Final Project: Predict Movie Quality

Alexander Wu

March 22, 2017

Abstract

What traits do great movies share? Is there a way to determine the quality of a movie before it is released? Will Wonder Woman be a good movie? I address these questions by exploring the data and applying two different classification models: binomial logistic regression and random forests. I conclude that good movies (IMDb score > 7.5) are longer, tend to have fewer faces in posters, and have a high number of facebook likes for the director and actors involved. Also, the logisitc regression and random forests are both decent classifiers (both with accuracy ~85%). Finally, I apply my random forests classifier to predict that Wonder Woman will not have an IMDb score above 7.5.

Introduction

The problem I address is in this project is predicting what makes a good movie. Since there are so many new movies that come out each year, it would be nice if we could determine which movies will be good without relying on critics or our own instincts (which can be unreliable at times).

The dataset comes from kaggle.com and was published by Chuan Sun on August 22, 2016. There are 28 variables, 12 of which are categorical. There are variables directly about the movie (like IMDb score, title year, duration, etc.) and variables related to the people involved in the movie (directors, actors, and critics). I use R to perform my analyses on the data.

In the first half of my report, I engage in some exploratory data analysis. The purpose of this is twofold: to uncover interesting insights from the data, and secondly to select appropriate variables for building a classification model used to predict if a movie will be good (has a IMDb score of at least 7.5). In second half of my report, I compare the logistic regression and random forests model in terms of how well they classify good movies. I also analyze insights gained from these models. I didn't choose to include other classifiers like k-Nearest Neighbors because these classifiers don't work as well when dealing with categorical data. Although the classifiers I chose only do a decent job at classifying (~85% accuracy), I was able to learn which variables were important in determining a good movie.

Part 1: Exploratory Data Analysis

Data exploration

Below, I list the variables of the dataset in question.

Variable Name	Description
director_name	director name
director_facebook_likes	director facebook likes
actor_1_name	actor 1 name
actor_1_facebook_likes	actor 1 facebook likes
actor_2_name	actor 2 name
actor_2_facebook_likes	actor 2 facebook likes
actor_3_name	actor 3 name
actor_3_facebook_likes	actor 3 facebook likes
cast_total_facebook_likes	cast total facebook likes
movie_facebook_likes	movie facebook likes
imdb_score	IMDb rating score (0-10)

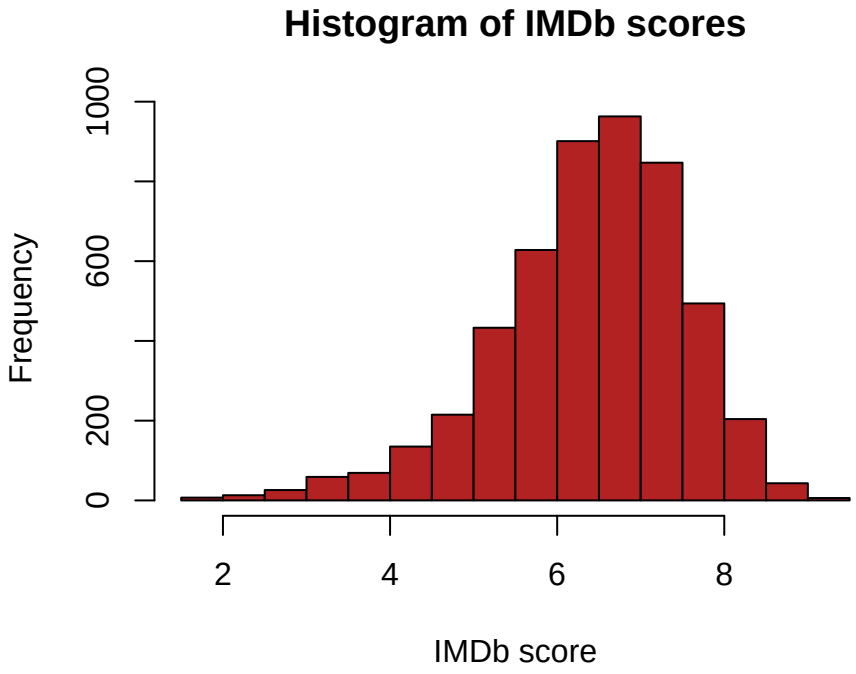
Variable Name	Description
num_voted_users	number of voted users
num_critic_for_reviews	number of reviews from critics
num_user_for_reviews	number of reviews from users
facenumber_in_poster	human faces in primary poster
movie_title	movie title
title_year	title year
duration	duration
country	country
genres	genres
color	color
aspect_ratio	aspect ratio
content_rating	content rating
plot_keywords	plot keywords
language	language
budget	budget
gross	gross
movie_imdb_link	movie IMDb link

I do some data exploration in order to determine which variables I want to use in my classification models. First, I load my data from the csv file.

```
imdb <- read.csv("movie_metadata.csv",
                 stringsAsFactors = TRUE,
                 strip.white = TRUE,
                 na.strings = c("NA",""))
```

Let's take a look at the distribution of the scores.

```
hist(imdb$imdb_score,col="firebrick",
     main = "Histogram of IMDb scores",
     xlab = "IMDb score")
```



It looks like a nice normal distribution. I believe movies with an IMDb score of 7.5 or higher constitutes a "good" movie.

```
imdb <- mutate(imdb,good_movie = as.factor(ifelse(imdb$imdb_score >= 7.5,"Yes","No")))
num.good_movie <- nrow(imdb[imdb$good_movie == "Yes",])
num.good_movie
```

```
## [1] 887
```

```
good_movie.ratio <- num.good_movie/nrow(imdb)
good_movie.ratio
```

```
## [1] 0.1758874
```

There are 887 good movies that make up 17.6% of the dataset. Let's see if we have any missing data.

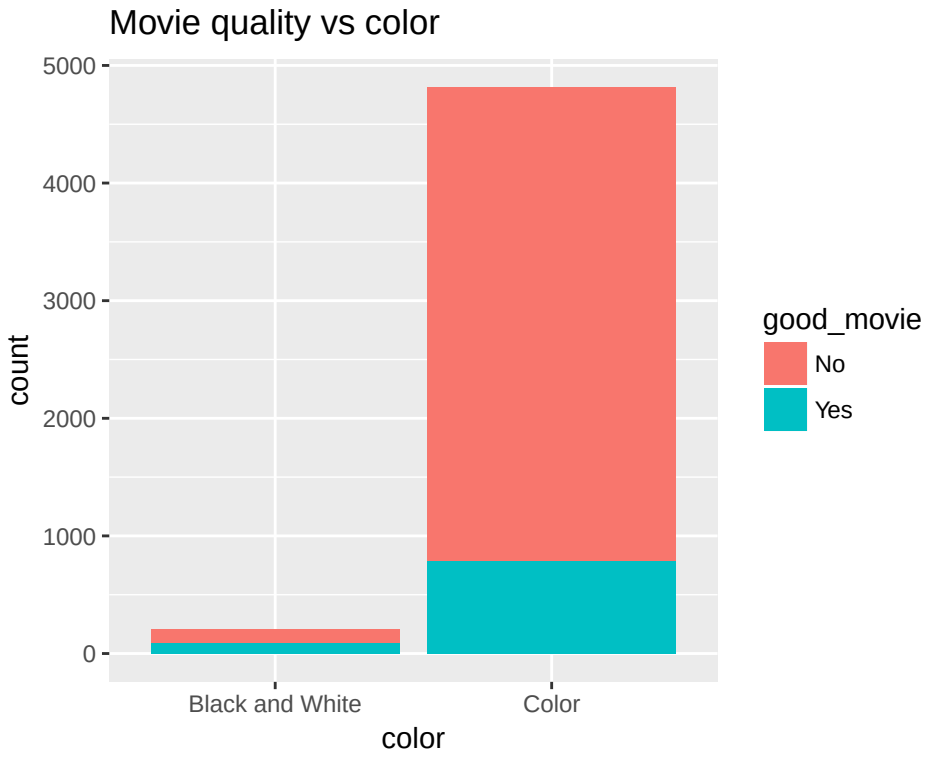
```
sapply(imdb,function(x) sum(is.na(x)))
```

```
##                      color             director_name
##                         19                       104
##    num_critic_for_reviews                  duration
##                         50                        15
##   director_facebook_likes    actor_3_facebook_likes
##                        104                        23
##              actor_2_name    actor_1_facebook_likes
##                         13                         7
##                      gross                    genres
##                        884                         0
##              actor_1_name               movie_title
##                         7                         0
##           num_voted_users cast_total_facebook_likes
##                         0                         0
##              actor_3_name      facenumber_in_poster
##                        23                        13
##             plot_keywords           movie_imdb_link
##                       153                         0
##      num_user_for_reviews                  language
##                        21                        12
##                   country            content_rating
##                         5                       303
##                    budget                title_year
##                       492                       108
##    actor_2_facebook_likes                imdb_score
##                        13                         0
##              aspect_ratio      movie_facebook_likes
##                       329                         0
##                 good_movie
##                         0
```

There is a lot of missing data about the gross (884 missing) and budget (492 missing) of movies.

Movie quality vs color

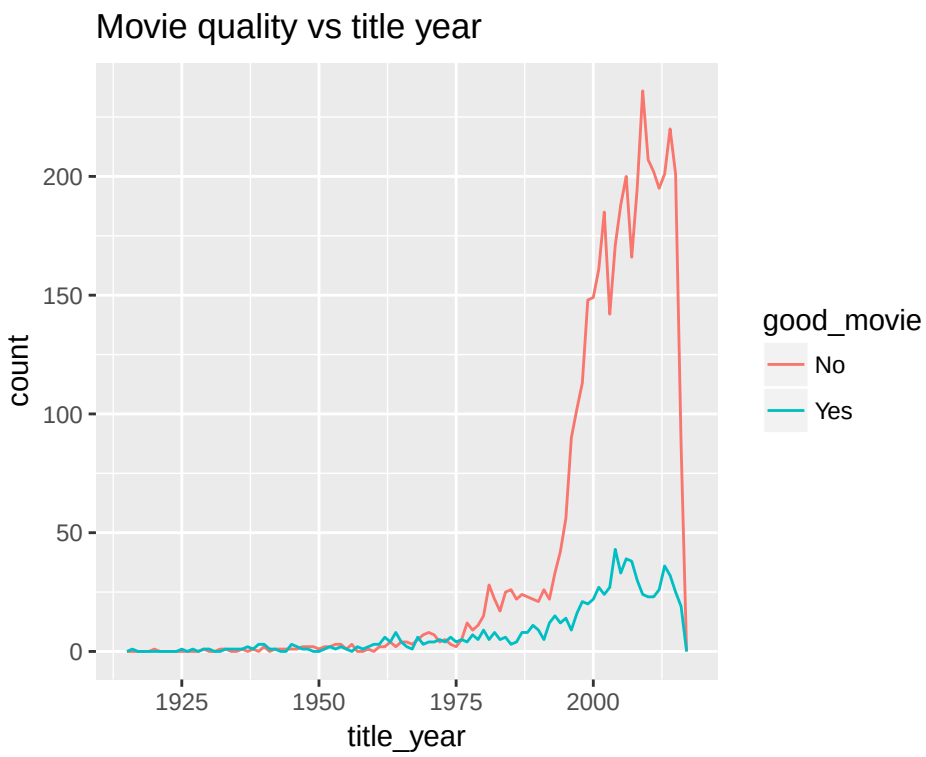
```
temp <- imdb[c("good_movie","color")]
temp <- na.omit(temp)
ggplot(temp, aes(color,fill=good_movie)) +
  geom_bar() +
  ggtitle("Movie quality vs color")
```



It looks like black and white movies tend to be better movies judging based on the proportion of good movies to bad movies.

Movie quality vs title year

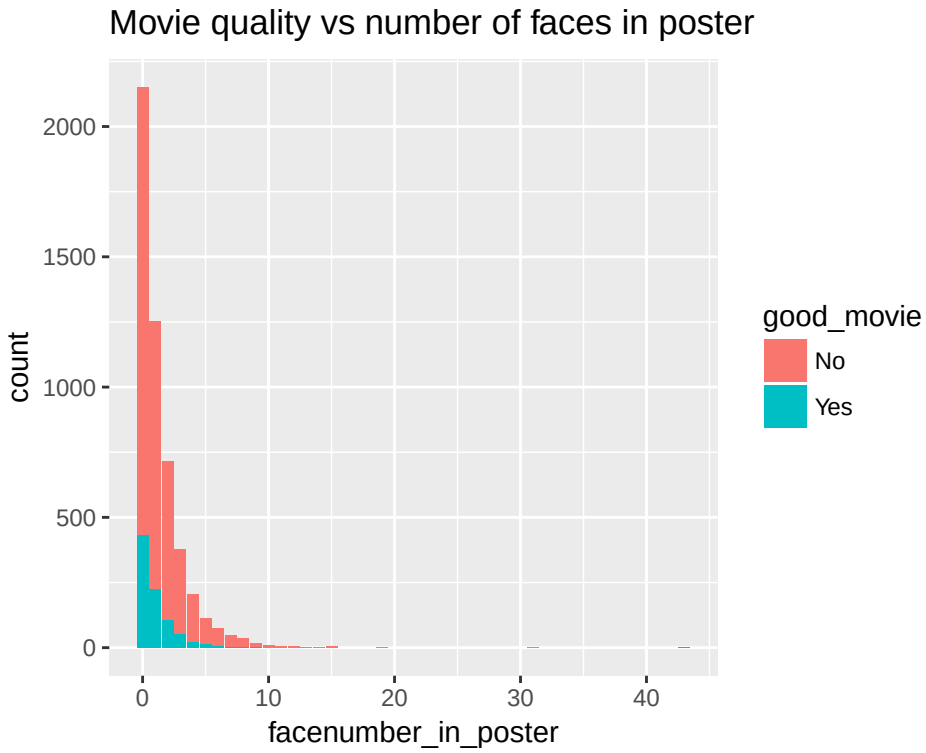
```
temp2 <- imdb[c("good_movie","title_year")]
ggplot(temp2, aes(title_year, colour = good_movie)) +
  geom_freqpoly(binwidth = 1) +
  ggtitle("Movie quality vs title year")
```



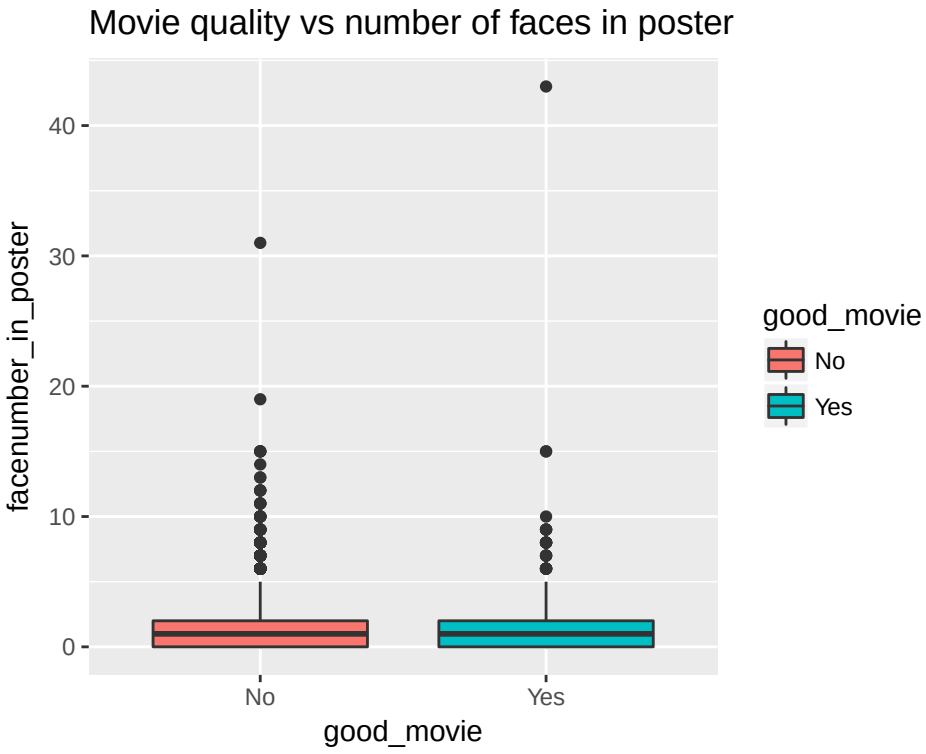
There is larger proportion of good movies to bad movies for movies made before 1990. Perhaps this is because old movies that are bad are forgotten and only the good things from the past are kept.

Movie quality vs number of faces on movie poster

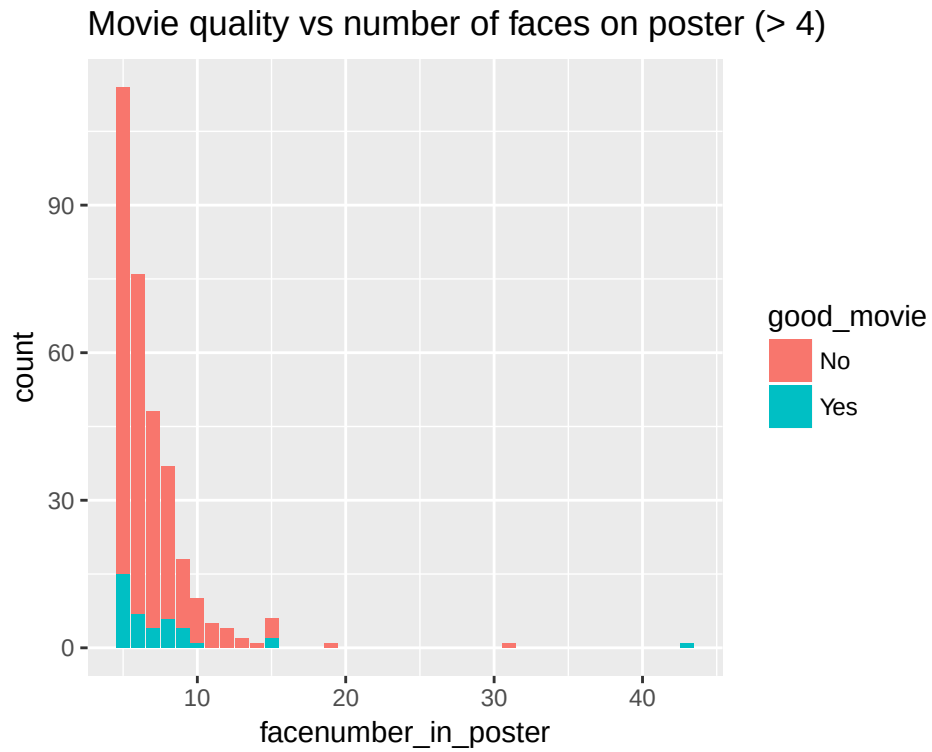
```
ggplot(imdb, aes(facenumber_in_poster,fill=good_movie)) +
  geom_bar() +
  ggtitle("Movie quality vs number of faces in poster")
```




```
ggplot(imdb, aes(good_movie,facenumber_in_poster)) +
  geom_boxplot(aes(fill=good_movie)) +
  ggtitle("Movie quality vs number of faces in poster")
```




```
temp3 <- subset(imdb, facenumber_in_poster > 4, select = c("good_movie","facenumber_in_poster"))
ggplot(temp3, aes(facenumber_in_poster,fill=good_movie)) +
  geom_bar() +
  ggtitle("Movie quality vs number of faces on poster (> 4)")
```




```
good.movies <- subset(imdb,good_movie=="Yes")
bad.movies <- subset(imdb,good_movie=="No")
# Summary statistic of director facebook likes of good movies
pander(summary(na.omit(good.movies$facenumber_in_poster)))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	0	1	1.137	2	43

```
# Summary statistic of director facebook likes of bad movies
pander(summary(na.omit(bad.movies$facenumber_in_poster)))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	0	1	1.421	2	31

It is rare for a good movie to have more than 10 faces on the movie poster. Good movies tend to have fewer faces in the movie poster.

Movie quality vs director facebook popularity

Below, I use the natural log of director facebook likes because the number of likes ranges from 0 to 23000, which spans many orders of magnitude and is hard to see in a boxplot.

```
ggplot(imdb, aes(good_movie,log(director_facebook_likes))) +
  geom_boxplot(aes(fill=good_movie)) +
  ggtitle("Movie quality vs log of director facebook likes")
```


```
# Summary statistic of director facebook likes of good movies
pander(summary(na.omit(good.movies$director_facebook_likes)))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	0	103	1731	453.2	22000

```
# Summary statistic of director facebook likes of bad movies
pander(summary(na.omit(bad.movies$director_facebook_likes)))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	8	44	478.6	168	23000

Good movies tend to have directors that are more popular on facebook.

Movie quality vs top 3 actors/actresses facebook popularity

```
ggplot(imdb, aes(good_movie,log(actor_1_facebook_likes))) +
  geom_boxplot(aes(fill = good_movie)) +
  ggtitle("Movie quality vs log of actor 1 facebook likes")
```

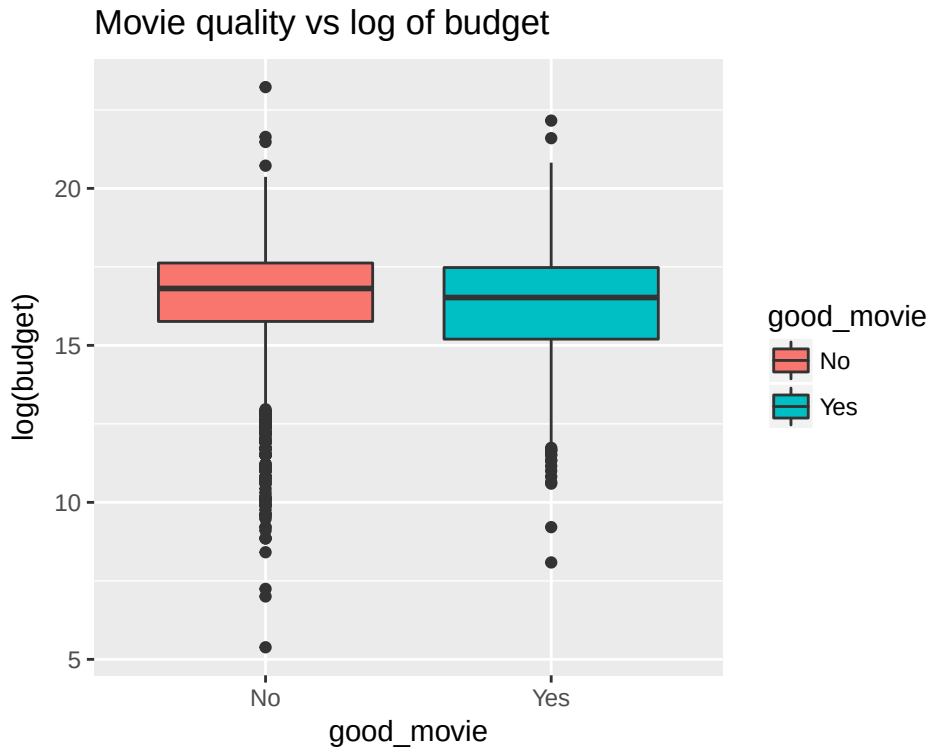

```
ggplot(imdb, aes(good_movie,log(actor_2_facebook_likes))) +
  geom_boxplot(aes(fill = good_movie)) +
  ggtitle("Movie quality vs log of actor 2 facebook likes")
```


```
ggplot(imdb, aes(good_movie,log(actor_3_facebook_likes))) +
  geom_boxplot(aes(fill = good_movie)) +
  ggtitle("Movie quality vs log of actor 3 facebook likes")
```

There doesn't seem to be a relation between quality of movie and facebook popularity of the top three actors/actresses.

Movie quality vs budget

```
ggplot(imdb, aes(good_movie,log(budget))) +
  geom_boxplot(aes(fill = good_movie)) +
  ggtitle("Movie quality vs log of budget")
```




```
# Summary statistic of director facebook likes of good movies
pander(summary(na.omit(good.movies$budget)))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3250	3988500	1.5e+07	43373970	3.9e+07	4.2e+09

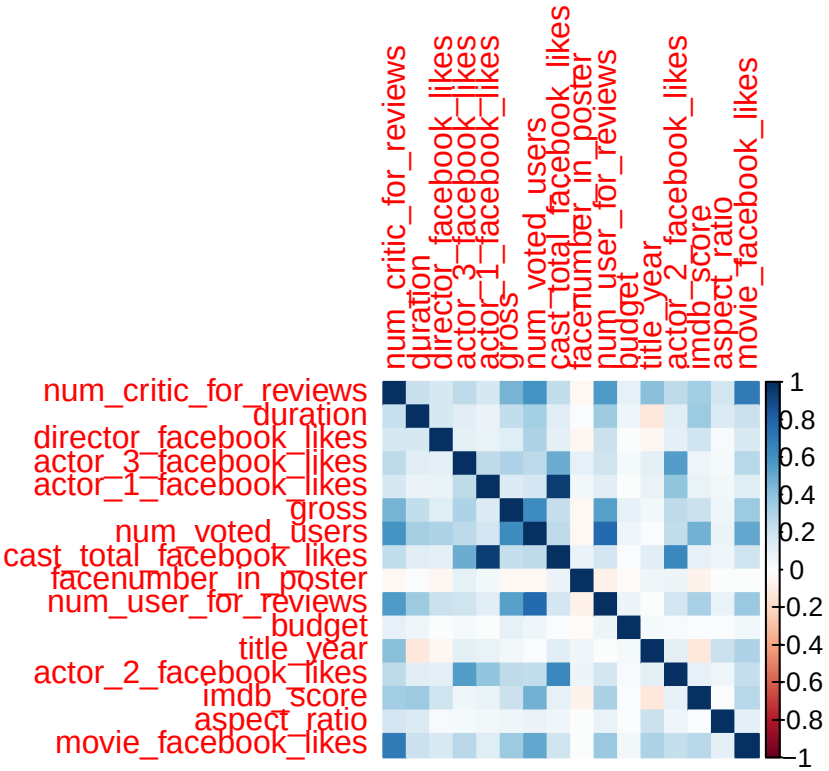
```
# Summary statistic of director facebook likes of bad movies
pander(summary(na.omit(bad.movies$budget)))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
218	7e+06	2e+07	39018589	4.5e+07	1.222e+10

There doesn't seem to be a relation between budget and movie quality. A good movie takes more than just money. You probably won't make a great movie with a budget of $3000 however.

Correlation analysis

```
imdb.continuous.vars <- select_if(imdb, is.numeric)
imdb.continuous.vars <- na.omit(imdb.continuous.vars)
imdb.cor <- cor(imdb.continuous.vars)
corrplot(imdb.cor, method="color")
```



This correlation plot tells us many things. "cast_total_facebook_likes" is heavily correlated with "actor_1_facebook_likes" and a little less correlated with "actor_2_facebook_likes" and "actor_3_facebook_likes" (because the actors are part of the cast). Also, older movies tend to be longer than newer movies. Not surprisingly, "movie_facebook_likes", "num_voted_users", "gross", and "num_user_reviews" are all correlated with each other.

Variables to be used in models

After exploring our data, it is time to come up with variables used to train our model. Many variables are not applicable (like IMDb link), so I removed these. Some other variables are collinear with other variables (like "cast_total_facebook_likes"), so I remove these also. Since I am interested in prediction, I remove "movie_facebook_likes", "num_voted_users", "num_critic_for_reviews", "num_user_for_reviews", and "gross" because these information will not be available before a movie is released. In the end, I am left with the following:

Variable Name	Description
director_facebook_likes	director facebook likes
actor_1_facebook_likes	actor 1 facebook likes
actor_2_facebook_likes	actor 2 facebook likes
actor_3_facebook_likes	actor 3 facebook likes
facenumber_in_poster	human faces in primary poster
title_year	title year
duration	duration
color	color
budget	budget

```
ms <- imdb[,c("good_movie",
              "director_facebook_likes",
              "duration",
              "actor_1_facebook_likes",
              "actor_2_facebook_likes",
              "actor_3_facebook_likes",
```

```
              "facenumber_in_poster",
              "title_year",
              "color",
              "budget")]

ms <- na.omit(ms)
```

Part 2: Model fitting

Creating testing and training set

I will create the testing and training set that will be used to build and validate two different models: logistic regression and random forest. To do this, I random select 70% of the rows for training and use the rest for testing. I choose this method rather than k-fold cross validation or leave-one-out cross-validation because I am concerned that creating a random forest will take much more time with these other resampling methods.

```
set.seed(3)
train <- sample(1:nrow(ms), 0.70*length(ms$good_movie))
ms.train <- ms[train,]
ms.test <- ms[-train,]

# Original Data
pander(table(ms$good_movie)/nrow(ms))
```

No	Yes
0.8324	0.1676

```
# Training Data
pander(table(ms.train$good_movie)/nrow(ms.train))
```

No	Yes
0.8352	0.1648

```
# Testing Data
pander(table(ms.test$good_movie)/nrow(ms.test))
```

No	Yes
0.8259	0.1741

The training and testing data sets have roughly the same proportion of good movies to bad movies as the original data set, a good sign! We are now ready to train and test our logistic regression and random forests models.

Logistic Regression

```
glm.fit.train <- glm(good_movie ~., data=ms.train, family=binomial)
glm.prob.test <- predict(glm.fit.train, type="response", newdata = ms.test)
glm.pred.test <- as.factor(ifelse(glm.prob.test<0.5,"No","Yes"))
glm.error <- table(glm.pred.test, true=ms.test$good_movie)
```

```r
# Confusion matrix
pander(glm.error)
```

	No	Yes
No	1079	192
Yes	36	43

```r
# Accuracy rate
sum(diag(glm.error))/sum(glm.error)
```

```
## [1] 0.8311111
```

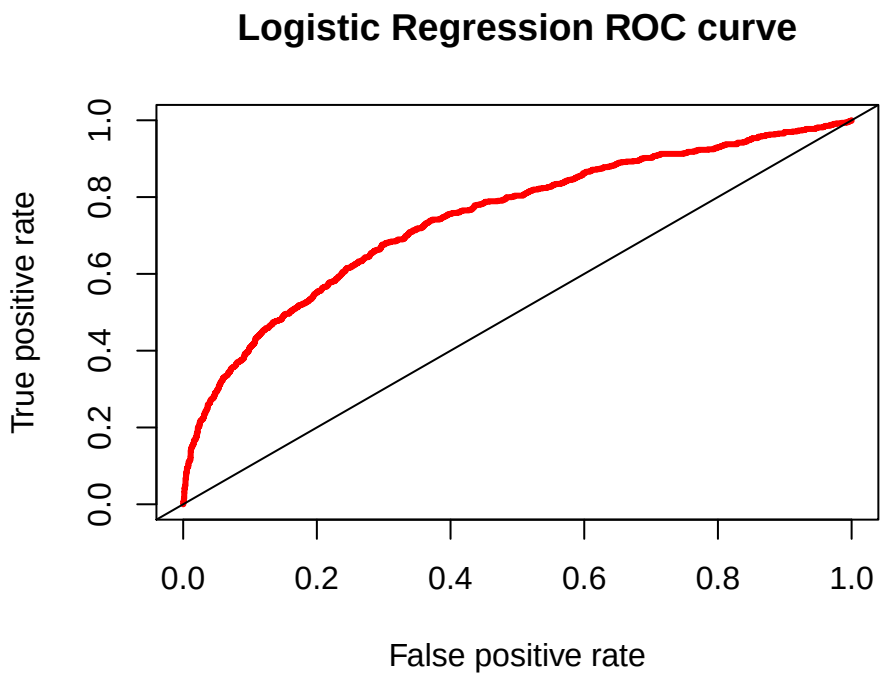
```r
# True positive rate
glm.error[1,1]/sum(glm.error[1,])
```

```
## [1] 0.8489378
```

```r
# True negative rate
glm.error[2,2]/sum(glm.error[2,])
```

```
## [1] 0.5443038
```

```r
glm.fit <- glm(good_movie ~., data=ms, family=binomial)
glm.prob <- predict(glm.fit, type="response", newdata = ms)
glm.pred <- prediction(glm.prob, ms$good_movie)
glm.perf <- performance(glm.pred, measure="tpr", x.measure="fpr")
plot(glm.perf, col=2, lwd=3, main="Logistic Regression ROC curve")
abline(0,1)
```




```r
glm.auc <- performance(glm.pred, "auc")@y.values
glm.auc
```

```
## [[1]]
```

```
## [1] 0.743271
```

The AUC was computed to be 0.7433. Logistic regression is a decent model.

```
pander(summary(glm.fit))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	48.83	6.343	7.699	1.372e-14
director_facebook_likes	6.995e-05	1.145e-05	6.108	1.007e-09
duration	0.02585	0.001882	13.74	6.207e-43
actor_1_facebook_likes	1.362e-06	3.067e-06	0.4441	0.657
actor_2_facebook_likes	2.196e-05	1.083e-05	2.027	0.04265
actor_3_facebook_likes	6.788e-06	2.545e-05	0.2667	0.7897
facenumber_in_poster	-0.1045	0.02696	-3.878	0.0001054
title_year	-0.02618	0.003187	-8.215	2.121e-16
colorColor	-1.046	0.1764	-5.932	3e-09
budget	-8.677e-11	2.797e-10	-0.3102	0.7564

(Dispersion parameter for binomial family taken to be 1)

Null deviance:	4068 on 4499 degrees of freedom
Residual deviance:	3533 on 4490 degrees of freedom

Title year and duration are the most significant variables in this model since the summary indicated a very small p-value ($< 2 \times 10^{-16}$). Similarly, this summary also indicates that director facebook likes, number of faces in poster, and color are important factors, although not as much as duration or title year. Interestingly, the number of facebook likes of actor 2 is significant, but the number of facebook likes of actor 1 and actor 3 are not. We see that budget does not really affect the quality of a movie. We can see how each variable affects our prediction by observing coefficients in the "Estimate" column. From the sign of these coefficients, we see that good movies tend to have fewer faces, tend to be older films, and tend to be black and white. Also, longer movies and movies with high number of actor/director facebook likes tend to be better films.

```
# McFadden R^2
pander(pR2(glm.fit))
```

llh	llhNull	G2	McFadden	r2ML	r2CU
-1766	-2034	535.3	0.1316	0.1122	0.1885

We observe a McFadden R^2 of 0.1316 which is decent, but not very good.

Random Forests

I will now try using a random forest model.

```
set.seed(3)
# Fit random forest model
rf.fit.train <- randomForest(good_movie ~., ntree=500, data = ms.train)
pander(rf.fit.train)
```

Call: randomForest(formula = good_movie ~ ., data = ms.train, ntree = 500) Type of random forest: classification Number of trees: 500 No. of variables tried at each split: 3

OOB estimate of error rate: 14.6349%

Table 16: Confusion Matrix

	No	Yes	class.error
No	2538	93	0.03535
Yes	368	151	0.7091

```
rf.pred.test <- predict(rf.fit.train, newdata = ms.test)
rf.error.test <- table(rf.pred.test, ms.test$good_movie)
# Confusion matrix
pander(rf.error.test)
```

	No	Yes
No	1068	178
Yes	47	57

```
# accuracy rate
sum(diag(rf.error.test))/sum(rf.error.test)
```

```
## [1] 0.8333333
```

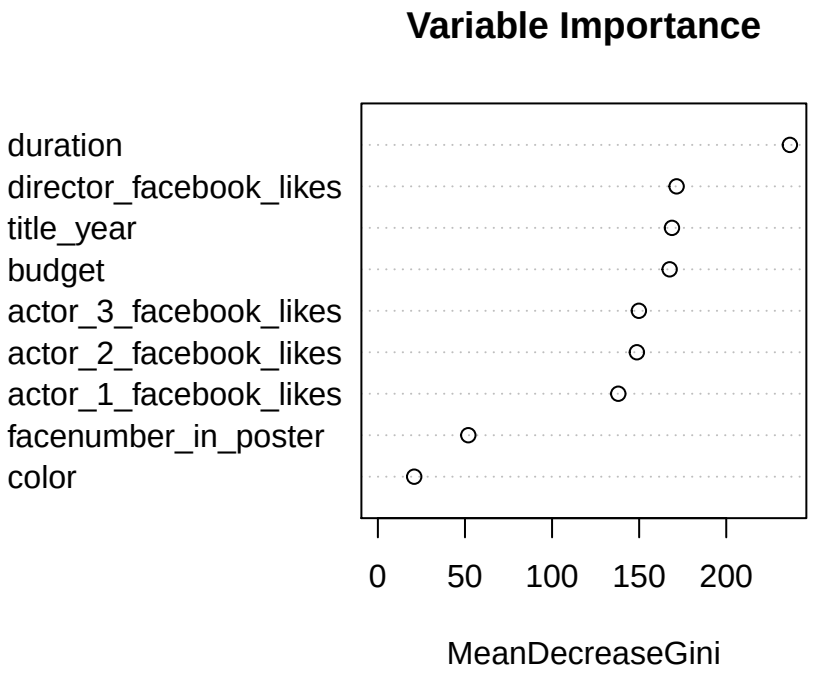
```
# true positive rate
rf.error.test[1,1]/sum(rf.error.test[1,])
```

```
## [1] 0.8571429
```

```
# true negative rate
rf.error.test[2,2]/sum(rf.error.test[2,])
```

```
## [1] 0.5480769
```

```
set.seed(3)
rf.fit <- randomForest(good_movie ~., ntree=500, data = ms)
# Variable Importance
varImpPlot(rf.fit,
           sort = TRUE,
           main="Variable Importance")
```



From the variable importance plot, we see that duration is the strongest factor in determining the quality of a movie. Director/actor/actress facebook popularity and budget are also important factors, although not as strong as duration. The number of faces on the movie poster and color are weaker factors that still need to be taken into account.

I wonder if Wonder Woman will be a good movie

I will use the random forests classifier to determine if Wonder Woman will be a good movie. In order to use this model to predict however, I will need to extract the necessary information beforehand.

With a bit of googling, I can easily find out the required information. The director is Patty Jenkins and her facebook page has 602 facebook likes. The duration of the movie is not yet known, so I will take the average of the other 3 movies in the DC Extended Universe (since Wonder Woman will likely be a similar movie). I find the average duration of Man of Steel (143 minutes), Batman v Superman: Dawn of Justice (151 minutes), and Suicide Squad (123 minutes) to be 139 minutes. According to IMDb, the top 3 actors are Gal Gadot, Robin Wright, and Chris Pine with 8,317,204, 136,556, and 954,997 facebook likes respectively. The movie poster has 1 face in it and the movie will be released in 2017 in color. IMDb tells me the budget is roughly $120,000,000, which is above average for our dataset.

Now that I have all the necessary information for my random forest model, let's see what it tells me!

```
Wonder.Woman.movie <- data.frame("director_facebook_likes" = 602L,
                                 "duration" = 139L,
                                 "actor_1_facebook_likes" = 8317242L,
                                 "actor_2_facebook_likes" = 136556L,
                                 "actor_3_facebook_likes" = 179164L,
                                 "facenumber_in_poster" = 1L,
                                 "title_year" = 2017L,
                                 "color" = "Color",
                                 "budget" = 120000000)
# Indicate that color could be either "Color" or "Black and White"
levels(Wonder.Woman.movie$color) <- levels(ms[1,]$color)

# Predict using the random forest model
```

```
pander(predict(rf.fit, Wonder.Woman.movie, type="prob"))
```

No	Yes
0.642	0.358

When I input the data, the random forest model returns to me that there is a 64% chance that the movie will not have an IMDb score > 7.5, so I conclude that Wonder Woman will likely not be a good movie. Perhaps our model would tell us that Wonder Woman is a good movie if the actors and directors were more well known and the movie was longer.

Conclusion

In this project, I used explored the dataset and used logistic regression and random forests to find out what traits great movies possess, whether it is possible to predict the quality of a movie, and whether Wonder Woman will be a good movie. These are the results: * Duration was the most important factor: great movies tend to be longer * Great movies tend that have fewer faces in the theatrical poster * Number of facebook of director and top three actors is an important factor * The performance of logistic regression and random forests on this dataset are similar * According to my random forests model, Wonder Woman won't be a great movie (i.e. it will have an IMDb score $<$ 0.75) Although there are some trends that great movies share, it is still hard to accurately predict whether or not a movie will be great before it is released. There are many other questions that this dataset can answer. Here are some directions for further work that I am particularly interested in: What genres are popular and does this change through time? Which directors and actors usually work together? Finally, I would like to thank Professor Oh for all the data mining knowledge he has taught me and for helping me navigate through this project.

References

RStudio https://www.rstudio.com/

Dataset (by Chuan Sun): https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset

Kaggle analysis of data set (by Chuan Sun): https://blog.nycdatascience.com/student-works/machine-learning/movie-rating-prediction/

Wonder Woman IMDb: http://www.imdb.com/title/tt0451279/

How to Perform a Logsitic Regression in R (by Michy Alice) https://datascienceplus.com/perform-logistic-regression-in-r/

PSTAT 131 Lecture 7 - Logistic regression (by Professor Oh): https://gauchospace.ucsb.edu/courses/pluginfile.php/991691/mod_resource/content/0/lecture-07.pdf

ggplot2 library help: http://docs.ggplot2.org/0.9.3.1/