## **Machine learning and predective analysis with movie data**

Author: Jinhang Jiang

## Introduction

This machine learning project is the final project for the Business Analytics Capstone Class (BSAN 480, INSTR: Chris Claterbos) at the University of Kansas School of Business. The purpose of the project is to study patterns and derive insights from the Movielens data and IMDB data to predict the performance and the rating of a movie in the future.

## Scope of Work

For the main concentration, the project will focus on performance analysis and prediction. I firstly cleaned, created, and then analyzed the given tables (Appendix A-1 & A-2) to understand patterns and stories that otherwise may have gone untold. Secondly, I created a tag table with 7,075,944 records (Appendix A-3) and did a sentimental analysis with the data. I also created some dashboards to visualize the significant findings. Finally, I tried and compared several different predictive models to forecast profitability performance, as well as the ratings.

I used SQL, R, and Python as the programming languages. I used packages and algorithms, such as pandas, numpy, Node2Vec, NetworkX, and K-means for analysis. RStudio, Jupyter Notebook, Oracle Developer were used as IDE, and Oracle Analytics Cloud was used to create dashboards.

## Data

* movie\_new.csv

This table contains 17,071 English-speaking movie records from the past 114 years, including some important features we are looking for, like movie ID, revenue, budget, release data, cast information, rating, genres, and so on. Excluding the tag table, all the other tables that focused on some specific feature analysis were generated from this table (Further details in Appendix A-4).

I did a log transformation for budget and revenue, respectively, since they are too big in numbers and may create outliers when we develop predictive models. I broke the column genre into a dummy matrix for both model development purpose and network analysis.

Since some movies do not have many votes (perhaps because they did not have a Facebook account or something) while the other of them have a huge number of votes, it would not be reasonable considering it directly as a variable in the model. So, I also did a tree cut transformation and broke it into three groups.

* genre\_matrix.csv

This is the transformed dummy matrix of the genre. It has 17,071 rows and 21 columns. Column 1 is the movie ID, and column two 2 to 21 are all the types of genres that are listed in the original table. The output looks like the following:

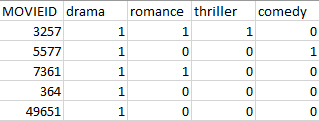


Figure 1 - Partial View of genre\_matrix.csv

In Figure 1, we can see that the table is populated with 1s ("equal to 1" means the movie has this genre listed) and 0s ("equal to 0" means the movie is not in this genre). Take "movieid = 3257" as an example: this film's genres are drama, romance, and thriller.

* tag.csv

It includes movies id, tag, and relevance score, which has 7,075,944 observations. The relevance score ranges from 0 to 1, representing how strongly movies exhibit particular properties represented by tags.

## Analysis Process

#### **Analysis and some sample queries:**

At the beginning of the analysis, to study and understand the data better, I used SQL (Appendix A-5) to produce some basic information about the movie data.

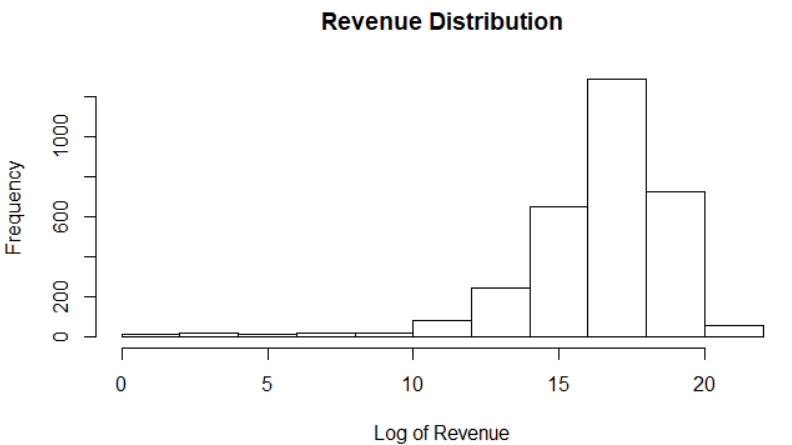
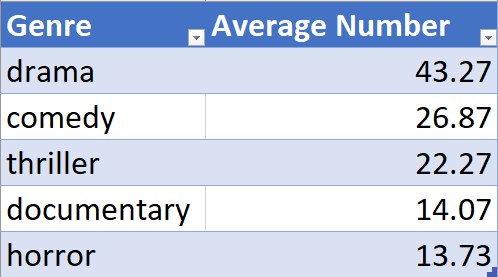
 

Figure 2 - Log(100 million) ≈ 16.1 Figure 3 - Top five genres released annually

Over the most recent 15 years (01/01/2015 – 12/31/2019), the average number of released English-speaking films is **105.64**. And the average number of the movies with revenue over $100 million is **8.5**, which means over the last 15 years, approximately eight out of every 100 films annually would have revenue exceeding $100 million.

The five most popular genres are drama, comedy, thriller, documentary, and horror. According to the analysis of the genre table, the documentary is the 11th in the total numbers of the dataset, meaning documentary must have become increasingly popular over the past 15 years.

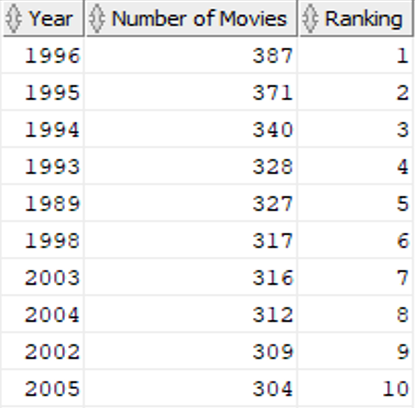


Figure 4 – 90s, Cinema’s fairytale decade

Figure 4 shows that the 90s take up half of the top ten (four of the top five) most productive years in the past 114 years of cinema history. The reason why the cinema industry was booming in the 90s could be that the low-budget independent films unceasingly rose and maintained their popularity in the industry within the decade. Also, in the late 90s, Netflix started to offer rental DVDs service, which could be another reason. Moreover, with the introduction of the Internet and personal computers, it may shape the consumers' movie-watching practices and stimulate the industry as well. More information can be found in Appendix A-6.

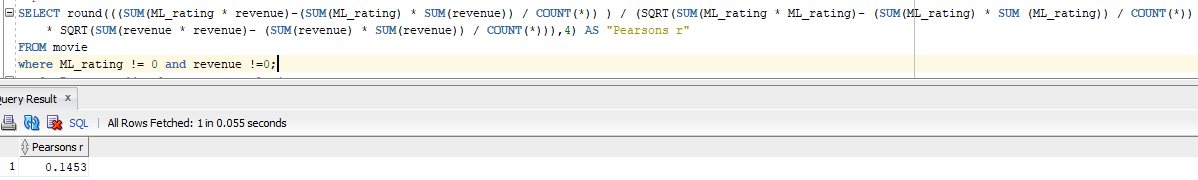


Figure 5 – Rating and Revenue aren’t strongly correlated

According to the definition of Pearson’s Product-moment Correlation Coefficient, the test gives a measurement from -1 for a perfect negative correlation to 1 for a perfect correlation. And a correlation of 0 means that there is no relationship between the two. Given the result is **0.1453** (Figure 5), I do not think there is a strong link between rating and revenue. It could be due to the range of the time in this dataset is too large, and the amount of revenue for an individual movie has significantly increased over the years.



Figure 6 – Top ten most profitable movie in history

I also did a quick analysis (Figure 6) of profitability by looking at the profit margin. Some very famous movies are on the list, like Modern Times. These movies are not necessary to be the best movies in the history, but they are definitely worth some further study if you are interested.

#### Dashboards

#### Network Analysis of Genres

To study and predict the genre combinations of top rated movies, I decided to do a deep learning network analysis, using the algorithms of Node2Vec and K-means.

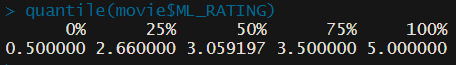


Figure 7 – Top rating cuts off at 3.5

To find out top rated movies, I firstly did a quick check of the quantile of the ML\_Rating (Figure 7) and decided to cut at 3.5, where is top 25% (4335 records). Then I created a “top rating adjacency matrix of genre” (Appendix A-7) that looks like the following:

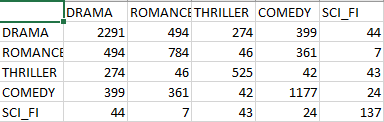


Figure 8 – Adjacency Matrix

Then I use the Node2Vec package in python (Appendix A-8) to generate a set of k-means features for the selected genre data. Then I used R (Appendix A-9) to do the rest k-means analysis and plotting.

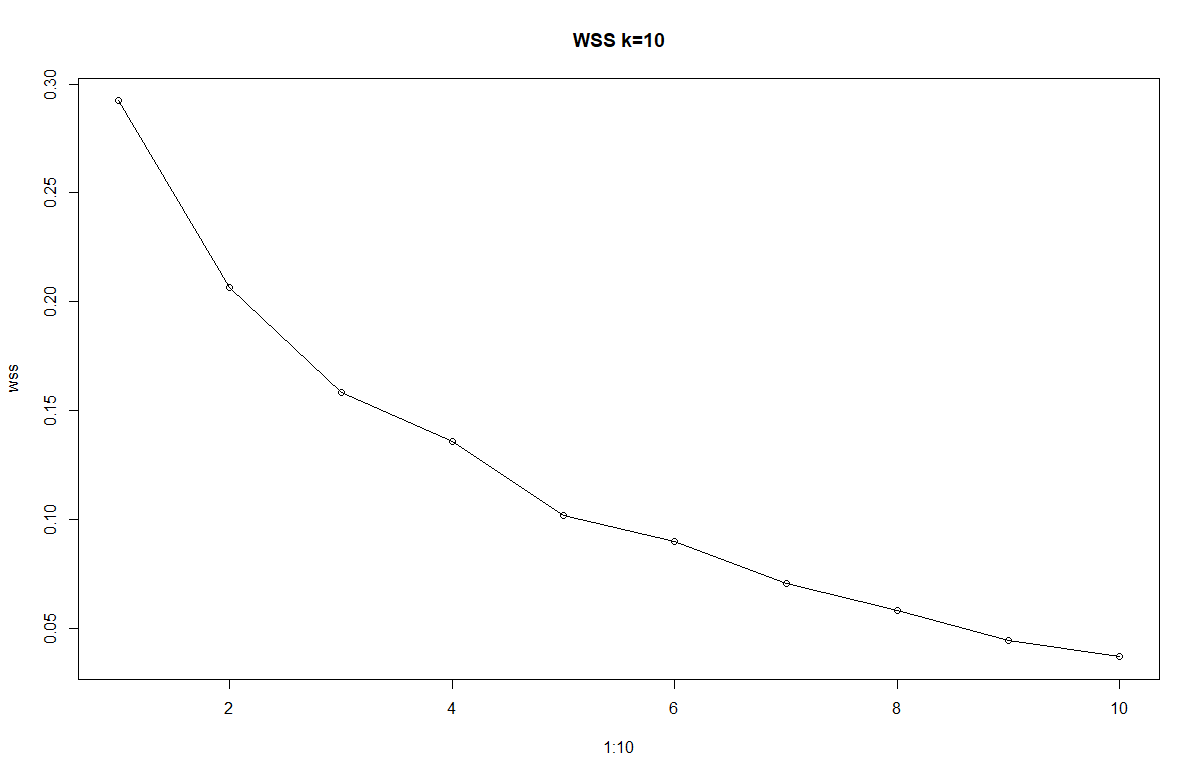
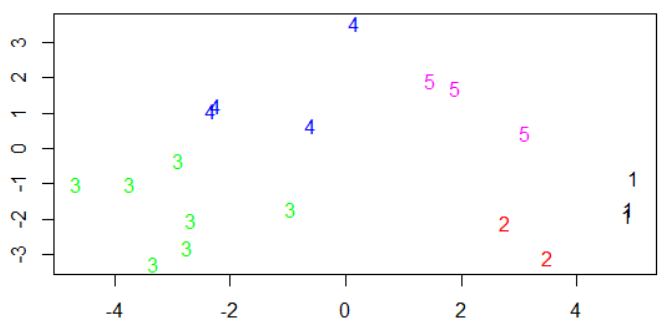
 

Figure 9 - Elbow curve shows at 5 Figure 10 – K-means plot

Based on Figure 9, I decided to use k=5 since it is where the elbow curve shows. And the final plot looks like Figure 10, which is pretty neat and clear. I ran a subsample afterward to gain insights about the popular genre combinations for top-rated movies.

The genre combination of the first subsample group of top-rated movies is composed of “**drama, comedy, animation, and children**.”

The genres of the second group are “**drama, thriller, and IMAX**.”

The genres of the third group are “**drama, adventure, action, mystery, war, documentary, and film-noir**.”

The genres of the fourth group are “**drama, romance, musical, crime, and fantasy**.”

The genres of the fifth group are “**drama, sci-fi, western, and horror**.”

By studying the popular genre combinations above, movie producers will be able to make a better decision on what kind of film scripts they should pick. And this model also can be applied to an advanced recommendation algorithm for streaming platforms, like Netflix. For example, if a customer watched a lot of movies with genres in drama and comedy, it will be reasonable to recommend the customer some animation or children movie.

#### Natural Language Processing Analysis

In this step, I was very interested in studying the relationship between the TAGs and the high Relevance scores. A quantile test shows that the top 10% relevance scores cut off at **0.2905**. So, I subsampled the TAG table, and get a new table with 707,734 records.



Figure 11 – Word Clouds I

Figure 11 shows that there are some common and meaningless words, like “good”, “great”, and “best”. I removed some of them to help gain a better understanding.

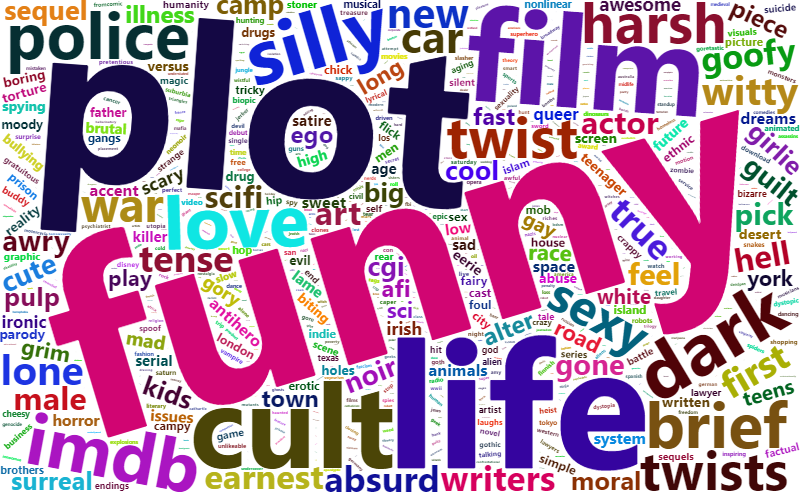


Figure 12 – Word Clouds II

Figure 12 looks like there are plenty of emotional words and expressions, so I believe it is worth a sentiment analysis.

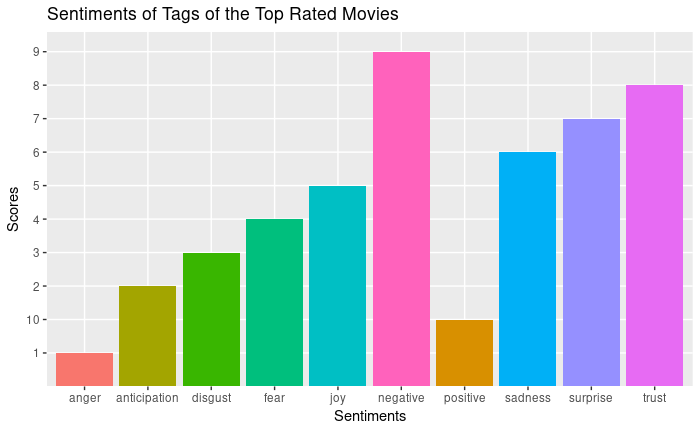


Figure 13 – Sentimental Analysis Plot

Surprisingly, the final output for the sentimental analysis (Figure 13) for the high rating movies suggests that the sentiment of “negative” has the highest score, and sentiment of “joy” is in fifth place. Perhaps, in this case, a quite portion of the top rating movies are associated with the negative attitude, or maybe the sentiment analysis mistook the “intense” or something similar for “negative.”

#### Model Development and Comparison

## Results

What insights and results were found.

## Conclusion

Summary of work and insights / recommendations.

# APPENDIX A

You may check the partial codes and relevant tables on my [GitHub](https://github.com/jinhangjiang/BSAN480-FinalProject).

1. [SQL code](https://github.com/jinhangjiang/BSAN480-FinalProject/blob/master/Scripts/MovieTable_sqlcommand.txt) for cleaning and creating the original table
2. [R code](https://github.com/jinhangjiang/BSAN480-FinalProject/blob/master/Scripts/Model_analysis.R) for modifying the movie table
3. [SQL code](https://github.com/jinhangjiang/BSAN480-FinalProject/blob/master/Scripts/Tag_creation.sql) for creating the tag table
4. [Details and references](https://github.com/jinhangjiang/BSAN480-FinalProject/tree/master/Reference) for the tables
5. [SQL code](https://github.com/jinhangjiang/BSAN480-FinalProject/blob/master/Scripts/FinalProject.sql) for the basic analysis and queries
6. “[Film History of the 1990s](https://www.filmsite.org/90sintro.html)” by Tim Dirks
7. [SQL code](https://github.com/jinhangjiang/BSAN480-FinalProject/blob/master/Scripts/Top_Rating_Genre_Matrix.txt) for creating the top\_rating\_genre\_matrix
8. [Python code](https://github.com/jinhangjiang/BSAN480-FinalProject/blob/master/Scripts/Movie_genre_network.py) for generating Node2Vec features of genres
9. [Code](https://github.com/jinhangjiang/BSAN480-FinalProject/blob/master/Scripts/Genre_networks.R) for creating genre networks in R
10. [Sentiment Analysis Code](https://github.com/jinhangjiang/BSAN480-FinalProject/blob/master/Scripts/NLP_Analysis_Report.pdf) of the tags in R