YouTube Trending Video Dashboard

Kaustubh[1]

[1]*UG Student, Maharaja Agrasen Institute of Technology, Dept. of IT, Delhi, India*

---***---

Abstract - *YouTube is not only serving as an entertainment platform for the films and the television industry but it has also emerged as a learning platform for many students. Content creators on YouTube, commonly known as "YouTubers" are pushing their content every single day and hour to be relevant to their audiences. It's a known fact that the YouTube algorithm is not a publicly available code and it is kept private for most of the time. Also, YouTube has shared any intuition about what factors are considered for a video to be in trending section which leaves its audience in dilemma about posting videos. The trending section is where some videos are listed as trending and it is designed in such a way that every user of this platform checks it once in a while. This dashboard aims to present the trending section of YouTube for India in terms of the trends and observations which could be helpful for creators to push their content to more audiences.*

***Key Words*: YouTube, Trending Section, Metadata Analysis, Python, Data Representation, Pandas, Plotly, Heroku Deployment**

1. INTRODUCTION

A particular video content on YouTube is not limited not to the actual video content being displayed to the user. A user visiting a YouTube video is encountered with likes, video title, description, comments, publishing date, publishing channel, number of subscribers of the channel and much more elements. All these features directly or indirectly contribute to a video being on the trending section.

This dashboard can help in finding and comparing key aspects of a YouTube video and help budding creators, businesses to align their content to the most popularly used key aspects be included in the trending section in future.

2. LITERATURE SURVEY

YouTube is one of the most popular platforms and a lot of research has been done on it. Despite its importance, YouTube trending videos analysis has yet to be thoroughly investigated. Trending video analysis still has a lot of room for improvement.

Ouyang, Li, and Li investigated the prediction of internet video popularity. The popularity forecasting problem was divided into two tasks in this study: video popularity prediction and video view count prediction. With a core set of variables and numerous categorization algorithms, researchers first anticipate the future popularity levels of videos. The study then used sophisticated regression models to forecast the number of views based on the popularity levels. (Ouyang, Li and Li, 2016) [1].

The impact of meta-data elements such as title, tag, thumbnail, and description on the popularity and trendiness of YouTube videos was investigated by Hoiles, Aprem, and Krishnamurthy (Hoiles, Aprem and Krishnamurthy, 2017).[2] Their study used a variety of Machine Learning algorithms to predict the YouTube video popularity based on the video's meta-features as well as other factors like the number of subscribers.

S. Amudha et al. looked into the YouTube popular video metadata. The study employed an unsupervised dataset and the Decision Tree technique from Machine Learning to estimate the most effective courier service. The study used a views ratio per category to provide a simplified output of views, likes, dislikes, and comments scatter plot. Using pre-processing analysis, the thesis aids in understanding the value of these attributes (S. Amudha et al., 2020) [3].

3. DATA GATHERING

We had to create a Python script to fetch the data from the YouTube Data API. The YouTube data API has a clean interface to obtain any type of data from their platform. It authenticates via API keys. The key was generated by registering an account under Google Cloud Platform (GCP), then creating a project and then enabling the YouTube API under that project. The current thresholds for this API are maximum of 10k calls per day which is more than enough for our project [4].

The data returned by the YouTube API is of the nested JSON format that further needs data wrangling. Also, one issue we had to deal with was the paginated results. The JSON results were paginated with each page having token for the next one. Here is one of the JSON output for a video as show in Fig-1.

```json
{
    "kind": "youtube#video",
    "etag": "-Hp3huEh0ze_OVolPzJh9QM55oI",
    "id": "qj7dsKQE7gk",
    "snippet": {
      "publishedAt": "2022-05-06T14:01:45Z",
      "channelId": "UCOV4B25tLhyLkGB_N8olPpw",
      "title": "Video Title",
      "description": "Video Description",
      "channelTitle": "Ruh",
      "tags": [
        "#mujjuu___14",
        "#deepeshzo",
        "#sevengers",
        "#ajaysharma",
        "#payalteena",
        "#pyushjoshi",
        "#sadiqahmed"
      ],
      "categoryId": "22",
      "liveBroadcastContent": "none",
      "localized": {
        "title": "Video Title",
        "description": "Video Description"
      }
    },
    "statistics": {
      "viewCount": "10478337",
      "likeCount": "818643",
      "favoriteCount": "0",
      "commentCount": "24646"
    }
  },
```


Fig -1: JSON Output for one of the Videos

4. DATA WRANGLING AND STORAGE

Data wrangling usually entails converting and mapping data from one raw format to another in order to facilitate data consumption. The data should not have any null, special or unrecognized characters.[5]

We created the logic to fetch the data but now the question was how to manage all the files and load them at the same time. We used glob to find all the filenames with the extension ".csv" and loaded them one by one in master Pandas DataFrame. So now every time there is new csv file in the data loading folder, that file contents will be loaded automatically into the master DataFrame.

After loading this DataFrame, there were several cleaning and memory reduction processes which were applied on this DataFrame. These steps include:

- Reducing memory usage: All the integer data types are inferred as int64 which is not an efficient approach to store a value as small as one digit. It affects speed of execution for further data analysis. Therefore, these also need to be in the correct integer range.[6]

- Filling null values. These are the rows whose values are null. We can see that null values are denoted by NaN. These null values are replaced with an empty string, in case of string type column and 0, in case of integer column.

- The date columns should be in datetime format used by Pandas. It helps in accessing the specific date attributes. Also, the time zone returned by the YouTube is in the UTC. We need to convert this into Indian IST time.

- Title and description cleaning for emojis.

- Populating the category data for the videos

- Calculating the title length of the video

One of the data cleaning functions, implemented in Python is shown in Fig-2.

```python
def dataClean_Filling(df: pd.DataFrame) -> pd.DataFrame:
    df["description"] = df["description"].fillna(value="")
    df["channelTitle"] = df["channelTitle"].fillna(value="")
    df["tags"] = df["tags"].fillna(value="")
    df["trending_date"] = pd.to_datetime(df["trending_date"], utc=True)
    df["publishedAt"] = pd.to_datetime(df["publishedAt"])
    df["publishedAt"] = df.set_index("publishedAt").tz_convert("Asia/Kolkata").index
    df["cleanedTitle"] = df["title"].apply(emoji_free_text)
    df["cleanedDescription"] = df["description"].apply(emoji_free_text)
    df = addCategories(df)
    df["fullyCapitalizedTitle"] = df["cleanedTitle"].apply(capitalizedTitle)
    df["cleanedTags"] = df["tags"].apply(cleanTags)
    df["title_length"] = df["cleanedTitle"].apply(lambda x: len(x.split()))
    return df
```

Fig -2: Pandas DataFrame Cleaning function

In the Fig-2 function, an input for Pandas DataFrame is taken and then various transformations are performed on this DataFrame. The description, channel title and tags columns are filled with empty string. The publishing date is converted to Pandas datetime format and then the time zone conversion takes place. We also removed the emojis from the channel title and description. The logic to add the categories corresponding to the video is also implemented here.

5. AUTOMATING ELT USING GITHUB ACTIONS

GitHub actions are the CI/CD pipelines that allows the applications to be automatically trigger updating process and if everything passes the checks, automatically pushes the application in the deployment stage.[9] These pipelines are the heart of our application. We wrote the fetching script in a way that it takes in the command line arguments for the YouTube API and the GitHub API.

The YouTube API key is used to authenticate while fetching data from the YouTube. The GitHub API is used to push the files to the required destination folder. The GitHub actions script is written using YAML language. GitHub has made a process for making an action to be used by other users. One has to create a file with YML extension in the root of the action repository with ".github/workflows" path which contains all the input-output of the action. Also, it defines where the action will be run, the environment.

We wrote an integration script (Fig-3) to run every day at one particular time, checkout the repository, install the required libraries, run the fetching script with the required command-line arguments, generate the CSV file and push the file to the GitHub repository.

```yaml
name: Data Updation
on:
  schedule:
      # Runs at the end of every day
      - cron: '0 23 * * *'
  workflow_dispatch:

jobs:
  job_1:
    name: data-addition
    runs-on: ubuntu-latest
    steps:
        - name: Repo Checkout
          uses: actions/checkout@v2
        - name: Python Setup
          uses: actions/setup-python@v3
          with:
            python-version: '3.8'
        - name: Install Packages
          run: |
            python -m pip install --upgrade pip
            pip install -r requirements-cron.txt
        - name: Data Extraction Script
          run: python extract_data.py ${{ secrets.YOUTUBE_API }} ${{ github.workspace }}
        - name: commit files
          run: |
            git config --local user.email "action@github.com"
            git config --local user.name "GitHub Action"
            git add -A
            git commit -m "update data" -a
        - name: push changes
          uses: ad-m/github-push-action@v0.6.0
          with:
            github_token: ${{ secrets.GITHUB_TOKEN }}
            branch: main
```

Fig -3: YAML file to fetch YouTube API daily for data

6. HEROKU INTEGRATION

The dashboard we built will work for all the use cases on local system only. To enable this to be used by general public, this needs to be created as public dashboard where anyone check the current trends.

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. [10] For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. Therefore, we choose Heroku to deploy our Python dashboard.

Now the problem we faced here is the revoked GitHub Oauth tokens from the Heroku. In the recent data breach at Heroku, several GitHub Oauth tokens were stolen and used for unauthorized access. This led to Heroku revoking GitHub access to directly deploy applications. [11]

What this means is that now we cannot enable automatic deploys from the Heroku dashboard and now we had to manually write the script to redeploy the dashboard as soon as the data file is pushed to the GitHub repository. We found the already existing solution to deploy applications via GitHub actions.

There, now as soon as there is a "push" activity in the repository, the deploy script is triggered which redeploys the dashboard. The script for this continuous deployment is shown in Fig-4.

```yaml
name: Deploy

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: akhileshns/heroku-deploy@v3.12.12
        with:
          heroku_api_key: ${{secrets.HEROKU_API_KEY}}
          heroku_app_name: "youtube-trending-dashboard"
          heroku_email: "kaustubhgupta1828@gmail.com"
```

Fig -4: YAML file to deploy dashboard on every new push to the repository

7. PLOTLY DASHBOARD

A dashboard for analytics is a set of widgets that enable quick data viewing. A dashboard is a reporting tool that allows website analysts to measure numerous data such as online conversions, visitors, and page views to conveniently monitor the operation of a website. The analytics dashboard is flexible, powerful, and simple to use, which is one of its best advantages.[12]

We created a dashboard that is interactive in nature and presents the charts in the form of visualizations. The visualizations are created keeping in mind that an average person can understand the context of the findings by just looking at the graphs.

This dashboard will be helpful for anyone who wants to know to gather knowledge about how YouTube trending videos can be perceived from a mathematical point of view. It can also be helpful for YouTube content creators, who are working hard to get their video into the trending section

One can look at the scatter plot for trending videos publishing hours. Dots are used to represent values for two different numeric variables in a scatter plot. The values for each data point are indicated by the position of each dot on the horizontal and vertical axes.

Scatter plots are used to visualize the relationships between variables. A scatter plot's main aim is to show and observe relationships between two numeric variables. When looking at the data as a whole, the dots in a scatter plot reflect not just the values of individual data points, but also patterns.[13] The Fig-5 shows the dashboard preview for the scatter plot between the number of trending videos and publishing hours.

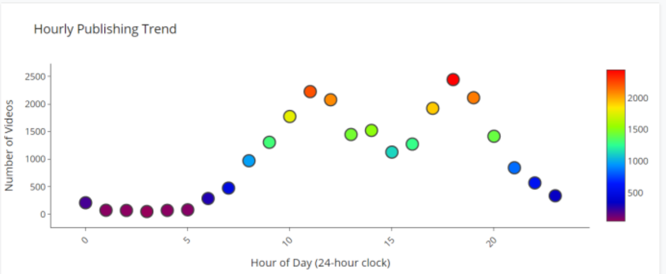

Fig -5: Scatter plot for number of trending videos per publishing hour

Another type of plot that can be generated for representing the number of trending videos on different weekdays is a line plot. A line plot is a graph that displays data frequency on a number line. When comparing fewer than 25 numbers, a line plot is the best option. It's a quick and easy way to arrange the information [14]. Below Fig-6 shows the dashboard preview for line plot for the number of trending videos on weekdays.

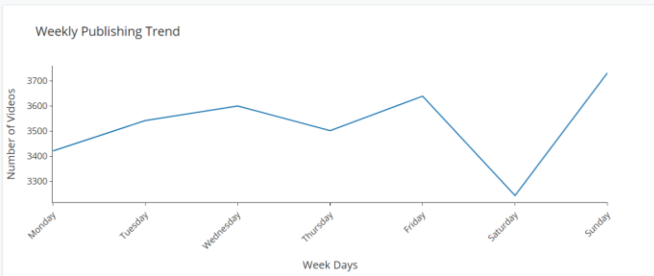

Fig -6: Line plot for the number of trending videos on weekdays.

Next plot that can be generated for representing whether the Video Title is fully capitalized or not using a Pie chart. A pie chart is a pictorial representation of data in the form of a circular chart or pie where the slices of the pie show the size of the data.

A list of numerical variables along with categorical variables is needed to represent data in the form of a pie chart. The arc length of each slice and consequently the area and central angle it forms in a pie chart is proportional to the quantity it represents. Fig-7 shows the dashboard preview for pie chart for whether the video title was fully capitalized or not.

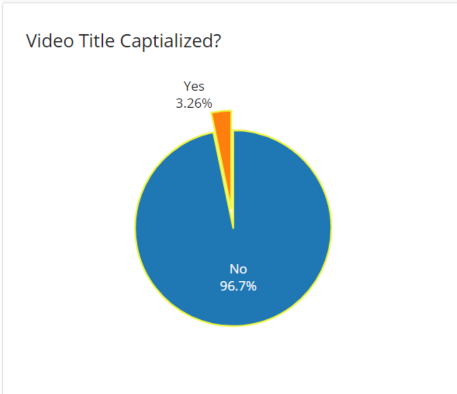

Fig -7: Pie chart for whether the video title was fully capitalized or not.

Next plot that can be generated for representing the top 10 trending channels of time using a Bar chart. A bar chart may be either horizontal or vertical. The important point to note about bar charts is their bar length or height—the greater their length or height, the greater their value. Bar charts are one of the many techniques used to present data in a visual form so that the reader may readily recognize patterns or trends.

Bar charts usually present categorical variables, discrete variables or continuous variables grouped in class intervals. They consist of an axis and a series of labelled horizontal or vertical bars. The bars depict frequencies of different values of a variable or simply the different values themselves. The numbers on the y-axis of a vertical bar chart or the x-axis of a horizontal bar chart are called the scale. Fig-8 shows the dashboard preview for bar chart for top 10 trending channels of time.

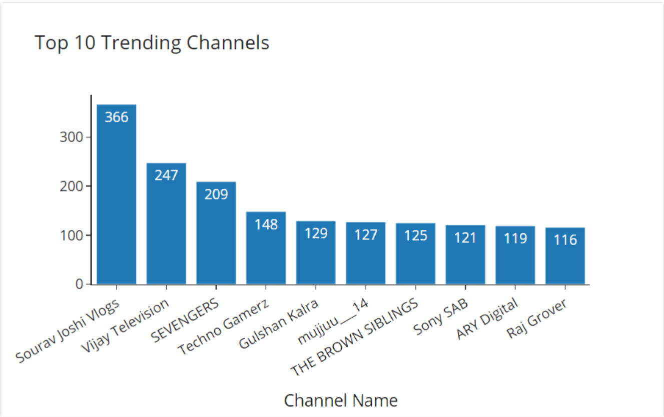

Fig -8: Bar chart for top 10 trending channels of time.

Next plot that can be generated for representing the video title length distribution using a histogram. A histogram can be defined as a set of rectangles with bases along with the intervals between class boundaries. Each rectangle bar depicts some sort of data and all the rectangles are adjacent. The heights of rectangles are proportional to corresponding frequencies of similar as well as for different classes. Let's learn about histograms more in detail. A histogram graph is a bar graph representation of data.

It is a representation of a range of outcomes into columns formation along the x-axis. in the same histogram, the number count or multiple occurrences in the data for each column is represented by the y-axis. It is the easiest manner that can be used to visualize data distributions. Fig-9 shows the dashboard preview for histogram for video title length distribution

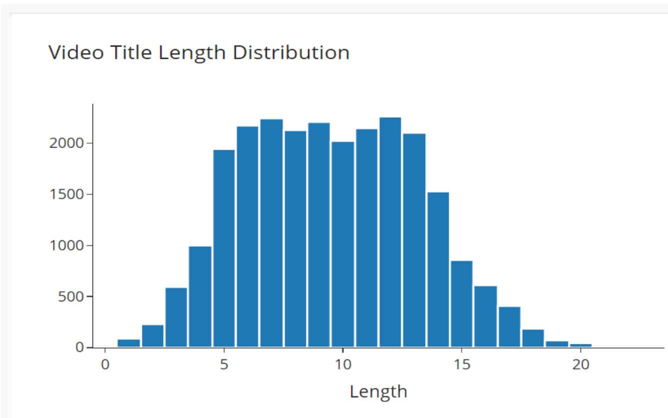

Fig -9: Histogram for video title length distribution

8. Conclusion

Our findings for measuring, assessing, and comparing essential characteristics of YouTube popular videos were reported in this study. Knowing the optimal time to publish a video to YouTube isn't enough to get millions of views and make your video popular. Other elements to consider are good titles, good thumbnails, video SEO, proper tagging, and the number of subscribers, all of which are important in driving views for your material. Understanding these statistics will aid YouTube in not only developing better video processing algorithms, but also in making judgments for individual youtubers.

REFERENCES

[1] Ouyang, S., Li, C. and Li, X. (2016). A Peek Into the Future: Predicting the Popularity of Online Videos. IEEE Access, 4, pp.3026–3033

[2] Hoiles, W., Aprem, A. and Krishnamurthy, V. (2017). Engagement and Popularity Dynamics of YouTube Videos and Sensitivity to Meta-Data. IEEE Transactions on Knowledge and Data Engineering, 29(7), pp.1426–1437

[3] s. Amudha, Niveditha V.R, P S Raja Kumar and Radha Rammohan Shanthanam

[4] “YouTube API Overview”, https://developers.google.com/youtube/v3/getting-started

[5] “Data Wrangling”, https://www.altair.com/what-is-data-wrangling

[6] Wes McKinney (2011). pandas: a Foundational Python Library for Data Analysis and Statistics

[7] “Plotly language support”, https://en.wikipedia.org/wiki/Plotly

[8] “What is Canva”, https://www.tutorialspoint.com/what-is-canva-and-what-are-its-main-features

[9] Timothy Kinsman, Mairieli Wessel, Marco A. Gerosa, Christoph Treude (Mar 2021) How Do Software Developers Use GitHub Actions to Automate Their Workflows? arXiv:2103.12224

[10] PATRIK DANIELSSON, TOM POSTEMA, HUSSAN MUNIR (Jan 2021) Heroku-Based Innovative Platform for Web-Based Deployment in Product Development at Axis

[11] “The cyberattack on Heroku was worse than we thought”, https://www.protocol.com/bulletins/heroku-github-cyberattack-passwords-stolen

[12] Katrien Verbert, Sten Govaerts, Erik Duval, Jose Luis Santos, Frans Van Assche, Gonzalo Parra, Joris Klerkx. Learning Dashboards: An Overview and Future Research Opportunities

[13] Alper Sarikaya, Student Member, IEEE and Michael Gleicher, Scatterplots: Tasks, Data, and Designs

[14] Tracey L. Weissgerber, Natasa M. Milic, Stacey J. Winham, Vesna D. Garovic Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm