MS&E 226 Fundamentals of Data Science Project
Author: Taylor Song, Sandy Lee

1. Introduction

1.1 Data Collection

This dataset is from Kaggle and obtained from a website called data.world. It consists of 28 columns of variables and 5043 rows of movies. The link to the dataset and description of 28 variables is in Appendix Table 1. In terms of any concerns about data completeness and accuracy, the data accuracy seems reasonable while some movies have empty variables. We will discuss the data cleaning process in section 1.3. This dataset may not be comprehensive enough to predict overall behavior of all the movies in the world. However, we believe this dataset can be used as a learning purpose.

1.2 Problem Statement

Using the movie dataset, it would be interesting to explore what types of movies are more successful and analyze important factors that contribute to such success. Accordingly, we intend to predict IMDB movie scores using linear regression and classification models. In our regression analysis, we take IMDB scores as continuous response variable and analyze the rest of covariates in the dataset. For our classification model, we introduce a binary response variable which we define as 1 if IMDB score is greater than 7, and 0 otherwise. As an entertaining movie that appeals to audience often leads to commercial success, the results will be useful to movie production companies.

1.3 Data visualization

First, we look at the score distribution. Our histogram (Appendix Figure 1) illustrates that the distribution is left skewed with minimum score of 1.6 and maximum score of 9.5. In addition, the third quartile is 7.2, which suggests that we could possibly consider movies with a score greater than 7 as "good."

Then, we also explore the distribution of title year, from which we see that most of the movies in the dataset were released after 1990. Perhaps, advancement in technology during the 1990's led to this growth in movie production. We will consider movies released after 1990 for our study since data before this date might not be representative due to lack of records.

1.4 Data Cleaning and Exploration

1.4.1 Preparing Data

For the "country" variable, we decide to only keep data for movies produced in the USA. This is because the "budget" variable for movies produced in other countries was not converted to USD. To convert the budgets into USD, we would have to consider the exchange rate at the time which would be too temporal. We also removed 45 duplicate rows found in the dataset.

1.4.2 Missing values

For gross, budget, title year, and content rating, the values are hard to impute since each of them are specifically pertinent to each movie. Hence we will delete rows that have null values for gross, budget, and title year. We are left with 3857 observations after deleting the rows.

1.4.3 Removed Variables

- "movie_title" has been removed since it is an ID variable that has no bearing on our analysis. Likewise, movie_imdb_link" has been removed.
- Variables of names ("director_name," "actor_1_name," "actor_2_name," and "actor_3_name") have been removed since names in the dataset were too diverse. Likewise, "plot_keywords" variable has been removed.
- "Language" has been removed since we are only looking into movies produced in the USA. All of the movies produced in the USA in this dataset are produced in English.
- "Color" has been removed since only 4% of the movies are black and white, and this indicates that color variable can be deemed as nearly constant.
- "Genre" has been removed since each record of genres is often comprised of multiple types (ie. Drama|Music|Mystery|Romance|Thriller - in alphabetical order), and we couldn't come up with a fair way to assign the best matching genre for each record.

1.5 Exploring Correlations - ggpair

Before fitting a regression model, we will investigate the correlation between the variables. With our choices of response variable, we can find the following correlations between the covariates and the response variable from the ggpair plot attached in the Appendix Figure 2.

"Num_voted_users" has the highest positive correlation with imdb scores while "aspect_ratio" has the highest negative correlation. Meanwhile, "title_year" has the lowest positive correlation with imdb scores and "facenumber_in_poster" has the lowest negative correlation.

There are also several noticeable correlation characteristics between predictors. "Actor_1_facebook_likes" is highly correlated with the "cast_total_facebook_likes," with the correlation value of 0.95. "Facenumber_in_poster" and "duration" has highest negative correlation value of -0.138.

We try to look for any mutual correlations between the aspect ratio and number of voted users predictors identified above (Appendix Figure 3). Many movies fall under the 2.35, 1.85, and 1.37 ratio, which explains the congregated scatter dots. From the plot, we can tell that the higher the aspect ratio the more it is likely to review votes from users. Hence although the aspect ratio is strongly negatively correlated with the imdb score and number of voted users is strongly positively correlated with the imdb score, the two variables are positively correlated with each other.

2. Prediction - Regression

Our focus for this section is on building the best prediction model with the dataset we have. Instead of using the train-validate methodology, we decided to fit our models on all the data and evaluate performance across models via cross validation (10-fold CV). Using CV would yield a lower variance estimate of generalization error.

2.1 Linear Regression Model

Our baseline model is an OLS model using all the covariates. The train set RMSE is 0.7738, and the CV RMSE is 0.7870. The adjusted R-squared is 0.4053. In addition, comparatively large p-values show there are numerous covariates such as director_facebook_likes and content_rating that are not significant, suggesting they are not very relevant for our model given the other covariates.

From the residual vs fitted values plot (Appendix Figure 4), we observe that there is a general trend of curvature (higher fitted values, on average, have residuals greater in magnitude). This suggests that including higher order terms could help improve our model.

2.2 Linear Regression Model with Higher Order Terms

This model is based on our baseline model, to which we add second order terms for all continuous covariates except for facenumber_in_poster and title_year. We used the 'poly' function for orthogonal polynomials. As expected with the added terms, the model fits the train data better with higher R-squared of 0.4613 and a lower train set RMSE of 0.7346. It follows that the model has a higher variance. We also find a slight improvement in the CV RMSE which is now 0.7730.

2.3 Stepwise Selection

We saw from our baseline model that there are many insignificant variables. Here, we use forward stepwise selection to explore whether eliminating some of the candidate variables from the model yields improved performance. From the model found using AIC, we obtain a train RMSE of 0.7760, and this is very close to its CV RMSE of 0.7836. This is not surprising since stepwise selection prevents overfitting by penalizing model complexity. The CV RMSE for this model, however, is generally going to be an overly optimistic estimate of test error since the same data is being used repeatedly to make selection decisions. Using the train-validation methodology could somewhat mitigate this, but we stick with CV for objective comparisons with other models.

2.4. Interaction Terms

If particular covariates have large effect on each other, it is worth considering including interaction terms. We decided to investigate covariates that show high correlation from the ggpair plot or that depend on each other such as gross and budget.

Interaction between covariates of the following has been added:

Gross : num_critic_for_reviews, num_voted_users : num_critic_for_reviews, num_voted_users : gross, num_user_for_reviews : num_critic_for_reviews, num_user_for_reviews : gross, num_user_for_reviews : num_voted_users, budget : num_critic_for_reviews, budget : num_user_for_reviews, movie_facebook_likes : num_critic_for_reviews.

Linear regression model after adding the interaction terms resulted in a train RSME of 0.7614 and CV RMSE of 0.7838 which is slightly lower than the CV RMSE from the linear regression model. As we added new covariates, adjusted R-squared increased to 0.4213. By capturing and including relationship between covariates in the interaction terms, we were able to decrease the RSME.

2.5 Ridge Regression

The coefficients of the linear regression model is small, each ranging from the power of 10^{-1} to the power of 10^{-10}. This may mean that the covariates need to be scaled and standardized. We decided to standardize the covariates and run ridge regression. The train RSME from ridge is 0.800 and the CV RMSE is 0.62550. The train RMSE is high since the objective is no longer to minimize sum of squared errors. By adding penalties to the covariates, ridge regression may have been able to capture the multicollinearity of our dataset.

2.6 Discussion

Of the different prediction methods (See Appendix Table 2 for comparison), we believe ridge regression provides the best prediction for the response variable. With regularization, it produces more stable parameters compare to OLS models and thus should be less prone to overfitting training data (as we can see from train RSME). On the contrary, it generalizes better to new data, which is our objective for prediction. We think the test error should be larger than the CV RMSE we currently see. It is an underestimate of test error since we used the cv.glmnet() function on the training dataset to choose λ (lambda.1se) that resulted in a low CV error. Code in Appendix.

3. Prediction - Classification

We focus on binary classification where the outcome is 1 if IMDB score is greater than 7, and 0 otherwise (see Section 1.2 for more explanation). Our objective for the classification task is to choose a model with AUC closer to 1 as possible since the average 0-1 loss will not capture our prediction (ie. we have a higher number of outcomes with IMDB less than 7, which means we could achieve a low 0-1 loss by simply predicting 0 all the time). We evaluate performance on our models using test-validation methodology.

3.1 Naive Bayes Classifier

Our baseline is the Naive Bayes classifier. We measure the average 0-1 loss and the Area Under Curve (AUC) to score the classification models. Its average 0-1 loss on the training data is 0.2436 and the AUC is 0.7681. Meanwhile, the average 0-1 loss on the validation set is 0.2762 and its AUC is 0.6287.

3.2 Logistic Regression

Our second classifier is logistic regression using all covariates. For logistic regression, The 0-1 loss of the training set is 0.1731 while its AUC is 0.8577. Meanwhile, the average 0-1 loss on the validation set is 0.21690 and its AUC is 0.8180. We can see that logistic regression is a better predictor compared to the Naive Bayes Classifier as average 0-1 loss decreased and the AUC is closer to 1. Such difference in performance may be due to Naive Bayes Classifier assuming that all covariates are conditionally independent, hence introducing bias. It is hard to say that all covariates are independent in our dataset.

3.3 Random Forest

Finally, we decided to implement a Random Forest model with R's default parameters since Random Forest is extremely flexible with both categorical and continuous variables. In addition, its Ensembling Learning technique helps reduce overfitting problem in decision trees and also reduce the variance

(thereby improving the accuracy). There is zero misclassification in case of prediction on train data set. When applied to the validation set, 0-1 loss is 0.1963, and the AUC is 0.8778.

3.4 Discussion
We have a high accuracy for training data for the random forest model since its tree was built based on the training data. Of the different prediction methods (See Appendix Table 3 for comparison), we believe the random forest classifier provides the best prediction on the validation set based on AUC, our evaluation metric for model performance. Code in Appendix.

4. Prediction on the test set
See Appendix Table 4 for comparison. From Part 1 of the project, we chose ridge regression as the best model for regression. The 10-fold CV RSME error of the model was 0.6255, and we observe an error of 0.7876 when applied to the held out test set. This is as we had expected: since we used the cv.glmnet() function on the training dataset to choose λ for the model, CV error was likely an underestimate of test error.

For classification, we chose the random forest classifier from which we had an AUC of 0.8778, and we have the test error of 0.9161. It is interesting to see a higher test set AUC compared to that of the train set. This indicates that we had used a robust approach to estimate the generalization error of our model. That is, we used a validation set to measure the AUC of our random forest classifier, so we were already getting a fairly good estimate. The test set and validation set AUC are close, and a slightly higher AUC estimate on test set may be possible.

5. Inference
In this part of the project, we plan to try out some of the methods we learned in the class for inference. For simplicity of analysis, we use the baseline linear regression model from Part 1 resulting from the elimination of very correlated features.

5.1 Significance From the output, we see that a number of coefficients are significant at different levels. We decided to choose to $\alpha = 0.05$ to report statistical significance. The following variables are statistically significant at the 95% level:
content_ratingPG-13, actor_1_facebook_likes, actor_2_facebook_likes, num_critic_for_reviews, duration, director_facebook_likes, num_voted_users, num_user_for_reviews, budget, title_year

These represent when p-values are less than 0.05, where a p-value for each test gives the probability of observing a test statistic as extreme as the observed, respective t-value, if the null hypothesis is true (when coefficient is zero). The results are consistent with our analysis and intuition stated in Part 1 of the project. However, the mathematics used to compute the p-value involves various distributional assumptions that we don't believe to hold in our data. Notably, t-test depends on the assumptions that the population model is linear, includes all of the right covariates, and has independently and identically distributed errors with mean zero and a constant variance. In addition, the fact that the coefficient is not statistically significant doesn't necessarily mean it is not important to the model. Rather, it could mean that we simply do not have enough evidence to reject the null hypothesis.

5.2 Fitting a model Fitting our chosen model on the test data helps to validate our findings of significance levels. Our interpretations of p-values and significance become more relevant as the model selection is no longer based on the data we fit it on. When the model is fitted on the test data, not all coefficients have the same statistical significance compared to when the model is fitted on the train data (See Appendix Figure 5 and 6).

Covariates with coefficients of lower statistical significance:
actor_1_facebook_likes, cast_total_facebook_likes, budget, actor_2_facebook_likes

Covariates with coefficients of higher statistical significance:
content_ratingPG, movie_facebook_likes

This change is attributed to the fact that our data was random, and also that we had a finite amount of data (less evidence to determine whether a coefficient is zero). Slight changes in p-values were therefore as expected. For instance, we observe that our covariate actor_2_facebook_likes is significant at the 86% level so we now fail to reject the null hypothesis. Previously, our training data suggested that this covariate is significant at the 99% level and therefore relevant to include it in our model.

This is also an example of a post-selection inference problem where we ended up choosing the covariate favorably from the training set because it appeared to be significant. By doing so, we favorably biased our selection of p-values. In addition, the changes in coefficients can also be explained by the problem of collinearity. For most of our covariates, coefficients did not fluctuate too much from the results we had seen before which gives support to our model. However, there are some covariates with different magnitudes of coefficients. This can be partly explained by problems like collinearity where the effect of a covariate is not solely determined by the covariate itself but also incorporates its correlation with other covariates.

5.3 Bootstrap As another way to test the statistical significance of our coefficients, we can use the bootstrap (case resampling) to get an estimate of the sampling distribution of the estimates of the coefficients of the linear regression. We used the technique to calculate 10,000 samples, from which the bias and standard error of coefficients were calculated. Using this information, we can build a 95% confidence interval for each coefficient: $[\widehat{B_i} - 1.96 * \widehat{SE}\ ,\ \widehat{B_i} + 1.96 * \widehat{SE}]$ (See Appendix Table 5). This assumes that the sampling distribution is normal. To compare our results from what R gave us in its standard regression output, we look at the standard errors. Generally, R generated and bootstrap standard errors reside in the same neighborhood for every coefficient . There are slight differences, and in particular, bootstrap standard errors are on average marginally higher. We have used sufficient data points (10,000 i.i.d samples) so the sample size does not account for such differences. Differences might occur because the assumption on the normality of the distribution of the estimate of the linear regression coefficients does not hold. If the normality assumption indeed failed to hold, the estimates from the bootstrap method would be more robust and accurate since no assumptions are necessary for the bootstrap.

5.4 Include all Covariates Our chosen model does not include all covariates we had available. Therefore, we run an inference analysis on the training data with a model including most of the covariates we excluded and see if there is any difference in coefficients and their significance.

The following are the coefficients that were statistically significant at the 99.9% level when we did not include all covariates for our final model: Num_critic_for_reviews, duration, director_facebook_likes, num_voted_users, num_user_for_reviews, budget, title_year

The following are the coefficients that are statistically significant at the 99.9% level when we included all covariates: Num_critic_for_reviews, duration, actor_1_facebook_likes, director_facebook_likes, num_voted_users, cast_total_facebook_likes, num_user_for_reviews, budget, title_year, actor_2_facebook_likes

Interestingly, most of the coefficients that are statistically significant at the 99.9% level are similar with those from our final model where we chose certain covariates. We can also view that the p-values for several covariates such as actor_1_facebook likes and actor_2_facebook_likes changed so that they are included within the 99.9% statistically significant coefficients. This may be explained by the collinearity between certain covariates. As we include all the covariates, collinearity amongst certain coefficients that was absent in our final model may have been introduced, thus resulting in some additional significant covariates.

5.5 Potential Problems There is an inherent risk in running simultaneous, multiple hypothesis tests on coefficients. By some random chance, coefficients can appear to be large even though there isn't a clear underlying relationship. This is where we observe false positives. In particular, because we used a cut-off of 0.05, we should expect 5% of our discoveries to be false positives. However, given that we started out with 16 coefficients (0.05 x 16 = 0.8), it is unlikely that at least one of the coefficients that we found to be significant is a false positive. The Bonferroni correction would alter our effective p-value to about 0.003. Many of the coefficients that were previous significant at 95% level would no longer be significant.

Then, we mentioned the problem of collinearity with our analysis in part (c). If we see different magnitudes of coefficients when the model is fitted on the test data vs. train data, this could be a flag for collinearity. To provide a concrete example, we saw that the magnitude of the coefficient was different for actor_2_facebook_likes when the model was fitted on the test data. Actor_2_facebook_likes is probably correlated with cast_total_facebook_likes, which made it difficult to determine whether the magnitude of the actor_2_facebook_likes coefficient we saw from the train data was directly due to the covariate itself or its indirect effect from its correlation with cast_total_facebook_likes. We therefore see different magnitudes of coefficients when the model is fitted on test data.

In part (b), we also briefly addressed the problem of post-selection. We had chosen our covariates and model based on the training set, and therefore optimized their significance, favoriably biasing our selection of p-values. We consequently observed changes in significance levels when the model was fitted on the test set.

5.6 Causal Relationship We would not be willing to interpret the relationships we found that are significant as causal relationships. As discussed in part (e) above, when there exists collinearity or a high correlation between two predictors, we cannot derive causal relationships. A high correlation between the predictors makes it difficult to determine the sole impact of each predictor on the response variable. In addition, there could also be other confounding variables that are not present among the covariates, and these could also be affecting our covariates. For instance, we had excluded the covariate title_year due to missing values. This covariate would influence other covariates like actor_facebook_likes (since the popularity of actors tend to change over time) and num_user_for_reviews (ie. there are probably more reviews for more recent movies vs. 1970's when the internet was not readily available). Taking all this into account, we are not convinced of causal relationships of our significant coefficients on a macro level.

6. Discussion

Our models could be used to predict factors that contribute to high IMDB scores. They should primarily be used for prediction but not inference as we do not know whether the models reflect the population model itself. It is not recommended to make causal inference as we cannot reason about other significant covariates that the dataset may not have included (as further explained in part (f) of Section 2), introducing omitted variable bias. Correlation amongst covariates and/or outcome that may be useful for prediction cannot necessarily explain inference and causation.

Our models will guide movie directors, investors, and other related personnel to forecast a movie's IMDB score so they could predict and succeed in promoting their movies. However, possible pitfall could result from expecting causal relationship between the covariates and the observation. For instance, users should be aware that casting actor 1 with the higher facebook likes does not necessarily mean higher IMDB score. Hence as discussed previously, users should be aware of the pitfall that the dataset may consist due to collinearity and omitted variable bias.

During the data cleaning process, we decided to explore movies that were released after 1990. This was due to the lack of data of movies released before 1990 and our aim to best reflect the advancement in technology. First, it is critical to note that evolution of technology (ie. new social network) may completely change considered covariates (ie. movie_facebook_likes if Facebook's usage popularity declines over time), and in this case, new data collection and model development should be done. In addition, the popularity of movies depends on a number of factors, especially because the trend in the movie industry and people's preferences are always changing. To account for changes in technology and people's preferences over time, we may need to refit our model at least every few years. It is inherently difficult to determine exactly how often they should be refitted because it depends on a particular context.

There are few things we want to point out regarding the choices we made in our data analysis. First, we would explain our data preprocessing step as well as data transformation, and clearly elaborate on why we decided to retain certain features for the final model. We would also warn how some decisions we made when manually choosing our covariates were based on our intuition and domain knowledge.

Second, we would like to point out that we had to delete columns that contained names of actors and directors as the names in the dataset were too diverse. Instead, we relied on their presence on social media as a gauge. Our client should be aware that such method may misrepresent how the actors and directors affect the response variable.

We would notify of our concerns discussed in part (e) of Section 2, especially how our model might suffer from problems like post-selection inference. We would also point out issues with multiple hypothesis testing that even though we have 16 covariates and use a cut-off of 5%, it is possible to observe one or two of our covariates to appear significant by some random chance.

While the data seem reasonably extensive and reliable, other reasonable covariates could have been collected to reflect the population model. As discussed in part (b), we would want to collect data from recently developed and most used social network. Thus we would consider collecting data that extends beyond number of facebook likes, such as number of instagram followers or number of twitter followers. Also, it would be helpful to know what kind of technology the movie incorporates such as IMAX, 3D, and so on in addition to the color and aspect_ratio covariates.

If we were to attack the same dataset again, we would try different techniques such as the k-nn classifiers for both classification and regression. Also, rather than deleting rows of omitted values, we would want to try imputing the values so more data can be reflected when building our models.

Appendix

color	Movie colorization. “Color” or “Black and White”	**country**	Produced country
director_name	Name of director	**gross**	Gross earnings in dollars
num_critic_for_reviews	Number of critics’ reviews on imdb	**genres**	Genre of movie
duration	Duration of movie	**budget**	Budget in dollars
director_facebook_likes	Number of likes of director on Facebook	**movie_title**	Title of the movie
actor_1_facebook_likes	Number of likes of actor 1 on Facebook	**num_voted_users**	Number of people who voted for the movie
actor_2_facebook_likes	Number of likes of actor 2 on Facebook	**cast_total_facebook_likes**	Total number of likes of entire cast on Facebook
actor_3_facebook_likes	Number of likes of actor 3 on Facebook	**facenumber_in_poster**	Number of actors’ faces in poster
actor_1_name	Name of actor 1	**movie_imdb_link**	Imdb link to the movie
actor_2_name	Name of actor 2	**language**	Language
actor_3_name	Name of actor 3	**content_rating**	Content rating
plot_keywords	Keywords of movie plot	**title_year**	Released year
num_user_for_reviews	Number of users who wrote a review	**imdb_score**	imdb score of movie
aspect_ratio	Movie aspect ratio	**movie_facebook_likes**	Number of likes of movie on Facebook

Table 1. Variable Description (https://data.world/popculture/imdb-5000-movie-dataset)

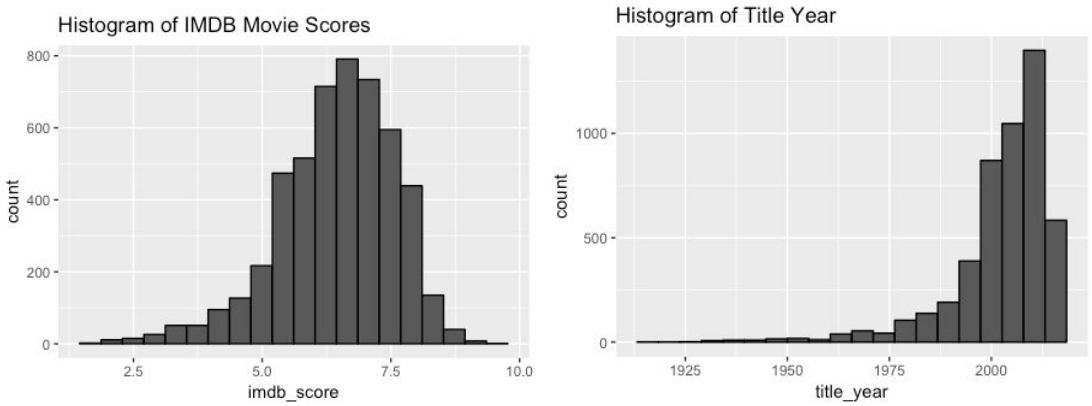

Figure 1. Histogram of IMDB movie scores, Histogram of title year

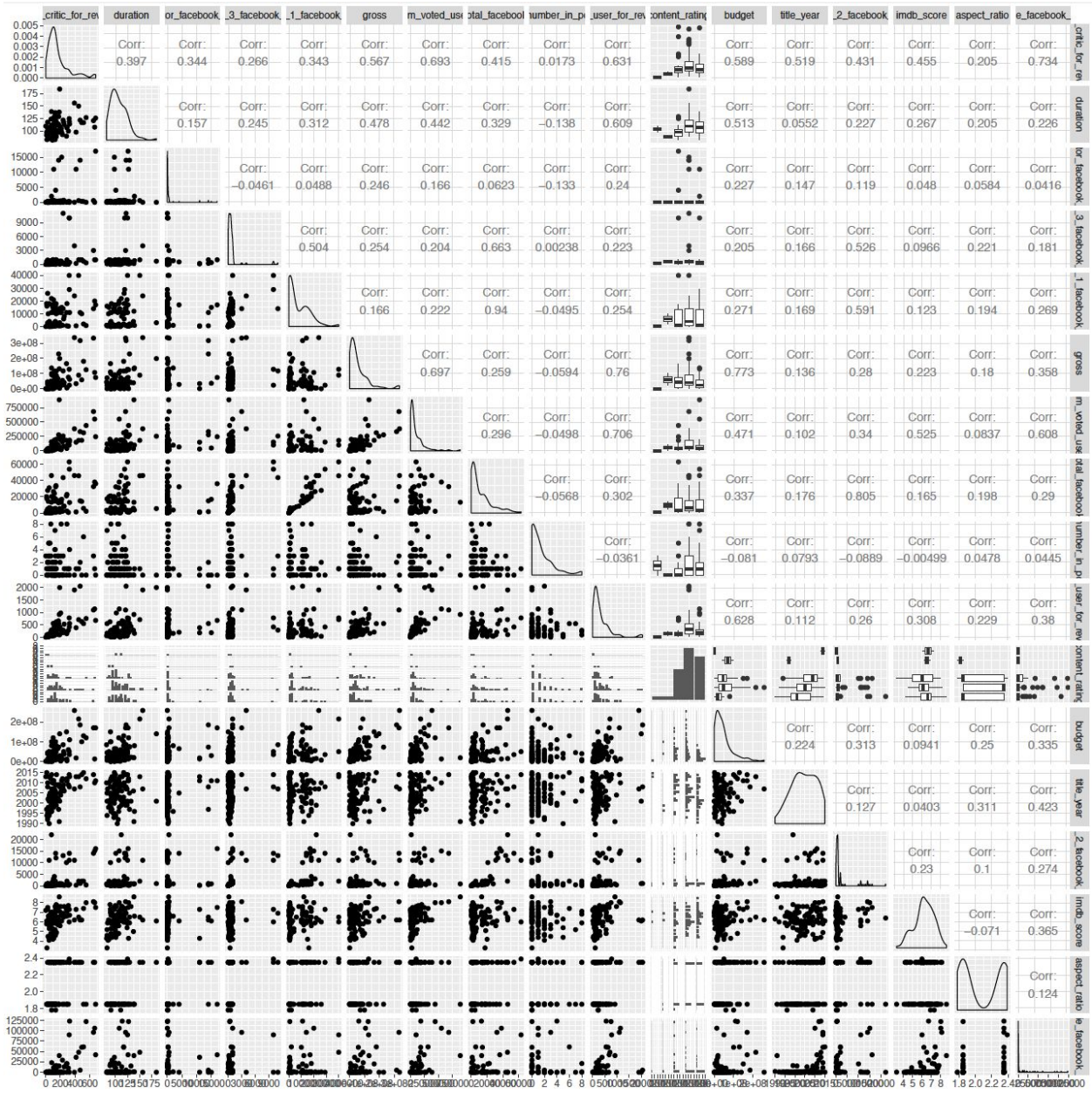

Figure 2. Ggplot of Covariates

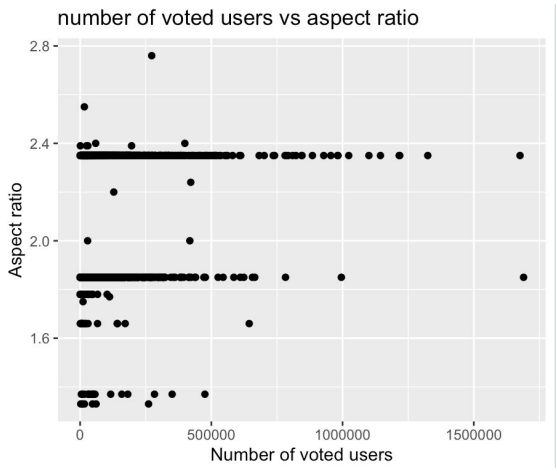

Figure 3. Number of voted users vs. Aspect ratio graph

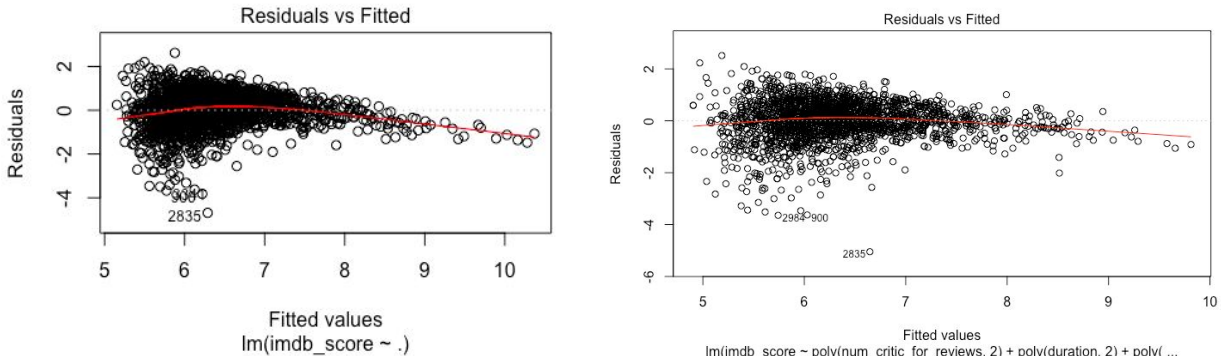

Figure 4. Residuals vs. Fitted of
1)Linear regression Model and 2) Linear regression Model with higher order term

	Train RMSE	10-fold CV RMSE
Linear Regression Model	0.7738	0.7870
Higher Order Terms	0.7346	0.7730
Stepwise Selection	0.7760	0.7836
Interaction Terms	0.7614	0.7838
Ridge Regression	0.8001	0.6255

Table 2. Comparison of Train RMSE and CV RMSE of Regression

	Training set		Validation set	
	0-1 Loss	AUC	0-1 Loss	AUC
Naive Bayes Classifier	0.2436	0.7681	0.2762	0.6287
Logistic Regression	0.1731	0.8577	0.2169	0.8180
Random Forest	-	-	0.1963	0.8778

Table 3. 0-1 Loss and AUC values of Classifiers

Regression - Ridge Regression code

```
#standardizing train dataset

sd_train = train
sd_train <- sapply(sd_train, function(x) if(is.numeric(x)){
  scale(x, center = TRUE, scale = TRUE)
} else x)
sd_train = as.data.frame(sd_train)

x <- model.matrix(imdb_score~., sd_train)[,-15] #trim off score column
y <- sd_train %>% dplyr::select(imdb_score) %>% unlist() %>% as.numeric()

#ridge model
ridge <- cv.glmnet(x,y, standardize = TRUE, alpha = 1)
ridge$lambda.min
ridge$lambda.1se

i <- which(ridge$lambda == ridge$lambda.1se)
mse.1se<- ridge$cvm[i]
mse.1se


plot(ridge)
cvridge <- cv.glmnet(x,y)

plot(cvridge)
#CV error
ridge_cv <- cvridge$lambda.min # best cross-validated lambda
```

```
response2 <- predict(ridge, x)
#train RMSE
sqrt(mean((response2 - sd_train$imdb_score)^2))
```

Classification - Random Forest Classifier code

```
library(randomForest)
set.seed(123)
rf <- randomForest(factor(imdb_score) ~ . , data = train, mtry = 5)
rf

#train 0-1 loss is zero
predTrain <- predict(rf, train, type = "class")
table(predTrain, train$imdb_score)
trainLoss = sum(predTrain != train$imdb_score)
table(predTrain, train$imdb_score)

#validate 0-1 loss
predValidate<- predict(rf, validate, type = "class")
table(predValidate, validate$imdb_score)
valLoss = sum(predValidate != validate$imdb_score)
avgValLoss = valLoss/length(validate$imdb_score)
avgValLoss

#validate AUC error
rf_val <- predict(rf, validate, type="prob")[,2]
rf_val <- prediction(rf_val, validate$imdb_score)
auc_train2 <- performance(rf_val, measure = "auc")@y.values[[1]]
auc_train2
```

		Train Set Error	Test Set Error
Ridge Regression	RSME	0.6255 (10-fold CV)	0.7876
Random Forest Classifier	AUC	0.8778	0.9161

Table 4. Comparison of test error

```
Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                6.460e+01  7.346e+00   8.795  < 2e-16 ***
num_critic_for_reviews     3.023e-03  2.738e-04  11.039  < 2e-16 ***
duration                   1.370e-02  1.004e-03  13.652  < 2e-16 ***
director_facebook_likes    3.274e-07  5.631e-06   0.058  0.95363
actor_3_facebook_likes     3.583e-05  2.379e-05   1.506  0.13216
actor_1_facebook_likes     4.869e-05  1.533e-05   3.176  0.00151 **
gross                      2.692e-10  3.836e-10   0.702  0.48288
num_voted_users            3.368e-06  2.138e-07  15.748  < 2e-16 ***
cast_total_facebook_likes -4.612e-05  1.530e-05  -3.014  0.00261 **
facenumber_in_poster      -1.517e-02  7.794e-03  -1.946  0.05179 .
num_user_for_reviews      -6.017e-04  6.900e-05  -8.720  < 2e-16 ***
content_ratingG           -4.885e-01  3.400e-01  -1.437  0.15088
content_ratingNC-17       -1.030e-02  5.505e-01  -0.019  0.98507
content_ratingPG          -4.790e-01  3.217e-01  -1.489  0.13665
content_ratingPG-13       -7.818e-01  3.199e-01  -2.444  0.01460 *
content_ratingR           -5.720e-01  3.196e-01  -1.790  0.07360 .
budget                    -5.758e-09  5.649e-10 -10.192  < 2e-16 ***
title_year                -2.958e-02  3.669e-03  -8.064 1.21e-15 ***
actor_2_facebook_likes     5.127e-05  1.609e-05   3.187  0.00146 **
aspect_ratio              -7.769e-02  6.864e-02  -1.132  0.25783
movie_facebook_likes      -9.456e-07  1.183e-06  -0.800  0.42400
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7776 on 2167 degrees of freedom
Multiple R-squared:  0.4108,    Adjusted R-squared:  0.4053
F-statistic: 75.53 on 20 and 2167 DF,  p-value: < 2.2e-16
```

Figure 5. Coefficients of the OLS model on Train Set

```
Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                6.452e+01  1.605e+01   4.020 6.67e-05 ***
num_critic_for_reviews     3.956e-03  5.472e-04   7.230 1.71e-12 ***
duration                   1.373e-02  1.769e-03   7.758 4.49e-14 ***
director_facebook_likes    1.535e-06  1.177e-05   0.130 0.896282
actor_3_facebook_likes     7.372e-05  4.905e-05   1.503 0.133489
actor_1_facebook_likes     5.428e-05  2.731e-05   1.988 0.047344 *
gross                      1.061e-09  8.750e-10   1.212 0.225941
num_voted_users            2.572e-06  4.410e-07   5.832 9.56e-09 ***
cast_total_facebook_likes -5.285e-05  2.709e-05  -1.951 0.051557 .
facenumber_in_poster      -2.131e-02  1.798e-02  -1.185 0.236414
num_user_for_reviews      -5.527e-04  1.505e-04  -3.672 0.000265 ***
content_ratingG           -3.156e-01  6.488e-01  -0.486 0.626855
content_ratingPG          -1.219e+00  6.136e-01  -1.987 0.047411 *
content_ratingPG-13       -1.352e+00  6.051e-01  -2.234 0.025906 *
content_ratingR           -1.073e+00  6.033e-01  -1.779 0.075770 .
budget                    -3.142e-09  1.302e-09  -2.414 0.016131 *
title_year                -2.971e-02  8.055e-03  -3.688 0.000249 ***
actor_2_facebook_likes     4.396e-05  2.956e-05   1.487 0.137512
aspect_ratio               1.953e-01  1.538e-01   1.270 0.204688
movie_facebook_likes      -5.191e-06  2.076e-06  -2.500 0.012725 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8378 on 528 degrees of freedom
Multiple R-squared:  0.4713,    Adjusted R-squared:  0.4522
F-statistic: 24.77 on 19 and 528 DF,  p-value: < 2.2e-16
```

Figure 6. Coefficients of the OLS model on Test Set

Covariates	$\widehat{SE}$	**CI at 95%**
(Intercept)	7.939877e+00	[4.92E+01, 8.03E+01]
num_critic_for_reviews	2.780832e-04	[2.71E-03, 3.80E-03]

duration	9.904000e-04	[1.04E-02, 1.43E-02]
director_facebook_likes	4.054885e-06	[-8.37E-06, 7.52E-06]
actor_3_facebook_likes	2.481538e-05	[1.57E-05, 1.13E-04]
actor_1_facebook_likes	1.578384e-05	[2.94E-05, 9.13E-05]
gross	3.781366e-10	[-4.42E-11, 1.44E-09]
num_voted_users	2.348367e-07	[2.77E-06, 3.69E-06]
cast_total_facebook_likes	1.565947e-05	[-8.90E-05, -2.76E-05]
facenumber_in_poster	8.967194e-03	[-4.48E-02, -9.61E-03]
num_user_for_reviews	7.873581e-05	[-8.11E-04, -5.02E-04]
budget	5.638664e-10	[-6.34E-09, -4.13E-09]
title_year	3.977074e-03	[-3.77E-02, -2.21E-02]
actor_2_facebook_likes	1.701285e-05	[2.73E-05, 9.40E-05]
movie_facebook_likes	1.232700e-06	[-4.42E-06, 4.16E-07]

Table 5. $\widehat{SE}$ **and Confidence Interval from Bootstrap**

Code:

```
#ridge model
ridge <- cv.glmnet(x,y, standardize = TRUE, alpha = 1)
ridge$lambda.min
ridge$lambda.1se

i <- which(ridge$lambda == ridge$lambda.1se)
mse.1se<- ridge$cvm[i]
mse.1se

plot(ridge)
cvridge <- cv.glmnet(x,y)

plot(cvridge)
#CV error
ridge_cv <- cvridge$lambda.min # best cross-validated lambda
response2 <- predict(ridge, x)
#train RMSE
sqrt(mean((response2 - sd_train$imdb_score)^2))
```

```
sd_test = hold_out_test
sd_test <- sapply(sd_train, function(x) if(is.numeric(x)){
  scale(x, center = TRUE, scale = TRUE)
} else x)
#Leaves out factor variable I don't know why it returns numeric variable for content tho
sd_test = as.data.frame(sd_test)
set.seed(123)

x <- model.matrix(imdb_score~., sd_test)[,-15] #trim off score column
y <- sd_train %>% dplyr::select(imdb_score) %>% unlist() %>% as.numeric()

response3 <- predict(ridge, x)
sqrt(mean((response3 - sd_test$imdb_score)^2))

set.seed(123)
model = lm(imdb_score ~., data = train)
model.cv = cvFit (model, data = train, y = train$imdb_score, K = 10, na.rm=TRUE)
print(summary(model))
sqrt(mean((predict(model, train) - train$imdb_score)^2))
print(model.cv)
sqrt(mean((predict(model, test) - test$imdb_score)^2))

library(boot)
set.seed(123)

# bootstrapping with 10000 replications
results <- boot(data = train, statistic=bs,R=10000, formula=imdb_score ~.)
ci <- boot.ci(results, conf = 0.95)
```