Automatic Playlist creation

Gopal Ramesh Dahale

August 21, 2020

The Code is divided into roughly 5 sections.

- **Youtube Video Data**: This deals with extracting data from youtube and putting it into a dataframe and writing the data to csv.

- **Data Preprocessing**: This deals with cleaning the data i.e. dropping duplicate rows, formatting dates and time, filling missing values ans z-score normalization.

- **Working with the YoutubeVideoData class**:Data is collected on 18-08-2020 (data1) and 20-08-2020 (data2). The combined data is written to 11840520_data.csv. Both the data is cleaned (df and df2) and then written to 11840520_cleaned_data.csv.

- **Working on score and Playlist Creation**: Creation of a new dataframe(bigdata) which contains intersection of above cleaned data with added columns for change in viewCount, likeCount etc.
  Playlist creation is done by using the Scoring function. It based on the Borda method. It sorts the Video_IDs based on the likeCount,viewCount etc and then applies the Borda method. The scored videos are used to create playlist using the method topVideos() which returns the topmost videos such that the length of a playlist is 7 hours. This is done for each topic.
  The final playlists are displayed for each topic.

- **Data Visualisation**: Describes distribution of viewCount, likeCount, dislikeCount and commentCount for each topic. Top ten videos from each topic are displayed with score. A score vs duration barplot is created for each playlist to determine what is the best duration for video with which a user is comfortable. Field correlations and heatmap are shown for all topics.
  Some other visualisation like 'wordcloud' shown (for fun).

- **Which APIs did you use?**
  YouTube Data API v3

- **Which filters did you use?**
  To search for youtube videos, part = 'snippet' and type = 'video' are used in youtube.search.list(). 'PageToken' is also used to get the next 50 videos.
  To get statistics (views, likes, dislikes, commentCount,favoriteCount, duration, published date) of videos, part = 'statistics' and part = 'contentDetails' are used youtube.videos.list().

- **What were the challenges you faced in this assignment?**
  Biggest challenge was of learning and implementing the scoring function. Working with pandas dataframe and writing data to csv was quite tricky for me the first time. Writing so much code without any stratergy made my code a mess but then I used OOP which gave more clarity to the code.
  Data visualisation for another tough part for me because initially I didn't knew what to visualise and why to visualise.

- **What did you learn from this data analysis exercise?**
  I learned how to use API's and get the data from youtube. Learned about the usage of dataframe and csv together. Improved my understanding over the Borda's method as I have implemented it. Data visualisation helped to gain an understanding of how the playlist look and the distribution of score helped to describe what should be a good duration of a video to have a good score in the playlist. WordCloud is a fun feature of seaborn and helps to see what words are present that are closely related to the topic.

References:

- Kaggle notebook for trending youtube video metadata analysis

- For visualisation

- For using seaborn