

Mini Project Report *on*

“YouTube Data Analysis”

Submitted by

Sumedh Kamble	S1032180007	PC01
Shreyan Yoge	S1032180058	PC05
Shriya Padhi	S1032181174	PD25
Taha Bohra	S1032181349	PC58

Under the Guidance of

Prof. Vaishali Suryawanshi

At



Dr. Vishwanath Karad

**MIT WORLD PEACE
UNIVERSITY** | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

School of Computer Engineering and Technology

Contents

Abstract	I
List of Abbreviations.....	II
List of Figures.....	III
List of Tables.....	IV
1. Introduction.....	1
2. Motivation.	12
3. Problem Definition	12
4. Problem Statement.....	12
5. Objectives	12
6. Tools used	12
6.1 Hadoop Distributed File System	12
6.1 Hive	12
6.3 Power BI	12
7. Dataset description.....	12
8. System architecture.....	12
9. Data analysis.....	12
10. Output.....	12
11. Visualization screenshots	12
12. Conclusion.....	12
13. References.....	12

Abstract

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. We live today in a digital world a tremendous amount of data is generated by each digital service we use. This vast amount of data generated is called Big Data. YouTube is one of the best examples of services that produce a massive amount of data in a brief period. We demonstrate how we can extract insightful information from YouTube dataset using Big Data Analytics. Data extraction of a significant amount of data is done using Hadoop and MapReduce to measure performance. Hadoop is a system that offers consistent memory. Storage is provided by HDFS (Hadoop Distributed File System) and MapReduce analysis. MapReduce is a programming model and a corresponding implementation for processing large data sets. Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop. Power BI is a business analytics service by Microsoft. It aims to provide interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards. It is part of the Microsoft Power Platform. These are the technologies used for the same.

List of Abbreviations

BDA Big Data Analytics

BI Business Intelligence

HDFS Hadoop Distributed File System

POSIX Portable Operating System Interface

API Application Programming Interface

SQL Structured Query Language

FINRA Financial Industry Regulatory Authority

UI User Interface

CSV Comma Separated Values

List of Figures

Figure 1: Hadoop Distributed File System Architecture.....	I
Figure 2: Hive System Architecture.....	I
Figure 3: Power BI System Architecture.....	I
Figure 4: YouTube Dataset.....	I
Figure 5: System Architecture for YouTube Dataset Analysis	I
Figure 6: Hive Query: Create Table.....	I
Figure 7: Hive Query: Display count of total videos.....	I
Figure 8: Hive Query: Display count of total videos with ratings more than 4.7 .I	
Figure 9: Hive Query: Analyze video length.....	I
Figure 10: Hive Query: Maximum views based on category.....	I
Figure 11: Hive Query: Minimum views based on category.....	I
Figure 12: Hive Query: Analyze video engagement.....	I
Figure 13: Hive Query: Top 10 video uploaders.....	I
Figure 14: Hive Query: Top 10 latest videos.....	I
Figure 15: Hive Query: Count of videos with category comedy.....	I
Figure 16: Hive Query: Count of comments on EvilSquirrelPictures channel....	I
Figure 17: Data visualization: Number of comments by category.....	I
Figure 18: Data visualization: Number of ratings by category	I
Figure 19: Data visualization: Time interval by category.....	I
Figure 20: Data visualization: Video length by category	I
Figure 21: Data visualization: Number of views by category.....	I
Figure 22: Data visualization: Average of ratings by category.....	I
Figure 23: Data visualization: Number of comments by uploader name.....	I
Figure 24: Data visualization: Average of ratings by uploader name.....	I
Figure 25: Data visualization: Time interval by uploader name.....	I
Figure 26: Data visualization: Average video length by uploader name.....	I
Figure 27: Data visualization: Number of views by uploader name.....	I
Figure 28: Data visualization: Ratings by uploader name	I

List of Tables

Table 1: Hadoop Distributed File System Pros and Cons.....	I
Table 2: Hive Pros and Cons.....	I
Table 3: Power BI Pros and Cons.....	I

1. Introduction

Entertainment has become a necessity of life for us and there is enough content today to keep us engaged for every moment of the rest of our lives. So one thing we can be sure of is that no one is going to die of boredom in the near foreseeable future. The internet and the rise of online streaming platforms have paved the way for a new golden age of Television. It may come as a surprise, but there is a good possibility that old reliable T.V. could be permanently replaced by the modern marvels of the internet.

These platforms are both proliferating into the global market as well as evolving in terms of technology. To begin looking at what big data has been doing in the entertainment industry we need to first have a look at one of the earliest examples of online mainstream entertainment- YouTube.

2. Motivation

Online Entertainment as we perceive it today was born in 2005 when YouTube came to be. The online journey of mass-produced entertainment thus started. YouTube has always been a free forum for everyone to upload whatever they deemed fit to be called entertainment. Yet, the amount of popularity it has amassed is astounding.

The key here is audience engagement and Big Data is the magic tool that facilitates this. Big Data offers valuable insights into audience's temperaments and their preferences which can further be used to strategize content creation. From marketing strategy to creating the content itself, there are a lot of aspects that are influenced deeply by Big Data.

3. Problem Definition

The YouTube platform is increasingly under competitive pressure to not only acquire customers but also understand their customers' needs to be able to optimize customer experience and develop long standing relationships. By sharing their data and allowing relaxed privacy in its use, customers expect YouTube to know them, form relevant interactions, and provide a seamless experience across all touch points.

Content is the life-blood of this organization. Our role is to recognize trends that drive strategic roadmap for innovation, new features, and services. Being able to react in real time and make the customer feel personally valued is only possible through advanced analytics.

4. Problem Statement:

Our aim is to generate insightful information to be able to anticipate customer needs, deliver relevant products, personalize and optimize customer experiences in YouTube by using Big Data Analytics.

5. Objectives:

1. To extract meaningful insights from the YouTube dataset.
2. To uncover hidden patterns present in YouTube data.
3. To extract unknown correlations between certain parameters.
4. To understand market trends and customer preferences to enhance YouTube customer base.

6. Tools used:

1. Hadoop distributed file system:

The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file system written in Java for the Hadoop framework. Some consider it to instead be a data store due to its lack of POSIX compliance, but it does provide shell commands and Java application programming interface (API) methods that are similar to other file systems. A Hadoop instance is divided into HDFS and MapReduce. HDFS is used for storing the data and MapReduce is used for processing data.

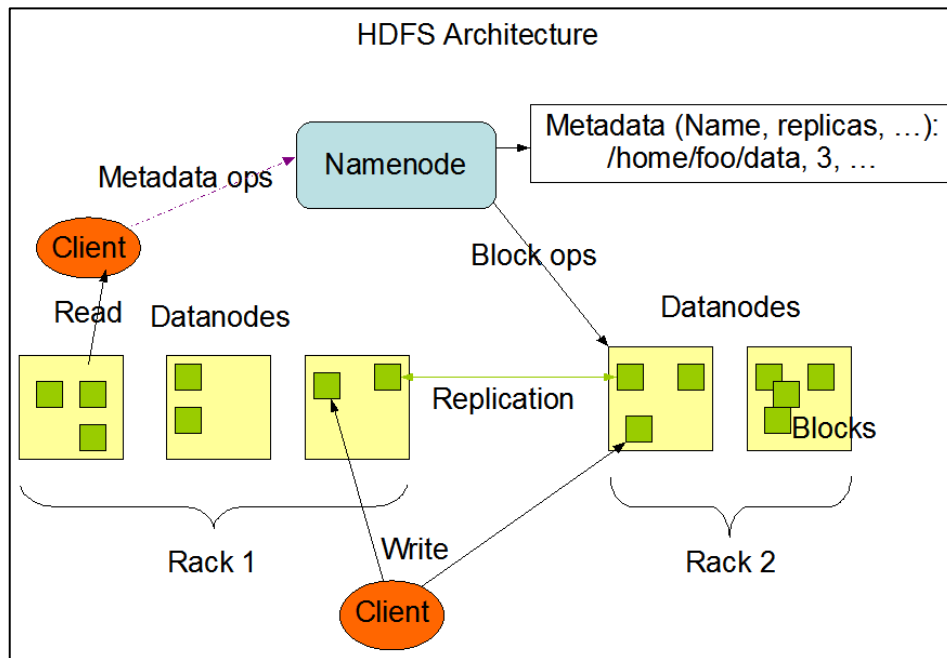


Figure 1: Hadoop Distributed File System Architecture

Pros	Cons
Scalable	Latency
Cost effective	Security
Compatible	Supports Batch Processing only
Easy to use	Latency
Varied data sources	No Real Time Data Processing

Table 1: Hadoop Distributed File System Pros and Cons

2. Hive:

Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop. Traditional SQL queries must be implemented in the MapReduce Java API to execute SQL applications and queries over distributed data. Hive provides the necessary SQL abstraction to integrate SQL-like queries (HiveQL) into the underlying Java without the need to implement queries in the low-level Java API. Since most data warehousing applications work with SQL-based querying languages, Hive aids portability of SQL-based applications to Hadoop. While initially developed by Facebook, Apache Hive is used and developed by other companies such as Netflix and the Financial Industry Regulatory Authority (FINRA). Amazon maintains a software fork of Apache Hive included in Amazon Elastic MapReduce on Amazon Web Services.

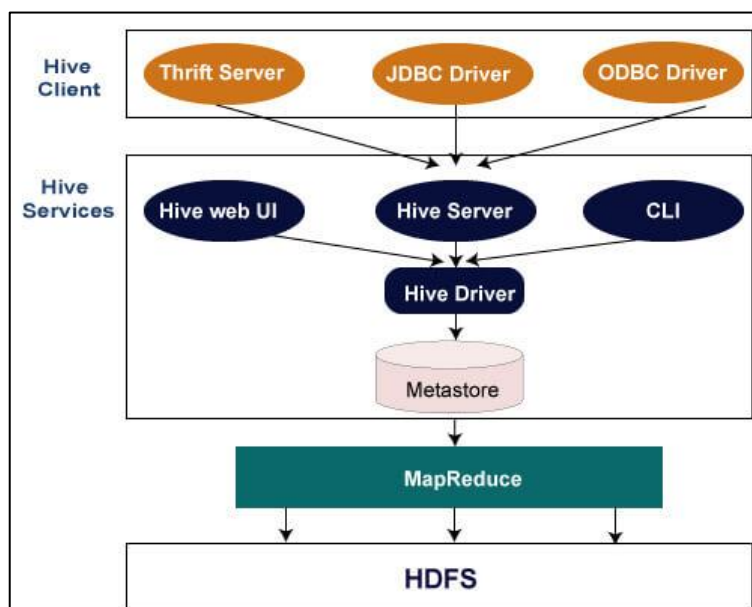


Figure 2: Hive System Architecture

Pros	Cons
Supports Hadoop	Not all standard SQL supported
MapReduce is easy	Limited built-in functions

Table 2: Hive Pros and Cons

3. Power BI:

Power BI is a business analytics service by Microsoft. It aims to provide interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards. It is part of the Microsoft Power Platform.

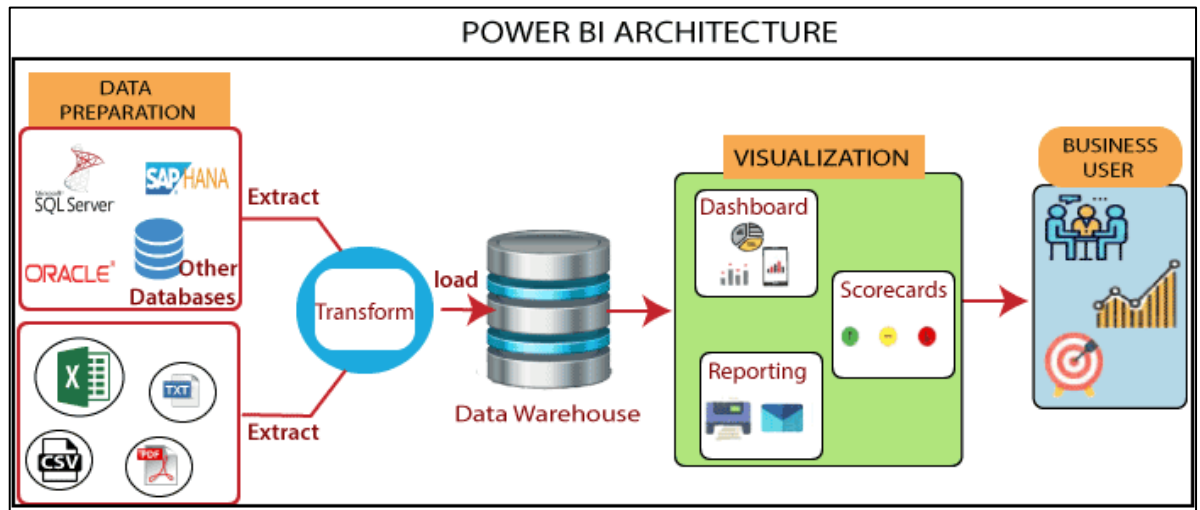


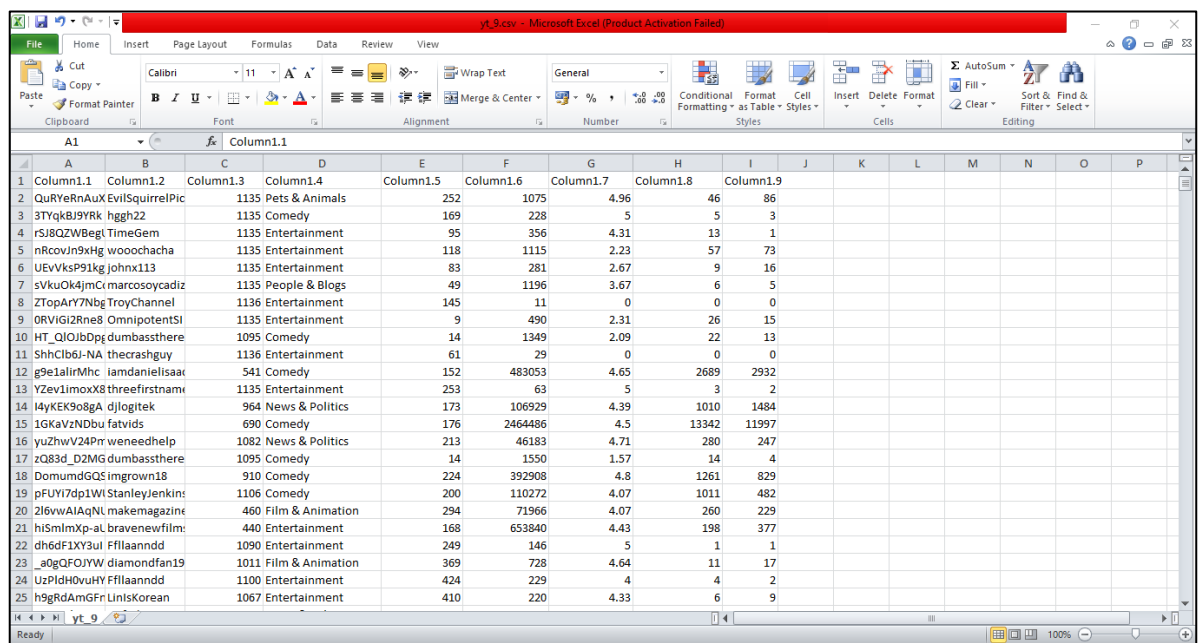
Figure 3: Power BI System Architecture

Pros	Cons
Range of custom visualizations	Difficult to master
Cost effective	Crowded UI
High Data connectivity	Rigid Formulas
Connected with Excel	Limited Data Handling in free version

Table 3: Power BI Pros and Cons

7. Dataset description:

The dataset is named as yt9.csv and is a YouTube dataset. It consists of 9 columns and 4100 rows. The attributes are video id, channel name, time interval when video was uploaded, video category, video length, number of views, ratings, number of ratings and number of comments. These attributes contribute to carry out analytics so as to obtain insightful information.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	Column1.1	Column1.2	Column1.3	Column1.4	Column1.5	Column1.6	Column1.7	Column1.8	Column1.9							
1	Column1.1	Column1.2	Column1.3	Column1.4	Column1.5	Column1.6	Column1.7	Column1.8	Column1.9							
2	QuRYeRnAuXEvilSquirrelPic		1135	Pets & Animals	252	1075	4.96	46	86							
3	3TYqKBJ9YRk hggh22		1135	Comedy	169	228	5	5	3							
4	rSj8QZWBegL TimeGem		1135	Entertainment	95	356	4.31	13	1							
5	nRcovJn9xHg woochacha		1135	Entertainment	118	1115	2.23	57	73							
6	UEvKsP91kg johnx113		1135	Entertainment	83	281	2.67	9	16							
7	sVkuOk4jmCr marcosoycadiz		1135	People & Blogs	49	1196	3.67	6	5							
8	ZTopARy7Nbg TroyChannel		1136	Entertainment	145	11	0	0	0							
9	ORVIGi2Rne8 OmnipotentSi		1135	Entertainment	9	490	2.31	26	15							
10	HT_QlOJbDpjdumbassthere		1095	Comedy	14	1349	2.09	22	13							
11	ShhClb6J-NA thecrashguy		1136	Entertainment	61	29	0	0	0							
12	g9e1alirMhc iamdaniellisaai		541	Comedy	152	483053	4.65	2689	2932							
13	Yzev1ImoxX8threefirstnam		1135	Entertainment	253	63	5	3	2							
14	l4yKEK9o8gA dJlogitek		964	News & Politics	173	106929	4.39	1010	1484							
15	1GKaVzNDbu fatvids		690	Comedy	176	2464486	4.5	13342	11997							
16	yuZhwV24Prr weneedhelp		1082	News & Politics	213	46183	4.71	280	247							
17	zQ83d_D2MG dumbassthere		1095	Comedy	14	1550	1.57	14	4							
18	DomumdGQ5imgrown18		910	Comedy	224	392908	4.8	1261	829							
19	pFUYi7dp1WlStanleyJenkins		1106	Comedy	200	110272	4.07	1011	482							
20	2l6vwAIAqNL makemagazine		460	Film & Animation	294	71966	4.07	260	229							
21	hi5mlmXp-aL bravenewfilm		440	Entertainment	168	653840	4.43	198	377							
22	dh6dF1XY3ul Ffllaanndd		1090	Entertainment	249	146	5	1	1							
23	_a0gQFOJYW diamondfan19		1011	Film & Animation	369	728	4.64	11	17							
24	UzPlidH0vuhV Ffllaanndd		1100	Entertainment	424	229	4	4	2							
25	h9gRdAmGFr LinsKorean		1067	Entertainment	410	220	4.33	6	9							

Figure 4: YouTube Dataset

8. System Architecture:

The CSV file containing the YouTube dataset is uploaded on the Hadoop Distributed File System by running commands on Cloudera Command Prompt. Hive is used to run queries on the data to analyze it. For the same it uses Map Reduce technology. Lastly, the CSV file is used to generate data visualization using Power BI.

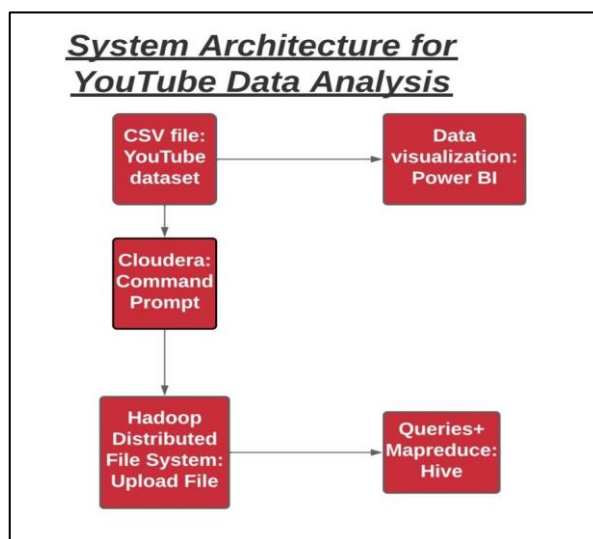


Figure 5: System Architecture for YouTube Dataset Analysis

9. Data analysis (Hive queries)

1. Create table.

```
create table ytdata(vid_id string, uploader_name int,  
interval_fromytcreated_toupload int, catagory string, video_length  
int, no_of_views int, ratings float, no_of_ratings int, no_of_comments  
int) row format delimited fields terminated by ',' stored as textfile  
location '/user/cloudera/ytdata';
```

2. Display the count of total videos in database.

```
select count(*) from ytdata;
```

3. Display the count of videos with ratings greater than 4.7.

```
select count(*) from ytdata where ratings > 4.7;
```

4. Analyse category wise popularity of videos based on ratings.

```
select category,avg(ratings) as average from ytdata group by category order by  
average;
```

5. Check if longer videos are more popular than shorter ones.

```
select "longer_videos_rating",avg(ratings) as longer_videos_avg_ratings from  
ytdata where video_length > 200;  
select "Shorter Videos Ratings",avg(ratings) as longer_videos_avg_ratings from  
ytdata where video_length < 200;
```

6. Display category wise maximum views count.

```
select category ,max(no_of_views) as max_views from ytdata group by  
category;
```

7. Display category wise minimum views count.

```
select category ,min(no_of_views) as min_views from ytdata group by  
category;
```

8. Display top 3 engaging videos. It is measured as-

number of ratings + number of comments /number of views

```
select vid_id,( (no_of_ratings+no_of_comments)/no_of_views) as  
engagement_value from ytdata order by engagement_value desc limit 3 ;
```

9. Display top 10 video uploaders.

```
select uploader_name,count(*)as count from ytdata group by uploader_name  
order by count desc limit 10;
```

10. Display 10 oldest videos uploaded

```
select vid_id,interval_frommytcreated_toupload from ytdata where  
interval_frommytcreated_toupload > 0 order by interval_frommytcreated_toupload  
limit 10;
```

11. Display 10 latest videos uploaded

```
select vid_id,interval_frommytcreated_toupload from ytdata where  
interval_frommytcreated_toupload > 0 order by interval_frommytcreated_toupload  
desc limit 10;
```

12. Display number of videos in comedy category.

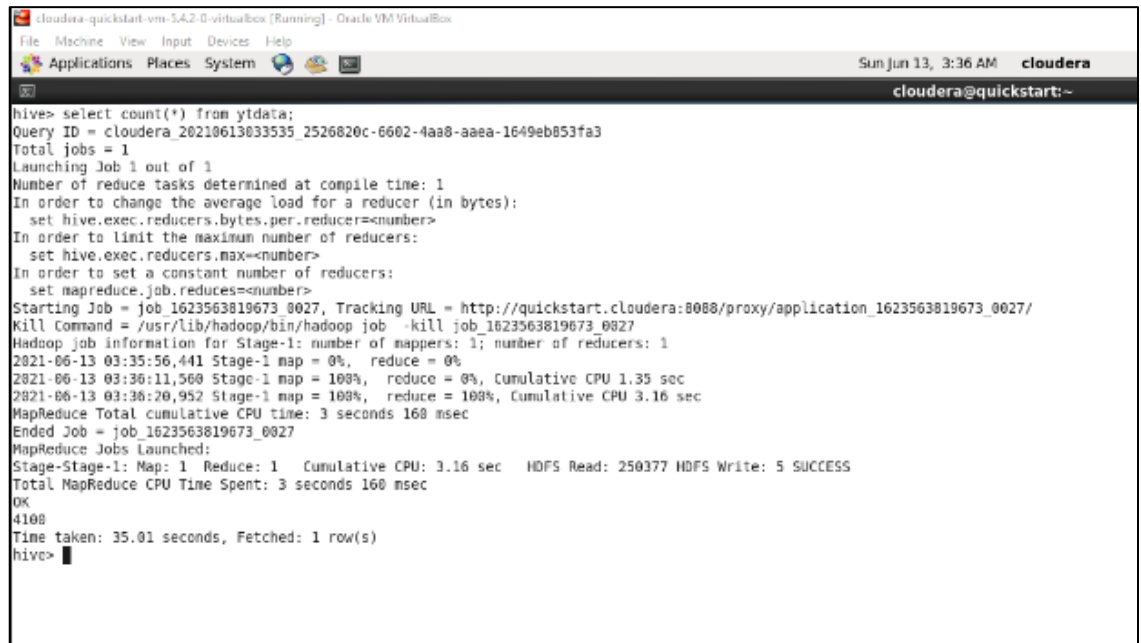
```
select count(*) from ytdata where category='Comedy';
```

13. Display total comments on EvilSquirrelPictures channel.

```
select sum(no_of_comments) from ytdata where uploader_name =  
'EvilSquirrelPictures';
```

10. Output

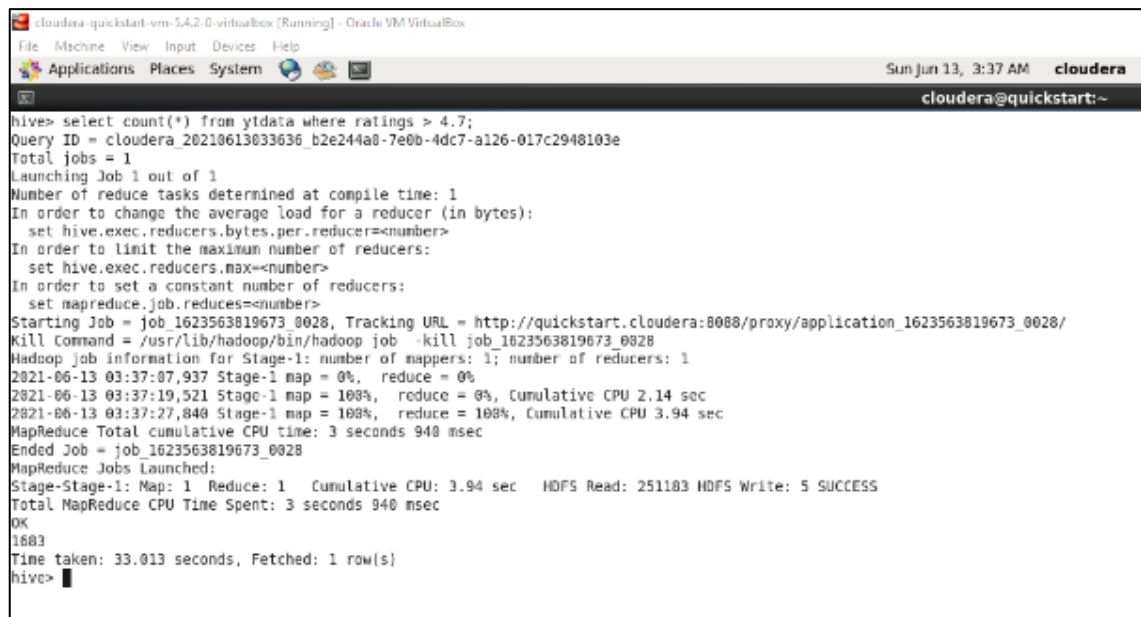
1. Create table.



```
cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Sun Jun 13, 3:36 AM cloudera
cloudera@quickstart:~
hive> select count(*) from yldata;
Query ID = cloudera_20210613033535_2526820c-6602-4aa8-aaaa-1649eb853fa3
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0027, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0027/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0027
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:35:56,441 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:36:11,560 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.35 sec
2021-06-13 03:36:20,952 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.16 sec
MapReduce Total cumulative CPU time: 3 seconds 160 msec
Ended Job = job_1623563819673_0027
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.16 sec HDFS Read: 250377 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 160 msec
OK
4108
Time taken: 35.01 seconds, Fetched: 1 row(s)
hive>
```

Figure 6: Hive Query: Create Table

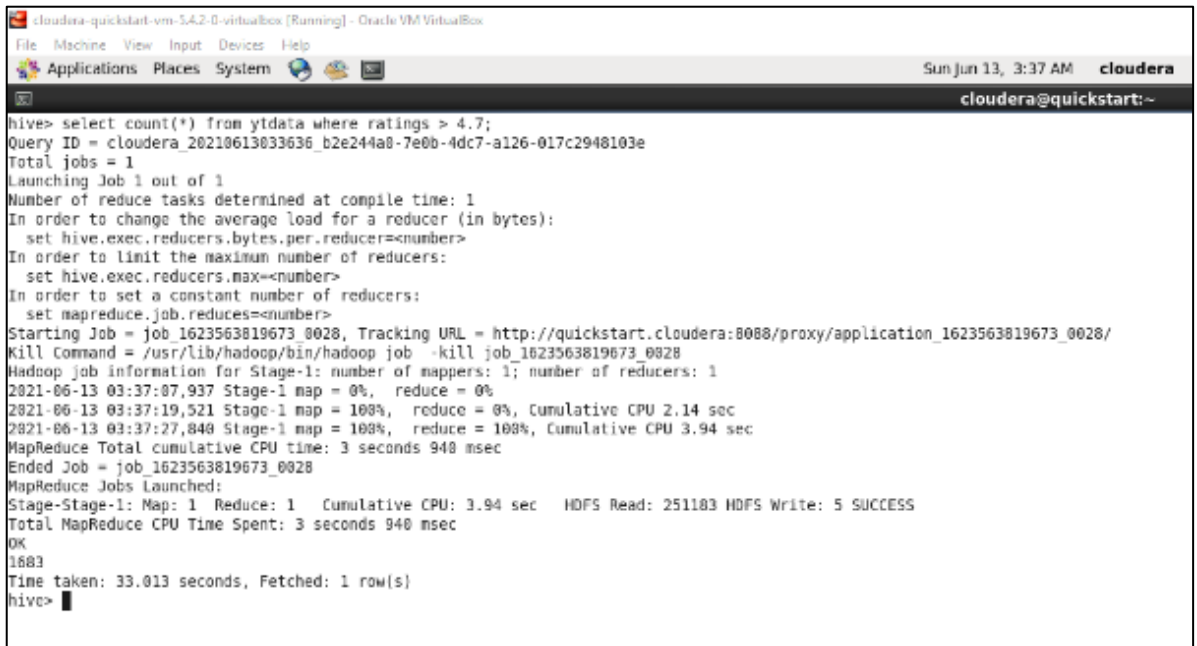
2. Display the count of total videos in database.



```
cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Sun Jun 13, 3:37 AM cloudera
cloudera@quickstart:~
hive> select count(*) from yldata where ratings > 4.7;
Query ID = cloudera_20210613033636_b2e244a0-7e0b-4dc7-a126-017c2948103e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0028, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0028/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0028
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:37:07,937 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:37:19,521 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.14 sec
2021-06-13 03:37:27,840 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.94 sec
MapReduce Total cumulative CPU time: 3 seconds 940 msec
Ended Job = job_1623563819673_0028
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.94 sec HDFS Read: 251103 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 940 msec
OK
1883
Time taken: 33.013 seconds, Fetched: 1 row(s)
hive>
```

Figure 7: Hive Query: Display count of total videos

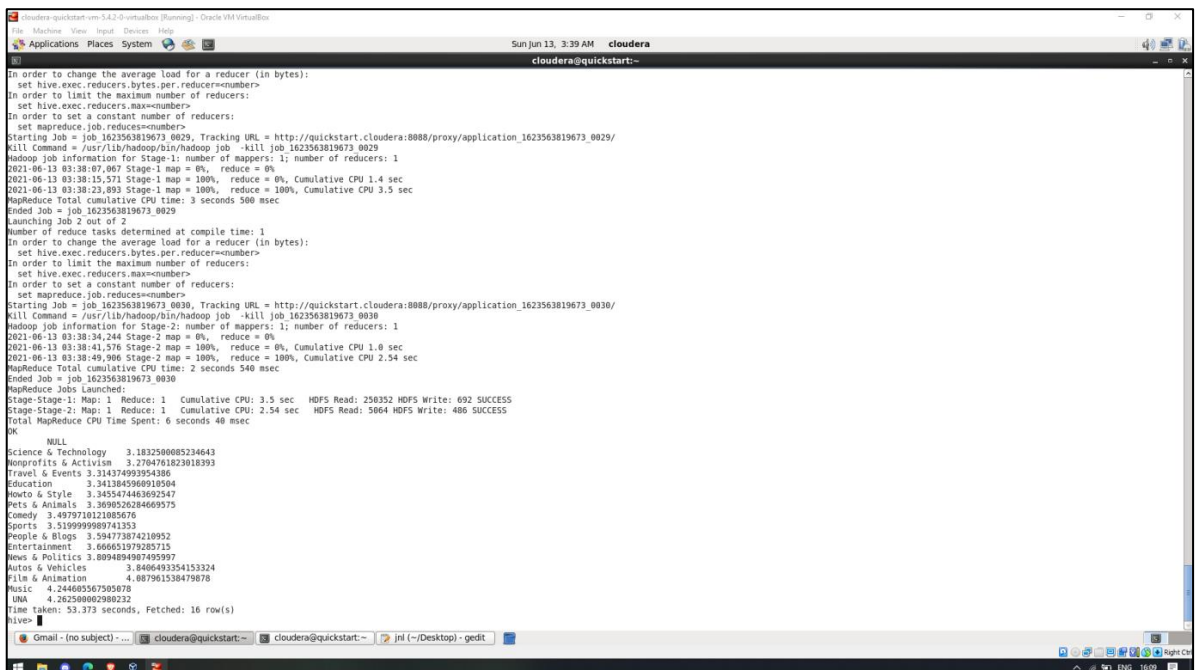
3. Display the count of videos with ratings greater than 4.7.



```
cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Sun Jun 13, 3:37 AM cloudera
cloudera@quickstart:~
hive> select count(*) from yldata where ratings > 4.7;
Query ID = cloudera_20210613033636_b2e244a0-7e0b-4dc7-a126-017c2948103e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0028, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0028/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0028
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:37:07.937 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:37:19.521 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.14 sec
2021-06-13 03:37:27.840 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.94 sec
MapReduce Total cumulative CPU time: 3 seconds 940 msec
Ended Job = job_1623563819673_0028
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.94 sec HDFS Read: 251183 HDFS Write: 5 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 940 msec
OK
1683
Time taken: 33.013 seconds, Fetched: 1 row(s)
hive>
```

Figure 8: Hive Query: Display count of total videos with ratings more than 4.7

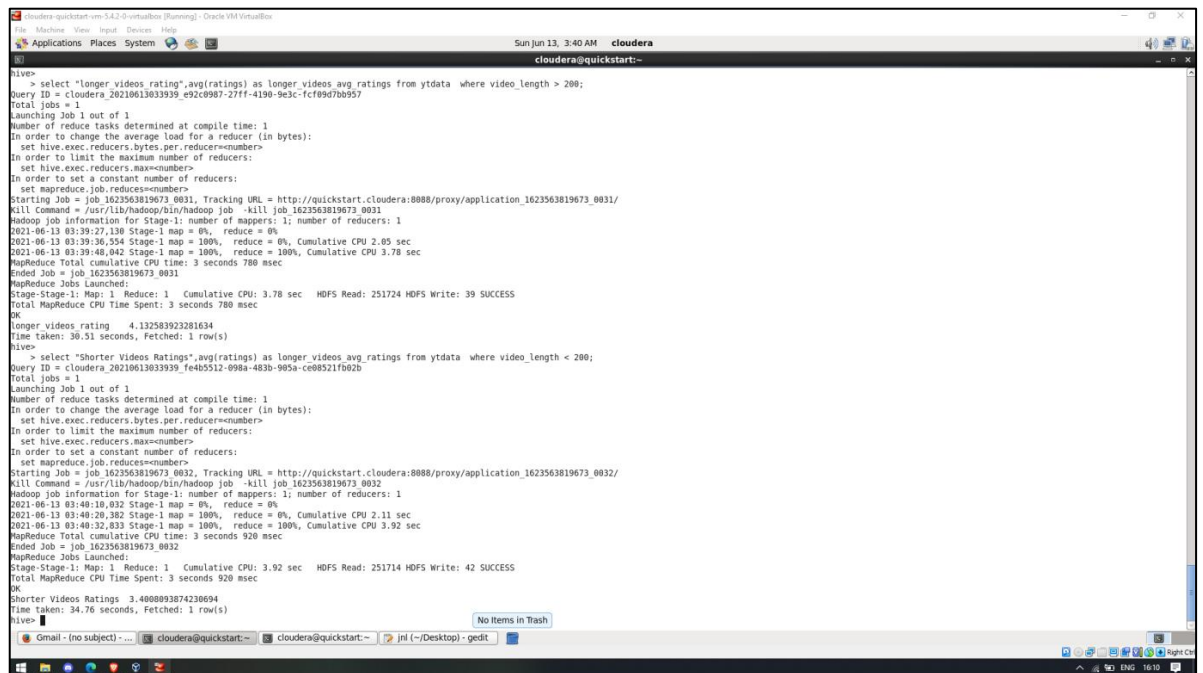
4. Analyze category wise popularity of videos based on ratings.



```
cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Sun Jun 13, 3:39 AM cloudera
cloudera@quickstart:~
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0029, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0029/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0029
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:38:07.067 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:38:15.571 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.4 sec
2021-06-13 03:38:23.893 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.5 sec
MapReduce Total cumulative CPU time: 3 seconds 500 msec
Ended Job = job_1623563819673_0029
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0030, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0030/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0030
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2021-06-13 03:38:34.244 Stage-2 map = 0%, reduce = 0%
2021-06-13 03:38:41.576 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.0 sec
2021-06-13 03:38:49.906 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 2.54 sec
MapReduce Total cumulative CPU time: 2 seconds 540 msec
Ended Job = job_1623563819673_0030
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.5 sec HDFS Read: 250352 HDFS Write: 692 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 2.54 sec HDFS Read: 5664 HDFS Write: 480 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 40 msec
OK
NULL
Science & Technology 3.1832500085234643
Nonprofits & Activism 3.2704761823918393
Travel & Events 3.314374993954306
Education 3.3413845960910504
Howto & Style 3.3455474463692547
Pets & Animals 3.3899526284669575
Comedy 3.4979710121085676
Sports 3.5199999089741353
People & Blogs 3.594773874210952
Entertainment 3.606651979205715
News & Politics 3.809494987495997
Autos & Vehicles 3.8406493354153324
Film & Animation 4.087961538479878
Music 4.244603567505078
UNA 4.262500062980232
Time taken: 53.373 seconds, Fetched: 16 row(s)
hive>
```

Figure 8: Hive Query: Analyze category wise video popularity

5. Check if longer videos are more popular than shorter ones.



```
hive> > select 'longer videos rating',avg(ratings) as longer_videos_avg_ratings from ytdata where video_length > 200;
Query ID = cloudera_20210613033939_e92c0987-27ff-4190-9e3c-fcf09d7b0957
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0031, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0031/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0031
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:39:27.130 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:39:36.554 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.05 sec
2021-06-13 03:39:40.042 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.78 sec
MapReduce Total cumulative CPU time: 3 seconds 780 msec
Ended Job = job_1623563819673_0031
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.78 sec HDFS Read: 251724 HDFS Write: 39 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 780 msec
OK
longer_videos_rating 4.132583923281634
Time taken: 30.51 seconds, Fetched: 1 row(s)
hive> > select 'Shorter Videos Ratings',avg(ratings) as longer_videos_avg_ratings from ytdata where video_length < 200;
Query ID = cloudera_20210613033939_f4b5512-098a-483b-905a-ce08521f0b2b
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0032, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0032/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0032
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:40:18.032 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:40:20.382 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.11 sec
2021-06-13 03:40:32.033 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.92 sec
MapReduce Total cumulative CPU time: 3 seconds 920 msec
Ended Job = job_1623563819673_0032
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.92 sec HDFS Read: 251714 HDFS Write: 42 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 920 msec
OK
Shorter Videos Ratings 3.4008093874230694
Time taken: 34.76 seconds, Fetched: 1 row(s)
hive>
```

Figure 9: Hive Query: Analyze video length

6. Display category wise maximum views count.



```
hive> select category,max(no of views) as max_views from ytdata group by category;
Query ID = cloudera_20210613034242_2e00c8ba-b039-4e20-99c3-1f140eea8b6e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0034, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0034/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0034
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:42:46.122 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:43:01.996 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.51 sec
2021-06-13 03:43:12.464 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.53 sec
MapReduce Total cumulative CPU time: 3 seconds 530 msec
Ended Job = job_1623563819673_0034
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.53 sec HDFS Read: 250690 HDFS Write: 329 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 530 msec
OK
NULL
UNRA 711384
Autos & Vehicles 1762369
Comedy 11807201
Education 1369522
Entertainment 18235463
Film & Animation 65341925
Howto & Style 6101232
Music 33754615
News & Politics 6024441
Nonprofits & Activism 80967
People & Blogs 5766247
Pets & Animals 27721690
Science & Technology 3234852
Sports 5360384
Travel & Events 3818296
Time taken: 47.407 seconds, Fetched: 16 row(s)
hive>
```

Figure 10: Hive Query: Maximum views based on category

7. Display category wise minimum views count.

```
cloudera@quickstart:~$ hive> select category ,min(no of views) as min views from ytdata group by category;
Query ID = cloudera_20210613034343_7c5a6691-7fd3-41b6-b052-9f19b42c1008
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0035, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0035/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0035
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:43:51,729 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:43:59,165 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.34 sec
2021-06-13 03:44:06,485 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.05 sec
MapReduce Total cumulative CPU time: 3 seconds 50 msec
Ended Job = job_1623563819673_0035
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.05 sec HDFS Read: 250690 HDFS Write: 247 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 50 msec
OK
      NULL
UNA    159
Autos & Vehicles      41
Comedy      11
Education      21
Entertainment      6
Film & Animation      4
Howto & Style      30
Music      0
News & Politics      25
Nonprofits & Activism      17
People & Blogs      3
Pets & Animals      18
Science & Technology      19
Sports      0
Travel & Events      4
Time taken: 26.708 seconds, Fetched: 16 row(s)
hive>
```

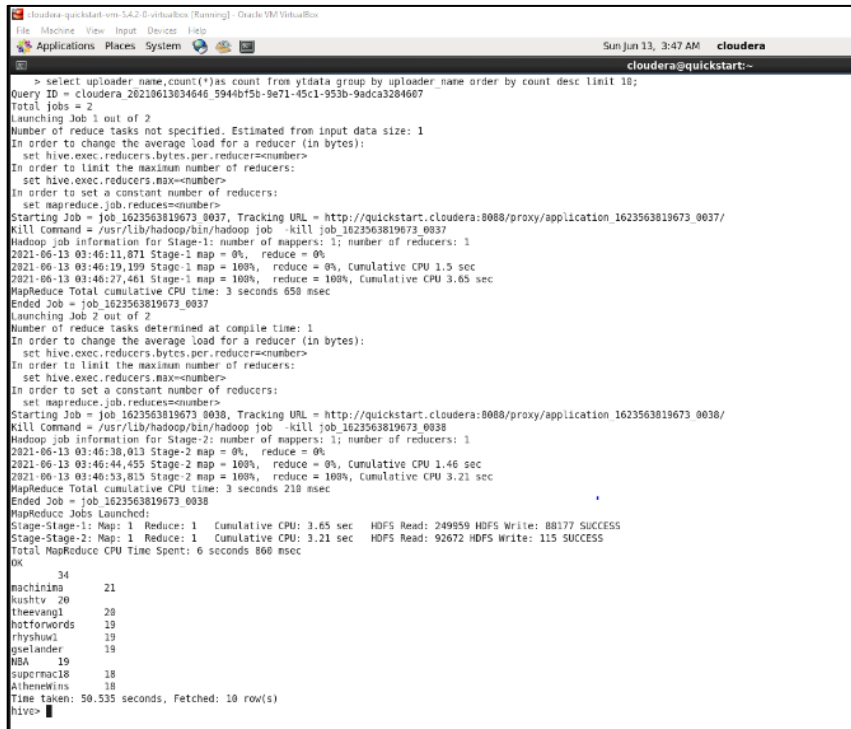
Figure 11: Hive Query: Minimum views based on category

8. Display top 3 engaging videos. It is measured as- number of ratings + number of comments /number of views

```
cloudera@quickstart:~$ hive> select vid id,{ (no of ratings+no of comments)/no of views} as engagement_value from ytdata order by engagement_value desc limit 3 ;
Query ID = cloudera_20210613034545_2ebe02b5-41a1-4da4-aa14-8293478fa02d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0036, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0036/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0036
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:45:12,252 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:45:24,309 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.69 sec
2021-06-13 03:45:37,962 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.53 sec
MapReduce Total cumulative CPU time: 4 seconds 530 msec
Ended Job = job_1623563819673_0036
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.53 sec HDFS Read: 250949 HDFS Write: 92 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 530 msec
OK
gAtFNr3G6uW      1.583941605839416
4Du5CM0uPhQ      1.2087175188600168
vpkT8bUI204      0.8623918878616165
Time taken: 36.084 seconds, Fetched: 3 row(s)
hive>
>
```

Figure 12: Hive Query: Analyze video engagement

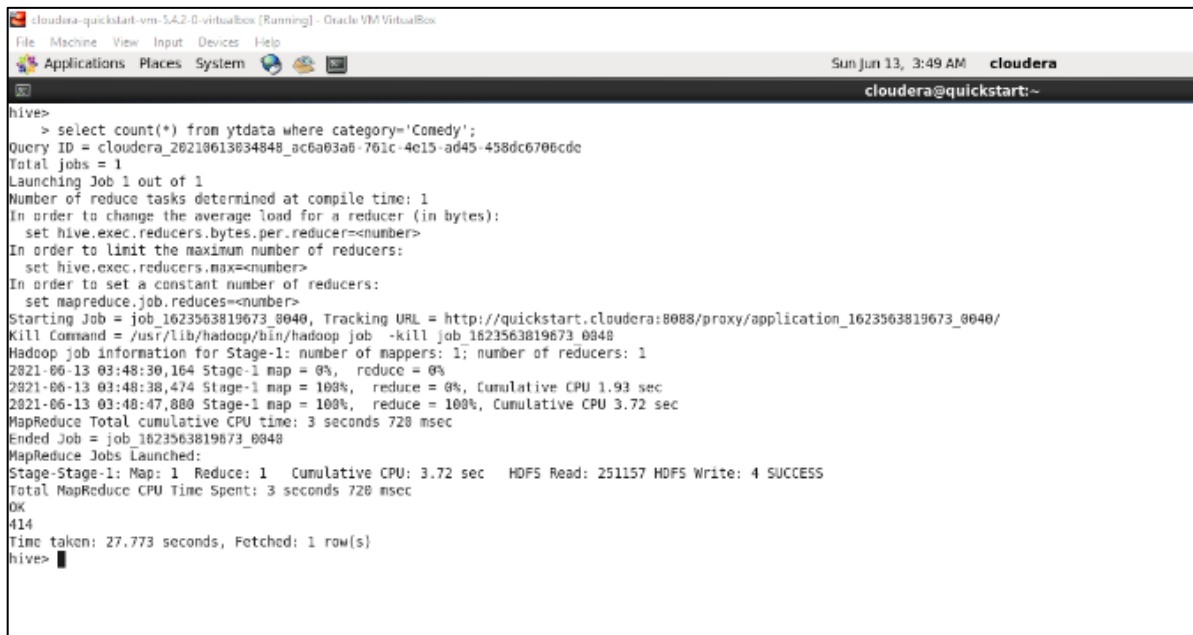
9. Display top 10 video uploaders.



```
cloudera-quickstart-vm-5.4.2.0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System Sun Jun 13, 3:47 AM cloudera
cloudera@quickstart:~$
> select uploader name,count(*)as count from ytdata group by uploader name order by count desc limit 10;
Query ID = cloudera_20210613034646_5944bf5b-9e71-45c1-953b-9adca3284607
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0037, Tracking URL = http://quickstart.cloudera:8080/proxy/application_1623563819673_0037/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0037
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:46:11,871 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:46:19,199 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.5 sec
2021-06-13 03:46:27,461 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.65 sec
MapReduce Total cumulative CPU time: 3 seconds 650 msec
Ended Job = job_1623563819673_0037
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0038, Tracking URL = http://quickstart.cloudera:8080/proxy/application_1623563819673_0038/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0038
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2021-06-13 03:46:38,013 Stage-2 map = 0%, reduce = 0%
2021-06-13 03:46:44,455 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.46 sec
2021-06-13 03:46:53,813 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 3.21 sec
MapReduce Total cumulative CPU time: 3 seconds 218 msec
Ended Job = job_1623563819673_0038
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.65 sec HDFS Read: 249859 HDFS Write: 88177 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 3.21 sec HDFS Read: 92672 HDFS Write: 115 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 860 msec
OK
34
machining      21
kushTV         20
theevangl      20
hotforwords    19
rhythmic       19
gselander      19
NBA            19
supermac18     18
AlheneXins     18
Time taken: 50.535 seconds, Fetched: 10 row(s)
hive>
```

Figure 13: Hive Query: Top 10 video uploaders

10. Display 10 latest videos uploaded



```
cloudera-quickstart-vm-5.4.2.0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System Sun Jun 13, 3:49 AM cloudera
cloudera@quickstart:~$
hive>
> select count(*) from ytdata where category='Comedy';
Query ID = cloudera_20210613034848_ac6a03a6-761c-4e15-ad45-458dc6706cdc
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0040, Tracking URL = http://quickstart.cloudera:8080/proxy/application_1623563819673_0040/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0040
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:48:30,164 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:48:38,474 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.93 sec
2021-06-13 03:48:47,800 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.72 sec
MapReduce Total cumulative CPU time: 3 seconds 720 msec
Ended Job = job_1623563819673_0040
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.72 sec HDFS Read: 251157 HDFS Write: 4 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 720 msec
OK
414
Time taken: 27.773 seconds, Fetched: 1 row(s)
hive>
```

Figure 14: Hive Query: Top 10 latest videos

11. Display number of videos in comedy category.

```
cloudera@quickstart:~$ hive> select sum(no_of_comments) from ytdata where uploader name = 'EvilSquirrelPictures';
Query ID = cloudera_20210613035050_e8deb02c-7f56-43f7-8d35-3fe17a36d52d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0042, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0042/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0042
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:50:20,767 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:50:34,850 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.99 sec
2021-06-13 03:50:46,692 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.72 sec
MapReduce Total cumulative CPU time: 4 seconds 720 msec
Ended Job = job_1623563819673_0042
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.72 sec HDFS Read: 251204 HDFS Write: 3 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 720 msec
OK
06
Time taken: 39.604 seconds, Fetched: 1 row(s)
hive>
```

Figure 15: Hive Query: Count of videos with category comedy

12. Display total comments on EvilSquirrelPictures channel.

```
cloudera@~$ hive> select sum(no_of_comments) from ytdata where uploader name = 'EvilSquirrelPictures';
Query ID = cloudera_20210613035050_e8deb02c-7f56-43f7-8d35-3fe17a36d52d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1623563819673_0042, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1623563819673_0042/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1623563819673_0042
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2021-06-13 03:50:20,767 Stage-1 map = 0%, reduce = 0%
2021-06-13 03:50:34,850 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.99 sec
2021-06-13 03:50:46,692 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.72 sec
MapReduce Total cumulative CPU time: 4 seconds 720 msec
Ended Job = job_1623563819673_0042
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.72 sec HDFS Read: 251204 HDFS Write: 3 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 720 msec
OK
06
Time taken: 39.604 seconds, Fetched: 1 row(s)
hive>
```

Figure 16: Hive Query: Count of comments on EvilSquirrelPictures channel

12. Visualization Screenshots

1. Number of comments grouped by category.

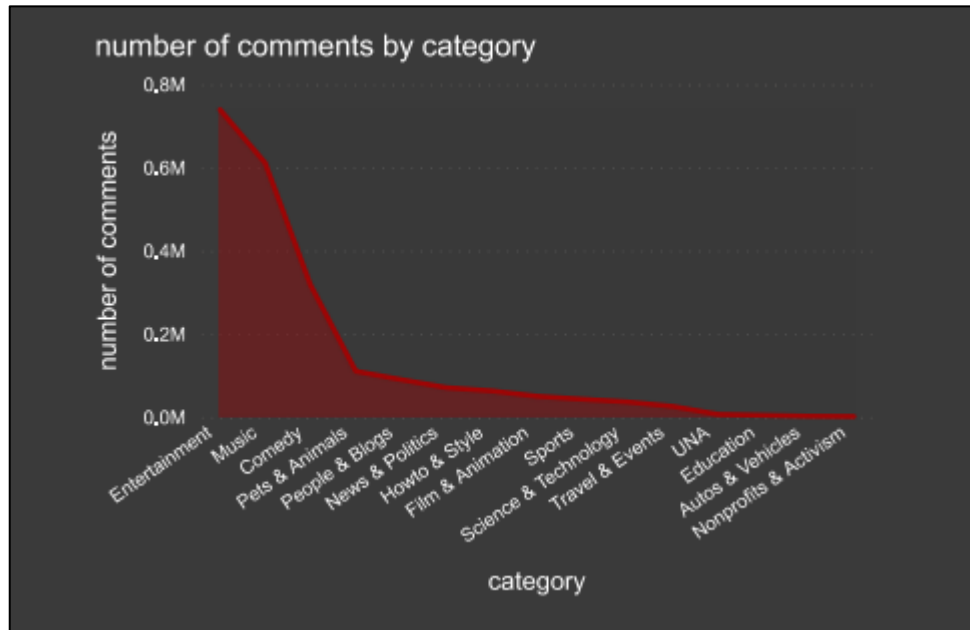


Figure 17: Data visualization: Number of comments by category

2. Number of ratings grouped by category.

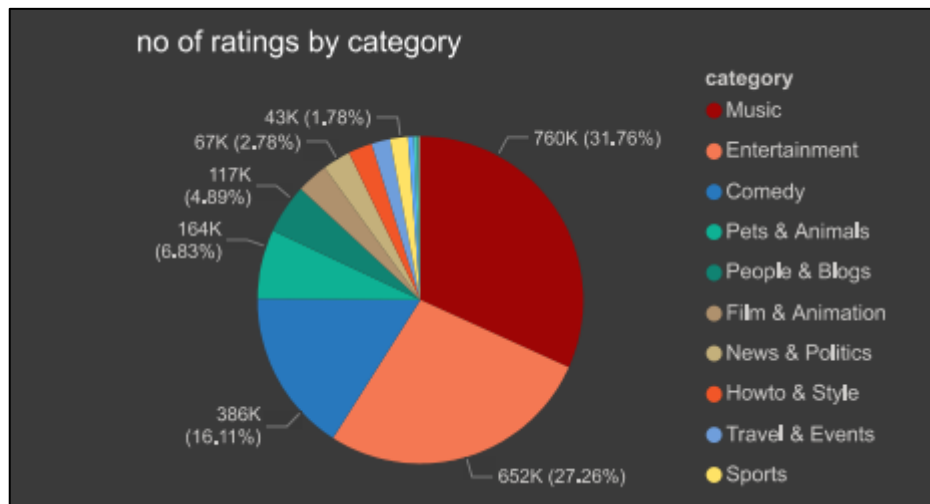


Figure 18: Data visualization: Number of ratings by category

3. Time interval from when YouTube was created to when the video was uploaded grouped by category.

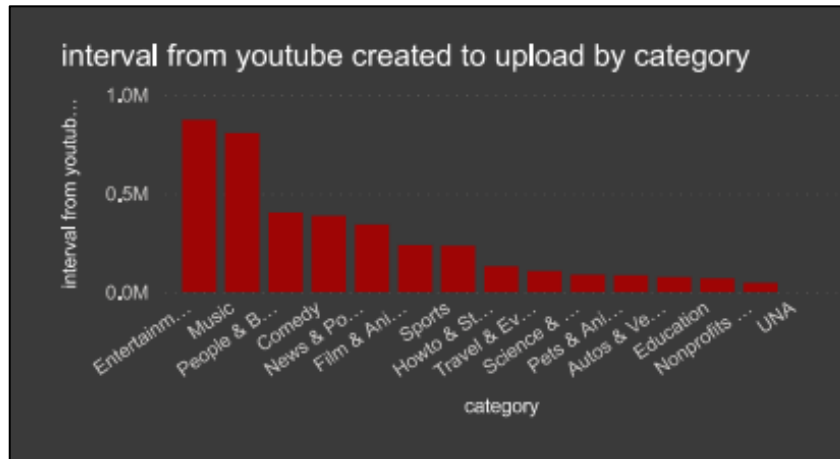


Figure 19: Data visualization: Time interval by category

4. Video length grouped by category.

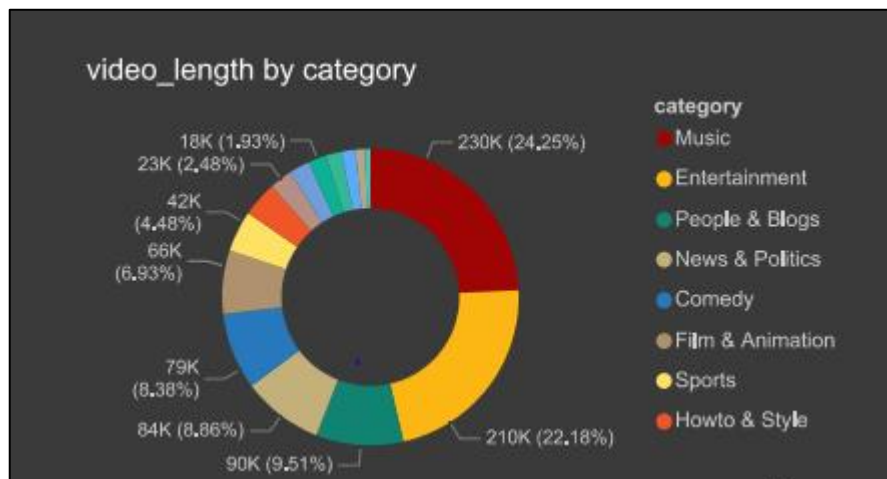


Figure 20: Data visualization: Video length by category

5. Number of views grouped by category.

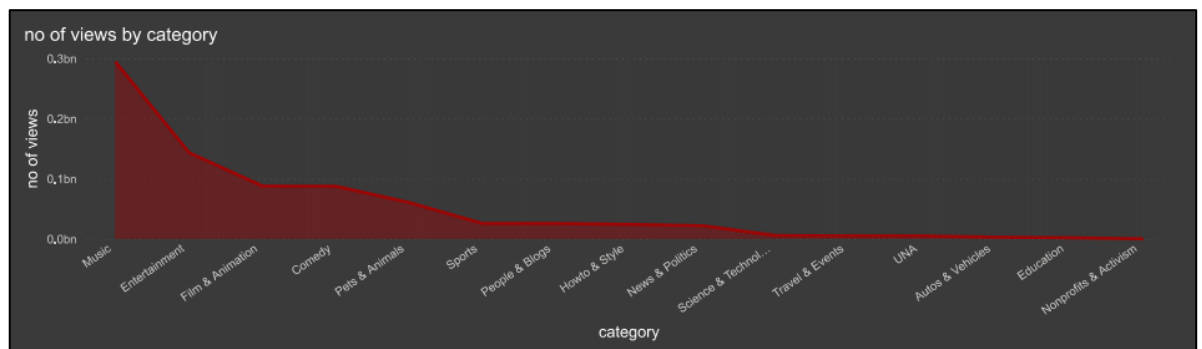


Figure 21: Data visualization: Number of views by category

6. Average of ratings grouped by category.

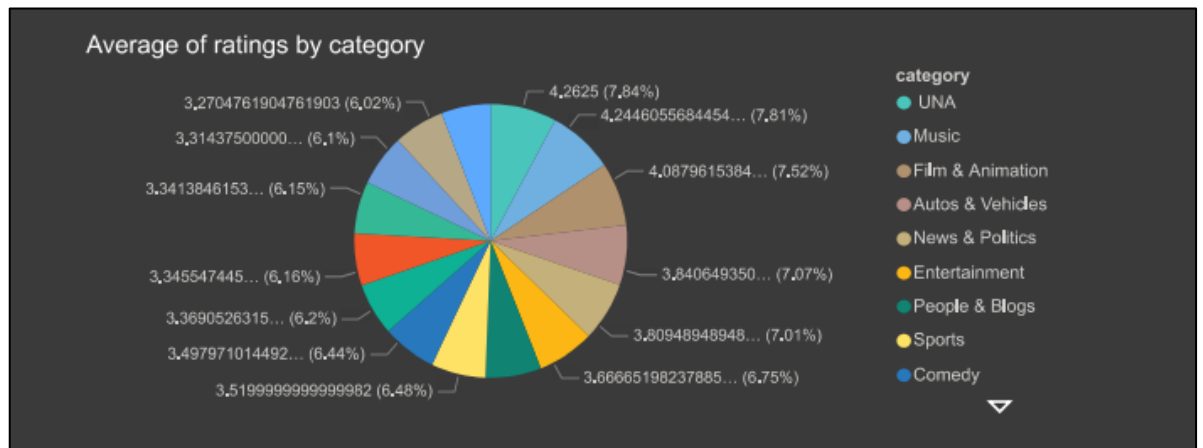


Figure 22: Data visualization: Average of ratings by category

7. Number of comments grouped by uploader name.

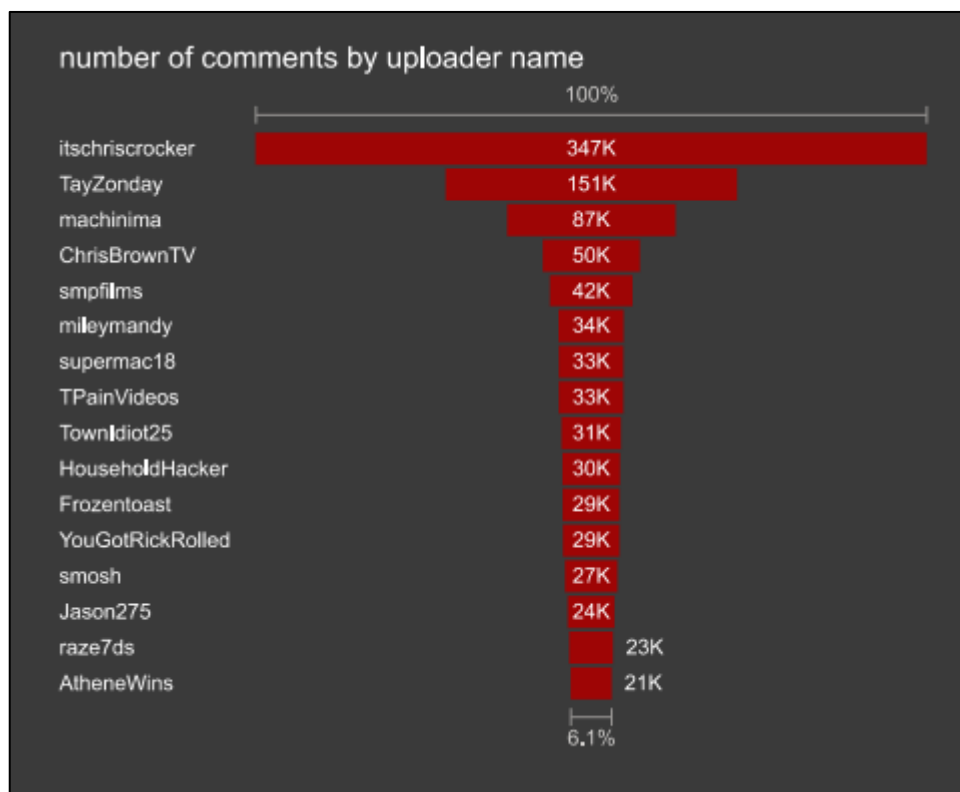


Figure 23: Data visualization: Number of comments by uploader name

8. Average of ratings grouped by uploader name.

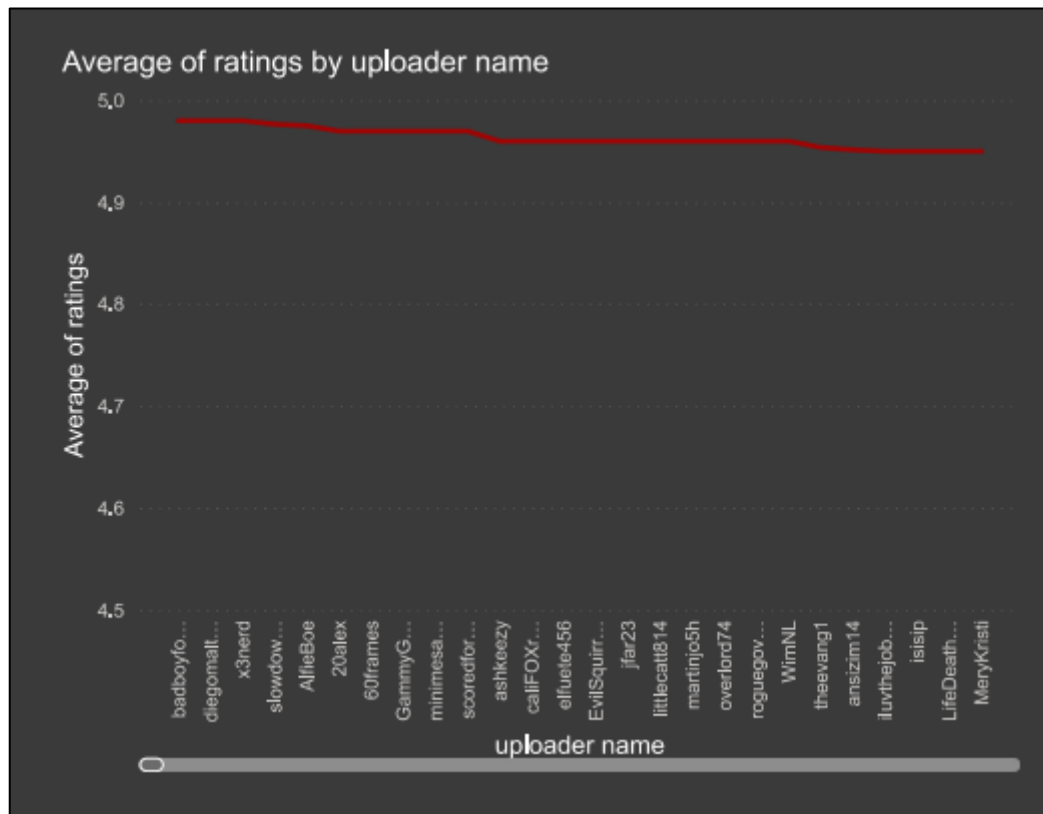


Figure 24: Data visualization: Average of ratings by uploader name

9. Time interval from when YouTube was created to when the video was uploaded grouped by uploader name.



Figure 25: Data visualization: Time interval by uploader name

10. Average of video length grouped by uploader name.

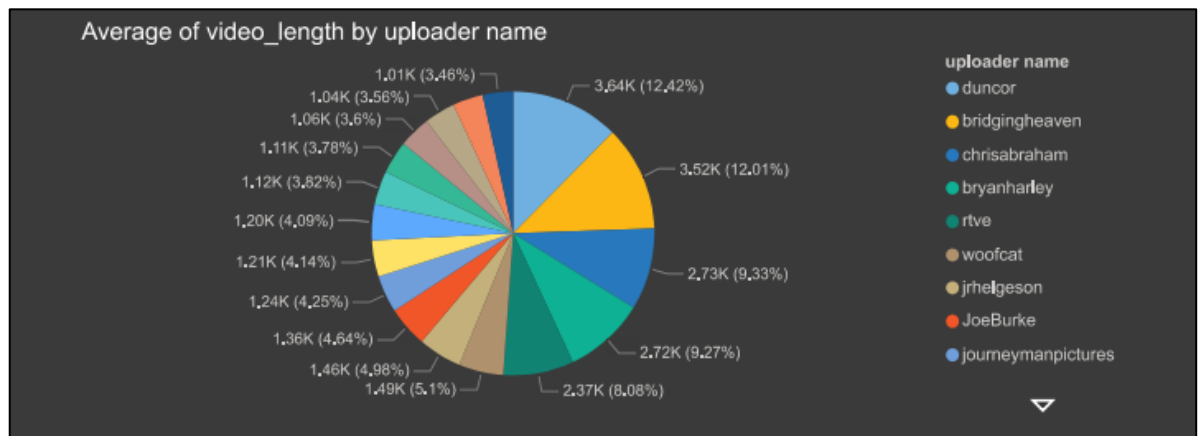


Figure 26: Data visualization: Average video length by uploader name

11. Number of views grouped by uploader name.

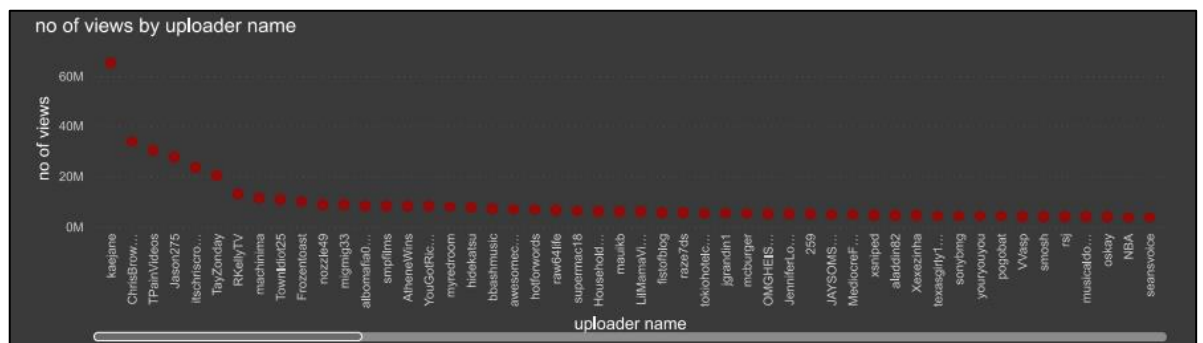


Figure 27: Data visualization: Number of views by uploader name

12. Ratings grouped by uploader name.

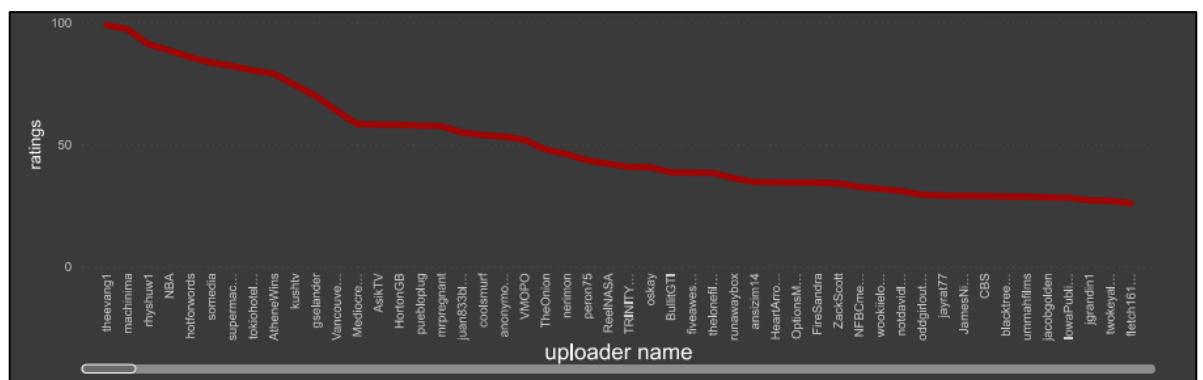


Figure 28: Data visualization: Ratings by uploader name

13. Conclusion

Entertainment has become a necessity of life for us and there is enough content today to keep us engaged for every moment of the rest of our lives. YouTube is one of the best examples of services that produce a massive amount of data in a brief period. We have demonstrated how we can extract insightful information from YouTube dataset using Big Data Analytics. For the same purpose, we have used technologies like Hadoop, Hive and Power BI.

14. 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] P. Merla and Y. Liang, "Data analysis using hadoop MapReduce environment," 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 4783-4785, doi: 10.1109/BigData.2017.8258541.

- [3] R. Harsh, G. Acharya and S. Chaudhary, "Big Data Hysteria, Cognizance and Scope," 2018 4th International Conference for Convergence in Technology (I2CT), 2018, pp. 1-8, doi: 10.1109/I2CT42659.2018.9057878.

- [4] J. Sang and C. Xu, "On Analyzing the 'Variety' of Big Social Multimedia," 2015 IEEE International Conference on Multimedia Big Data, 2015, pp. 5-8, doi: 10.1109/BigMM.2015.60.