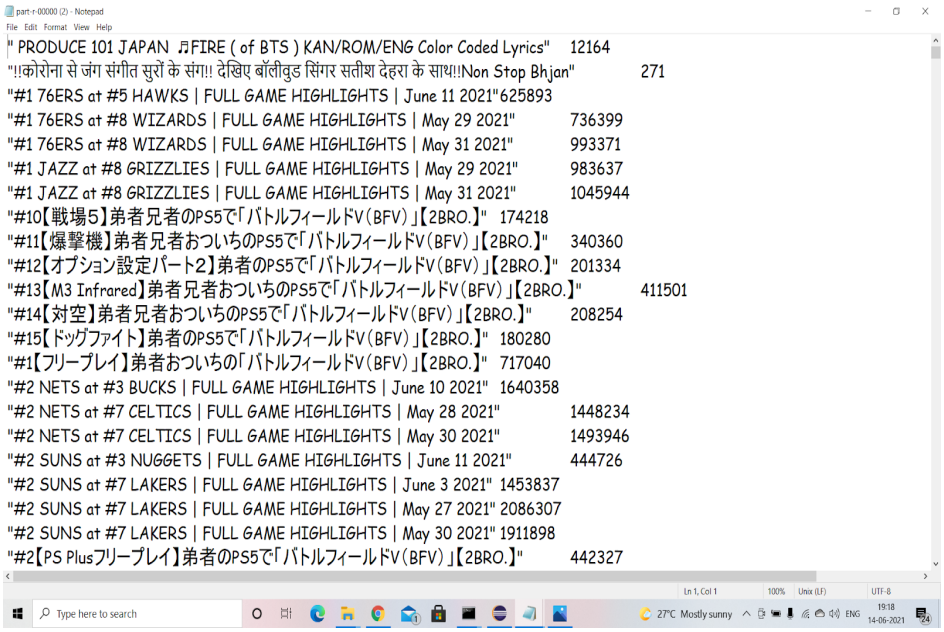



Fig.9:Views received by each video

## 6.3 Cassandra Queries

- **Table Creation and Import of Dataset**

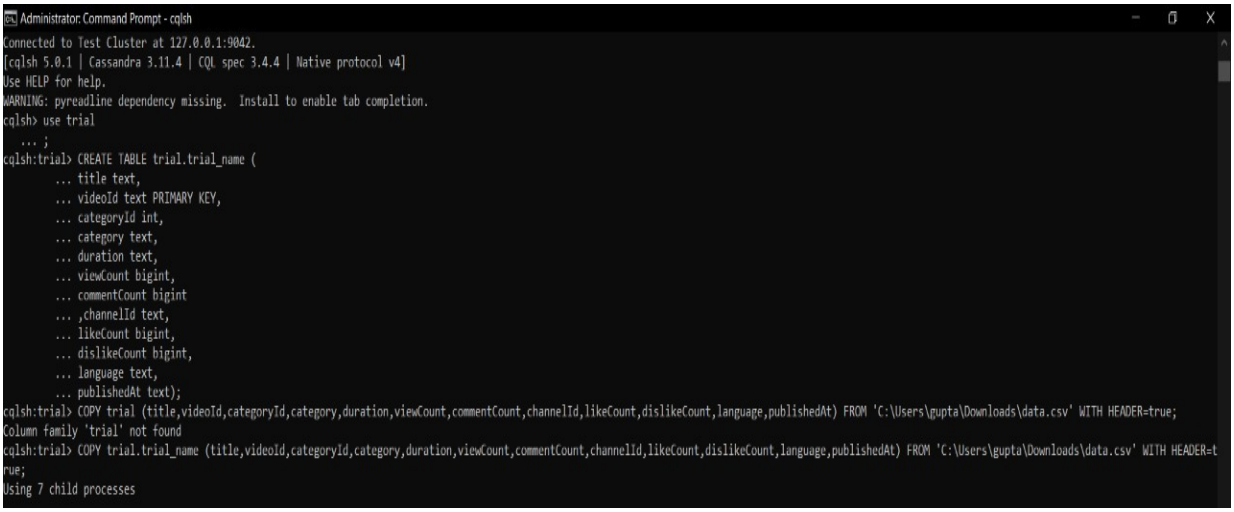



Fig.10:Table Creation and Dataset Import

- **Dataset imported in Cassandra DBs**

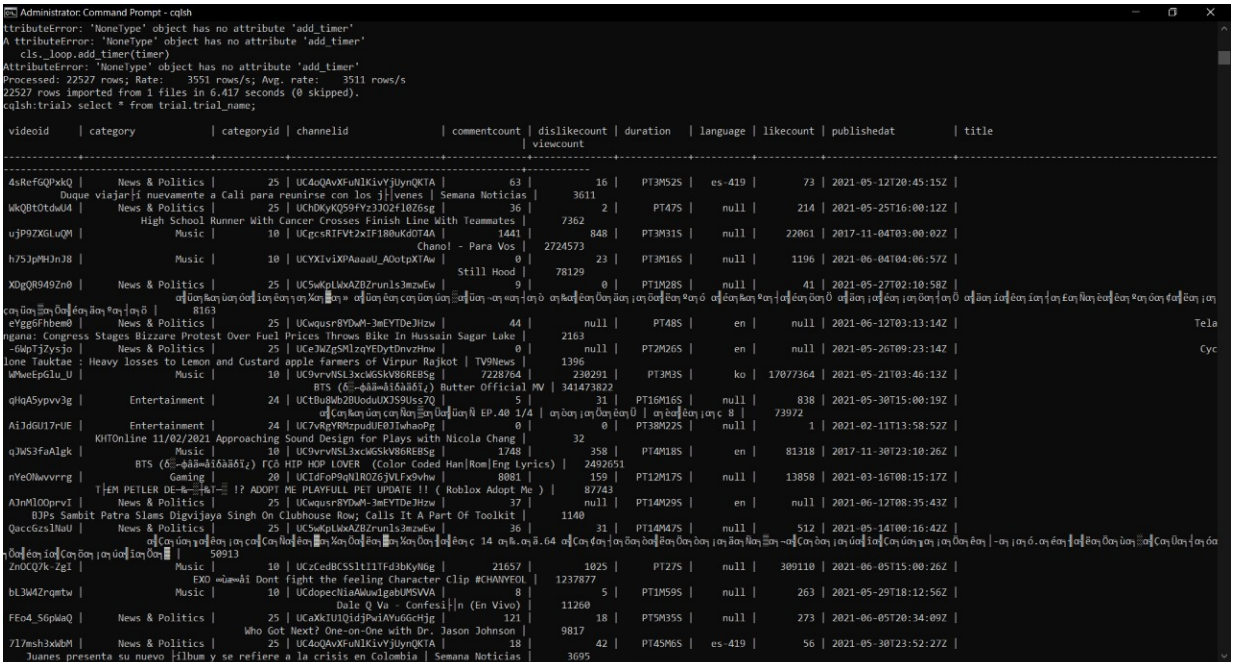



Fig.11:Displaying all records

- **Aggregation queries using CQL**

## Video id and view count from gaming category

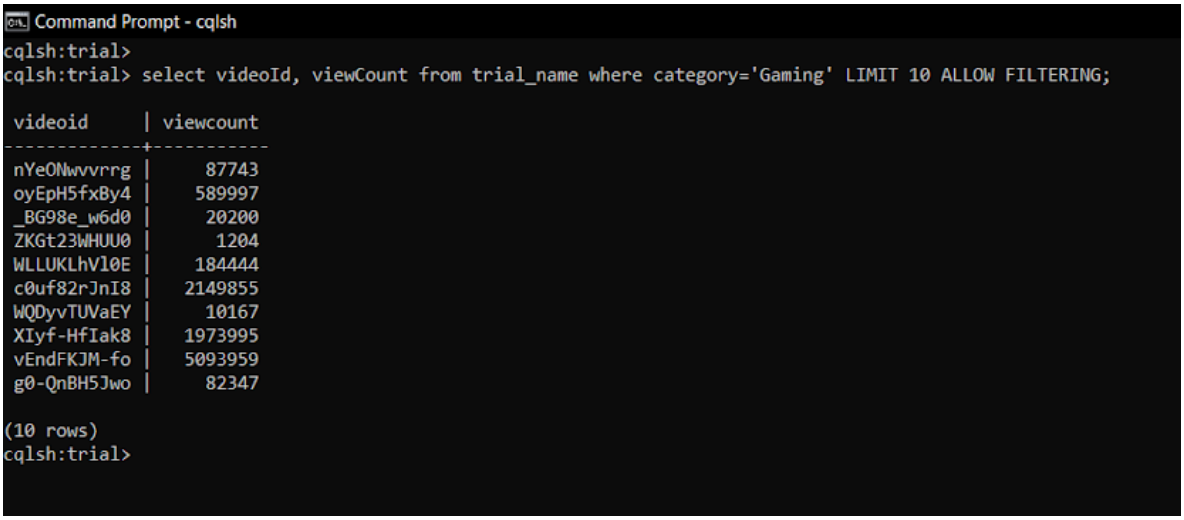



Fig.12:Video id and view count for gaming

## Delete a video from gaming category

```
cqlsh:trial> DELETE FROM trial_name where videoid='ZKGt23WHUU0';
cqlsh:trial> select videoId, viewCount from trial_name where category='Gaming' LIMIT 10 ALLOW FILTERING;

 videoid     | viewcount
-------------+-----------
 nYeONwvvrrg |     87743
 oyEpH5fxBy4 |    589997
 _BG98e_w6d0 |     20200
 WLLUKLhVl0E |    184444
 c0uf82rJnI8 |   2149855
 WQDyvTUVaEY |     10167
 XIyf-HfIak8 |   1973995
 vEndFKJM-fo |   5093959
 g0-QnBH5Jwo |     82347
 qeOooiDR27g |     15721

(10 rows)
cqlsh:trial>
```

Fig.13:Delete one record from gaming

## Maximum likes obtained by any video

```
cqlsh:trial> select min(likeCount) from trial_name;

 system.min(likecount)
-----------------------
                     0

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:trial>
```

Fig.14:Maximum likes obtained by any video

## Maximum likes obtained by any video

```
cqlsh:trial> select max(likeCount) from trial_name;

 system.max(likecount)
-----------------------
              28793939

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:trial>
```

Fig.15: Minimum likes obtained by any video

## 6.4 Frontend

- **Homepage:**

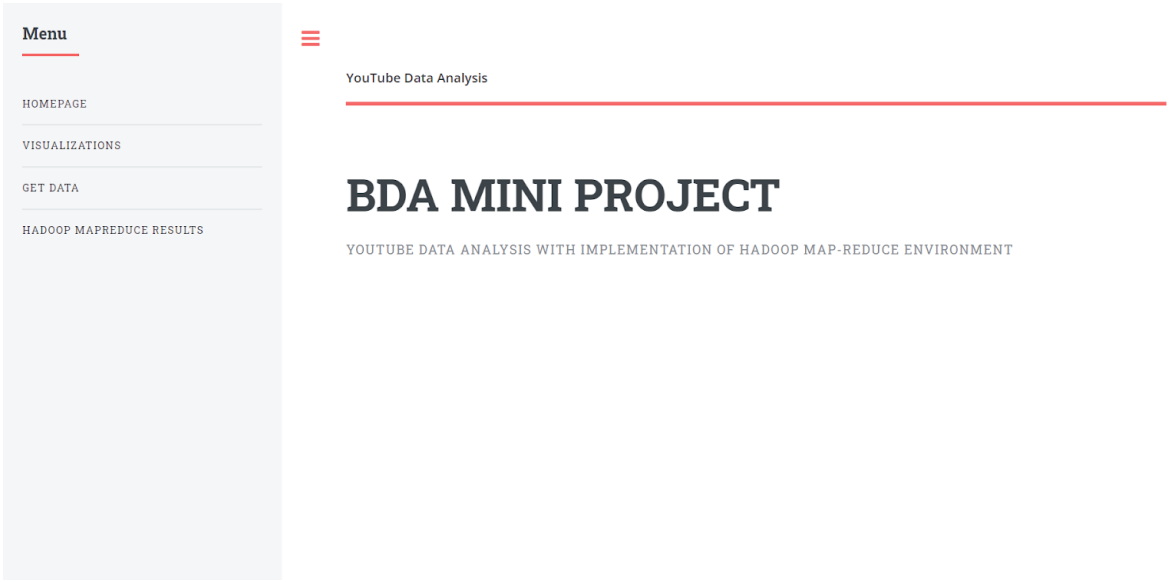



Fig.16: Homepage

- **PowerBI Visualizations:**

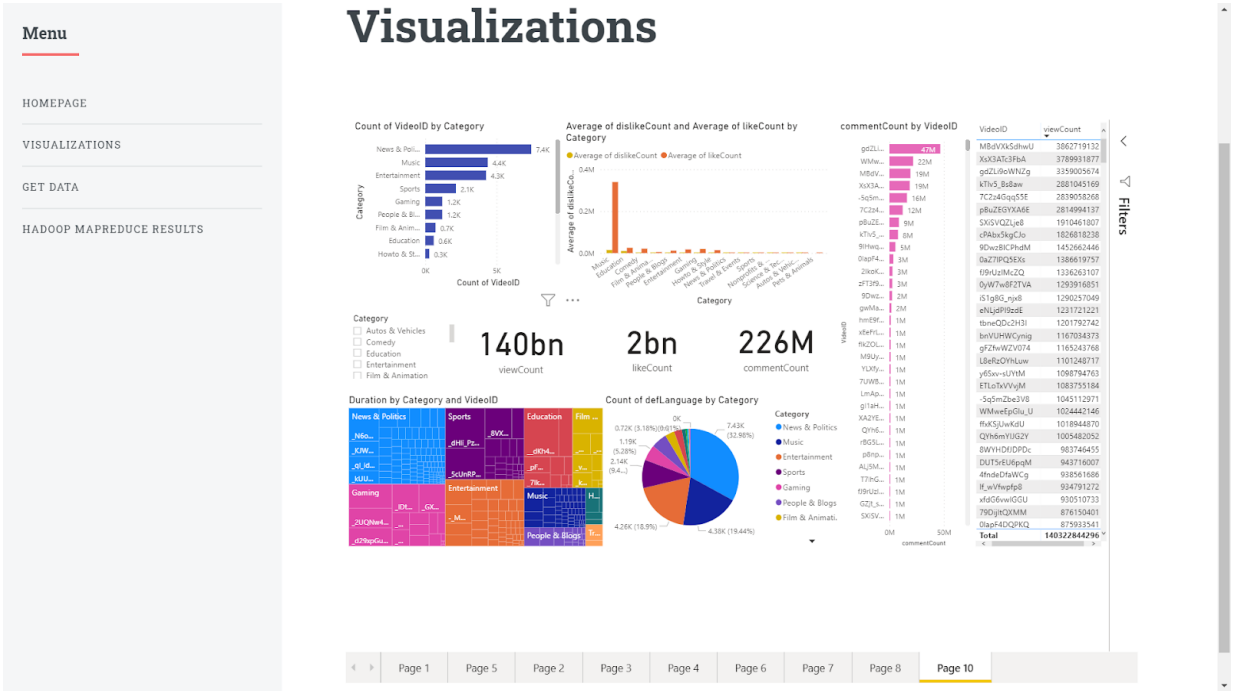



Fig.17: Visualizations embedded in a webpage

- **Hadoop Map-Reduce Outputs:**

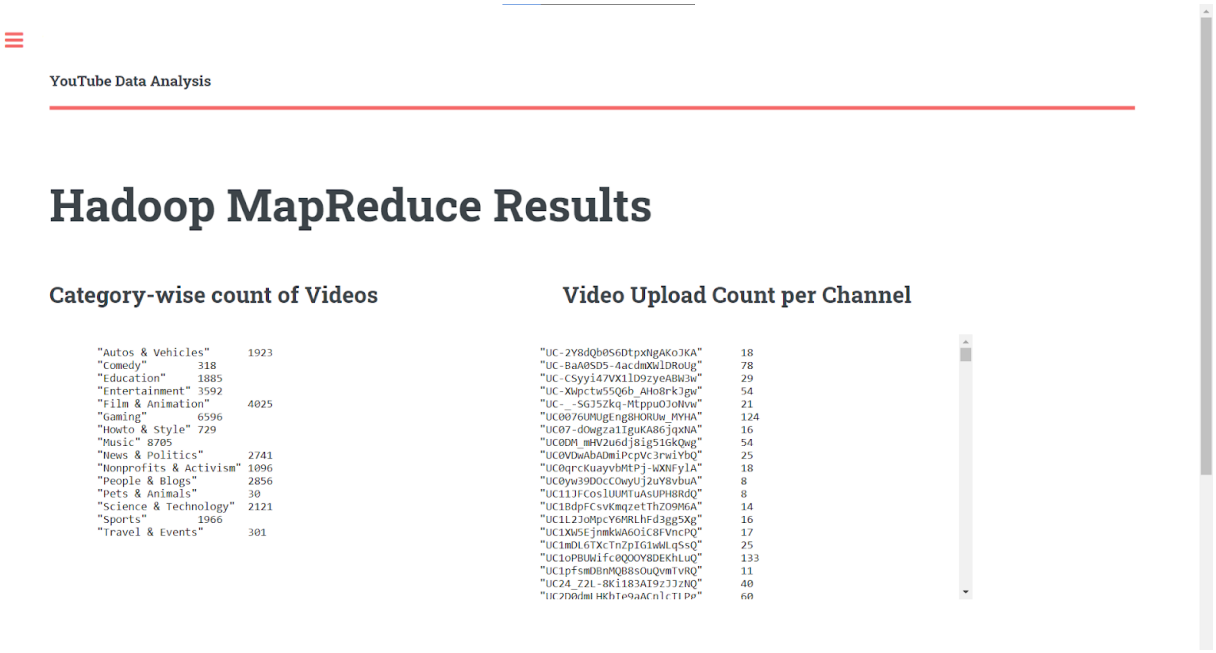



Fig.18: Hadoop MapReduce Output embedded in a webpage

# 7. VISUALIZATION SCREENSHOTS

- **Visualizations filtered by Education Category**

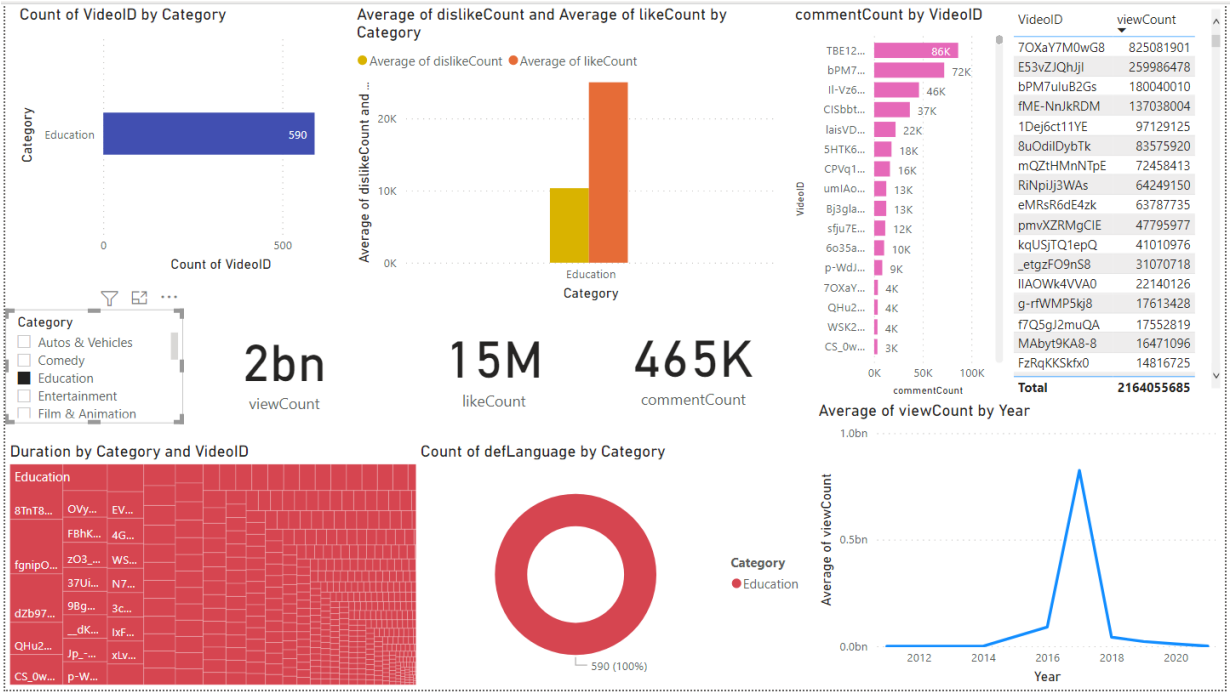



Fig.19: Visualizations filtered by education category

- **Dashboard**

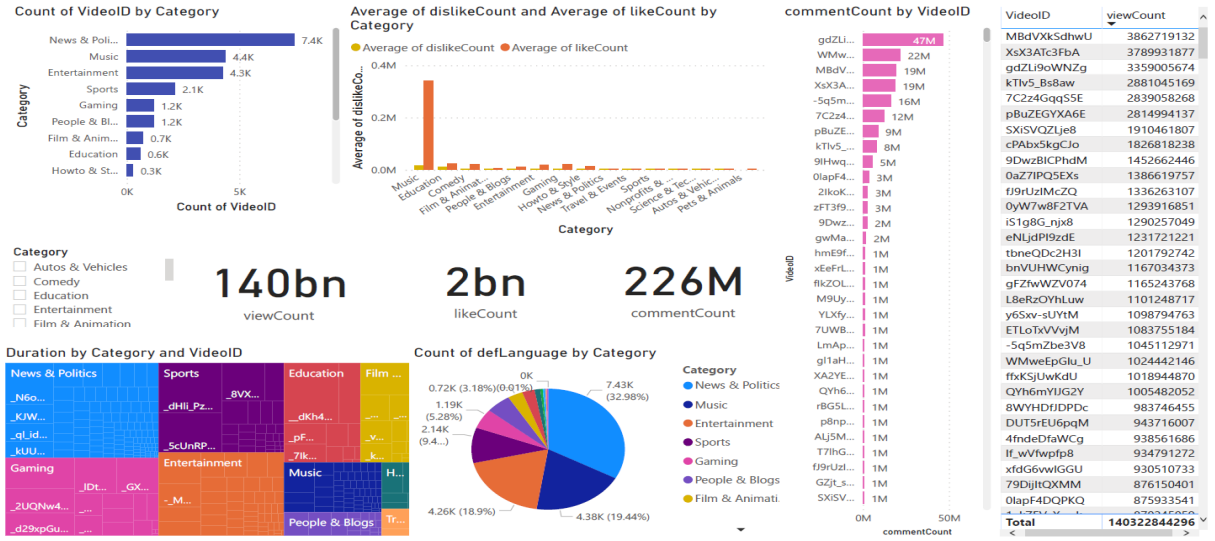



Fig.20: PoweBI Dashboard

- **Key Influencers**

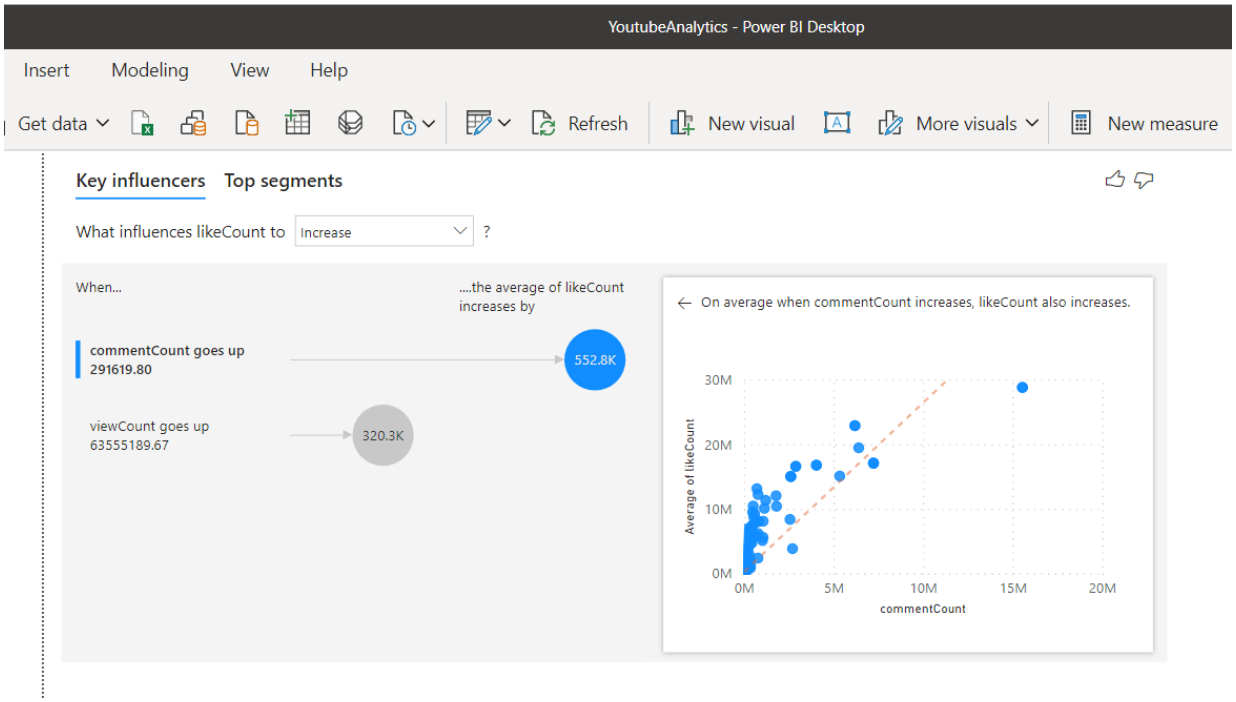



Fig.21:Key Influencers

# 8. CONCLUSION AND FUTURE WORK

## 8.1 Conclusion

In the digitalized era today, companies use YouTube for marketing and promoting their products and brand by uploading their product advertisement video to YouTube and movie makers promote their movies by uploading songs and movie trailers to YouTube. The measure of how well the product and movie is received by the public are determined by the number of views, likes (ratings) and comments on the video. This project intends to hit on those key areas which companies and organizations use or can use to measure their product's/movie's success against their competitors. As seen from the methodology, the basic algorithm retrieves reports to better understand and view statistics and trends for users' channels depending on the number of views and likes not only on their respective videos but also check if their competitors' are at the top. Another output result gives us insights on what categories of videos interest the public more. This can be done by analyzing the top video categories. This also helps budding YouTubers who upload YouTube videos to earn money. Thus, we used MapReduce based on JAVA, Cassandra CQL queries and Power BI tool to gain insights from the extracted Youtube data.

## 8.2 Future Scope

This project can be further advanced by designing a MapReduce algorithm to perform sentiment analysis on YouTube video comments. Also, an algorithm on comment analysis can help analyze and identify the numbers of trolls harassing authentic users and spam users. Designing a MapReduce algorithm for analysing how many users click on the ads or how many users enable or disable Adblock for their website can give a very interesting and important insight to the company. These algorithms can also be implemented on any social networking site promoting ads.

# 9. REFERENCES


1. F. Shaikh, D. Pawaskar, A. Siddiqui and U. Khan, "YouTube Data Analysis using MapReduce on Hadoop," 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2018, pp. 2037-2041, doi: 10.1109/RTEICT42901.2018.9012635.
2. Hota, S. (2018). Big Data Analysis on YouTube Using Hadoop And Mapreduce. International Journal of Computer Engineering in Research Trends, 5, 98-104.
3. Shelke, M.B., 2017. YDA: Youtube Data Analysis Using Hadoop and Mapreduce. Open Access International Journal of Science and Engineering, 2(11).