MS&E226 Project: Exploration of Movie Revenues and Ratings, Part 1
Nicole Nie, Jeremy Bao

Project Topic

Are you a loyal IMDB user? Do you follow any movie celebrity on Facebook or Twitter? In this project, we look into both conventional and modern aspects of movies. Part 1 concerns dataset description and interesting pattern discovery. Please find related tables and graphs Appendix.

Dataset description

We start with a dataset provided by from Kaggle. It contains 28 variables for 5043 movies, spanning across 100 years in 66 countries (Table 1). These include basic facts (title, year, duration, color, country, language), director and cast and their Facebook likes, content and type (genres, keywords, content rating), financial performance (budget and box office gross), and feedback statistics (movie Facebook likes, number of critic reviews and voted users, and IMDB scores). Combinations of these many dimensions allow us to explore interesting findings. Since there are no clear patterns in NA values, we simply remove them. Afterwards, the training set still has over 3000 observations.

Before exploring exiting insights, here are some facts about the dataset (Graph 1):
1) Most movies in this dataset are within time span from 1980 to 2016, with a majority produced after 2000.
2) Average movie duration is 110 min, with standard deviation of 22.
3) IMDB scores average around 6.5, with standard deviation of 1.1.
4) Content ratings mostly fall into PG-13, R and PG.
5) Documentary, comedy, and thriller are the top three genres in this dataset.
6) The categories of country and language are distributed quite unevenly. We suggest splitting movies into US / non-US and English / non-English.

Response Variables

The correlation matrix (Graph 2.1) indicates that IMDB score or gross revenues could play as continuous response variable. It is also common sense that the two indicators reflect a movie's overall market performance. We could further create a binary response variable "Profit" by subtracting budget from gross (loss – 0, profit – 1), since the movies with profit and loss are nearly half and half in this dataset. Plus, it is an interesting topic for prediction.

Covariates Correlations

1) Numeric variables with response variables (Graph 2.1)
We sort covariates with respect to coefficient of correlation (r) in descending order (Table 2).
(a) No numeric variable has strong correlation with IMDB score, while gross have quite strong correlations with budget ($r = 0.68$) and number of user reviews ($r = 0.52$).
(b) Correlated variables that both response variables share in common are number of user reviews, number of critic reviews, and duration.
(c) A clear distinction is that IMDB score correlates much less with budget than gross does.
In both cases, title year plays little role, but further examination of their relationships indicates it would be better to categorize it into 4 bins: Before 1960s (Hollywood Golden Age), 1960-1980s (New Hollywood), 1980 – 2000s (Contemporary), and After 2000 (rise of social network).

2) Irrelevant variables (Graph 2.2)
The variable "number of face in poster" seems attractive: Comedies may use a lot of faces in poster, while Horrors almost always use one (THE one!). On the contrary, movie of any genre is possible to have one face in poster. Face number is quite an irrelevant noise and its interactions with genres, country, language, and content ratings are still little relevant. Other irrelevant variables are string variables such as movie title and people's names.

3) Interaction terms
Although aspect ratio itself seems to be irrelevant, it might interact with certain genres such as War / History which may involve spectacular scenes (Graph 2.3). Similarly, content ratings itself is not quite relevant but might somehow measure the movie's impacts in genres such as Crime (Graph 2.4).

4) Associate correlations
Such correlations exist in actors' Facebook likes (Graph 2.5): actor 1 ~ actor 2 ~ cast total, and actor 2 ~ actor 3 ~ cast total. In on-going analysis, it may be better to use one of them (e.g. cast total) as the proxy for actor popularity to avoid collinearity problems.

Interesting Insights
We found two interesting patterns that may somehow reflect people's tastes.
1) Average IMDB score actually goes down as time goes by. Are reviewers becoming more critical, or are movies worse off? (Graph 3.1).
2) R rating movies receive higher average score than PG or PG-13. People do fancy stimulation and thrills.

We also find some answers to the following questions:
Do directors shooting more movies actually have better performance? What about the actors/actresses? (Graph 3.2 and 3.3).
While top productive directors list is something, average IMDB scores and Return On Investment (ROI) of their movies sometimes tell a different story. Though some highly productive directors achieved high IMDB scores and high ROI (e.g. Steven Spielberg), others directors did not (e.g. Renny Harlin). Also, some directors do well only on ROI or only on scores. On the contrary, directors such as George Lucas did not shoot so many movies, but many of his work are most profitable. Similar patterns could be found in top actors.

Concerns of Data Quality
1) Survival bias. We use Facebook likes as a proxy for popularity. However, Facebook was not founded until recently, and passed-away directors and their movies may not have Facebook accounts. Such observations have value zero in the dataset.
2) Every movie has quite distinct key words, but are less helpful than user comments (not in current dataset) in sentiment analysis. We plan to put it aside on the risk of loosing a part of information.
3) Overall, this dataset contains around 5,000 movies whereas existed movies exceed 40,000. We do not have enough information on whether the provider's data collection involved any bias.

Appendix

Table 1 Variables overview

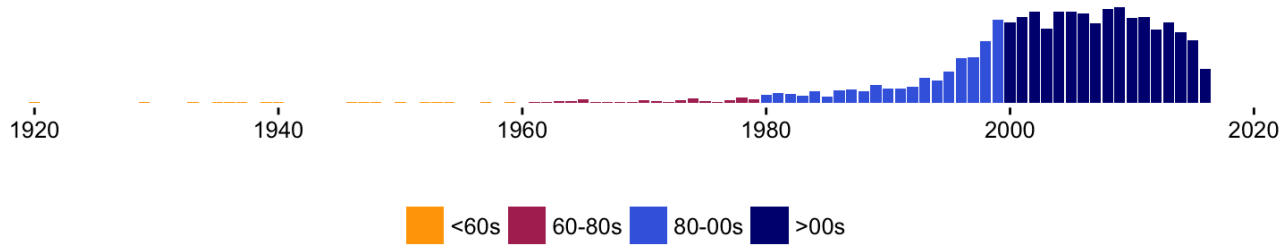
Variable(s)	Type	Example
movie_title	string	Avatar
title_year	integer	2009
color	string, categorical	Black, White
duration	integer (minute)	12
country	string, categorical	USA
language	string, categorical	English
director_name, actor_1_name, actor_2_name, actor_3_name	string	Anthony Russo
director_facebook_likes, actor_1_facebook_likes, actor_2_facebook_likes, actor_3_facebook_likes, cast_total_facebook_likes, movie_facebook_likes	integer	681
21 genre variables: Action, Adventure, Crime …	bool, categorical	TRUE, FALSE
plot_keywords	string	blood|godzilla|monster|sequel
facenumber_in_poster	integer, categorical	0
aspect_ratio	real number, categorical	2.35
num_voted_users, num_user_for_reviews, num_critic_reviews	integer	371639
content_rating	string, categorical	PG-13
budget, gross	integer	250000000
imdb_score	real number	6.9

Table 2 Correlation Coefficients

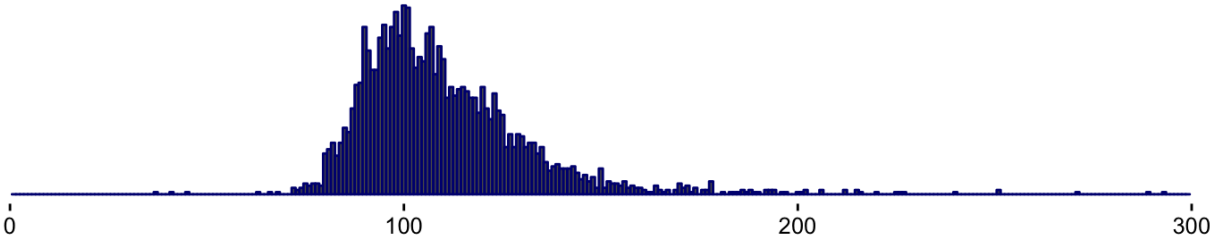
Response variable	0.5 < |r| < 1 Strong	0.3 < |r| < 0.5 Moderate	0.1 < |r| < 0.3 Weak	|r| < 0.1 Very weak
IMDB score	-	duration, number of user reviews, number of critic reviews, movie FB likes, director FB likes	gross, cast total FB likes	title year, budget
gross	budget, number of user reviews	number of critic reviews, cast total FB likes, duration	movie FB likes, IMDB score, director FB likes	title year

Graph 1: Movie Distributions

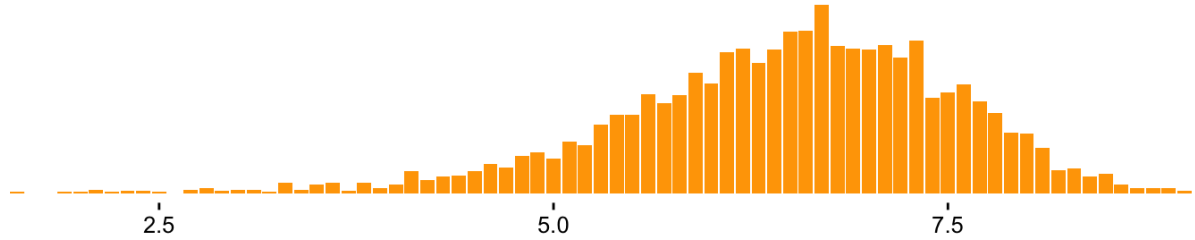
Graph 1.1 movie amount



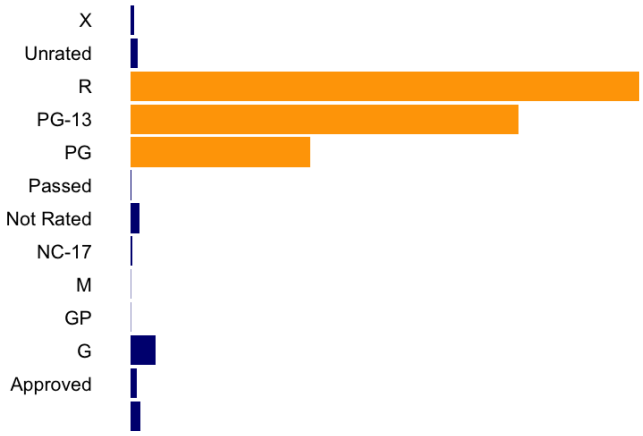
Graph 1.2: movie duration



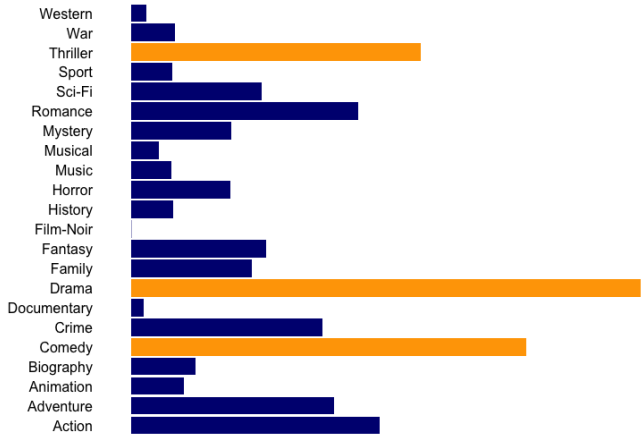
Graph 1.3: IMDB scores



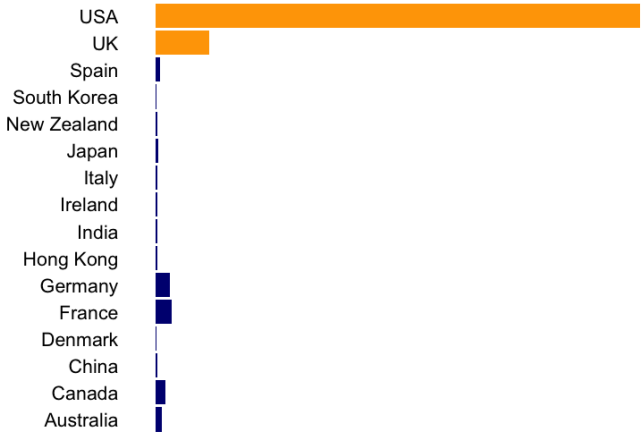
Graph 1.4: content ratings



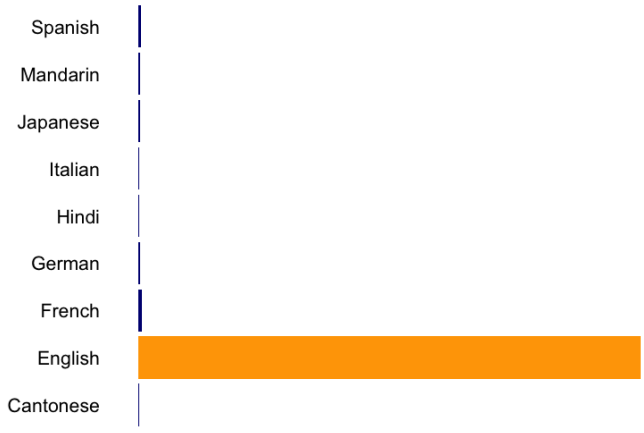
Graph 1.5: genres



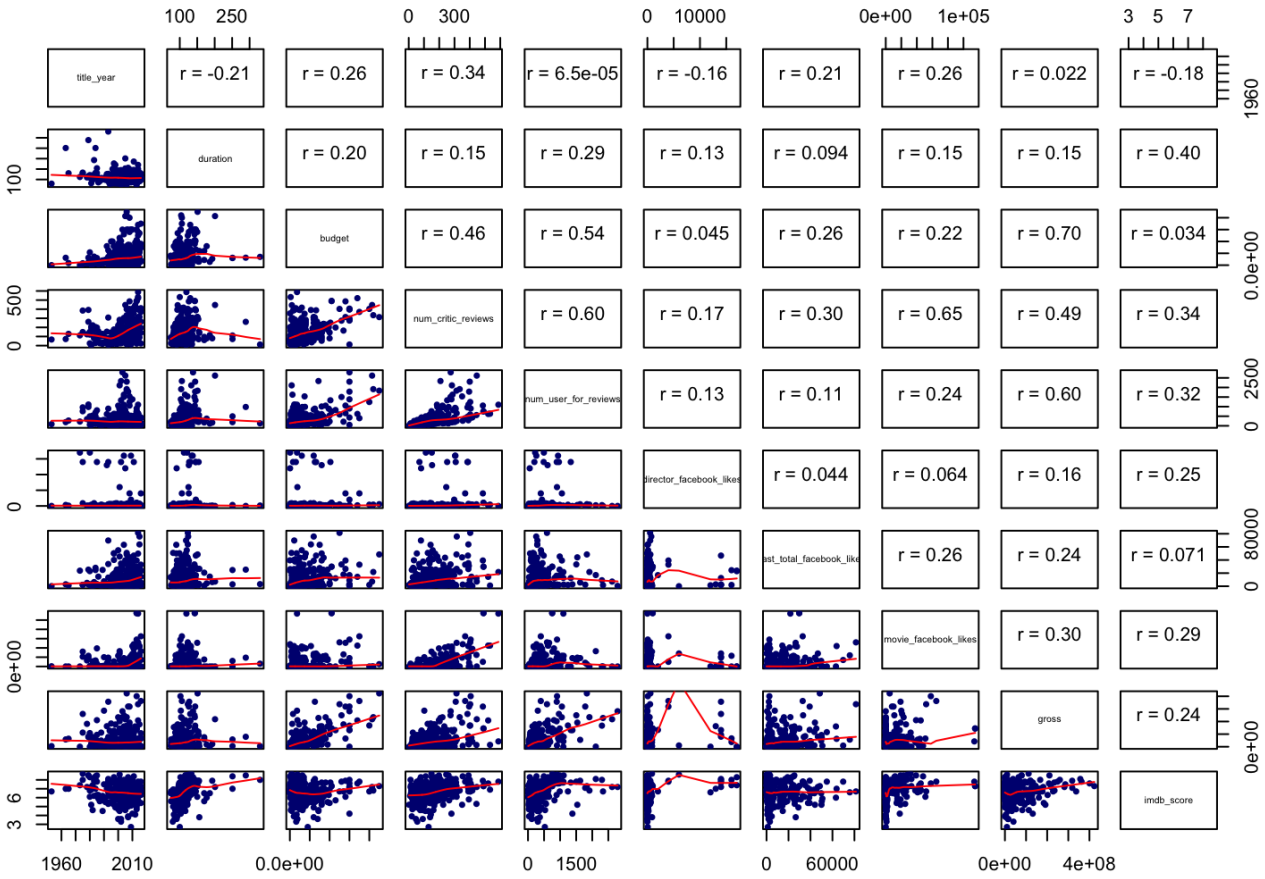
Graph 1.6: countries



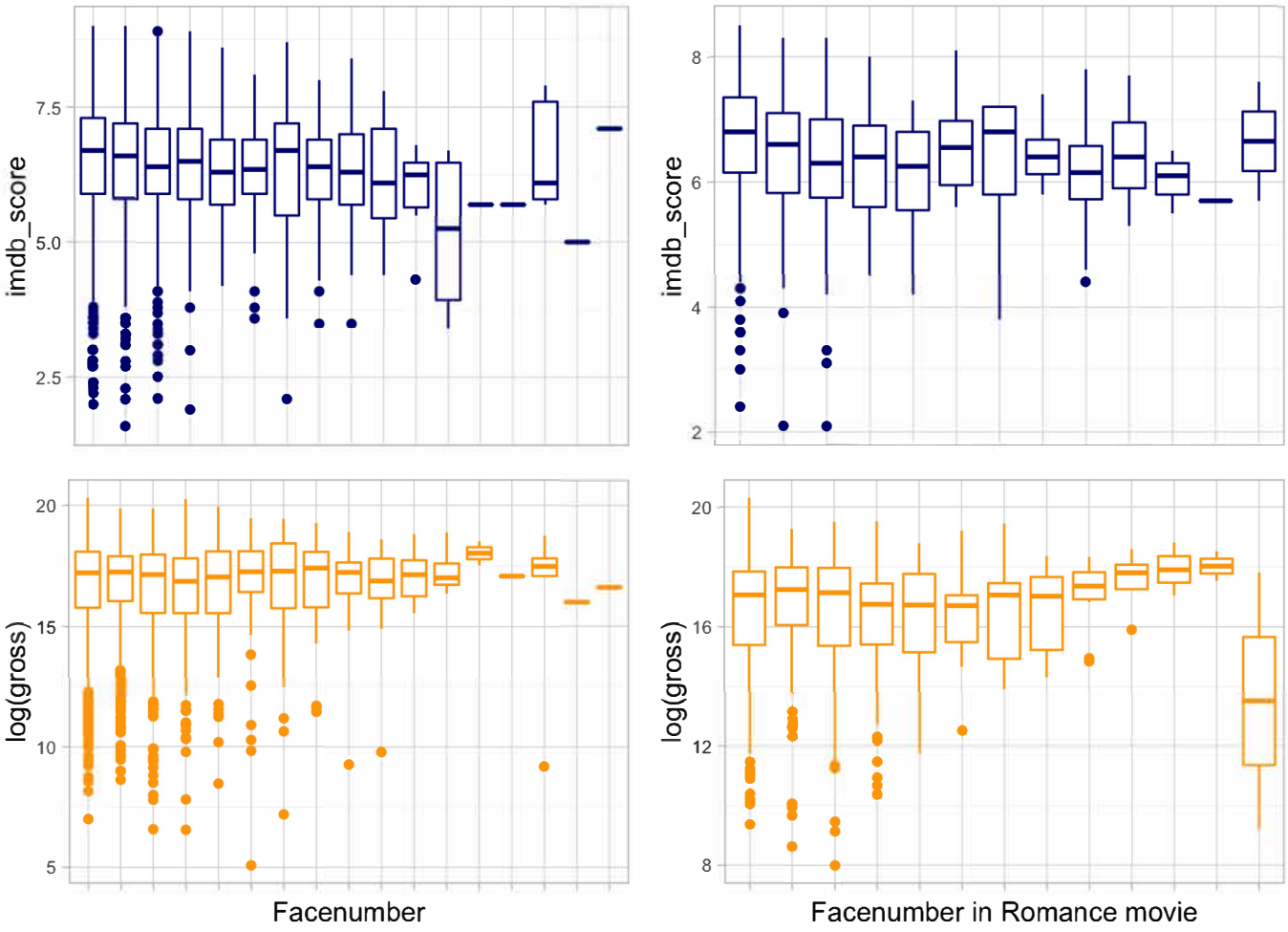
Graph 1.7: languages



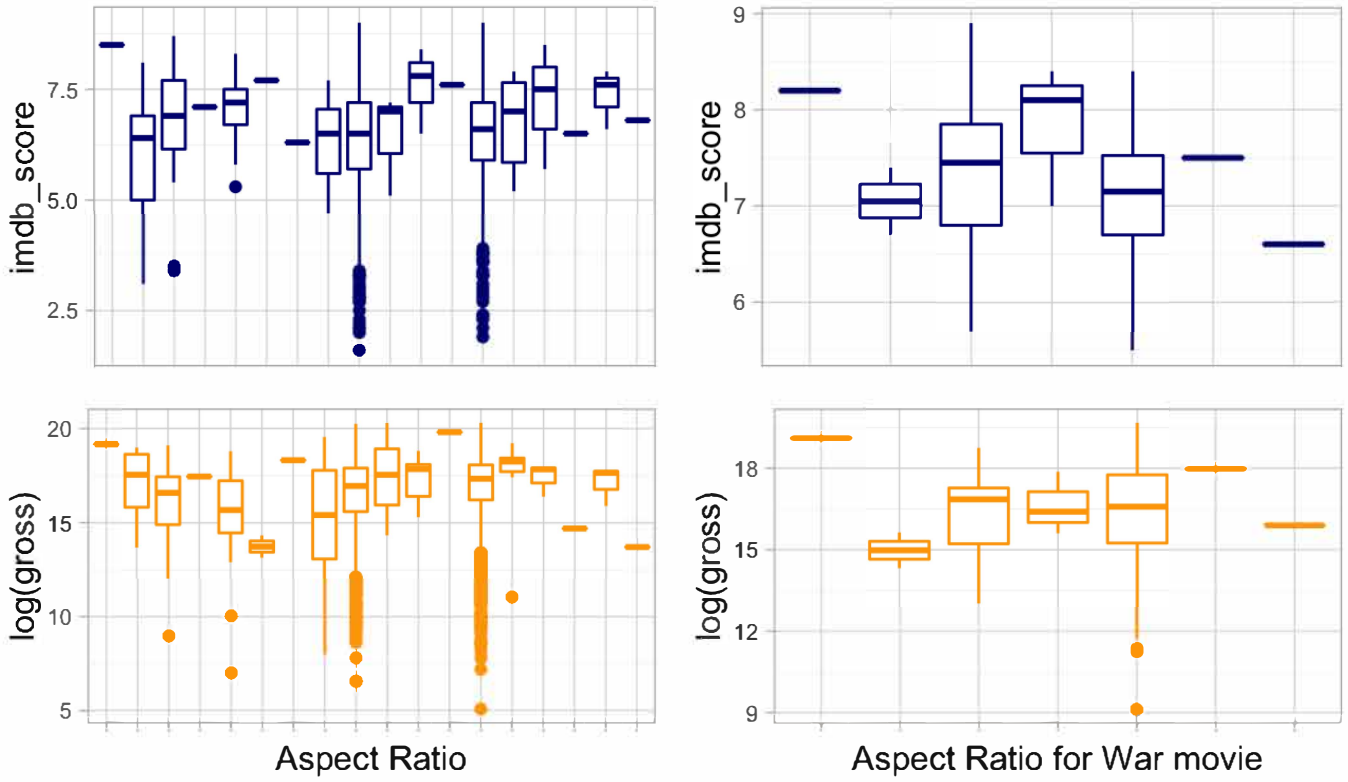
Graph 2.1: Numeric Variables Scatterplot Matrix



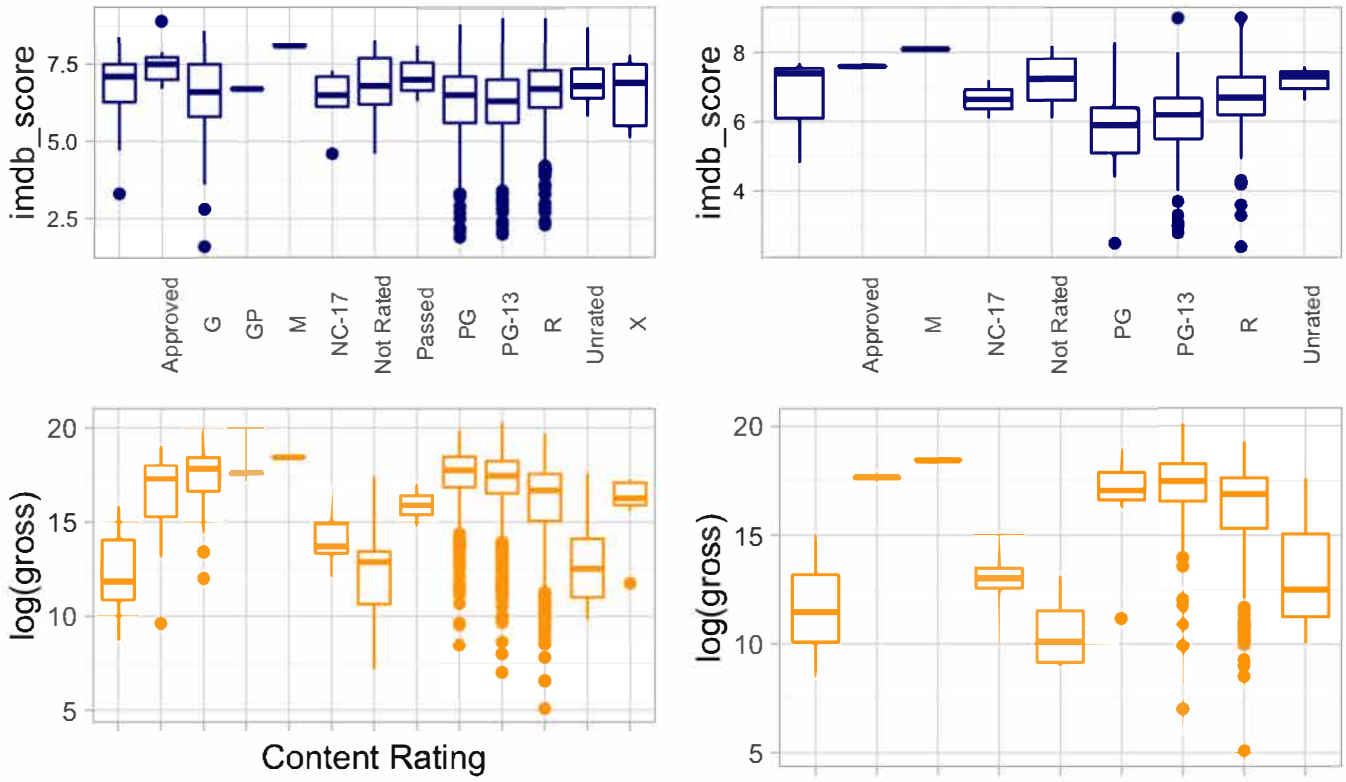
Graph 2.2: Facenumber may be irrelevant



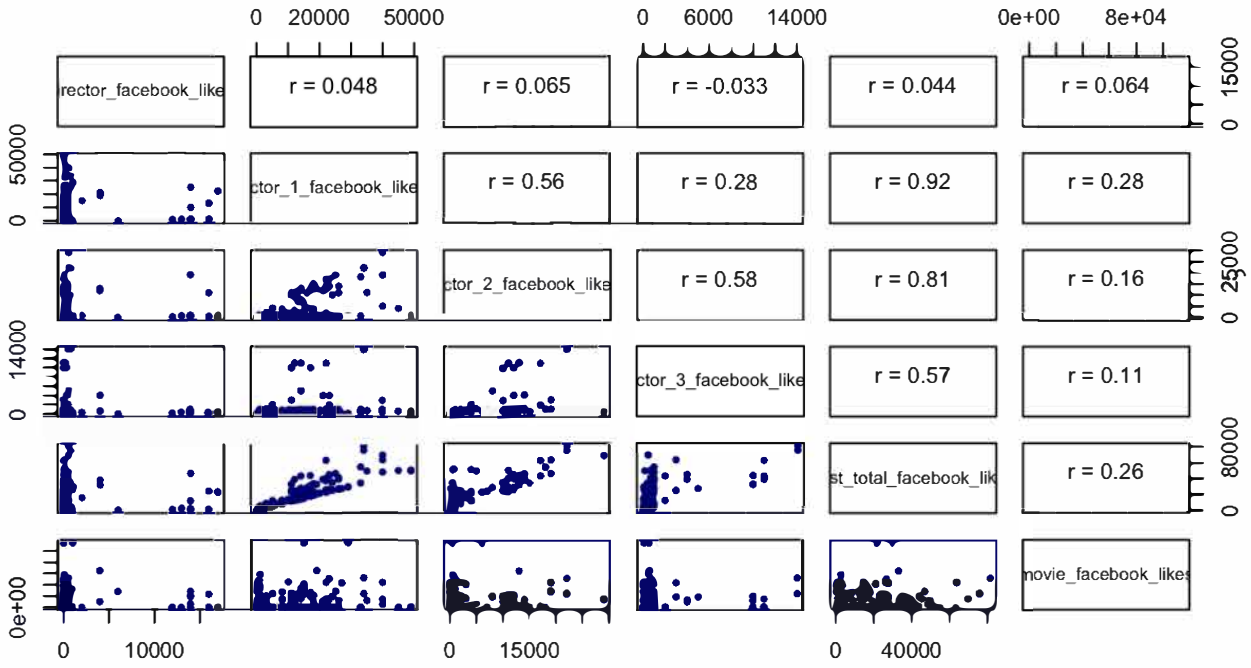
Graph 2.3: Aspect Ratio may interact with genre



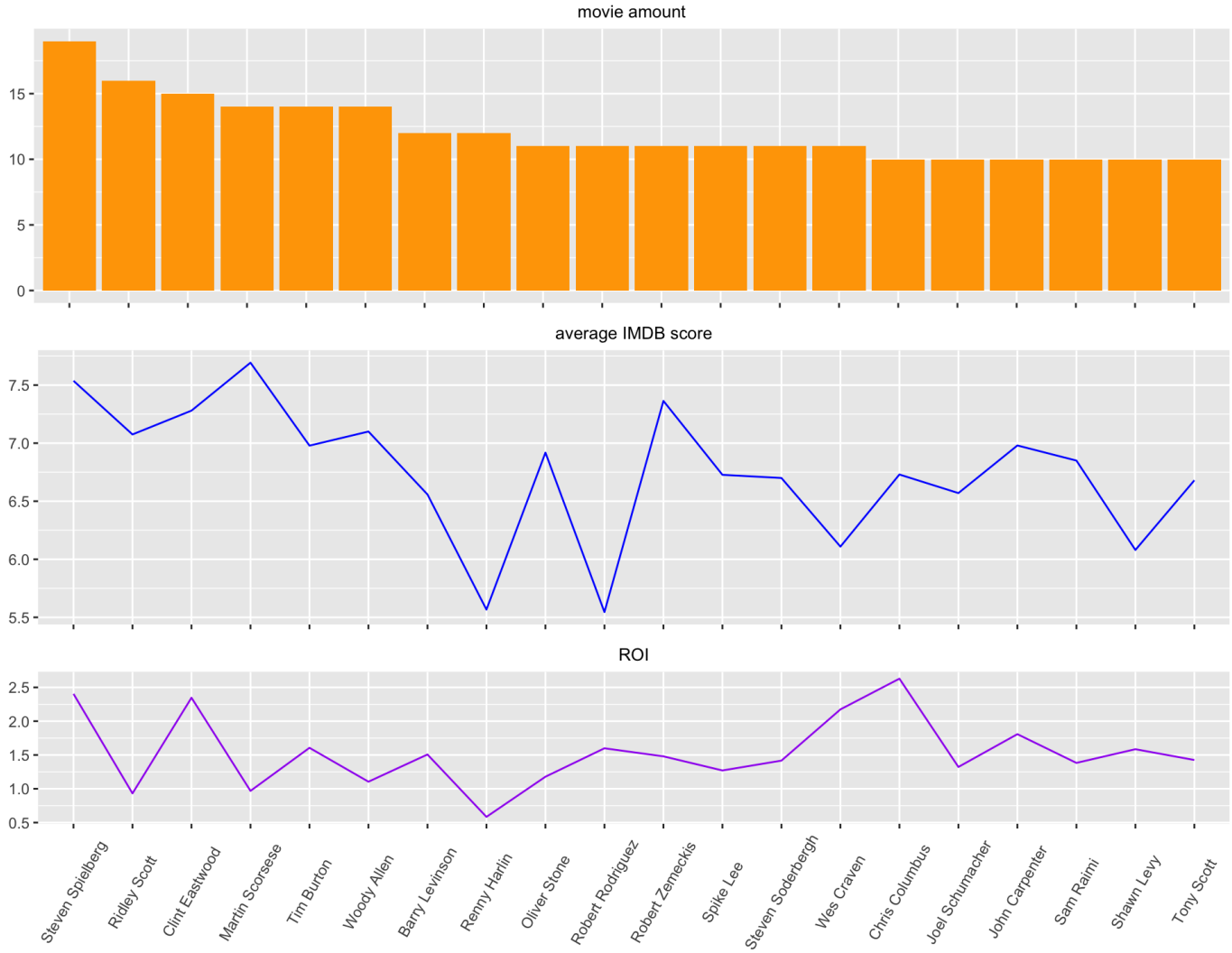
Graph 2.4: Rating may interact with genre



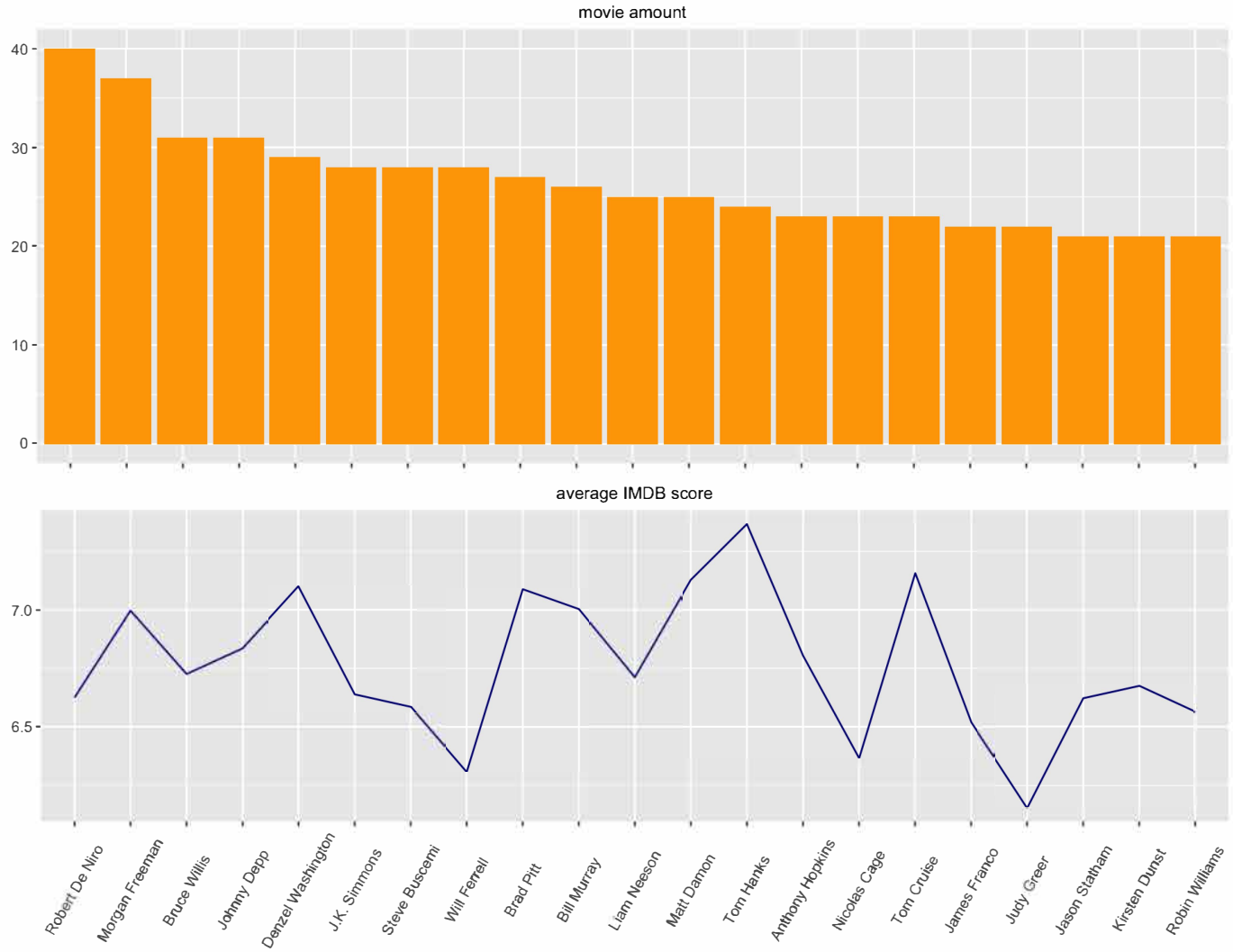
Graph 2.5: Facebook Likes Scatterplot Matrix



Graph 3.2: Top productive directors' move performace



Graph 3.3: Most starred actors' movie performace



MS&E 226 Project Part 2

Jeremy Bao, Nicole Nie

In this part we continue our exploration and prediction on IMDB movie dataset. After basic data cleaning, we find the best regression model in prediction for IMDB score, and best classification model in prediction for whether the movie profits.

1) Data cleaning

In the data cleaning process, firstly we split 20% of the data as the test set. Then we remove duplicate values and values that include NA, which we find not very useful. We also delete string covariates including the names of actors and directors, as well as the movies' key words. These are hard to deal with in our models.

Secondly, we find that most movies in the data set are in English (2883 out of 3018). Since movies in other languages can cause significant prediction error, we decide to only build models for English movies, and eliminate data with "language" not equal to "English".

Thirdly, we try to classify some covariates to make them easier to analyze. For "title_year", we classify it to "<60s", "60-80s", "80-00s", ">00s". For "genre", we separate it into 22 covariates, each indicating one genre. For movies which involves a specific genre, that value for that genre will be "true", otherwise "false". For "content_rating", we separate it into four covariates, "rateR", "ratePG", "rateG", "ratePG13". These are the four ratings with the highest number of movies, and the value will be "Y" if the movie falls into this rating. If all four covariates have values "N", that means this movie falls into other ratings, which is quite rare in our database. For "Aspect_ratio" and "color", we deal with it with the same method. For covariate "country", we change it to "USOrNot" (Y/N), for most of the English movies come from the US.

2) Regression

In this part, we try to build regression models to predict the IMDB score. The distribution of IMDB score can be seen below **(Figure 1)**. We have tried several techniques, including logarithm transformation, forward stepwise selection, lasso and ridge, adding higher order terms and interaction terms, etc. We use 10-fold cross validation to estimate the prediction error.

Baseline models

We set the baseline model to always predicts the average, which has a cross validation RMSE of **1.058**. Based on that, we build other models and compare the CV RMSE to it **(Table 2)**.

Logarithm transformations and forward stepwise selection

We then make logarithm transformations for covariates with great magnitude in number, including gross and budget, covariates related to different kinds of reviews, etc. We don't do

this for facebook_likes related variables since some of them are 0. We build a model that involves all the covariates after logarithm transformation, and get a cross validation RMSE of **0.723**. Based on that, we run forward stepwise selection to select useful variables, which results in a model with a lower RMSE of **0.720**.

Lasso and Ridge regression

We run lasso and ridge regression to build models. We find these methods not very useful here, and the λ that minimizes RMSE is close to zero **(Figure 2)**. The RMSE is higher than that of forward stepwise selection (**0.722** for lasso and **0.727** for ridge).

Adding interaction terms and higher order terms

We add these terms based on two references: one is that we run forward stepwise selection with all interaction terms included and select some interaction terms that cause a great decrease in AIC; the other is that we select covariates which we think may have great impacts on IMDB scores and add interaction terms and higher order terms for these covariates. Here we also delete the covariate "cast_total_facebook_likes" because we find it has a strong correlation with "actor_1_facebook_likes" (ρ=0.92), and we delete it to avoid collinearity.

Final Model

Our final model is below:

imdb_score ~ moviel + Drama + directorl + Animation + budgetl + votedl + Horror + grossl + duration + USorNot + userReviewl + ColorOrNot + criticl + Action + Fantasy + Thriller + Comedy + Crime + actor3l + actor1l + Musical + ratePG13 + rateR + Family + actor2l + I(votedl^2) + I(criticl^2) + grossl:budgetl + moviel:Action + moviel:Drama + moviel:Family + moviel:votedl + Drama:decades + budgetl:duration + decades:criticl + moviel:decades + rateR:moviel

Explanation of variable names can be seen in the appendix **(Table 1)**. It has a cross validation RMSE of **0.692**, which is lowest of all models. We believe this cross validation for RMSE can be an estimate of the test error, but we are concerned that this may be an optimistic estimate, for this model involves several higher order terms and interaction terms that may cause the model to overfit the training data.

3) Classification

We fit binary classification models to predict whether a movie profits (1) or loses money (0). The models used are: logistic regression, SVM, KNN, and random forest. Among them, random forest outperforms others by giving a prediction accuracy of 76%.

Covariates selection

In part 1 we notice some variable that may not have high correlation with response variables but we are not sure. Simply deleting them would lead to loss of hidden information, so we use

a full model and a log model (taking log of budget and number of reviews) in all situations and add regularization term when necessary.

Model selection:

Taking baseline model as a benchmark, we compare the average accuracy of the above models by repeating 10-fold cross validation for 5 times to decide the best model **(Table 3)**.

1) Baseline model

The baseline simply takes the frequencies of the two labels in the data, and predicts whichever is higher for all new data. Cross validation accuracies are around 54%, which is reasonable considering the number of positive and negative labels are nearly the same in the training set.

2) Logistic regression

We fit four models of this kind: full model and log model, and their versions with or without regularization term. They all yield average accuracy above 70%. The log model with regularization has the best accuracy of 74.9%, possibly because logarithmic transformation rescales the covariates to reasonable levels and regularization relieves overfitting to some extent.

3) SVM

Full model and log model under SVM do not differ much in CV prediction accuracy, though the latter gives a slightly better rate of 75.4%. The process of tuning SVM parameters and fitting the models has been quite computationally expensive, and the result is a black box.

4) K Nearest Neighbors

Because our covariates are a mixture of numeric and categorical variables, it would be hard to calculate the "distance" between two examples. We thus assume all categorical variables as discrete numeric variables. This approach proves unsuccessful, for the full model gives accuracy under 70% and log model even worse than baseline model.

5) Random Forest

Finally, we fit a random forest model which achieves average CV accuracy of 76.2%.

Overall, random forest had the best performance. It gave an estimated prediction accuracy of 76.2% or 24.8% test error rate. We then fit the model again on whole training set and observe the top important covariates **(Figure 3)**: those related to user reviews, budget, IMDB score, duration, and Facebook likes.

Appendix

1) Regression

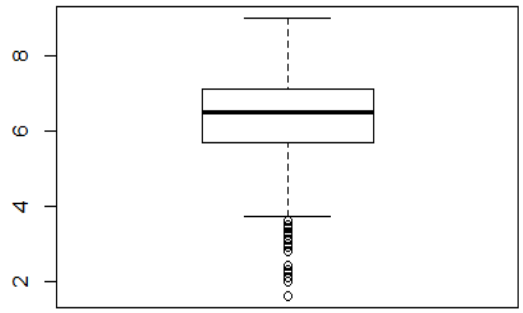


Figure 1 IMDB score distribution

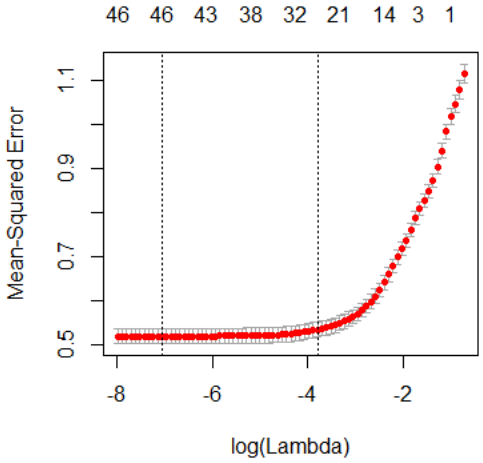


Figure 2 λ-MSE relationship in Lasso

Table 1 Explanation of variable names

Variable name	**Explanation**	**Variable name**	**Explanation**
budgetl	log(budget)	dacades	Classification variable on when the movie went out ("<60s", "60-80s", "80-00s", ">00s")
grossl	log(gross)	ratePG13	Whether this movie's rating is PG13 (Y/N) Similar for rateR and ratePG
votedl	log(num_voted_users)	Animation	Whether this movie's genres include animation (Y/N) Similar for other variables related to genres, like Action, Fantansy, etc.
criticl	log(num_critic_reviews)	USOrNot	Whether this movie is from the United States (Y/N)

Table 2 RMSE of regression models

Model Type	Baseline	Log-Full	Log-Stepwise Selection	Lasso	Ridge	Best model
CV RMSE	1.058	0.723	0.720	0.722	0.727	0.692

Best Model

```
model_IMDB_d <- lm(formula = imdb_score ~ movie_facebook_likes + Drama +
director_facebook_likes + Animation + budgetl + votedl + Horror + grossl +
duration + USorNot + userReviewl + ColorOrNot + criticl + Action + Fantasy +
Thriller + Comedy + Crime + actor_3_facebook_likes + actor_1_facebook_likes +
Musical + ratePG13 + rateR + Family + actor_2_facebook_likes + I(votedl^2) +
I(criticl^2)    +    grossl:budgetl    +    movie_facebook_likes:Action    +
movie_facebook_likes:Drama       +       movie_facebook_likes:Family       +
movie_facebook_likes:votedl    +    Drama:decades    +    budgetl:duration    +
decades:criticl + movie_facebook_likes:decades  + rateR:movie_facebook_likes ,
data = train_log)

cv_IMDB_d <- cvFit(model_IMDB_d, data = train_log, y = train_log$imdb_score,
K = 10, seed = 10)
cv_IMDB_d
```

2) Classification

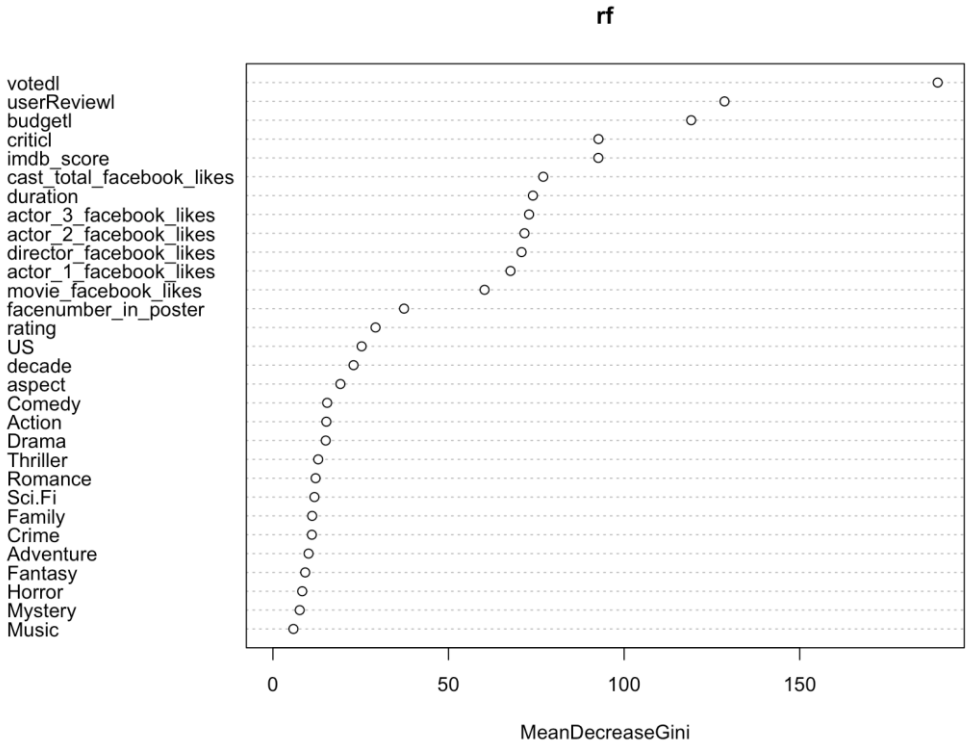


Figure 3 Variable importance in random forest

Table 3 Model performance

Average CV Accuracy	Baseline	Logistic	Regularized logistic	SVM	KNN	Random Forest
Full model	54.2%	72.7%	72.7%	74.7%	65.1%	76.1%
Log model	54.2%	74.7%	74.9%	75.4%	54.1%	76.2%

Model Selection:

```
source("CVAccuracy.R")

# full and log baseline
CVAccuracy(train, "simple")
CVAccuracy(train_log, "simple")

# full and log logit ( + regularization)
CVAccuracy(train, "logit")
CVAccuracy(train_log, "logit")
CVAccuracy(train, "logitReg")
CVAccuracy(train_log, "logitReg")

#full and log svm
svm_tune <- tune(svm, train.x=train[,-ncol(train)],
train.y=train[,ncol(train)],
              kernel="radial", ranges=list(cost=2^(-3:3), gamma=2^(-3:3)))
CVAccuracy(train, "svm")
svm_tune <- tune(svm, train.x=train_log[,-
ncol(train_log)],train.y=train_log[,ncol(train_log)],
              kernel="radial", ranges=list(cost=2^(-3:3), gamma=2^(-3:3)))
CVAccuracy(train_log, "svm")
```

```
#full and log knn
CVAccuracy(data.matrix(train), "knn")
CVAccuracy(data.matrix(train_log), "knn")

#full and log random forest
CVAccuracy(train, "rf")
CVAccuracy(train_log, "rf")
```

Cross Validation:

```
library(e1071)
library(class)
library(randomForest)
library(glmnet)

CVAccuracy <- function(train, model) {
  set.seed(1)
  N = 5;
  all.accuracy = rep(NA, N)
  for (n in 1:N) {
    K = 10;
    #Randomly shuffle the data
    train <- train[sample(nrow(train)),]
    folds <- cut(seq(1,nrow(train)),breaks=K,labels=FALSE)
    accuracy = rep(NA,K)
    for(i in 1:K){
      #Segement data by fold
      testIndexes <- which(folds==i,arr.ind=TRUE)
      testData <- train[testIndexes, ]
      trainData <- train[-testIndexes, ]
      if (model == "simple") {
        predict <- sum(trainData[,ncol(train)] == T) > 0.5
        fit <- rep(predict, nrow(testData))
      }
      if (model == "logit") {
        logit <- glm(isProfit ~ ., data = trainData, family = "binomial")
        fit <- (predict(logit, newdata = testData[,-ncol(train)], type =
"response") < 0.5)
      }
      if (model == "logitReg") {
        x = as.matrix(trainData[,-ncol(trainData)])
        y = as.matrix(trainData[,ncol(trainData)])
        cvfit = cv.glmnet(x, y, family = "binomial", type.measure = "class")
        fit <- predict(cvfit, newx = as.matrix(testData[,-ncol(train)]),
                       s = "lambda.min", type = "class")
      }
      if (model == "svm") {
        svmfit <- svm(isProfit ~ ., data = trainData, kernel = "radial",
                      cost = 1, gamma = 0.125)
        fit <- predict(svmfit, newdata = testData[,-ncol(train)])
      }
      if (model == "knn") {
        fit <- knn(trainData, testData, trainData[,ncol(train)], k = 5)
      }
      if (model == "rf") {
        rf <- randomForest(isProfit ~ ., data = trainData)
        fit <- predict(rf, testData[,-ncol(train)])
      }
      accuracy[i] <- sum(fit == testData[,ncol(train)]) / nrow(testData)
    }
    all.accuracy[n] = mean(accuracy, na.rm = T)
  }
  return(mean(all.accuracy))
}
```

Best Model:

```
rf <- randomForest(isProfit ~ . , data = train_log)
varImpPlot(rf)
importance(rf)
```

MS&E 226 Mini Project Part 3

Jeremy Bao, Nicole Nie

In this final part, we applied best models from part 2 on test set and examined the results. We also discussed the reliability of the output and the overall modeling procedures, and wrapped up with a project summary.

Test set performance

Regression

In the test set, our best prediction model has the RMSE of 0.6502, which is even smaller than the cross validation RMSE 0.6923. They are on a similar magnitude, so we believe that the CV RMSE can be a decent estimation of the test RMSE. The decrease in RMSE may derive from random sampling.

Classification

The estimated accuracy for random forest model from CV is 0.76, and prediction error 0.24. For the test set, prediction error is 0.23, slightly lower than CV estimated error. We may also deem the difference coming from random sampling and could conclude that CV did quite a good job in estimating our model's performance.

Inference

1) significant coefficients

Table 1 lists the statistical significance of different variables in the training and the test set. We can see that when we fit the model with the training set, we have 23 statistically significant covariates, while the number greatly decreases to 7 when fitted to the test set.

The statistical significance of a coefficient means: if this coefficient is 0, the probability of seeing the data we have is low (the exact probability was shown below **Table 1**). It suggests this coefficient is worth including in the model.

We don't quite believe in this result for two reasons. Firstly, we actually conducted "post-selection inference" here by first using plenty of methods to select the model and doing inference afterwards. By doing so, we favorably biased our selection of p-values, and therefore greatly increased the chance of making the covariates we selected statistically significant.

Secondly, there are several assumptions for the population model before we do analysis on statistical significance (The population model is linear; the errors are normally distributed; the errors are i.i.d., with mean zero). We believe the population model for our dataset can be complicated and may not perfectly obey these assumptions.

The result on the test set also shows this. When using different data to do inference, the statistically significant covariates dramatically decrease. The statistical significance analysis carried out on the test set can be more trustworthy. The result also indicates that our regression model is probably not a good model for inference.

Table 1 Significant covariates and confidence intervals

	Statistical Significance for Training Set	Statistical Significance for Test Set	Confidence interval from original R calculation	Confidence Interval from Bootstrap
(Intercept)	***		(6.8729, 13.7520)	(5.933, 13.825)
movie_facebook_likes			(-0.000119, 4.504E-05)	(-0.00012, 0.001062)
DramaTRUE			(-0.8676, 0.5323)	(-0.8339, 0.6080)
director_facebook_likes	*		(3.9629E-07, 1.7866E-05)	(2.739E-06, 1.554E-05)
AnimationTRUE	***	*	(-0.6177, 0.9065)	(0.6219, 0.9099)
budgetl			(-0.0799, 0.2002)	(-0.0950,0.2070)
votedl	***	**	(-1.5694, -1.0463)	(-1.6223, 1.0247)
HorrorTRUE	***	***	(-0.4620, -0.2601)	(-0.4643, -0.2574)
grossl			(-0.04677, 0.2206)	(-0.0840, 0.2521)
duration	***	**	(0.01583, 0.05176)	(0.0155, 0.0558)
USorNotN	***		(0.1062, 0.2449)	(0.1051, 0.2465)
userReviewl	***	.	(-0.2994, -0.1882)	(-0.3123, -0.1758)
ColorOrNotN	***		(0.1471, 0.4777)	(0.1730, 0.4126)
criticl			(-0.4699, 0.9106)	(-0.5182, 1.2099)
ActionTRUE	***		(-0.3054, -0.1530)	(-0.3064, 0.1520)
FantasyTRUE	**		(-0.2194, -0.0549)	(-0.2255, -0.0477)
ThrillerTRUE	***		(-0.1949, -0.05479)	(-0.1952, 0.0553)
ComedyTRUE	***		(-0.2157, -0.0829)	(-0.2208, -0.0789)
CrimeTRUE			(-0.01442, 0.1313)	(-0.0140, 0.1283)
actor_3_facebook_likes	*		(-3.2164E-05, -4.1823E-08)	(-3.2173E-05, -3.4760E-06)
actor_1_facebook_likes			(-1.0427E-06, 2.2517E-06)	(-1.46279E-06, 1.76932E-06)
MusicalTRUE			(-0.1351, 0.1952)	(-0.1752, 0.2309)
ratePG13N	***		(0.10244, 0.2995)	(0.07684, 0.3213)
rateRN			(-0.0300, 0.1682)	(-0.0468, 0.1837)
FamilyTRUE	**		(-0.328, -0.06970)	(-0.3585, -0.0438)
actor_2_facebook_likes			(-7.00977E-06, 6.1544E-06)	(-4.486E-06, 7.0588E-06)
I(votedl^2)	***	***	(0.07549, 0.1007)	(0.7489, 0.1025)
I(criticl^2)	*	*	(-0.07666, -0.008893)	(-0.1092, 0.0102)
budgetl:grossl	.		(-0.01567, 0.001046)	(-0.01733,0.003244)
movie_facebook_likes:ActionTRUE			(-1.3818E-06, 5.1828E-06)	(1.2450E-06, 5,4886E-06)
movie_facebook_likes:DramaTRUE	**		(-8.5510E-06, -2.1480E-06)	(-8.5511E-06, -1.7078E-06)
movie_facebook_likes:FamilyTRUE	***		(5.2339E-06, 1.5600E-05)	(5.9267E-06, 1.6213E-05)
movie_facebook_likes:votedl			(-1.7955E-06, 1.4660E-06)	(-2.1537E-06, 2.1755E-06)
DramaFALSE:decades60-80s			(-4.0355, 1.5158)	(-4.0230, 2.2985)

DramaTRUE:decades60-80s			(-4.0102, 1.7009)	(-3.8839, 2.2199)
DramaFALSE:decades80-00s			(-4.6523, 0.6247)	(-4.0593, 1.9314)
DramaTRUE:decades80-00s			(-4.2286, 1.2431)	(-4.0593, 1.9313)
DramaFALSE:decades>00s	*	**	(-5.3861, -0.1355)	(-5.3333, 0.7981)
DramaTRUE:decades>00s			(-4.7920, 0.6570)	(-4.6309, 1.3169)
budgetl:duration	**		(-0.002465, -0.000399)	(-0.00269, -0.00037)
criticl:decades60-80s			(-0.4444, 0.8716)	(-0.5836, 0.8375)
criticl:decades80-00s			(-0.2848, 0.9768)	(-0.4611, 0.9351)
criticl:decades>00s			(-0.2130, 1.0418)	(-0.3917, 1.001969)
movie_facebook_likes:decades60-80s			(-5.0628E-05, 0.0001157)	(-0.00106, 0.000109)
movie_facebook_likes:decades80-00s			(-4.2559E-05, 0.0001166)	(-0.001051, 0.0001134)
movie_facebook_likes:decades>00s			(-4.1139E-05, 0.0001174)	(-0.001052, 0.0001133)
movie_facebook_likes:rateRN	**		(1.1171E-06, 6.5555E-06)	(1.4546E-06, 6.6945E-06)

P values: '***' means p-value between 0 and 0.001; '**' means p-value between 0.001 and 0.01; '*' means p-value between 0.01 and 0.05; '.' means p-value between 0.05 and 0.1; ' ' means p-value between 0.1 and 1.

2) Bootstrap analysis on confidence intervals

We can see that the confidence interval calculated by bootstrap is basically similar to what R gave us in standard regression output.

The bootstrap does not require assumptions of the population model while the confidence interval we get from R requires several assumptions as we mentioned earlier. It probably indicates that the population model basically obeys the three assumptions.

3) Comparison with full model

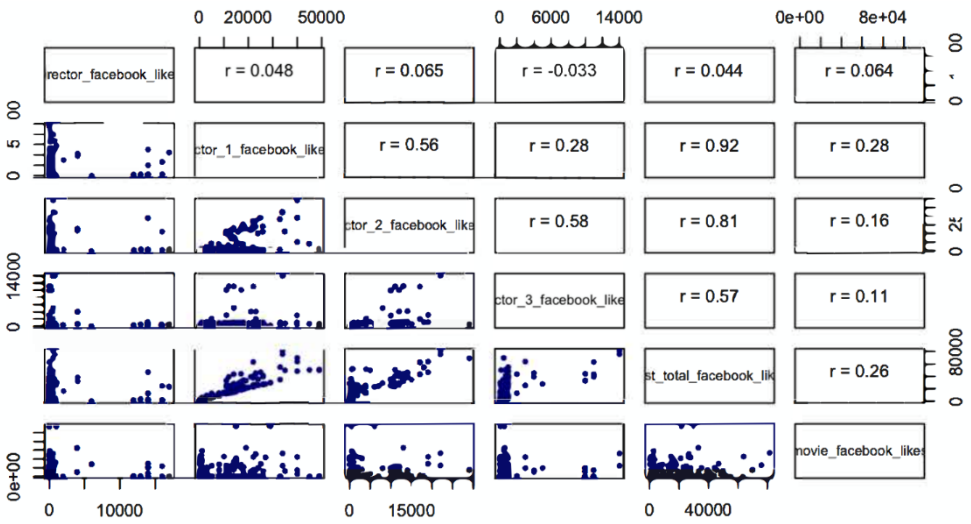
Table 2 Some regression coefficients of best model v.s. full model

covariate	**Full model**	**Best model**
director_facebook_likes	1.533e-05	9.131e-06
actor_3_facebook_likes	3.701e-05	-1.610e-05
votedl	5.136e-01	-1.308e+00
userReviewl	-1.913e-01	-2.438e-01
duration	8.748e-03	3.375e-02
USorNot	1.574e-01	1.755e-01
ColorOrNot	2.825e-01	2.973e-01
Animation	8.141e-01	7.621e-01
Action	-1.674e-01	-2.293e-01
Family	-1.334e-01	-1.989e-01
Fantasy	-1.213e-01	-1.372e-01
Horror	-4.283e-01	-3.610e-01
Thriller	-1.597e-01	-1.249e-01
Comedy	-1.615e-01	-1.493e-01
ratePG13	3.963e-01	2.010e-01

Most coefficients changed. Some were still in compatible value levels, while others changed a lot and even had opposite sign **(Table 2)**. For example, coefficient of numeric variable actor_3_facebook_likes decreased to negative value from full model to our best model, while coefficient of categorical variable USorNOT and ColorOrNot only increased very slightly. For those coefficients that changed dramatically, there may be loss of related information in the covariates that we dropped in model selection: whether collinearity or hidden variable. For those that did not change much, we guess they were quite independent variables that were not strongly affected by other variables: holding other things equal, their association with the response variable is always nearly at constant.

4) Potential problems

Collinearity: In part 1 of the project we discovered that some pairs in the four Facebook covariates (actor 1-3 and total cast) have quite strong correlation (r as high as 0.8 or 0.9, see **Graph 1**). This suggested some collinearity between them. Such associative correlation exists in several variables related to user reviews and votes as well, but not that strong when we explored the data in the first stage. Although the coefficients significance our final best model successfully filtered out several Facebook covariates, some collinearity effect of "user" covariates may still affect our analysis.



Graph 1 Associative correlation between Facebook variables

Multiple hypothesis testing: Our best model used 37 covariates in all, with 23 of them being significant at p-value of 0.05 or lower. At a cut-off level of 0.05, the number of false positives we expect is $1.85 \approx 2$. Therefore, not all of the 23 "significant" covariates are really significant. Using Bonferroni correction, we adjusted the cut-off level down to $0.05/37 \approx 0.00135$. Under the new cut-off level, the number of significant covariates reduced to 16.

Post-selection inference: This problem may be more serious in our model than the above two. Because of the imbalance in data (much more English movies than other languages), we dropped non-English examples for convenience and therefore dropped the variable "English" as well. We treated several categorical variables (rating, aspect ratio, etc.) in this way for similar reasons. In this sense, our dataset and covariates had been selected favorably to some extent. Although we had 23 significant covariates at last, some of them may not be

that significant had we remained all data and covariates as they were.

5) Reasoning on causal relationship

We would rather interpret the significant relationships as correlation rather than causation. A main reason is that all the covariates were not directly related with a movie's content or user's comments (in words), which are in reality the most influential factors to IMDB score. Another reason is that our response variable and some covariates may affect each other so that there may be two-way relationship. One of such covariates that might confound our causal analysis is gross: gross revenue may be the cause as well as the result of our response variable – IMDB score.

Project Summary

1) Use of the model

We think our two models should be primarily used for prediction, i.e. they can be used to predict the IMDB score of a new movie (so people can decide to watch it or not) or whether this new movie can make profits (it can be a reference for the movie team). It is probably not good enough for inference, because we don't know exactly about the population model, and there are still factors in the population model we haven't covered. A case in point is the movies' keywords or keywords from users' comments. We believe these should be somehow considered in the population model. However, dealing with such covariates requires natural language processing and we currently cannot take them into consideration. Therefore, it is a better idea to use this model for prediction instead of inference. The decrease in statistically significant variables when we fit the model on the test set also shows it is probably not a good model for inference.

2) Explanatory power over time

We believe our model needs refitting frequently. One reason is that people's ideas about different movies change quickly over time, and are highly related to the economic, societal and cultural factors, all of which can change with time. Another reason is that the model includes covariates of "facebook_likes", which are updated quickly and can be unstable. We suggest refitting our model once a month.

3) Awareness for application

There is a series of data cleaning and data transformation that could be found in our code in part 2, mainly dealing with categorical variables and genres. A top awareness is that this model is only used for movies whose language is English.

4) Data quality

If possible, we would like to add on covariates that dive more deeply into user review, e.g. comment key words.

5) Possible improvement

On a second attack, we would spend more timing parsing the plot keywords that are relevant to movie content. This covariate is actually presented in original dataset, but processing the words basket would be much more time-consuming than our current approach that we did not manage to do so. But it is definitely an attempt worth trying.