Rowdy Rooster

Boya Li, David Moon, EunJeong Heo

Technical Report

[Topic]

Up-To-Date YouTube Top Trending Videos Analysis

: YouTube maintains a list of the top trending videos on the platform. According to Variety magazine, “To determine the year’s top-trending videos, YouTube uses a combination of factors including measuring users’ interactions (number of views, shares, comments and likes). Note that they’re not the most-viewed videos overall for the calendar year”.

[Inspiration]

• Analyze trending video data over the previous one year to discover possible variations between variables across separate English-speaking countries/ regions.

• Analyzing what factors affect how popular a YouTube video will be.

• Sentiment analysis in a variety of forms

• Statistical analysis over time

[Purpose of the ETL]

• Provide up-to-date summary statistics in English-speaking countries/ regions for YouTube creators.

• Enable to track successful factors of trending videos

• Enable to compare the top trending channels/YouTubers followed by video categories

Rowdy Rooster

Boya Li, David Moon, EunJeong Heo

[Extract_Kaggle API]

1) The source of data is extracted from kaggle 'youtube-trending-video-dataset'[1] using kaggle API -- This dataset includes several months (and counting) of data on daily trending YouTube videos. Data included US, GB, CA with up to 200 listed trending videos per day. Each region's data is in a separate file. Data includes the video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count.
    a) ** The original Kaggle API code is attached on ipynb.file, however, due to the Kaggle API's error itself, we initially used the downloaded dataset. We can update the code once the Kaggle API issues are fixed.
2) pgAdmin 4 used to format data for later loading. A database was created and appropriate columns using the query tool.

[Transform_Data Cleanup]

1) The type of transformation needed for this data

   dataframe 1

   - Cleaning & Merging

```
# import file
US = pd.read_csv('Dataset/US_youtube_trending_data.csv')
GB = pd.read_csv('Dataset/GB_youtube_trending_data.csv')
CA = pd.read_csv('Dataset/CA_youtube_trending_data.csv')

US['country'] = 'US'
GB['country'] = 'GB'
CA['country'] = 'CA'
frames = [US, GB, CA]

#merge
df = pd.concat(frames).drop_duplicates()
```

   dataframe 2

[1] https://www.kaggle.com/rsrishav/youtube-trending-video-dataset

Rowdy Rooster

Boya Li, David Moon, EunJeong Heo

- load json file on jupyter notebook

```
with open("CA_category_id.json")as f:
    data = json.load(f)
```

```
data
```

```
{'kind': 'youtube#videoCategoryListResponse',
 'etag': 'kBCr3I9kLHHU79W4Ip5196LDptI',
 'items': [{'kind': 'youtube#videoCategory',
   'etag': 'IfWa37JGcqZs-jZeAyFGkbeh6bc',
   'id': '1',
   'snippet': {'title': 'Film & Animation',
    'assignable': True,
    'channelId': 'UCBR8-60-B28hp2BmDPdntcQ'}},
  {'kind': 'youtube#videoCategory',
   'etag': '5XGylIs7zkjHh5940dsT5862m1Y',
   'id': '2',
   'snippet': {'title': 'Autos & Vehicles',
    'assignable': True,
    'channelId': 'UCBR8-60-B28hp2BmDPdntcQ'}},
  {'kind': 'youtube#videoCategory',
   'etag': 'HCjFMARbBeWjpm6PDfReCOMOZGA',
   'id': '10',
   'snippet': {'title': 'Music',
```

- get particular keys and values (‘id’ and ‘title’) from the dictionaries

Get 'id' and 'title' from dictionary

```
from pprint import pprint
list_of_items = data['items']
```

```
d = []
for i in list_of_items :
    d.append(
        {
            'id' : i['id'],
            'title': i['snippet']['title']
        }
    )

pd.DataFrame(d)
```

- convert dataframe 1 and dataframe 2 to csv file

```
df.to_csv('US_GB_CA_merged.csv')
```

```
df.to_csv('video_category.csv', index=False)
```

[Load _Analysis]

2) The type of final production database to load the data into (relational or non-relational).
relational

Connect to local database

```
engine = create_engine('postgresql+psycopg2://postgres:postgres@localhost/ETL-youtube')
```

```
# confirm tables
engine.table_names()
```

```
df.to_sql(name='youtube', con=engine, if_exists='replace', index=False)
```

```
# Read the table data from SQL
pd.read_sql_query('select * from youtube', con=engine).head()
```

A relational database is a type of database that stores and provides access to data points that are related to one another. The given datasets are intuitive, straightforward in representing data in tables. Therefore, each row in the table is a record with a unique ID called the primary key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

** Characteristic of non-relational database
- Easy to distribute data
- Easy for replication and recovery
- Useful when data needs to be processed at high speed
- It is easy if it needs to be loaded continuously, such as logs, and wants to store large amounts of data.
- since there is no fixed form of schema, it is possible to flexibly change the data model