YouTube Trend Analysis

A Mini Project Report Submitted by

Khushi S Bhimani
(4NM20AI021)

Krishna M S
(4NM20AI022)

UNDER THE GUIDANCE OF

Ms. Rakshitha
Associate Professor

Department of Artificial Intelligence and Machine Learning Engineering

In partial fulfillment of the requirements for the

BIG DATA ANALYTICS–20AM505

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Nitte(DU) established under Section 3 of UGC Act 1956 | Accredited with 'A+' Grade by NAAC

☎ 08258 - 281039 – 281263, Fax: 08258 – 281265

December 2022

NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎ : 08258 - 281039 - 281263, Fax: 08258 - 281265

Department of Artificial Intelligence and Machine Learning Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

Certified that the mini project work entitled

"Youtube Trend Analysis"

is a bonafide work carried out by

Khushi S Bhimani
(4NM20AI021)

Krishna M S
(4NM20AI022)

in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Artificial Intelligence and Machine Learning Engineering prescribed by Visvesvaraya Technological University, Belgaum during the year 2022-2023.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The mini project report has been approved as it satisfies the academic requirements in respect of the mini project work prescribed for the Bachelor of Engineering Degree.

13/12/22
Signature of Guide

Signature of HOD

Evaluation

Name of the Examiners	Signature with Date
1. MAHESH B.L.	13/12/2022
2. Rakshitha	13/12/22

ACKNOWLEDGEMENT

We believe that our mini project will be complete only after we thank the people who have contributed to make this mini project successful.

First and foremost, our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our mini project work at our college and providing us with all the needed facilities.

I acknowledge the support and valuable inputs given by**, Dr. Sharada U Shenoy** the Head of the Department, Artificial Intelligence and Machine Learning Engineering, NMAMIT, Nitte

We express our deep sense of gratitude and indebtedness to our guide **Ms. Rakshitha,** Associate Professor Artificial Intelligence and Machine Learning Engineering, for her inspiring guidance, constant encouragement, support and suggestions for improvement during the course for our mini project.

We also thank all those who have supported us throughout the entire duration of our mini project.

Finally, we thank the staff members of the Department of Artificial Intelligence and Machine Learning Engineering and all our friends for their honest opinions and suggestions throughout the course of our mini project.

Khushi S Bhimani

Krishna M S

TABLE OF CONTENTS

CHAPTER 1

1.1 ABSTRACT

YouTube is one of the most popular platforms for making money online. The biggest challenge that every youtuber face is “which videos grab more user’s attention(views)?” more specifically, what should be the video content so that the video can obtain more user views. In order to answer this question, we have made big data analytics on Kaggle dataset (YouTube’s trending video statistics) alternatively, we can also collect data in real-time using YouTube data API. Our Kaggle dataset contains information such as video_id, trending_date, views, likes, dislikes, title, channel, tags, category, etc. we provide a complete solution about choosing actual video content using both tags column (example: NBA| "Basketball"| "Sports") and views column of dataset as input and running an algorithm similar to word count algorithm in MR where mappers split the tags by ‘|’ character, associate view count as initial counter and reducers reduce the similar tags. This algorithm outputs tag view counts and when visualized though tools like tableau youtubers can decide which video content to choose. We use spark framework to run this algorithm because spark provides faster data processing, minimal implementation code compared to Hadoop. Also, we can perform real-time data processing using spark.

1.2 INTRODUCTION

Pyspark is an Apache Spark and Python partnership for Big Data computations. Apache Spark is an open-source cluster-computing framework for large-scale data processing written in Scala and built at UC Berkeley's AMP Lab, while Python is a high-level programming language. Spark was originally written in Scala, and its Framework PySpark was later ported to Python through Py4J due to industry adaptation. It is a Java library built into PySpark that helps Python interact with JVM objects dynamically; therefore, to run PySpark, you must also have Java enabled in addition to Python and Apache Spark.

Spark programmes operate independently on a cluster, which is a collection of computers linked together to perform computations on vast amounts of data, with each computer called a node, some of which are master nodes and others slave nodes. PySpark is used in distributed systems; in distributed systems, data and measurements are distributed because these systems combine the resources of lesser computers and potentially provide more cores and capacities than even a powerful local single computer. Check out How Data Analysis Using Pyspark is done.

1.3 SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS :

- Windows 7 or above / Linux.
- Python 2.7 or above.
- Jupyter Notebook.

3.2 HARDWARE REQUIREMENTS :

- Graphics Processing Unit (GPU).
- Intel Core i3 processor or above

CHAPTER 2

2.1 PROBLEM STATEMENT

Youtubers earn money from advertisers based on their video views and youtubers often ponder with this question “what type of videos should I make in order to get more user views?”. youtubers should always be aware of the current video trend and analysing YouTube’s big data helps youtubers cope up with the current trend and earn money through video views.

However, YouTube contains large amounts of real-time data (big data), and its difficult to analyse this large data with normal analytical tools so we chose spark for data analytics because spark offers real time data streaming and faster data processing.

2.2 PROPOSED SOLUTION

Data Crawling

The first step in solving this problem is data crawling. Trending YouTube videos data can be crawled through

- YouTube Data API
- Datasets available in online resources like Kaggle.

We collected big data set from Kaggle which contains 6 months of trending videos data on daily basis. We have also crawled small data set from YouTube API due to time constraint of this project.

Data source

https://www.kaggle.com/datasnaek/youtube-new#USvideos.csv---trending

Number of rows : 50,000

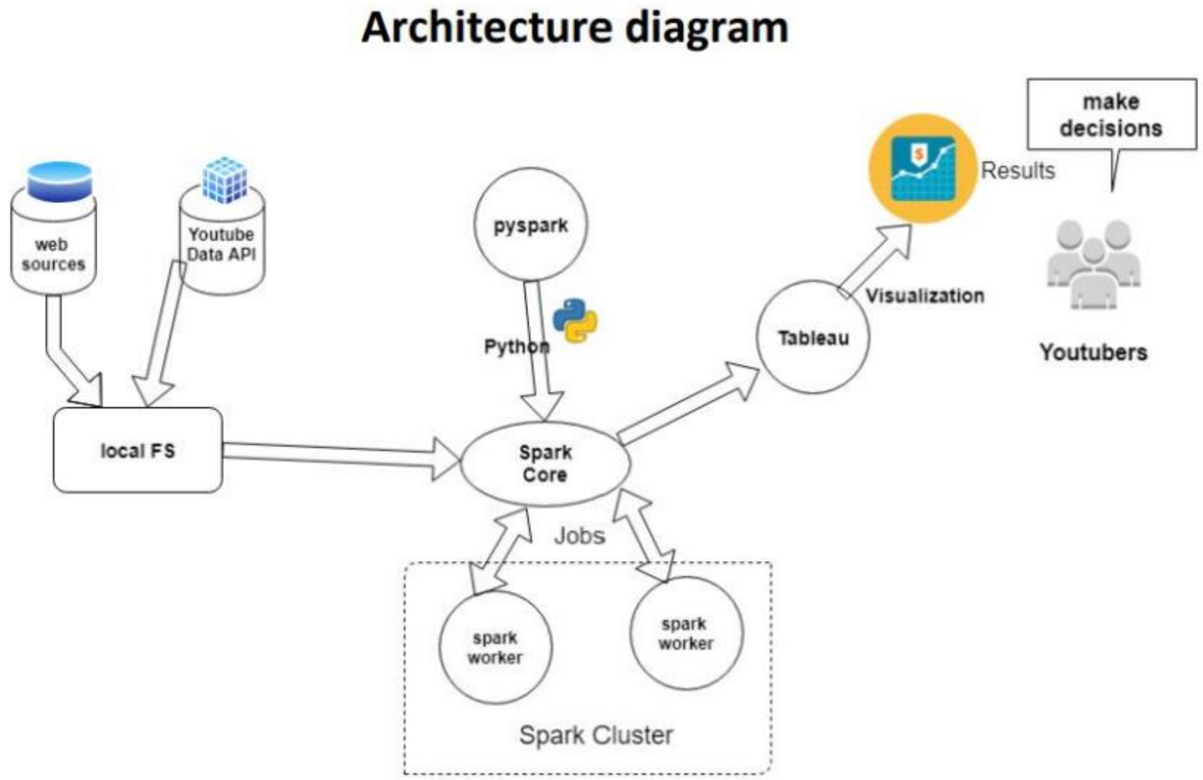
Columns : video_id, trending_date, title, channel_title, category_id, publish_time, tags, views, likes, dislikes, comment_count, thumbnail_link, comments_disabled, ratings_disabled, video_error_or_removed, description

Programming Model

We used python as our programming language because of its simplicity

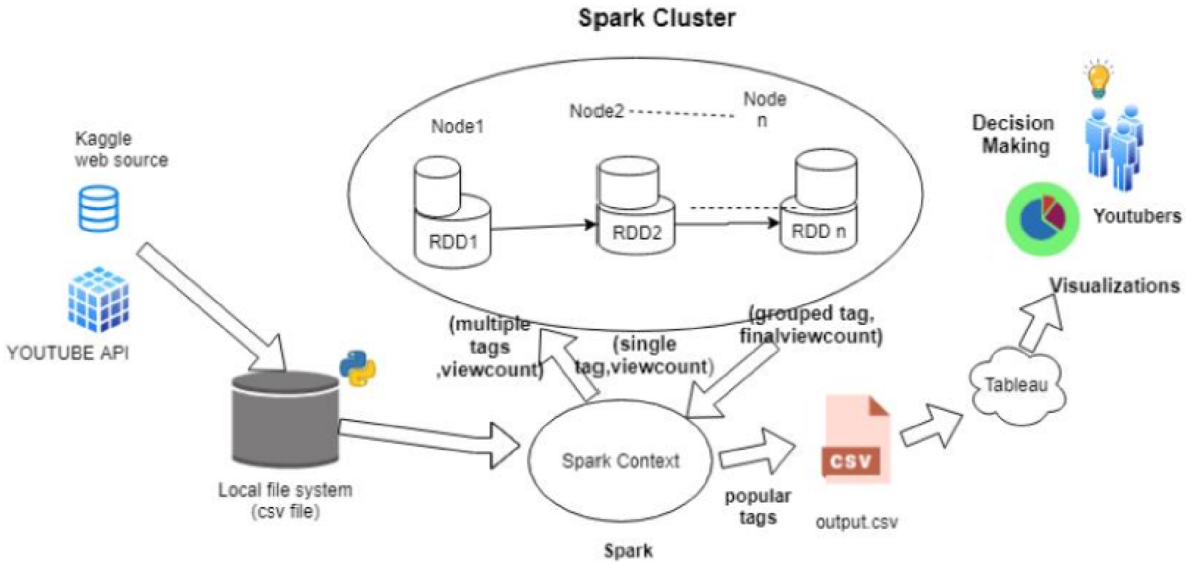
and ease of use. In order to code in python, we used pyspark API provided

by Apache spark.



The data extracted from YouTube API and Kaggle dataset are saved to local File system. Data from local file system is loaded to spark engine using pyspark API (python). Spark sends data and tasks (map, reduce) to worker nodes to do parallel processing. Map and reduce tasks for our use case is like wordcount algorithm. Finally spark sends output back to local system where output is fed to tableau for visualization and the results are used for decision making.

Data Pipeline



Data collected from different sources are saved locally and fed to spark.

Spark transmits data into RDDS to its worker nodes. All the workers nodes execute their tasks and sends the sorted output back to local fs.

Outputs are visualised in tableau for decision making.

2.3 SOLUTION COMPOSITION

Algorithm

1. Load dataset to RDD in spark.

2. Splits tags column by “|” and associate view count for each tag.

3. Apply log on view count and reduce will group similar tags by adding their respective view counts.

4. Sort data by view count and write top 35 tags and counts to disk as csv.

CODE

```
import findspark

findspark.init()

from pyspark.context import SparkContext

from pyspark.sql.session import SparkSession

from operator import add

import math

import pandas as pd


#Creating Spark Context

sc = SparkContext.getOrCreate()

#Creating Spark Session

spark = SparkSession(sc)

#Readind data set(csv file),dropping null values and selecting only tabs and views column

tags_views = spark.read.csv("file:///C:\\Users\\Hp\\A BDA PROJECT\\Youtube-Big-data-Analytics-using-Spark-master\\USvideos.csv", inferSchema = True, header = True).dropna().select("tags","views")
```

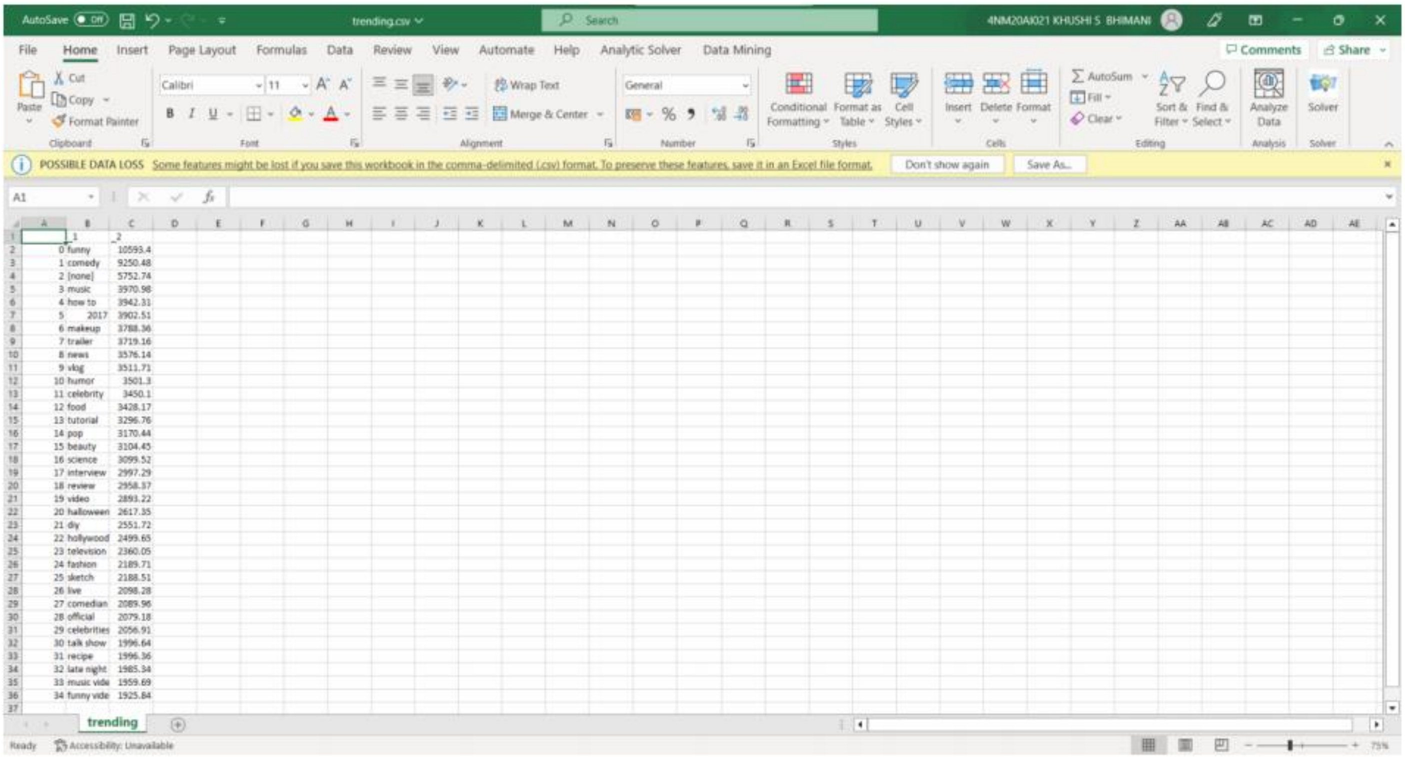
```
#Mapping by splitting tags with "|" character,
def tags_split(x):
    tags=x["tags"].split("|")
    result=[]
    for every in tags:
        #Associating view count as counter and applying log on view count because views count will
be really large
        if not str(x["views"]).isdigit() or every==None:
            continue
        if x["views"]!=0:
            result.append((every.strip("\"").lower(),math.log(x["views"])))#Stripping unnecessary
characters
    return tuple(result)
rdd1=tags_views.rdd.flatMap(tags_split).reduceByKey(add)#reduce by similar tags and adding its
view count
#Top Tags are queried from RDD by Sorting RDD's in descending order of view count
toptags=rdd1.takeOrdered(35, key = lambda x: -x[1])
df=spark.createDataFrame(toptags)
#Writing back to Disk
print(df.head(5))
pandasDF = df.toPandas()
print(pandasDF)
pandasDF.to_csv("E:\\trending.csv")
# df.repartition(1).write.csv(path="file:///D:\\trending.csv")
spark.stop()
```

- This code is implemented with python and well commented.
- We read data from local file system and selected only “tags” and “views” column into RDD.
- We then split tags by “|” character, associate each tag with view count and reduce by same key.

We apply log on views because they could sum to large values, so we minimise them with log.

- Finally, we sort the top trending tags by view count and write back to disk.

2.4 OUTPUT



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...

	A	B	C
1		_1	_2
2	0	funny	10593.4
3	1	comedy	9250.48
4	2	[none]	5752.74
5	3	music	3970.98
6	4	how to	3942.31
7	5	2017	3902.51
8	6	makeup	3788.36
9	7	trailer	3719.16
10	8	news	3576.14
11	9	vlog	3511.71
12	10	humor	3501.3
13	11	celebrity	3450.1
14	12	food	3428.17
15	13	tutorial	3296.76
16	14	pop	3170.44
17	15	beauty	3104.45
18	16	science	3099.52
19	17	interview	2997.29
20	18	review	2958.37
21	19	video	2893.22
22	20	halloween	2617.35
23	21	diy	2551.72
24	22	hollywood	2499.65
25	23	television	2360.05
26	24	fashion	2189.71
27	25	sketch	2188.51
28	26	live	2098.28
29	27	comedian	2089.96
30	28	official	2079.18
31	29	celebrities	2056.91
32	30	talk show	1996.64
33	31	recipe	1996.36
34	32	late night	1985.34
35	33	music vide	1959.69
36	34	funny vide	1925.84

CHAPTER 3

3.1 CONCLUSION

• From this word cloud visualization, it is evident that during the time data is collected “comedy and funny” videos grabbed more user views.

• Any youtuber can now choose video content with the help of trend visualized.

• As time passes, the trend changes so it is always better to do realtime data processing with spark streaming API to cope up with the trend

3.2 SUMMARY

❖ Spark took less than 5 minutes to process 50000 records which is incredibly fast compared to other tools.

❖We can analyse both real-time and periodic data using spark.

❖Spark provides various inbuilt programming capabilities; in our use case we use pyspark.

❖ Data analysed through spark can be fed to visualization tools like tableau for gaining knowledge and decision making.

3.3 BIBLIOGRAPHY

- https://medium.com/big-data-engineering/how-to-install-apache-spark-2-x-in-your-pc-e2047246ffc3
- https://www.youtube.com/watch?v=639JCua-bQU
- https://spark.apache.org/docs/1.2.0/programming-guide.html
- https://www.codementor.io/@jadianes/spark-python-rdd-basics-du107x2ra