



NEW YORK INSTITUTE OF TECHNOLOGY

College of Engineering and Computing
Science

Project Report

Top Subscriber on YouTube

NIRAJ RAJ

Term: Summer 2023

DTSC 701 – Introduction to Big Data

Professor:
Liangwen Wu

Table of Content

1. Introduction.....	3
2. Dataset overview.....	3
3. Channel Category distribution, KNN and Confusion Matrix.....	4
4. AWS spark cluster setup.....	5
5. SSH access.....	7
6. The final output of Spark.....	8
7. Conclusion.....	9

1. Introduction

In the digital age, YouTube has emerged as a global platform where content creators, artists, and organizations showcase their creativity, share knowledge, and engage with audiences on an unprecedented scale. With millions of channels and billions of videos, YouTube represents a rich landscape of diverse content genres, captivating millions of viewers around the world. As part of this vibrant ecosystem, understanding the characteristics and dynamics of YouTube channels becomes pivotal.

This report embarks on an exploratory journey through the "Most Subscribed 1000 YouTube Channels" dataset, meticulously curated from the vast YouTube universe and sourced from Kaggle. With an unwavering focus on data-driven insights, the analysis encompasses two key pillars: the intricate world of channel categorization and the strategic implementation of machine learning techniques, specifically K-Nearest Neighbors (KNN) classification. Through the fusion of these analytical facets, we aim to illuminate the essence of YouTube content diversity and demonstrate the potential of intelligent algorithms in categorization tasks.

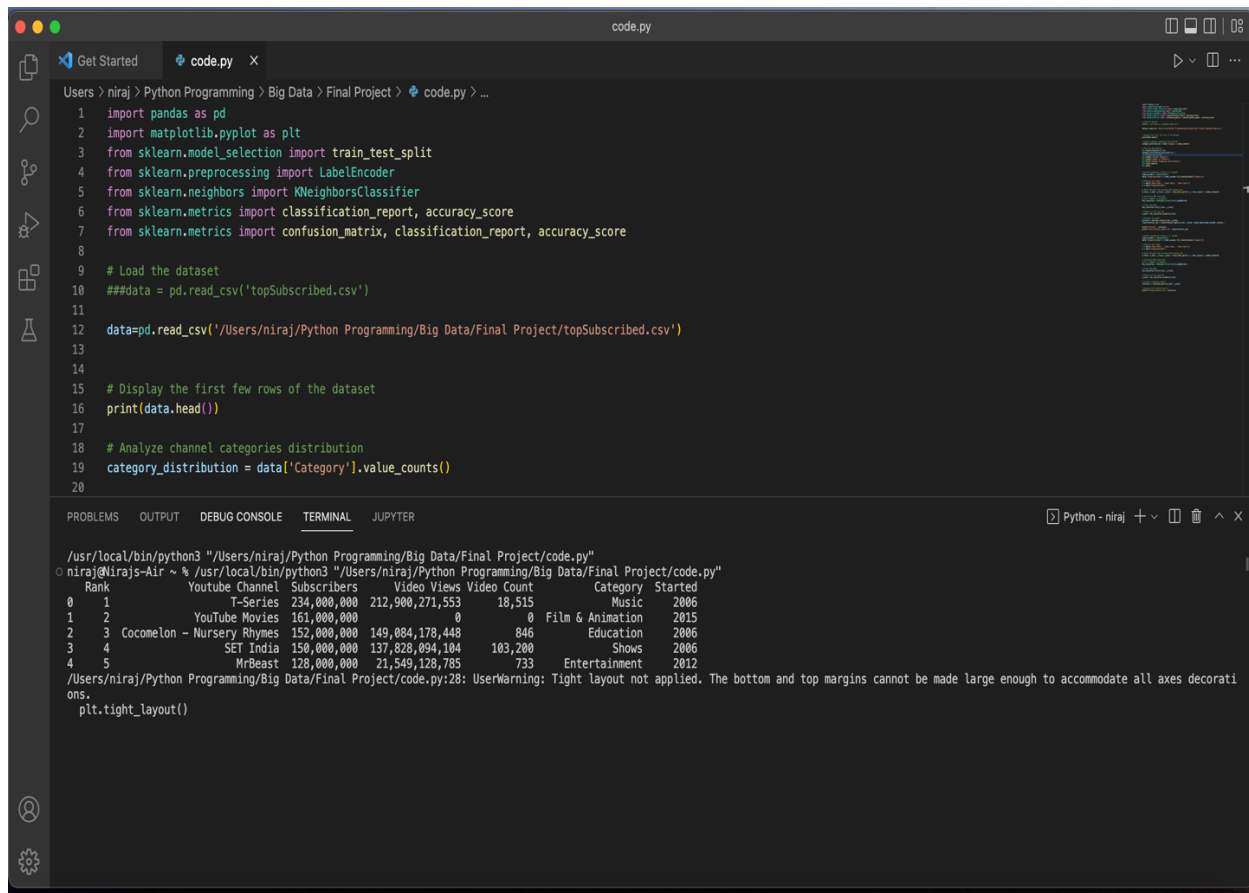
2. Dataset Overview

The dataset under investigation includes a broad view of the top 1000 YouTube channels with the most subscribers. Each entry in the dataset contains a wealth of channel-specific information, such as rank, title, subscriber count, video views, video count, categorical classification, and the channel's foundational year of creation. The foundation of our analysis is made up of this collection of characteristics.

Link: <https://www.kaggle.com/datasets/themrityunjaypathak/most-subscribed-1000-youtube-channels>.

3. Channel Categories Distribution

In the age of digital media, channel categorization serves as an illuminating compass, navigating users through the labyrinth of content offerings. The elucidation of category distribution is akin to unraveling the tapestry of human interests and preferences interwoven across the YouTube spectrum. By meticulously sifting through the dataset and deploying visualizations, we gleaned insights into the ebbs and flows of content genres that captivate the YouTube audience.



```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import LabelEncoder
5 from sklearn.neighbors import KNeighborsClassifier
6 from sklearn.metrics import classification_report, accuracy_score
7 from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
8
9 # Load the dataset
10 ##data = pd.read_csv('topSubscribed.csv')
11
12 data=pd.read_csv('/Users/niraj/Python Programming/Big Data/Final Project/topSubscribed.csv')
13
14
15 # Display the first few rows of the dataset
16 print(data.head())
17
18 # Analyze channel categories distribution
19 category_distribution = data['Category'].value_counts()
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

/usr/local/bin/python3 "/Users/niraj/Python Programming/Big Data/Final Project/code.py"

niraj@Nirajs-Air ~ % /usr/local/bin/python3 "/Users/niraj/Python Programming/Big Data/Final Project/code.py"

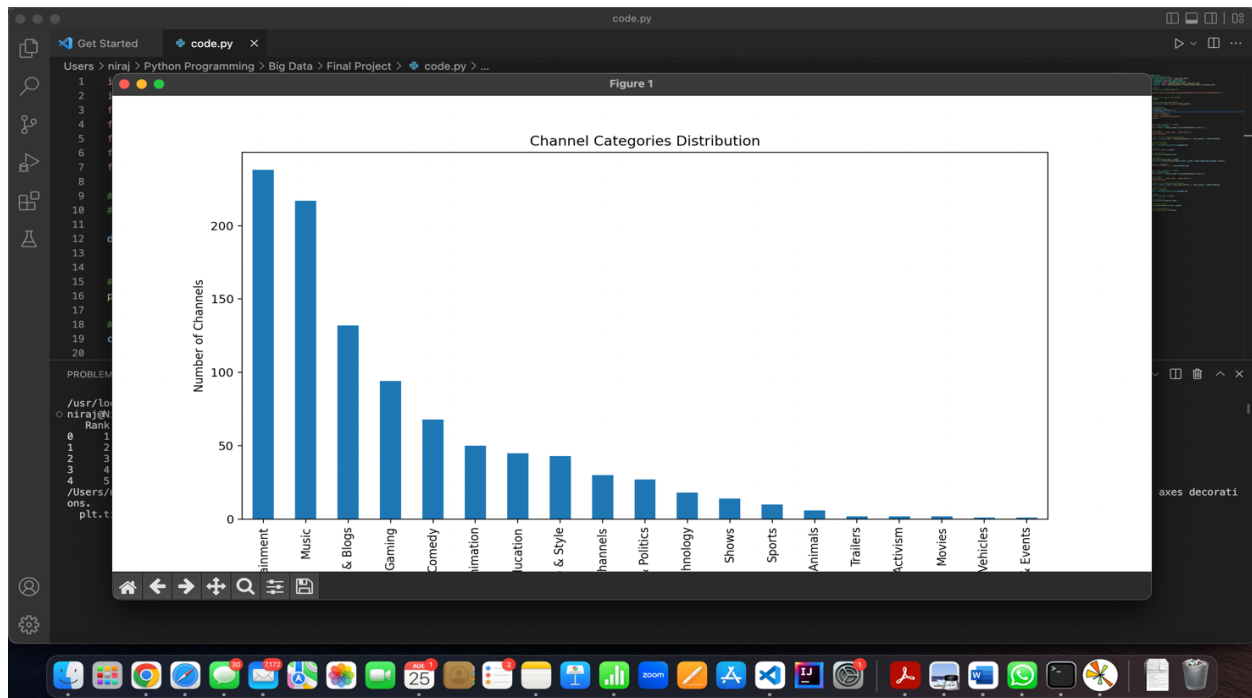
Rank	Youtube Channel	Subscribers	Video Views	Video Count	Category	Started	
0	1	T-Series	234,000,000	212,900,271,553	18,515	Music	2006
1	2	YouTube Movies	161,000,000	0	0	Film & Animation	2015
2	3	Cocomelon - Nursery Rhymes	152,000,000	149,084,178,448	846	Education	2006
3	4	SET India	150,000,000	137,828,094,104	103,200	Shows	2006
4	5	MrBeast	128,000,000	21,549,128,785	733	Entertainment	2012

/Users/niraj/Python Programming/Big Data/Final Project/code.py:28: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to accommodate all axes decorations.
plt.tight_layout()

K-Nearest Neighbors (KNN) Classification & Confusion Matrix

In an era dominated by digital prowess, the application of machine learning algorithms breathes life into data, transforming it from mere numbers into predictive insights. The K-Nearest Neighbors (KNN) algorithm emerges as a torchbearer in the realm of classification. Our voyage into the application of KNN unfolds as we harness its power to predict channel categories based on features such as subscriber count, video views, and video count.

The ensuing bar chart orchestrates the dance of channel categories, echoing the rhythmic cadence of subscriber preferences. The orchestra is led by the crescendo of the "Music" category, harmonizing with the serenades of "Shows," "Entertainment," and "Education," each revealing a unique note in the symphony of YouTube.



Python code is on GitHub.

GitHub Link: <https://github.com/niraj2810988/FinalProject>

4. AWS Spark setup

The canvas grows beyond the confines of local processing as we go toward scalability and computational power. Utilizing cloud resources, the AWS Spark cluster spreads its wings and orchestrates the dance of data analysis. We entered the world of distributed computing, where data flows in torrents and insights come from the convergence of processing nodes, with the deployment of three instances and the magic of Spark setup.

1. Visit the AWS Website: Go to the official Amazon Web Services website at <https://aws.amazon.com/>
2. Sign Up: Click "Create an AWS Account" and follow the instructions. You'll need to give your email, password, and billing info.
3. Verify Identity: Complete identity verification, which might include using a phone number for a verification code.
4. Set Up Payment: Enter payment details to enable billing for your AWS account.
5. Accessing EMR: Access the "Amazon EMR" service through the AWS Management Console. Initiating Cluster Creation: Initiate the process by selecting the "Create cluster" button, which will lead you to the configuration of your EMR cluster.

6. Defining Cluster Settings: Assign a suitable name to your cluster and specify your preferred Spark application and version. Choosing Instance Types: Opt for 3 instance types for both master and core nodes.
7. Determining Instance Count: Indicate the desired number of core instances, in your case, select 3 instances. Configuring Security and Access: Set up security groups, opt for an EC2 key pair (if SSH access is needed), and configure other access parameters.
8. Commencing Cluster Launch: After reviewing your configurations, finalize by selecting the "Create cluster" button to launch your Spark cluster.

My cluster Updated less than a minute ago [Actions](#)

Summary

Cluster info	Applications	Cluster management	Status and time
Cluster ID j-ONRXF9YN1LYG	Amazon EMR version emr-6.12.0	Log destination in Amazon S3 Logging not configured	Status ✔ Waiting
Cluster configuration Instance groups	Installed applications Spark 3.4.0, Zeppelin 0.10.1	Persistent application UIs Spark History Server YARN timeline server	Creation time August 24, 2023, 18:55 (UTC-04:00)
Capacity 1 Primary 3 Core 0 Task		Primary node public DNS ec2-3-15-222-135.us-east-2.compu te.amazonaws.com Connect to the Primary Node using SSH	Elapsed time 55 minutes, 17 seconds

Properties | Bootstrap actions | Instances | Steps | Applications | Configurations | Monitoring | Events | Tags (1)

Operating system [Info](#) | **Cluster logs** [Info](#) | **Cluster termination** [Info](#)

Amazon Linux release | Archive log files to Amazon S3 | [Edit cluster termination](#)

Instances (4) [Info](#) [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input type="checkbox"/>	-	i-0e44eb45e6f9eb228	✔ Running	m5.xlarge	✔ 2/2 checks passed	No alarms	us-east-2b
<input type="checkbox"/>	-	i-0fc27329a4d1f58e6	✔ Running	m5.xlarge	✔ 2/2 checks passed	No alarms	us-east-2b
<input type="checkbox"/>	-	i-070340f28eaf1cbfb	✔ Running	m5.xlarge	✔ 2/2 checks passed	No alarms	us-east-2b
<input type="checkbox"/>	-	i-0e73c3097ffe370d0	✔ Running	m5.xlarge	✔ 2/2 checks passed	No alarms	us-east-2b

Select an instance

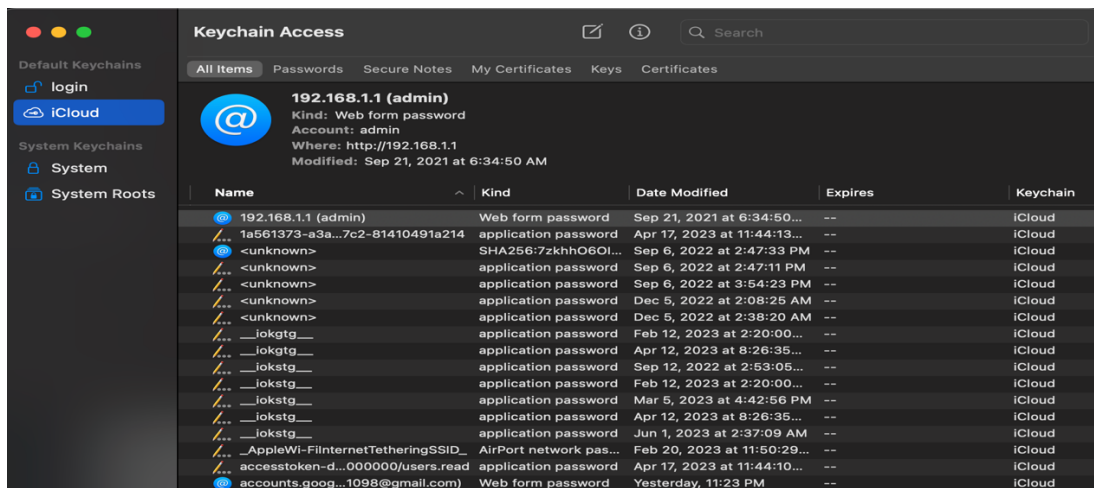
5. SSH Access

1. SSH Key: If you configured an EC2 key pair during cluster creation, you can use it for SSH access.
2. Terminal: On your local machine, open a terminal window.

```
niraj — zsh — 80x24

[Enter passphrase (empty for no passphrase): ]
[Enter same passphrase again: ]
Your identification has been saved in cd -downloads
Your public key has been saved in cd -downloads.pub
The key fingerprint is:
SHA256:dhs6hsL5k/+6/dMZ6wHVxew9MP2m/7jc+cOSZn2H/M4 niraj@Nirajs-MacBook-Air.local
The key's randomart image is:
+----[RSA 4096]-----+
|          .o.|
|         o o+|
|        +.+|
|       ..=|
|      S o . o.|
|     . . o o o .o|
|    + ..+ . o=*|
|   oo. o .**0=|
|  .o++o.+o*=E|
+-----[SHA256]-----+
[niraj@Nirajs-MacBook-Air ~ % ssh -i ~/BIGDATA.pem hadoop@ec2-3-15-222-135.us-east-2.compute.amazonaws.com
Warning: Identity file /Users/niraj/BIGDATA.pem not accessible: No such file or directory.
cd
```

3. Navigate to Key Pair: If you created a new EC2 key pair, navigate to the directory where you saved the private key file (.pem) on your local machine.



4. Change Key Permissions: To ensure secure access, restrict the permissions of the private key file:


chmod 400 mykeypair.pem


5. SH Command: Use the SSH command to establish an SSH connection to one of the instances. Replace your-key-pair.pem with the actual path to your private key file and public-ip with the instance's public IP address:

[`ssh -i ~/mykeypair.pem hadoop@ec2-3-15-222-135.us-east-2.compute.amazonaws.com`](#)

Connect to the primary node using SSH



 This cluster was launched without an EC2 key pair, this must be configured during cluster launch to enable SSH access.

You can connect to the Amazon EMR primary node using SSH to perform actions like running interactive queries, examining log files, submit Linux commands, and view web interfaces hosted on Amazon EMR clusters. [Learn more](#) 

Windows

Mac/Linux

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.

2. To establish a connection to the primary node, enter the following command. Replace `~/mykeypair.pem` with the location and filename of the private key file (.pem) that you used to launch the cluster.

```
ssh -i ~/mykeypair.pem hadoop@ec2-3-15-222-135.us-east-2.compute.amazonaws.com
```

3. Enter yes to dismiss the security warning.

[View web interfaces hosted on Amazon EMR clusters](#) 

Close

6. Connected: You are now connected to the instance via SSH. You can use this terminal to interact with the instance, deploy Spark jobs, and manage the cluster.

