# CSC 455 - Project Proposal (R3)
# Topic: Data Analysis on YouTube Videos & Twitter

Prem Subudi, Yuan Zhang, Amiya Renavikar

**Description:**

The trending hashtag on YouTube is a good way to keep up with updates and societal changes and also reveal one's popularity or status in social network. We propose a hypothesis about a person's popularity from trend of YouTube videos.

For this project, we are going to gather the datasets for the name, uploads, number of views, and channel type for the currently trending YouTube artists. Also, we will collect the data for the number of Twitter followers. Then we will perform data analysis over those datasets to estimate the popularity of the artists and generate the rateScore and viewScore as a general metric. Using this analysis, we are going to support and verify our hypothesis.

**Hypothesis:**

If a user is popular in YouTube, then she/he will most likely to have most followers on twitter.

**Methods to get real time data:**

1. **Web Crawler**: This is for collecting information from any webpage. Firstly, we understand how the html is implemented. Then we use python's beautifulsoup package to open the url and select elements containing data that we need in html. And then use regular expression parser to get information that we need.
2. **With Youtube API:** There is YouTube API that implemented by google developers. The first step is to use create Oauth account and get 'client_secrest.json' file from: https://console.developers.google.com/apis/credentials With this file, we can use youtube api to get data. By sending get request based on code provided by YouTube API, we can get number of views, likes or dislikes and all other content for the YouTube video.

**Implementation of hypothesis:**

a) **Explanation:** If a celebrity is famous on one social media, then they are likely to be searched and followed on different social media platform by viewers who have access to multiple channels or social networks . This way a popular celebrity will have more viewers and followers and also have higher status in the social network as well.

b) **Steps:**

*I.* *Import 5000 youtubers with its twitter url and youtuber url:* https://socialblade.com/youtube/top/5000/mostsubscribed contains profile link and name of 5000 youtubers, 'getYoutubers.py' file collects each youtuber's name and his/her

socialblade profile url. Then the program will go to each profile url to get its youtube link, twitter link in 'youtubers_update.csv' file.

II. *Data cleaning of 'youtubers_update_1.csv':*
'datacleaning1.py' deletes rows that contain missing value in twitterlink(twitter's url) And updated file is 'youtubers_update_1.csv'. Then 'datacleaning2.py' deletes the rows that have zero number of views (invalid value). And then the table will updated to 'youtubers_update_2.csv' file. With these steps, the number of rows decreases to 2446.

III. *Get all videos of each artist with the number of likes and dislikes.*
With the YouTube url in 'youtubers_update_2.csv' file, we can get the channel id for each channel. In 'getvideoinformation.py' file, we use suffix of the youtube url to get its channel id and then use youtube api to get the playlist id for each channel. With the playlistid, we can see all videos, and like or dislike information from youtube api. But unfortunately, we can only get maximum 50 videos for each video because it is restricted by youtube api. With these information, we can create files for each channels that contain the like and dislike count for each video. Then we generate an '(index).csv' file in list folder

IV. *Get number of followers of each user from Twitter*
In 'youtubers_update_2.csv' file, there are twitter url links for each youtuber. In getfollowers.py, we iterate through twitter and use web crawler approach to get follower information. And then we update the table to 'youtbers_update_3.csv'.

V. *Calculate the popularity of each youtuber's videos*
For each of the videos, we will use number of likes, dislikes, views and ratings as factors to calculate its popularity. We assume that better feedback means the video is popular. The algorithm we are going to use is Wilson score algorithm of normal distribution (detailed explanation in data analysis section). This algorithm will comprehensively evaluate the likes and number of voting. This is the algorithm:

$$n = u + v$$

$$p = u/n$$

$$S = (p + \frac{z_\alpha^2}{2n} - \frac{z_\alpha}{2n}\sqrt{4n(1-p)p + z_\alpha^2}) \,/\, (1 + \frac{z_\alpha^2}{n})$$

**u** is for like(positive feedback)
**v** is for dislike(negative feedback)

**n** represents total interactions(likes + dislikes)
**p** is proportion of positive feedback and
**z** is quartile of this normal distribution
**S** is the popularity of this video.
After getting the result, we insert it into corresponding row of .csv file.
*Reference: Wilson, E. B. (1927), "Probable Inference, the Law of Succession, and Statistical Inference," Journal of the American Statistical Association, 22, 209-212.*

VI. *Update the tables by implementing scores for each user*
In 'FillViewScore.py', by iterating all files in list folder, I use wilson score algorithm to get the rate score for each video and calculate the its average for rateScore. And then use wilson normal distribution algorithm to get video score. I update these rateScore and videoScore columns for each user in 'youtubers_update_4.csv''.

VII. *Data cleaning for missing value in twitter followers*
In 'dataclearning3.py', I delete all rows with missing values in followers and update it to 'youtubers_update_5.csv' file. In 'youtubers_update_5.csv', we have 2270 rows.

**Data Analysis:**

**Enhancing Wilson's normal distribution Algorithm (Updated by pksubedi)**

**Part 1 - Why Wilson Algorithm of normal distribution? (pksubedi)**

Wilson Algorithm is very important for calculating the random and uneven distribution of data. For example the YouTube artists have uneven number of likes and dislikes as well as their followers. In order to use the uneven or some range of values for the large distribution of data, we have to use the lower bound of scores, upper bound of scores and confidence interval. Specifically in this situation:- lower & upper bound of the number of likes, dislikes, and so on. For instance, I used the top 25 percentile of the values of likes, dislikes, and followers, to analyze the hypothesis. So this type of clustering idea is only supported by the Wilson's algorithm of normal distribution. Hence, I find this algorithm the best fit for the data analysis of this project.

**Data Analysis 1 - Details (pksubedi)**

In order to perform the data analysis to figure out the consistency of the user's popularity in two different social media sites, I grab the twitter user with the maximum number of followers among the list of users from the provided dataset (youtuber_update_5.csv) file using python built in function. Similarly, I grab the maximum rate score and view score for the youtube artist or user. Then I extract the top 25% of the youtube users with maximum scores (both view and rate scores) and top 25% of the twitter users with maximum followers and stored them into the list/group. Then I analyze those two lists to see if there is/are any commonality of the user(s). And based on the result given by my program (DataAnalysis.py), it is true that the users who fall

within the top 25 percentile by the popularity in youtube also fall within the same percentile range in twitter group based on the number of their followers. Hence this proves our hypothesis, "If an artist is popular in YouTube, then she/he will most likely to have most followers on twitter."

**Data Analysis 2 - (Yuan)**

I am using k-means algorithm analyze the data, which helps to cluster group with similar value for ratescore and followers. The reason that I do not use view scores to evaluate is because they are too small and there is not much discrepancies or the difference in the different values.

Steps:

As I mentioned above, I have implemented k-means in 'DataAnalysis2.py.' After reading data in youtubers_update_5.csv, I rebuild the dataset to make it only have information of rate-scores and the number of followers. And with this new data set, I use *sklearn* package to generate k-mean cluster with list of cluster center. And then I use four variables for each quartiles number of rate-score. And then I have created four lists. If the cluster's center is located in one of these four numbers, it will be appended to its corresponding list. And then I used another four variables for each quartiles number of followers. And then calculate how many cluster center are located in each score-rate list within each range. I try to set number of cluster to 10, 100, 200(I did not make it higher because we only have around 2000 data set), The result is shown below:

```
I cluster the data set into 10 groups
there are 3 clusters in 0-25% of rateScore, 33% in 0-25% of number of followers
there are 7 clusters in 25-50% of rateScore, 0% in 25-50% of number of followers
there are 0 clusters in 50-75% of rateScore, 0% in 50-75% of number of followers
there are 10 clusters in 75-100% of rateScore, 90% in 75-100% of number of followers
```

```
I cluster the data set into 100 groups
there are 20 clusters in 0-25% of rateScore, 5% in 0-25% of number of followers
there are 56 clusters in 25-50% of rateScore, 3% in 25-50% of number of followers
there are 15 clusters in 50-75% of rateScore, 0% in 50-75% of number of followers
there are 85 clusters in 75-100% of rateScore, 84% in 75-100% of number of followers
```

```
I cluster the data set into 200 groups
there are 46 clusters in 0-25% of rateScore, 19% in 0-25% of number of followers
there are 92 clusters in 25-50% of rateScore, 6% in 25-50% of number of followers
there are 39 clusters in 50-75% of rateScore, 17% in 50-75% of number of followers
there are 161 clusters in 75-100% of rateScore, 80% in 75-100% of number of followers
```

**Conclusion:**

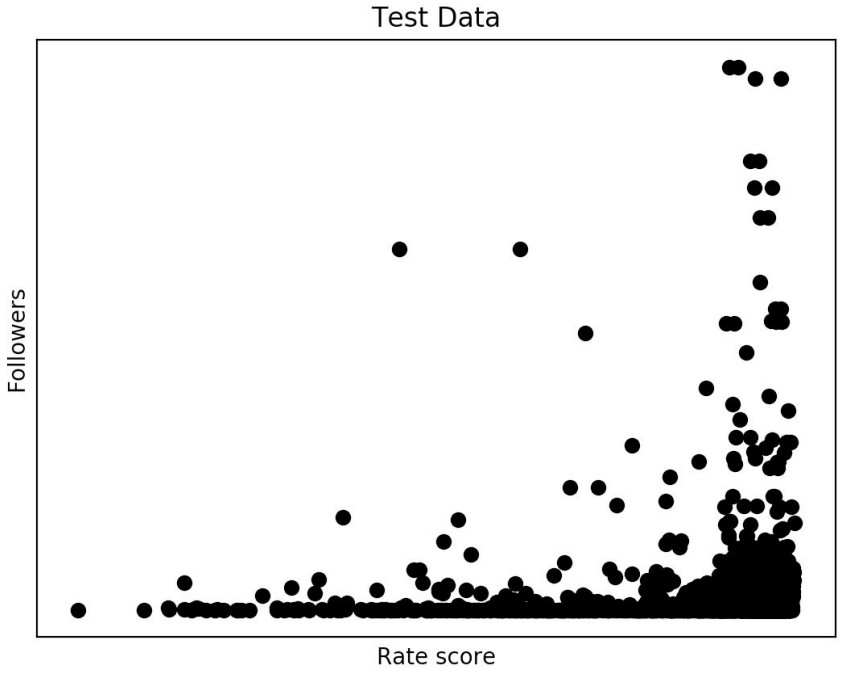
From the result, I find that most cluster tend to set to 75-100% of rate score. It preliminarily testified that those popular youtubers(with higher score) are tend to group together because they have similar number of followers. Consequently I came to the result that, at least

80% of the cluster of popular youtubers also popular in twitter because they are located in 75-100% high range of number of follower. 80% is a strong indication of that if a user is popular in youtube, he/she is likely to have most followers in twitter.

**Data Analysis 3 - Linear Regression method (Amiya)**

Linear Regression is useful for finding a relationship between two continuous variables (one independent and other dependent). Using this method, I am able to establish a statistical and deterministic relationship between our two variables - twitter followers and youtube rate score. I am not using view Scores since they have minor discrepancies and do not tell me much about a user's liking. I use the 'youtube_update_5.csv' dataset to obtain information about the two variables. Many artists have been observed and their rate score on Youtube and number of Twitter followers have been recorded. The goal is to design a model that can reflect the high range of twitter followers, given the rate score on Youtube. Using the given data, I obtain a linear model.

I estimate that the range of twitters followers should be consistent with the rate score as seen in the graph below. The function for the line of regression is plotted by the program itself. In the plot, the trend seems to be consistent as predicted for both variables. Both variables both fall near the line of regression in the bottom right of the image. From this statistical relationship, we can conclude that there is a strong correlation between the number of twitter followers and the rate score of the Youtuber.



**Special technical challenges:**

- We had trouble with implementing the Twitter API at first because we decided to implement it using the API provided by the Twitter developers website. However, we found that it was restricted to a maximum number of 200 follower details to be displayed as output, hence we switched to using REST APIs with Python.
- There were some discrepancies between the results for the Twitter API since our output file was initially using twitter handle to make the GET requests. Out of 2000+ users, the output csv file was only printing out 700 lines of data or so. Hence, we changed our implementation to use the user's Twitter URL and the issue was fixed.
- The challenges part of web crawler is that some web data are not well formatted. When I got its each profile url in socialblade, some of them need do another search steps to get to real url, while some of them did not need to. So I used some if else statements for handling those invalid cases. The same thing happened when I did web crawler in twitter. Some of them put the number of followers with different selectors. So I also need to write codes for handling these cases. It cost me a lot of time on running them again and again for finding out those special cases.
- When I implemented data analysis of k-means methods, I tried to plot them out with a fancy picture. But the index when iterating each row is matching predict one. But when I print the length, they are all same. I still can't plot the cluster despite it is not relative to my analysis and result.
- For the linear regression implementation, I had trouble generating the line of regression so I used functions to calculate it.

**Takeaways:**

From this project, we demonstrate the concept of constructing a 'good' argument in which our premise provides good reasons to be true. We have accomplished this by generating our hypothesis and proving it to be true from the results of our two data analysis methods. We have also understood that the likelihood of an individual being popular on multiple social media platforms is high if they have already established themselves on one platform. In terms of Social Computing, we believe that the data analysis performed in this project has added to the knowledge of social media analytics. The knowledge of users' behavior could be beneficial in implementing a potential marketing strategy for both the social media platforms. This project also helps the reader to populate the ideas of argumentation based on the data found in the internet or media channels.

**References:**

https://www.jianshu.com/p/4d2b45918958
http://www.evanmiller.org/how-not-to-sort-by-average-rating.htm