Approach/Assumptions and APIs:

1. Make Youtube API calls using the youtube ‘search()’ API (.search().list(q='subject',part='snippet',type='video',maxResults=50)) to get search results - Title, Video ID, PublishedAt for a particular Topic/subject and storing them in lists. Also, store the Time of the Query in another list.
2. Then use the youtube 'videos()' API (.videos().list(part=["statistics","contentDetails"],id=VideoIDs)) to get the statistics (likes, dislikes, views) and Duration of video using the video IDs that I had saved in lists. Store these new values in list as well.
3. I filled ‘-1’ in the like/dislike count of videos whose ike/dislike counts were hidden.
4. I stored the search results for each of the 3 subjects in files- subject1.csv, subject2.csv and subject3.csv after the first search and then into subject1_2.csv, subject2_2.csv and subject3_2.csv for the second search (on another day)
5. Then, I removed the duplicate values (same video results) from these individual files as well as the rows with Like/Dislike count == -1 (hidden likes)
6. Created one Data Frame each for individual subjects for video ids that were common in the search results at different days and calculated the increase/decrease in the videos’ views, likes, dislikes.
7. Then I converted the Video Duration from the youtube format into hours, added z-score columns for the numerical value columns
8. Then I calculated the Scores according to my scoring function using the z-scores.
9. Ranked the subjects in descending order of their scores and created individual playlists of max 10 hours for each subject based on the best scores.

Challenges Faced and what I learnt from the Assignment:

1. The code that I had inititally written didn’t take into account the videos whose like/ dislike counts were hidden. This caused me some errors and took quite some time to figure out the problem.
2. The youtube search() API didn’t return more than 50 results at a time so I had to switch to the next pages using the ‘nextPageToken’.
3. I faced the most problems in handling the data, taking into account the different possible values of the fields, converting them into readable format and debugging small errors that arose from the process. Like, I couldn’t find out why my code gave an error when everything was typed correctly. It took me some time to realize that the error was not in the code but rather in my assumption that there should be no duplicate search results returned by the youtube search() API.
4. The most important thing I learnt from doing this assignment is that I should first make a rough higher level outline of the things to be done, keeping a note of the most important things to be analysed before starting to type the code so that I can avoid changing the code which I had typed without proper planning.