A GLM Approach to Analyse IMDb Scores

Group 9

March 20, 2023

Introduction

IMDb scores are an excellent way to gauge the performance of movies. In this project, we use Generalised Linear Models (GLM) on a set of movies to determine the properties that influence their IMDb scores.

Data description

The dataset used for this analysis contains information on 3001 movies and was collected from the IMDb database. It include 7 features namely:

- film_id: The unique identifier for the film
- year: Year of release of the film in cinemas
- length: Duration (in minutes)
- budget: Budget for the films production (in $1000000s)
- votes: Number of positive votes received by viewers
- genre: Genre of the film
- rating: IMDB rating from 0-10

Step 1: Data exploration

The following packages were used for the analysis.

```
# Load required packages
library(tidyverse)
library(ggplot2)
library(scales)
library(dplyr)
library(broom)
library(pROC)
library(MASS)
```

Firstly, we read the data file and name it "films".

```
# Read data
films <- read.csv("dataset9.csv", header = TRUE)
# View data structures and variable types
str(films)
```

```
## 'data.frame':    3001 obs. of  7 variables:
##  $ film_id: int  45327 55943 7752 34995 21585 20729 16345 39560 274 25005 ...
##  $ year   : int  1984 2001 1999 1970 1939 1961 1978 1975 1999 1998 ...
##  $ length : int  103 60 105 135 117 90 95 110 20 101 ...
##  $ budget : num  14.4 10.2 13.4 11.6 17 10.7 14.7 14 12 13.4 ...
##  $ votes  : int  17 11 3216 73 1988 7 134 8 5 1645 ...
##  $ genre  : chr  "Comedy" "Documentary" "Documentary" "Comedy" ...
##  $ rating : num  8 8.1 7.9 7.1 8 2.8 8.3 2.4 8.1 8.6 ...
```

```
# View variable distribution
summary(films)
```

```
##     film_id           year          length           budget
##  Min.   :   16   Min.   :1895   Min.   :  1.00   Min.   : 1.20
##  1st Qu.:14874   1st Qu.:1957   1st Qu.: 71.25   1st Qu.:10.10
##  Median :29673   Median :1983   Median : 90.00   Median :12.10
##  Mean   :29709   Mean   :1976   Mean   : 81.57   Mean   :11.98
##  3rd Qu.:44660   3rd Qu.:1997   3rd Qu.:100.00   3rd Qu.:14.00
##  Max.   :58753   Max.   :2005   Max.   :555.00   Max.   :23.40
##                                 NA's   :127
##      votes            genre              rating
##  Min.   :    5.0   Length:3001        Min.   :0.8
##  1st Qu.:   11.0   Class :character   1st Qu.:3.7
##  Median :   30.0   Mode  :character   Median :4.7
##  Mean   :  655.8                      Mean   :5.4
##  3rd Qu.:  118.0                      3rd Qu.:7.8
##  Max.   :92437.0                      Max.   :9.2
##
```

The dataset contains information about movies released between 1895 and 2005. While the length of the movies in our dataset is between 1 and 555 minutes, 98% (approx.) of the movies have a duration less than 150 minutes.

Moreover, the mean value of the variable `votes` is significantly greater than its median and therefore, its distribution is skewed to the right.

Since our goal is to determine the features that affect the IMDb score of a movie, we consider `rating` as the response variable. The least value observed for rating in our dataset is 0.8, while the greatest value is 9.2. The distribution of IMDb scores is shown in figure 1.

Secondly, we use correlation matrix to check if there there is a collinearity problem.

```
# Correlation matrix
cor(films[,c("year", "length", "budget", "votes", "rating")])
```

```
##               year length     budget       votes      rating
## year    1.00000000     NA 0.03139738  0.09221425 -0.03525934
## length          NA      1         NA          NA          NA
## budget  0.03139738     NA 1.00000000  0.03137437  0.22510750
## votes   0.09221425     NA 0.03137437  1.00000000 -0.04854771
## rating -0.03525934     NA 0.22510750 -0.04854771  1.00000000
```

We do not observe any significant correlation among these five numeric variables.

Step 2: Data Visualizations

Firstly, we visually look at the distribution of scores.

```
# Draw a histogram (show the number of movies with different ratings,
# and the color distinguishes the fuzzy interval with a score of about 7)
ggplot(films, aes(x = rating, fill = factor(rating > 7))) +
  geom_histogram(alpha = 0.5, binwidth = 0.5) +
  scale_fill_manual(values = c("#E69F00", "#56B4E9"),
                    name = "Rating > 7",
                    labels = c("No", "Yes")) +
  labs(title = "Distribution of IMDB ratings", x = "IMDB rating", y = "Frequency") +
  theme_classic()
```

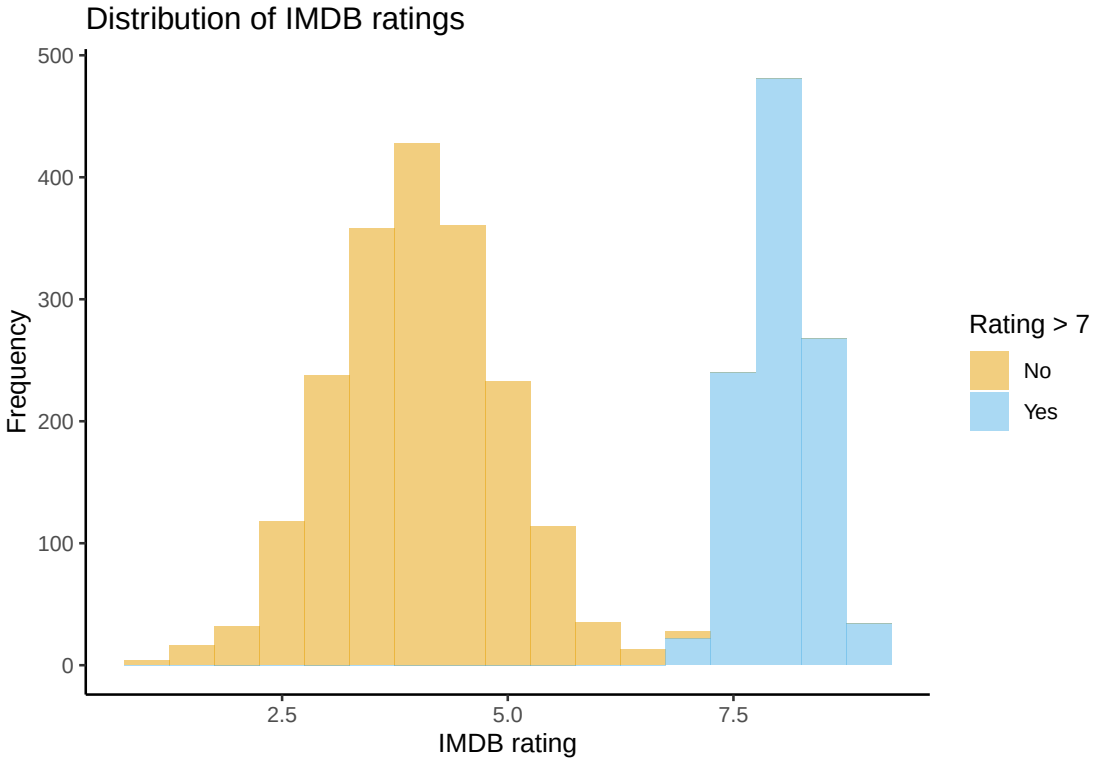

Figure 1: Distribution of IMDb ratings

The histogram (figure 1) shows that the IMDb scores follow a bimodal distribution i.e., it has two peaks. This indicates that we separate the data into two subgroups: one with IMDb scores ranging from 0 to 7, and another with scores ranging from 7 to 10. And there are more films which have scores lower than 7.

Secondly,we draw a scatter plot to show the relationship between budget, rating and genre.

```
ggplot(films, aes(x = budget, y = rating)) +
  geom_point(aes(size = votes, color = genre)) +
  scale_color_brewer(palette = "Set2", name = "Genre") +
  labs(title = "Relationship between budget, rating and genre",
       x = "Budget", y = "IMDB rating") +
  theme_classic()
```

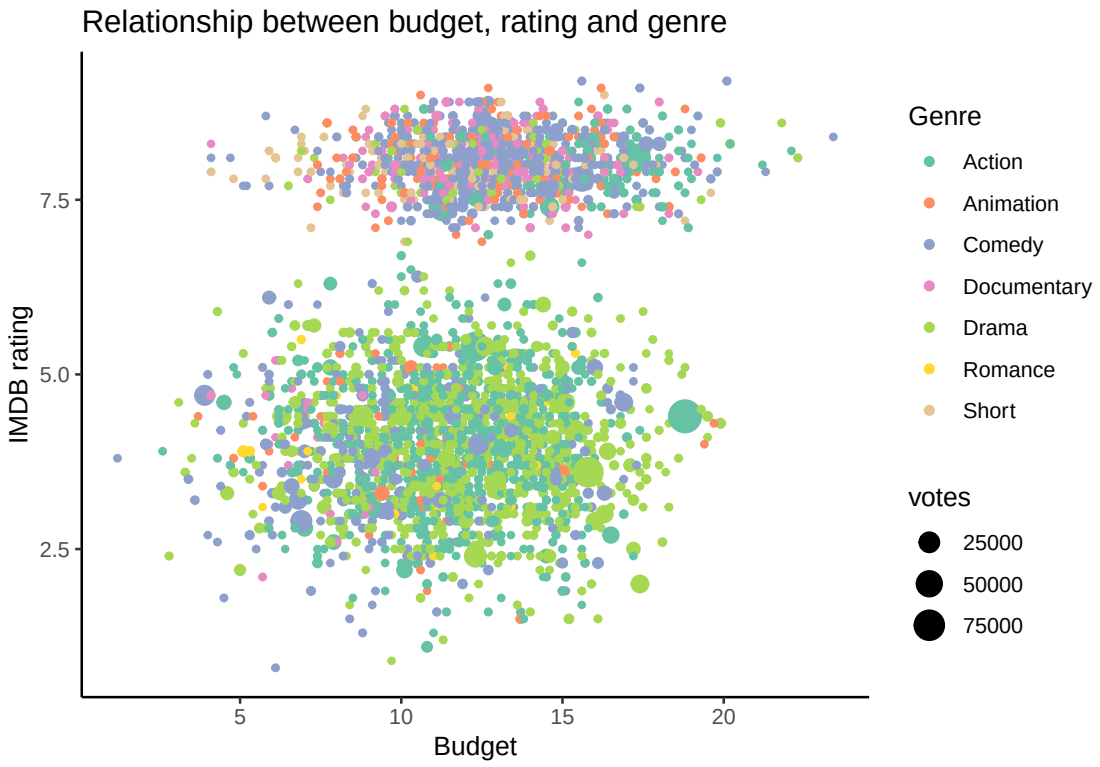

Figure 2: Relationship between budget, rating and genre

The scatter plot (figure 2) shows that:

- Less popular films have significantly higher ratings.
- Blockbuster films receive higher levels of attention.

Thirdly, we draw a line chart by year.

```
ggplot(films, aes(x = year, y = rating, group = 1)) +
  stat_summary(fun = mean, geom = "line", color = "red", size = 1) +
  stat_summary(fun = median, geom = "line", color = "blue", size = 1) +
  scale_x_continuous(breaks = seq(1930, 2020, by = 10),
                     labels = seq(1930, 2020, by = 10)) +
  labs(title = "Trend of IMDB rating", x = "Year", y = "IMDB rating") +
  theme_classic()
```

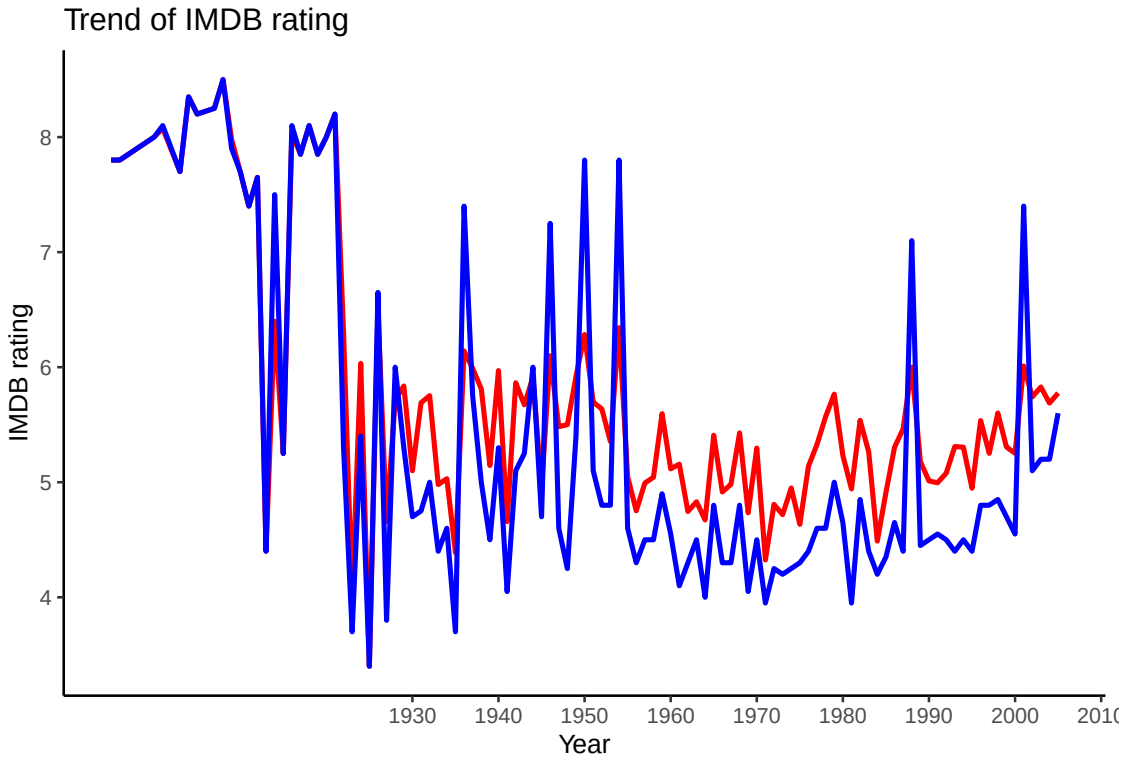

Figure 3: Trend of IMDB rating

In this plot (figure 3), the blue line represents the trend of mean value. And the red one represents the median value. We could find that the mean is more volatile than the median.

Step 3: Data preprocessing

In order to make the dataset more suitable for analysis we have implemented the following changes.

3.1. Handling missing values

The variable `length` has 4.2% missing values. Since this is a relatively small value, we have decided to fill the missing values.
And we plot different counts of different length of films to decide whether it is better to use the mean or the median.

```
ggplot(data=films,aes(length))+
  geom_histogram(color='black',fill='gray60',binwidth = 1)+
  labs(title = "Counts of different lengths")
```

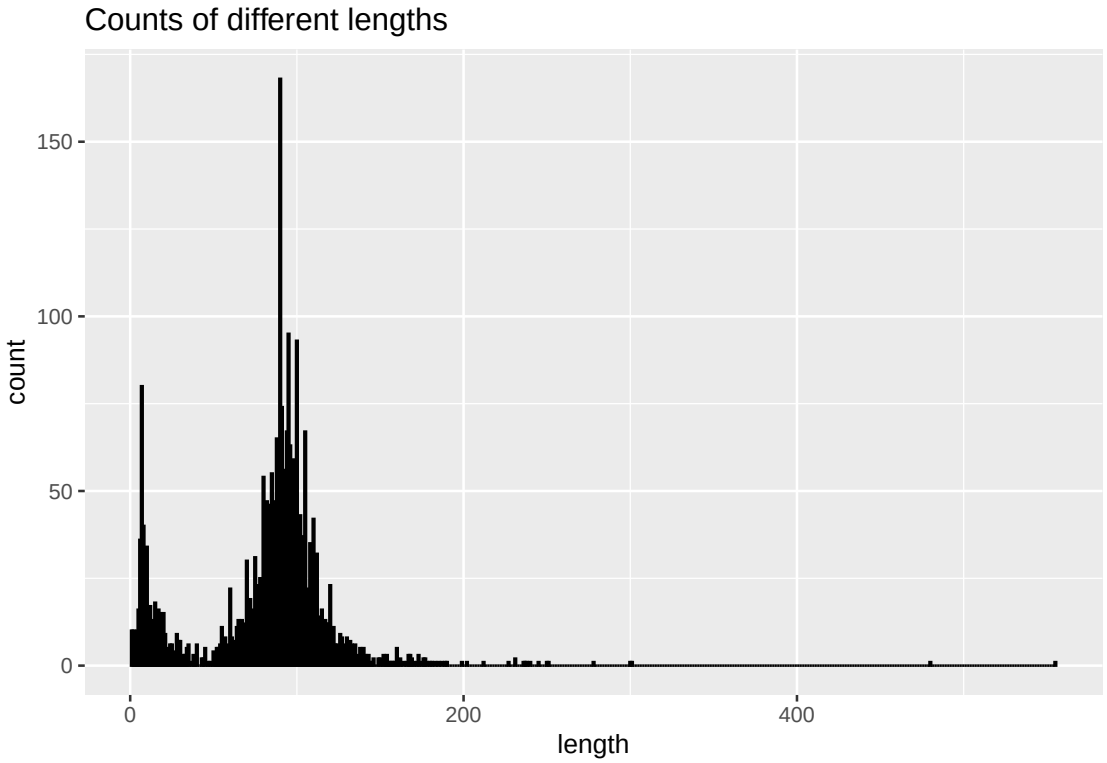

Figure 4: Counts of different lengths

From the plot above(figure 4), we could find that it follows a bimodal distribution i.e., it has two peaks. In this case ,it is not appropriate to replace them with the mean length of the film. Hence, we decide to fill them by median values.

```
# Check for missing values
sum(is.na(films))
```

```
## [1] 127
```

```r
# Check which column has missing values
colSums(is.na(films))
```

```
## film_id    year  length  budget   votes   genre  rating
##       0       0     127       0       0       0       0
```

```r
#Fill missing values
median_length <- median(films$length, na.rm = TRUE)
films$length[is.na(films$length)] <- median_length
# Check if missing values were filled successfully
sum(is.na(films))
```

```
## [1] 0
```

3.2. Feature transformation

We have performed transformations for columns `year` and `length` to convert them to continuous variables. Since `genre` is a qualitative feature, we used the *one-hot encoding technique* to transform it into binary. And finally, considering the bimodal distribution of IMDb scores, we binarize the variable `rating` by choosing the threshold value 7.

```r
# Convert year and length to continuous variables
# (Because time and length can be regarded as unlimited values)
films$year <- as.numeric(as.character(films$year))
films$length <- as.numeric(as.character(films$length))
# one-hot encoding
films <- cbind(films, model.matrix(~genre-1, data=films))
# Binarize the IMDB score
films$rating2 <- ifelse(films$rating > 7, 1, 0)
```

After all the above operations are performed, the dataset contains 3001 observations with 15 features.

Step 4: Model Fit

Genaralised Linear Model (GLM) based on logistic regression was chosen to model the data.

```
# Fitting a GLM model
fit <- glm(rating2 ~ year + length + budget + votes + genreAction +
             genreAnimation+genreComedy + genreDocumentary + genreDrama +
             genreRomance, data=films, family=binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = rating2 ~ year + length + budget + votes + genreAction +
##     genreAnimation + genreComedy + genreDocumentary + genreDrama +
##     genreRomance, family = binomial, data = films)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9859  -0.3410  -0.1143   0.1945   2.9726
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -8.279e+00  5.702e+00  -1.452  0.14656
## year             4.762e-03  2.840e-03   1.677  0.09357 .
## length          -6.005e-02  3.572e-03 -16.812  < 2e-16 ***
## budget           5.160e-01  2.885e-02  17.882  < 2e-16 ***
## votes            4.239e-05  1.526e-05   2.777  0.00548 **
## genreAction     -4.704e+00  1.055e+00  -4.458 8.28e-06 ***
## genreAnimation  -4.998e+00  1.058e+00  -4.721 2.34e-06 ***
## genreComedy     -1.376e+00  1.054e+00  -1.305  0.19187
## genreDocumentary 6.055e-01  1.103e+00   0.549  0.58287
## genreDrama      -6.167e+00  1.059e+00  -5.822 5.83e-09 ***
## genreRomance    -5.486e+00  1.322e+00  -4.149 3.33e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3879.3  on 3000  degrees of freedom
## Residual deviance: 1550.9  on 2990  degrees of freedom
## AIC: 1572.9
##
## Number of Fisher Scoring iterations: 7
```

The variables `length`, `genreAction`, `genreAnimation`, `genreComedy`, `genreDrama` and `genreRomance` take negative values for coefficient estimates. This means that higher values of these variables are associated with a lower likelihood of the `rating` variable taking on a value of 1.

The p-values for the variables `year`, `genreComedy` and `genreDocumentary` are greater than 0.05 and hence they are not statistically significant in this model.

```
# Extract Coefficients and Standard Errors
coef_df <- tidy(fit, exponentiate = TRUE) %>%
```

```
  filter(term != "(Intercept)") %>%
  mutate_if(is.numeric, list(~ round(., 2))) %>%
  mutate(
    lower = estimate - 1.96 * std.error,
    upper = estimate + 1.96 * std.error
  )
```

```
# plot the coefficients
ggplot(coef_df, aes(x = term, y = estimate)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2) +
  coord_flip() +
  labs(
    title = "Effect of predictors on movie rating",
    x = "", y = "Odds Ratio"
  ) +
  theme_minimal()
```

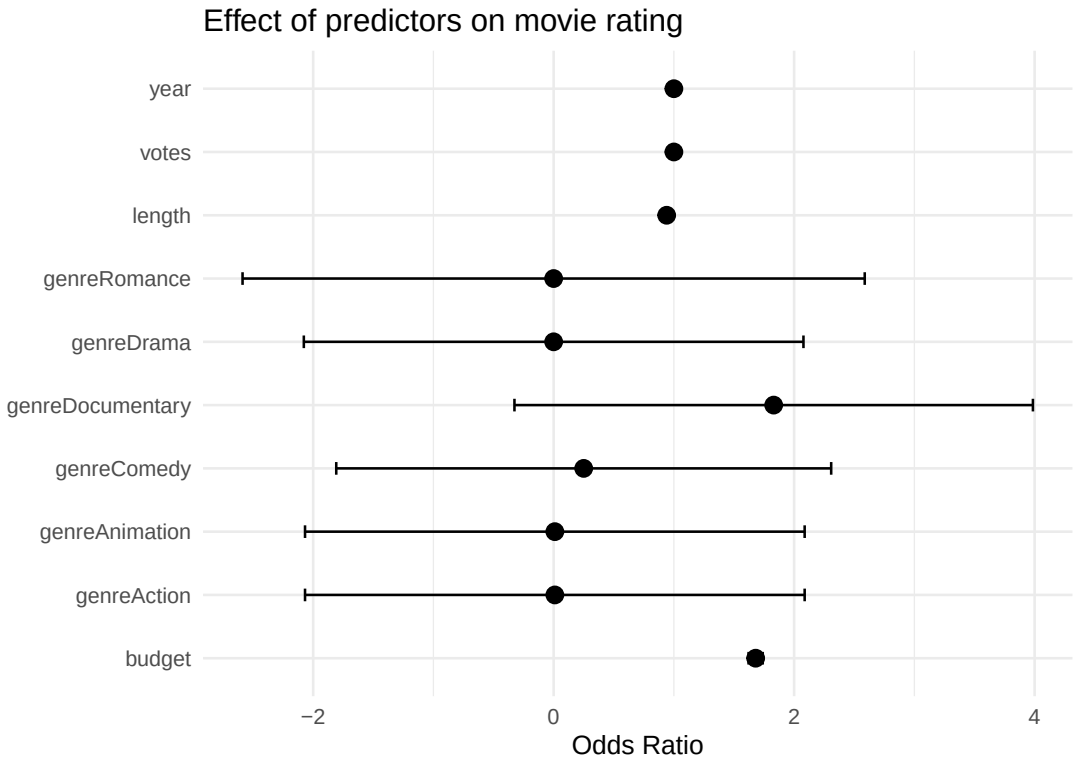

Figure 5: Effect of predictors on movie rating

According to the results, the following conclusions can be drawn:

- Year (year) has a coefficient of 1, indicating that there is no significant difference in the impact of the year on the score.
- The coefficient of film length (length) is 0.94, and the confidence interval does not include 1, indicating that the film length has a negative impact on the score, that is, the longer the film length, the lower the score.

- The coefficient of budget (budget) is 1.68, and the confidence interval does not include 1, indicating that the budget has a positive impact on the score, that is, the higher the budget, the higher the score.
- The coefficient of the number of votes (votes) is 1, indicating that the impact of the number of votes on the score is not significantly different.
- None of the different genres had a significant effect.

Step 5: Variables Selection and Models Selection

The observations from variable p-values indicate that some variables are statistically insignificant and hence can be removed from the equation.

```
# After selecting variables:
# According to GLM fit summary, we choose the variables with p-value lower than 0.05
fit2 = glm(rating2 ~ length + budget + votes + genreAction + genreAnimation + genreDrama + genreRomance
fit3 = glm(rating2 ~ length + budget , data=films, family=binomial)
table1 = matrix(c(fit$aic, fit2$aic, fit3$aic), dimnames = list(c("fit","fit2","fit3"), c("AIC")))
table1
```

```
##           AIC
## fit  1572.895
## fit2 1615.509
## fit3 2736.135
```

The AIC value of the first models is less than that of the second and third model. Therefore, the first model is preferred.

```
# Store the variables that need to be scatter plotted in a data frame
variables <- c("year", "length", "budget", "votes")
plot_data <- films[variables]
```

Then we try to plot a scatterplot matrix to find if there is interaction terms.

```
# Plot a scatterplot matrix
pairs(plot_data)
```

It can be seen from the scatter plot matrix (figure 6) that the scatter plots between length and year, votes and year, length and votes, budget and length, budget and votes all present a triangular shape, that is, there may be interactions. And as for year and budget, the pattern of that plot is rectangular. Hence, these five interaction terms are put into the model for comparison

```
# test interaction
films$year_length <- films$year * films$length
films$year_votes <- films$year * films$votes
films$length_votes <- films$length * films$votes
films$budget_length <- films$budget * films$length
films$budget_votes <- films$budget * films$votes

fit4 <- glm(rating2 ~ year + length + budget + votes + genreAction + genreAnimation + genreComedy + geni

table2 = matrix(c(fit$aic, fit4$aic), dimnames = list(c("fit","fit4"), c("AIC")))
print(table2)
```

```
##           AIC
## fit  1572.895
## fit4 1550.086
```

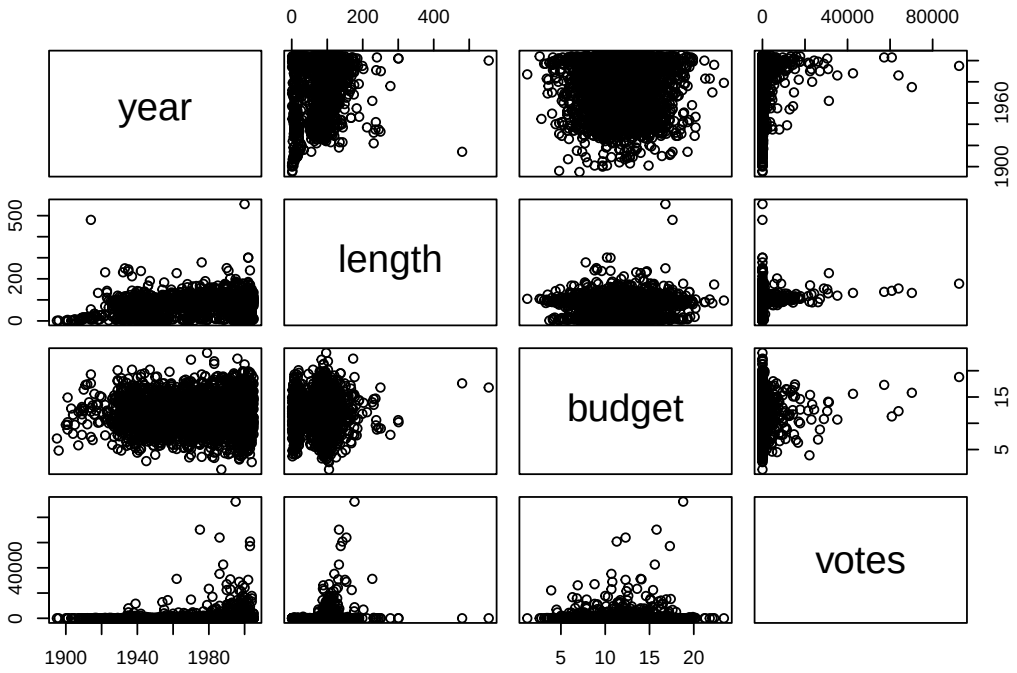

Figure 6: Scatterplot matrix

```
anova(fit, fit4, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: rating2 ~ year + length + budget + votes + genreAction + genreAnimation +
##     genreComedy + genreDocumentary + genreDrama + genreRomance
## Model 2: rating2 ~ year + length + budget + votes + genreAction + genreAnimation +
##     genreComedy + genreDocumentary + genreDrama + genreRomance +
##     year_length + year_votes + length_votes + budget_length +
##     budget_votes
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      2990     1550.9
## 2      2985     1518.1  5   32.809 4.106e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Compared AIC values of these two model, we choose fit4 which has lower AIC value. And due to the result of anova analysis, p-value is less than 0.05 which indicates that there is significant difference between these two model.

Step 6: Model summary

The entire dataset was randomly divided into two groups: training and testing sets, with the random seed specified so that different runs produced the same results. The splitting is done in such a way that the

training set contains 70% of the data. These subsets were used to plot the ROC curve and calculate the AUC value.

```
set.seed(123)
train_index <- sample(1:nrow(films), size = round(nrow(films) * 0.7), replace = FALSE)
train_data <- films[train_index, ]
test_data <- films[-train_index, ]
# Predict the test data using the fitted model
test_data$predicted <- predict(fit4, newdata = test_data, type = "response")
# Calculate the ROC curve and AUC
roc_data <- roc(test_data$rating2, test_data$predicted)
roc_auc <- auc(roc_data)
roc_auc
```

```
## Area under the curve: 0.9502
```

```
plot(roc_data,
  main = "ROC Curve for Movie Rating Model",
  xlab = "False Positive Rate", ylab = "True Positive Rate",
  print.auc = TRUE, auc.polygon = TRUE, grid = c(0.2, 0.2), col = "darkblue"
)
abline(a = 0, b = 1, lty = 2, col = "gray")
```

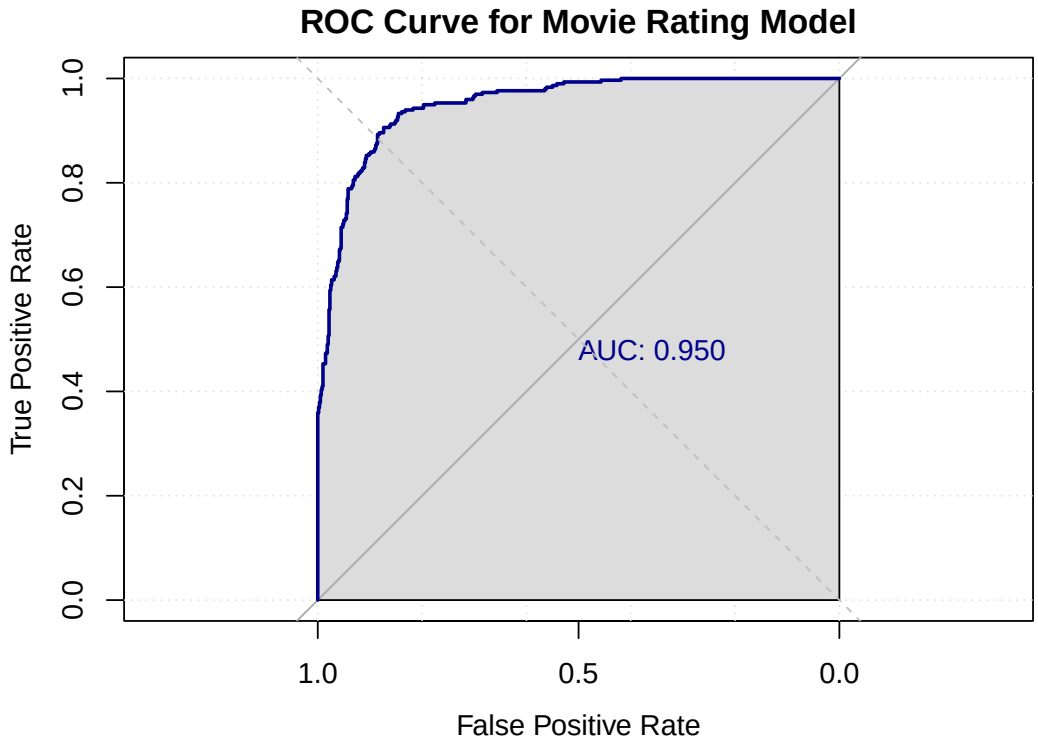

Figure 7: ROC Curve for Movie Rating Model

This result shows that the model works well for predicting the classification of observations in the test dataset, because the area under the ROC curve is 0.9502, which means that the model can distinguish positive and negative examples to a large extent.

```r
# View model results
summary(fit4)
```

```
##
## Call:
## glm(formula = rating2 ~ year + length + budget + votes + genreAction +
##     genreAnimation + genreComedy + genreDocumentary + genreDrama +
##     genreRomance + year_length + year_votes + length_votes +
##     budget_length + budget_votes, family = binomial, data = films)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8881  -0.3261  -0.0997   0.1663   3.0854
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       6.690e+00  1.157e+01   0.578 0.563268
## year             -1.286e-03  5.812e-03  -0.221 0.824911
## length           -2.690e-01  1.564e-01  -1.720 0.085436 .
## budget            2.331e-01  5.772e-02   4.039 5.36e-05 ***
## votes            -4.505e-03  4.230e-03  -1.065 0.286808
## genreAction      -4.289e+00  1.085e+00  -3.953 7.72e-05 ***
## genreAnimation   -5.024e+00  1.096e+00  -4.583 4.58e-06 ***
## genreComedy      -8.073e-01  1.088e+00  -0.742 0.457985
## genreDocumentary  1.252e+00  1.145e+00   1.094 0.274046
## genreDrama       -5.810e+00  1.088e+00  -5.339 9.34e-08 ***
## genreRomance     -5.107e+00  1.317e+00  -3.879 0.000105 ***
## year_length       8.102e-05  7.891e-05   1.027 0.304547
## year_votes        2.298e-06  2.119e-06   1.084 0.278148
## length_votes      1.127e-06  7.332e-07   1.537 0.124369
## budget_length     3.885e-03  7.093e-04   5.477 4.31e-08 ***
## budget_votes     -1.384e-05  5.608e-06  -2.468 0.013591 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3879.3  on 3000  degrees of freedom
## Residual deviance: 1518.1  on 2985  degrees of freedom
## AIC: 1550.1
##
## Number of Fisher Scoring iterations: 7
```

We utilize the *odds ratio* to measure the strength of association of features used in the model and IMDb rating.

```r
# Extract Coefficients and Standard Errors
coef_df <- tidy(fit4, exponentiate = TRUE) %>%
  filter(term != "(Intercept)") %>%
  mutate_if(is.numeric, list(~ round(., 2))) %>%
  mutate(
    lower = estimate - 1.96 * std.error,
```

```
    upper = estimate + 1.96 * std.error
  )
```

```
# plot the coefficients
ggplot(coef_df, aes(x = term, y = estimate)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2) +
  coord_flip() +
  labs(
    title = "Effect of predictors on movie rating",
    x = "", y = "Odds Ratio"
  ) +
  theme_minimal()
```

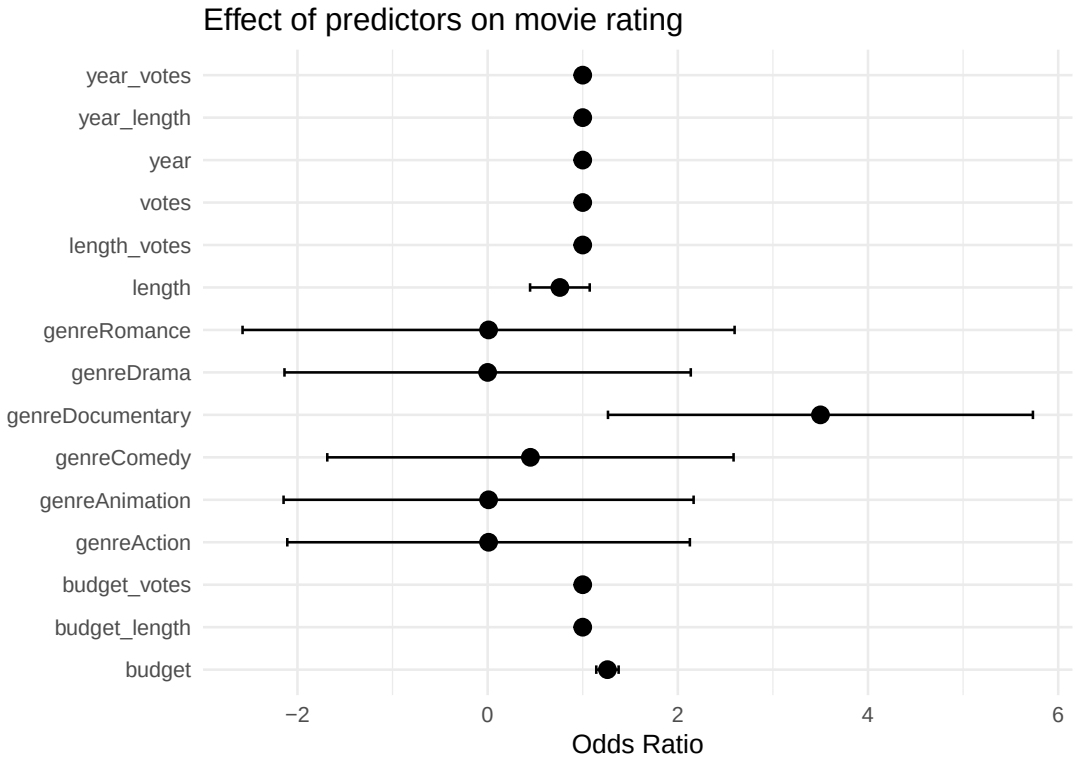

Figure 8: Effect of predictors on movie rating

According to the results, the following conclusions can be drawn:

- `Intercept`: The log odds of the response variable being 1 when all independent variables are 0 in this model is 6.690.
- `Year`: For each increase of 1 year, the log odds of a movie being classified as rating2 = 1 will decrease by 1.286e-03 units.
- `Length`: For each increase of 1 minute in movie length, the log odds of a movie being classified as rating2 = 1 will decrease by 0.2690 units. The p-value obtained is 0.085, which is not significant enough.

- `Budget`: For each increase of 1 unit in budget, the log odds of a movie being classified as rating2 = 1 will increase by 0.2331 units. The p-value is sufficiently low and hence the variable is significant.
- `Votes`: For each increase of 1 vote, the log odds of a movie being classified as rating2 = 1 will decrease by 4.505e-03 units, with a p-value of 0.306. The variable is not significant enough.
- `genreAction, genreAnimation, genreDrama` and `genreRomance`: Compared to genreThriller, the log odds of movies of other genres being classified as rating2 = 1 are -4.289, -5.024, -5.810, and -5.107 units, respectively, with p-values less than 0.05, and therefore significant.
- `genreComedy` and`genreDocumentary`: Compared to genreThriller, the log odds of romance movies being classified as rating2 = 1 is -8.073e-01 and 1.252 units, respectively. They both have p-value greater than 0.05, hence not significant.
- `year_length`: For each increase of 1 year and 1 minute in movie length, the log odds of a movie being classified as rating2 = 1 will increase by 8.102e-05 units, with a p-value of 0.304, which is not significant enough.
- `year_votes`: For each increase of 1 year and 1 vote, the log odds of a movie being classified as rating2 = 1 will increase by 2.298e-06 units. It has p-value of 0.278, which is not significant enough.
- `length_votes`: For each increase of 1 minute in movie length, the log odds of a movie being classified as rating2 = 1 will increase by 1.127e-06 units, with a p-value of 0.124, which is not significant enough.
- `budget_length`: For each increase of 1 unit in budget and 1 minute in movie length, the log odds of a movie being classified as rating2 = 1 will increase by 3.885e-03 units, with a p-value significantly less than 0.05, which is significant.
- `budget_votes`: For each increase of 1 unit in budget and 1 vote, the log odds of a movie being classified as rating2 = 1 will decrease by 1.384e-05 units, with a p-value of 0.014, which is significant.

Step 7: Conclusion

The report analyzes IMDb data to extract features that affect the rating of movies. The model developed for the analysis has an AUC value of 0.95, which is very promising. Based on the results obtained, we can conclude that budget and genre have a significant effect on determining whether the rating of a movie is greater than 7. Besides, budget-length and budget-votes interactions have also influenced the ratings.

For further research, we use stepwise regression method to do model optimization.

```
step.fit4 <- stepAIC(fit4, direction = "both", trace = FALSE)
summary(step.fit4)
```

```
##
## Call:
## glm(formula = rating2 ~ length + budget + genreAction + genreAnimation +
##     genreDocumentary + genreDrama + genreRomance + year_length +
##     length_votes + budget_length + budget_votes, family = binomial,
##     data = films)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8926  -0.3259  -0.0978   0.1824   3.0843
##
## Coefficients:
```

```
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       3.413e+00  7.539e-01   4.527 5.98e-06 ***
## length           -2.655e-01  7.706e-02  -3.445 0.000571 ***
## budget            2.340e-01  5.707e-02   4.099 4.15e-05 ***
## genreAction      -3.500e+00  1.824e-01 -19.188  < 2e-16 ***
## genreAnimation   -4.268e+00  3.623e-01 -11.781  < 2e-16 ***
## genreDocumentary  2.023e+00  3.786e-01   5.345 9.04e-08 ***
## genreDrama       -5.031e+00  2.545e-01 -19.767  < 2e-16 ***
## genreRomance     -4.354e+00  7.989e-01  -5.450 5.05e-08 ***
## year_length       7.910e-05  3.837e-05   2.061 0.039259 *
## length_votes      1.645e-06  6.124e-07   2.687 0.007210 **
## budget_length     3.856e-03  6.955e-04   5.544 2.96e-08 ***
## budget_votes     -1.391e-05  5.955e-06  -2.336 0.019485 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3879.3  on 3000  degrees of freedom
## Residual deviance: 1521.1  on 2989  degrees of freedom
## AIC: 1545.1
##
## Number of Fisher Scoring iterations: 7
```

We get a model with higher AIC value. And all the variables in this model is significant.