Movie Success Prediction Using Twitter Sentiment

Allison Lollo
Project URL: https://github.com/lolloall/Movie-Success-Prediction-Using-Twitter-Sentiment.git

ABSTRACT

Social media contains massive amounts of valuable data that can be used to predict outcomes. This project attempts to create a model that can predict how well a movie will do at the box office based on tweets using sentiment analysis. More specifically it will attempt to predict the IMDB rating. I formulated the problem as a regression problem. To see how well the twitter data can be used to make this prediction this project will compare and contrast two separate models; one model will use only features obtained from the movie itself, the other model will add new features containing information from Twitter. Unfortunately the results of this project were unable to show much support, but with the right improvements(listed in further sections) this project could potentially be very successful.

1.INTRODUCTION

1. In recent years Twitter has largely been a source of commentary on main stream pop culture, more specifically movies. People tend to publicly express their feelings about things on social media platforms. Because of this, this project attempts to create a model for predicting how well a movie will do by combining twitter sentiment data, along with various other features such as genre, cast, and revenue. Sentiment is known as a view of or attitude toward a situation or event. Basically an opinion. Each tweet from Twitter has valuable information on the overall attitude and feeling based on key words that appear in each tweet. By using sentiment analysis we can determine how positive or negative the tweet was. From there if we filter out only tweets that contain the titles of movies we are interested in we can then gain an understanding of what the general mood is surrounding that movie title thus helping us make our overall prediction.

2. The goal of this project is to show that twitter contains valuable information when making predictions. In order to achieve this goal and to show that this is the case two separate models will be created. One model will contain only features about the movie such as revenue and genre. The second model will contain all of the same features but in addition it will add two new features, the average tweet sentiment of the movie title and the number of tweets that contain that movie title.

3. The data collected consists of 4 CSV files and a large set of Twitter data. The CSV files include information on what movies are being used in the prediction, along with any information needed to extract necessary features of each movie. The Twitter data contains tweets ranging from years 2015-2016 that will contain titles of movies that we are interested in.

4. While collecting the data there were several issues encountered. The first being that the twitter data available was only from the year 2016. This proved to be an issue when finding datasets that contained movie data. Most available movie datasets do not contain recent information thus limiting the amount of data that was available to work with. This issue further made the preprocessing step more difficult by having to set a time window of when to collect Twitter data for each tweet. This issue was resolved by comparing the movie release date with the date of each tweet with a window of 2 days after the release date. The next issue encountered with the data was that the datasets being used contained information spread across multiple files. This made the preprocessing step more difficult by having to merge data from multiple sources. By using Pandas Dataframes and the merge functionality this allowed to be able to combined the data. The next challenge was finding a good data source that would allow to actually make the prediction. The question of "what does it mean for a movie to do well at the box-office?" is a hard question to determine. The solution chosen was to tie in a dataset that contained ratings of the movies. The only downside was that these data sets do not contain more recent information, thus limiting the amount of data even further. Another challenge was the overall processing of all of the twitter data. This step took the longest. To get through this the first step was to find a way to most efficiently filter through the data and extract important information that would help in the prediction. After several attempts all the data was picked through and put put into text files that was used for further preprocessing. Unfortunately, with twitter data, there is so many ways an individual can say something. For example if a full movie title is "Batman v Superman: Dawn of Justice" in reality people may only reference it by "Batman v Superman". Finding different variations for hundreds of movie titles proved to be very difficult. To remedy some issues movie titles were checked for all upper case, all lower case, and title case.

5. The project was hoping to show that using twitter data could help improve the results of the initial model. Unfortunately, this was not the case. The data remained relatively the same with slight variation.

2.DATA

In total this project used data from 4 different datasets. The first dataset used was a dataset collected from the site Kaggle called "The Movies Dataset"(URL: https://www.kaggle.com/rounakbanik/the-movies-dataset) From this dataset two CSV files were used. The first CSV file used was the "movies_metadata.csv" file which contained valuable information such as movie title, release data, revenue, genre, and budget. The next file was the "credits.csv" file which contained information on the cast such as main actor, director, and producer. Once the information from both of these files were pulled they had to be combined by using an ID value specific to each movie. The next dataset used was the "IMDB Movies Dataset" collected from Kaggle(URL: https://www.kaggle.com/orgesleka/imdbmovies#imdb.csv). This data set was a CSV file containing information on the IMDB rating of movies up to the year 2016. This information was put into a Pandas dataframe and then combined with the larger dataframe on the condition that the titles were the same. The next dataset used was called "Academy Awards Oscars: Nominees and Winners 1927 to Present" from the site datahub.io(URL: https://datahub.io/rufuspollock/oscars-nominees-and-winners#resource-oscars-nominees-and-winners_zip). This CSV file contained information on names of all winners and nominees. From this data a count for the main actor and director of each movie in the over all dataframe was collected which represented how many awards they each won or were nominated for. This information was then added to the over all data frame. The last source of data used was the Twitter data collected from the year 2016. By searching tweets for movie titles two features were gained, the average tweet sentiment for each movie title and the overall number of tweets that spoke about each movie title. This information was then added to the overall data frame.

The data collected is all centered around the year 2016. This is due to the fact that the twitter data is from the year 2016. The raw datasets had to be filtered down to only movies that had a release date in the year 2016. The raw movie datasets were very large containing data on over 45,000 movies but this spanned many years. So in order to use the twitter data this raw data had to be skimmed down. Unfortunately there is not many datasets containing information on movies post 2015 so the final data set ended up being around 250 movies in total. In order to get an accurate representation of how well a movie does during its release, the twitter data had to be filtered to only tweets that were within a 2 day time window of the release date of each movie. This allowed to be able to see how much "buzz" was going on about the movie following its release. All missing data had to be discarded due to the nature of the data. The final data frame for the model with out the twitter data contained 6 features and the final data frame using the twitter data contained 8 features.

The majority of the time working on the project was spent in the preprocessing stage. Starting with the first file "movies_metadata.csv". Although this file was relatively simple to process the first thing that had to be done was to get only movies released in 2016. This required having to slice the date string and extract the year to ensure it was the correct year. One step that also needed to be done with this file was to calculate the gross revenue or profit. This data set contained the budget as well as the revenue so the gross revenue was then calculated. The next step that had to be done was obtain the genre. Each movie didn't just have one genre, there was a list of multiple genres as a string. Instead of using all genres, the first genres was selected as the main genre for that movie. This once again required parsing through the string to find the first genre listed. The next file to work with was the "credits.csv" file. This file was more difficult to work with because all of the data was stored as one consecutive string format. Trying to use JSON ended up not working so in order to obtain the information that was needed I had to parse though the string and find key words that would give me the information I needed. These key words included finding 'cast', what would follow that is all the actors of the movie, however I was only interesting in the first actor listed. The next thing that needed to be searched for was "'job': 'Director'" which would then follow with the director for the movie. Once these two files were preprocessed and put into data frames they then had to be combined. They were combined on an ID value that corresponded to each movie. This then created one larger data frame. The next file "IMDB-Movie-Data.csv" only needed to be filtered through by only obtaining movies with a release date in 2016 and then putting this new information into a new data frame. This new data frame and the other larger data frame then had to be combined but this time combining them on matching movie titles. The next step in preprocessing was collecting information from the "awards.csv". This involved having to iterate through the movies in our large data frame, and then looking through the rows in the awards file and to keep track of a count for both the main actor and the director as to how many awards each of them have either won or been nominated for. Once this information was found it was added to our larger data frame. Because this is a regression problem, the genre field had to be encoded making it a numeric value. This involved encoding the genre into binary versions where there would be a 1 for the genre it was and 0 where it wasn't. There were 15 genre types in total which in turn added 15 new columns to the overall data frame. The last preprocessing step involved working with the twitter data which was the longest step. The first step was to actually obtain the tweets of interest. This was achieved by writing a script that would iterate through tweets in each file, then check if the tweet contained any key words. Key words being movie titles we were interested in. Other variations of movie titles were also checked such as all upper case, all lowercase, and title case. These collected tweets were broken up in to 4 separate files. After all the data was collected it had to be further filtered by ensuring we had tweets only from a 2 day time window of each release date for the movies. This was accomplished by converting the tweets timestamp and the movies release date to Datetime objects and then using Timedeltas to create a time window. Once the tweets were fully filtered the python module TextBlob was used to actually calculate the sentiment of each tweet along with a count of the number of tweets for each movie. These two new features were again added to the large final data frame.

There were two separate Dataframes used, one with out the Twitter data and one with the Twitter data. There are 6 predictor attributes being used for the first model. Those features are budget, genre, gross revenue, number of votes, main actor award count, and director award count. The second model used had all of these same features along with two new predictor attributes of average tweet sentiment and number of tweets. This problem was a regression problem so the response attribute that we were trying to predict was the IMDB rating of the movie. This is, a rating that scales from 1-10.

After preprocessing the final table contained 8 predictor attributes; budget, genre, gross revenue, number of votes, main actor award count, director award count, tweet count, and average tweet sentiment. The last column is the response attribute which is the IMDB rating. Unfortunately because of the limited data available

the final table was relatively small. In total there are 233 rows and 23 columns.

3.METHODOLOGY

This problem as a whole was predictive modeling that utilized regression. This project utilized a linear regression model from Python's Scikit-Learn package in order to create the predictions. The data was divided into 30% test and 70% train. The overall workflow is as follows

1. Find useful datasets and data to be used.
2. preprocess data and create Panda dataframes that contain the relevant information we need to make our prediction
3. combine all the dataframes into one larger table
4. with the final table divide data into training and test sets using sklearn
5. train the model using linear regression model.
6. apply the model on the test set
7. Use root mean square error and R-sqaure values to evaluate the model and its performance
8. plot final results
9. Collect and preprocess twitter data. Combine twitter information into the over all table
10. repeat steps 4-8 with the new table that contains the twitter data
11. Compare results from both models.

Below is a list of files that are used in the creation of this project that collects, preprocesses, and creates the model:

- step1.py: this is the python file that was used to filter through the twitter data.
- preprocessing_with_twitter.ipynb: this is the Jupyter notebook file that is used to preprocess and prepare a model for the table containing the twitter data.
- preprocessing_no_twitter.ipynb: this is the Jupyter notebook file that is used to preprocess and prepares a model for the table that does not utilize the twitter data.

The software used for this project included Anaconda's Jupyter Notebook. In addition, numerous python modules were used including Pandas, datetime, TextBlob, numpy, and sklearn.

4.EXPERIMENTAL EVALUATION

This section describes the experimental setup and results obtained.

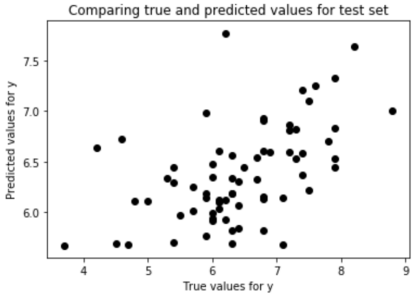
4.1.Experimental Setup

1. This project was built using a Mac OS.
2. In order to test whether this project does in fact show that twitter data helps to improve the prediction of the IMDB rating a baseline model was also created. This baseline model contained all of the same features just without using the twitter information.
3. The evaluation metrics that were used were root mean square error and R-sqaure values.

4.2.Experimental Results

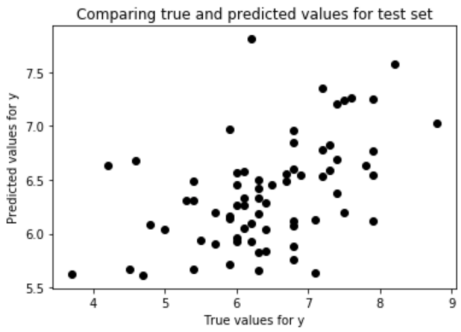
This project involved the creation of a prediction model that would predict the IMDB rating of a movie. A linear regression model was built using Sklearn package of Python. The data was split into 30% train and 70% test. In order to see the performance of the model the Root Mean Square error and the R-square values were calculated. The coefficients for each variable are also displayed, along with the intercept value. The data was also plotted into a scatter plot. Two separate models were created, one with twitter data and one with out. The values of the baseline model without twitter data is shown below, along with the values of the model with the twitter data. The scatter plots for each models are also shown below.

1. Results from the model not containing Twitter data.




```
Root mean squared error = 0.8439
R-square = 0.2626
Slope Coefficients: [-7.16293105e-10  6.63550529e-10  4.40784496e-06  5.11057870e-02
 -1.70606049e-02 -3.97369970e-01 -1.13185559e-01  7.88338885e-02
  2.19856237e-02  2.37723113e-01  5.17597936e-01 -2.49886092e-01
 -3.45363247e-01  3.50815684e-01 -4.19697207e-01 -6.93929652e-02
  1.21476121e-01 -3.10056798e-01 -1.69265795e-01  7.45785264e-01]
Intercept: 6.0803353840325505
```

2. Results from model containing Twitter data




```
Root mean squared error = 0.8546
R-square = 0.2438
Slope Coefficients: -9.537172549292599e-10
```

As shown from the results this project was unfortunately unsuccessful. The goal was to show improvements in the analysis of the two models. This would mean that it would have to show that the Root mean squared error value had decreased and the R-square value increased. However, this was not the case, the values slightly changed between the two models but the Root Mean square error value slightly increased and the R-square value slightly decreased. But for the most part they remained fairly the same. There were many reasons that this project proved not to be successful and there are many improvement that could help make this a successful project, these improvements are outlined below in the Conclusions section. But to briefly discuss, the first issue was with the limited amount of data. It is very hard to successfully train a model of this scale with such a small amount of data. I am sure that with much more data, these features chosen would be able to create a good predictor model. The other big issue is with the handling of the twitter data. Spending more time working with the twitter data and finding different alternatives to determining what tweets are spoken about each movie would need to be done.

5.CONCLUSIONS

This project provides the foundation of what could be a potentially very successful project. Although this project failed to achieve the goal, it could however be achieved with more time and more data. It provides a workflow of how to go about setting up a dataset to be used for this regression problem. This project could definitely use many improvements. One big improvement would be utilizing datasets that have more information. Because such limited information is available about movies from the year 2016 it was hard to make an accurate prediction. The next major improvement would be the filtering of the twitter data. Having a way to isolate what is a movie title and what isn't was a difficult part of this project. For example with a movie titled “Sing” Many twitter users could be tweeting this word without the association of the actual movie itself because it is a common word or phrase. Having a way to only use movies that aren’t common words or phrases would help the results of this project and filter out faulty information. The next improvement from the twitter perspective would be to have more variations of movie titles. Titles such as “Batman v Superman: Dawn of Justice” could be split to also recognize that “Batman v Superman” alone is also the same. The last improvement that could be made is to do a more thorough analysis of the features selected and ensure that there is no correlated attributes that are interfering with the results.

6.REFERENCES

1. Banik, R. (2017, November 10). The Movies Dataset. Retrieved from https://www.kaggle.com/rounakbanik/the-movies-dataset
2. Leka, O. (2016, November 15). IMDB Movies Dataset. Retrieved from https://www.kaggle.com/orgesleka/imdbmovies#imdb.csv.
3. Datopian. (n.d.). Academy Awards Oscars: Nominees and Winners 1927 to Present. Retrieved from https://datahub.io/rufuspollock/oscars-nominees-and-winners#resource-oscars-nominees-and-winners_zip