Citadel Datathon Summary Report

SUBMISSION BY: Team 22

Colin Flaherty, Jonathan Yuan, Zijian Li, Abhinav Singh

Executive Summary

Our high-level goal is to discover the primary drivers of customers' preferences for movie consumption, and then use these drivers to generate insights regarding what kinds of movies customers prefer. The end goal is to suggest a framework for designing an ideal movie driven by customer preferences. In order to accomplish this goal, we consider both customer-generated movie attributes ("tags") and standard attributes such as movie budget and genre. In general, we use mean customer-generated ratings as a proxy for customer preferences and complement that with an analysis of the spreads of these ratings.

Customer-generated "tags" provide a convenient way of categorizing movies. By examining what tags correlate strongly with customer preferences, we are able to better understand what kinds of movies are preferred by customers. In general, we found customer-generated tags to be highly correlated with customer preferences, which is expected. We then investigate further to discover what tags are correlative with positive ratings and what kinds of movies. We call those tags that are strongly correlated with customer preferences "ratings-driven tags".

Building on this theme, we explore what kinds of movies tend to be "controversial". Controversy within these ratings is an interesting abstraction because it alludes to the polarization in opinion for each movie and points at a different aspect of the distribution of consumer preference. We suspect it would be relevant to the outcome of each movie's average rating and general performance. To assist with this goal, we define metrics to quantify "controversy" and then ask questions about what kinds of movies tend to be controversial, and how controversial movies tend to fare in the box office and in the Oscar Awards.

We discovered that ratings and controversial scores are worthwhile predictors of success in terms of a movie's gross profit and potential to win awards and saw that they are mutually negatively correlated. Using these findings, we set out to both build accurate models for both of these metrics using relevancy tags and to statistically understand which tags are drivers behind them, and obtained a very large R^2 value of approximately 0.99. This allows us to not only build a comprehensive view into the exact keywords from a customer perspective that influence movie success, but also construct a framework to potentially design a movie in line with consumer preferences. We believe this work is highly relevant when understanding the drivers of consumer preferences at a granular level, and have suggested ways to improve and next steps for expanding upon our results.

Predicting Movie Ratings From Movie Attributes

To catch an overall understanding of the correlation between the relevance score of movies to user-generated tags, we started with a linear model to investigate if there is any linear relationship between them. Let x_i^j denote the relevance score between movie j and tag i, and let y^j denote the average rating of movie j.

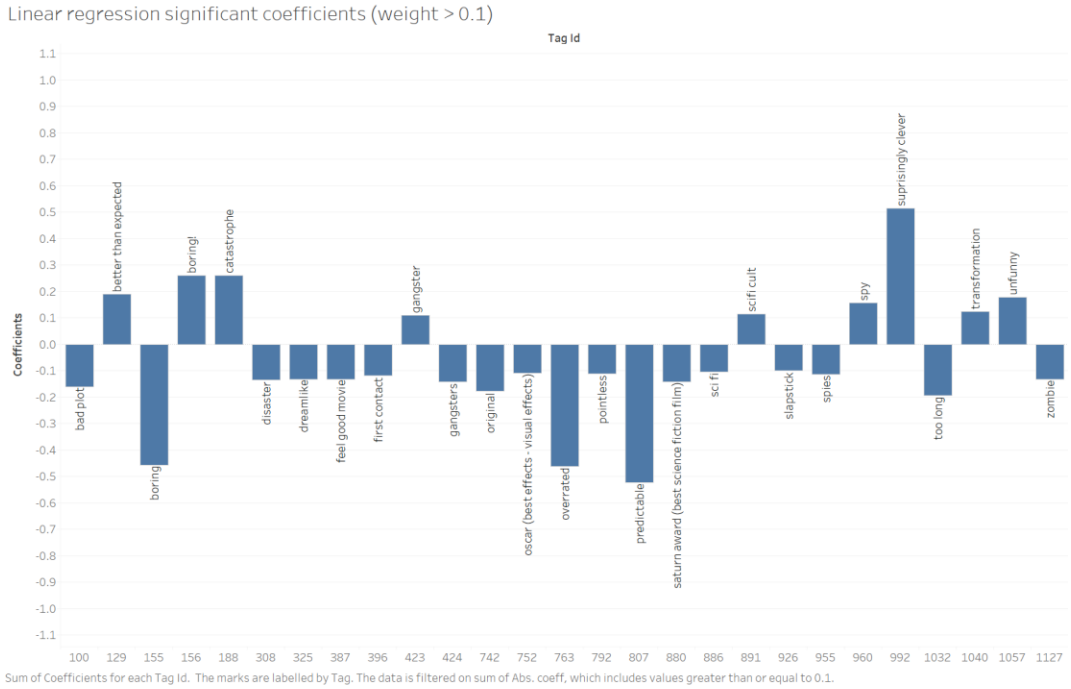
For$X^j = [1, x_1^j, x_2^j, \ldots, x_{1128}^j]^T$, we assume:

$y^j = \Theta \cdot X^j$ where $\Theta = [\theta_0, \theta_1, \ldots, \theta_{1128}]$ is chosen so that

$M.S.E. = \frac{1}{13176}\sum_{j=1}^{13176} \left(y^j - \Theta \cdot X^j\right)^2$ is minimized.

As our first model, we naively took all of the relevance scores between 1128 tags and 13176 movies as the input and the average rating of these movies as output. This yields us a linear model with coefficient of determination (R^2) = 0.985. This means that a good portion of variance in the average rating of movies can be explained by these relevance scores. The picture below shows 27 tags with the most weighted

coefficients. Nevertheless, without testing any assumptions a linear model presumes, we are not ready to make conclusions about these coefficients.



Observe that some prominent tags have similar meanings but lead to opposite effects. For example, both tag <boring> and <boring!> are very prominent but have opposite effects. This suggests our model suffers from collinearity among the features suggesting that the terms have interactions among them. In an ideal linear regression scenario, each of the independent variables should have a direct impact on the dependent variable Y and not on any of the other X variables but the above coefficients suggest that if the value of the tag <boring> is changed, we'll have an impact on the <boring!> tag as well, which further effects the prediction of Y. This makes the interpretability of the model difficult and we need to handle such interaction terms in our linear regression model.

To better evaluate this model, we randomly split our whole dataset into training, cross validation, and testing subsets, with size ratio 6:2:2. Fitting the same model only on the training set yields us a mean squared error (MSE) of 0.00309 on the training set and 0.00427 on the test set. This suggests our model is potentially a good predictor of the average rating based on the relevance scores, but it currently suffers from overfitting the data.

Now we split our efforts working on 2 different directions:

1. Improve the accuracy of this predictor.
2. Derive more meaningful statistical inferences

Unfortunately, these two goals could not be achieved simultaneously. More details will come later.

Accuracy Improvement

The MSE of our predictor on the test set is around 38% higher than that on the training set. This suggests our model is overfitting. To get more accuracy prediction, we tried the following two approaches to neutralize overfitting:

1. Drop tags that are not prominent.
2. Introduce regularization.

Non-Prominent Tags

As we noticed that some tags are linguistically similar to one another, such as <boring> and <boring!>, <gangster> and <gangsters>, these tags may have strong correlations in terms of their relevance to movies. These strongly correlated tags are the ones that we will remove first.

For each pair of tags, we computed the correlation between, where correlation between tag i and tag k is defined as

$$Corr(i,k) = \frac{\sum_{j=1}^{13176} \left(x_i^j - \underline{x_i}\right)\left(x_k^j - \underline{x_k}\right)}{(13176-1)s_i s_k},$$

where s_i, s_k are the corrected sample standard deviations of x_i^j and x_k^j.

Strongly correlated tags (with correlation > 0.95) are the following:

```
>>>
tag #387 <feel good movie> and tag #388 <feel-good> has correlation 0.9528
tag #696 <nazi> and tag #697 <nazis> has correlation 0.9694
tag #886 <sci fi> and tag #890 <scifi> has correlation 0.9570
tag #887 <sci-fi> and tag #890 <scifi> has correlation 0.9563
tag #955 <spies> and tag #960 <spy> has correlation 0.9536
tag #987 <super hero> and tag #989 <superhero> has correlation 0.9644
tag #1067 <vampire> and tag #1069 <vampires> has correlation 0.9856
tag #1121 <world war ii> and tag #1126 <wwii> has correlation 0.9504
```

More interestingly, some tags are highly correlated for non-linguistic reasons. For example:

```
>>>
tag #152 <book> and tag #153 <book was better> has correlation 0.9013
tag #232 <comic book> and tag #987 <super hero> has correlation 0.8629
tag #697 <nazis> and tag #1121 <world war ii> has correlation 0.8528
tag #1079 <vietnam> and tag #1080 <vietnam war> has correlation 0.9367
tag #747 <oscar (best actress)> and tag #750 <oscar (best directing)> has correlation
0.8062
tag #747 <oscar (best actress)> and tag #759 <oscar (best supporting actress)> has
correlation 0.8580
tag #750 <oscar (best directing)> and tag #758 <oscar (best supporting actor)> has
correlation 0.8057
```

As we can see, tags with similar linguistic meaning tend to have high correlation. What to mention is that this correlation is computed based on statistics on user behaviors, during which only the id of a tag is used. The system did not try to interpret its meaning or analyse similarity in its character components. But our model is able to tell the similarity in linguistic meaning of the tags. We think our model will help identify words with similar meaning, even when they look very different or even in different languages, in natural language processing. Furthermore, if we are able to generate tag data about things beyond just movies, for example by implementing a tag system on Amazon.com on general commodities, we should be able to discover much more associations between different words.

Moreover, some tags are not correlated for linguistic reasons. For example, tag <comic book> and <superhero> have strong correlation, while “comic book” and “superhero” describe very different objects in English. This correlation is likely to be associated with the popularity of superhero comic books. It is of independent interest what we can infer from these correlations. With some cutoff correlations, we are to remove tags that have higher correlation to existing tags.

Regularization

Another attempt is made by introducing regularization terms. Now to get the coefficients, we try to minimize the regularized MSE:

$$regularized\ MSE = MSE + \alpha \sum_{i=1}^{1128} \theta_i^2$$

The regularization term penalizes coefficients not close to 0, which in terms reduce model complexity and overfitting. With some repeated model fitting with different combinations of cutoff-correlation and alpha, we found that when cutoff-correlation = 1 (not removing any tags) and alpha =0.0003 we get the best MSE on the cross validation set. Choosing such hyperparameters, we yield a model scoring a MSE of 0.00422

on the test set. As collinearity does not affect prediction accuracy, we did not try to limit the correlation of features at this point. This issue will be addressed in the next part.

Some other attempts are made, such as introducing genres, film year and budget as regression features, removing tags based on cross-validation MSE. None of these attempts made noticeable improvements to prediction accuracy. To further improve the accuracy of our predictor, obtaining more training data will be likely to help, but this is beyond the scope of this project.

Statistical Inference

An accurate predictor does not mean a good model. To evaluate the model and make sound inferences, assumptions for linear regression models must be tested. Regularization is not applied for this part.

Multicollinearity

As some features in this model are highly correlated, it is clear that the multicollinearity assumption is violated. As a result the value and significance level of each individual coefficient in our model is not reliable. To reduce multicollinearity within the features, we would like to iteratively remove features based on its variance inflation factor(VIF). However, with 1128 features and 13,176 data points included, computing the VIF is expensive as each VIF requires performing a linear regression. But notice that most of our linear collinearity comes from linguistic similarity between the two tags, thus bilinearity contributes a big part in our collinearity problem. Thus we could optimize this process by first removing features that are linear correlated. With a correlation-cutoff of 0.5, we have filtered 764 tags that are correlated with some remaining tag. Then we filter the remaining tags if it has a VIF greater than 5. This way we have significantly reduced the multicollinearity in our features. Now we have only 318 tags remaining as features.

```
>>>
Tag #2 <007 (series)> is removed due to correlation(0.7502) with tag #1 <007>.
Tag #151 <bond> is removed due to correlation(0.8744) with tag #1 <007>.
Tag #361 <espionage> is removed due to correlation(0.5777) with tag #1 <007>.
Tag #573 <james bond> is removed due to correlation(0.783) with tag #1 <007>.
...
In total 764 tags are removed due to correlation.
Column  relevance to tag 8  is dropped due to VIF  6.0395317372551505
Column  relevance to tag 18  is dropped due to VIF  9.410869484658841
Column  relevance to tag 19  is dropped due to VIF  10.527630411785598
Column  relevance to tag 21  is dropped due to VIF  21.14063516273654
```

But at the same time, the accuracy of the model dropped as features were removed. Now our model makes a MSE of 0.041 on the training set, with R^2 value dropped to 0.827.

Heteroscedasticity

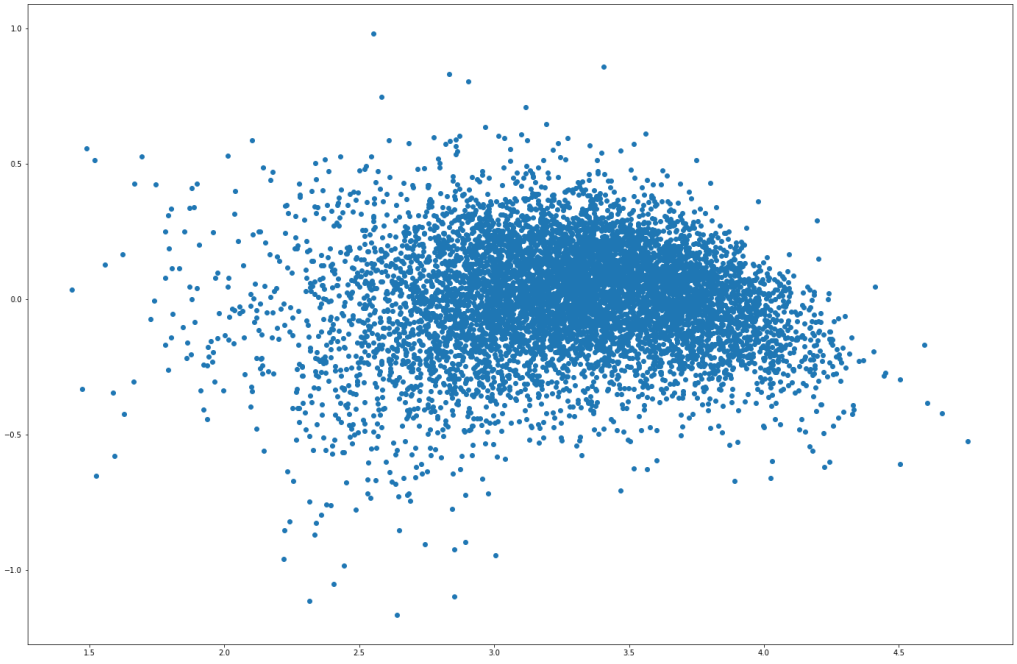
We perform a Breusch-Pagan test to determine if heteroscedasticity is present.

H0: Homoscedasticity is present.

Ha: Homoscedasticity is not present.

```
>>>
[('Lagrange multiplier statistic', 1147.1761516429183),
 ('p-value', 4.210030478776378e-94),
 ('f-value', 4.04957047319814),
 ('f p-value', 1.048535613566696e-104)]
```

Since p-value < 0.05, we reject the null hypothesis. We have sufficient evidence to say that heteroscedasticity is present in the regression model. This will affect our estimations of the significance of terms in our model. A fitted value vs. residual plot is provided below. According to the plot, it is unlikely we can fix heteroscedasticity with a simple transform of ratings.



Autocorrelation

To detect autocorrelation, we preform the Durbin Watson Test:

H0: There is no first order autocorrelation.

Ha: There exists first order autocorrelation.

```
>>>
Durbin-Watson:       1.978
```

Since Durbin-Watson score is between 1.5 and 2.5, we consider the residuals not autocorrelated.

Normality of residuals

We perform the Jarque Bera test and the Omnibus test for normality.

H0: Residuals are normally distributed.

Ha: Residuals are not normally distributed.

```
>>>
Omnibus:      438.122       Durbin-Watson:       1.978
Prob(Omnibus):      0.000  Jarque-Bera (JB):    834.518
Skew:  -0.409                Prob(JB):          6.12e-182
Kurtosis:     4.365          Cond. No.     113.
```

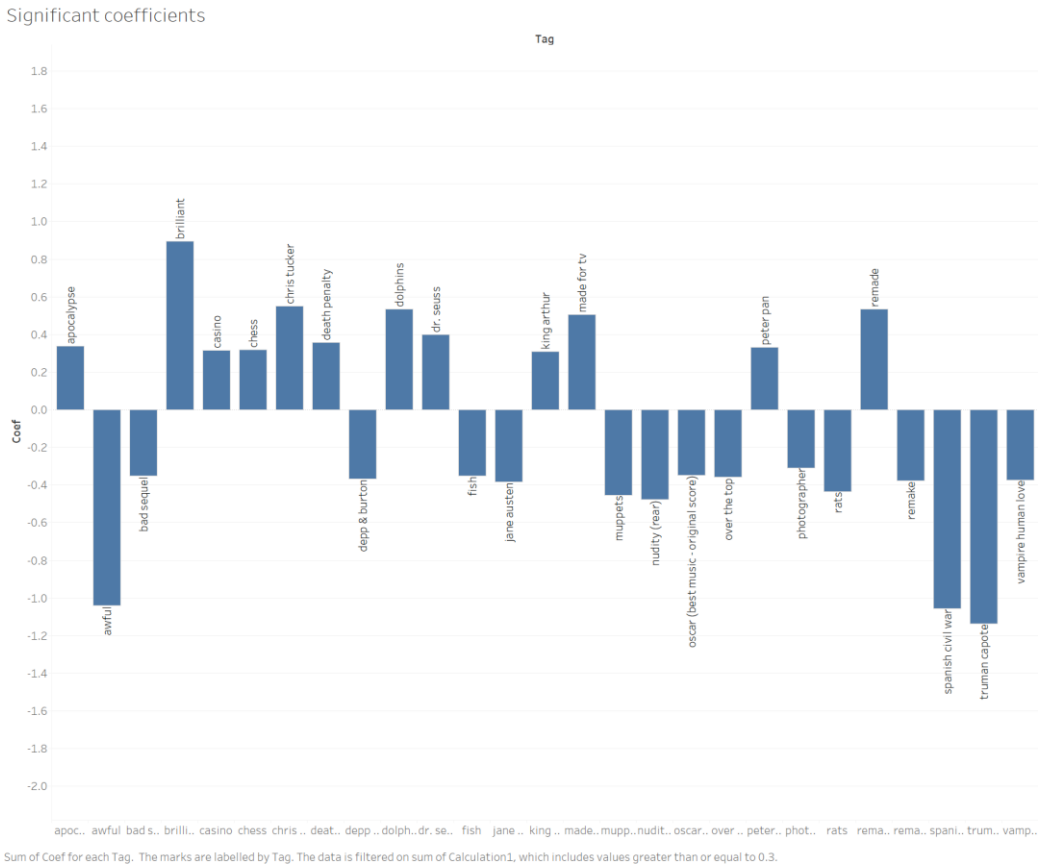
With p-value < 0.05, we reject the null hypothesis. There is evidence that our residuals are not normally distributed.

```
>>>
residual mean =  -1.0381743957026131e-16
```

But we believe that the residuals have mean 0.

Coefficients

The following is a plot of the most prominent tags in our linear model. Now that we have a model with low multicollinearity, these coefficients provide us a better understanding that tags tend to have more prominent effect on the rating of a movie. Meanwhile, we should keep in mind that the significance of these coefficients may not be accurate due to violations of some of the assumptions of linear regression models.



Controversy Scores

Whereas the above analysis focuses on a deep dive into the determinants of average ratings, to further explore the distribution of ratings amongst movies and audiences, we focused on a notion of "controversy" within the movie ratings that may allude to polarization in opinions of the reviewers. When analyzing controversy, we hoped to capture these differences in opinion amongst viewers captured by differences in the spread of the ratings. We can define proxy metrics by targeting the variance of viewer ratings in two ways.

We defined hard controversy as the weighted standard deviation of ratings:

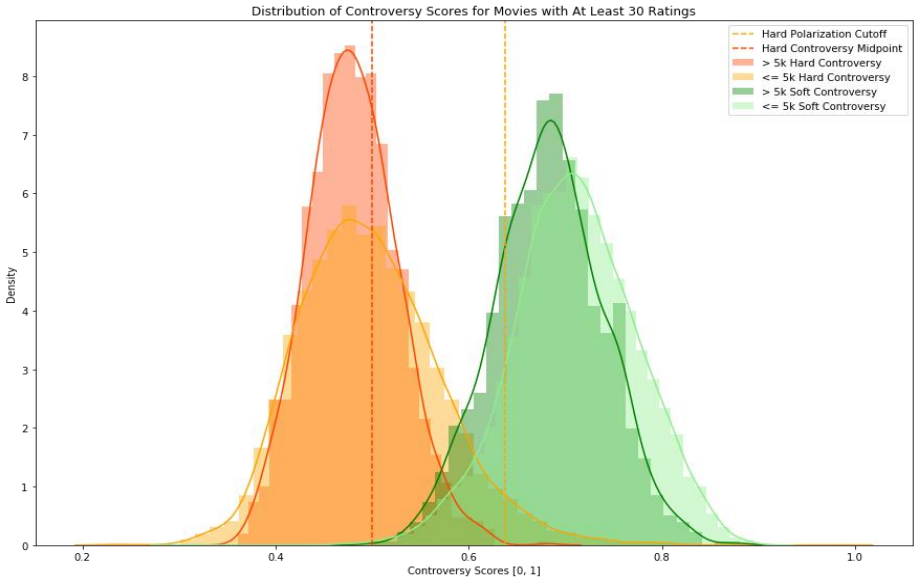
$$H = \frac{1}{c_H}\left[\sum_{i=0.5}^{5} p_i(r_i - \underline{r})^2\right]^{\frac{1}{2}}$$

and soft controversy as follows[1]:

$$S = 1 - \frac{1}{c_S}\left[\sum_{i=0.5}^{5} (p_i - 0.1)^2\right]^{\frac{1}{2}}$$

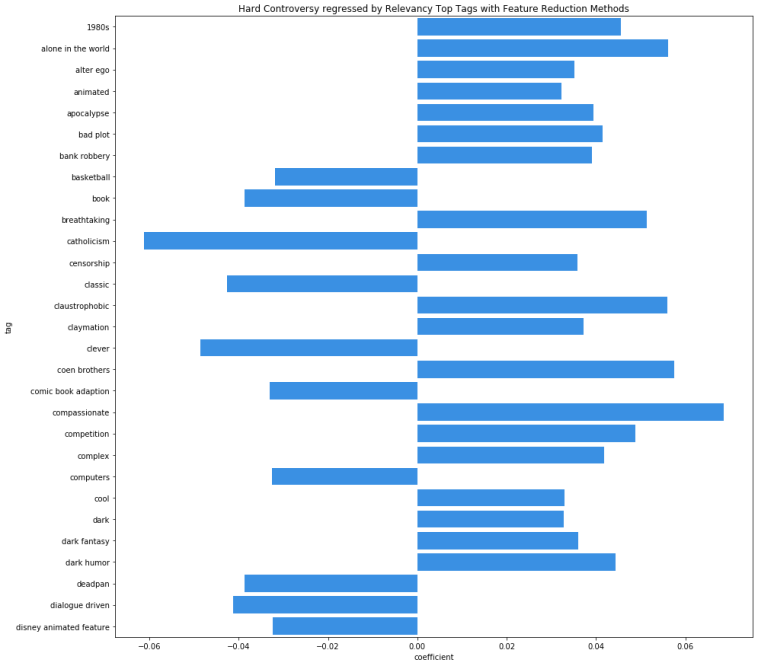
These are the two controversy scores we analyzed with our dataset, each serving as normalized scores from [0, 1] where a higher score implies a higher degree of controversy. In the case of hard controversy, we define it in terms of a weighted, normalized standard deviation, whereas in the case of soft controversy, we define it in terms of a measure of peakedness of the distribution, maximized where each rating occurs with probability 0.1.

[1] Inspiration for these "controversy" scores was derived from those utilized in (*Amendola, 3)*

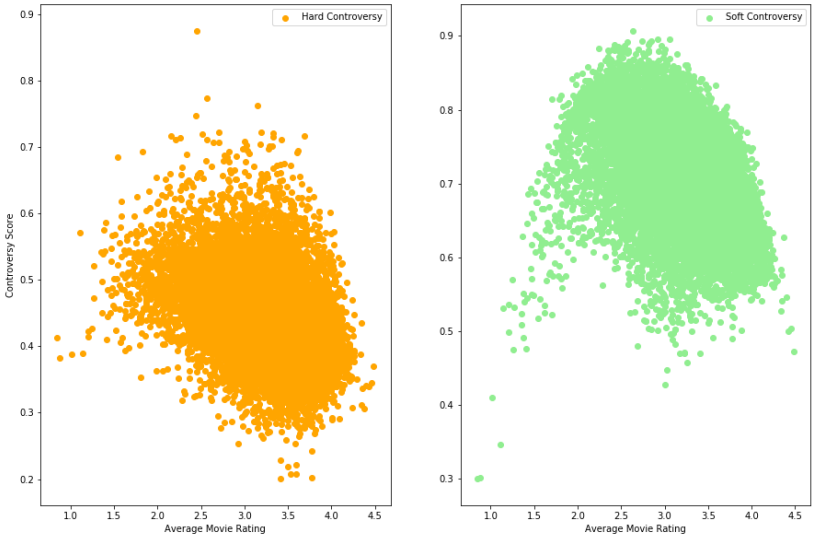


With these controversy scores, we extended the previous analysis on average viewer ratings to see if we could derive meaningful significance by regressing on tags and other movie attributes, and to build a strong model for prediction. We bucketed the scores based on the hard controversy thresholds of 0.5 (the midpoint) and 0.638 (the normalized cutoff threshold of a completely flat distribution). The 0.638 threshold denotes the controversy score where the distribution of ratings is completely flat (no peak), and anything beyond that would be considered bimodal, or polarized. We noticed, however, that there were less than 200 movies that fell into this category, highlighting the finding that extreme polarizations of rating scores were very rare. Another key insight from these controversy scores is that controversy tends to decrease as the number of reviews increases. The >5k and <=5k labels in the above graph describe the distribution of controversy for movies with greater or less than/equal to 5k reviews respectively. We can intuitively think of this as when a movie gets more views, it becomes less controversial, or as the fact that more controversial movies have fewer interested audiences. We will explore this more in the next section.

Following the method outlined in the previous section with average ratings, we built similar models with relevancy tags to predict these two controversy scores. We found that by reducing the feature space by iteratively removing features based on high correlations, their VIF scores, and using a regularization term, hard controversy yielded a much smaller MSE of 0.00175, with minimal overfitting - only 4.8% larger than the training dataset. A sample of large magnitude coefficients on these tags that survived the feature selection process is shown below, with quite a few being very intuitive. “Dark humor” and “bad plot”, for example, has a positive coefficient on increasing controversy, while “classic” and “clever” tend to reduce the controversy scores.



In terms of inferential results, we found a few interesting things. We suspected the average ratings and controversy scores are correlated. When running a regression, it seems that a film's hard and soft controversy are both significantly negatively correlated with ratings.



The two plots hint at the fact that increased movie ratings correspond to decreasing controversy levels, and the subtleties between the two controversy scores: hard controversy gives us more of an idea of polarized distributions, and those tend to decrease as ratings increase. As for soft controversy, we get a measure of how flat the distribution is, and those tend to be maximized at around the center of the distribution of average movie ratings. So we can understand the property that increasing a very low rating helps increase controversy scores, but there is a tradeoff between ratings and controversy scores when increasing scores from around 2.5 towards the maximum. This tradeoff between rating and controversy scores is an important point of consideration for movie makers, since a controversial movie might sacrifice some ratings for other benefits, like increased viewers, profit, and potential for winning an award. More discussion on these points can be found in the following section.

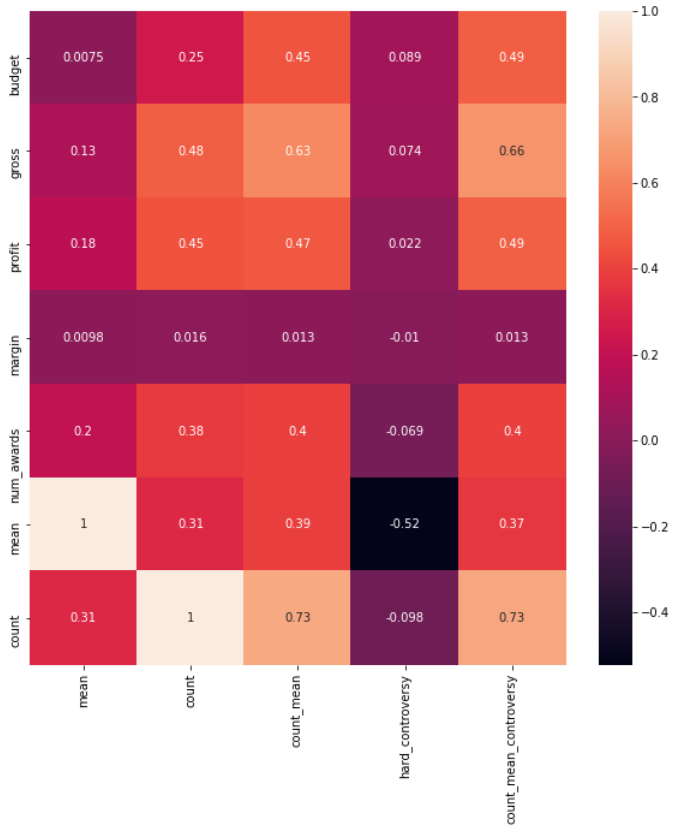
Why Try To Predict Ratings and Controversy?

At this point, we would like to summarize what we have discovered so far, and why these discoveries "matter". We began with a model trying to predict customers' sentiment for a movie based on what kind of movie it is. Since we neither can know the "true" rating of a movie nor fully encode all attributes of a movie, we use proxy statistics to estimate both the underlying customer sentiment distribution and the underlying movie distribution. To estimate the customer sentiment distribution, we use the mean rating of ratings provided by actual customers. Secondly, we approximate the underlying movie distribution by "encoding" a movie in terms of a genome of tags. This genome of tags was provided to us, and it yields machine learning-generated "relevancy" scores associating each movie with each tag in the genome. The vector of relevancy scores for a movie constitutes that movie's encoding, and is used to predict the movie's mean rating.

We took this approach because the ability to predict a movie's ratings based on its encoding is not only useful in itself but also provides a secondary benefit: It can be used to generate movie ideas that might be warmly received by customers. For example, if "Pixar" wanted to produce a movie that would be "based on a tv show" and set in the "18th century", then our model could be used to determine what kinds of ratings could be expected. ("Pixer", "based on a tv show", and "18th century" are all tags.)

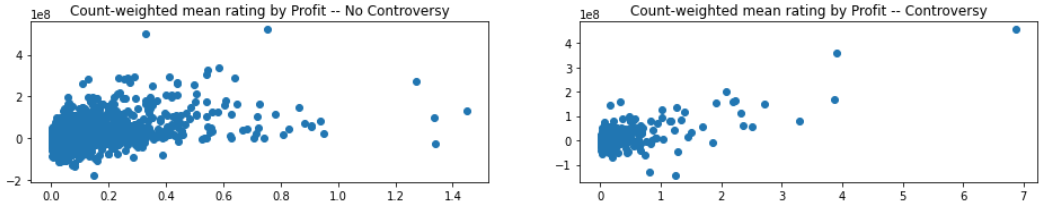
After creating this model, we then tried to predict what kinds of movies tend to be "controversial". To assist us in this endeavour, we defined "controversiality" in terms of standard deviation of ratings. (If a movie's ratings exhibit greater variance, then this means that ratings are clustered around low and high ratings.)

Now, we ask the question: Why try to predict ratings and controversy? For this, we turn to two other important metrics for movie products: profit and number of Oscar awards. If ratings and controversy are predictive of profit (or at least gross revenue) and number of Oscar awards, then creating movies that receive high ratings and are controversial / not controversial is clearly important from a business perspective.



This correlation map yields some interesting facts: Firstly, controversy is significantly inversely proportional with ratings (-0.52 correlation). This suggests that when movies are controversial (most ratings are 1 or 5 stars), most of the reviews will be 1 stars, not 5 stars. Secondly, the interaction term between mean ratings ("mean") and number of ratings ("count) yields a correlation of 0.63 with gross revenue and a correlation of 0.47. While there is little correlation between mean rating and profit margin, this may be due to production failures. Even if a move is very well-received, it might not have a high margin due to all of the costs of creating a movie.

While this analysis suggests that movie studios should avoid highly controversial movies due to their strong inverse correlation with ratings, this is in fact not necessarily the case.



An analysis of highly-controversial movies (standard deviation in the 90th percentile or higher) shows that ratings for these movies are a significant predictor of profit, with an R^2 of 0.74.

Other Models

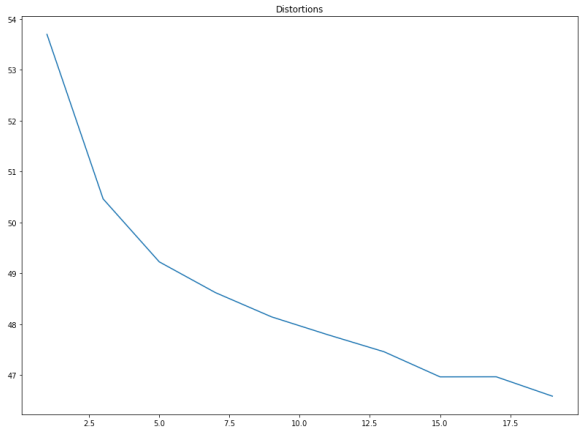
The work in this section is designed to augment our current models and provide alternative outlooks into the problem that serve as strong next steps. We can use some of this information as input to our current models, and will explain the modeling approach and insights obtained below.

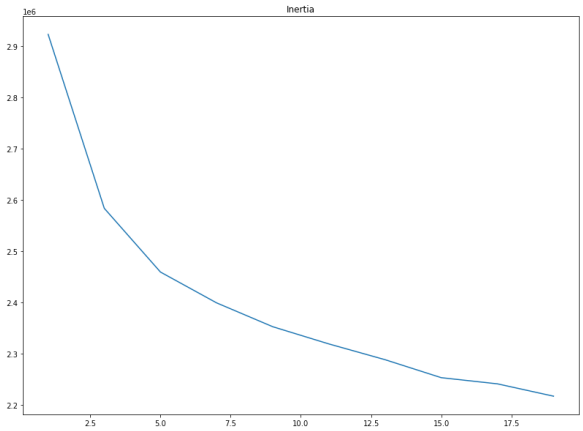
1. User rating prediction using k-means clustering.

We tried to come up with a predictive model using K-means clustering to predict the rating a user might give to a movie based on the information that how the user rated the most popular movies. Popularity is judged on the basis of the number of ratings that a movie receives i.e. more the number of ratings a movie has received, more popular the movie is. To train the model we selected top 1000 users based on the maximum number of ratings/reviews they provided and selected top 1000 movies which have the highest number of ratings. Using this, a k-means clustering model was trained which had the feature vector as the ratings that a particular user gave to the top 1000 movies and the model returned which cluster this user belonged to. Once we were able to identify in which cluster did the user belong to, we were able to predict the rating of a movie which the user has not yet rated. This can be done by taking the mean of all the ratings available for that particular movie in the predicted cluster.

During the analysis, we had to identify the number of clusters that the model should be trained on. We estimated the ideal k* value by using the elbow method. Elbow method is one of the most popular training algorithms to predict the best k value for the model. The elbow method runs k-means clustering on the dataset for a range of k values and then for each value of k computes a distortion score. When the distortion score is plotted against the k value, we were able to identify an elbow in the visual plots. An elbow is defined as an inflection point in the chart which basically indicates that the underlying model fits best at that particular point where k=k*.

Distortion is calculated as the average of the squared distances from the cluster centers of the respective clusters. We used the Euclidean distance metric to calculate this distance.





On observing the above plot for distortions against k values, we are able to find an inflection point at k = 10 which fits the model the best. Hence the number of clusters is equal to 10 for the final model. After getting a trained model with k = 10, if given the feature vector X where X represents the ratings given by a test user to the top 1000 movies, we can predict which cluster does the user belong to. Furthermore, after getting the cluster information, we can predict the rating that a user might give to a movie that he/she has not rated by taking the average rating of all the ratings of that particular movie in that cluster.

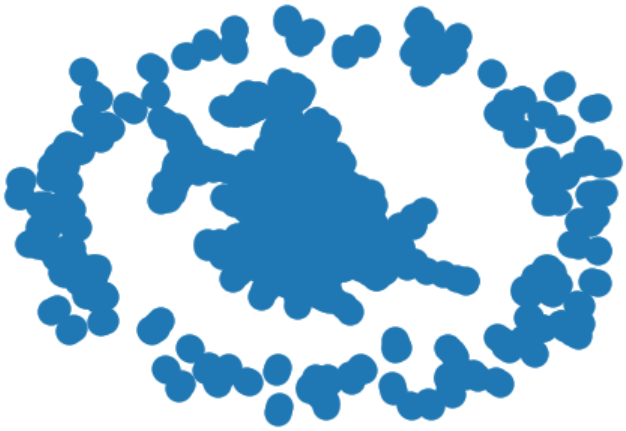
This analysis can easily be extended to the complete movie set rather than just the top 1000 movies, we selected just top 1000 movies in interest of time and computation complexity.

2. User community detection using Girvan Newman algorithm.

One of the most relevant features of graphs representing real systems is community structure, or clustering, i. e. the organization of vertices in clusters, with many edges joining vertices of the same cluster and comparatively few edges joining vertices of different clusters. Such clusters, or communities, can be considered as fairly independent compartments of a graph, playing a similar role like, e. g., the tissues or the organs in the human body. We tried to detect similar communities in user graphs using a very popular algorithm for community detection called Girvan-Newman algorithm. The Girvan–Newman algorithm detects communities by progressively removing edges from the original network. The connected components of the remaining network are the communities. Instead of trying to construct a measure that tells us which edges are the most central to communities, the Girvan–Newman algorithm focuses on edges that are most likely "between" communities.

For our dataset, we first generated a user graph, where each node in the network represented a user, and two users will have an edge among them if they have rated a similar number of movies based on some computation. For our analysis, we decided to go ahead and work on a graph where all users rated a movie only 5 stars, we also filtered out the rating data which was after January 1, 2015 for the genre '***Adventure***'. Two users had an edge in between them if they had rated the median number of the same movies. On constructing this user graph, we had 1887 user nodes which were connected using 111491 edges. As we can see this is a gigantic network and due to the computational bandwidth, we decided to move ahead with the above mentioned filters to perform our analysis.

Community graph



On completion of the analysis, from the above community graph we can infer that a huge amount of users end up rating 5 to similar types of adventure movies. It suggests that all users have similar taste and preference in adventure movies which allows them to form a large community. The smaller communities around the bigger chunk suggest that there exists some groups of users, which rated an adventure movie as 5, but which was not liked or appreciated by the bigger crowd. It can be interpreted as that the movies liked by these users were not part of the mainstream adventure movies hence creating an outlier bounding to the mainstream set.

As we can see, by just performing our analysis on such a small subset of data, we were able to identify patterns and infer about user preferences, we believe that if this study is extended on the entire dataset, we'll be able to get even more information on user preferences and what movies lie in the mainstream group. Using community information, we can perform sophisticated and dedicated linear regression analysis on these subset datasets to get further insights on the datasets and relationships between the above analysed report data.

Citations

1. F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. <https://doi.org/10.1145/2827872>
2. Amendola L, Marra V and Quartin M (2015) The evolving perception of controversial movies. Palgrave Communications. 1:15038 doi: 10.1057/ palcomms.2015.38.