Data Collection

The aim of this project is to build a model than can predict the number of views a YouTuber will have depending on his/her presence on other social media platforms such as Instagram and Twitter. Compiling data was the hardest part since no such dataset is available online. A YouTuber example "TheEllenShow" has 10,990 uploads, 35.3 millions subscribers and 17,939,915,162 views on YouTube. TheEllenShow also has an Instagram account with 17,939,915,162 followers, 383 following and 8172 posts with includes pictures and videos. On Twitter, TheEllenShow has an account named "Ellen DeGeneres' with 79050757 followers, 20621 tweets and 1880 likes.

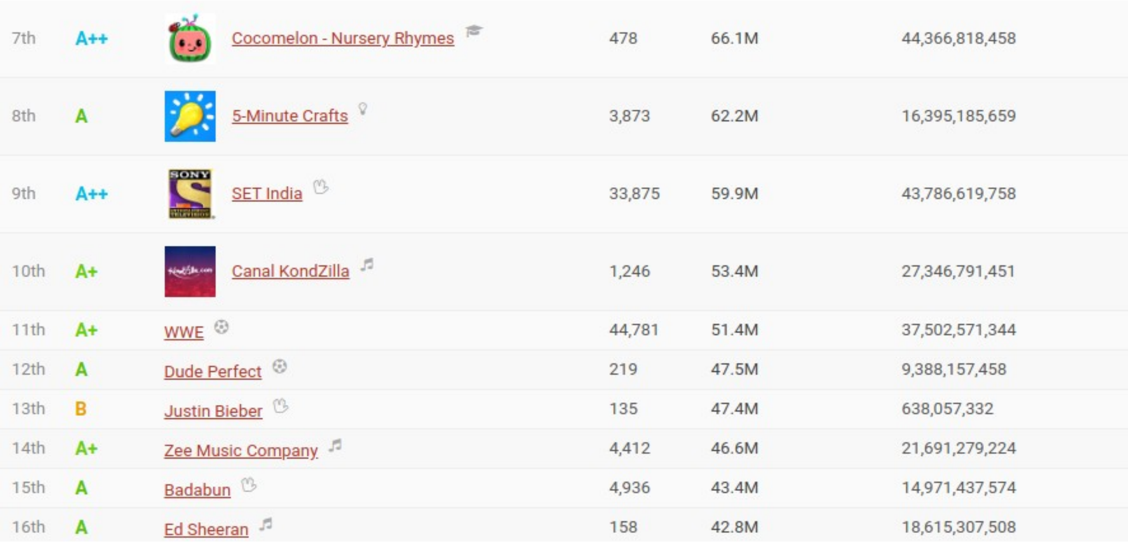
This information is represented in the Dataset.csv file as followed:

TheEllenShow;10,990;35200000.0;17,939,915,162;79800000.0;383;8,172;79050757;20621;1880

1. YouTube Username
2. Number of uploads on YouTube
3. Number of subscribers on YouTube
4. Total number of views of YouTube
5. Number of followers on Instagram
6. Number of following by user on Instagram
7. Number of posts of Instagram
8. Number of followers on Twitter
9. Number of tweets on Twitter
10. Number of likes on Twitter

Manually selecting YouTube accounts and looking their perspective Instagram and Twitter accounts would be too tedious and time consuming. Ideally, we want to analyze the most successful YouTubers, that are channels with the highest number of subscribers. *https://socialblade.com/* is a realtime database of social media platforms that contains the top 5000 most subscribed YouTube channels. We used this as the base for our dataset.

Pulling data from Socialblade



7th	A++	Cocomelon - Nursery Rhymes	478	66.1M	44,366,818,458
8th	A	5-Minute Crafts	3,873	62.2M	16,395,185,659
9th	A++	SET India	33,875	59.9M	43,786,619,758
10th	A+	Canal KondZilla	1,246	53.4M	27,346,791,451
11th	A+	WWE	44,781	51.4M	37,502,571,344
12th	A	Dude Perfect	219	47.5M	9,388,157,458
13th	B	Justin Bieber	135	47.4M	638,057,332
14th	A+	Zee Music Company	4,412	46.6M	21,691,279,224
15th	A	Badabun	4,936	43.4M	14,971,437,574
16th	A	Ed Sheeran	158	42.8M	18,615,307,508

Using the python library BeautifulSoup, we were able to extract the following information: the channel's name, number of uploads, subscribers count and view counts as shown in the picture above. This was done by extracting the appropriate 'div' tags and 'attrs' styles from the HTML response the python Request received from Socialblade.

Next step is for each of these accounts, we want to find their corresponding Instagram and Twitter if any. One challenge is finding the ID for each channel. For example, from the above picture, "Cocomelon - Nursery Rhymes" is a channel's name and is not an id/primary key. In order to locate the "about page" which contains links for other social media accounts, we need the id. For "Cocomelon - Nursery Rhymes", its channelId is UCbCmjCuTUZos6Inko4u57UQ. The URL for the about page would be: https://www.youtube.com/channel/UCbCmjCuTUZos6Inko4u57UQ/about

Finding the channelid

We can find the channedId using the YouTube data API [1].

```
def get_channel_username_API(name):
    os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1"
```

```
 api_service_name = "youtube" api_version = "v3"
client_secrets_file = "secret.json" # Get credentials and create an API client flow =
google_auth_oauthlib.flow.InstalledAppFlow.from_client_secrets_file(
 client_secrets_file, scopes)
credentials = flow.run_console()
youtube = googleapiclient.discovery.build(
api_service_name, api_version, credentials=credentials)
request = youtube.search().list(
    part="snippet",
    q=name,
    type="channel",
    maxResults=1,
)
response = request.execute()
print(response)
```

This returns a json object and channelId can be found in items['channelId'].

In the about page, we can use BeautifulSoup to grab the page content and scrape for all links with 'a' tags and href =True to look for the keywords '*instagram.com%2F*' and '*twitter.com%2F*'. The username will be the word that follows. Example:

```
<a class="yt-simple-endpoint style-scope ytd-channel-about-metadata-renderer" href="/redirect?event=channel_description&amp;q=http%3A%2F%2Finstagram.com%2Ftheellenshow&amp;SRe9vx9uZBh8MTU3NDk3MTA0MEAxNTc0ODg0NjQw">
```

The Instagram username is theellenshow.

Using these steps, we can build a *username_dataset* with all channels that have both an Instagram and Twitter account. *username_dataset* contains YouTube username, Instagram username and Twitter username as records. Note: It is not possible to get usernames/links for other social media accounts directly via the YouTube data API. These links can only be retrieved from the about page.

Getting Information from Instagram

Extracting the number of post,followers and following from Instagram proved to be quite challenging. Instagram prevents users from constantly sending requests by limiting replies and preventing web scrapping.

First Attempt:

By adding the /?__a=1 keyword after the URL, Instagram would return a json file which contains the information we need.

Example: https://www.instagram.com/theellenshow/?__a=1 would return the relevant information as shown below:

```
{"logging_page_id":"profilePage_18918467"," ... alse,"graphql":{"user": ... "edge_followed_by":
{"count":80035565}, "followed_by_viewer": false," edge_follow":{"count":384}, ...
edge_owner_to_timeline_media":{"count":8189,"page_info" ... }
```

where:

number of followers would be: json_reply[‘graphql’][‘user’][‘edge_followed_by’][‘count’]
number of following would be: json_reply[‘graphql’][‘user’][‘edge_follow’][‘count’]
number of posts would be: json_reply[‘graphql’][‘user’][‘edge_owner_to_timeline_media’][‘count’]

This was simplified by removing unnecessary data.

While this technique worked, we could only retrieved around 150 results before getting blocked resulting in a none return from the python request function.

As mentioned above, it is also impossible to scrape the website using BeautifulSoup without being blocked.

Second Attempt

We referred to “*https://www.benlcollins.com/spreadsheets/import-social-media-statistics/*” that uses a google sheet to extract information from the XML. The following formula :

```
=IMPORTXML(<Cell_with_URL>,"//meta[@name='description']/@content")
```

with URL https://www.instagram.com/theellenshow/ would output: "79.8m Followers, 383 Following, 8,172 Posts - See Instagram photos and videos from Ellen DeGeneres(@theellenshow)". From this, we could extract the number of followers, following and posts and build a dataset for Instagram. It took over 4 days to build this dataset since we were constantly getting blocked and had to swap in between the two methods each time.

Getting Information from Twitter

Twitter allows for web scrapping using BeautifulSoup. By making a python get request for the URL: ‘https://twitter.com/<username>’ we were able to access the html content and extract information as shown in the next page. We repeated this process for each twitter account and created the Twitter dataset. Generating this dataset took around 3 hours for approx. 2000 usernames.

Final step was merging all the datasets to create the Dataset.csv file using *username_dataset* to link them. This step is fairly straight forward.

```
    For each record in Instagram_dataset
    Youtube_username =  Find from username_dataset using Instagram_username
    Twitter_username = Find form  username_dataset using Instagram_username
    Youtube_Record = Find from youtube_dataset using Youtube_username
    Twitter_record = Find from twitter_dataset using Twitter_username
    Merge Instagram_record, Youtube_Record and Twitter_Record
```

The final dataset has around 1200 records.

Note: We also tried including Facebook in our model, but it is impossible to scrape Facebook pages and we were unable to secure a development access key from Facebook.

```
def get_twitter_info(username):
    temp = requests.get('https://twitter.com/'+username)
    bs = BeautifulSoup(temp.text,'lxml')
    try:
         follow_box = bs.find('li',{'class':'ProfileNav-item ProfileNav-item--followers'})
        tweets_box = bs.find('li',{'class':'ProfileNav-item ProfileNav-item--tweets is-active'})
        likes_box = bs.find('li',{'class':'ProfileNav-item ProfileNav-item--favorites'})
        num_followers = follow_box.find('a').find('span',{'class':'ProfileNav-value'}).get('data-count')
        num_tweets = tweets_box.find('a').find('span',{'class':'ProfileNav-value'}).get('data-count')
        num_likes = likes_box.find('a').find('span',{'class':'ProfileNav-value'}).get('data-count')
        return num_followers,num_tweets,num_likes
    except Exceptions as e:
        print(str(e))
```

References

[1] Search YouTube Data API Google Developers. (n.d.). Retrieved November 27, 2019, from https://developers.google.com/youtube/v3/docs/search.