PREDICTING MOVIE PREFERENCE

Summary of Findings and Description of Research

By: Jared B. Andrews, William A. Bates, Erik D. Mumm, and Cole J. Richmond

UC San Diego

December 2nd, 2019

Julian McAuley

Abstract

This paper proposes a recommendation system for movies based on a user's film preference. Our model's success is measured by comparing the predicted film preference with the actual responses of users, leveraging historical user behavior data. Features included movie genre, popularity, production company, and similarity to other movies preferred by the user. Experiment results with movies from the MovieLens dataset show our model to achieve a 95% accuracy and an 80% balanced accuracy rate in predicting user preference.

Background and Introduction

In a world of Rotten Tomatoes, Metacritic, and movie review sites alike, actually reading whole written reviews and reflections has gone by the wayside for people looking to critics for whether or not to see a film. Nowadays, it's all about numbers, ratings, and what's "Recommended for You". While seeing a number or some stars doesn't tell you much, it tells you something. Unlocking the hidden meaning behind what's encoded in the ratings allows us make decisions about user's preferences when it comes to watching movies.

A number of websites and forums allow users and movie goers alike to submit movie reviews and aggregate them into an average. Review sites have allowed the common movie lover to express their opinion on films. Many of these sites allow users to rate films on a certain scale and their votes are then culled into an overall rating and ranking for any particular film. Some of these community driven review sites allow users to register and submit reviews. This means that they are a form of open access poll, and have the same advantages and disadvantages; notably, there is no guarantee that they will be a representative sample of the film's audience. In some cases, online review sites have produced wildly differing results to scientific polling of audiences. Likewise, reviews and ratings for many movies can greatly differ between the different review sites, although there are certain movies that are well-rated (or poorly-rated) across the board. By using a stable benchmark dataset, we hope to quell any sort of uncertainty that comes with the variable nature of movie reviews.

Dataset Description

Our group chose to use the ‘MovieLens 20M Dataset’ for this assignment. MovieLens is run by GroupLens, a research lab at the University of Minnesota that develops new experimental tools and interfaces for data exploration and recommendation. The ‘MovieLens 20M Dataset’ contains 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users. The dataset includes tag genome data with 12 million relevance scores across 1,100 tags.

`rating.csv`

All ratings are contained in this file. Each row in this dataset represents one rating of one movie by one user. Ratings are made on a 5-star scale, with half-star increments (0.5 stars - 5.0 stars). The distribution of all ratings in the dataset is as follows:

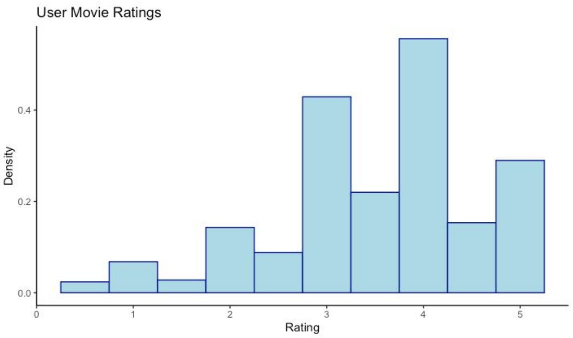

Figure 1: Histogram depicting the distribution movie ratings

The average rating among all movies in the dataset is 3.525529 with a standard deviation of 1.051989. It is clear that users favor giving out whole-star ratings to movies as opposed to half-star ratings. Our group was curious to figure out why this is the case and see if we could discover any trends among users that prefer half-star ratings over whole-star ratings. We can also see user ratings broken up by the month the rating was given in:

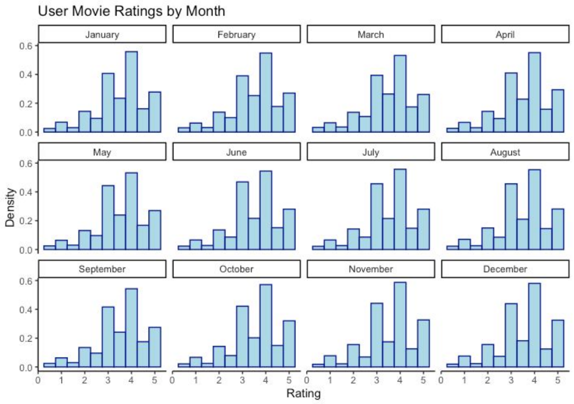

Figure 2: Collection of histograms depicting the distribution movie ratings by month

This visualization doesn't reveal anything significant in terms of how a user rates movies throughout the year, but yet we still see the same trend in whole-star and half-star ratings.

`tags.csv`

Each line of this file represents one tag applied to one movie by one user where tags are user-generated metadata about movies. Each tag is typically a single word or short phrase where the meaning, value, and purpose of a particular tag is determined by each user. There are 38,643 unique tags in this dataset. To get an idea of which tags were the most common we plotted those words/phrases with a frequency greater than or equal to 1,000:

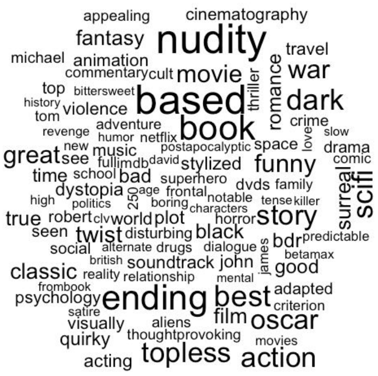

Figure 3: A tag cloud of top user-assigned tags for all movies

It seems as though users apply a tag to a movie based on either what was most salient to them (e.g. soundtrack, nudity, aliens) or what the overall theme of the movie was (e.g. sci fi, fantasy, war).

`movies.csv`

Information about all 27,000 movies is contained in this file. Movie titles were entered manually or imported from https://www.themoviedb.org/, and include the year of release in parentheses. The break-up of types of genres represented in this dataset is as follows:

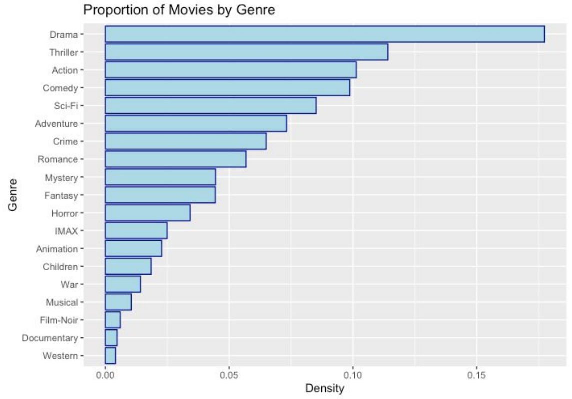

Figure 4: Ordered bar chart depicting the make-up of genres

Those movies that are considered to be of the genre ‘Drama’ make up 17.7% of the dataset with ‘Thriller’ (11.4%), ‘Action’ (10.1%), and ‘Comedy’ (9.9%) rounding out the top four.

`links.csv`, `genome-scores.csv`, `genome-tags.csv`

These three detests serve as complements to the three above. The `links.csv` dataset contains unique identifiers that can be used to link to other sources of data for a particular movie. The `genome-scores.csv` and `genome-tags.csv` datasets contain information regarding how strongly movies exhibit particular properties represented by user-generated tags (atmospheric, thought-provoking, realistic, etc.). The tag genome was computed by the GroupLens research lab who used a machine learning algorithm on user-contributed content including tags, ratings, and textual reviews.

Predictive Task and Model Development

Model Motivation

Many people rely on critics to gauge the quality of a film, while others use their instincts. We identified that a majority of ratings cluster in the high-ratings area with a small amount of ratings appearing in the low-ratings area. Therefore, we are interested to see what qualities those low-rated movies possess that cause audiences to frequently give them less than favorable reviews. Moreover, we would like our model to capture a user’s genre preferences which, in turn, may give us an idea of whether or not they would prefer to watch a new movie presented to them. With information about users’ movie preferences and the movies themselves, our group was curious to see how accurately we could predict a user’s preference for certain movies in the form of a star rating.

Predictive Task

We wanted to try our hand at one of the most frequently-attempted problems in data science: building an effective recommender system for movies. Specifically, we compressed the user ratings into ones, if the rating was at least 3.0, and zeros, if the rating was less than 3.0, and set out to predict whether or not a user would “like” a given movie (ie. give it a rating greater than or equal to 3.0). The model will be evaluated using a simple accuracy, taking the number of correct predictions over the total number of predictions, in addition to a balanced accuracy measure. The balanced accuracy sums the proportions of correct predictions for each label and divides by the number of labels.

Given a choice between two options, the most naive classifier would randomly predict either ‘one’ or ‘zero’ for every data point, yielding an accuracy of 0.5. The next most naive baseline which we could consider is the predictor that predicts the most common rating at every point, regardless of attributes. As about 88% of the ratings were ones, the predictor would predict ‘one’ for every point and return 88% accuracy. This would be a good baseline to use to determine whether the features of the data are at all useful. The baseline balanced accuracy of the random classifier would also be 0.5, and the baseline balanced accuracy of the mode classifier would be 0.5 in this case. Therefore, when validating our model, we should be looking for accuracy scores of greater than 0.88 and balanced accuracy scores greater than 0.5.

Model Selection

We felt that a binary classifier would best represent our task at hand. When people are deciding to watch a movie or not, they want to know whether they will have a positive or negative experience watching it. With numeric rating systems so variable and user-specific, we felt that this problem would best be approached through binary classification. As such, learning models that we considered included SVMs, logistic regressors, and random forest classifiers. The specific model decision would be decided based on our available data.

The features available to us were many. Some were more easily accessible, such as genre, but would need to be encoded to be used in the model. Also generally available in the starting dataset were the userId, movieId, and user’s rating of the movie. As noted earlier, the rating variable was processed to compress it from 0.5-5.0 at intervals of 0.5 to a binary 0/1 variable.

Production company was pulled from the Movie Database API, as were budget and release date. Budget was inputted as-is, while production company was one-hot encoded. The release date was then separated into month and year and one-hot encoded. These were the given features, though we also developed additional features using techniques such as sentiment analysis and similarity, which will be discussed in the next section.

We knew that we would need a classifier that is effective in binary situations with unbalanced labels as well as capable of handling many features. Due to many categorical variables in the data, such as genre and production company, there would be a large number of columns as a result of one-hot encoding. For these reasons, we chose to use a Random Forest Classifier, which is generated from combining the results of numerous decision trees made from certain features of samples of data points. This classifier is particularly effective in cases of data with many features, due to its separation of the features to reduce tree correlation, and cases of severely imbalanced labels.

Model Development

In the creation of our model, we took inspiration from models developed in previous homework and from previous categorization machine learning models found in our research. Similar to our book/user recommendation model in homework #3, we developed a “movie popularity” feature. We considered the most watched movies that made up 50% of total movie watches in the dataset.

To try and quantify similarities between user/movie pairs, we attempted to calculate the maximum, minimum and mean of the cosine similarities between both users that had watched a particular movie and movies that had been watched by a particular user. Due to the sheer size of our dataset, this cosine similarity calculation exceed the maximum runtime and we were unable to use the cosine similarity statistics. As a result, we used Jaccard similarities to capture such similarities. For each user/movie pair, let the user be u1 and the movie be m1, we took the set of movies that the u1 had seen, let this set be M, and the set of users that had seen m1, let this set be U. For each movie in M, let this movie be m2, we calculated and stored the Jaccard similarities between the set of users that had seen m2 and U. After doing so, we took the maximum, mean, and minimum of the resulting set of Jaccard similarities as features for the user/movie pair. We calculated a similar set of Jaccard similarity statistics between movie sets, creating an additional three features.

To utilize the movie tag in our dataset, we employed the VADER sentiment analysis tool to capture capture the tag's sentiment. Running each tag through the VADER Sentiment Analysis function, we were given a polarity score which the positivity of a tag. Using thresholds found in VADER research [5], we encoded each tag as positive, neutral, and negative with values of 1,0 and -1, respectively.

To choose which classification learning method to employ, we look at our dataset. Since many of our features are categorical, e.g., production company, popularity feature, month, our data can be assumed to be nonlinear. In addition, our data is high-dimensional with nearly 550 features and is unbalanced as roughly 88% of the reviews are classified as positive. Considering this, we choose to employ a Random Forest Classifier in our model. The Random Forest Classifiers is a nonlinear classifier, as opposed to Linear Regression and SVC which are both linear classifiers. Furthermore, Random Forest Classifiers don't overfit as the number of trees increases and are able to fit unbalanced data [7] by specifying the class weights parameter when initializing the model.

To perform hyper-parameter tuning, we employed random search cross-validation. This cross-validation method involves setting up a grid of hyperparameter values and selecting random combinations to train the model. The best model was then selected as our final model.

Literature Review

Existing Dataset Related Literature

As mentioned briefly in the Introduction of this report, this project utilizes an existing dataset of movie reviews and ratings. This dataset was made public by GroupLens, which is a research lab in the Department of Computer Science and Engineering at the University of Minnesota. GroupLens created MovieLens, which is a web-based recommender system and user community designed to recommend movies to users based on their film preferences. The MovieLens datasets have been published and updated since 1998 and are downloaded hundreds of thousands of times each year.

MovieLens has several data sets available for new research and analysis. The 20

Million review dataset is recommended and includes reviews of more than 27,000 movies by 138,000 users. Our team modified this dataset only slightly; we one-hot encoded the genre and month features, created the like/dislike column based on the rating column, and removed rows that contained null values. After doing so, our dataset consisted of about 180,000 rows. To supplement this dataset, our team then used The Movie Database's (https://www.themoviedb.org/) API to acquire data on the production budget, the production company, the number of reviews, and the release year for each movie. Upon further review, we found that many movies had budgets of $0 and other very small values. As a result, production budget was not utilized in our final model. In addition, there were hundreds of production companies, many of which had produced less than 10 movies. When one-hot encoded, this feature would produced a massive, sparse matrix. To deal with the sparsity, we only kept production companies that had produced over 50 movies. Using these roughly 250 production companies, we one-hot encoded the production company feature.

Related Datasets

As mentioned above, the MovieLens dataset our team downloaded is used and/or referenced in hundreds of studies each year. Most of these studies and downloads are generated by users for educational purposes. Examples of education-driven efforts and analysis can be found on Kaggle.com. Students and researchers have used the dataset to gather information and statistics on the industry, to complete personal research projects, and to build basic recommender systems. An example set of research questions, developed by Yogesh Patil [3], is as follows:

- What are the Top 3 movies by Occupation of users?
- What are the Top 3 movies in each Genre of movies?
- What are the Top 3 movies for each Age Group of users?
- What are the Top 3 genres released in Summer (May – July)?

Looking beyond MovieLens, there is an abundance of research on the entertainment industry, specifically focused on movies. Many of these studies use IMDb (Internet Movie Database) as their dataset, as it is a well established platform for movie reviews in the industry. One such study was conducted by Karl Persson at the University of Skoevde, titled: *Predicting Movie Ratings: A comparative study on random forests and* *support vector machines* [1]. This study analyzed the results of recommender systems “using pre-release attributes such as cast, directors, budget and movie genres” [1]. The results of the study found consistently better performance from the random forests over the support vector machines, but the authors did note that the results were very similar and did not deem these results conclusive.

Another related study, conducted by Michael Lash and Kang Zhao at the University of Iowa, created a decision support system for movie profitability [2]. This study uses data jointly from IMDb and BoxOfficeMojo, focusing on “who” is in the cast, “when” the movie will be released, and “what” a movie is about [2]. The study found that their recommender system vastly outperformed existing methods of predicting profitability.

Current Studies

Some of the most recent research regarding movie success focuses on social media and user personality, using similar techniques to those currently employed in personalized advertising. A study produced by the GroupLens research group, titled: Exploring the Value of Personality in Predicting Rating Behaviors: A Study of Category Preferences on MovieLens, found that incorporating personality types into user-specific recommendations created slightly better recommendations [4]. Other related research has incorporated social media chatter and social media personality types to determine additional methods to customize recommendations for the users.

Conclusions and Findings of Relevant Work

Very similar studies of user-behavior based recommender systems have outputted similar accuracies to our model. As mentioned above, some studies compared models like Random Forest and Support Vector Machines, hoping to find the optimal model. Described below, our conclusions detail a very successful and accurate model in predicting whether or not a user will watch a movie. Most models developed in a similar manner have similarly high accuracies. As a general statement, it is fair to say that there are many methods available to develop a successful recommender system.

Results and Conclusions

With our predictive model, we were able to achieve an accuracy score of **0.9511** and a balanced accuracy score of **0.802**. These results were achieved using a Random Forest Classification model. Due to the dimensions of our dataset being so high, our group attempted to fit a number of decision tree classifiers on various sub-samples of the MovieLens film dataset and used averaging to improve the predictive accuracy and control over-fitting. Also, we employed Randomized Search Cross-Validation for the purpose of tuning our parameters. This model outperformed our attempts at both a Logistic Regression and Support Vector Classifier model which provided accuracy scores of **0.8797** and **0.792** respectively. Moreover, our Support Vector Classifier model achieved a balanced accuracy score of **0.500** which performed no better than a trivial predictor.

Similar to the read prediction task we saw in the previous assignment, our group hypothesized that popular movies (i.e. movies that are watched by a large number of people) were more likely to receive a favorable review as opposed to less popular movies. Movies that made up the top half of all movies in terms of viewership were labeled as being “Popular”. The remaining films were marked as being “Not Popular”.

Our group then made an attempt to implement a cosine similarity feature amongst the movies in the dataset, however the runtime necessary to compute this feature was far too long given the amount of movies contained in our dataset. As a result, the cell would routinely crash every time we attempted to construct this feature. Optimization would be needed to implement a cosine similarity. Instead, we turned to jaccard similarity in order to try and quantify similarity instead. To build this feature, we calculated the maximum, minimum, and average jaccard similarity scores between both users and movies for each (movie, user) pair.

Another feature that we initially included in our model was the budget for a given movie. Our thought was that movies with a higher budget would have higher production value and people would enjoy the movie more as a result. However, this did not prove to be true. Including the budget in our predictive model did little to nothing to improve the accuracy score which told us something about what’s important (or not important) to the audience. It is not the case that movies with high budgets will receive rave reviews from all types of users. Whether you’re a film savant or a common moviegoer you prefer to watch films that align with what you’re interested. This is why our similarity measures worked well to predict a user’s movie preference. In general, people enjoy watching movies that are similar to ones that they have watched and enjoyed in the past. The same goes for preference in the opposite direction. If a user didn't enjoy a movie about the Battle of Gettysburg (e.g. *Gettysburg* (1993)) it is more than likely that same user will give a low rating to another film depicting an event from the American Civil War.

Works Cited

1. Karl Persson, Joe Steinhauer, Alexander Karlsson (2015). PREDICTING MOVIE RATINGS: A comparative study on random forests and support vector machines
2. Michael T. Lash and Kang Zhao (2016). Early Predictions of Movie Success: the Who, What, and When of Profitability
3. Yogesh Patil (2019). MovieLens 100K Data Analysis
4. Raghav Pavan Karumur, Tien T. Nguyen, Joseph A. Konstan. (2016) Exploring the Value of Personality in Predicting Rating Behaviors: A Study of Category Preferences on MovieLens
5. Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.
6. F. Maxwell Harper and Joseph A. Konstan (2015). The MovieLens Datasets: History and Context
7. Chen, Chao, et al. “Using Random Forest to Learn Imbalanced Data.” Berkeley Statistics, https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf.