Movie Tweet – Sentiment Analysis and Recommendation Module of Movies

Preet Parikh
Albany ID – 001353411
prparikh@albany.edu
Computer Science Department
University at Albany
Albany, USA

Abstract - Sentiment Analysis means classification of a text or a sentence according to their overall positivity or negativity. Here, we analyze sentiments of tweets of movie to derive the rating and review of the movie on the box office. We also form clusters using unsupervised algorithm to classify the tweets of the movie. We also propose a recommendation module for the movies using association rule mining.

1) Introduction

Almost everyone who watch movies considers getting a review and Rating of the movie before spending hours over it. So, one would always love to get reviews and ratings based on multiple public opinions rather depending on just one person. Now-a-days as the advent of social media is increasing it is very important to take into account social media networks while coming to some results which include public response and opinion. Tweeter is one place where multiple people share their opinion publicly without any hesitation. Twitter is one such highly used social media network where people share their opinion and reaction over various topics they are a part of or something which is trending.

Recommendation has become an important part in everybody's life. Recommendation helps in narrowing down the choices of a particular and hence enhancing the utilization of time by not wasting time on something which is not recommended. Obviously, one would not like to stick to only to recommend things, but recommendations make life easy.

As we're watching movies on regular basis, which motivated us to build this system. Also, there are lots of confusion about selecting the movies everywhere and people don't know what they should go for. The movie watchers can use this application to see the reviews of the movie and based on the user's likings we will give user suggestion of movies of same genre. People can easily use some online available reviews and ratings and come to the deductions. They can also rely on reviews and ratings of a famous personality in the field. But, the problem with such reviews and ratings is that those are opinion of just one personality who might have different mindset than others. The reviews and opinions are also often paid to have good mouth of word. Hence, it becomes challenging to rely on such reviews and ratings. Here, we can address the challenge by not relying on single opinion but collecting opinions of vast diversity who are not directly related and deriving results from them.

The project collects tweets based on the movie entered by the user, gives rating and reviews based on sentiment analysis, K-means clusters and word cloud. The other module of the project demonstrates the Recommendation Model based on Association Rule Mining.

2) Related Works

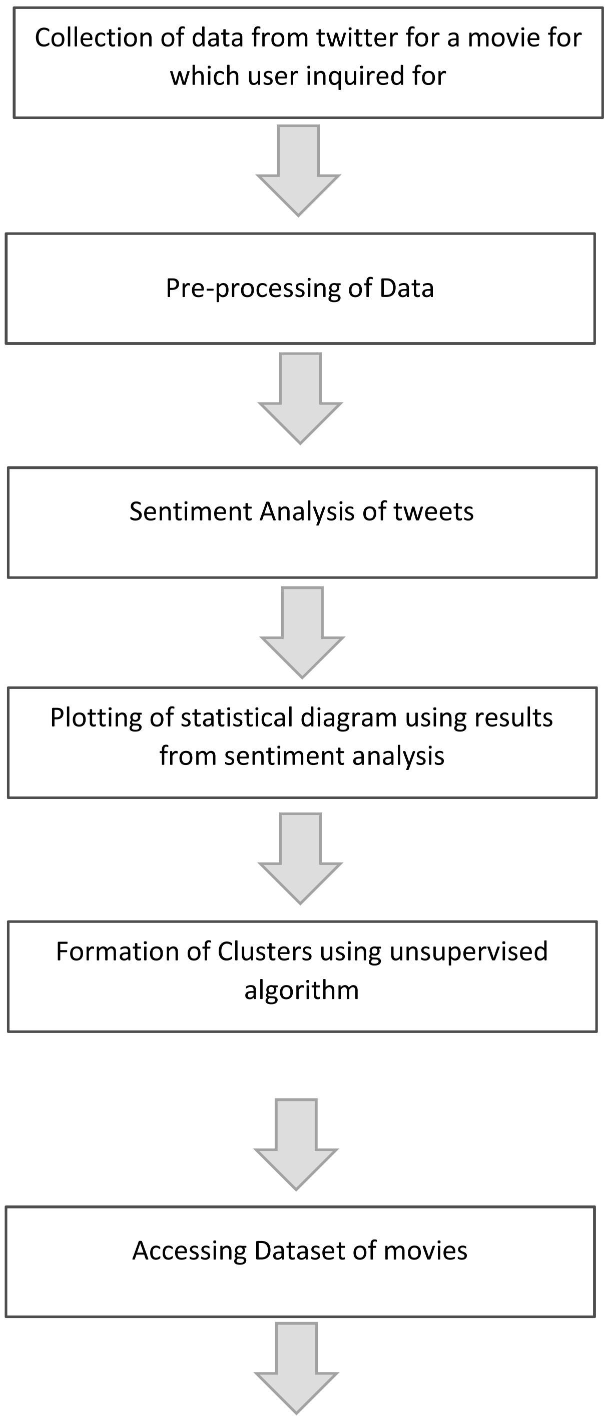
"Bag of words meets Bag of Popcorn' by Kaggle is one such related work in this field. In this work, they have discussed sentiment analysis by using special labeled data set which consists of 50,000 movie reviews. The sentiment of the review is mentioned as a binary number here. Rating < 5 is denoted as 0 and Rating >= 7 is denoted as sentiment score of 1.

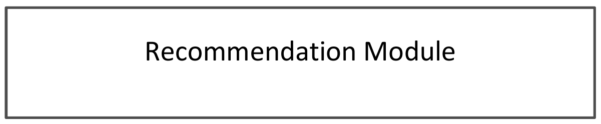
There is one other related work on sentiment analysis which uses dataset from the rotten tomatoes. Here, they have done sentiment analysis on a scale of 5 labels: negative, somewhat negative, neutral, somewhat positive and positive.

Movie Recommendation has been gaining attention since last couple users. Netflix, Hulu and Amazon Prime use such recommendation system on very large base. This service implements this system on a large platform using multiple complex algorithm which includes but not limited to user's personal watching history, user's search history, similar genre and number of views of a video.

3) Proposed approaches

The implementation of the system requires following steps to get the aimed results. The diagram below shows the steps required to reach end results:


Collection of data from twitter for a movie for which user inquired for
Pre-processing of Data
Sentiment Analysis of tweets
Plotting of statistical diagram using results from sentiment analysis
Formation of Clusters using unsupervised algorithm
Accessing Dataset of movies



Here, first we target to collect tweets from twitter for the movie entered by the user. After collection of tweets, its pre-processing and proper storage is done. Tweets are then analyzed to get idea of number of positive, negative and neutral tweets. This classification is done using polarity of the tweets using pattern module. Later, using values we get from the analysis of the tweets we plot statistical diagram using matplotlib library. The statistical diagrams give idea of the whole analyzed data in one view. Hence, plotting is very important. We also get the percentage value of the classification of the tweets. Later, clusters are formed using k-means algorithm and tweets for different clusters is stored in different files in proper directory of the movie. Then, for recommendation module we use large dataset of about 30000 movies from all genre and all languages. This is very rich data set of all the movie released and it also contains movies which are not yet released but are announced. Accessing this data set and using association rule mining for the genre of the movies, we can get the recommendation of the movies from the dataset based on the genre we watch and love, recommended genre and the year we want movie list of. We will discuss each of this implementation in detail in this paper.

4) Data Collection

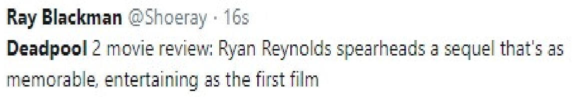
Data Collection is very important part of the project as all the later parts of the project depend on the data which has been collected. One would get better results if the data collected is proper and accurate. Now, the important question which arises here is: What data to collect?? As our project is based on sentiment analysis of movies, hence we want data related to movies. Here, our focus for collection of data is limited only to twitter but for extension we can also collect data from Instagram, Facebook and other social media networks.

For collection of data from twitter here we use twitter REST API. Python provides rich twitter API for developers which can be used to collect tweets of users from twitter on specific topic. Here, tweets collected are real time and also historical up to some time frame. REST API limits user allowing collecting only 180 tweets for a 15-minute time frame. The twitter API gives access only after verifying 4 keys which are unique for every developer. The tweets are collected based on keyword and/or location given to the system. Here, for this we used only 1 query which consisted of keyword as "Movie Name" which is entered by user. Here, is a code snippet of the query which was used.

```
query = movie+' AND movie AND review'
MAX_ID = None
tweets = api.GetSearch(query,count=100,
max_id=MAX_ID, result_type='mixed')
```

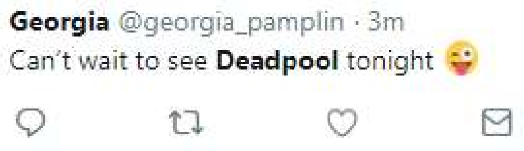
Here, we use query which has movie name which has been entered by the user. It is used along with the keywords movie & review using AND operator. The reason behind using AND operator along with the keyword 'Movie' is that we want data/tweets only specific to the movie and not other data. For eg – If we enter a Movie Name: Game Night, then without the AND operator it would give us data/tweets also related to word Game Night which is used on social media very often. Hence, that data would be of no use. Similarly, the reason behind AND operator with the keyword review is that we want tweets only which has some review regarding to that movie in it. Other tweets might or might not have the data which is actually targeted in the collection of the tweets.

For Eg:


Ray Blackman @Shoeray · 16s
Deadpool 2 movie review: Ryan Reynolds spearheads a sequel that's as memorable, entertaining as the first film

This tweet here has the keyword both movie and review and has the perfect data as we wanted.

Whereas,


Georgia @georgia_pamplin · 3m
Can't wait to see Deadpool tonight 😜

This tweet here only has the keyword 'Deadpool', which gives no idea about the review of the movie
Here, we haven't used any geo location filter for the collection of the tweets as it doesn't matter from what location does the tweet comes from if it solves our purpose of knowing the review of the movie.

The tweets collected here are processed further to analyze and get results.

5) Data Storage & Pre-processing

After the tweets are crawled using the REST API, they are subject to storage and pre-processing. The collected tweets are stored in a text file which would be named by the movie name the user enters. This text file is saved in the directory which will then be created after collecting tweets. The name of the directory will be same as the movie name or the text file.

The REST API gives us tweets which contains all the details about the tweets along with the metadata of the tweets like tweet id, user id, date and time of creation, etc. Here, we only need the text of the tweet which was posted by some user on twitter. Hence, we extract only the text part of the tweet and save it.

The Code snippet for the above implementation is as below:

```
for raw_tweet in tweets:
    tweet =
json.loads(str(raw_tweet))
    info={"text":tweet['text']}

f=open(""+movie+"/"+movie+".txt",
'a+')
    f.write(json.dumps(str(info)))
```

The data stored here is further subject to pre-processing. For pre-processing we remove stop words from the collected tweet file. The data after storage and pre-processing is further made available to the sentiment analysis module which is a very important module of the project.

6) Sentiment Analysis

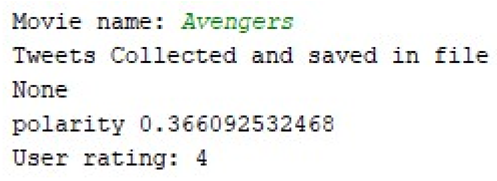
Sentiment Analysis is defined as "The process of computationally identifying and categorizing opinions expressed in a piece of text". In simple words, classification of a text or a sentence according to their overall positivity or negativity. Sentiment Analysis include several tasks but here our target will be pretty narrowed down to get the sentiment of the tweets and map them to the ratings. We classify our tweets under 3 labels: Positive, Negative and Neutral.

Here, in our project, we used pattern library for the task of the sentiment analysis. First, let us understand how pattern library works. Pattern library breaks up a given sentence or text in to parts to identify different parts of text as noun, verb, adjective, etc. It contains a database of adjectives and adverbs which contain negative and positive ratings for the negative and positive sentiment. It is also subject to other things in the text. Based on this, one can figure out the polarity of the sentence by combining ratings of different parts of speech.

The pattern.en sentiment() method gives us (polarity, subjectivity)-tuple for the given sentence. Here, we only use pattern.en.polarity method to get the polarity of each tweet in the file which was stored earlier. The Code snippet for the same can be seen below:

```
f=open(""+movie+"/"+movie+".txt").r
eadlines()
b=sum(1 for line in f)
for line in f:
    a=pattern.en.polarity(line)
```

Further, we find average of the polarity of the file and then we map this to the user rating based on some research. The average enables us to know the overall polarity of the whole collected data. This average polarity is mapped to a number between 0 and 5 which denotes the user rating of the movie.



```
Movie name: Avengers
Tweets Collected and saved in file
None
polarity 0.366092532468
User rating: 4
```

A Screenshot of the result can be seen above which first asks to enter a movie name, which on entering collects tweets and saves in a file. Further, the overall polarity of the file is calculated and is mapped to a number between 0 and 5. This number denotes the user rating.

We have extended this module to give user an idea of what percentage of the users gave positive review and what percentage of the users gave negative review. For this, we checked the polarity of each tweet line. Next, we initialized 3 counters which counts the number of positive, negative and neutral tweets.

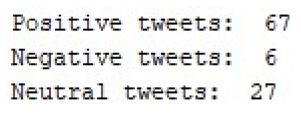
A code snippet for the above implementation and result can be seen below:

```
for line in f:
    a=pattern.en.polarity(line)
    if a>0 and a<=1:
        pos=pos+1
print "Positive tweets: ", pos

for line1 in f:
    a=pattern.en.polarity(line1)
    if a<0 and a>=-1:
        neg=neg+1
print "Negative tweets: ", neg

for line2 in f:
    a=pattern.en.polarity(line2)
    if a==0:
        nut=nut+1
print "Neutral tweets: ", nut
```

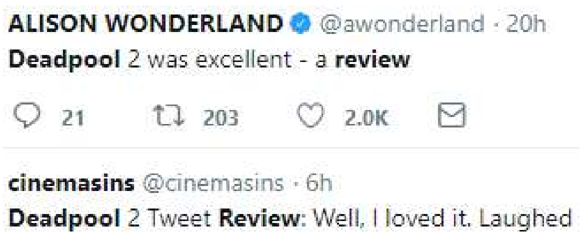
Implementing this code, gives us following output:



```
Positive tweets:  67
Negative tweets:  6
Neutral tweets:  27
```

Now, let us see some examples of positive, negative and neutral tweets:

1)Positive tweets:

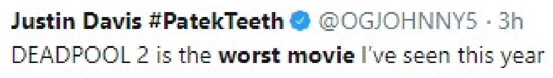


ALISON WONDERLAND @awonderland · 20h
Deadpool 2 was excellent - a review

21 203 2.0K

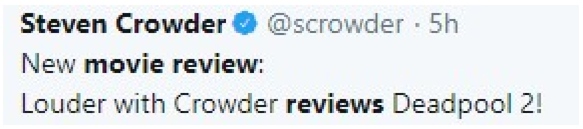
cinemasins @cinemasins · 6h
Deadpool 2 Tweet Review: Well, I loved it. Laughed

2)Negative Tweets:



Justin Davis #PatekTeeth @OGJOHNNY5 · 3h
DEADPOOL 2 is the worst movie I've seen this year

3)Neutral Tweets:

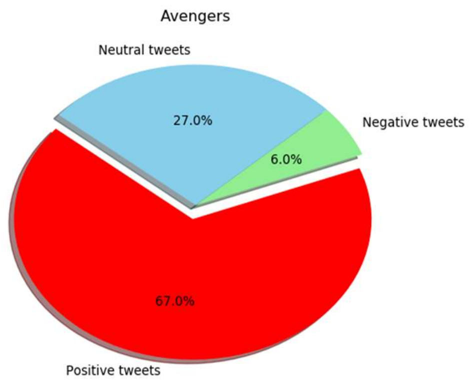


Steven Crowder @scrowder · 5h
New movie review:
Louder with Crowder reviews Deadpool 2!

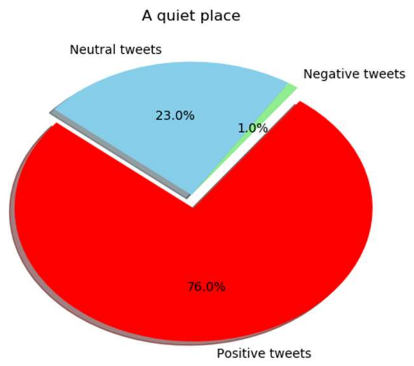
7) Statistical Diagram

Statistical Diagrams are of great use to represent statistical data graphically. They help in analyzing data quickly. It is the best representation of the given statistical data. There are many ways in which statistical data can be plotted in diagrams according to different types of data and representation required. Here, we select pie chart to represent our data because it gives a good idea of what percentage of the tweets are positive sentiment and what percentage are negative.

For the implementation of the Pie Chart we have used pyplot module of the matplotlib library which is very rich library for statistical diagrams.
Following are some of the pie- charts which were plotted on the basis of sentiment analysis of the tweets:
The above pie-chart shows the positive, negative and neutral tweets for the Avengers movie:



The below pie-chart shows data analysis of the sentiment of tweets for movie 'A quite place'.



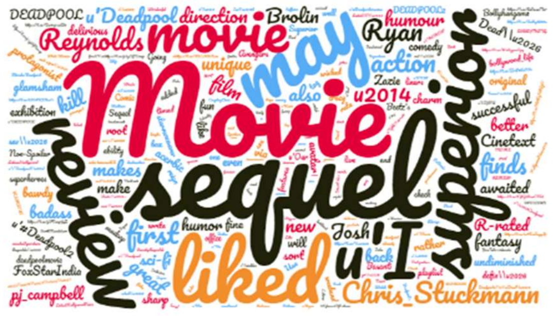
8) Cluster Formation

Here, we have made classification of the tweets using unsupervised clustering algorithm: K-means. K-means is an iterative clustering algorithm. It starts by taking k random points as cluster centers, then takes average of assigned cluster center points. It stops when no assignments change. Here, we take k =3 and form only 3 clusters because we can classify tweets only as positive, negative and neutral. The code snippet for k-means clustering can be seen below:

```
km = KMeans(n_clusters = 3, n_init
= 100) # try 100 different initial
centroids
km.fit(X)
```

9) Wordcloud

Wordcloud is a visual representation which is used to identify trends and patterns in the particular dataset or text file. This trends and pattern would otherwise can remain unclear as it is not easy to analyze each and every review of the movie. Here, common words which may get overlooked in a text file are highlighted and represented bigger in the visual format. Better the frequency of the appearance of a word in the dataset, more highlighted that word is in the visual representation. Let us see one such example of the wordcloud generated from the tweets collected.



Here, the above representation shows wordcloud of a latest released movie- Deadpool 2. This wordcloud was made using tweets of the movie. Hence, one can easily say using the wordcloud that the movie is a sequel of a movie, and the positive keywords like superior and liked shows that the responses on the movie are good and people liked the movie.

10) Dataset

For, further implementation of the module we will require use of dataset which has list of all the movies released along with the genre of the movie. Here, we have a dataset which contains about 30000 movies along with the released year and genre of the movie. The dataset contains movies from all over the world in all languages. The dataset has following attributes:

1) ID – Primary key of the database(Index)
2) Movie Title – This attribute contains the movie name along with the year it was released in.
3) Genre – The genre in which it falls.

So, the dataset has 3 columns and about 30000 rows.

11) Recommendation

Here, we have implemented small module which recommends movie to the user using association rule mining. Association rule mining is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. Here, in this module, we have defined history of some users of watch movies. Here, we have not taken into account the movies but instead a more generalized idea, which is genre. Here, we have defined user history of watching movies of genre. The history looks like below:

```
["Comedy, Family, Drama, Romance",
 "Action, Sport, Sci-Fi, Thriller",
 "Romance, Documentary, Horror,
Action",
 "Horror, Crime, Romance, Sci-Fi",
 "Thriller, Comedy, Documentary,
Drama",
 "Action, Crime, Sport, Romance",
 "Thriller, Sci-Fi, Crime, Action"]
```

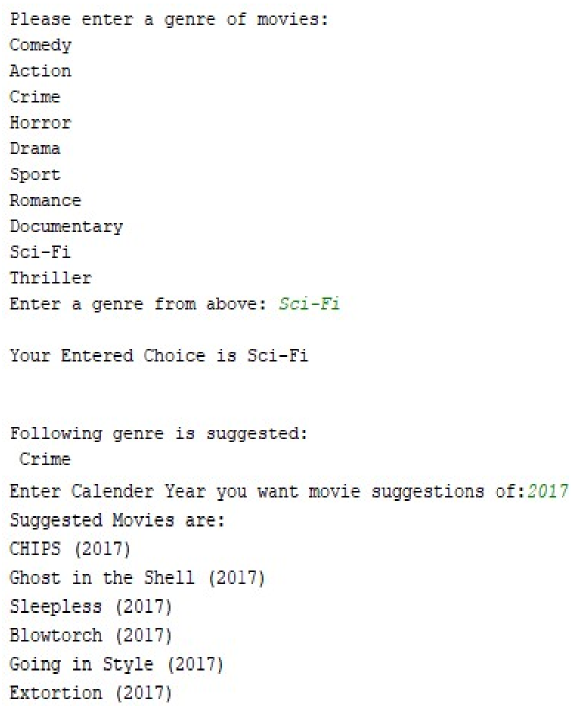
The above is the history of 7 users who watch movies. This history is here defined in terms of genre. This data is subject to raw data which is fed into recommendation module. The recommendation module is solely based on the association rule mining. The association rule mining uses Orange library and makes rules according to confidence and support described in the code. Here, we have chosen support = 0.2 and we take into consideration that association rule which has maximum confidence among all. This is done using loop and assigning value of confidence to a variable if it is greater than the previous value and then lastly taking into consideration only the one rule which has highest confidence out of all. The code snippet for the above implementation can be seen below:

```
for idx, d in enumerate(data):
    if idx is 7:
        print '\nYour Entered
Choice is
{0}'.format(raw_data[idx])
        for r in rules:
            if r.applies_left(d)
and not r.applies_right(d):
                if r.confidence >
maximum:
                    printing_str =
r
                    maximum =
r.confidence
```

Here, we ask user their choice of genre from the list of genres. When a user enters a genre, the implementation of association rule mining takes into account this genre and recommends a genre according to the rules and maximum confidence. Later, the module asks user to enter a year of their choice and then taking into consideration the recommended genre and year entered by the user and using the dataset, the Recommendation module gives recommendation of the movies to the user. The code snippet for the same is as follow:

```
aline = open('movies.dat.txt',
'r').readlines()
result = []
print "Suggested Movies are:"
for x in aline:
    movie_dat = x.split('::')[2]
    s_movie = movie_dat
    if var[2:] in s_movie:
        movie = x.split('::')[1]
        if year in movie:
            print movie
```

The below screen shots show implementation of the recommendation module.



```
Please enter a genre of movies:
Comedy
Action
Crime
Horror
Drama
Sport
Romance
Documentary
Sci-Fi
Thriller
Enter a genre from above: Sci-Fi

Your Entered Choice is Sci-Fi


Following genre is suggested:
 Crime
Enter Calender Year you want movie suggestions of:2017
Suggested Movies are:
CHIPS (2017)
Ghost in the Shell (2017)
Sleepless (2017)
Blowtorch (2017)
Going in Style (2017)
Extortion (2017)
```

Hence, in this way the recommendation module works using the association rule mining.

12) Result and Analysis

Here, we have tried and implemented two modules in two different python files. One module collects tweets and analyzes it and performs sentiment analysis on the collected tweets to rate the movie and to predict the response of the movie on the social media. It also shows this prediction in the form of statistical diagram using pie-chart. We have also made cluster using k-means algorithm and implemented wordcloud to get visual representation of the review of the movie. This helps one in knowing what words are more used to describe the movie.

We have also implemented recommendation module using association rule mining which takes genre input. This entered genre by the user is subject to the raw data/rules which is historical viewership of the genre by different users. Using support value as 0.2 and extracting only the rules with maximum confidence a genre is recommended to the user and the system asks user to enter a year for which the user wants recommendation. Based on recommended genre and year entered by the user, the module recommends a series of movies for a user.

13) Limitation and Challenges

As we are using REST API to collect tweets we can get only a number of 100 tweets in one run of the code after the user enters the movie name. As everything in our module is based on the movie entered by the user so at that point if no previous searches are made for the movie then we have to make analysis based on the 100 tweets. We tried solving this problem by making multiple searches for the same movie and then making it available to the user to get better accuracy in the results, but this is not a potential challenge. Moreover, using the REST API we only get tweets of latest movies only. So, what if one wants to get results of some 10 years old released movie? For that one won't get enough results on social media. A solution to this problem would be to design a potential web crawler which would enable us to collect old historical tweets.

One of the important challenge in working with text file is data-preprocessing. Data pre-processing is challenging yet important task. Other challenge was how to represent statistical data of the movie rating and tweet analysis so that user can have the idea of the data very easily. Plotting the diagram is one such good option. Also, working with large dataset with about 30000 rows is very challenging. We have to focus on black values as well as any unimportant data which might be a part of dataset.

14) Conclusion

Here, we proposed a model which works on the user reviews of the movie which are then combined, and a conclusion is brought. This conclusion about the user review is brought using sentiment analysis of the tweets and combining all the sentiment analysis of a movie. This conclusion is mapped to integer value and a movie is rated on a scale of 0 to 5. A statistical diagram can also be obtained which shows positive, negative and neutral responses on the movie in percentage value. A visual representation of the reviews of the movie is done using wordcloud. Also, we proposed here a recommendation module using genre of the movie based on association rule mining.

15) Future Scope

In future, we can focus on making a web crawler which would then allow us to get old historical tweets. Also, we can make the system more accurate by collecting more tweets and saving the tweets collected for one search and using it for other search after appending the new tweets when different user searches for similar movies again and again. Apart from Association rule mining, other different modules like Logistic Regression can be used for better accuracy of recommendation by combining it with Association Rule Mining.

References

1) C. C. Aggarwal, *Data mining: the textbook*. Cham: Springer, 2015
2) Computational Linguistics & Psycholinguistics Research Center: https://www.clips.uantwerpen.be/pattern
3) Orange Library Documentation : https://docs.orange.biolab.si/3/data-mining-library/
4) https://www.datascience.com/blog/k-means-clustering
5) Sentiment Analysis on Movie Reviews : https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews
6) Bag of words meets bag of popcorn : https://www.kaggle.com/rajathmc/bag-of-words-meets-bags-of-popcorn