Movie review prediction using twitter data

Ravi Teja Kakara(SUID - 328347031)

Prasanth Chakravarthy Gajula(SUID - 250389453)

Manideep Ramineni(SUID - 393059154)

Sri Satya Praveen Gatti(SUID - 375865092)

Ramireddy Singareddy(SUID - 532326351)

Ratna Manikanta Ajay Kumar Kakarala(SUID - 814378982)

Social Media and Data Mining (CIS - 600)

Syracuse University

New York

May 2021

Table Of Contents

Abstract

Movies are undoubtedly one of the major entertainments to the people. Generally, people glance at the movie reviews and decide whether to watch a particular movie or not regardless of its genre. Social Media has been a part of human life and millions of people are expressing their views, opinions, likes, dislikes and sharing their joy with others through social media. So, we came up with an idea of predicting the review of a movie by considering thousands of valuable views regarding a movie and predicting it's review. Prediction of movie review by analyzing user sentiments expressed through tweets on the Twitter platform. This is done by classifying tweets into positive, negative, and neutral responses by performing Sentiment Analysis using Vader. Finally, we predict the degree of opinion on the movie and visualize it based on various factors such as age, region, gender etc. These features are integrated into a web application.

Keywords: Movie, Twitter, Sentiment Analysis, Vader, web application.

1.Introduction:

Movies started entertaining people all over the world in the early 1800's. Soon they became one of the most loved and popular entertainments. A movie has many stages starting from bringing it on to sets to releasing it in movie theatres. This is where people enjoy their movies. One important aspect that affects a movie is movie review. Most of the people are wary about watching a movie regardless of its cast, crew, genre, and ratings. So, they go through reviews of movies and decide on whether to watch a movie or not.

Moreover, Social Media which started in the late 1900s changed the way we see the world. A statement or news can be known throughout the world within no time through social media. Social Media has made the world a better place by increasing the reachability of news and people's voices. People started being active over various social media and expressed their feelings such as happiness, joy, support, and rejections through social media like Facebook, Twitter, Instagram, LinkedIn etc.

So, we thought of why we shouldn't make use of this mass pool of generic people's opinions and views over social media and predict the review of a movie. When a movie is released, people from all over the world talk about the movie over social media like Twitter. We chose to do so because a collective people review might be more authentic than a small group of people reviews.

We collected tweets from Twitter regarding a movie for which the user would like to know the review and perform Sentiment Analysis to predict the review of a movie.

2. Prior Work:

The paper “Survey on mining subjective data on the web” [1], was used to understand the different Sentiment Analysis techniques available to our disposal. Once we found techniques that are most suitable to our application, we researched each of them in detail and we will cite the most critical papers, webpages and journals in this subsection.

1.The website “Using VADER to handle sentiment analysis with social media text”[2]

2. The webpage, “Simplifying Sentiment Analysis using VADER in Python”[3] provided a lot of fundamental insights on how VADER performs sentiment analysis and how it can be used on Social Media Text.

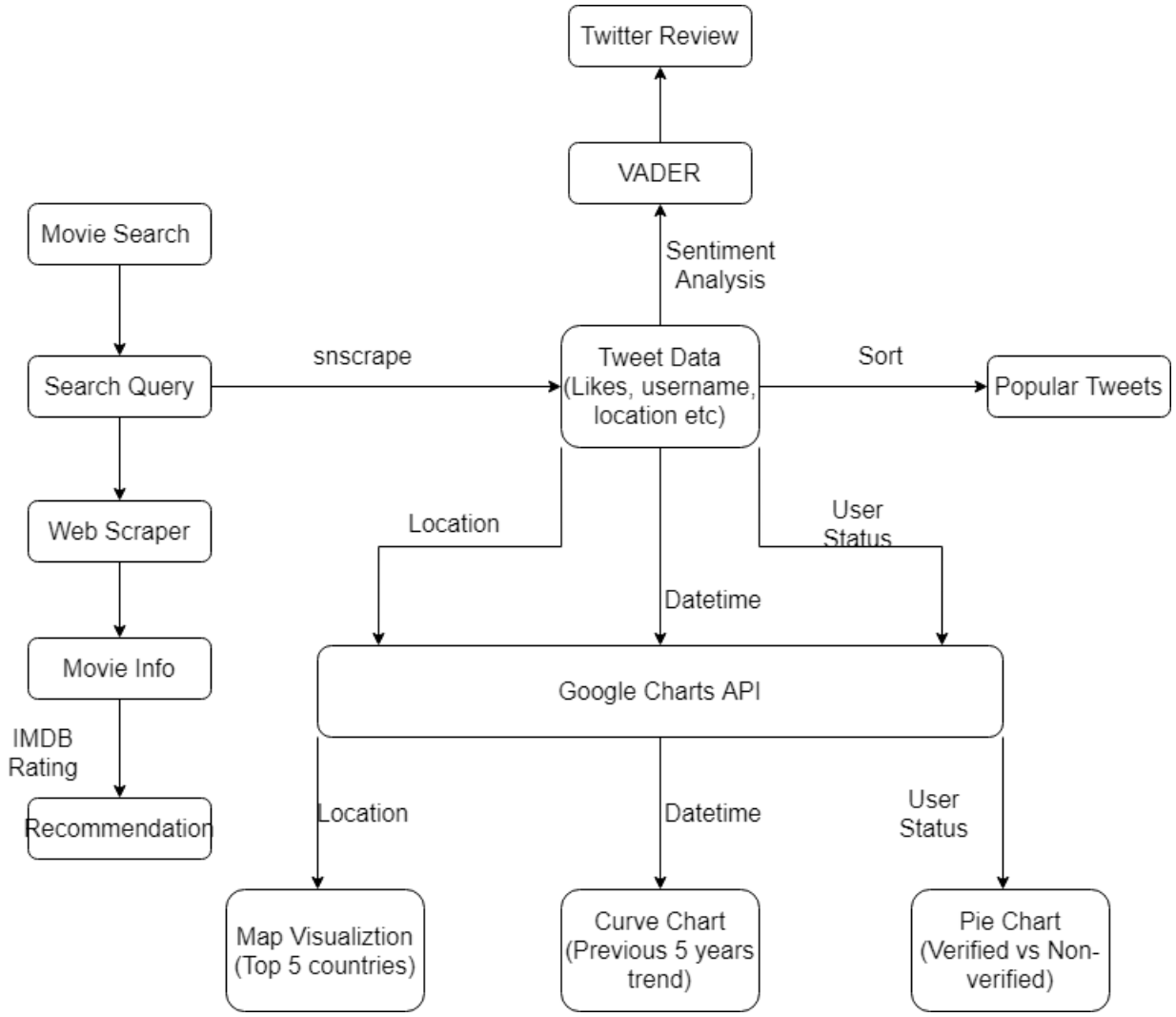
3.By using the paper[4] we could analyse the reviews of customers on various movies by implementing three algorithms namely K Nearest Neighbours, Logistic Regression and Naive Bayes and provide conclusive remarks.

4. The paper [5] enabled us to describe the development, validation, and evaluation of VADER (for Valence Aware Dictionary for sEntiment Reasoning) by using a combination of qualitative and quantitative methods to produce, and then empirically validate, a gold-standard sentiment lexicon that is especially attuned to microblog-like contexts.

5. The website Word Clouds and the values of simple visualisations helped us to build word clouds from the tweets extracted and identify the most used words in those tweets. This will help us to identify the most common words used by the twitter users for a particular movie.

6. The website https://github.com/JustAnotherArchivist/snscrape enabled us to acquire some knowledge about snscrape. By using it we scrape things like user profiles, hashtags, or searches e.g. the relevant posts.

3. Architecture:



An interactive web page is developed which allows the user to search for a movie of his choice. As soon as the user enters a movie name, the webpage fetches the information regarding the movie like cast and crew, ratings, etc., using web scraping. We recommend the user whether to watch a movie or not based on IMDB rating. On the other hand, we fetch tweets related to the movie using snscrape from Twitter. Furthermore, we are showing different visualizations based on regions from which the tweets were tweeted, trend of tweets over a span of 5 years and user's status.

4. Data:

4.1 Data Collection:

This project includes two kinds of data. One is data that contains all the information related to the movie. The other data is the tweets related to the movie. These two kinds of data are collected using two techniques namely Web Scraping[6] and snscrape[7]. Let us discuss these techniques.

4.1.1 Web Scraping:

We are using data scraping used for extracting data from websites. The web scraping software might directly access the World Wide Web using HTTP or a browser. This is generally an automated process and implemented using a bot or web crawler.

As soon as the user enters a movie name along with the date, we are using a web scraper to get the data related to the movie. We scraped the data from a website called “reelgood.com” dynamically. We scraped data like movie description, cast and crew, IMDB rating, genre, trailer link and displayed it on our web page.

Requests:

Request is a library built on python. The Requests library allows us to send HTTP requests extremely easily. It doesn’t require the user to manually add the search string to the URLs, or to form-encode the PUT and POST data. Using the requests library, we downloaded the HTML code at the URL after the search string is appended to the URL.

Requirements

requests officially support Python 2.7 & 3.5 or higher.

Installation

Command to install snscrape:

```
python -m pip install requests
```

bs4:

bs4 is popularly known as Beautiful Soup. Beautiful Soup is a library which is also built on python that makes it easy to scrape information from web pages. It sits atop of HTML or XML

parser providing pythonic idioms for iterating, searching, and modifying the parse tree. bs4 organizes the downloaded HTML code using requests library by preventing name squatting.

Requirements

requests officially support Python 3 or higher.

Installation

Command to install snscrape:

```
pip install bs4
```

4.1.2 snscrape

snscrape is a scraper for social networking services (SNS). It scrapes things like user profiles, hashtags, or searches and returns the discovered items, e.g. the relevant posts. some features listed here may only be available in the current development version of snscrape.

The following services are currently supported:

- Facebook: user profiles, groups, and communities (aka visitor posts)
- Instagram: user profiles, hashtags, and locations
- Reddit: users, subreddits, and searches (via Pushshift)
- Telegram: channels
- Twitter: users, user profiles, hashtags, searches, threads, and list posts

Requirements

snscrape requires Python 3.8 or higher. The Python package dependencies are installed automatically when you install snscrape. one of the dependencies, lxml, also requires libxml2 and libxslt to be installed.

Installation

Command to install snscrape pip3 install snscrape.

Usage

To get all tweets by Jason Scott (@textfiles) we use

snscrape twitter-user textfiles

To redirect the output to a file for further processing, e.g. in bash using the filename twitter-@textfiles we use

snscrape twitter-user textfiles >twitter-@textfiles

To get the latest 100 tweets with the hashtag #archiveteam:

snscrape --max-results 100 twitter-hashtag archiveteam

4.2 Data Preprocessing:

Sorting the fields of dataset:

The dataset that we got after scraping Twitter using snscrape has a variable called likes_count. We sorted the whole dataset based on the likes_count in the decreasing order of their likes_count. This helped us in displaying the top 10 Tweets that are tweeted regarding the movie.

Handling of Null Values:

There are few null values in the dataset associated with the columns ‘Tweet’, ‘Username’ and ‘location’. Null values in ‘location’ are imputed using the ‘mode’ method. Fields in ‘Tweet’, ’Username’ with the null values are dropped.

In [194]:
```
missing_val = pd.DataFrame(data.isnull().sum())
missing_val = (missing_val/len(data))*100
missing_val.reset_index()

missing_val = missing_val.rename(columns = {'index': 'Variables', 0: 'Missing_percent'})
missing_val
```

Out[194]:

	Missing_percent
Unnamed: 0	0.000000
Datetime	0.000000
Tweet	0.014000
ID	0.000000
likes_count	0.000000
Username	0.014000
verified	0.000000
followers	0.000000
friends	0.000000
favourites	0.000000
statuses	0.000000
location	28.167718

Feature Extraction/Engineering:

As part of feature engineering, we did the below tasks:

- Extracted new variables ‘year’, ‘month’, ‘day’, ‘hour’ from the existing ‘Datetime’ column.

```
In [198]:
import datetime
data['year'] = data['Datetime'].dt.year
data['month'] = data['Datetime'].dt.month
data['day'] = data['Datetime'].dt.day
data['hour'] = data['Datetime'].dt.hour
```

- Changed the datatype of variables ‘location’ and ‘verified’ to category because they can be classified as categorical data.

```
data['location']=data['location'].astype('category')
data['verified']=data['verified'].astype('category')
```

Feature Selection:

Feature selection plays a vital role in data analysis. There is a high chance of redundancy due to the dependent variables because they carry the similar data to the target variable. Such variables can be filtered out from the dataset by doing a correlation test. Correlation matrix is constructed using the method corr() which consists of values between -1 and 1. A Variable must be dropped if the correlation(positive or negative) is pretty high.

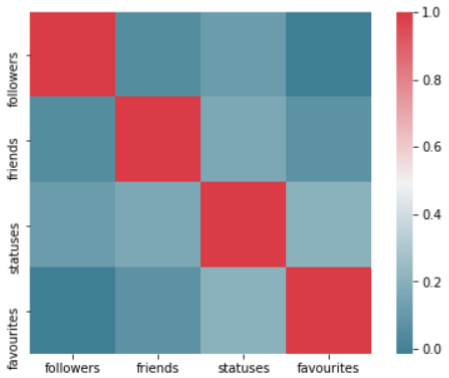
- If the value tends to -1 then the variables are negatively correlated.
- If the value tends to 1 then the variables are positively correlated.

In [199]:
```
cnames = ['followers','friends','statuses','favourites']
corr = data.loc[:,cnames]
#Set the width and hieght of the plot
f, ax = plt.subplots(figsize=(7, 5))

#Generate correlation matrix
corr1 = corr.corr()

#Plot using seaborn library
sns.heatmap(corr1, mask=np.zeros_like(corr1, dtype=np.bool), cmap=sns.diverging_palette(220, 10, as_cmap=True),
            square=True, ax=ax)
```

Out[199]: <AxesSubplot:>



The above graph was constructed using seaborn library and the boundaries are defined from 0 to 1. If the value is '0' then the variables are negatively correlated, if the value is '1' then the variables are positively correlated to each other.

5. SENTIMENT ANALYSIS :

”The method of detecting whether a piece of text is positive, negative, or neutral is known as sentiment analysis.” Sentiment analysis[8] is used by data analysts in large corporations throughout the world to analyze public opinion, conduct market research, track product and brand reputation, and analyse customer experiences. In various industries, sentiment analysis is used in a variety of ways. It allows businesses to identify customer sentiment toward products, brands or services in online conversations and feedback. Sentiment analysis models can flag any situation which is not expected and thus enable you to take action right away. The process of tagging text by sentiment is very subjective and is easily influenced by one's thoughts, beliefs and even personal experiences.

5.1 Preprocessing

As text is the most unstructured form, it needs extensive cleaning. These pre-processing procedures aid in the conversion of noise from high-dimensional features to low-dimensional space, allowing for the extraction of as much correct information from the text as possible.We have applied following techniques before actually feeding the data to sentiment analysis models.

1. **Tokenization**

   For a given character sequence Tokenization is the process of transforming the text into pieces, called tokens.

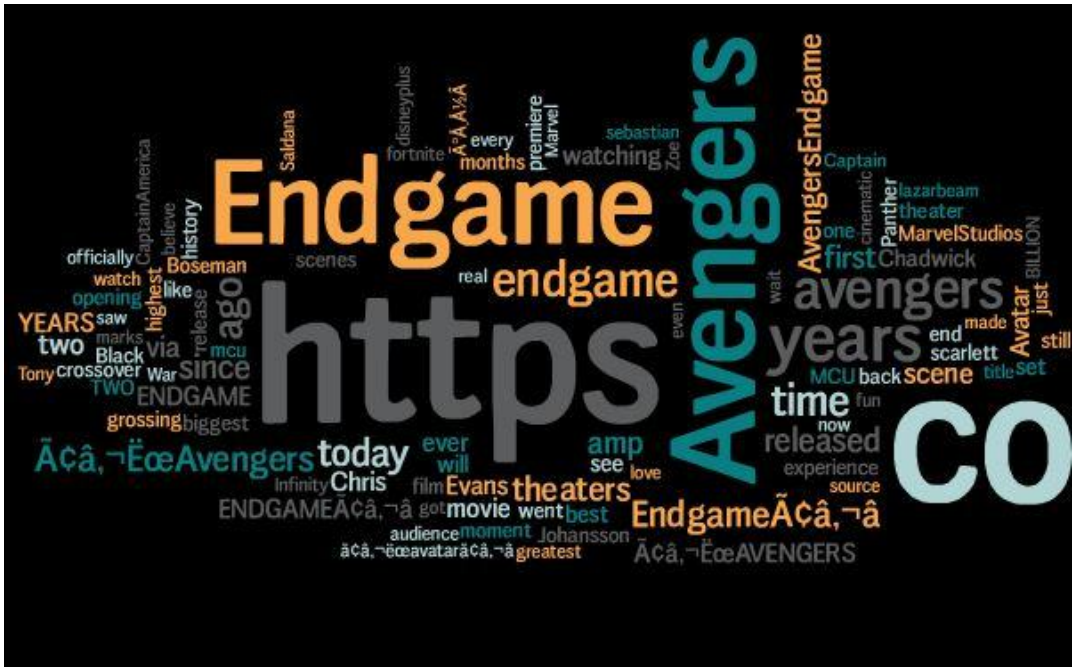
2. **Stopwords**

   The most commonly occurring words that do not contribute majorly to the context of the data are called Stopwords. They generally do not add any value to the sentence's meaning. These stopwords are usually removed before sentiment analysis is performed.

3. **Stemming**

   Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even if the stem itself is not a valid word in the Language. We have used NLTK python library, which is the one stop solution for tokenizing, stemming and removing the stop words from the text.

5.2 Word Cloud



Word cloud[9] for the most liked tweets about the movie "The Endgame"

Above we can see the word cloud generated from the tweets which are having the most likes. The words in the above word cloud are the most appeared words among those tweets. Among those words, the words which are appearing with large and thick font are the most repeated words among the tweets. And the words with small and thin font are the least appeared words. The word cloud is generated using the "pro word cloud" Add-on in the microsoft word.

What are Word Clouds?

Word clouds (also known as text clouds or tag clouds) work in a simple way: the more a specific word appears in a source of textual data (such as a speech, blog post, or database), the bigger and bolder it appears in the word cloud.

A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is.

Also known as tag clouds or text clouds, these are ideal ways to pull out the most pertinent parts of textual data, from blog posts to databases. They can also help business users compare and contrast two different pieces of text to find the wording similarities between the two.

5.3 Models and Methodologies Used

VADER[10] (Valence Aware Dictionary and sEntiment Reasoner) is a sentiment analysis tool that is specifically attuned to sentiments expressed in social media using lexicon and rule-based analysis. It makes use of a combination of sentiment lexicon, which is a list of lexical features (e.g., words) which are usually labelled according to their semantic orientation as either negative or positive.

VADER has been quite successful when judging the sentiment of social media texts such as twitter comments, Tweets, FB posts etc. The reason behind VADERs success is that even for text having slang, punctuations, emotions or unstructured text, it is able to judge how positive or negative a sentiment is.

5.4 Why VADER outperforms?

The production of lexicons is extremely time consuming and very expensive thus they are rarely updated meaning they lack the current age slangs which might be in any dictionary.

VADER analyses sentiments primarily based on below key points, which enables it to precisely predict the sentiments of the text

1. **Punctuation**: The use of an exclamation mark(!), increases the magnitude of the intensity without modifying the semantic orientation. For example, “The food here is good!” is more intense than “The food here is good.” and an increase in the number of (!), increases the magnitude accordingly.

2. **Capitalization**: Using upper case letters to emphasize a sentiment-relevant word in the presence of other non-capitalized words, increases the magnitude of the sentiment intensity. For example, “The food here is GREAT!” conveys more intensity than “The food here is great!”

3. **Degree modifiers**: Also called intensifiers, they impact the sentiment intensity by either increasing or decreasing the intensity. For example, “The service here is extremely good” is more intense than “The service here is good”, whereas “The service here is marginally good” reduces the intensity.

4. **Conjunctions**: Use of conjunctions like “but” signals a shift in sentiment polarity, with the sentiment of the text following the conjunction being dominant. “The food here is great, but the service is horrible” has mixed sentiment, with the latter half dictating the overall rating

5. **Preceding Tri-gram**: By examining the tri-gram preceding a sentiment-laden lexical feature, we catch nearly 90 percent of cases where negation flips the polarity of the text. A negated sentence would be “The food here isn’t really all that great”.

5.5 VADER implementation in Jupyter(Screenshot):

```
In [242]:  #VADER IMPLEMENTATION
           analyzer = SentimentIntensityAnalyzer()

           def text_sentiment_vader(text):
               vs = analyzer.polarity_scores(text)
               return int(vs.get("compound")>0)

           predictions = x_test.Tweet.map(lambda x : text_sentiment_vader(x))
```

5.6 VADER implementation for UI(Screenshots):

Tweets are appended to a list and VADER is applied on that list. Polarity scores are calculated for each and every tweet. Finally, the compound scores are appended to the variable sentiments.

```
tweets_list = tweets_df2['Tweet'].tolist() #select tweet column of all tweets
for i in tweets_list:
    sentiment.append(analyzer.polarity_scores(i)) #find polarity score of each tweet
for i in sentiment:
    sentiments.append(i['compound']) #select compoud key from the dictionary generted by above code for each tweet
movie.sentiments = sentiments #store all compounds of all tweets
```

Tweets are classified to ‘positive’, ‘neutral’ and ‘negative’ based on the sentiment scores obtained. This is shown in the below screenshot.\

```
#calculate mean of all the compounds of all tweets
 (variable) movie: Movie s)/100000
movie.sent_val = round(((sent_val1+1)/2)*100)

#Classify movie revie as negative or neutral or positive based in mean value
if movie.sent_val < 40:
    movie.review = "Negative"
elif movie.sent_val >= 40 and movie.sent_val < 70:
    movie.review = "Neutral"
else:
    movie.review = "Positive"
```

5.7 AFINN

AFINN[11] is an English word list whose scores range from -5(most negative) to +5 (most positive). The English language dictionary in the current version of the lexicon is AFINN-en-165.txt which contains over 3,300+ words with a polarity score associated with each word. Afinn requires data cleaning and pre-processing of data before actually running the sentiment analysis which is different from VADER.

Which is best?

VADER has a better performance than AFINN and is doing a better job in terms of accuracy. We handled the data with 100k records and implemented AFINN and VADER without considering any sample. Our output is classified into ‘positive’, ‘negative’, ‘neutral’ based on the polarity scores obtained from both the algorithms.

- Output is classified as ‘positive’ if polarity score > 0.5
- Output is classified as ‘negative’ if polarity score < -0.5
- Output is classified as ‘neutral’ if polarity score ranges between -0.5 and 0.5

The results of analysis showed that VADER has an accuracy of over 70% and AFINN has an accuracy of about 67%. Here, VADER slightly out performed than AFINN. VADER is easy to implement because it does not need separate preprocessing, whereas AFINN requires data cleaning and preprocessing before applying it for sentiment analysis.

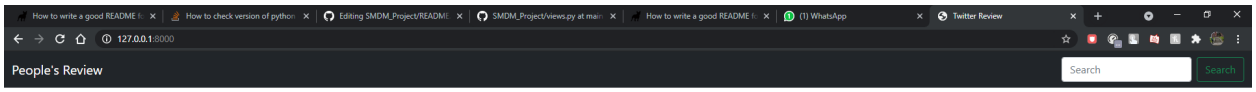
Model	Accuracy
AFINN	**67.8%**
VADER	**70.001%**

6. Results:

6.1 Overview of the Web Page:

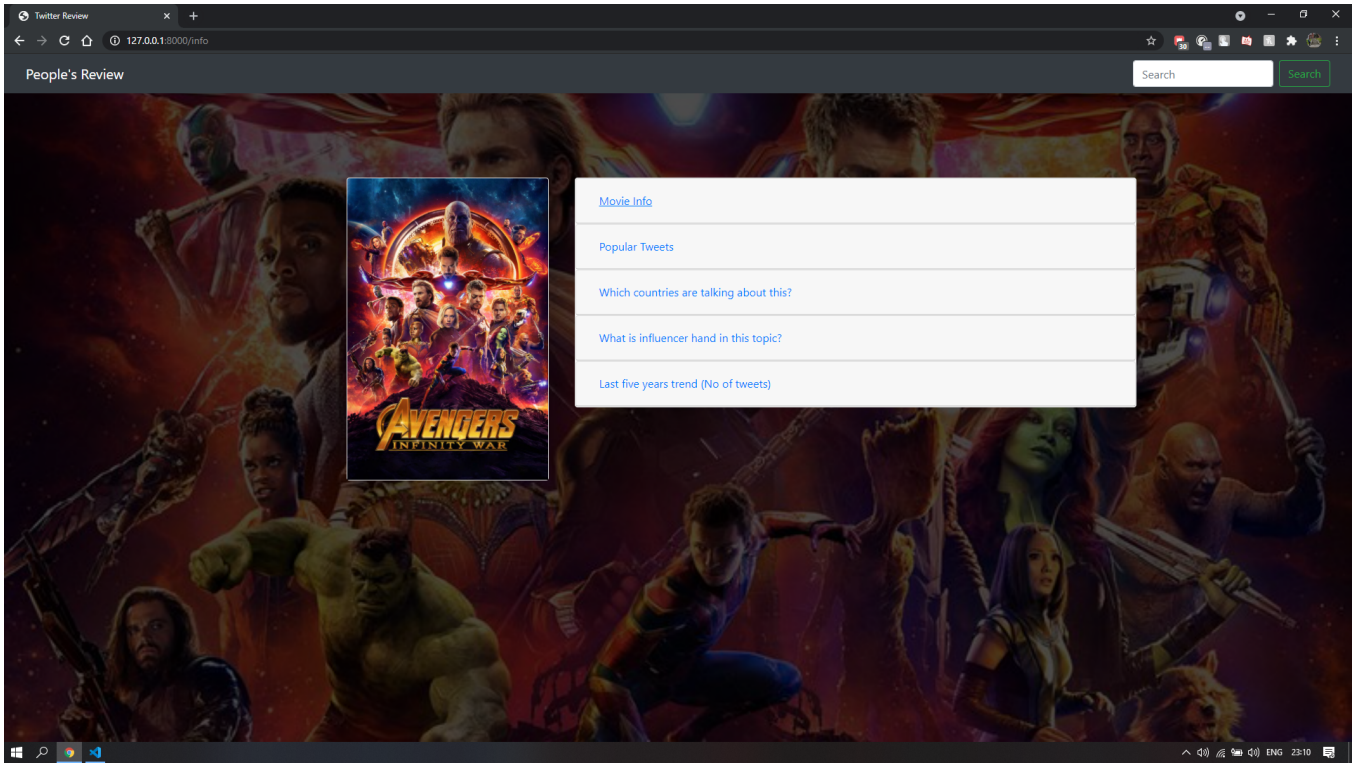
The search box allows the user to enter the movie name along with the year. And the user clicks the search button. The web page is developed using Django Framework which is a python based framework. The webpage runs on a local host. To develop the front-end of the web page, HTML, CSS and JavaScript are used. The back-end is solely developed in python. BootStrap cloud services are used to style the web page.

Initially, the webpage will have a search box allowing the user to enter the movie name along with the year in which it has been released.



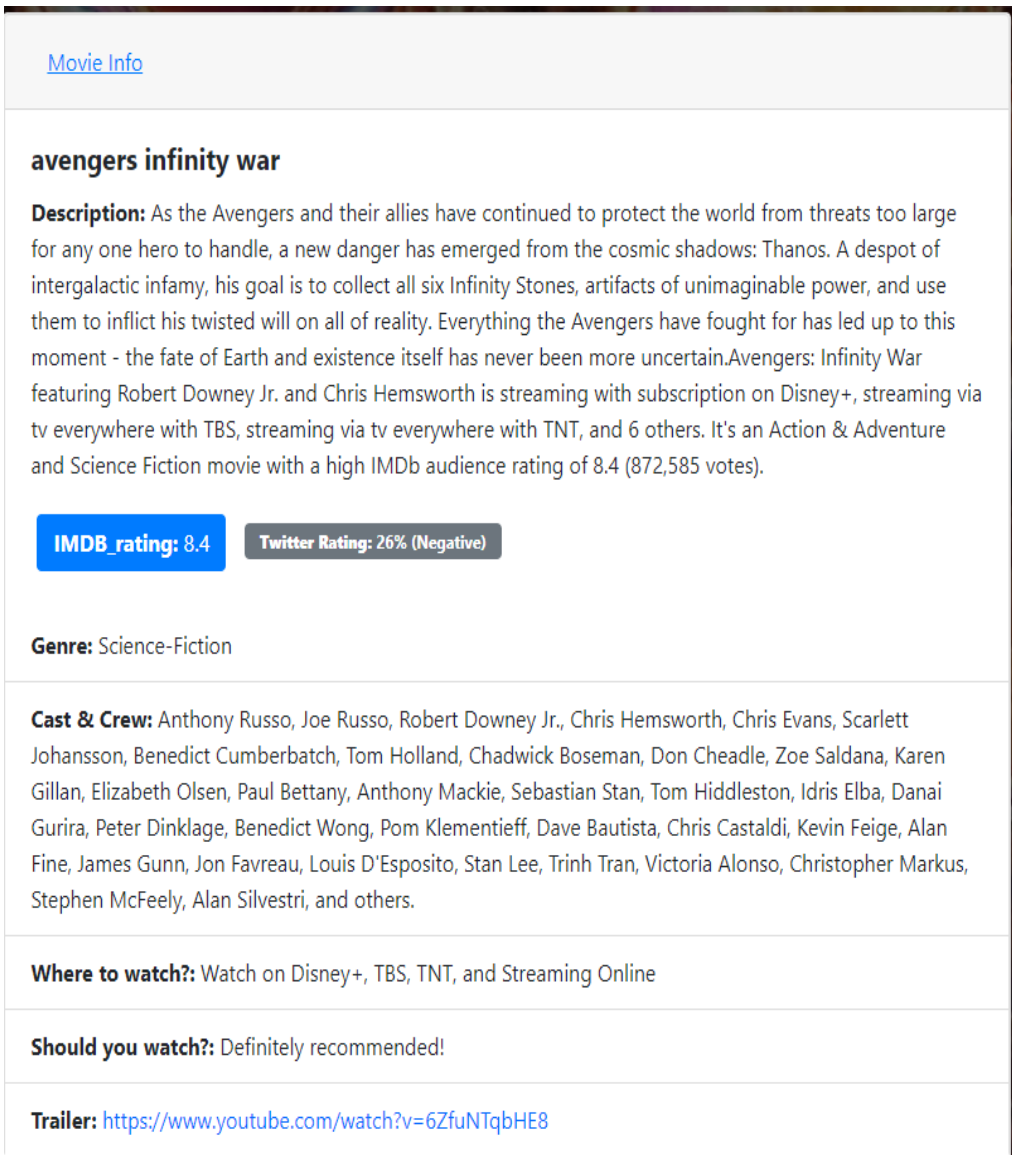
6.2 Home Page:

As soon as the user clicks the search button, the below page is loaded on to the screen. It shows the movie poster and other tabs. By the time this page is loaded, the whole Sentiment Analysis and visualization would already be done in the background and are readily available for the user to explore. However, it takes around 17 minutes to load the webpage if we take 100k tweets for Sentiment Analysis and 1 minute if we take 5k tweets.



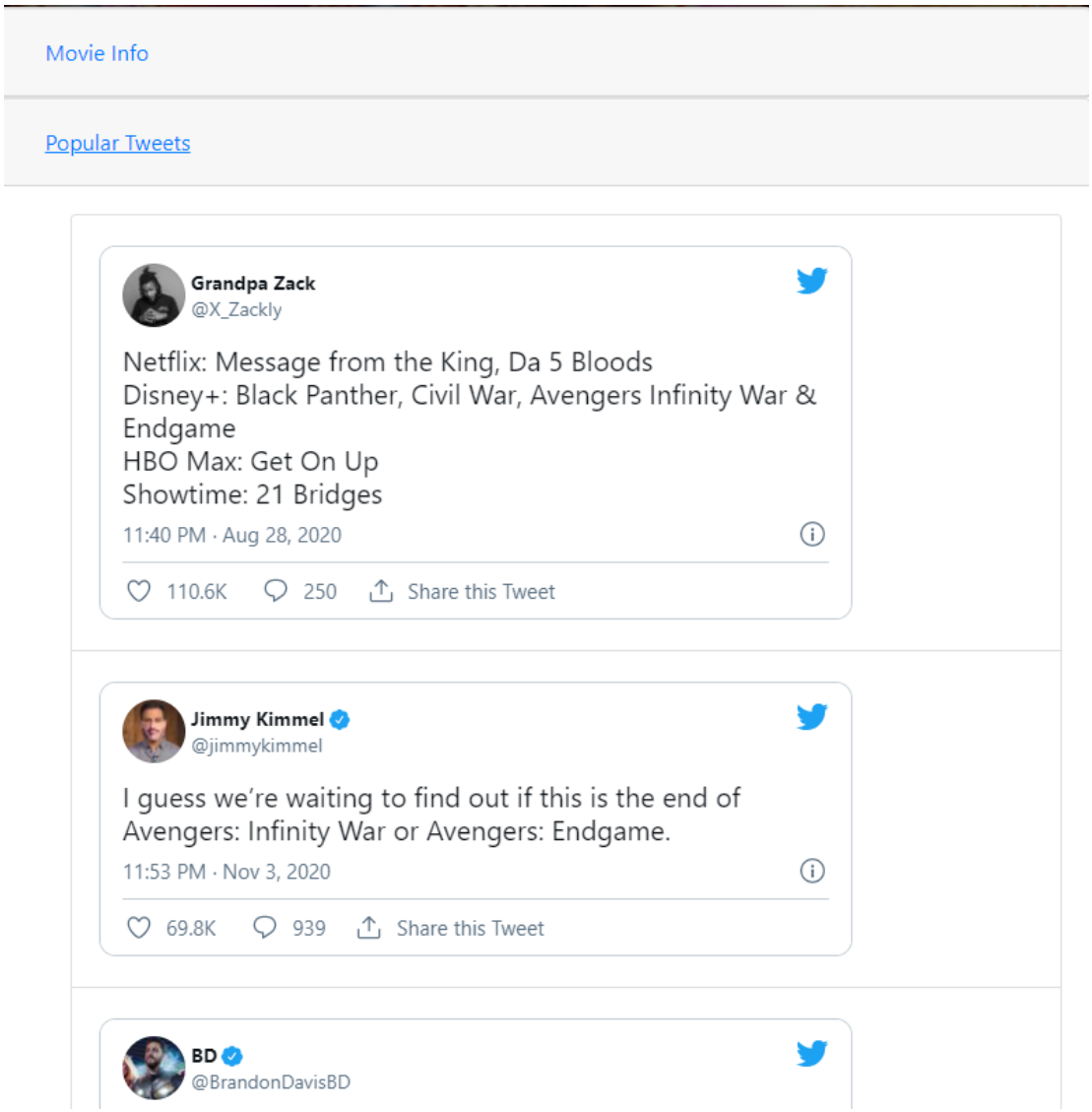
6.3 Movie info:

The movie info tab displays all the information regarding the movie that is fetched using web scraping. First it displays the movie title, the description, IMDB rating, Genre, Cast & crew, Where to watch(Online OTT Platforms), Should you watch, Trailer (youtube link to the trailer). We have another component called Twitter Rating, which is the rating we get after performing Sentiment Analysis.



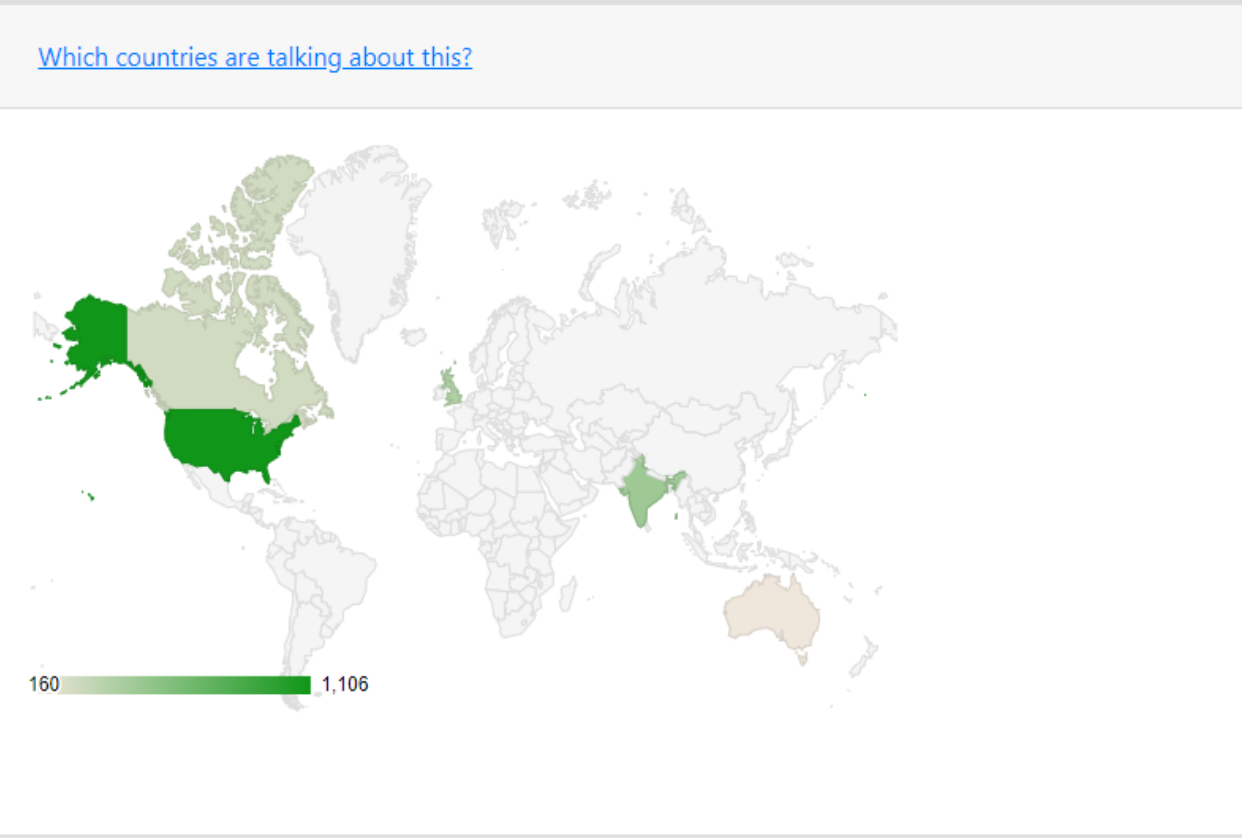
6.4 Popular Tweets:

We are displaying the top 10 tweets under the popular tweets tab. The tweets are ordered in decreasing order of their likes_count. From these tweets, the first 10 tweets are fetched using a unique id that each tweet has. The user can like the content, comment and either share the tweet from the webpage. If the user intends to do such actions, the web page automatically redirects to Twitter and the user can like or do the above mentioned actions.



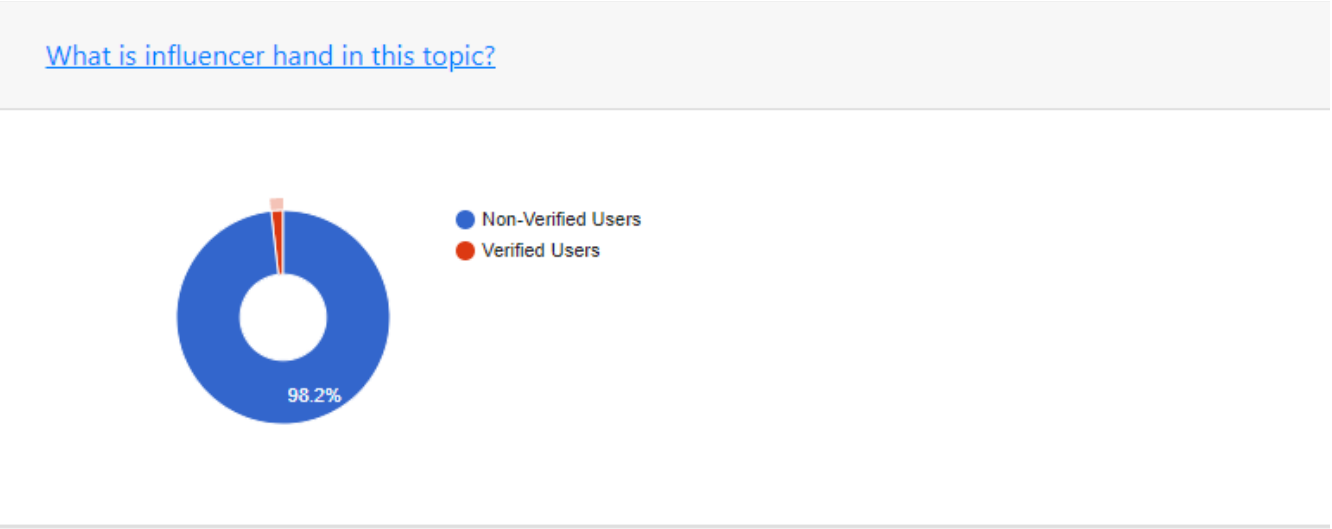
6.5 Which countries are talking about this?:

This tab has a geo chart that depicts the top 5 countries that contributed the most number of tweets regarding the movie. To achieve this a library called pycountry has been used that contains the names of all the countries in the world. When the user hovers over a country, it displays the number of tweets made by the people from that country.



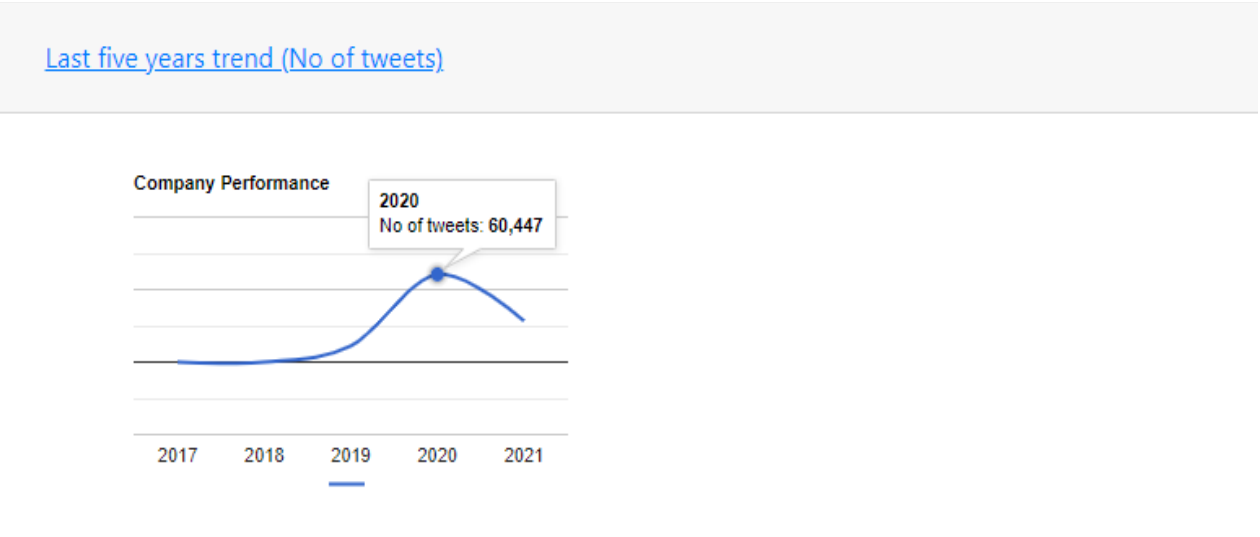
6.6 What is the influencer's hand in this topic?

This tab contains the pie-chart depicting the kind of users that influenced the movie review the most. This graph is generated using the verified attribute in the dataset. The pie-chart shows the percentage of verified users and non-verified users on Twitter that made tweets about the movie.



6.7 Last five years trend:

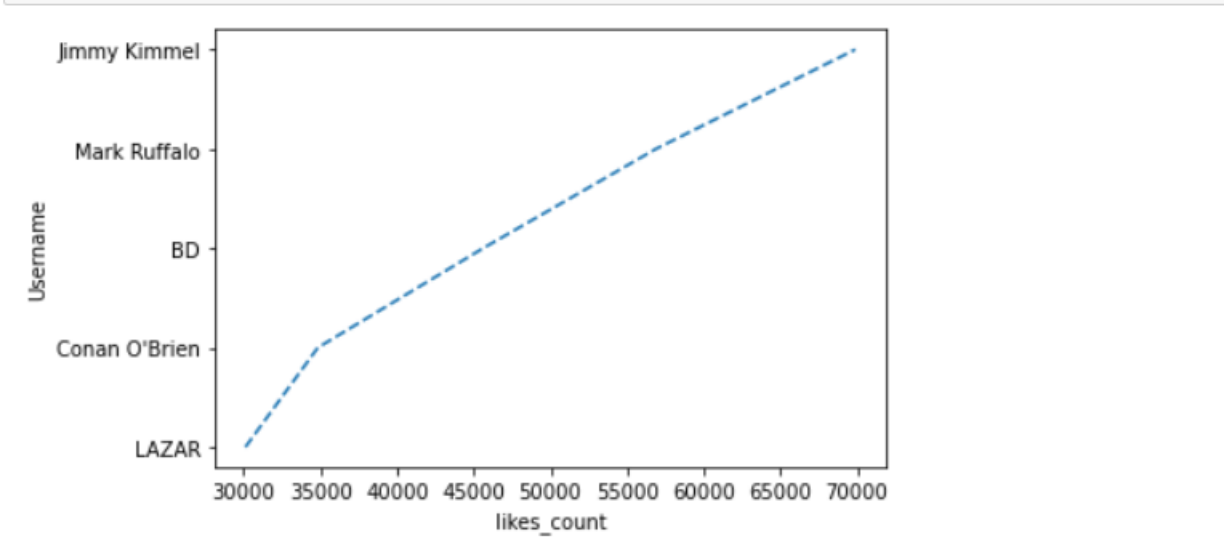
The graph depicts the trend of the movie on Twitter over the span of 5 years. This graph is based on the number of tweets tweeted regarding the movie in that particular period. When the user hovers over the line in the graph, he/she can see the total number of tweets made in that year.



7 Additional Analysis:

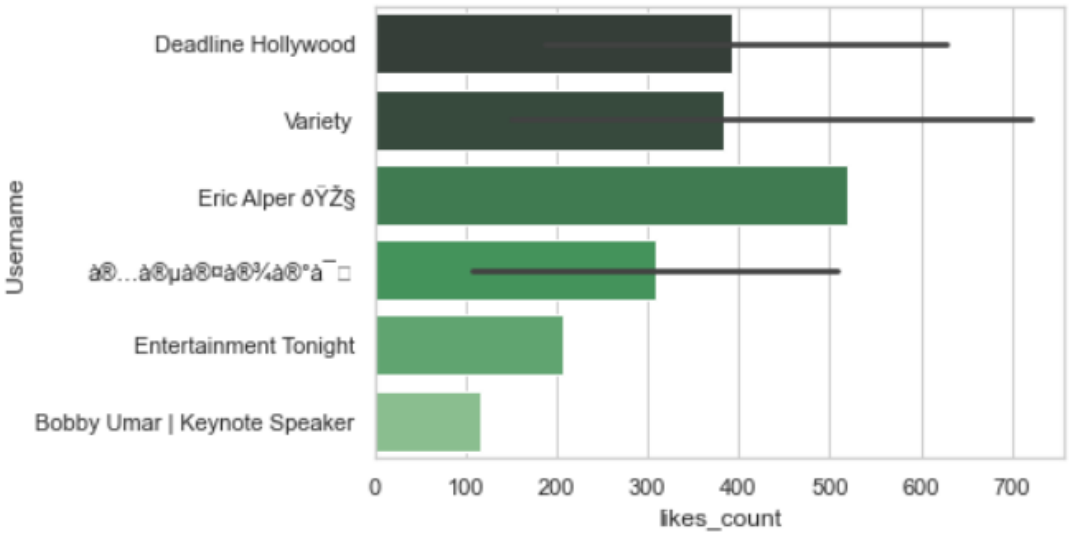
7.1 Dotted line plot between verified users and likes:

Graph shows the number of likes obtained for the tweets posted by the top 5 verified users. This analysis shows that the tweets posted by the users ‘Jimmy’, ‘Mark’,’BD’, ‘Conan’ ,‘LAZAR’ have greater impact towards the movie review.



7.2 Bar plot between users(having more friends) and likes count:

Movie reviews can also be impacted by the tweets from the users who have more friends. Below graph contains the user names on y-axis having friends more than a lakh and the likes obtained for the tweet on x-axis. This analysis shows that their tweet has a greater reach and this might have a significant change on the review (either positive or negative). In the bar chart, the color of each bin depicts the tweet as positive or negative. If the color palette is darker the sentiment score tends towards positive.



7.3 Active users count:

As part of exploratory data analysis,

- There are 2610 users with more than 50k followers.
- There are 17684 users with more than 50k tweets.
- There are 5 users whose tweet has more than 50k likes.

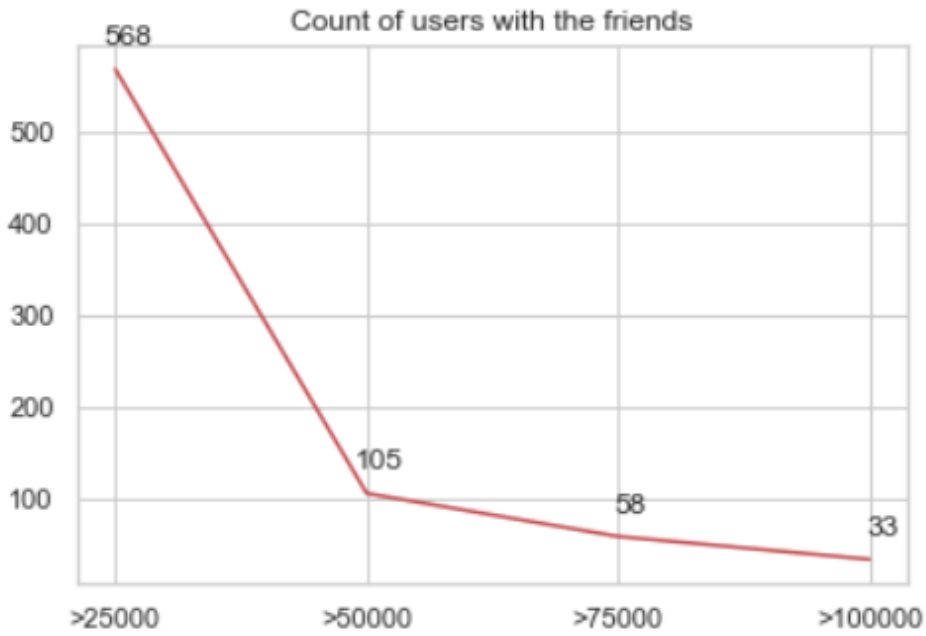
By our analysis, we can state that these users are more active in twitter than other users.

In [350]:
```
#count of users with followers more than 50000
print("Count of users with followers more than 50000 is->",
      data1[data1['followers'] > 50000]['Username'].count())
print("Count of users with Statuses more than 50000 is->",
      data1[data1['statuses'] > 50000]['Username'].count())
print("Count of tweets with likes more than 50000 is->",
      data1[data1['likes_count'] > 50000]['Tweet'].count())
```

```
Count of users with followers more than 50000 is-> 2610
Count of users with Statuses more than 50000 is-> 17684
Count of tweets with likes more than 50000 is-> 5
```

7.4 Line graph between count of users and friends range:

This graph contains the number of users with more than (25k, 50k, 75k, 100k) friends. It can be inferred that tweets posted by these users might have a larger impact on the movie review due to the greater reach.



8. Conclusion:

As we have discussed, most of the people glance at movie reviews prior to watching the movie. People should go through a lot of reviews and interpret the statements made by others in various media like news, social media etc. This can be a tedious and tiring task. So, we developed a web page that allows the user to search for a movie and the user gets an instant review which is more reliable than going through a bunch of tweets. The movie review is comprehended through the thorough analysis of thousands of tweets using Sentiment Analysis. The webpage also provides the user with visualizations regarding various aspects affecting the movie review.

To achieve this, we collected data from the internet using web scraping and from Twitter using snscrape. Displayed the data collected by scraping the web on our webpage, and performed Sentiment Analysis on a dataset of hundred thousand tweets related to the movie to predict the review of a movie. Sentiment Analysis is done using VADER which is a module in nltk python library. VADER is intelligent enough to map the lexical structure to intensity of emotions. And it

generates key value pairs of positive, negative, neutral and compound as keys and their respective fractions as values. We calculated the review by averaging the compound values which are calculated by normalizing the positive, negative and neutral scores.

Moreover, to provide the users with interactive visualizations such as geo charts and pie charts, Google Charts API has been used. These graphs allow the user to interact with them. The user can like, share and comment on the tweets. The user can know the number of tweets tweeted when hovered over the geo chart and linear graphs.

The webpage provides the user with a unique experience of learning the review of a movie and the factors impacting it.

9. Future Scope:

There is always room for development. Let us discuss such developments that are possible in this project. In this project, the data we collected for Sentiment Analysis is restricted both in numbers and from where it is collected. The data is extracted from Twitter and hundred thousand tweets related to the movie are fetched. This is because it costs us a lot of time to fetch more tweets. The review can be more reliable if we collect the opinions of people posted on other platforms as well.

Moreover, Twitter doesn't allow us to know the gender and age of the twitter user. If that data were available, we could extend our analysis to how different age groups are affecting the review of a movie and what different gender groups think about the movie.

10. Bibliography

[1] Survey on mining subjective data on the web https://link.springer.com/article/10.1007/s10618-011-0238-6

[2]Sentiment analysis using vader https://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html

[3] simplifying sentiment analysis using vader: https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f

[4] https://www.ijcaonline.org/archives/volume182/number50/mamtesh-2019-ijca-918756.pdf

[5]Vader:https://www.semanticscholar.org/paper/VADER%3A-A-Parsimonious-Rule-Based-Model-for-Analysis-Hutto-Gilbert/bcdc102c04fb0e7d4652e8bcc7edd2983bb9576d

[6] Web Scraping using python: https://realpython.com/beautiful-soup-web-scraper-python/

[7]Snscrape:https://github.com/JustAnotherArchivist/snscrape#:~:text=snscrape%20is%20a%20scraper%20for,items%2C%20e.g.%20the%20relevant%20posts.&text=Instagram%3A%20user%20profiles%2C%20hashtags%2C%20and%20locations

[8]Sentiment Analysis: https://link.springer.com/chapter/10.1007/978-981-15-0135-7_31#:~:text=Sentiment%20analysis%20of%20a%20movie,training%20and%20testing%20the%20data.

[9] Word Cloud: https://boostlabs.com/blog/what-are-word-clouds-value-simple-visualizations/

[10]Vader:https://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html

[11] Sentiment Analysis using Affin https://www.geeksforgeeks.org/python-sentiment-analysis-using-affin/