Analytical Report on Movies

Each year, Hollywood releases vast numbers of movies to diverse audiences all over the world. Of these scores of films, only a scant few are chosen to win an Academy Award, and fewer still win the Award for Best Picture. How might it be possible to predict in advance whether a film will win an Academy Award for Best Picture? Is it possible to assess the quality of a movie based on its relative probability of winning an Academy Award? Answers to such questions would likely be useful for film critics looking to assess the Oscar chances for the current year's films, as well as for individuals within the Motion Picture Academy itself looking to assess the relative quality of the films presented to them for consideration as an aid in their decision-making process.

Please note that all code used to generate the graphs and output displayed herein is available for review in Appendix D under the appropriate headers.

Initial Preparations

In order to answer these questions, I began by obtaining two datasets containing both general movie data and data indicating whether films were nominated for academy awards. All of these films had at least won an Academy Award for Best Picture. The general movie dataset contains imdb data obtained from Kaggle at https://www.kaggle.com/antoniobap/imdb-v3/data. The academy award data was obtained from the University of Waterloo at https://cs.uwaterloo.ca/~s255khan/oscars.html.

Before loading the datasets, I loaded the dplyr, tidyr and dummies libraries.

Data Cleaning

Preliminary cleaning procedures

Intending to merge these two datasets into one, I began by changing the names of the columns in the imdb dataset (hereinafter, imdb) to match those in the academy awards dataset (hereinafter, acad_awd). I renamed movie_title, title_year and imdb_score to be name, year and rating, respectively.

I then trimmed all leading and trailing whitespace from the names in both datasets and removed trailing question marks from the names in the imdb set. I also changed the names in both datasets to uppercase for consistency. I changed one title in imdb, "Birdman", which used the film's extended title to match the shorter title used in acad_awd. There were two dashes in the nominations column[1] in the acad_awd set that I changed to 1's, signifying that those films had at least won an Oscar for Best Picture.

The na's in certain key columns in imdb, budget, gross and duration, had values imputed to them based on the means of their respective columns as a whole. (The mean for duration was rounded to ensure an integer value.)

A number of duplicated names were observed in the imdb set as well, and these were removed using imdb <- distinct(imdb, name, .keep_all = TRUE). I then subsetted the imdb data for durations greater than 60 in order to eliminate the television shows that were present in the data, and then removed blank names.

The Merge Process

The merge process proved to be fairly complex owing to the different columns in each dataset. In order to prevent loss of data from the acad_awd set, the merge was accomplished in two distinct stages.

[1]The original variable name "nominations" refers to the total number of nominations received by films that won a Best Picture Academy Award. All films listed in the acad_awd dataset were winners of a Best Picture Oscar.

First, a merge_acad_awd set was created, dropping all columns except the name and nominations columns. Following this, I created a merged dataset called movies by performing a full join of merge_acad_awd and imdb by name (movies <- full_join(merge_acad_awd, imdb, by = "name")). This merge was effective except for those movies that were in the acad_awd set, but not in the imdb set. All data except for their names and nominations were effectively lost in the merge. As there were 17 such films (out of a total of 87 acad_awd films), this proved to be an unacceptable loss.

Accordingly, a second separate merge of the 17 films into movies was necessary in order to at least preserve duration, ratings and genre-related data. In order to make the genre-related data in acad_awd for these films compatible with the movies set, I created binaries from the genre1 and genre2 columns using the dummies library. These binaries had to first be extracted into 2 separate dataframes, binary1 and binary2, in order to prevent overwriting in situations where a film was both a comedy and a drama, for example. The two binary sets were then bound together using cbind and the column of blanks, V1, was removed. The combination of the two sets in this manner created duplicate column names such that, for example, Biography could be added to Biography.1 (as in binaries$Biography <- binaries$Biography + binaries$Biography.1) so that any ones would cancel out any zeroes in order to allow for the presence of two genres simultaneously. The duplicate columns were then removed. I then attached this final set of binaries to acad_awd using cbind and called the resulting dataset merge2_acad_awd. I then dropped all extraneous columns from merge2 (genre1, genre2, release, metacritic and synopsis) and limited this dataset to only those films that did not appear in imdb (using merge2_acad_awd <- merge2_acad_awd[which(merge2_acad_awd$name %!in% imdb$name),] and a custom definition, '%!in%' = Negate('%in%')). I then performed the second merge by doing a full join of merge2 with movies, initially into a test dataset called movies2. After examining movies2, I overwrote movies with movies2. This final merge proved to be successful as the maximum amount of data appeared to be preserved.

Final preparations

Following the final merge, I created two subset dataframes based on the age of the films. The first set consisted of all movies between 1927 and 2014 inclusive (the years common to both of the original datasets). Any films in this set having NA's in the nominations column were changed to 0 to signify that they did not win a Best Picture Oscar. The second set consisted only of newer movies produced from 2015-2016, and naturally had no nominations data. This set could be used for a comparison to the older movies to show how trends in filmmaking, such as budgeting and the types of movie genres commonly produced, may have changed over time. Both sets then required a small amount of additional cleaning to remove additional NA's and irrelevant columns (due to lack of relevancy to older films, such as Facebook likes, for example), after which they were ready for analysis.

Data Analysis

Initial Subsetting and Analysis

I began by subsetting the older movie set, consisting of films produced from 1927 through 2014, into another set called won_movies, indicating those movies that had won a Best Picture Academy Award and received other Oscar nominations, thus allowing for a closer analysis of the characteristics of Oscar-winning movies in order to compare them to the set as a whole. As many of the variables were either character variables or binary variables (i.e., the genre variables), the str and summary functions for the sets revealed only scant information concerning the ratings, duration and nominations variables. The maximum number of nominations received by a film was 14. Ratings and duration will be covered in further detail below. Other variables analyzed were the films' language and country of production, their initial budget and amount ultimately grossed at the box office, the varieties and frequencies of film genres and the frequencies of particular directors and actors repersented in the datasets.

Language and Country

I ran tables contrasting language and country for the older movies, won_movies and the new movies sets, using table(won_movies$language, won_movies$country), for example:

```
##
##           China France Germany UK Unknown USA
##   English     1      1       2 12      18  53
```

Both the older and newer films were primarily English, even where they were made in other countries. In the case of the Oscar-winning films, as illustrated above, all of the movies were in English, typically in either the US or the UK[2].

Duration and Ratings

I then ran the mean, median, range, variance (var) and standard deviation (sd) for the durations and ratings of the older, newer and Oscar-winning films. Ratings as used here refer to IMDB ratings given on a scale from 1 (being the worst) to 10 (being the best).

The results for duration indicated that the average length of Oscar-winning movies was longer than the length of other movies in general (approximately 140 minutes for Oscar-winning movies compared to 108 and 107 minutes for older and newer movies, respectively.) There was also a greater variance in won_movies (approximately 1303), but a lower maximum at 240 minutes compared to 330 minutes in the older movies. However, this 330 minute length in the older movies set appears to be an outlier, since the film occupying the 99th percentile in duration had a length of a more modest 188 minutes. The newer movies had the least variance (at approximately 318) and the lowest average length.

Predictably, the results for rating indicated that the Oscar-winning movies were rated higher than the general population or the newer movies, the average and median ratings being approximately 7.8 and 8 respectively compared to 6.4 and 6.6 for the older movies and 6.1 and 6.3 for the newer movies. The Oscar-winning movies also had lower variances than the other sets. (0.43 compared to 1.22 and 1.58 for the older movies and the newer movies.) The maxima for all sets was at or near 9.2, but the minima varied. The older movies and the newer movies had lows of 1.6 and 2.2 respectively, while won_movies had a much greater minimum of 5.8. The smaller variance in won_movies is therefore likely due to those films having been rated consistently high. (The averages for the general population of older movies and newer movies, both here and for duration were substantially the same, with the slightly greater ratings and durations for the older movies set likely because this set also contains the Oscar-winning movies.)

Thus it appears that Oscar-winning films typically have average durations of two to two and a half hours in length with high imdb ratings close to 8 out of 10.

Budget and Gross

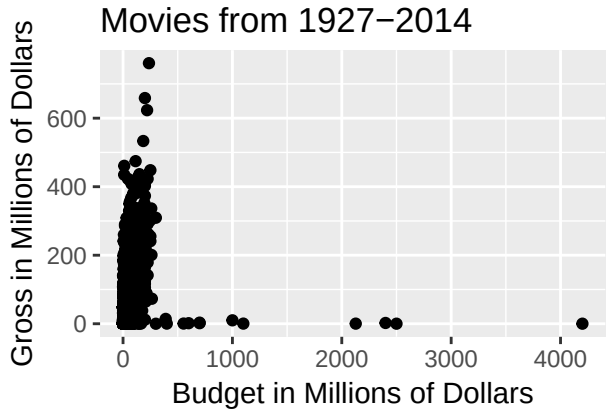
I then ran the mean, median, range, variance (var) and standard deviation (sd) for the gross and budget of the older, newer and Oscar-winning films.

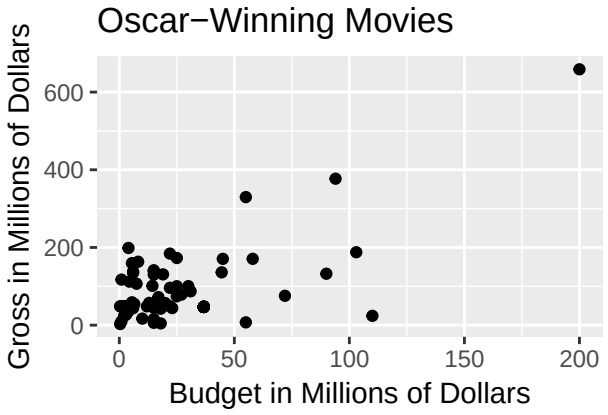
The variances for both gross and budget across all datasets were enormous (on the order of at least 10^{14}), likely due in part to the fact that the dollar figures for both variables are not adjusted for inflation, and the data runs from 1927 through 2014. (There were, however, still large variances of this order within the newer movies data for 2015 and 2016, but this could simply result from the set containing a wide variety of films in terms of budget and distribution. For example, the range of the gross for the newer movies was between $1711 and $652,177,271.) Therefore, I chose to focus more on the relationship between the gross and budget variables.

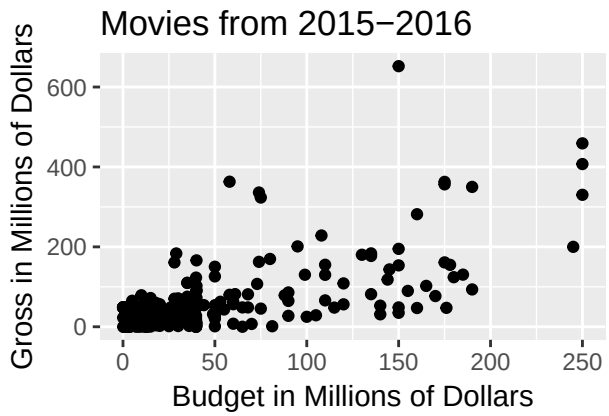
[2]The more extensive tables for the older and newer movies are available in Appendix A, where the first table corresponds to the older films, inclusive of the films from won_movies, and the second set contains the newer films.

The Oscar-winning movies and the newer movies both displayed a fairly positive correlation between a film's budget and the amount it ultimately grossed, with correlations of 0.63 and 0.64, respectively. Thus, the greater the budget for a film the more likely that such films would gross a large amount at the box office.

On the other hand, the general population of older movies showed a much lower correlation between these two variables at only 0.20. In this instance, it did not appear to matter much what a film's initial budget was; most films appeared to gross much more, while movies with very large budgets (over a billion dollars) grossed very little. Perhaps this was due to older filmmakers knowing how to make a quality film for very little money, stretching their dollar as much as possible, while the extremely large budgets were likely indicative of poor planning and organization which would have contributed to the films' overall commercial failure. However, it should again be noted that the financial figures in the older movies dataset were not adjusted for inflation, and therefore likely also contributed the lower correlation, and thus, to the form of the scatterplot for the older movies shown here. The relationships between budget and gross scaled in millions of dollars for all three sets are displayed below.







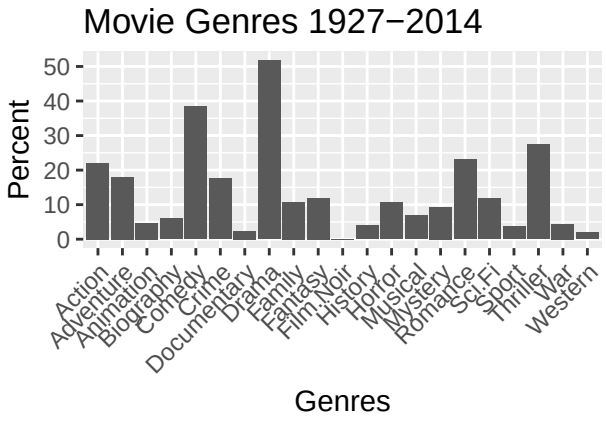
Therefore, it appears that Oscar-winning films typically have budgets not much greater than $100 million and gross not much more than $200 million.

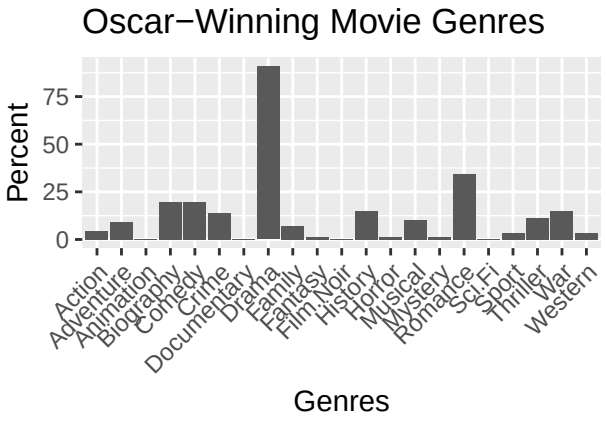
Genres

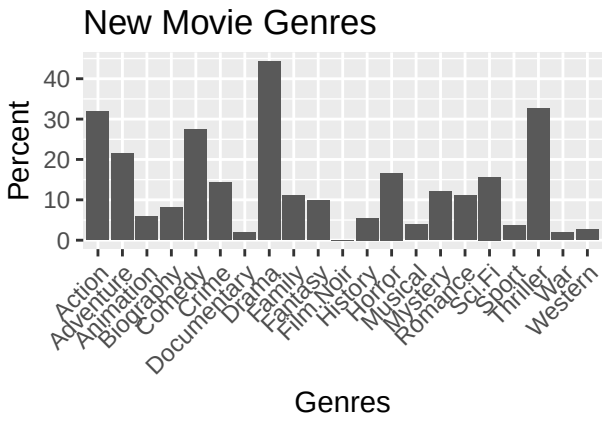
There were 21 total genres that were counted in each dataset: Action, Adventure, Biography, Comedy, Crime, Drama, Musical, Western, Family, Film Noir, History, Mystery, Romance, Sport, War, Fantasy, Sci Fi, Thriller, Documentary, Horror and Animation. It should be noted, however, that the Film Noir category was not actually represented in the data, despite its being given its own genre category by the creator of the dataset. It may be that those movies falling into this genre were subsumed into other genres in the course of the dataset's creation and the category was never removed.

The most common genre in every case was Drama. The least common genres, aside from Film Noir, were

Westerns for the general movie population from 1927-2014, Sci Fi, Documentary, Animation, Fantasy, Horror and Mystery for the Oscar-winning movies and Documentary for the newer movies. Plots showing the relative percentages of each genre in greater detail are below.







As will be observed, the Oscar-winning movies are dominated primarily by Drama and Romance movies. This is also the case for the movies from 1927-2014, but with greater representation from Comedies, Thrillers and Action and Adventure films. The influence of changing public tastes may be seen in the New Movies, where Thrillers, Action, Adventure, Sci Fi and Horror (less conventionally "dramatic" genres) are more common than in prior years. Additionally, Animation is a relatively new category at the Academy Awards, thus accounting for its absence from the Oscar-winning dataset.

Directors and Actors

I then analyzed each of the datasets for the most frequently occuring directors and actors. In the general dataset for Movies from 1927-2014 I created a subset of only movie names and director names called director1 and from there a list called dir_count1 using count(director1, director1$director_name). I then obtained the director name with the maximum count using dir_count1$‘director1$director_name‘[dir_count1$n == max(dir_count1$n)] and the directors occupying above the 99th percentile in frequency using dir_count1$‘director1$director_name‘[dir_count1$n > quantile(dir_count1$n, .99)]. This revealed Steven Spielberg as being the most popular director here, while 16 other directors occupied positions above the 99th percentile. These other top directors were Barry Levinson, Brian De Palma, Clint Eastwood, Joel Schumacher, Martin Scorsese, Oliver Stone, Renny Harlin, Ridley Scott, Robert Rodriguez, Robert Zemeckis, Ron Howard, Spike Lee, Steven Soderbergh, Tim Burton, Tony Scott and Woody Allen.

Using a similar procedure for the Oscar-winning movies and newer movies, I found that the top directors for the Oscar-winning movies (all tied for the maximum) were Billy Wilder, Clint Eastwood, David Lean, Elia Kazan, Francis Ford Coppola, Frank Capra, Fred Zinnemann and Milos Forman. Tied for maximum

frequency among the directors of the newer movies were David O. Russell, James Wan, Jaume Collet-Serra, Patricia Riggen, Robert Schwentke, Roland Emmerich and, once again, Steven Spielberg. It thus appears that Spielberg is popular among audiences new and old, but less so among those who decide the Best Picture Oscar winners at the Motion Picture Academy (although Spielberg did direct one Oscar-winning movie in the dataset, "Schindler's List").

In order to analyze the frequency of the actors, I had to stack the three separate actor columns in each dataset (actor_1_name, actor_2_name and actor_3_name) to form a single column outside the datasets using for the first dataset, for example, actors2 <- stack(list(act1 = won_movies$actor_1_name, act2 = won_movies$actor_2_name, act3 = won_movies$actor_3_name)) and removing those entries that were unknown using actors2 <- subset(actors2, actors2$values != "Unknown"). I then used the same procedures as above with the directors in order to obtain the actors with the maximum counts and the actors occupying positions above the 99th percentile in frequency. (The exception was the Oscar-winning actors, where I used the 90th percentile since percentiles above the 90th percentile were the same as the maximum).

The most common actor for the movies from 1927-2014 was Robert De Niro. 54 other actors had frequencies above the 99th percentile, appearing in more than 18 films per person. The most common actor among the actors in the Oscar-winning films was Morgan Freeman, although Robert De Niro is in the 90th percentile of top actors. The other top (90th percentile) actors in the Oscar-winning movies were Al Pacino, Anthony Hopkins, Beth Grant, Claude Rains, Clint Eastwood, Colin Firth, Jack Hawkins, John Gielgud, Judd Hirsch, Karl Malden, Kate Winslet, Leonardo DiCaprio, Marlon Brando, Meryl Streep, Oliver Reed, Ray Walston, Robert Duvall, Robert Shaw, Scoot McNairy and Susannah York. Finally, the most frequent actors appearing in the new movies were Chris Hemsworth and, once again, Robert De Niro, while the other actors appearing above the 99th percentile in frequency were Bradley Cooper, Johnny Depp, Scarlett Johansson and Tom Hardy.

It should be noted here that the compiler of these datasets did not always include the most important actors in a given film, for example, leaving out Arnold Schwarzenegger and Sharon Stone from "Total Recall". This may account for the presence of actors here such as "Scoot McNairy" in the Oscar-winning dataset, a lesser known actor who did indeed appear in the Oscar-winning movies "12 Years a Slave" and "Argo". Thus, the actors variables may have limited predictive utility for those films that included lesser-known actors at the expense of omitting the main actors, although this did not occur in every case.

Summary of Statistical Analysis Results

Based on my examination of the characteristics of the films in the movie datasets, it appears that a number of factors characterize successful potentially Oscar-winning movies. Drama films made in English in either the US or UK that are moderately long (approximately two to two and a half hours in length), with high imdb ratings (nearly 8 out of 10) and relatively large budgets and high box office values (budgets not much greater than $100 million and grossing not much more than $200 million) appear to be films that are most likely to win Academy Awards for Best Picture.

While the caliber of directors and actors involved in a film's production may be persuasive evidence that a movie will turn out to be successful, it is not determinative. For example, while Steven Spielberg was involved in the production of an Oscar-winning film, he was also involved in many that were not. Likewise, Morgan Freeman (the most frequently appearing actor among the Oscar-winning films) acted not only in Oscar-winning films such as "Unforgiven", but also in less successful films such as "Evan Almighty" and "Ted 2".

Accordingly, I would use variables such as duration, language, country, rating, budget, gross and the genre variables as predictor variables when attempting to make predictions as to whether a film is likely to win an Academy Award for Best Picture, rather than looking to the director and/or actors involved in the film's production. Such variables are likely to also prove useful in assessing the overall quality of a film, assuming that the subjective nature of "quality" may be quantitatively defined as, for example, the number of other awards that a film won or the number of other awards for which they were nominated. These ideas will be further put to the test in my Regression Analysis report.

Regression Analysis

Although the nominations variable has numbers indicating the number of awards for which a movie was nominated, all of the movies from the original acad_awd dataset were Best Picture Oscar winners. Therefore, the real issue here is simply whether a movie won or not. The number of awards won is irrelevant. Accordingly, after creating the training set and testing set, I converted nominations to a simple binary "Yes/No" variable.

Since nominations, now binary, is the response variable, I chose logistic regression as the method for making predictions here. It remained only to choose the best predictor variables for the model.

Constructing a Model

I began by uniting the previously divided datasets (based on the year the films were released) into a single dataset called all_movies. I then divided this set again for training and testing purposes; 80% of all_movies became a training subset called movies_train, while the remaining 20% was set aside as the testing set called movies_test. The randomization seed was set to 123 to ensure reproducibility.

I then created a number of preliminary models with different combinations of predictor variables and assessed each one to determine which was optimal. The variables that appeared to be of primary significance were duration (length of a film measured in minutes), gross and rating (a film's imdb rating, on a scale of 1 to 10, with 10 being the highest). Based on my prior statistical analysis of the movies dataset that indicated the prevalence of some genres over others among Oscar-winning movies, I also determined that the genres should also be included as predictor variables in the model.

After proceeding through a number of models both with and without the genre variables and one with an interaction term between gross and budget that proved to be of little significance, I ultimately settled upon two models. The first model, called movies.mod7, included duration, gross, rating, the genre variables as well as a language variable (indicating the language in which the movie was made). The language variable was not significant in itself, but appeared to have positive synergistic effects on the model in conjunction with the other variables. Movies.mod7 had one of the lowest residual deviance scores (at 408.53) among the other models with high significance scores for duration, gross and particularly rating, with a probability score ($\Pr(>|z|)$) on the order of 10^{-16}.

I then ran a forward step model (called fwd_model) on the variables duration, gross, rating, budget, language, country and the genre variables to find an optimal model. The step model ultimately chose as predictor variables rating, country, duration, budget, gross, country (the country in which the film was produced), and the genre variables Drama, Mystery, Sci.Fi, Animation, Romance, Fantasy, Documentary, Horror, War and Family. Rating remained the most significant variable, though not as significant as in movies.mod7. $\Pr(>|z|)$ was on the order of 10^{-13}, rather than 10^{-16} for rating, while budget, gross and duration also appeared highly significant as well. The residual deviance for fwd_model was much lower, however, at 304.01 (a difference of 104.49) as revealed by running the anova function on both models. As these two models appeared to be the strongest, I decided to evaluate them both further.

Evaluating the models

I ran the predict functions on both models over the training set with an initial threshold of .5 and then ran ROC curves for both using the pROC package. The optimal thresholds were shown to be 0.045 and 0.055 with Areas Under the Curve (AUC) of 0.947 and 0.971, respectively. I then re-ran the predictions using these new optimal thresholds and obtained the following confusion matrices. (The coefficients and probabilities for each of the variables can be viewed in Appendix B here [3].)

```
##
## pred1_train   No  Yes
```

[3]The first set of coefficents and probabilities corresponds to movies.mod7, while the second set corresponds to fwd_model.

```
##         No  3243   12
##         Yes  288   61
```

```
##
## pred2_train   No  Yes
##         No  3353    5
##         Yes  178   68
```

There was a slightly lower overall accuracy for the first model (91.7% vs. 94.9%), with higher False Negative and higher False Positive rates in the first model. Movies.mod7 had a Sensitivity of 83.6%, a Specificity of 91.8%, a False Negative rate of 16.4% and a False Positive rate of 8.2%, while fwd_model had a Sensitivity of 93.2%, a Specificity of 95.0%, a False Negative rate of 6.8% and a False Positive rate of 5.0%.

I then ran the predict functions on both models over the test set with an initial threshold of .5 and then ran ROC curves for both using the pROC package. The optimal thresholds were shown to be 0.034 and 0.046 with Areas Under the Curve (AUC) of 0.867 and 0.853, respectively, thus showing a lower AUC under the test set for fwd_model. (All ROC curves may be viewed in Appendix C [4].) I then re-ran the predictions using these new optimal thresholds. Initially, it appeared that fwd_model was superior in most respects, but this changed once applying the predctions to the test set. The confusion matrices for the test set were as follows (using the new optimal thresholds of 0.034 and 0.046, respectively):

```
##
## pred1  No Yes
##   No  774   2
##   Yes  99  12
```

```
##
## pred2  No Yes
##   No  812   4
##   Yes  61  10
```

There was again a lower overall accuracy for the first of the models (88.6% vs. 92.7%, respectively). For movies.mod7 there was a Sensitivity of 85.7%, a Specificity of 88.7%, a False Negative rate of 14.3% and a False Positive rate of 11.3%, while for fwd_model there was a Sensitivity of 71.4%, a Specificity of 93.0%, a False Negative rate of 28.6% and a False Positive rate of 7.0%. Thus, in this instance movies.mod7 had a higher Sensitivity and lower False Negative rate than fwd_model. (However, it should be noted that the proportion of positives ("Yes" values for nominations) in the test set population was lower than that in the training set population, 1.6% vs 2.0% respectively, so the probability of getting a "Yes" hit for nominations was greater in the training set population.) Therefore, although fwd_model appeared to be superior in most respects, the first model may be preferred for this dataset where greater sensitivity is desired. It was for this reason that I ultimately chose movies.mod7 to test its ability to make predictions based on specific movies contained in the test set.

Applying the model

I chose two films from the dataset that both had equal values for rating in order to control for the strength of the rating variable by itself. The films I chose were "The Artist", a film I believed could be Oscar-worthy, and "Casino Royale", a film I doubted would be Oscar-worthy. Both films had imdb ratings of 8, just above the average rating of 7.8 for Oscar-winning films. Applying movie.mod7 to a dataframe containing these films' values for the variables in the model, I obtained the following:

```
##               name        fit     se.fit rating
## 22      THE ARTIST 0.14514413 0.05291039      8
## 1475 CASINO ROYALE 0.03848228 0.03382990      8
```

[4] The first two ROC curves correspond to the curves for movies.mod7 and fwd_model, respectively, on the training set, while the second two curves correspond to the curves for movies.mod7 and fwd_model, respectively, on the test set.

As can be seen, "The Artist" has a much higher likelihood of winning (approximately 14.5% vs. 3.8%, with very small standard errors), despite both movies having imdb ratings of 8 [5]. As revealed by the imdb profile for both films available on imdb.com, "The Artist" actually won the Best Picture Oscar for 2012, and was nominated for many other awards, including a Golden Globe. "Casino Royale" won a Saturn for Best Action/Adventure/Thriller film, a BAFTA award for Best Sound and was nominated for a number of other awards, but did not win an Oscar or a Golden Globe. Therefore, imdb rating is not the only determining factor in predicting whether a film will win a Best Picture Oscar. Indeed, given the low predictive threshold of .034 used here, this model correctly predicted that "The Artist" would win a Best Picture Oscar and was also able to make (or at least confirm) a "subjective" assessment as to the quality of both films, if such quality can be measured by the relative probability of winning a Best Picture Oscar and by the other awards won by each film. On the other hand, the model made a Type I error in predicting that "Casino Royale" would also win a Best Picture Oscar, although this error was very small. The probability that "Casino Royale" would win was approximately 3.8%, while the prediction threshold used here was 3.4%, so the film's probability of winning was really borderline here as it was only over the threshold by less than 1%.

A likely explanation for this small Type I error is that certain key predictor variables had values for "Casino Royale" that were more characteristic of Oscar-winning films. For example, in addition to having a high imdb rating, the duration (normally a fairly strong predictor) for "Casino Royale" was 144 minutes, just slightly above the average of 140 minutes for Oscar-winning films. Additionally, "Casino Royale" grossed approximately $167 million, higher than the average Oscar-winning gross of approximately $84 million, but within the upper quartile for such films (between approximately $100 million and $377 million, excluding an extreme outlier). Nevertheless, the film was correctly predicted to have a much lower probability of winning a Best Picture Oscar than "The Artist". Therefore, this model appears to be accurate on the whole, but may be overly sensitive where a particular film has values close to characteristic values for Oscar-winning films for one or more strong predictor variables [6].

Concluding Remarks and Recommendations

While the current dataset and model appear to be accurate on the whole, there may be some areas for further investigation and improvement. First, higher quality data could be obtained that adjusts all financial variables for inflation. This data should also include improved actor information, such that the three actors listed for any given film are all principal actors in the film, rather than just any three actors that appeared in the film. With improved actor data, it may be possible to make more accurate predictions using the director and actor variables on a subset of the overall movies dataset that looks only at the most frequently appearing directors and actors over a particular span of years. (For example, Cary Grant likely appears frequently, but should not be used in a span of years from 1985 through the present.) Another potential avenue for investigation may be in parsing out the most frequent of the plot keywords for use as predictor variables.

When using this dataset and model on a new movie for which one intends to evaluate its Oscar potential, look up the film on imdb.com and obtain the following pieces of information: the film's imdb rating, its length in minutes, the amount in dollars the film grossed at the box office, the language in which the film was made, and all of the film's genre categories (i.e., whether the film is an Action, Adventure, Biography, Comedy, Crime, Drama, Musical, Western, Family, History, Mystery, Romance, Sport, War, Fantasy, Sci.Fi, Thriller, Documentary, Horror or Animation). Once this information is obtained, construct a dataframe with each

[5] It should be noted here that fwd_model produced incorrect results, in part because the model left out key genre variables (Action, Adventure, Thriller, Comedy). It showed both films with very low probabilities of winning, with "Casino Royale" at a much higher probability than "The Artist" (approximately 1% vs. less than 1 millionth of 1%). This model also includes budget in addition to gross, both of which are non-inflation adjusted variables. In addition to having a below average gross compared to Oscar-winning films, "The Artist" also had a below average budget (approximately $15 million compared to approximately $25 million for Oscar-winning films.) It therefore appears that more stable models should contain all of the genre variables, as the first model does, and fewer high variance predictor variables. The first model, movies.mod7, therefore appears to be more stable when exposed to different types of data than fwd_model.

[6] In this case, likely the high rating in conjunction with a duration near the average Oscar-winning length and, perhaps, the high gross amount.

one of these pieces of information as a single column [7]. Following this, run the predict function over this dataframe using the movies.mod7 model [8]. The output will be a decimal value indicating the probability that the film in question will win an Academy Award for Best Picture. If this number is greater than the threshold value chosen in advance (in the preceding example, the optimal threshold was found to be .034)[9], this indicates that the movie in question is likely to win a Best Picture Oscar. If the same information for two or more movies are inserted into the dataframe, the movies may be compared as to their relative probabilities of winning a Best Picture Oscar.

Therefore, using this model as suggested it may be possible to predict in advance whether a film will win an Academy Award for Best Picture and to assess its relative quality[10].

Appendix A

```
##
##              Afghanistan Argentina Aruba Australia Bahamas Belgium Brazil
##   Aboriginal           0         0     0         1       0       0      0
##   Arabic               0         0     0         0       0       0      0
##   Aramaic              0         0     0         0       0       0      0
##   Bosnian              0         0     0         0       0       0      0
##   Cantonese            0         0     0         0       0       0      0
##   Chinese              0         0     0         0       0       0      0
##   Czech                0         0     0         0       0       0      0
##   Danish               0         0     0         0       0       0      0
##   Dari                 1         0     0         0       0       0      0
##   Dutch                0         0     0         0       0       0      0
##   Dzongkha             0         0     0         1       0       0      0
##   English              0         0     1        47       1       2      1
##   Filipino             0         0     0         0       0       0      0
##   French               0         0     0         0       0       0      0
##   German               0         0     0         0       0       0      0
##   Greek                0         0     0         0       0       0      0
##   Hebrew               0         0     0         0       0       0      0
##   Hindi                0         0     0         0       0       0      0
##   Hungarian            0         0     0         0       0       0      0
##   Icelandic            0         0     0         0       0       0      0
##   Indonesian           0         0     0         0       0       0      0
##   Italian              0         0     0         0       0       0      0
##   Japanese             0         0     0         0       0       0      0
##   Kannada              0         0     0         0       0       0      0
##   Kazakh               0         0     0         0       0       0      0
##   Korean               0         0     0         0       0       0      0
##   Mandarin             0         0     0         0       0       0      0
##   Maya                 0         0     0         0       0       0      0
##   Mongolian            0         0     0         0       0       0      0
```

[7] Use the following template code to accomplish this, filling in the appropriate values within the parentheses following the c after each variable: new_movie_data <- data.frame(duration = c(), rating = c(), gross = c(), language = c(), Action = c(), Adventure = c(), Biography = c(), Comedy = c(), Crime = c(), Drama = c(), Musical = c(), Western = c(), Family = c(), History = c(), Mystery = c(), Romance = c(), Sport = c(), War = c(), Fantasy = c(), Sci.Fi = c(), Thriller = c(), Documentary = c(), Horror = c(), Animation = c())

[8] Use the following code to run the predict function: predict(movies.mod7, newdata = new_movie_data, type = "response")

[9] Please note that the lower the threshold chosen, the greater the sensitivity of the model. However, the False Positive rate will also increase as a result.

[10] Where such quality is measured by the relative probability of winning a Best Picture Oscar and by the other awards won by each film, verifiable on imdb.com

```
##   None                0         0     0         0       0       0       0
##   Norwegian           0         0     0         0       0       0       0
##   Persian             0         0     0         0       0       0       0
##   Polish              0         0     0         0       0       0       0
##   Portuguese          0         0     0         0       0       0       6
##   Romanian            0         0     0         0       0       0       0
##   Russian             0         0     0         0       0       0       0
##   Spanish             0         4     0         0       0       0       0
##   Swedish             0         0     0         0       0       0       0
##   Thai                0         0     0         0       0       0       0
##   Vietnamese          0         0     0         0       0       0       0
##   Zulu                0         0     0         0       0       0       0
##
##              Bulgaria Cambodia Cameroon Canada China Colombia
##   Aboriginal        0        0        0      0     0        0
##   Arabic            0        0        0      0     0        0
##   Aramaic           0        0        0      0     0        0
##   Bosnian           0        0        0      0     0        0
##   Cantonese         0        0        0      0     1        0
##   Chinese           0        0        0      0     2        0
##   Czech             0        0        0      0     0        0
##   Danish            0        0        0      0     0        0
##   Dari              0        0        0      0     0        0
##   Dutch             0        0        0      0     0        0
##   Dzongkha          0        0        0      0     0        0
##   English           1        1        1    105     3        0
##   Filipino          0        0        0      0     0        0
##   French            0        0        0      6     0        0
##   German            0        0        0      0     0        0
##   Greek             0        0        0      0     0        0
##   Hebrew            0        0        0      0     0        0
##   Hindi             0        0        0      1     0        0
##   Hungarian         0        0        0      0     0        0
##   Icelandic         0        0        0      0     0        0
##   Indonesian        0        0        0      0     0        0
##   Italian           0        0        0      0     0        0
##   Japanese          0        0        0      0     0        0
##   Kannada           0        0        0      0     0        0
##   Kazakh            0        0        0      0     0        0
##   Korean            0        0        0      0     0        0
##   Mandarin          0        0        0      0    18        0
##   Maya              0        0        0      0     0        0
##   Mongolian         0        0        0      0     0        0
##   None              0        0        0      1     0        0
##   Norwegian         0        0        0      0     0        0
##   Persian           0        0        0      0     0        0
##   Polish            0        0        0      0     0        0
##   Portuguese        0        0        0      0     0        0
##   Romanian          0        0        0      0     0        0
##   Russian           0        0        0      0     0        0
##   Spanish           0        0        0      0     0        1
##   Swedish           0        0        0      0     0        0
##   Thai              0        0        0      0     0        0
##   Vietnamese        0        0        0      0     0        0
```

```
##   Zulu              0        0        0      0     0        0
##
##              Czech Republic Denmark Dominican Republic Egypt Finland
##   Aboriginal               0       0                  0     0       0
##   Arabic                   0       0                  0     1       0
##   Aramaic                  0       0                  0     0       0
##   Bosnian                  0       0                  0     0       0
##   Cantonese                0       0                  0     0       0
##   Chinese                  0       0                  0     0       0
##   Czech                    1       0                  0     0       0
##   Danish                   0       4                  0     0       0
##   Dari                     0       0                  0     0       0
##   Dutch                    0       0                  0     0       0
##   Dzongkha                 0       0                  0     0       0
##   English                  1       6                  0     0       0
##   Filipino                 0       0                  0     0       0
##   French                   0       0                  0     0       1
##   German                   0       0                  0     0       0
##   Greek                    0       0                  0     0       0
##   Hebrew                   0       0                  0     0       0
##   Hindi                    0       0                  0     0       0
##   Hungarian                0       0                  0     0       0
##   Icelandic                0       0                  0     0       0
##   Indonesian               0       0                  0     0       0
##   Italian                  0       0                  0     0       0
##   Japanese                 0       0                  0     0       0
##   Kannada                  0       0                  0     0       0
##   Kazakh                   0       0                  0     0       0
##   Korean                   0       0                  0     0       0
##   Mandarin                 0       0                  0     0       0
##   Maya                     0       0                  0     0       0
##   Mongolian                0       0                  0     0       0
##   None                     0       0                  0     0       0
##   Norwegian                0       0                  0     0       0
##   Persian                  0       0                  0     0       0
##   Polish                   0       0                  0     0       0
##   Portuguese               0       0                  0     0       0
##   Romanian                 0       0                  0     0       0
##   Russian                  0       0                  0     0       0
##   Spanish                  0       0                  1     0       0
##   Swedish                  0       0                  0     0       0
##   Thai                     0       0                  0     0       0
##   Vietnamese               0       0                  0     0       0
##   Zulu                     0       0                  0     0       0
##
##              France Georgia Germany Greece Hong Kong Hungary Iceland India
##   Aboriginal      0       0       0      0         0       0       0     0
##   Arabic          1       0       1      0         0       0       0     0
##   Aramaic         0       0       0      0         0       0       0     0
##   Bosnian         0       0       0      0         0       0       0     0
##   Cantonese       0       0       0      0         8       0       0     0
##   Chinese         0       0       0      0         0       0       0     0
##   Czech           0       0       0      0         0       0       0     0
##   Danish          0       0       0      0         0       0       0     0
```

```
##   Dari            0       0       0       0         0       0       0     0
##   Dutch           0       0       0       0         0       0       0     0
##   Dzongkha        0       0       0       0         0       0       0     0
##   English        80       1      77       0         5       1       1     5
##   Filipino        0       0       0       0         0       0       0     0
##   French         60       0       0       0         0       0       0     0
##   German          0       0      14       0         0       0       0     0
##   Greek           0       0       0       1         0       0       0     0
##   Hebrew          0       0       0       0         0       0       0     0
##   Hindi           0       0       0       0         0       0       0    25
##   Hungarian       0       0       0       0         0       1       0     0
##   Icelandic       0       0       0       0         0       0       1     0
##   Indonesian      0       0       0       0         0       0       0     0
##   Italian         0       0       0       0         0       0       0     0
##   Japanese        0       0       0       0         0       0       0     0
##   Kannada         0       0       0       0         0       0       0     1
##   Kazakh          1       0       0       0         0       0       0     0
##   Korean          0       0       0       0         0       0       0     0
##   Mandarin        0       0       0       0         3       0       0     0
##   Maya            0       0       0       0         0       0       0     0
##   Mongolian       0       0       0       0         0       0       0     0
##   None            0       0       0       0         0       0       0     0
##   Norwegian       0       0       0       0         0       0       0     0
##   Persian         1       0       0       0         0       0       0     0
##   Polish          0       0       0       0         0       0       0     0
##   Portuguese      0       0       0       0         0       0       0     0
##   Romanian        0       0       0       0         0       0       0     0
##   Russian         0       0       0       0         0       0       0     0
##   Spanish         0       0       0       0         0       0       0     0
##   Swedish         0       0       0       0         0       0       0     0
##   Thai            0       0       0       0         0       0       0     0
##   Vietnamese      0       0       0       0         0       0       0     0
##   Zulu            0       0       0       0         0       0       0     0
##
##              Indonesia Iran Ireland Israel Italy Japan Kyrgyzstan Libya
##   Aboriginal         0    0       0      0     0     0          0     0
##   Arabic             0    0       0      0     0     0          0     0
##   Aramaic            0    0       0      0     0     0          0     0
##   Bosnian            0    0       0      0     0     0          0     0
##   Cantonese          0    0       0      0     0     0          0     0
##   Chinese            0    0       0      0     0     0          0     0
##   Czech              0    0       0      0     0     0          0     0
##   Danish             0    0       0      0     0     0          0     0
##   Dari               0    0       0      0     0     0          0     0
##   Dutch              0    0       0      0     0     0          0     0
##   Dzongkha           0    0       0      0     0     0          0     0
##   English            0    1      11      0    12     4          1     1
##   Filipino           0    0       0      0     0     0          0     0
##   French             0    0       0      0     0     0          0     0
##   German             0    0       0      0     0     0          0     0
##   Greek              0    0       0      0     0     0          0     0
##   Hebrew             0    0       0      3     0     0          0     0
##   Hindi              0    0       0      0     0     0          0     0
##   Hungarian          0    0       0      0     0     0          0     0
```

```
##   Icelandic          0    0       0      0     0     0          0     0
##   Indonesian         1    0       0      0     0     0          0     0
##   Italian            0    0       0      0     9     0          0     0
##   Japanese           0    0       0      0     0    14          0     0
##   Kannada            0    0       0      0     0     0          0     0
##   Kazakh             0    0       0      0     0     0          0     0
##   Korean             0    0       0      0     0     0          0     0
##   Mandarin           0    0       0      0     0     0          0     0
##   Maya               0    0       0      0     0     0          0     0
##   Mongolian          0    0       0      0     0     0          0     0
##   None               0    0       0      0     0     0          0     0
##   Norwegian          0    0       0      0     0     0          0     0
##   Persian            0    3       0      0     0     0          0     0
##   Polish             0    0       0      0     0     0          0     0
##   Portuguese         0    0       0      0     0     0          0     0
##   Romanian           0    0       0      0     0     0          0     0
##   Russian            0    0       0      0     0     0          0     0
##   Spanish            0    0       0      0     0     0          0     0
##   Swedish            0    0       0      0     0     0          0     0
##   Thai               0    0       0      0     0     0          0     0
##   Vietnamese         0    0       0      0     0     0          0     0
##   Zulu               0    0       0      0     0     0          0     0
##
##              Mexico Netherlands New Line New Zealand Nigeria Norway Peru
##   Aboriginal      0           0        0           0       0      0    0
##   Arabic          0           0        0           0       0      0    0
##   Aramaic         0           0        0           0       0      0    0
##   Bosnian         0           0        0           0       0      0    0
##   Cantonese       0           0        0           0       0      0    0
##   Chinese         0           0        0           0       0      0    0
##   Czech           0           0        0           0       0      0    0
##   Danish          0           0        0           0       0      0    0
##   Dari            0           0        0           0       0      0    0
##   Dutch           0           4        0           0       0      0    0
##   Dzongkha        0           0        0           0       0      0    0
##   English         1           1        1          11       1      3    1
##   Filipino        0           0        0           0       0      0    0
##   French          0           0        0           0       0      0    0
##   German          1           0        0           0       0      0    0
##   Greek           0           0        0           0       0      0    0
##   Hebrew          0           0        0           0       0      0    0
##   Hindi           0           0        0           0       0      0    0
##   Hungarian       0           0        0           0       0      0    0
##   Icelandic       0           0        0           0       0      0    0
##   Indonesian      0           0        0           0       0      0    0
##   Italian         0           0        0           0       0      0    0
##   Japanese        0           0        0           0       0      0    0
##   Kannada         0           0        0           0       0      0    0
##   Kazakh          0           0        0           0       0      0    0
##   Korean          0           0        0           0       0      0    0
##   Mandarin        0           0        0           0       0      0    0
##   Maya            0           0        0           0       0      0    0
##   Mongolian       0           0        0           0       0      0    0
##   None            0           0        0           0       0      0    0
```

```
##   Norwegian       0          0        0          0       0      4    0
##   Persian         0          0        0          0       0      0    0
##   Polish          0          0        0          0       0      0    0
##   Portuguese      0          0        0          0       0      0    0
##   Romanian        0          0        0          0       0      0    0
##   Russian         0          0        0          0       0      0    0
##   Spanish        12          0        0          0       0      0    0
##   Swedish         0          0        0          0       0      0    0
##   Thai            0          0        0          0       0      0    0
##   Vietnamese      0          0        0          0       0      0    0
##   Zulu            0          0        0          0       0      0    0
##
##              Philippines Poland Romania Russia Slovakia South Africa
##   Aboriginal           0      0       0      0        0            0
##   Arabic               0      0       0      0        0            0
##   Aramaic              0      0       0      0        0            0
##   Bosnian              0      0       0      0        0            0
##   Cantonese            0      0       0      0        0            0
##   Chinese              0      0       0      0        0            0
##   Czech                0      0       0      0        0            0
##   Danish               0      0       0      0        0            0
##   Dari                 0      0       0      0        0            0
##   Dutch                0      0       0      0        0            0
##   Dzongkha             0      0       0      0        0            0
##   English              1      1       2      1        1            7
##   Filipino             0      0       0      0        0            0
##   French               0      0       0      0        0            0
##   German               0      0       0      0        0            0
##   Greek                0      0       0      0        0            0
##   Hebrew               0      0       0      0        0            0
##   Hindi                0      0       0      0        0            0
##   Hungarian            0      0       0      0        0            0
##   Icelandic            0      0       0      0        0            0
##   Indonesian           0      0       0      0        0            0
##   Italian              0      0       0      0        0            0
##   Japanese             0      0       0      0        0            0
##   Kannada              0      0       0      0        0            0
##   Kazakh               0      0       0      0        0            0
##   Korean               0      0       0      0        0            0
##   Mandarin             0      0       0      0        0            0
##   Maya                 0      0       0      0        0            0
##   Mongolian            0      0       0      1        0            0
##   None                 0      0       0      0        0            0
##   Norwegian            0      0       0      0        0            0
##   Persian              0      0       0      0        0            0
##   Polish               0      1       0      0        0            0
##   Portuguese           0      0       0      0        0            0
##   Romanian             0      0       1      0        0            0
##   Russian              0      0       0      8        0            0
##   Spanish              0      0       0      0        0            0
##   Swedish              0      0       0      0        0            0
##   Thai                 0      0       0      0        0            0
##   Vietnamese           0      0       0      0        0            0
##   Zulu                 0      0       0      0        0            1
```

```
##
##              South Korea Soviet Union Spain Sweden Switzerland Taiwan
##   Aboriginal           0            0     0      0           0      0
##   Arabic               0            0     0      0           0      0
##   Aramaic              0            0     0      0           0      0
##   Bosnian              0            0     0      0           0      0
##   Cantonese            0            0     0      0           0      0
##   Chinese              0            0     0      0           0      0
##   Czech                0            0     0      0           0      0
##   Danish               0            0     0      0           0      0
##   Dari                 0            0     0      0           0      0
##   Dutch                0            0     0      0           0      0
##   Dzongkha             0            0     0      0           0      0
##   English              5            0    18      0           2      0
##   Filipino             0            0     0      0           0      0
##   French               0            0     0      0           0      0
##   German               0            0     0      0           1      0
##   Greek                0            0     0      0           0      0
##   Hebrew               0            0     0      0           0      0
##   Hindi                0            0     0      0           0      0
##   Hungarian            0            0     0      0           0      0
##   Icelandic            0            0     0      0           0      0
##   Indonesian           0            0     0      0           0      0
##   Italian              0            0     0      0           0      0
##   Japanese             0            0     0      0           0      0
##   Kannada              0            0     0      0           0      0
##   Kazakh               0            0     0      0           0      0
##   Korean               6            0     0      0           0      0
##   Mandarin             0            0     0      0           0      1
##   Maya                 0            0     0      0           0      0
##   Mongolian            0            0     0      0           0      0
##   None                 0            0     0      0           0      0
##   Norwegian            0            0     0      0           0      0
##   Persian              0            0     0      0           0      0
##   Polish               0            0     0      0           0      0
##   Portuguese           0            0     0      0           0      0
##   Romanian             0            0     0      0           0      0
##   Russian              0            1     0      1           0      0
##   Spanish              0            0    13      0           0      0
##   Swedish              0            0     0      4           0      0
##   Thai                 0            0     0      0           0      0
##   Vietnamese           0            0     0      0           0      0
##   Zulu                 0            0     0      0           0      0
##
##              Thailand Turkey   UK United Arab Emirates Unknown  USA
##   Aboriginal        0      0    1                     0       0    0
##   Arabic            0      1    0                     1       0    0
##   Aramaic           0      0    0                     0       0    1
##   Bosnian           0      0    0                     0       0    1
##   Cantonese         0      0    0                     0       0    1
##   Chinese           0      0    0                     0       0    0
##   Czech             0      0    0                     0       0    0
##   Danish            0      0    0                     0       0    0
##   Dari              0      0    0                     0       0    1
```

```
##   Dutch             0      0    0                    0       0    0
##   Dzongkha          0      0    0                    0       0    0
##   English           2      0  385                    0      20 3375
##   Filipino          0      0    0                    0       0    1
##   French            0      0    0                    0       0    0
##   German            0      0    0                    0       0    0
##   Greek             0      0    0                    0       0    0
##   Hebrew            0      0    0                    0       0    0
##   Hindi             0      0    0                    0       0    1
##   Hungarian         0      0    0                    0       0    0
##   Icelandic         0      0    0                    0       0    0
##   Indonesian        0      0    1                    0       0    0
##   Italian           0      0    1                    0       0    0
##   Japanese          0      0    0                    0       0    1
##   Kannada           0      0    0                    0       0    0
##   Kazakh            0      0    0                    0       0    0
##   Korean            0      0    0                    0       0    0
##   Mandarin          0      0    0                    0       0    0
##   Maya              0      0    0                    0       0    1
##   Mongolian         0      0    0                    0       0    0
##   None              0      0    0                    0       0    1
##   Norwegian         0      0    0                    0       0    0
##   Persian           0      0    0                    0       0    0
##   Polish            0      0    0                    0       0    0
##   Portuguese        0      0    1                    0       0    0
##   Romanian          0      0    0                    0       0    0
##   Russian           0      0    0                    0       0    0
##   Spanish           0      0    0                    0       0    7
##   Swedish           0      0    0                    0       0    0
##   Thai              3      0    0                    0       0    0
##   Vietnamese        0      0    0                    0       0    1
##   Zulu              0      0    1                    0       0    0
##
##              West Germany
##   Aboriginal            0
##   Arabic                0
##   Aramaic               0
##   Bosnian               0
##   Cantonese             0
##   Chinese               0
##   Czech                 0
##   Danish                0
##   Dari                  0
##   Dutch                 0
##   Dzongkha              0
##   English               1
##   Filipino              0
##   French                0
##   German                2
##   Greek                 0
##   Hebrew                0
##   Hindi                 0
##   Hungarian             0
##   Icelandic             0
```

```
##   Indonesian            0
##   Italian               0
##   Japanese              0
##   Kannada               0
##   Kazakh                0
##   Korean                0
##   Mandarin              0
##   Maya                  0
##   Mongolian             0
##   None                  0
##   Norwegian             0
##   Persian               0
##   Polish                0
##   Portuguese            0
##   Romanian              0
##   Russian               0
##   Spanish               0
##   Swedish               0
##   Thai                  0
##   Vietnamese            0
##   Zulu                  0
```

```
##
##             Australia Belgium Brazil Canada Chile China Czech Republic
##   Cantonese         0       0      0      0     0     0               0
##   Chinese           0       0      0      0     0     1               0
##   English           2       1      0      6     1     2               1
##   French            0       0      0      0     0     0               0
##   Hebrew            0       0      0      0     0     0               0
##   Hindi             0       0      0      0     0     0               0
##   Japanese          0       0      0      0     0     0               0
##   Korean            0       0      0      0     0     0               0
##   Mandarin          0       0      0      0     0     1               0
##   Panjabi           0       0      0      1     0     0               0
##   Portuguese        0       0      1      0     0     0               0
##   Romanian          0       0      0      0     0     0               0
##   Russian           0       0      0      0     0     0               0
##   Slovenian         0       0      0      0     0     0               0
##   Spanish           0       0      0      0     0     0               0
##   Tamil             0       0      0      0     0     0               0
##   Telugu            0       0      0      0     0     0               0
##   Urdu              0       0      0      0     0     0               0
##
##             France Germany Greece Hong Kong India Ireland Israel Italy
##   Cantonese      0       0      0         1     0       0      0     0
##   Chinese        0       0      0         0     0       0      0     0
##   English        5       1      1         0     0       1      0     0
##   French         3       0      0         0     0       0      0     0
##   Hebrew         0       0      0         0     0       0      1     0
##   Hindi          0       0      0         0     1       0      0     0
##   Japanese       0       0      0         0     0       0      0     0
##   Korean         0       0      0         0     0       0      0     0
##   Mandarin       0       0      0         0     0       0      0     0
##   Panjabi        0       0      0         0     0       0      0     0
```

```
##   Portuguese      0       0      0         0     0       0      0     0
##   Romanian        0       0      0         0     0       0      0     0
##   Russian         0       0      0         0     0       0      0     0
##   Slovenian       0       0      0         0     0       0      0     0
##   Spanish         0       0      0         0     0       0      0     1
##   Tamil           0       0      0         0     1       0      0     0
##   Telugu          0       0      0         0     1       0      0     0
##   Urdu            0       0      0         0     0       0      0     0
##
##              Japan Mexico New Zealand Pakistan Panama Romania Russia
##   Cantonese      0      0           0        0      0       0      0
##   Chinese        0      0           0        0      0       0      0
##   English        2      2           1        0      1       0      0
##   French         0      0           0        0      0       0      0
##   Hebrew         0      0           0        0      0       0      0
##   Hindi          0      0           0        0      0       0      0
##   Japanese       1      0           0        0      0       0      0
##   Korean         0      0           0        0      0       0      0
##   Mandarin       0      0           0        0      0       0      0
##   Panjabi        0      0           0        0      0       0      0
##   Portuguese     0      0           0        0      0       0      0
##   Romanian       0      0           0        0      0       1      0
##   Russian        0      0           0        0      0       0      1
##   Slovenian      0      0           0        0      0       0      0
##   Spanish        0      1           0        0      0       0      0
##   Tamil          0      0           0        0      0       0      0
##   Telugu         0      0           0        0      0       0      0
##   Urdu           0      0           0        1      0       0      0
##
##              Slovenia South Korea Spain Taiwan  UK USA
##   Cantonese         0           0     0      0   0   0
##   Chinese           0           0     0      0   0   0
##   English           0           1     2      0  26 228
##   French            0           0     0      0   1   0
##   Hebrew            0           0     0      0   0   1
##   Hindi             0           0     0      0   0   0
##   Japanese          0           0     0      0   0   0
##   Korean            0           1     0      0   0   0
##   Mandarin          0           0     0      1   0   0
##   Panjabi           0           0     0      0   0   0
##   Portuguese        0           0     0      0   0   0
##   Romanian          0           0     0      0   0   0
##   Russian           0           0     0      0   0   0
##   Slovenian         1           0     0      0   0   0
##   Spanish           0           0     0      0   0   0
##   Tamil             0           0     0      0   0   0
##   Telugu            0           0     0      0   0   0
##   Urdu              0           0     0      0   0   0
```

Appendix B

```
##                      Estimate   Std. Error          z value     Pr(>|z|)
```

```
## (Intercept)        3.528372e-17 2.967310e+04 -0.0012766819038 9.989814e-01
## duration           1.009888e+00 4.584548e-03  2.1463077931601 3.184843e-02
## rating             8.149042e+00 2.554915e-01  8.2112337021562 2.189273e-16
## gross              1.000000e+00 2.191532e-09  2.6241214582571 8.687280e-03
## languageArabic     1.307725e+00 3.540929e+04  0.0000075767930 9.999940e-01
## languageAramaic    5.031115e-01 5.659824e+04 -0.0000121371890 9.999903e-01
## languageBosnian    4.402886e+01 5.659824e+04  0.0000668721347 9.999466e-01
## languageCantonese  1.352111e+01 3.287820e+04  0.0000792090991 9.999368e-01
## languageChinese    2.490283e+02 5.659824e+04  0.0000974865356 9.999222e-01
## languageCzech      1.236620e-01 5.659824e+04 -0.0000369305261 9.999705e-01
## languageDanish     1.313713e-01 3.643904e+04 -0.0000557019987 9.999556e-01
## languageDari       2.199350e-01 4.489594e+04 -0.0000337318508 9.999731e-01
## languageDutch      1.064267e-01 4.478615e+04 -0.0000500221400 9.999601e-01
## languageEnglish    7.111757e+07 2.967310e+04  0.0006093008465 9.995138e-01
## languageFilipino   2.668049e+00 5.659824e+04  0.0000173388348 9.999862e-01
## languageFrench     3.400176e-01 3.016717e+04 -0.0000357593404 9.999715e-01
## languageGerman     7.136684e-02 3.132458e+04 -0.0000842763720 9.999328e-01
## languageGreek      6.869923e-01 5.659824e+04 -0.0000066332833 9.999947e-01
## languageHebrew     1.056298e+00 4.223091e+04  0.0000012969227 9.999990e-01
## languageHindi      1.538513e-01 3.084991e+04 -0.0000606734015 9.999516e-01
## languageHungarian  1.156082e-01 5.659824e+04 -0.0000381204135 9.999696e-01
## languageIcelandic  1.039183e+00 5.659824e+04  0.0000006790870 9.999995e-01
## languageIndonesian 6.132196e+00 4.334683e+04  0.0000418381937 9.999666e-01
## languageItalian    2.685191e-01 3.326269e+04 -0.0000395287800 9.999685e-01
## languageJapanese   4.195267e-01 3.125700e+04 -0.0000277898706 9.999778e-01
## languageKannada    3.714251e-01 5.659824e+04 -0.0000174989195 9.999860e-01
## languageKazakh     2.128717e+00 5.659824e+04  0.0000133488158 9.999893e-01
## languageKorean     4.887001e-01 3.534378e+04 -0.0000202583409 9.999838e-01
## languageMandarin   5.038514e-01 3.113660e+04 -0.0000220150500 9.999824e-01
## languageMongolian  1.971023e-01 5.659824e+04 -0.0000286940458 9.999771e-01
## languageNone       6.902409e-01 4.304021e+04 -0.0000086132166 9.999931e-01
## languageNorwegian  1.014811e+00 4.215513e+04  0.0000003487709 9.999997e-01
## languagePersian    1.223889e-01 3.625279e+04 -0.0000579418017 9.999538e-01
## languagePolish     3.823062e-01 5.659824e+04 -0.0000169887519 9.999864e-01
## languagePortuguese 1.158353e-01 3.438076e+04 -0.0000626974544 9.999500e-01
## languageRussian    1.443617e+00 3.207894e+04  0.0000114452472 9.999909e-01
## languageSpanish    4.933741e-01 3.075595e+04 -0.0000229707605 9.999817e-01
## languageSwedish    2.426123e-01 3.635077e+04 -0.0000389617747 9.999689e-01
## languageThai       1.738178e+00 4.261544e+04  0.0000129727009 9.999896e-01
## languageVietnamese 2.311439e-01 5.659824e+04 -0.0000258791563 9.999794e-01
## languageZulu       6.154233e-01 4.511470e+04 -0.0000107602404 9.999914e-01
## Action             1.713349e-01 7.865384e-01 -2.2429104014040 2.490259e-02
## Adventure          7.128594e-01 5.235540e-01 -0.6464873236390 5.179638e-01
## Biography          8.358130e-01 4.015753e-01 -0.4466170263363 6.551516e-01
## Comedy             6.110956e-01 3.977491e-01 -1.2382223073155 2.156336e-01
## Crime              6.795498e-01 4.542737e-01 -0.8504227918453 3.950901e-01
## Drama              9.279655e-01 4.890722e-01 -0.1528624006274 8.785068e-01
## Musical            1.534848e+00 4.421744e-01  0.9689193994101 3.325854e-01
## Western            5.309869e-01 8.601634e-01 -0.7359275543284 4.617748e-01
## Family             1.450761e+00 5.758274e-01  0.6461799880064 5.181628e-01
## History            1.203269e+00 4.809605e-01  0.3847344302874 7.004342e-01
## Mystery            1.980136e-08 2.030486e+03 -0.0087356003520 9.930301e-01
## Romance            1.774042e+00 3.139403e-01  1.8260174844206 6.784763e-02
## Sport              9.078845e-01 6.643374e-01 -0.1454653419066 8.843435e-01
```

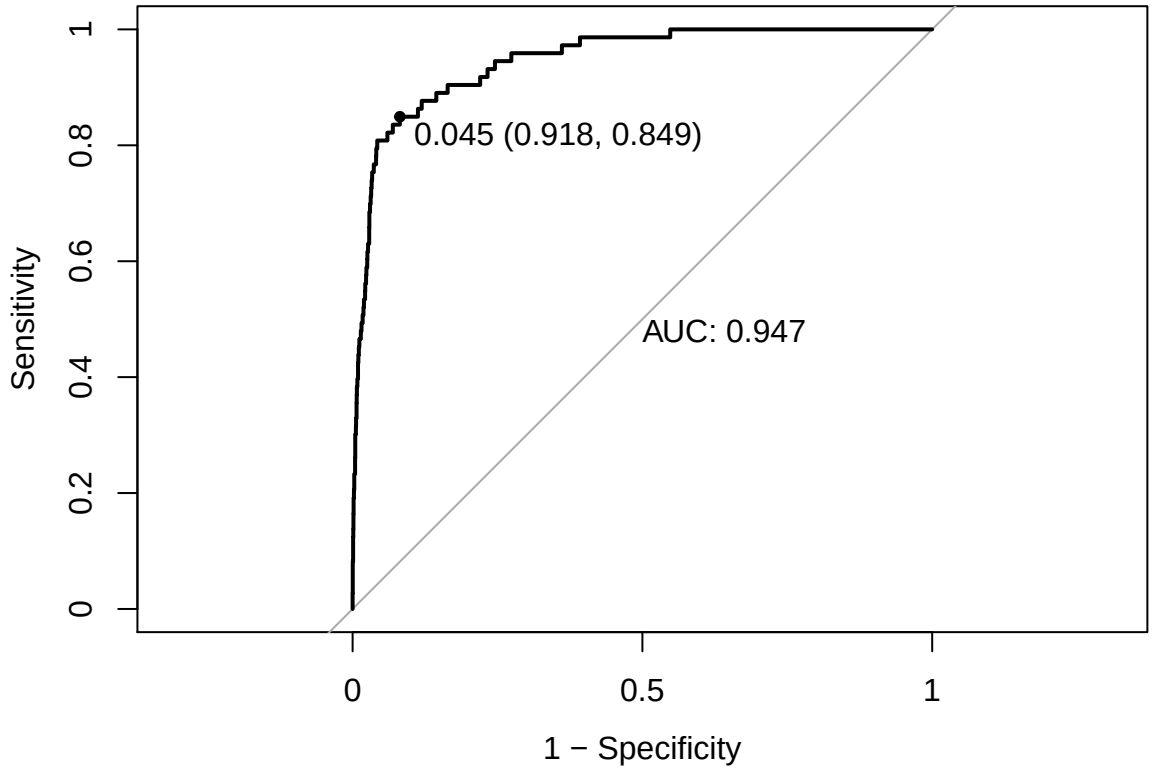
```
## War                2.141124e+00 4.525192e-01  1.6824281028744 9.248586e-02
## Fantasy            8.781741e-02 1.120371e+00 -2.1711518039524 2.991970e-02
## Sci.Fi             1.477134e-08 1.674458e+03 -0.0107680060888 9.914085e-01
## Thriller           6.231189e-01 5.064269e-01 -0.9340299114633 3.502885e-01
## Documentary        2.824181e-09 4.506957e+03 -0.0043677021885 9.965151e-01
## Horror             5.871690e-08 1.761476e+03 -0.0094526041881 9.924580e-01
## Animation          6.745957e-09 2.669249e+03 -0.0070485451579 9.943761e-01
```

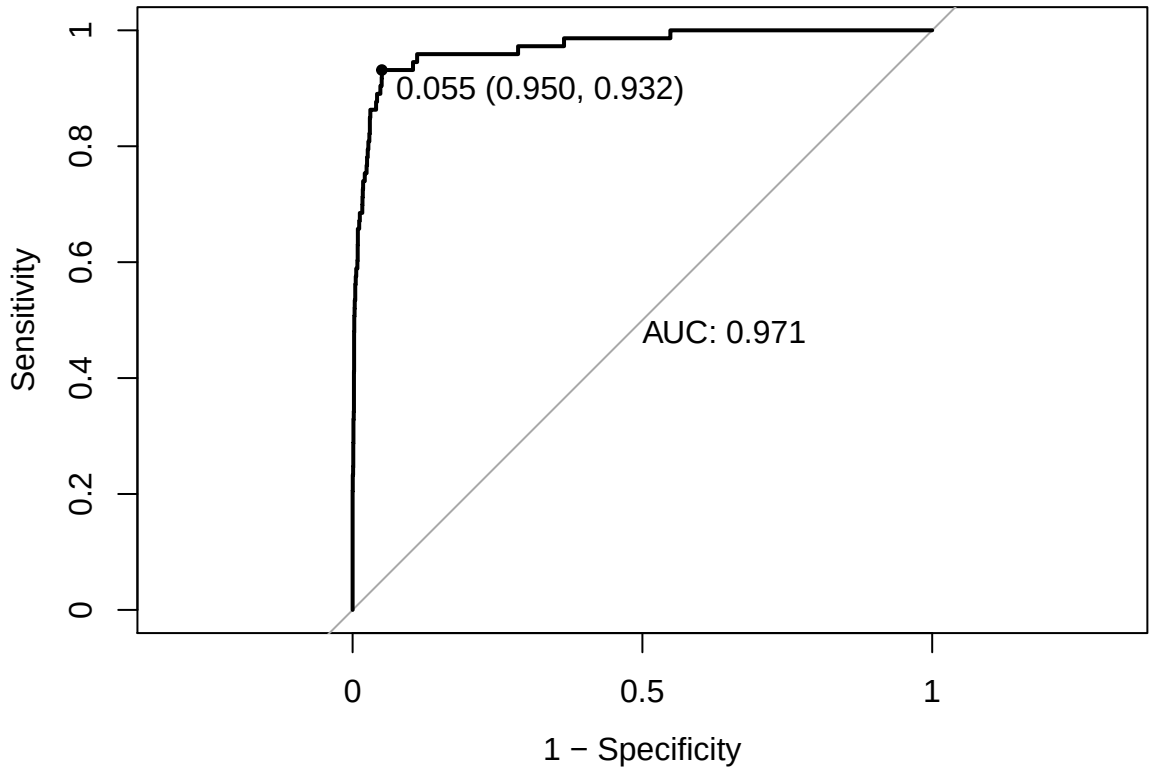
```
##                                Estimate   Std. Error         z value
## (Intercept)                1.823456e-18 4.819611e+04 -0.000847491567
## rating                     9.505520e+00 3.143450e-01  7.163698364033
## Drama                      1.345285e+00 4.840583e-01  0.612747555793
## countryArgentina           2.686804e+00 5.397412e+04  0.000018311599
## countryAustralia           1.753932e+00 4.855255e+04  0.000011572201
## countryBahamas             4.498812e+02 6.815962e+04  0.000089627610
## countryBelgium             7.546379e+10 6.819745e+04  0.000367270615
## countryBrazil              1.326149e-01 5.127579e+04 -0.000039400774
## countryBulgaria            1.217361e+09 6.817663e+04  0.000306849321
## countryCameroon            4.413238e+08 6.830442e+04  0.000291420237
## countryCanada              4.563197e+00 4.836628e+04  0.000031385990
## countryChina               4.203464e+08 4.819611e+04  0.000411995676
## countryColombia            6.193288e-01 6.815962e+04 -0.000007029368
## countryCzech Republic      6.311831e+00 5.748647e+04  0.000032049726
## countryDenmark             5.598157e-01 4.984121e+04 -0.000011639919
## countryDominican Republic  1.237045e+00 6.815962e+04  0.000003120994
## countryEgypt               1.089372e+07 6.830442e+04  0.000237227661
## countryFinland             1.540949e+00 6.815962e+04  0.000006343911
## countryFrance              1.372831e+00 4.833463e+04  0.000006555862
## countryGeorgia             3.355985e+01 6.815962e+04  0.000051545633
## countryGermany             2.270752e+08 4.819611e+04  0.000399218753
## countryGreece              4.249240e+00 6.815962e+04  0.000021225766
## countryHong Kong           6.598133e+00 4.981091e+04  0.000037878989
## countryHungary             1.139712e+37 6.815962e+04  0.001251861801
## countryIceland             2.241576e+00 6.815962e+04  0.000011842483
## countryIndia               6.421582e-01 4.874882e+04 -0.000009085769
## countryIndonesia           6.334410e-01 6.815962e+04 -0.000006698811
## countryIran                1.571271e-01 5.244861e+04 -0.000035285976
## countryIreland             1.415324e+00 4.996035e+04  0.000006952682
## countryIsrael              1.600827e+00 5.634437e+04  0.000008350798
## countryItaly               3.266499e-01 4.913533e+04 -0.000022771116
## countryJapan               4.012971e-01 4.869960e+04 -0.000018748681
## countryKyrgyzstan          1.298285e-02 6.815962e+04 -0.000063734604
## countryMexico              4.468282e-01 5.071651e+04 -0.000015884002
## countryNetherlands         4.560535e-01 5.407257e+04 -0.000014520210
## countryNew Line            8.621970e+03 6.815962e+04  0.000132953637
## countryNew Zealand         1.373736e+00 5.018777e+04  0.000006326916
## countryNorway              4.986567e+00 5.233744e+04  0.000030699776
## countryPeru                6.429216e+01 6.815962e+04  0.000061083642
## countryPhilippines         1.701206e+01 6.815962e+04  0.000041577737
## countryPoland              1.074946e+01 5.625353e+04  0.000042217003
## countryRomania             7.755449e-01 6.815962e+04 -0.000003729326
## countryRussia              2.665514e+00 4.979459e+04  0.000019688826
## countrySlovakia            2.293204e+02 6.815962e+04  0.000079741059
## countrySouth Africa        1.823215e+00 5.087421e+04  0.000011805621
```

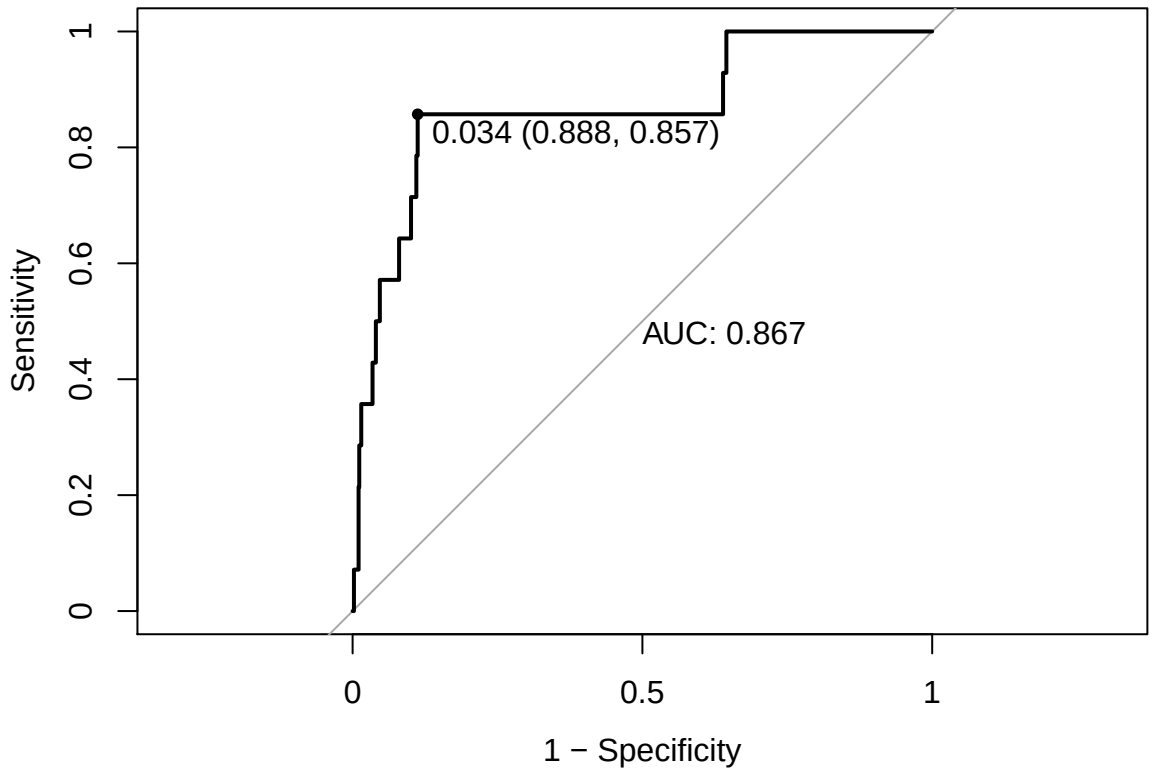
```
## countrySouth Korea          8.966824e-01 4.946701e+04 -0.000002204572
## countrySoviet Union         2.262329e+14 6.820458e+04  0.000484609448
## countrySpain                1.607049e+00 4.878485e+04  0.000009724325
## countrySweden               2.444778e-01 5.194710e+04 -0.000027116641
## countrySwitzerland          2.590074e+01 5.454893e+04  0.000059657843
## countryTaiwan               2.648058e-02 6.815962e+04 -0.000053277053
## countryThailand             8.283538e+02 5.005719e+04  0.000134235261
## countryTurkey               1.190475e+01 6.815962e+04  0.000036340248
## countryUK                   1.658539e+08 4.819611e+04  0.000392700102
## countryUnited Arab Emirates 9.042406e+06 6.830442e+04  0.000234500733
## countryUnknown              2.633194e+11 4.819611e+04  0.000545617322
## countryUSA                  7.658091e+07 4.819611e+04  0.000376666451
## countryWest Germany         1.293658e-02 5.266252e+04 -0.000082557688
## Mystery                     3.269671e-08 1.952706e+03 -0.008826722058
## Sci.Fi                      9.155309e-09 1.522820e+03 -0.012154377509
## duration                    1.015853e+00 4.884481e-03  3.220139464916
## budget                      1.000000e+00 8.566964e-09 -4.058877879390
## gross                       1.000000e+00 2.768576e-09  4.927872171842
## Animation                   3.130230e-09 2.271258e+03 -0.008621724990
## Romance                     2.037166e+00 3.438417e-01  2.069438878190
## Fantasy                     8.543880e-02 1.271443e+00 -1.934773574101
## Documentary                 7.061931e-09 4.445230e+03 -0.004222176452
## Horror                      7.447863e-08 1.662613e+03 -0.009871664582
## War                         2.175316e+00 4.840315e-01  1.605626281794
## Family                      2.543112e+00 6.116664e-01  1.525976101606
##                                 Pr(>|z|)
## (Intercept)                 9.993238e-01
## rating                      7.852909e-13
## Drama                       5.400433e-01
## countryArgentina            9.999854e-01
## countryAustralia            9.999908e-01
## countryBahamas              9.999285e-01
## countryBelgium              9.997070e-01
## countryBrazil               9.999686e-01
## countryBulgaria             9.997552e-01
## countryCameroon             9.997675e-01
## countryCanada               9.999750e-01
## countryChina                9.996713e-01
## countryColombia             9.999944e-01
## countryCzech Republic       9.999744e-01
## countryDenmark              9.999907e-01
## countryDominican Republic   9.999975e-01
## countryEgypt                9.998107e-01
## countryFinland              9.999949e-01
## countryFrance               9.999948e-01
## countryGeorgia              9.999589e-01
## countryGermany              9.996815e-01
## countryGreece               9.999831e-01
## countryHong Kong            9.999698e-01
## countryHungary              9.990012e-01
## countryIceland              9.999906e-01
## countryIndia                9.999928e-01
## countryIndonesia            9.999947e-01
## countryIran                 9.999718e-01
```

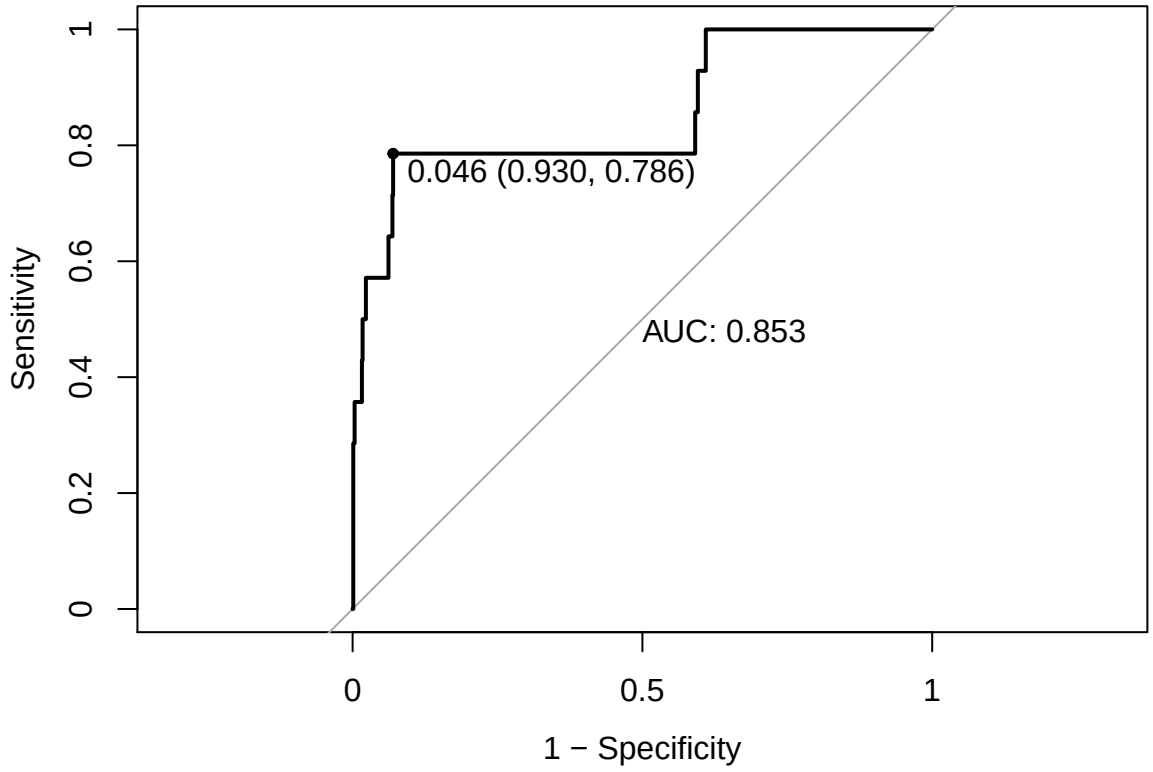
```
## countryIreland              9.999945e-01
## countryIsrael               9.999933e-01
## countryItaly                9.999818e-01
## countryJapan                9.999850e-01
## countryKyrgyzstan           9.999491e-01
## countryMexico               9.999873e-01
## countryNetherlands          9.999884e-01
## countryNew Line             9.998939e-01
## countryNew Zealand          9.999950e-01
## countryNorway               9.999755e-01
## countryPeru                 9.999513e-01
## countryPhilippines          9.999668e-01
## countryPoland               9.999663e-01
## countryRomania              9.999970e-01
## countryRussia               9.999843e-01
## countrySlovakia             9.999364e-01
## countrySouth Africa         9.999906e-01
## countrySouth Korea          9.999982e-01
## countrySoviet Union         9.996133e-01
## countrySpain                9.999922e-01
## countrySweden               9.999784e-01
## countrySwitzerland          9.999524e-01
## countryTaiwan               9.999575e-01
## countryThailand             9.998929e-01
## countryTurkey               9.999710e-01
## countryUK                   9.996867e-01
## countryUnited Arab Emirates 9.998129e-01
## countryUnknown              9.995647e-01
## countryUSA                  9.996995e-01
## countryWest Germany         9.999341e-01
## Mystery                     9.929574e-01
## Sci.Fi                      9.903024e-01
## duration                    1.281282e-03
## budget                      4.930909e-05
## gross                       8.312997e-07
## Animation                   9.931209e-01
## Romance                     3.850492e-02
## Fantasy                     5.301809e-02
## Documentary                 9.966312e-01
## Horror                      9.921237e-01
## War                         1.083560e-01
## Family                      1.270158e-01
```

Appendix C


Sensitivity
0
0.2
0.4
0.6
0.8
1
0.045 (0.918, 0.849)
AUC: 0.947
0
0.5
1
1 − Specificity


Sensitivity
0
0.2
0.4
0.6
0.8
1
0.055 (0.950, 0.932)
AUC: 0.971
0
0.5
1
1 − Specificity


Sensitivity
0
0.2
0.4
0.6
0.8
1
0.034 (0.888, 0.857)
AUC: 0.867
0
0.5
1
1 − Specificity


Sensitivity
0
0.2
0.4
0.6
0.8
1
0.046 (0.930, 0.786)
AUC: 0.853
0
0.5
1
1 − Specificity

Appendix D

I. Initial Preparations

The following code is needed as a basis to run the code used to generate the graphs and output displayed in this report.

```
library(dplyr)
library(ggplot2)
library(gridExtra)
library(dmm)
library(pROC)

options(scipen = 5)
train_movies <- read.csv("train_movies.csv", stringsAsFactors = FALSE)

#Remove invalid country name
train_movies$country[train_movies$country == "Official site"] <- "Unknown"

won_movies <- subset(train_movies, train_movies$nominations != 0)

new_movies <- read.csv("test_movies.csv", stringsAsFactors = FALSE)

all_movies <- read.csv("all_movies.csv", stringsAsFactors = TRUE)

#Remove excess X column
all_movies <- all_movies[, -1]

#Change na's to "Unknown" all_movies$nominations[is.na(all_movies$nominations)] <- "Unknown"

set.seed(123)

#Define negation of %in%
`%!in%` = Negate(`%in%`)

#Define training set as 80% of total and test set as 20% of total
movies_train <- all_movies[sample(nrow(all_movies), size = .8 * nrow(all_movies)), ]

movies_test <- all_movies[which(all_movies$name %!in% movies_train$name), ]

#Remove unknowns from movies_train
movies_train <- movies_train[movies_train$nominations != "Unknown", ]

#Convert nominations into a binary Yes/No factor
#Unfactor nominations to do numeric test
movies_train$nominations <- unfactor(movies_train$nominations)

movies_train$nominations <- ifelse(movies_train$nominations > 0, movies_train$nominations <- 1, movies_train$nominations <- 0)

#Convert nominations back into a factor and give "Yes/No" levels movies_train$nominations <- factor(movies_train$nominations, levels = c(0, 1), labels = c("No", "Yes"))

movies.mod7 <- glm(nominations ~ duration + rating + gross + language + Action + Adventure + Biography + Comedy + Crime + Drama + Musical + Western + Family + History + Mystery + Romance + Sport + War + Fantasy + Sci.Fi + Thriller + Documentary + Horror + Animation, family = binomial(), data = movies_train)

#Attempt forward stepwise model selection
mod.base <- glm(nominations ~ 1, family = binomial(), data = movies_train)

fwd.model <- step(mod.base, direction = "forward", scope = (~ duration + rating + gross + budget + language + country + Action + Adventure + Biography + Comedy + Crime + Drama + Musical + Western
```

+ Family + History + Mystery + Romance + Sport + War + Fantasy + Sci.Fi + Thriller + Documentary + Horror + Animation), trace = 0)

summary(fwd.model)

#Store forward model in fwd.model variable based on call shown in summary for ease of loading
fwd.model <- glm(formula = nominations ~ rating + Drama + country + Mystery + Sci.Fi + duration + budget + gross + Animation + Romance + Fantasy + Documentary + Horror + War + Family, family = binomial(), data = movies_train)

#Run predictions on both models on training set to evaluate models
pred_movies_train <- movies_train[, -3]

#Run predict function on both models
probs_train1 <- predict(movies.mod7, newdata = pred_movies_train, type = "response")
probs_train2 <- predict(fwd.model, newdata = pred_movies_train, type = "response")

pred1_train <- rep("No", 3604)
pred2_train <- rep("No", 3604)

#Clean movies test as above, removing unknown nominations and changing to a binary ("Yes/No")
#Remove unknowns from movies_test
movies_test <- movies_test[movies_test$nominations != "Unknown",]

#Convert nominations into a binary Yes/No factor
#Unfactor nominations to do numeric test
movies_test$nominations <- unfactor(movies_test$nominations)

movies_test$nominations <- ifelse(movies_test$nominations > 0, movies_test$nominations <- 1, movies_test$nominations <- 0)

#Convert nominations back into a factor and give "Yes/No" levels
movies_test$nominations <- factor(movies_test$nominations, levels = c(0, 1), labels = c("No", "Yes"))

#Remove new languages Dzongkha, Maya and Romanian from Test set not in train set (only 3 movies, none were winners)

movies_test <- movies_test[movies_test$language != "Dzongkha" & movies_test$language != "Maya" & movies_test$language != "Romanian",]

#Remove countries in test set not in train set (only 4 movies, none winners)
movies_test <- movies_test[movies_test$country != "Aruba" & movies_test$country != "Cambodia" & movies_test$country != "Libya" & movies_test$country != "Nigeria",]

#Remove response variable from test set
pred_movies <- movies_test[, -3]

#Run predict function on both models
probs1 <- predict(movies.mod7, newdata = pred_movies, type = "response")
probs2 <- predict(fwd.model, newdata = pred_movies, type = "response")

II. Language and Country

table(won_movies$language, won_movies$country)

III. Budget and Gross

plot1 <- ggplot(train_movies, aes((budget/10^6), (gross/10^6))) + geom_point() + labs(title="Movies from 1927-2014", x="Budget in Millions of Dollars", y="Gross in Millions of Dollars")

plot2 <- ggplot(won_movies, aes((budget/10^6), (gross/10^6))) + geom_point() + labs(title="Oscar-Winning Movies", x="Budget in Millions of Dollars", y="Gross in Millions of Dollars")
plot3 <- ggplot(new_movies, aes((budget/10^6), (gross/10^6))) + geom_point() + labs(title="Movies from 2015-2016", x="Budget in Millions of Dollars", y="Gross in Millions of Dollars")

grid.arrange(plot1, plot2, plot3, ncol=2)

IV. Genres

genres1 <- train_movies[, c("Action", "Adventure", "Biography", "Comedy", "Crime", "Drama", "Musical", "Western", "Family", "Film.Noir", "History", "Mystery", "Romance", "Sport", "War", "Fantasy", "Sci.Fi", "Thriller", "Documentary", "Horror", "Animation")]
genres2 <- won_movies[, c("Action", "Adventure", "Biography", "Comedy", "Crime", "Drama", "Musical", "Western", "Family", "Film.Noir", "History", "Mystery", "Romance", "Sport", "War", "Fantasy", "Sci.Fi", "Thriller", "Documentary", "Horror", "Animation")]
genres_new <- new_movies[, c("Action", "Adventure", "Biography", "Comedy", "Crime", "Drama", "Musical", "Western", "Family", "Film.Noir", "History", "Mystery", "Romance", "Sport", "War", "Fantasy", "Sci.Fi", "Thriller", "Documentary", "Horror", "Animation")]

genre_plot1 <- as.data.frame(colSums(genres1))
percent1 <- (genre_plot1$`colSums(genres1)` / nrow(train_movies)) * 100
plt1 <- ggplot(genre_plot1, aes(x=rownames(genre_plot1), y=percent1)) + geom_bar(stat = "identity", position = "dodge") + labs(x = "Genres", y = "Percent", title = "Movie Genres 1927-2014") + theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))

genre_plot2 <- as.data.frame(colSums(genres2))
percent2 <- (genre_plot2$`colSums(genres2)` / nrow(won_movies)) * 100
plt2 <- ggplot(genre_plot2, aes(x=rownames(genre_plot2), y=percent2)) + geom_bar(stat = "identity", position = "dodge") + labs(x = "Genres", y = "Percent", title = "Oscar-Winning Movie Genres") + theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))

genre_plot_new <- as.data.frame(colSums(genres_new)) percent_new <- (genre_plot_new$`colSums(genres_new)` / nrow(new_movies)) * 100

plt3 <- ggplot(genre_plot_new, aes(x=rownames(genre_plot_new), y=percent_new)) + geom_bar(stat = "identity", position = "dodge") + labs(x = "Genres", y = "Percent", title = "New Movie Genres") + theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))

grid.arrange(plt1, plt2, plt3, ncol=2)

V. Evaluating the Models

A. Confusion Matrices–Training Set

pred1_train <- rep("No", 3604)
pred2_train <- rep("No", 3604)

#Use optimal thresholds here
pred1_train[probs_train1 > .045] <- "Yes"
pred2_train[probs_train2 > .055] <- "Yes"

table(pred1_train, movies_train$nominations)

table(pred2_train, movies_train$nominations)

B. Confusion Matrices–Test Set

```
pred1 <- rep("No", 887)
pred2 <- rep("No", 887)

#Use optimal thresholds here
pred1[probs1 > .034] <- "Yes"
pred2[probs2 > .046] <- "Yes"

table(pred1, movies_test$nominations)

table(pred2, movies_test$nominations)
```

VI. Applying the Model

```
pred_Dat <- data.frame(duration = c(100, 144), rating = c(8, 8), gross = c(44667095, 167007184), language = c("English", "English"), Action = c(0, 1), Adventure = c(0, 1), Biography = c(0, 0), Comedy = c(1, 0), Crime = c(0, 0), Drama = c(1, 0), Musical = c(0, 0), Western = c(0, 0), Family = c(0, 0), History = c(0, 0), Mystery = c(0, 0), Romance = c(1, 0), Sport = c(0, 0), War = c(0, 0), Fantasy = c(0, 0), Sci.Fi = c(0, 0), Thriller = c(0, 1), Documentary = c(0, 0), Horror = c(0, 0), Animation = c(0, 0))

pred_Result <- pred_movies[pred_movies$name == "THE ARTIST" | pred_movies$name == "CASINO ROYALE", ]

pred_Result <- cbind(pred_Result, predict(movies.mod7, newdata = pred_Dat, type = "response", se.fit = TRUE))

pred_Result <- pred_Result[, c(1, 35, 36, 2:34, 37)]

pred_Result[, c(1:3, 5)]
```

VII. Appendix A

```
table(train_movies$language, train_movies$country)

table(new_movies$language, new_movies$country)
```

VIII. Appendix B

```
movies.mod.tab7 <- coef(summary(movies.mod7))
movies.mod.tab7[, "Estimate"] <- exp(coef(movies.mod7))

movies.mod.tab7

movies.mod.tab8 <- coef(summary(fwd.model))
movies.mod.tab8[, "Estimate"] <- exp(coef(fwd.model))

movies.mod.tab8
```

IX. Appendix C

```
movies_train$prob1 <- probs_train1
train_ROC1 <- roc(nominations ~ prob1, data = movies_train)
plot.roc(train_ROC1, legacy.axes = TRUE, print.thres = TRUE, print.auc = TRUE)
```

```
movies_train$prob2 <- probs_train2
train_ROC2 <- roc(nominations ~ prob2, data = movies_train)
plot.roc(train_ROC2, legacy.axes = TRUE, print.thres = TRUE, print.auc = TRUE)

movies_test$prob1 <- probs1
ROC1 <- roc(nominations ~ prob1, data = movies_test)
plot.roc(ROC1, legacy.axes = TRUE, print.thres = TRUE, print.auc = TRUE)

movies_test$prob2 <- probs2
ROC2 <- roc(nominations ~ prob2, data = movies_test)
plot.roc(ROC2, legacy.axes = TRUE, print.thres = TRUE, print.auc = TRUE)
```