CSC302 A2

Zixiao Ren, Qiming(Austin) Ye

Features

Features completed in A3

1. Playlist title keyword recommendation: Find a recommended list of keywords for playlist title (based on high-frequency keywords in the title of previous playlists)
2. Playlist description keyword recommendation: Find a recommended list of keywords for the description of playlist description (based on high-frequency keywords in the description of previous playlists)
3. Video title keyword recommendation page: Find a recommended list of keywords for the video title (calculate the total score of each keyword by summing up the favorite count, like count, view count, and comment count of all the videos that include the keyword in the titles. Sort keywords based on the total score)
4. Video description keyword recommendation page: Find a recommended list of keywords in the description of popular videos (calculate the total score of each keyword by summing up the favorite count, like count, view count, and comment count of all the videos that include the keyword in the descriptions. Sort keywords based on the total score)
5. Add-on feature:
    a. Sort the playlist table by video count and publish time
    b. Sort the video list table by publishing time, view count, like count, and comment count

Tasks:

- Features 1 and 2
    - Due date: Dec 2nd, 2022
    - Acceptance criteria: After clicking on the playlist button on the channel page, the user should be able to see two recommendation lists, one for the title and one for the description. Each list should include the most recommended 20 keywords and their frequencies that appeared in the previous playlist titles/descriptions.
    - Zixiao Ren: works on the recommendation algorithm
    - Qiming(Austin): works on the frontend and the recommendation algorithm
- Features 3 and 4
    - Due date: Dec 2nd, 2022
    - Acceptance criteria: After clicking on the video list button on the channel page, the user should be able to see two recommendation lists, one for the title and one for the description. Each list should include the most recommended 20 keywords and their total scores that appeared in the previous video titles/descriptions.

    - Zixiao Ren: works on the recommendation algorithm
    - Qiming(Austin): works on the frontend
- Feature 5
    - Due date: Dec 2nd, 2022
    - Acceptance criteria: The table content should be sorted based on the user's choice from the dropdown menu.
    - Qiming(Austin): works on the fronend

Note that the frontend design for features 1-4 was changed. The original design in A2 was that the keyword table and the info table should be shown on separate pages. There should be several buttons for redirections on the channel page. However, we found that it is not necessary to split these tables into different pages. If we add the keyword tables to the playlist and video list info pages, it would not only be easier for the user to find all the information all at once but also reduce the complexity of our code.

The recommendation algorithm design was also changed. The original idea was that we use the same way as how we calculate the keyword for the video title/description to calculate the playlist keyword. However, we found that the Youtube API does not allow us to get all the video information within a playlist directly. This means that if we want to know how popular a playlist is based on the popularity of the videos of that playlist, the number of times we call the Youtube API would exponentially increase. Hence, we decided to calculate the keyword based on the frequency at which it appeared in the previous playlist title/description.

Progress Demonstration

The frontend of features 1-4 is almost the same. The recommendation algorithm for the playlist keywords is also easy to be done since it only calculates the frequency of the keyword that appeared in the previous playlist titles/descriptions. Hence to balance the workload, these tasks plus the add-on feature were assigned to Qiming, and the recommendation algorithm for the video keywords was assigned to Zixiao. All the milestones were finished before Dec 2nd. The final wrap-ups including manual and automated testing and cloud deployment on Heroku were also finished on Dec 2nd. However, since we both had heavy workloads at the end of the semester, we did not have time for the documentation. We asked for an extension to work on the documentation after we finish all the final exams, which is Dec 15th.

Blameless Retrospective and Possible Future Developments

- Better frontend GUI design is needed
    - The dropdown menu is not well designed. It would be better to use the MUI library to build the dropdown menu.

- Find out a more static way to calculate and present data
    - It takes too long to load the data from Youtube API due to the large data set. This would lead the user to wait too long for the result andlead to an “NaN” issue when too much data to handle.
- Find out a way to make the project less depends on the Youtube API.
    - Each API Key can only make limit number of API calls, so we have to change the API Key manually if the number of API calls reaches the limit. It would be better if we can find a way to make our project less depends on Youtube APIs.