An Insightful Analysis of IMDB Data

Group 3: Arthur Krivoruk, Sibo Xu, Souravi

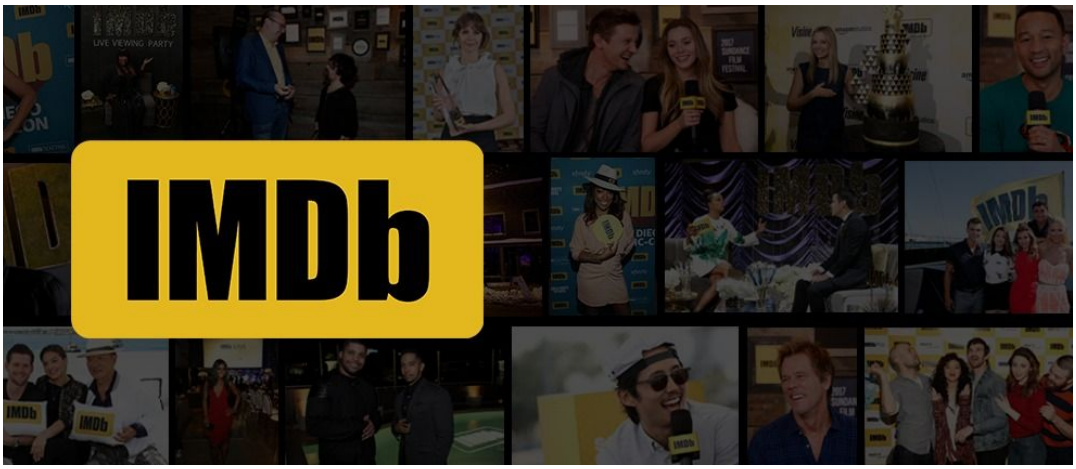


Table of Contents

1. Introduction

The success of a movie, measured by its gross profit, can be perceived as the product of several key factors.With that in mind, several of these factors are readily available through the internet, and with the correct algorithms, this data can be acquired, manipulated, and used to predict the gross revenue of future films. Such models would be invaluable to the firms who invest hundreds of millions of dollars on films and expect to see a significant return on these investments.

2. Data Scraping

The data of the project was scraped using Python code in several steps. Python packages including Beautifulsoup and Selenium were the main tools for the programming. For each movie, there was an unique movie ID in the IMDb database. Hence, a list of movie IDs was scraped and saved, then 18 different features of the movies and various reviews were scraped from the homepage of each movie. Meanwhile, two features (i.e. official Facebook page likes and follows) were scraped from each movie's official Facebook page.

There were two different datasets, movie review and movie gross used for the project. Movie gross profit dataset was including the gross profit for each movie and the most common 18 features which might have influence on the profit. The movie review dataset was only containing users reviews of different movies, which were 932 movies and 15~25 reviews (including review text and rates from the review writers) for each movie.

3. IMDb Review Analysis

3.1 EDA of Review Data Set

There were 21,368 rows in the dataset, where each movie had around 20 reviews. For each review, there was an ordinal score scaling from 1 to 10. The distribution of the score was shown in the *Figure 1 (Left).* As it could be seen, the distribution was not perfectly balanced. The most frequent rate was 1 point, and many of reviews were rated as 7~10 point. The imbalanced data set might introduce imbalanced classification result, which would be explained more in the latter part. The length of the reviews were shown in the *Figure 1 (Right).* The reviews were in similar length, so there wasn't necessary normalization. After cleaning and reorganizing, the data was ready for preprocessing.

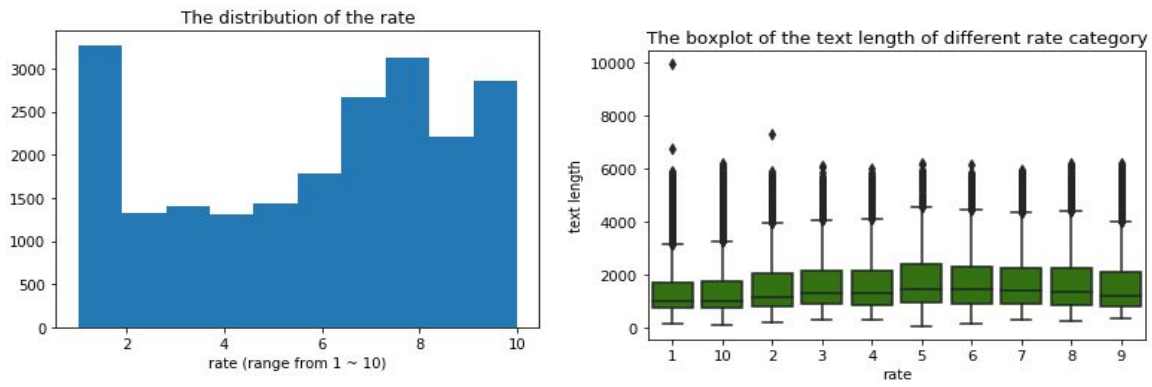


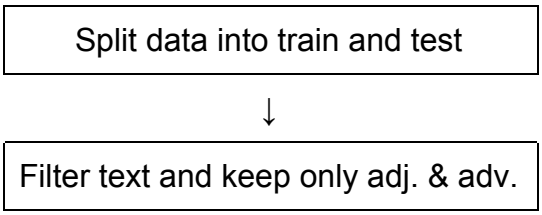
Figure 1. Histogram and boxplot of the movie review data set

3.2 Sentiment analysis using classification model

There were two different models to be discussed in this part. They were all based on the same review dataset, but were using distinguish methodology. The first method was performing classification (logistic regression) on the movie reviews. The goal was to classify if a review is either positive or negative.

3.2.1 Data preprocessing:

The main issue with the review data set was that it was all in plain-text format. The logistic regression algorithm would have a feature vector as input in order to perform the classification task[1]. One of the most commonly used representations was the bag-of-words approach, where each unique word in a text would be represented by one number[2].In this case, the order of the words was not considered in the process. Particularly, the text preprocessing method used for this part was tf-idf, short for term frequency–inverse document frequency, which was a numerical statistic that was intended to reflect how important a word was to a document in a collection or corpus. The complete preprocess was shown in the Figure 2. After splitting the dataset into train and test data, the spaCy was used for part-of-speech filtering. After trying to keep different combinations of part-of-speech tag and comparing the results of them, the best result giving by keeping only adjective and adverb of the text. Since most of the information of sentiment was expressed by adjective and adverb, this choice was also reasonable.



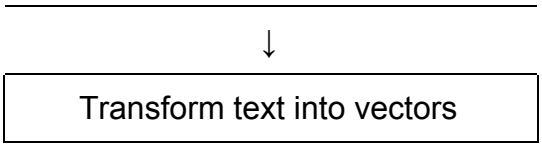


Figure 2. Text preprocessing

Then, the filtered text were transformed into vectors using TfidfVectorizer, where transformed each review into a vector. Since there were plenty of reviews, many zeros features for the presence of a word in the collection. The output was a matrix, which contained all reviews to be used for classification algorithm later.

3.2.2 Model and Results

The logistic regression algorithm were used for classification problem. There were 10 classes in the original dataset ranging from 1 to 10, where 1 was the most negative review and 10 was the most positive review.

For this multiclass problem, the confusion matrix of the test data was shown in the Figure 3. Most of the reviews were classified as 1, 2, 8 and 9. The performance of the classifier was poor. The reason was that there was no clearly boundary between two classes that made the classifier hard to work.

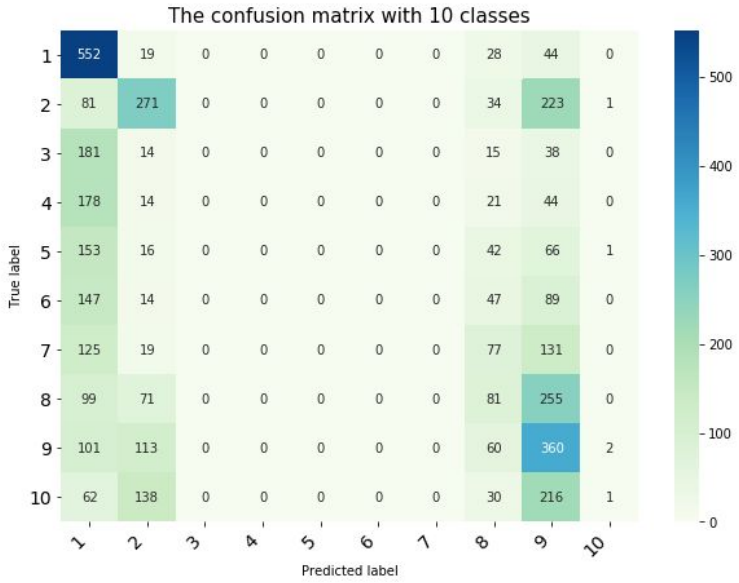


Figure 3. Confusion matrix for 10 classes

In order to improve the performance of the classification, a useful method would be reducing the number of classes. In the previous result, it showed that all the neutral reviews were classified incorrectly, but there was a clearly boundary between positive review and negative review. Hence, by reducing the number of classes, the result might be improved. After rearranging the rate of the review, there were only three categories

of the rate, i.e., negative, neutral and positive. The classification result was shown in the Figure 4. As it can be seen the recall for the positive class was 73.71% and the recall for the negative class was 67.25%. Meanwhile, the accuracy of the model was dramatically improved from 39% (with 10 classes) to 62.94%. The performance of the model was much better than the previous model. The model was performing well especially on the positive classes. This was because the rearranged data set had a much higher number of positive reviews than other two types of reviews. This meant that the model was more biased towards positive reviews compared to negative ones and neutral ones. But the model could also be improved especially on the neutral class, where precision was only 53.64%. But after tuning parameters of the model, the logistic regression model could not be improved anymore.

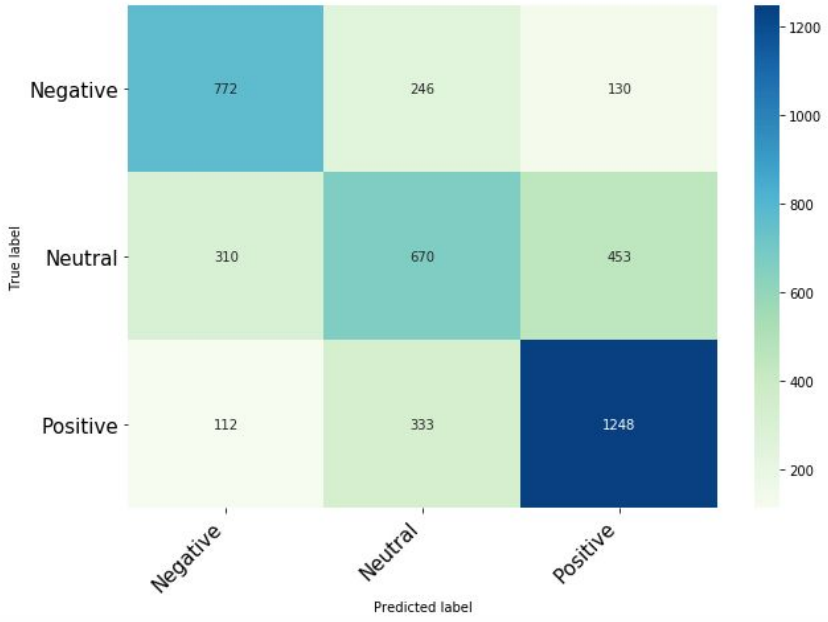


Figure 4. Confusion matrix for 3 classes

As mentioned before, the features for the classification was generated through tf-idf method, where each review was represented by a vector. Combining the features' name and the coefficients of the logistic regression model, the weight of words in reviews could be known. For example, the words with the highest weight are show in the left column of the Figure 5. They were all positive adjective or adverb that had positive influence on the probability. On the contrast, the right column of the Figure 5 showed the features with the lowest coefficient, where most of them were negative words. Hence, this logistic regression model worked well for this data set. And it would be useful to classify a review either positive or negative.

	feature	coef		feature	coef
64	brilliantly	17.919056	593	unfunny	-36.153004
228	highly	14.713949	282	laughable	-24.976116
48	beautifully	14.447535	175	forgettable	-24.230923
273	just right	13.719381	35	awful	-23.390179
433	pleasantly	13.205397	56	bland	-21.959345
111	definitely	12.540775	316	mediocre	-21.058868
150	excellent	12.532326	614	whatsoever	-19.184256
14	amazing	12.502997	438	poorly	-19.007989
550	superb	12.306540	623	wooden	-18.168074
164	fascinating	12.183011	553	supposedly	-17.685713

Figure 5. Coefficient of the features of logistic regression model (left column: highest; right column: lowest)

3.3 Lexicon based sentiment analysis

In this part, a lexical approach using three packages, the Vader, Senticnet and Pattern to determine the overall polarity of the movie review[3][4] was performed. Based on the sentiment scores given by these tools, a linear regression model was built to decide different weight of these tools for predicting sentiment score of a movie review. Those three packages were some of the commonly used ones. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media[5]. It gives positive, negative, neutral and compound score for a sentence or text. Only positive and negative were used for this project in order to keep the same format as other tools provided. Senticnet is concept-level sentiment analysis, that is, performing tasks such as polarity detection and emotion recognition by leveraging on semantics and linguistics instead of solely relying on word co-occurrence frequencies. It works only for separate words and generated a positive score or negative score of a given word. The Pattern module bundles a lexicon of adjectives that occur frequently in product reviews, annotated with scores for sentiment polarity (positive or negative) and subjectivity.

3.3.1 Data preprocessing:

The movie review data set was the same one as before. For this part, the preprocessing of the data was mainly transforming each review into scores given by three sentiment tools introduced before. Each tools would generate a positive score and a negative score respectively. Vader could only handle short text or sentences. Pattern and Senticnet worked in similar way on the words. Hence, movie reviews were tokenized to be the input for Pattern and Senticnet.

After preprocessing of the review text, several boxplots (Figure 6) were generated to show how those sentiment scores distributed across the entire dataset. On the left column, the relationship between the negative sentiment score and the review rating could be seen. The negative scores tended to be large when the actual review rating was negative (rate less than 4). On the contrary, when the review rating was positive (rate greater than 7), the positive sentiment scores were large. The sentiment scores from Vader and Pattern showed more clearly relationship to the rate. The scores from Senticnet was more evenly distributed across the entire range.

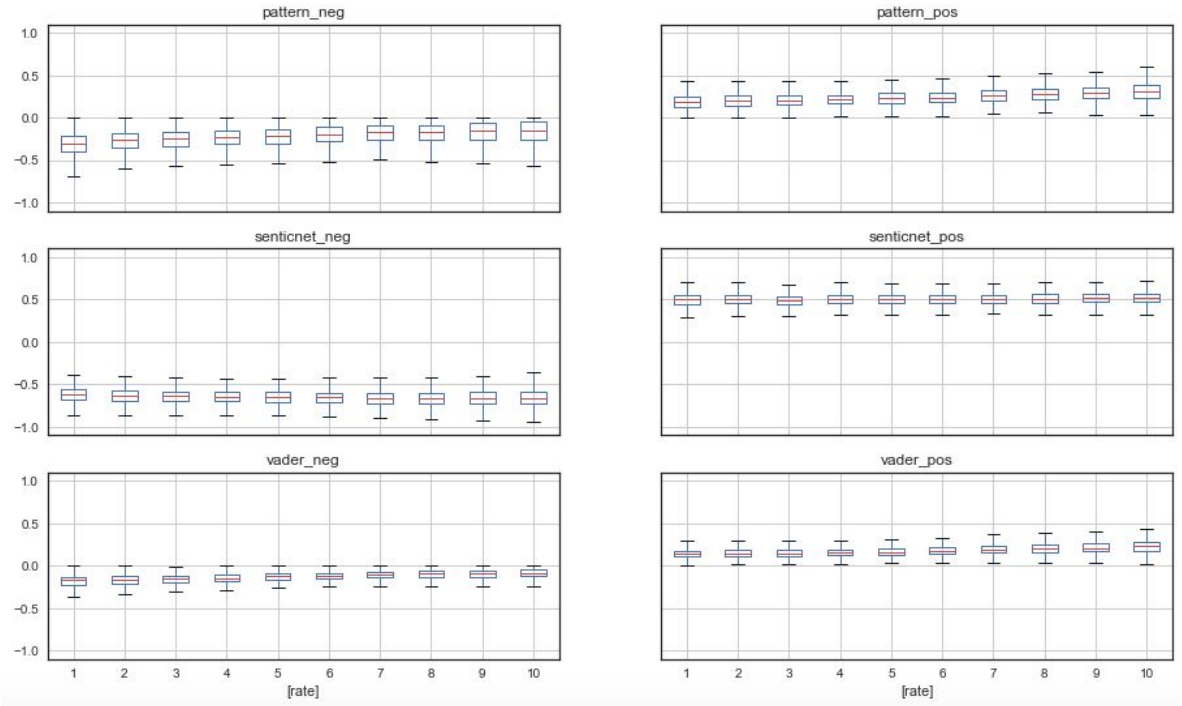


Figure 6. The relationship between sentiment score and the rating

3.3.2 Model and Results

The model was to predict the rating of a movie's review based on the sentiment score provided by Vader, Senticnet and Pattern. As the EDA shown before, there were some linear relationship between the rating and the sentiment score. Hence, a simple linear regression was used first to see how the sentiment score were related to rating. There were several steps of developing the models to reach a relatively little error. As it showed in the Figure 7, the first step was only fitting one sentiment score into the model. Then, the stepwise regression would be conducted to select the best combination of the features. Finally, ridge regression were performed to improve the model.

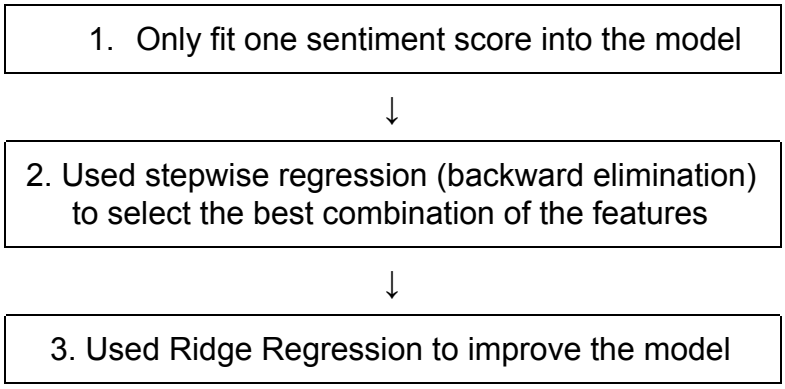


Figure 7. The steps of improving linear regression model

In order to evaluate the models, the root-mean-square error (RMSE) was calculated for each of the model. As it can be seen from the Table 1, three different sentiment scores were used separately to fit the model. As for the results, the Vader and Pattern had similar performance. They had reasonable adjusted R^2, hence they were better than the Senticnet score. In fact, only using the Senticnet score was not reliable to predict the rating of the review. When conducting the stepwise regression, all of the features were fitted into the model. Then, there were several ways for the model to select the best combination of the features. In this case, the standard was comparing the F statistics, which the lowest one was the best. As for the result, fitting all features happened to be the best combination of the model, that gave the best model. When three sentiment scores were all utilized to predict the rating, the RMSE of the model would only be 1.851, which meant that the prediction rating would only be 1.851 point more or 1.851 point less than the actual rating. This was an acceptable result, which could be used to provide sentiment score of a movie review. It would not give an accurate sentiment score, but would provide information of review's sentiment. For example, a predicted score for a review was 3.56, it meant that the review is a negative review or a review with many negative opinions. Then, in order to improve the model, a ridge regression model was used. Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of coefficients. However, as it could be seen, the ridge regression had just decreased the RMSE by 0.001.

Features	**Adj-R^2**	**MSE**	**RMSE**
Vader_pos Vader_neg	0.495	4.518	2.126
Pattern_pos Pattern_neg	0.496	4.466	2.113
Senticnet_pos Senticnet_neg	0.07	8.200	2.863

All features (backwards regression)	0.612	3.426	1.851
All features (Ridge Regression)	0.612	3.422	1.850
Intercept: 4.50 *Vader_pos: 7.78* *Vader_neg: 12.01*		*Pattern_pos: 7.52* *Pattern_neg: 5.42*	*Senticnet_pos: 3.38* *Senticnet_neg: 2.28*

The equation of predicting the sentiment of a review would be

Sentiment score = 4.50 + 7.78 Vader_pos + 7.52*Pattern_pos + 3.38*Senticnet_pos + 12.01*Vader_neg + 5.42*Pattern_neg + 2.28*Senticnet_neg*

In the equation, the '_pos' were the positive sentiment score from the packages, whereas the '_neg' were the negative sentiment score (negative number). The model could be used for predicting sentiment from other platforms, e.g. Twitter and Instagram. In these websites or applications, there were no rating system for the review or users' opinions. Hence, film companies could use this model to analyze audiences' opinions all over the Internet.

4. Gross Revenue Prediction Using Regression

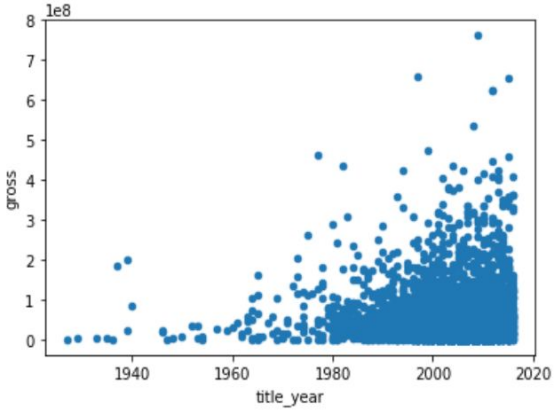
4.1 EDA & Data Processing

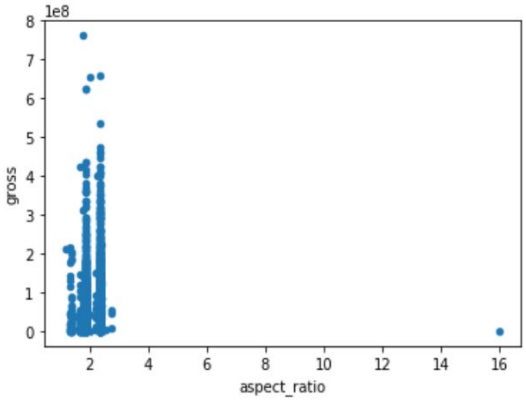
After the initial scraping process was completed the initial step was to make the data readable and able to be processed by the various python algorithms in use. To do so all of the columns with numeric data which were deemed so intuitively when the project idea was first conceived and planning for what data was to be scraped had occurred.

To complete the transition from string to float some of the columns needed to be changed significantly. These columns were a mix of both numeric data and string characters for they appeared that way from the data sources. Therefore, a column with values taking the form of "[$1,000,000,000 gross revenue]" required the special characters, and the phrasing "gross revenue" to be removed. Once that step was complete, using excel's formatting capabilities, these values were converted to be registered as numeric and were subsequently ready to be observed for future planning stages.

Some of the columns were deemed vital to the models as factors but were entirely made up of string characters. To allow these data points to interact with the model, dummy variables were created in their place; each with their own arbitrarily set values. In the case of the variable deemed ‘Color’, if a movie was in color, that value was set to zero and subsequently a black and white movie had a value of one for that column. The ‘Language’ variable was treated similarly The value of ‘English’ was set to the number zero and any alternative string was subsequently converted to have the value one. Lastly, this methodology was applied to the ‘rating’ variable. The four possible values that this factor could take were ‘PG’, ‘PG13’, ‘R’, and ‘Not Rated’. To divide them, rating suitable for children (‘PG’ and ‘PG13’) were given a value of zero and the other two granted a value of one, completing the transition of all vital variables to numeric values.

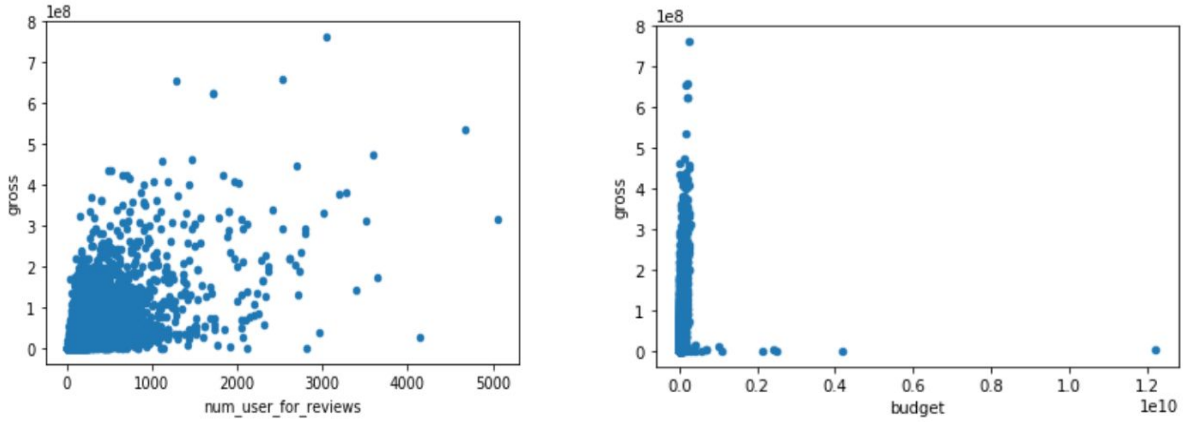
Initially, all of the independent variables were plotted against the dependent variable of gross revenue to check for outliers and inconsistencies. Two major issues arose when this was done. The first issue centered around the aspect ratio and the ‘Color’ variables. Due to the relatively short scope of the scraped data (roughly 7 years of data) these factors only had a single value for each observed data point, making these two columns utterly useless. To resolve this issue, more data was needed, and movie data encompassing the same factors was scraped for a much longer duration of roughly 40 years. Once again, each independent variable was plotted against gross revenue, and the results were as follows:





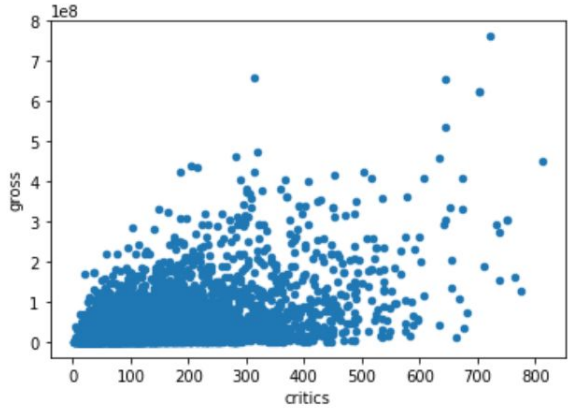
The leftward plot depicts the year in which a movie is initially released versus the gross revenue of the film. It seemingly has an exponential relationship and this data helps shed light on the growth of the movie industry as a whole. If upon closer analysis, the revenue generated by a later film is directly proportional to year, then the variable may not be as significant as one would believe when viewing this plot. At this stage, this variable would be a valuable addition to the model.

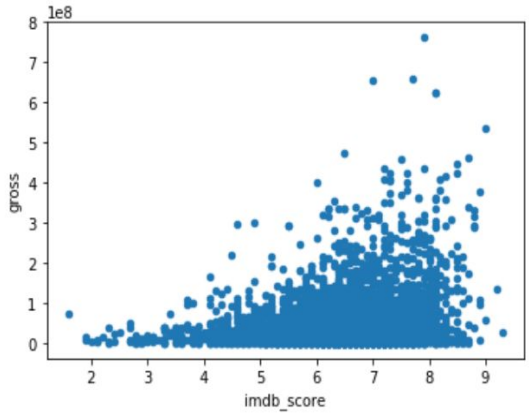
The right-hand graph shows the aspect ratio of the films encompassed within the dataset versus their respective gross revenue. At first glance, the aspect ratios take three main values which is reflexive of the evolution of film technology over the last several decades. The closest value to the origin represents the archaic ‘square’ aspect ratio found within older films during the birth of the medium. The middle value is a slightly rectangular aspect ratio intended to fit in televisions which were promptly increasing in terms of technological efficiency. The final value is the standard aspect ratio found in modern day television screens and theatres. There are some outliers in this dataset, and due to their values not being sensible, these instances were promptly removed. If this factor is found to be insignificant, then that would serve to imply that regardless of the current advent of technology, seeing films is a sought after pastime that will excel regardless of the current stage of technology.



The right figure shows the number of users who have left reviews of the film, but these users are regular movie goers. As seen in the graph, the gross revenue does seem to tangibly increase as more individuals voice their opinions about the film. This would attest to the need of films to heavily advertise to reach as many people as possible to maximize the likelihood of these customers to influence others to see the film as well. Were this factor to be significant, then this assertion would very much be justified.

The leftward graph shows a film’s budget versus the gross revenue, and this plot shows the very obvious biases and limitations of the data that is being worked with for the purposes of this project. As evidenced by the horizontal axis, most of these films have a similarly scaled budget. This is due to the movies which make up the dataset all being featured films, all with theatrical releases. This excludes many films, namely indie films which feature low budgets along with sometimes large critical and financial success. Moreover, there are a few very extreme outliers which depict very expensive movies that failed financially. These outliers were also removed since they failed due to unaccounted factors that are too subjective and complex for the models in the subsequent sections.





The right plot featuring the total number of critics versus gross revenue follows the same logic as the number of non-professional reviewers. If enough people voice their opinions about a film, it may entice additional patronage. Therefore, the relationship depicted in the scatter plot is subsequently intuitive and offers a similar proportional effect.

The final plot is the most intuitive of all of the scraped attributes. It relates the overall score that a film receives and the resulting gross revenue that the film is able to attain. A higher critical score indicates that reviewers believe that the average patron would find the film enjoyable spurring more people to see the film. Of course there are outliers due to extraneous factors such as innate fan bases, but upon inspection, the score seems to hold a very important proportional effect.

4.2 Model: Simple Linear Regression

The initial model used to discern the relationship between the various aforementioned factors and gross revenue was a simplistic linear regression. In regards to its implementation, the sklearn linear model was first attempted. This resulted in a number of complications due to various inconsistencies with the data's formatting such as dimension size and this methodology was then scrapped in favor of a more intuitive module. Instead, the statsmodel module was used in place of sklearn due to its more intuitive syntax and overall better quality[6].

The code itself is written in a format that emulated the coding language R, which was specifically created to be used for statistical problem solving. This is reflected within the model itself, for the core regression is written in the format of:

```
import statsmodels.formula.api as sm
sm.ols(dependent variable ~ variable 1 + variable 2 + … ).fit()
```
[1]

The largest boon of using this format is that the model is written with its proper mathematical notation, thus helping to rectify any transcription errors that may arise

when dealing with the data and code. Additionally, the module outputs a very easy to read table that clearly lists important attributed used in judging the model's overall effectiveness and significant features.

For this initial linear regression the following variables and shorthand variable names were used to regress against gross income:

x1 - number of critics
x2 - budget
x3 - length
x4 - voted users total
x5 - user review total
x6 - score
x7 - aspect ratio
x8 - language
x9 - year
x10 rating
x11 - color

Upon using these variables into the ols regression algorithm, the following outputs were available to be interpreted:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.492
Model:	OLS	**Adj. R-squared:**	0.491
Method:	Least Squares	**F-statistic:**	331.4
Date:	Mon, 30 Apr 2018	**Prob (F-statistic):**	0.00
Time:	18:56:14	**Log-Likelihood:**	-72287.
No. Observations:	3775	**AIC:**	1.446e+05
Df Residuals:	3763	**BIC:**	1.447e+05
Df Model:	11		
Covariance Type:	nonrobust		

The most striking aspect of this model is the very low percent of gross revenue is explained by these factors. With an r squared and adjusted r squared of 0.49, this model is very inefficient and offers very little in understand the problem at hand. This

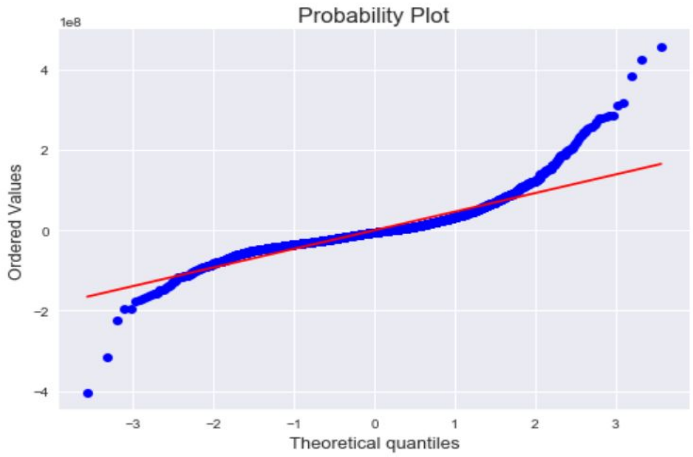
does, however, serves to establish a benchmark for which that improving upon the model can strive to beat.

Additionally, the model provided the coefficients of all variables and intercept along with all each features' statistical significance scores.

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.082e+09	2.08e+08	5.204	0.000	6.74e+08	1.49e+09
x8[T.1]	-2.57e+07	4.26e+06	-6.039	0.000	-3.4e+07	-1.74e+07
x10[T.1]	-3.354e+07	1.69e+06	-19.851	0.000	-3.69e+07	-3.02e+07
x11[T.1]	-2.144e+07	4.69e+06	-4.573	0.000	-3.06e+07	-1.22e+07
x1	1.012e+05	1.01e+04	10.045	0.000	8.15e+04	1.21e+05
x2	0.0146	0.004	3.942	0.000	0.007	0.022
x3	1.266e+05	4.12e+04	3.074	0.002	4.58e+04	2.07e+05
x4	215.6733	9.696	22.243	0.000	196.663	234.684
x5	1.372e+04	3353.693	4.091	0.000	7144.699	2.03e+04
x6	-4.118e+06	9.92e+05	-4.149	0.000	-6.06e+06	-2.17e+06
x7	-3.643e+06	2.42e+06	-1.505	0.132	-8.39e+06	1.1e+06
x9	-5.17e+05	1.04e+05	-4.994	0.000	-7.2e+05	-3.14e+05

Based on the above t values, there are no significant factors shown by the model. This would mean that this regression shows no significance between the above factors and gross revenue, which is very counterintuitive based on the industry as a whole. Due to this discrepancy, a more advanced model is required to get a better understanding of the intricacies of these factors.

To help better understand the model, some visualization tools were used to model the regression residuals and show any further discrepancies in the data. The first of such plots is a qq plot which was used using the spicy package[7].



As noted from the above plot, the data is relativistically normal meaning that most gross revenues seem to center around the industry average with few outliers. These are likely the few films that buck trends, such as *IT* which toppled box office number despite being a relatively cheap film to make due to child actor costs being minimalist.

4.3 Model: Linear Regression with Interaction Effects

To improve upon the previous model, interaction effects were added to the regression to help improve the overall efficiency of the model. Using the *Stats Models* module, interactions were added using the following format:

```
import statsmodels.formula.api as sm
sm.ols(dependent variable ~ … + x1*x2 + x1*x3 + x1*x4 +...).fit()
```

This format included each instance of two way interactions between each combination of all eleven factors that are being used.

Subsequently, the model revealed the following results:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.708
Model:	OLS	**Adj. R-squared:**	0.704
Method:	Least Squares	**F-statistic:**	183.9
Date:	Mon, 30 Apr 2018	**Prob (F-statistic):**	0.00
Time:	18:56:23	**Log-Likelihood:**	-71245.
No. Observations:	3775	**AIC:**	1.426e+05
Df Residuals:	3725	**BIC:**	1.429e+05
Df Model:	49		
Covariance Type:	nonrobust		

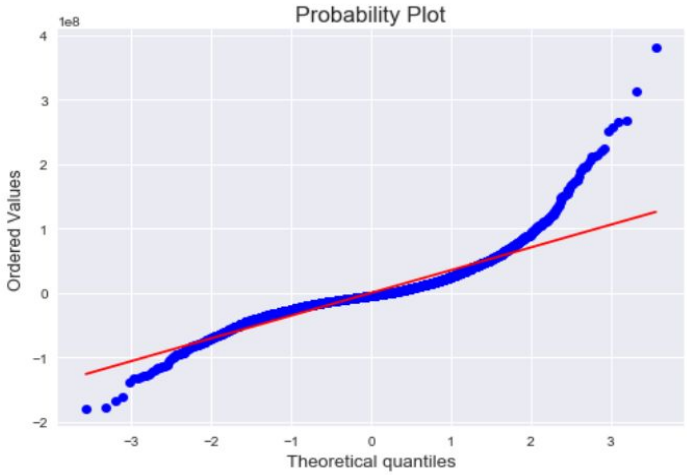
This model greatly improves upon the original model and helps it to explain the relationship between the factors and gross revenue by increasing the r squared by nearly half of the original metric. Due to this incredibly large improvement for this metric, the combined effect of factors is incredibly important. With that in mind, additional testing for higher level interactions were subsequently taken since these gains were attributed solely to second degree interactions only.

Additionally, these improvements translated to the coefficients and significances associated with this improved model. The corresponding results can be seen below:

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-8.018e+04	3.98e+04	-2.014	0.044	-1.58e+05	-2127.763
x8[T.1]	-3613.0476	9.26e+04	-0.039	0.969	-1.85e+05	1.78e+05
x10[T.1]	-7.716e+04	4.57e+04	-1.689	0.091	-1.67e+05	1.24e+04
x11[T.1]	-3.531e+05	1.41e+05	-2.501	0.012	-6.3e+05	-7.63e+04
x8[T.1]:x10[T.1]	3.638e+06	7.82e+06	0.465	0.642	-1.17e+07	1.9e+07
x8[T.1]:x11[T.1]	3.802e+06	1.16e+07	0.329	0.742	-1.89e+07	2.65e+07
x10[T.1]:x11[T.1]	-1.395e+07	1.03e+07	-1.357	0.175	-3.41e+07	6.2e+06
x1	-1.123e+07	2.25e+06	-4.997	0.000	-1.56e+07	-6.82e+06
x1:x8[T.1]	-3879.2060	4.25e+04	-0.091	0.927	-8.73e+04	7.95e+04
x1:x10[T.1]	2.316e+04	1.48e+04	1.560	0.119	-5940.337	5.23e+04
x1:x11[T.1]	-3.052e+04	4.16e+04	-0.734	0.463	-1.12e+05	5.11e+04
x2	-31.1798	4.063	-7.674	0.000	-39.145	-23.214
x2:x8[T.1]	-0.5166	0.032	-15.914	0.000	-0.580	-0.453
x2:x10[T.1]	0.1242	0.020	6.087	0.000	0.084	0.164
x2:x11[T.1]	-0.3275	0.184	-1.784	0.075	-0.688	0.033
x3	1.309e+07	2.22e+06	5.901	0.000	8.74e+06	1.74e+07
x3:x8[T.1]	3.013e+05	1.26e+05	2.387	0.017	5.38e+04	5.49e+05
x3:x10[T.1]	-1.111e+05	6.9e+04	-1.609	0.108	-2.46e+05	2.43e+04
x3:x11[T.1]	1.664e+05	1.75e+05	0.953	0.340	-1.76e+05	5.09e+05
x4	1.292e+04	2164.790	5.968	0.000	8675.887	1.72e+04
x4:x8[T.1]	-99.5621	51.134	-1.947	0.052	-199.815	0.691
x4:x10[T.1]	-80.3617	17.403	-4.618	0.000	-114.482	-46.241
x4:x11[T.1]	11.3363	29.277	0.387	0.699	-46.064	68.737
x5	1.557e+06	1.06e+06	1.475	0.140	-5.13e+05	3.63e+06
x5:x8[T.1]	1.16e+04	1.46e+04	0.792	0.428	-1.71e+04	4.03e+04

(For the sake of saving space all insteraction combinations were not shown here. To see all combinations of interaction effects and their respective coefficients and t statistics, please visit: the following link: https://github.com/zacwentzell/BIA660D_Group_3_Project/blob/master/Arthur's%20Files/project.ipynb)

Unlike the previous model, several variables and interactions are very much significant, notably, the number of reviewers, the number of critics, and total amount of reviews. Additionally, many of their interactions were significant as well, prompting one to believe that they are indeed keys to a film's success.



Just as the plot of the previous model, the data is relativistically normal which implies that most gross revenues seem to center around the industry average with few outliers.

The improvement of the linear regression was an important step forward and the interaction effect regression offered many insights. To form a concrete conclusion however, an additional test must be completed to assure the validity of this model's results. To do this, an ANOVA model transform is constructed with the interaction model as a base, to determine if these factors are truly significant.

4.4 Model: ANOVA

The anova transformation of a regression model is used to determine significance among several factors and their interaction effects. For the purposes of this project the ANOVA model is used to gauge the validity of any significance found in the previous models. To initiate the ANOVA transform, the following code was used:[3]

```
from statsmodels.stats.anova import anova_lm
anova_lm(model2, typ=2)
```

(Where model2 is the interaction effect regression model)

With that transform completed the ANOVA could be called, and in doing so, the table in which its values take are promptly displayed. The subsequent output is as follows:

	sum_sq	df	F	PR(>F)
x8	6.045750e+14	1.0	0.412400	5.207932e-01
x10	1.026380e+16	1.0	7.001274	8.179416e-03
x11	2.134355e+14	1.0	0.145591	7.028063e-01
x8:x10	3.175891e+14	1.0	0.216638	6.416409e-01
x8:x11	1.586771e+14	1.0	0.108239	7.421767e-01
x10:x11	2.700800e+15	1.0	1.842304	1.747634e-01
x1	8.314139e+15	1.0	5.671345	1.729418e-02
x1:x8	1.219223e+13	1.0	0.008317	9.273417e-01
x1:x10	3.569670e+15	1.0	2.434989	1.187397e-01
x1:x11	7.888567e+14	1.0	0.538105	4.632653e-01
x2	1.541102e+16	1.0	10.512357	1.196240e-03
x2:x8	3.712777e+17	1.0	253.260620	3.211900e-55
x2:x10	5.432142e+16	1.0	37.054413	1.265086e-09
x2:x11	4.663657e+15	1.0	3.181233	7.456990e-02
x3	1.662213e+15	1.0	1.133850	2.870246e-01
x3:x8	8.353925e+15	1.0	5.698484	1.702916e-02
x3:x10	3.795045e+15	1.0	2.588724	1.077118e-01
x3:x11	1.332682e+15	1.0	0.909066	3.404242e-01
x4	1.188882e+18	1.0	810.975029	1.456270e-161
x4:x8	5.557835e+15	1.0	3.791180	5.159836e-02
x4:x10	3.125951e+16	1.0	21.323128	4.011222e-06
x4:x11	2.197940e+14	1.0	0.149929	6.986257e-01
x5	4.622322e+15	1.0	3.153037	7.586742e-02
x5:x8	9.205798e+14	1.0	0.627958	4.281560e-01
x5:x10	1.873138e+16	1.0	12.777283	3.553043e-04
x5:x11	3.332087e+14	1.0	0.227293	6.335667e-01
x6	7.532987e+13	1.0	0.051385	8.206831e-01
x6:x8	7.434308e+15	1.0	5.071184	2.438478e-02
x6:x10	4.203501e+15	1.0	2.867345	9.047748e-02
x6:x11	8.244905e+15	1.0	5.624119	1.776554e-02
...	...	...	...	...

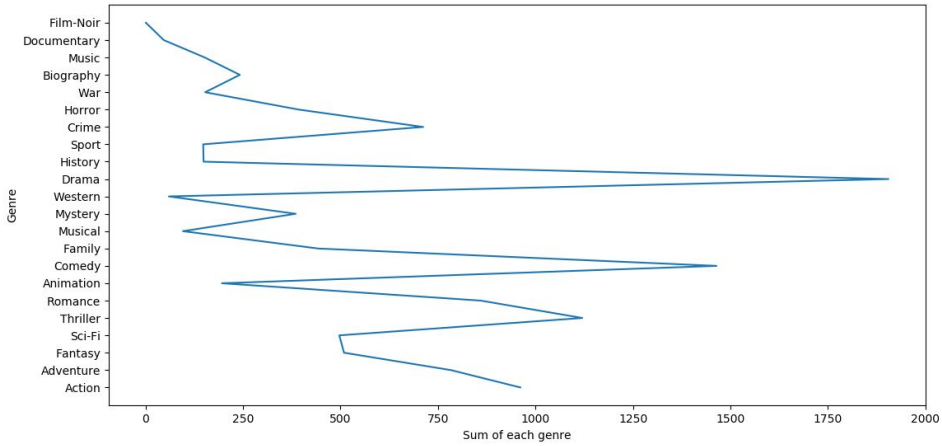
(The full list of factors and interaction effects is not provided. To see the output in its entirety please visit https://github.com/zacwentzell/BIA660D_Group_3_Project/blob/master/Arthur's%20Files/project.ipynb)

The ANOVA results concur with that of the interaction regression. The number of reviewers, the number of critics, and total amount of reviews are all once again heavily significant factors allowing us to form a conclusion based on fundamental understanding of the film industry and the consensus of the models.

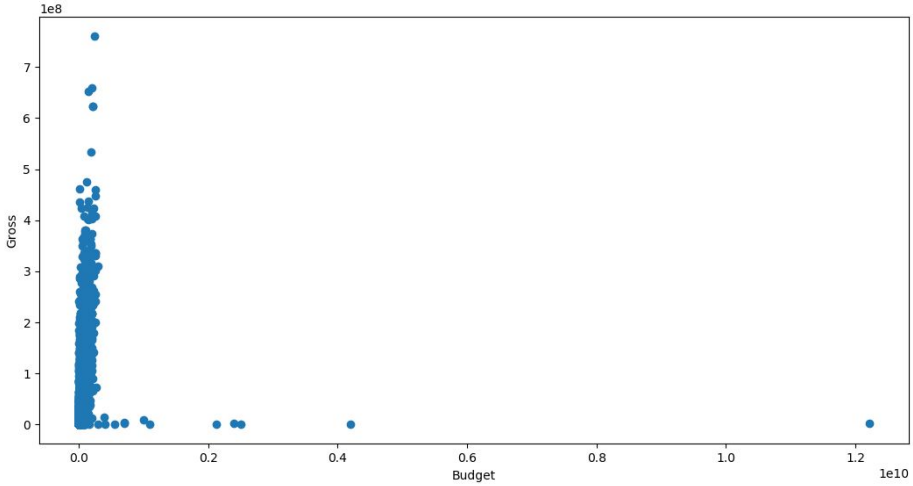
5. Random Forest

5.1 EDA

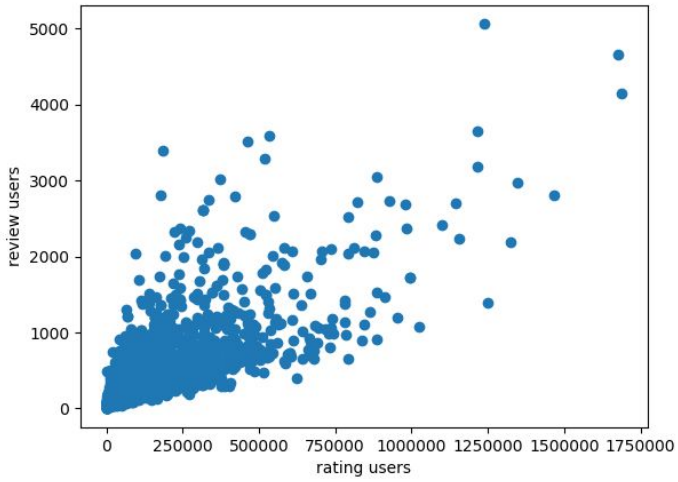
Formulating a graph between the genres and the total gross profit made by these genres.



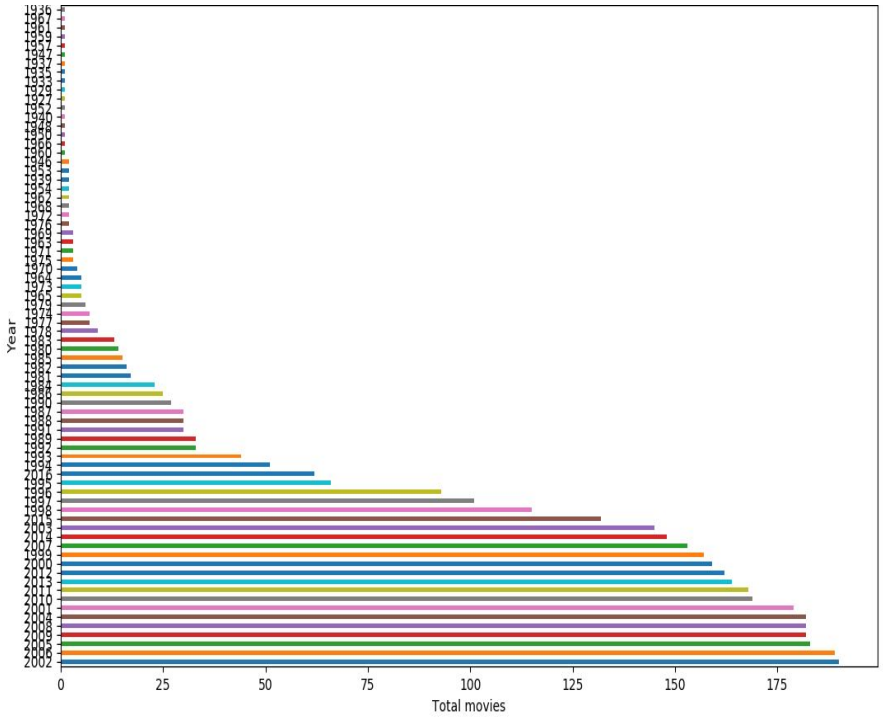
From the above graph we can say that, most of the movies are being made in the drama, followed by comedy and thriller genres. Thus, we can conclude that, most of the movies are successful in these genres compared to the other ones.



This graph shows the relationship between the gross value and the amount of money invested, i.e., budget of the movie. Most of the movies are made in about similar budget and making large profits. Although, there are some movies, the outliers which are having large budget but very few profits, showing that they failed miserably at the box office.



This graph shows how many users prefer rating to reviewing a particular movie. Most of the users prefer rating since it's more easier and quick method to evaluate the movie. The number of reviewers are comparatively very few that the rating users. The ratio is 1:250 for review user to rating user.



From this graph we can understand that, maximum number of movies were being made in 2002, which is 195. The graph is designed to show the number of movies in the increasing order.

5.2 Random Forest Regression

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

In our analysis, we created date set, “data” of the following:

```
x1= df['critics']

x2= df['gross']

x3= df['budget']

x4= df['imdb_score']

x5= df['num_user_for_reviews']

x6= df['num_voted_users']
```

Our x contains x1, x3, x4, x5 and x5, whereas the y is x2 i.e., Gross.

This is how the data set looks like:

	critics	budget	imdb_score	num_user_for_reviews	num_voted_users
0	723.0	237000000.0	7.9	3054.0	886204.0
1	302.0	300000000.0	7.1	1238.0	471220.0
2	602.0	245000000.0	6.8	994.0	275868.0
3	813.0	250000000.0	8.5	2701.0	1144337.0
4	462.0	263700000.0	6.6	738.0	212204.0

First we divide the data into training set and test set, where

```
X train shape= (3020, 5)
Y train shape=  (3020,)
X test shape= (755, 5)
Y test shape= (755,)
```

Then we perform the Random Forest regression to fit the model.

```
model = RandomForestRegressor(random_state=0)

fit= model.fit(X_train, y_train)
```

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
           oob_score=False, random_state=0, verbose=0, warm_start=False)
```

Upon analysis of the regression we get that,

```
Test data R-2 score: 0.606
Test data Spearman correlation: 0.735
Test data Pearson correlation: 0.779
```

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model. Or:

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%:

In our analysis, the R-square value is 60%, which is not a very satisfactory value to exhibit the correlation between the x and y.

Pearson's correlation is a measure of the linear relationship between two continuous random variables. It does not assume normality although it does assume finite variances and finite covariance. When the variables are bivariate normal, Pearson's correlation provides a complete description of the association. In our case, it is 77.9%, which is fairly significant.

Spearman's correlation applies to ranks and so provides a measure of a monotonic relationship between two continuous random variables. It is also useful with ordinal data and is robust to outliers (unlike Pearson's correlation). As per our analysis, the spearman correlation is 73.5%.

COEFFICIENT RELATION

Then, we use the feature_importances function on the model in order to predict the feature coefficient for each of the variables

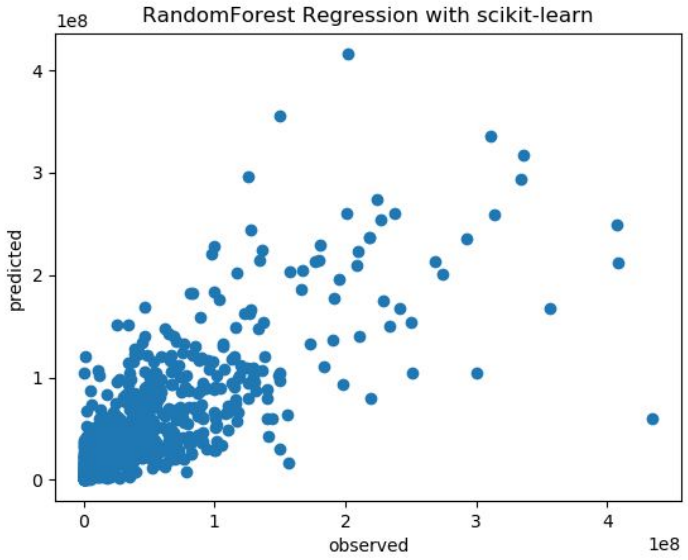
	4	1	0	3	2
values	0.507969	0.257112	0.093086	0.074056	0.067777

Thus, from this we can say that, the IMDB score and critics ratings are the most important and affecting factors for evaluating the Gross of a movie.

The equation we get is,

Gross= 0.507969*user_voting+ 0.257112*budget+ 0.093*critics_rating + 0.074*user_reviews+ 0.0677*imdb_rating.

From the analysis we can compare that, the random forest regression and linear regression give us different understanding of the effect of factors. There are no significant contributing factors.



Conclusion

The IMDb Review Analysis

There were two methods performed to analyze the sentiment of movies' review. The first one was using classification model to classify a review as one of the three classes, i.e., positive, neutral, and negative. This method was focusing on the text itself. And it gave a relatively good classification result, which could classify most of the positive and negative reviews correctly.

Meanwhile, a lexicon based algorithm was conducted to predict the sentiment score of a review. This algorithm utilized three sentiment tools, which were assigned by different weight using linear regression model. This could be used to predict rating of people's opinion from other platforms (e.g. Twitter, Facebook), which was not including users' rating system.

IMDb movie gross revenue prediction

There are several important notions that can be derived from which factors were significand and from which factors were not. Some of the most influential factors include the number of critics, the total number of reviews, and the total number of voted users.This indicates that marketing for a movie is one of the most important aspects of a successful film.

Due to the fact that the overall score of a film is not a heavily significant aspect of the model, this implies that being a critically successful film is not mutually exclusive with being a profitable film. Such a phenomenon can be attributed to niche films with a large innate fan base, such as superhero films.

Additionally due to the insignificance of technical aspects of a film such as aspect ratio, whether it's in color, or the length, this heavily favors the notion that a well recieved film can be created regardless of artistic norms. This is grounded in reality as the film *Sin City* was mostly in black in white, *A Quiet Place* was a fairly lengthy film that featured little to no speaking, which extended its length significantly, and *Titanic*, being an archaic film by today's standards all performed very well and grossed significant amounts of revenue.

Bibliography

[1]. Sentiment Analysis For Yelp Review Classification, The Tensorist Medium, Vivian Rajkumar,https://medium.com/tensorist/classifying-yelp-reviews-using-nltk-and-scikit-learn-c58e71e962d9

[2]. C. C. Aggarwal, Machine Learning for Text. Cham: Springer International Publishing, 2018.

[3]. Tungthamthiti, P., Shirai, K., & Mohd, M. (2014). Recognition of sarcasm in tweets based on concept level sentiment analysis and supervised learning approaches. In Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation, PACLIC 2014 (pp. 404-413). Faculty of Pharmaceutical Sciences, Chulalongkorn University.

[4]. T.P.SahuandS.Ahuja,"Sentiment analysis of movie reviews: A study on feature selection classification algorithms," in Proceeding of International Conference on Microelectronics, Computing and Communications, 2016, pp. 1–6.

[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]. https://www.statsmodels.org/stable/index.html is the official documentation for the Stats Models Module

[7]. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.probplot.html was used for the creation of the plot