Box Office Performance Prediction

Jia Rao(jr2254), Song Tang(st883), Wenchang Yang(wy286)

***Abstract*—The decision making process for a person who is planning to see a movie might sound like this: What are movies in theaters now? How do people rate them on Metacritic or Rotten Tomatoes websites? The person may search on Google for more information before purchasing the ticket. Moviegoers are now more actively exploring different movie options online. In this project, The objective is to use these information to predict box office performance of the opening weekend prior to its release date. In this way, movie marketers can adjust post-release marketing strategies based on accurate predictions: if the prediction of opening weekend performance is high, they may decide to release the movie in more theaters during the opening weekend in hopes of revenue growth.**

I. Exploratory Data Analysis

A. Data Collection

We used python to scrape movie data in 2016 from the following four websites or data sources: IMDB (imdb.com), Box Office Mojo (boxofficemojo.com), Metacritic (metacritic.com) and Google Trends (trends.google.com). The reason we only used 2016's data is that it's time-consuming to scrape, integrate and clean all the data. With 17 independent variables, 165 samples of movie should be able to support the prediction.

From IMDB, we collected the basic profile of movies (release date, run time, directors, actors, budget, MPAA rating, etc.). Here we only consider wide-released movies (defined by Box Office Mojo) which were released in 600 or more theaters, since other limited-released movies have too many missing values in the variables we acquired and the marketing strategies for them may be different. The total number of wide-released movies in 2016 is 165.

From Box Office Mojo, we collected the opening gross and the number of theaters during opening weekend for those 165 movies. In order to evaluate the influences of studios, directors, actors, writers and composers, we also collected the total gross of past movies for each studio, director, actor, writer and composer. Therefore, we can use the past cumulative gross for specific actors and directors to represent the star power of movies.

From Metacritic, we collected the score and date of critic reviews for each movie. The critic reviews are normally written prior to the movie release date, while the audience can only review after the release date, which means that we can only have critic reviews for our project. Besides, compared with other movie review platforms such as Rotten Tomatoes, Metacritic focuses on critic reviews and its data has fewer missing values.

From Google Trends, we collected the web search volume on each day for each movie. The period of data is from December 1, 2015 to January 1, 2017. Figure 1 shows an example of Google trends of movie *Rogue One*, *Caption America: Civil War* and *Finding Dory*. From the figure, we can find there is a dramatic increase of search volume before the release date.

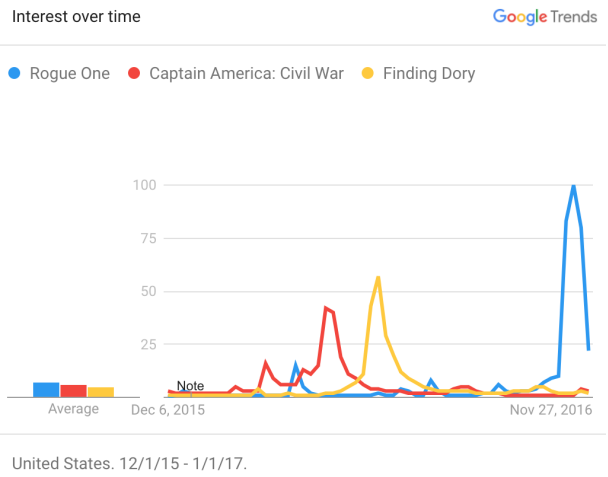

Figure 1: Google Trends

As Google Trends only releases score representing search interest relative to the highest point on the chart for the given region and time, so we calculated the relative score for each movie and the score for Google trends can represent the real web search volume.

B. Data Integration

We integrated all the collected data from 4 websites together to one movie database. As some movie names are different between IMDB, Box Office Mojo and Metacritic, we standardized the movie names and merged them together.

When integrating IMDB into database, we took four variables:

1) ***budget***: continuous variable denoting the budget of the movie
2) ***runtime***: continuous variable denoting the run time of the movie
3) ***MPAArating***: binary variable denoting whether the movie belongs to R class
4) ***genre***: categorical variable denoting the genre of the movie (recoded into dummy variables)

When integrating Box Office Mojo into database, we took two variables:

1) ***open_gross***: continuous variable denoting the opening weekend gross of the movie
2) ***open_thea***: continuous variable denoting the number of theaters during opening weekend

We also created three variables representing the power of studio, directors and actors:

1) ***Actor_gross***: continuous variable denoting the past cumulative gross of the actor before the movie
2) ***Director_gross***: continuous variable denoting the past cumulative gross of the director before the movie
3) ***studio_gross***: continuous variable denoting the cumulative gross of the studio from 2011-2015

To make it simple, we only considered the gross of lead director and lead actor, which should be able to represent the power of directors and actors. Also we considered the gross of the studio in the past 5 years. It is a little subjective to use the period of 5 years, but it should represent the recent power of studio.

When integrating Metacritic into database, as we need to predict opening weekend performance before release, we only took critic reviews before release date into account. We created two variables:

1) ***mean_score_mtc***: continuous variable denoting the average score given by critic reviewers before the release date
2) ***comment_count_mtc***: numeric variable denoting the count of critic reviews before the release date

When integrating Google Trends into database, we created three variables:

1) ***trend_a_day***: continuous variable denoting search query volume which is one day prior to the release date
2) ***trend_a_week***: continuous variable denoting average search query volume which is seven-day period prior to the release date
3) ***trend_two_week***: continuous variable denoting average search query volume which is from fourteen-day to seven-day period before the release date

C. Data Characteristics

The integrated database comprises of a variable matrix with 14 features and 165 rows of movies. The output we wish to predict is the box office gross of the opening weekend. Table 1 and 2 show the brief structure of the integrated database (genre is not shown).

Besides the variables shown in part B. Data Integration, we also created 3 more variables:

1) ***hot_season***: binary variable denoting if the movie opens during the summer or during November and December. The data is made from IMDB's release date information.
2) ***tier-a-series***: binary variable denoting if the movie is among the top 50 franchises sorted by total gross. The data is from boxofficemojo.com/franchises/.
3) ***strong_competition***: binary variable denoting if the movie is released in the month with large number of other films.

	name	open_gross	open_thea	budget	runtime	MPAArating	Sci-Fi	Adventure	Action	Animation	...	Actor_gross	studio_gross
0	Rogue One: A Star Wars Story	155081681	4157	200.0	133	0	1	1	0	0	...	5.477600e+07	8400.8000
1	Finding Dory	135060273	4305	200.0	103	0	0	0	0	1	...	6.333170e+08	8400.8000
2	Captain America: Civil War	179139142	4226	250.0	147	0	0	1	1	0	...	2.343813e+09	8400.8000
3	The Secret Life of Pets	104352905	4370	75.0	90	0	0	0	0	1	...	4.942307e+08	7358.0000
4	The Jungle Book (2016)	103261464	4028	175.0	105	0	0	1	0	0	...	2.981236e+09	8400.8000
5	Deadpool	132434639	3558	58.0	106	1	0	0	1	0	...	NaN	6160.9000
6	Zootopia	75063401	3827	150.0	108	0	0	0	0	1	...	4.000155e+08	8400.8000

Figure 2: Table 1 - Data example

	count	mean	std	min	25%	50%	75%	max
open_gross	165.0	2.141920e+07	3.136284e+07	227354.0000	4.294232e+06	1.111188e+07	2.381734e+07	1.791391e+08
open_thea	165.0	2.603800e+03	1.066413e+03	540.0000	1.945000e+03	2.886000e+03	3.384000e+03	4.370000e+03
budget	165.0	5.051455e+01	5.656766e+01	0.9000	1.000000e+01	2.500000e+01	6.500000e+01	2.500000e+02
runtime	165.0	1.085212e+02	1.556128e+01	76.0000	9.600000e+01	1.070000e+02	1.180000e+02	1.510000e+02
MPAArating	165.0	4.060606e-01	4.925911e-01	0.0000	0.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00
Director_gross	81.0	6.952929e+08	1.242949e+09	277000.0000	1.009515e+08	3.360583e+08	7.986025e+08	9.378314e+09
Actor_gross	124.0	1.706252e+09	1.338194e+09	277000.0000	7.343242e+08	1.266520e+09	2.240122e+09	7.411925e+09
studio_gross	163.0	4.540069e+03	3.194047e+03	0.0082	6.161000e+02	5.566000e+03	7.358000e+03	8.521900e+03
mean_score_mtc	155.0	5.824417e+01	1.576189e+01	7.6000	4.794436e+01	5.890323e+01	6.825636e+01	9.630435e+01
comment_count_mtc	155.0	2.989677e+01	1.244843e+01	1.0000	2.400000e+01	3.100000e+01	3.800000e+01	5.100000e+01
trend_a_week	150.0	7.271691e+01	1.114829e+02	0.0000	2.431804e+01	3.944063e+01	6.978476e+01	6.968789e+02
trend_two_week	150.0	3.426637e+01	4.636446e+01	0.0000	1.026076e+01	1.982754e+01	3.619571e+01	2.793938e+02
trend_a_day	150.0	1.270901e+02	2.140341e+02	0.0000	4.059938e+01	6.740012e+01	1.303992e+02	1.758049e+03
hot_season	165.0	4.121212e-01	4.937151e-01	0.0000	0.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00
tier-a-series	165.0	7.878788e-02	2.702275e-01	0.0000	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00
strong_competition	165.0	5.575758e-01	4.981859e-01	0.0000	0.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00

Figure 3: Table 2 - Summary Statistics

D. Data Cleaning

1) One-Hot Encoding: As the genre is categorical variable, we used one-hot encoding to change a variable of one column and n distinct values to n dummy variables. The number of columns for the genre variable is now increased from 1 to 19.

2) Dealing with missing values: There exist plenty of missing values in columns. The major missing values come from 84 missing values in *Director_gross* column and 41 missing values in *Actor_gross* column. Because all columns with the missing values are continuous variables, we decided to fill in missing entries with column mean.

E. Data Visualization

We created the pairwise scatter plot of the 12 variables in Figure 4, 5, 6, 7 and the correlation heat map in Figure 8.

From Figure 4, we can see from plot on the top-left corner shows the distribution of *open_gross*, it has a right-

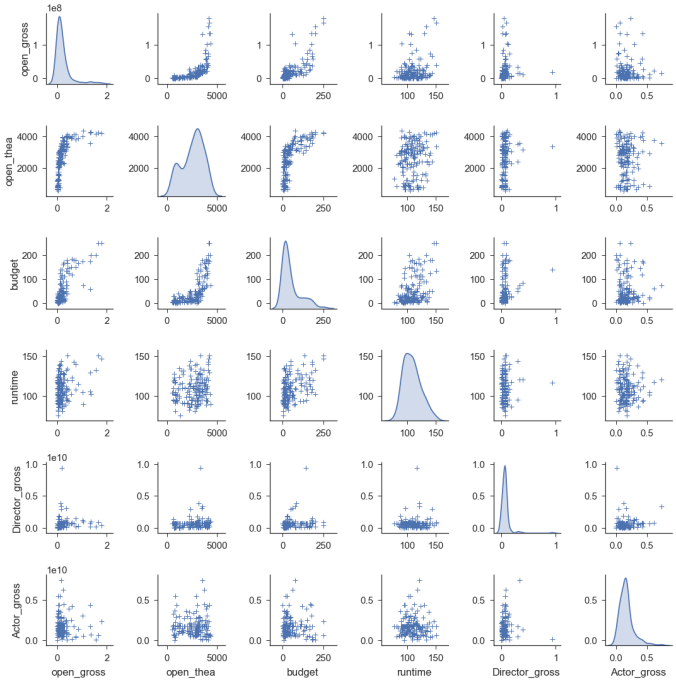

Figure 4: Pairwise plot between variables - part1

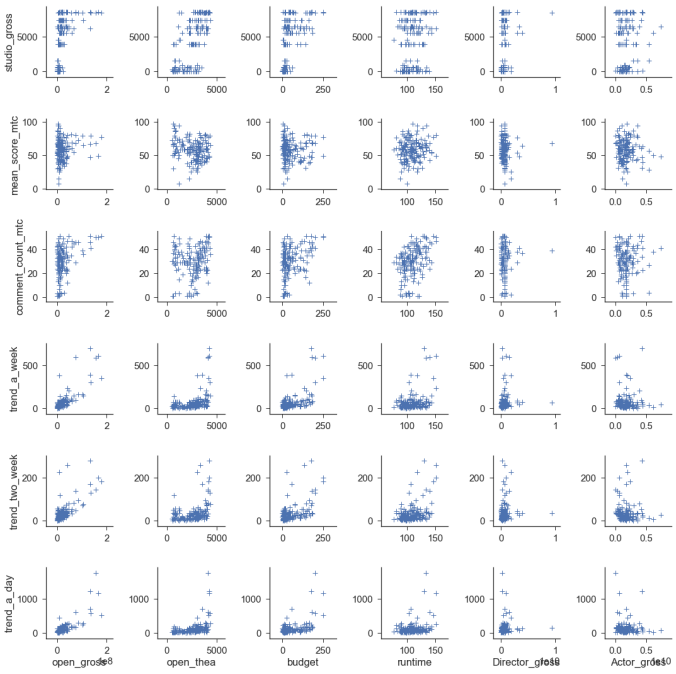

Figure 5: Pairwise plot between variables - part2

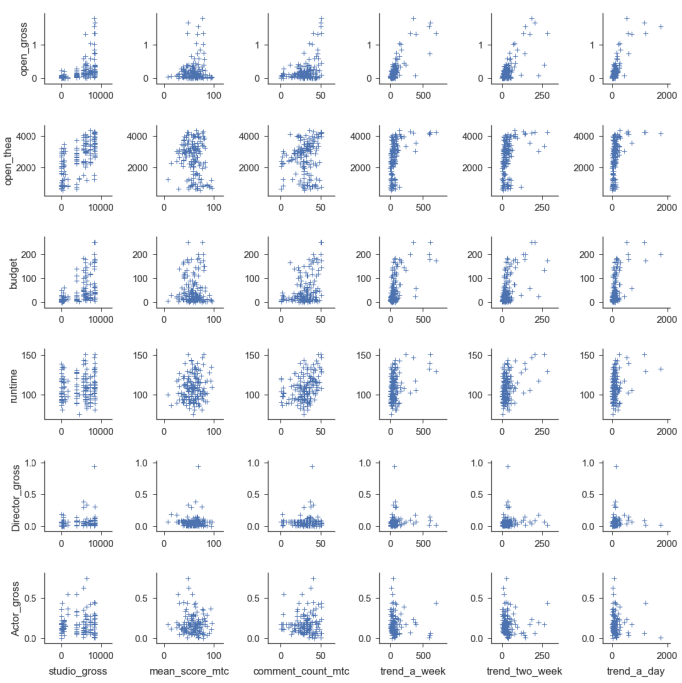

Figure 6: Pairwise plot between variables - part3

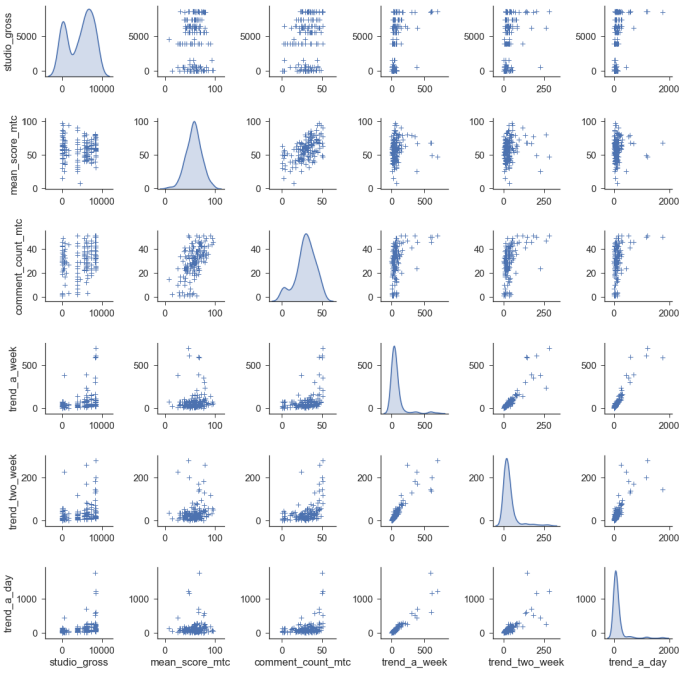

Figure 7: Pairwise plot between variables - part4

skewed, nearly normally distributed shape with the long tail representing some blockbuster movies.

From the figures, the independent variables - *trend_a_week*, *trend_a_day* and *trend_two_week* have strong linear correlation with *open_gross*, our dependent variable. It is clear that the Google trends is a good indicator for opening gross.

There are some other independent variables have strong correlation with *open_gross*. They are *open_thea* with 0.82

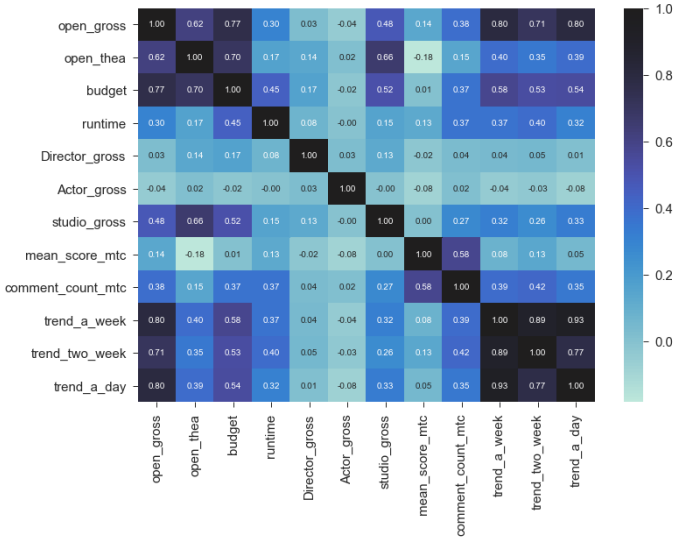

Figure 8: Correlation

correlation, *budget* with 0.77 correlation and *studio_gross* with 0.48 correlation.

However, *Director_gross*, *Actor_gross* and *mean_score_mtc* do not seem to have a strong correlation with *open_gross*. We can see from the plot that values for *Director_gross* indicate a small standard error. Also, to our surprise, the average score of critic reviews has low correlation with the opening gross. The reason might be that some movies like *Carol* and *The Moonligh(2016)*, with low gross but high review score, do not have enough budget to support marketing promotion, or the audience have different tastes of movie from the critics.

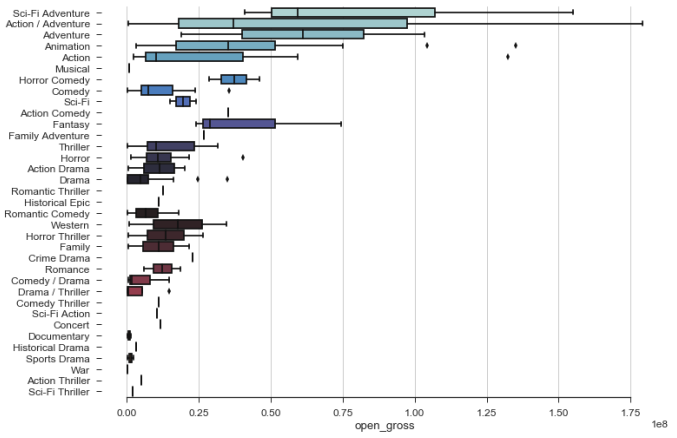

Figure 9: Box Plot of Genre

From Figure 9, we can get a sense of the variable genre's effect on the opening gross. The Sci-Fi Adventure and Adventure tend to yield the highest median returns. The success of Sci-Fi Adventure and Adventure movies can partly explain the popularity of Sci-Fi Adventure franchises such as Star War and Marvel movies. There are many genres that have low opening grosses such as Comedy, Horror and Action drama.

II. Model Selection

We split the data randomly into 80% training set and 20% test set.

Since we have a relatively small data set, if we partition the available data into three sets (training, validation and test set), we will drastically reduce the number of samples which can be used for model training. Therefore, for each model, we used a 5-fold cross-validation on the training set to calculate the λ of regularizer that minimizes the mean absolute error (MAE) of cross-validation. That λ was used in regularization along with the loss function to train the model, which was then applied to the test set. Finally, we compared the results in terms of each model's training error and test error.

Below are three models we used to run the regression.

A. Quadratic loss function with l2 regularization

We started from the quadratic loss function with l2 regularizer to get a sense of how our data set can fit with a generic regression model. As adding regularizers decreases the risk of over-fitting, we tried l2 regularizer first. The ridge regression is defined as below:

$$minimize \sum_{i=1}^{k}(y_i - w^T x_i)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

1) Cross-validation: We performed a 5-fold cross-validation to find out the λ of regularizer that minimizes the mean absolute error (MAE) of cross-validation. Figure 10 shows the results. After training data with different values of λ, we found that the smallest MAE on validation set of cross-validation occurs when λ = 7.

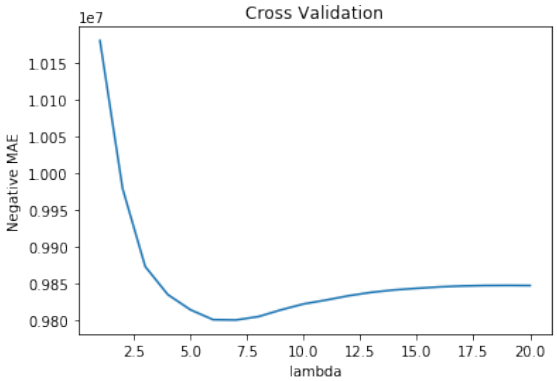

Figure 10: Cross-validation of Ridge regression

From the results above, we chose $\lambda = 7$ as our optimal regularization parameter.

2) Fit the model: We used the objective function with the optimal regularization parameter to fit the training set, then used the model trained on training set to make a prediction on test set. Table 1 shows the mean absolute error of the training set and test set.

Ridge Regression			
	$\lambda_{optimal}$	**Training Error**	**Test Error**
MAE	7	7.19^{e6}	9.58*e^6

Table I: Table for Ridge Regression

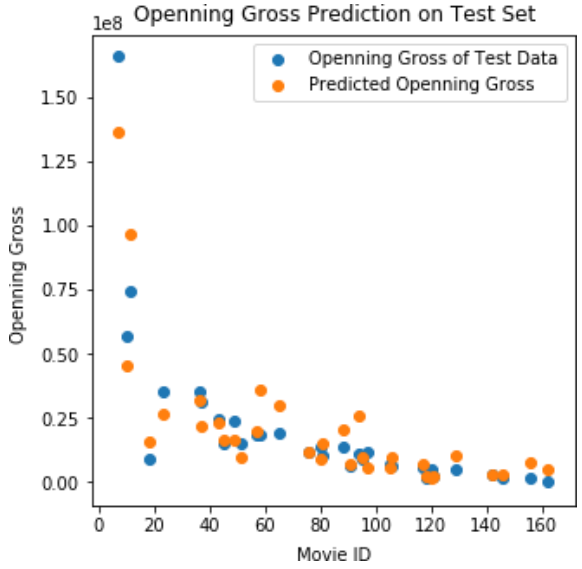

Figure 11: Prediction of Opening Gross on Test Set

Figure 11 shows our prediction for opening gross on test set. Movie ID on this plot refers to the unique ID we assigned to each movie in our data set sorted by opening gross. We can see that our model predicts better for movies with Movie ID$>$ 20, which are those having less popularity, and lower box office grosses than those with Movie ID$<$ 20. We can infer that the Ridge regression produces a more accurate result for movies which turned out to have lower opening grosses.

B. Quadratic loss function with l1 regularization

The ridge regression is defined as below:

$$minimize \sum_{i=1}^{k}(y_i - w^T x_i)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

Compared to l2 regularizer, l1 tends to produce a more sparse solution.

1) Cross-validation: Through the same procedure as ridge regression before, we got the following cross-validation results: From the results above, we found that when λ is $4.75 * e^7$,

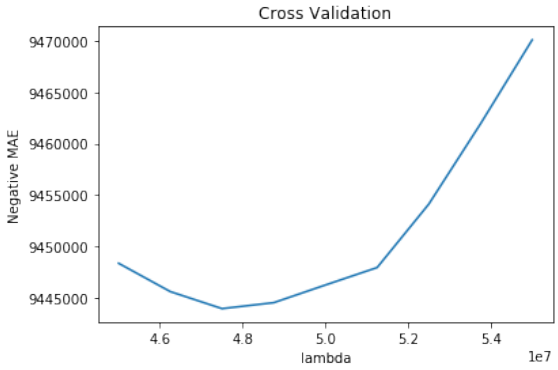

Figure 12: Cross-validation of Lasso regression

the mean absolute error of the model reaches minimum. The reason why this optimal $lambda$ is much larger than the one from Ridge regression ($\lambda = 7$) is that the l1 regularization got a l1 norm for coefficients w so the λ for l1 regularization should be larger than l2 regularization in order to prevent over-fitting by penalizing large w more.

2) Fit the model: Through the same procedure before, we got the following MAE results on training set and test set:

Lasso Regression			
	$\lambda_{optimal}$	**Training Error**	**Test Error**
MAE	4.75*e^7	7.74^{e6}	9.16*e^6

Table II: Table for Lasso Regression

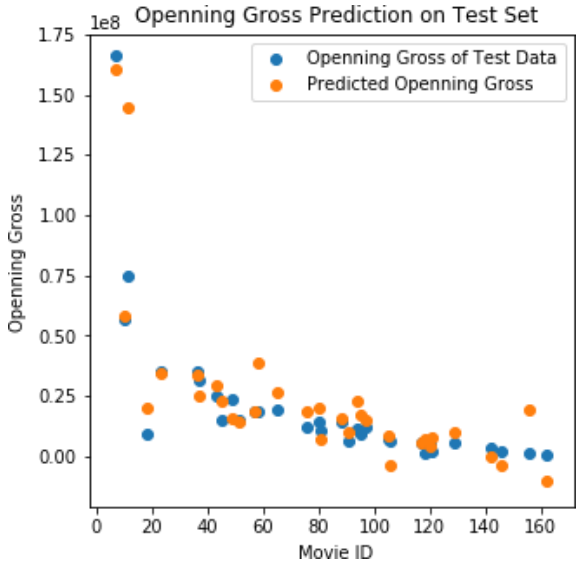

Figure 13: Prediction of Opening Gross on Test Set

From the results above, we found that Lasso regression also produces a more accurate prediction for movies with Movie ID

>20, so it fits better on movies which turned out to have lower box office grosses than movies with Movie ID <20. However, one thing that makes Lasso regression different from Ridge regression is that Lasso regression fits slightly better on test set than Ridge regression as the test error of Lasso regression is slightly lower than that of Ridge regression. To see it from graph: it fits significantly better on the first movie (the movie with highest box office gross).

C. Huber loss function with l2 regularization

According to the definition of Huber loss function, it is less sensitive to large outliers than quadratic loss function because it is linear-like when $w^T x$ is very large. Moreover, unlike l1 loss function, it is differentiable at anywhere, so it can produce more smooth solutions compared to l1 loss function. Since our data set contains many blockbusters' box office grosses (i.e. *Rogue One: A Star Wars Story*, *Captain America: Civil War*) which may need to be considered as outliers, we chose Huber loss function combined with l2 regularizer.

The objective function of this model is defined as below:

$$minimize(\frac{1}{n}\sum_{i=1}^{n} huber(y_i - w^T x_i) + \alpha||w||^2)$$

where the huber loss function is defined as:

$$huber(z) = \begin{cases} \frac{1}{2}z^2 & |z| \leq 1 \\ |z| - \frac{1}{2} & |z| > 1 \end{cases}$$

1) Cross-validation: Through the same procedure before, we got the following cross-validation results:

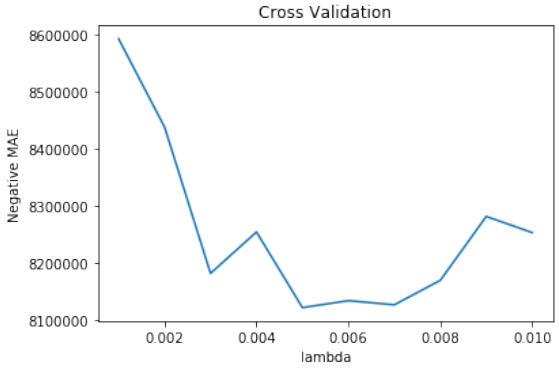

Figure 14: Cross-validation of Huber regression

From the results above, we found that when $\lambda = 0.005$, the MAE of our model reaches minimum. Compared to previous two models, the optimal λ for Huber regression is smallest. This might result from the fact that Huber regression is less sensitive to large outliers than Ridge regression.

2) Fit the model: Through the same procedure as Ridge regression and Lasso regression before, we got the following MAE results on training set and test set: From the results above, we found that like previous two regressions, Huber regression also predicts on test set more accurately for movies with Movie ID>20. However, the Huber regression has the lowest test error among all three models, which might result from Huber regression's better penalty efffcts on outliers than Ridge regression and Lasso regression.

Huber Regression			
	$\lambda_{optimal}$	Training Error	Test Error
MAE	0.005	7.38*e^6	8.37*e^6

Table III: Table for Huber Regression

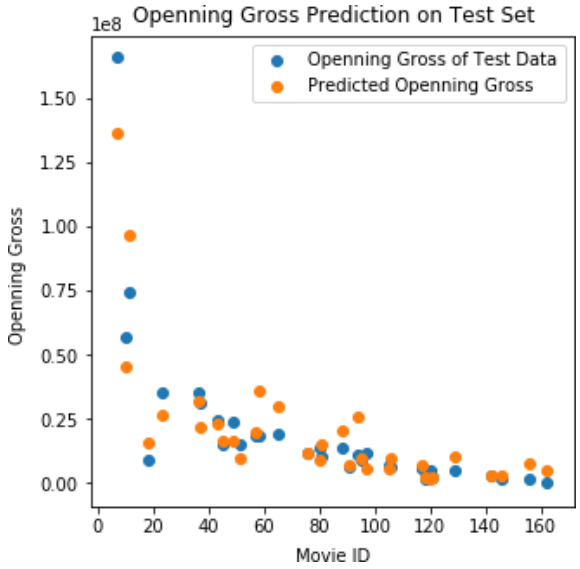

Figure 15: Prediction of Opening Gross on Test Set

D. Interpretation

Based on the results we got, Huber regression with l2 regularizer produces the lowest test error as well as training error. The reason behind this might come from its insensitivity to large outliers as our data set contains many blockbusters' box office grosses, which may be considered by the model as outliers. So the Huber regression with l2 regularizer eliminates the effects of outliers and appearances of crazy predictions.

We also found that our models predict better on small movies than those blockbusters in the sense that those small movies, which usually come from individual production studios, require smaller budgets and attract less social attention. These kinds of movies are less known to most people because they are more literary and more likely to earn stable and predictable box office grosses compared to those blockbusters

whose box office grosses could fluctuate greatly according to market popularity and effect of reputation.

III. Predicting Box Office One-Week Before Release

While predicting the gross of opening weekend the day before release is valuable as it is likely to be most accurate, but it doesn't leave much time for production companies and cinemas to react. What if we can predict the box office performance one week before the release? Therefore, we decided to apply the same regression methods with the same loss functions and regularizers to a special version of data set.

Compared with predicting one-day before release, the data set for this prediction lost two independent variables: *trend_a_week* and *trend_a_day* by Google Trends, so we have *trend_two_week* which is the average search query volume from fourteen-day to seven-day period before the release date. And we updated *mean_score_mtc* and *comment_count_mtc* by Metacritic to one-week before the release date.

Similar to predicting one-day before release, we tried Huber loss function with l2 regularization, lasso regression and ridge regression respectively. The results are below.

	$\lambda_{optimal}$	**Training Error**	**Test Error**
Huber Regression	0.0002	8.93*e^6	1.02*e^7
Lasso Regression	400000	7.86*e^6	1.08*e^7
Ridge Regression	6	8.37*e^6	1.29*e^7

Table IV: Table for Predicting Box Office Gross One Week Before Release

From the results above, similar to predicting one-day before release, we found that the Huber regression also produced the lowest test error compared to two other models. However, the test errors in three regression methods are larger than errors from predicting one-day before release. The reason is that we did not use two important independent variables: *trend_a_week* and *trend_a_day* generated by Google Trends.

IV. Conclusion

As moviegoers spend increasing time engaging in discussions and searching for information online, we could get a rough idea about the prospects of movies through their popularity. Knowing not only basic profiles of movies but also information like studio gross, Google trends, and Metacritic score, we are trying to take a step further by discovering relationships between opening gross and such types of information.

Among the linear models we explored, Huber regression with l2 regularizer outperforms the Ridge regression with l2 regularizer and Lasso regression with l1 regularizer. All of them predict the opening gross with higher accuracy on movies with small budgets and little popularity among people. This is a reasonable result because those small movies usually earn more stable and predictable box office grosses compared to those blockbusters whose box office grosses could fluctuate greatly according to their market popularity and reputation.

For the practical aspects of this project, by understanding accurate patterns, movie marketers could have a better ability to evaluate movies' quality and adjust their post-release marketing strategies accordingly to increase revenue. For future possible improvements, we could acquire more samples into our data set and try more robust and flexible models to get better predictions.