MSAN 621 - Group Assignment
IMDB 5000 Movie Data

Claire Broad, Andre Duarte, Brigit Lawrence-Gomez, Will Young
Corn on da CAWB

December 8, 2016

1 Introduction

1.1 Data

The data set used was the IMDB 5000 movie data available from Kaggle (https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset). This data set includes the top 5000 movies as ranked by production budget (source: http://www.the-numbers.com/movie/budgets/all). For each movie, a total of 28 features were available, which can be grouped into the following categories:

- Director / actor / movie Facebook likes
- Number of IMDB reviews
- Content Rating (PG-13, R, etc.)
- Production Budget
- Duration (run time of the movie)
- Release year
- Genre

For the gross revenue prediction, we scraped IMDB and added the following information:

- Release date
- Plot description
- Writer
- Awards
- Metascore

We also calculated an IMDB score for each actor and director and added this score as a feature. This IMDB score was calculated by averaging the IMDB score for each actor across every movie in which they appeared. A complete list of available predictors can be found in the appendix (3).

1.2 Goal

The goal of the project was tri-fold. First, we wanted to see whether we were able to predict movie revenue based on a combination of the features above. Second, we wanted to assess if we were capable of correctly labeling a movie's genre based on other descriptive features. Finally, we wanted to see if movies would form clusters based on their features alone.

2 Predict genre

2.1 Goal

The goal of the portion of the project was to use plot keywords to predict the movie's genre. We used two approaches: first, based on the plot keywords field in the data, and second, based on the plot description

provided. The theory was that certain keywords would be highly indicative of certain genres–e.g. 'spy' would be indicative of an action or thriller movie.

2.2 Method

The first task was bucketing the genres. The raw data listed up to five genres per movie, so we needed to pick the best descriptor to use as the primary genre. This process was subjective, and thus adjustments could be made to the bucketing process that might improve overall accuracy of the final model. The genre buckets that we chose were Drama, Romance, Comedy, Action, Thriller, Horror, Fantasy, and True (the latter encompassing documentaries, biopics, reality TV, etc.). We used a series of conditional statements to assign each movie to the best bucket for its specific set of genre tags. For example, a movie with genre listing mystery, fantasy, action wold be bucket as a thriller.

For the first pass at this problem, we worked with the keywords field from the raw data. This field gives a list of five keywords for each movie. We used a counter object to determine how many times each keyword shows up in the data. For this model, we used a 90/10 training/test split; therefore, our feature set was the list of all keywords that were present at least 10 times. We then used one-hot encoding ot create a vector for each movie indicating whether each of the top keywords was present in its keyword list. The algorithms we tested were K Nearest Neighbors, Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Decision Tree Classification, and Random Forest Classification. Overall, Logistic Regression had the highest performance of the models.

The problem with using the keyword field is that it is user sourced – the five keywords in the data dump are the top five in terms of upvotes on IMDB. Therefore, they are not always the most illustrative terms. For example, the keywords listed for "Monster's University," a children's movie, are "cheating|fraternity|monster|singing in a car|university." Because the reliability of this field seemed dubious, we thought we might do better by looking at the plot description field.

We tokenized the plot descriptions and removed stop-words. Then we used a counter object to determine the most common words, as before, but this time we tagged the word with the genre bucket of the movie – e.g. 'brother - fantasy' would get counted for Hercules, whereas 'brother - comedy' would get counted for Stepbrothers. We then pulled the top 1000 tagged terms from this counter as potential keywords. If a word showed up in the top 100 more than 4 times (that is, if it were one of the most common words for 5 or more genres), we excluded it from the feature set in order to cut down on potential noise. Finally, we stripped the genre tags and isolated the unique words, for a total of 388 keywords. The ensuing analysis was analogous to the previous case, with the one-hot encoding based on whether a given word was present in the movie's plot description. In this case, Linear Discriminant Analysis was the most accurate classifier.

2.3 Results

The keyword analysis consistently gave the correct genre bucket tag for just under 40% of the test set. In addition, we calculated an 'obscured prediction rate' by comparing the prediction to the original list of genres for the movie before bucketing. In the keyword analysis, this showed an additional $\sim$20% accuracy, for a total of about 55% of the test set being tagged with an accurate genre. Note the naive model that guesses the same genre each time would have an accuracy score of around 13% since there are eight genre buckets.

The plot description analysis had about the same main accuracy rate of 40%, with a slightly lower obscured prediction rate of 17%. In addition, we also looked at the individual accuracy rates within the genre buckets, but the rates varied greatly with each trial due to the relatively low n for each bucket in the test set; this would be a good place to implement more stringent cross-validation methods. If a given genre consistently has a very low prediction rate, that might indicate that the inclusion flow for that bucket needs adjusting.

Results are summarized in Table 1.

Table 1: Results of genre classification on test data

Method	Model Type	Prediction accuracy	Obscured accuracy	Overall accuracy
Plot keywords	Logistic Regression	40%	20%	60%
Plot description	LDA	40%	17%	57%

3 Predict revenue

3.1 Goal

The goal of this portion of the project was to predict gross revenue (USD) using some or all of the given predictors. We explored two questions:

- Can we accurately predict gross revenue using all of these factors?
- Can we accurately predict gross revenue using only the factors that would be known before the movie is released?

The first question will give us a baseline against which we can compare subsequent models. Since this questions allows the use of all available predictors, we expect this prediction to be the most accurate. We should also gain insight into which factors are strongly related to gross revenue.

For the second question, we limit ourselves to a business context. If we were planning production of a new movie what is our expected revenue given a certain set of features, and which of those features can we manipulate in order increase revenue? Framing the question in this context limits the predictors that we can use. We would not know movie Facebook likes, movie IMDB score, number of users used for review, number of user who voted for the movie, or IMDB metascore prior to its release.

3.2 Method

In addition to the data provided by IMDB, we also used genre (as defined in section 2: predict genre), release date, and actor and director IMDB score. For example, if Tommy Lee Jones was in three movies on the list with IMDB scores 6, 7, and 8, then Tommy has an actor IMDB score of 7. The same technique was used for the directors.

To clean the data for these experiments, we removed all movies with no or zero value for gross revenue. There were 926 movies where gross revenue was omitted. The movies produced in South Korea had gross revenue reported in Wons. Rather than attempting to adjust for exchange rate for the release date of each of these movies, we simply excluded them. This data set also included television series which we excluded. Some duplicate data appeared as well which was removed. In total we trimmed the data set from 5354 observations to 3431. Numeric predictor values were imputed with the mean value for that predictor. Null string values were imputed with “missing”.

We also explored subsetting the data. Since Facebook likes were often a strong predictor, we considered the relationship between when the movie was released and the number of Facebook likes. Indeed, movies released after Facebook grew in popularity have more Facebook likes than older movies. Note, it is possible to give a Facebook like an older movie, but this does not happen as often. To account for this, we trained one model that only included movies released after 2010.

Ultimately, we trained three different models:

- Model 1: Includes all available features to predict gross revenue (all years)
- Model 2: Limited Predictors - excludes those not available prior to movie release (all years)
- Model 3: Limited Predictors - excludes those not available prior to movie release, Post-2010 movies only.

To train the models, we randomly split the data into test and training sets with 70% of the data used in the training set and 30% of the data reserved for a test set. Some initial experimentation with a variety of models revealed a random forest would be the best model to use to to reduce MSE on the test set and return a predictor importance value. We tuned the depth of the Random Forest independently for each model using the OOB score.

3.3 Results

The results for all models are summarized in Table 2. The fourth column is the standard deviation of the test data. As expected, the first model with all available predictors gives the best results. Though all of the models explain a large portion of the overall variance, the overall variance is quite large. As a result the $RMSE$ values for the models range from 40 million to 60 million dollars. These predictions are far from accurate enough to use in planning. In light of this large variance in Gross Revenue, modeling was tested on log-transformed data. However, large variance still exists even after the log-transformation and none of our models developed using the log-transformed data outperformed the non-transformed data.

The more interesting information comes from the predictor scores. The top predictors for the first model were:

- number of voted users
- budget
- number of users for review
- rated R (indicator)
- actor IMDB score
- cast total facebook likes

The number of voted users carried half of the weight. The top predictors for the second and third models were:

- budget
- director IMDB score
- actor IMDB score
- title year (only in second model)
- release month
- rated R (indicator)
- Facebook likes (cast and director prior to movie)

Budget carried nearly half of the weight.

We are tempted to conclude budget is important in all models simply because the gross revenue values were not inflation adjusted, and budget is simply adjusting for inflation. However, the third model includes only years after 2011 where inflation has been relatively mild. From this we can conclude note there does appear to be a positive return on investment in most cases. This does not imply we can spend arbitrarily increase the budget to increase the gross revenue. But in general, investments in the movie appear to be rewarded.

Corresponding to that result, the inclusion of actors or use of directors previously involved in good movies (high IMDB scores) tends to increase revenue. This also aligns with the usefulness of including actor and director Facebook likes in the model. More popular actors and directors draw more people to the movies.

The title year carries weight for three reasons:

- inflation
- increased population
- increased popularity in movies

Year is not really something we can control, but it helps to make these adjustments in the model without calculating them explicitly.

The release month is an interesting variable. In fact, we see the highest grossing months are May, June, July, November, and December - the summer and winter holiday seasons. May and June top out at around 90 million dollars in average revenue and July, November and December all average between 60 and 80 million dollars. The next highest month is March at 50 million dollars and the other months are in the 20-30 million dollar range. The budgets also tend to be higher in these months as well though the differences are not quite as pronounced.

The content rating also carries a significant weight. Of the popular content ratings (G, PG, PG-13, and R), rated R movies are the lowest grossing. Rated R movies on average gross 35 million dollars while PG and PG-13 movies gross 80 and 70 million dollars on average respectively and G rated movies gross on average over 90 million dollars.

Table 2: Results of Random Forest Fits on Test Data

Model	R^2 Test	$RMSE$ Test	sd Test Data
1	0.7	4e7	7e7
2	0.5	5e7	7e7
3	0.5	6e7	8e7

4 Feature Visualization

4.1 Goal

The goal of this portion of the project was to explore how 16 of the 28 available features simplified to fewer features, and to visualize those movies which were similar to across those features. In other words, is it possible to say that some movies are close to each other based solely on the information included in the data set? As an example, we could hope to reveal how movies are similar to each other along non-obvious dimensions beyond genre.

4.2 Method

To accomplish this goal, we performed Principal Component Analysis (PCA). With this technique, we are able to calculate the first n principal components, i.e., the orthogonal linear combinations of our features that explain the most variability in our data set. In this way, we project our observations into this optimal reduced space. Movies that are located together are similar in terms of these new axes, which may or may not be interpretable. In particular, this is a useful step to visually assess if further clustering techniques should be applied.

Here, only numeric features were kept, as PCA does not accept categorical inputs. Missing values for numeric data were imputed using the mean value for that feature across movies. After this step, 16 features remained. The objective was to reduce the feature space to the first 3 principal components and assess whether we had clusters and/or could identify the meaning of each axis.

One final step before PCA was to scale the data, as the features were not comparable – movie budget and number of Facebook likes cannot be compared one-to-one for example. A min-max scaler was used to get all the features scaled to a [0-1] range. Thus, the transformed features had equal initial weight.

4.3 Results

By plotting the ratio of explained variance by principal component, as seen in Figure 1, we can see that the first three principal components explained 34.2%, 18.7%, and 11.8% of the total variance in the data

respectively. By using only these three components, as shown in Figure 2, we are effectively taking the axes that explain 64.7% of the total variability in the data set. Although this number is not exceptionally high (we should aim for around 80% of total variance explained by the principal components), we decided to use only the first three in order to make it easier to visualize the data.

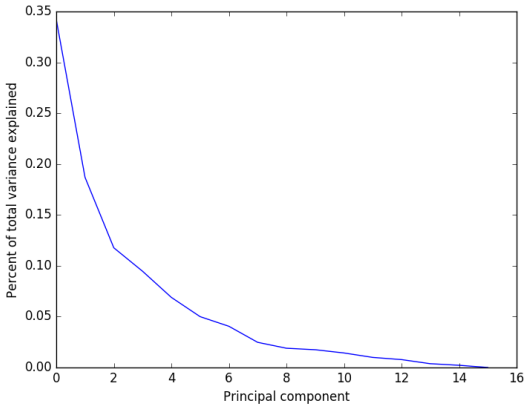

Figure 1: Scree plot from PCA clustering

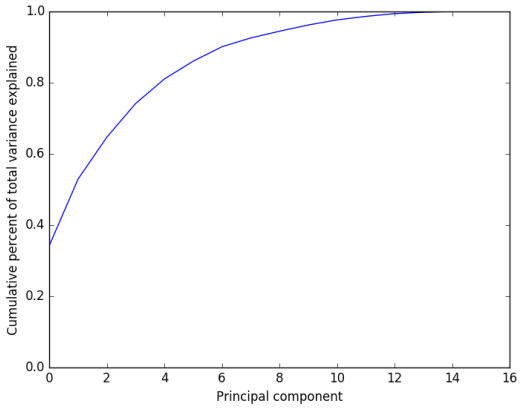

Figure 2: Cumulative scree plot from PCA clustering

The principal components are linear combinations of the 16 numeric features in the data set. Therefore, they may be able to separate and gather the data in ways which were not possible or accessible using the raw values. In order to try to infer the meaning of each principal component, we plotted the normalized data in this principal component space, as seen in Figures 3, 4, 5, and 6. While we could not at first identify what information each component was explaining, and the reason behind some of the clusters observed, we discovered that each principal component was highly correlated with one of our original features. Indeed, by coloring each point in the PCA plot by these original features, this correlation becomes visible:

- The first principal component roughly corresponded to the number of critics for review, as seen in Figure 3.
- The second principal component roughly corresponded to the movie's year of release, as seen in Figure 4.
- The third principal component roughly corresponded to the number of Facebook likes for the director, as seen in Figure 5.

Figure 6 shows a 3D scatter plot of movies using the color scheme for the year of release.

This result makes sense intuitively: IMDB is a website where people from around the world provide information about movies. It is inherently tied to the fact that the internet plays an ever-increasing role in our lives and many of our activities are social media-driven. Movies released many years ago will have had less "social buzz", and therefore their actors and directors will likely have less Facebook likes. Even well-known classics such as *Gone with the Wind* and *Modern Times* do not have comparable numbers of Facebook likes to modern blockbusters like *The Dark Knight Rises* or *Skyfall*. In addition, a contemporary director like Christopher Nolan has many Facebook likes, and his movies (the *Batman* trilogy, *Inception*, *Interstellar*) are watched by millions of people around the world who "like" it and comment and write reviews, hence the cluster that they form are centered around these features.

There are also other factors that could come into play, but were not present for our analysis. A rapidly growing world population that watches movies and increasing movie theater prices (the data is not corrected for inflation) are two such examples. Based on these results – and lack of clusters – we concluded that it was not worth it to further try to cluster the data set. Additionally, it is possible that other imputation techniques and data clean-up methods would produce slightly different clusters.

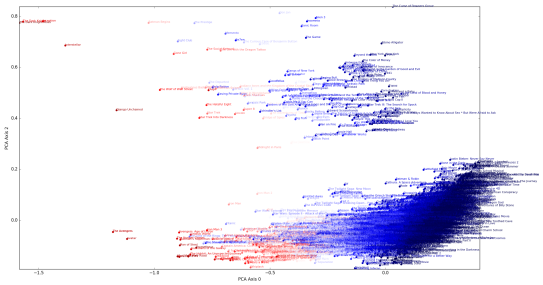

Figure 3: PCA scatter
Color: Num critics for review

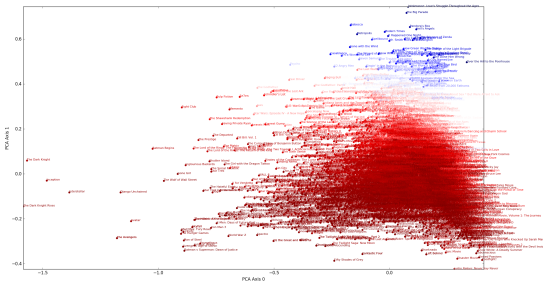

Figure 4: PCA scatter
Color: Year of title release

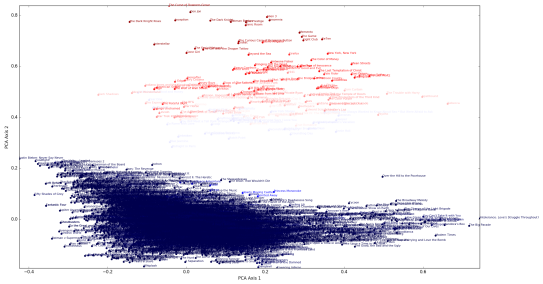

Figure 5: PCA scatter
Color: Director Facebook likes

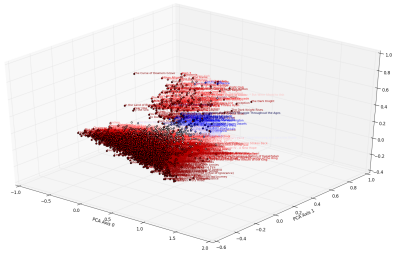

Figure 6: PCA 3D scatter
Color: Year of title release

5 Conclusion

With this assignment, we set out to explore and use several machine learning algorithms with different goals and assess their efficacy in predicting valuable information. We defined a three-pronged approach to perform regression, classification, and unsupervised clustering. Using these methods, we wished to assess whether we were able to predict movie gross and movie genre based on a set of features readily available from IMDB. In addition, we also wanted to see if movies would cluster by "closeness" and if we would be able to determine what causes two movies to perform similarly.

For prediction of movie genre, we used two tactics – by looking at the plot keywords and the plot description – and compared several classification algorithms. A Random Forest classification produced the best results, and we achieved 47% accuracy in prediction (with an additional 16% obscured prediction). This is significantly better than chance, which would be 12.5% since we used 8 genre buckets.

The revenue prediction experiments did not produce a model that is accurate enough to set budgets or present to investors. However, the results of the models did give us insight into which factors tend to increase or decrease the revenue of a movie. Movies tend to pay back investments in better actors and directors. The content rating as well as the month of release are also important and controllable factors. Our experiments indicate a G-rated kids movie released in May, directed by Christopher Nolan, and featuring Johnny Depp, Jennifer Lawrence, and Robert Downey Jr may not be the worst idea. *Lego Batman*? In future studies, we also like to consider the marketing budget to consider the impact of advertising on revenue.

With regard to parameter reduction, we discovered that the first three principal components, which explained around two-thirds of the variability within the data set given numerical features, were essentially three of the original features available in the data. We were not able to infer additional information based on this technique, and could not discern any visible clusters. However, it allowed us to realize that the IMDB movie data set is skewed toward more recent movies. This is largely due to the fact that internet adoption is quickly

growing, and social media has an increasing role in generating "buzz" for movies. It is worth noting the results of PCA analysis align with some to degree but not entirely with the results of the revenue predictions. The variables that explained most of the variance in the PCA were some of the more important variables in the Random Forest models as well.

During this project, we learned how to implement machine learning algorithms to real-world data. We had to deal with missing/wrong features, cleaning up the data, and engineering or fetching additional information to complete our analysis. The approach used to handle missing and wrong data can directly influence the results obtained by the analysis, and as such have to be meticulously determined. As is usual, some approaches produced better results than others, and we were able to generate insightful information from the original data.

6 Appendix

Table 3: IMDB 5000 Metedata

Variable	Description
Cast total Facebook likes	Total number of Facebook likes the movie cast has received over all time.
Title year	The year the movie was released.
Director Facebook Likes	The total number of Facebook likes the movie's director has received over all time.
Duration	The length of the movie in minutes.
Budget	The production budget of the movie.
Content rating	The content rating of the movie (G, PG, PG-13, R, etc.)
Genre	The single bucketed movie genre.
Release month	The month of the year the movie was released.
Actor IMDB score	Average IMDB score of all movies the actor has starred in.
Director IMDB score	Average IMDB score of all movies the director has directed.
Movie Facebook likes	The total number of Facebook likes the movie has received .
IMDB score	The IMBD score of the movie (user generated).
Meteascore	The critic score of the movie.
Number of users who reviewed movie	The total number of IMDB users who reviewed the movie.
Number of users who voted for movie	The total number of IMDB users who voted for the movie.