Analyzing Properties Influencing IMDB Ratings based on a Logistic Regression

Diwen Jiang, Wei Xiang, Zilu Liu

1 Introduction

IMDB ratings significantly influence audience preferences and the film industry. This study aims to analyze which factors contribute to higher IMDB ratings, specifically whether a movie receives a score greater than 7. Using the IMDB film data set, we employ logistic regression to identify key predictors. The findings provide insights into which factors drive higher ratings by data processing, visualization and model evaluation.

```
#Load the required R packages
library(naniar)
library(dplyr)
library(fastDummies)
library(lmtest)
library(car)
library(pROC)
library(brglm2)
library(ggplot2)
library(GGally)
library(gridExtra)
library(ResourceSelection)
library(caret)
```

2 Data cleaning

Before data modelling, we first check whether there are abnormal or missing values.

```r
#read the data
df <- read.csv("../Data/dataset09.csv", stringsAsFactors = FALSE)
str<-str(df)
```

```
'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 ...
```

```r
summary<-summary(df)
print(summary)
```

```
    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
```

At this stage, we can clearly observe that the **length** variable contains 127 missing values. No other obvious anomalies are present in the data set. However, further discussion and analysis are needed.

2.1 Check for assumption 1 and handle missing values

Therefore, we should handle the missing values.The first step is to check whether the missing values are Missing Completely at Random (MCAR).

```
mcar_test(df["length"])
```

```
# A tibble: 1 x 4
  statistic    df p.value missing.patterns
      <dbl> <dbl>   <dbl>            <int>
1         0     0       1                2
```

p value is equal to 1 (far greater than 0.05), meaning there is no systematic missing pattern. statistic is equal to 0 indicates that the MCAR assumption perfectly matches the missing data pattern. df is equal to 0 may suggest that the dataset has few variables or a simple missing pattern, implying that the missing data is completely random (MCAR) and can be directly removed.

```
df <- df %>% filter(!is.na(length))
```

2.2 Check for assumption 2

The dependent variable must be categorical, with at least one independent variable. Independent variables can be either continuous or categorical.

```
# convert y
df$high_rating <- ifelse(df$rating > 7, 1, 0)
```

Additionally, the "genre" variable is a multinomial categorical variable, so it needs to be converted into to a factor type.

```
df$genre <- factor(df$genre)
df$high_rating <- factor(df$high_rating)
```

Now, all variables meet the requirements of this assumption.

2.2.1 Convert the year column to year_since

```
df$years_since <- 2025 - df$year
```

Each observation must be independent. The categories of categorical variables (including both the dependent and independent variables) must be exhaustive and mutually exclusive.

2.3 Check for assumption 3

Check whether the observations are independent.

The "year" variable may introduce dependency in the data, so we use the Durbin-Watson test to check for autocorrelation:

```
# Set up linear regression model 1
model_1 <- glm(high_rating ~ year + length + budget + votes + genre ,
               data = df,
               family = binomial)

# Durbin-Watson checking
dwtest(model_1)
```

```
    Durbin-Watson test

data:  model_1
DW = 2.0619, p-value = 0.9514
alternative hypothesis: true autocorrelation is greater than 0
```

p-value is equal to 0.9514, which is far greater than 0.05, indicating that the independence assumption is satisfied.

```
# Check for duplicate values and abnormal patterns
dup<-sum(duplicated(df))
```

There are no duplicate values in the dataset.This result Indicates that no films are classified into multiple genres simultaneously.

2.4 Check for assumption 4

Logistic regression requires the assumption that: The minimum sample size should be 15 times the number of independent variables.The cleaned dataset contains 2,874 entries, which clearly supports this assumption.

2.5 Check for assumption 5

There should be no multicollinearity among independent variables.

VIF calculates the correlation between an independent variable and other independent variables.

```
# Set up linear regression model 2
model_2 <- glm(high_rating ~ year + length + budget
             + votes + genre,
             data = df, family = binomial)

# Calculate VIF
vif(model_2)
```

```
           GVIF Df GVIF^(1/(2*Df))
year   1.141535  1        1.068426
length 3.222731  1        1.795197
budget 1.585632  1        1.259219
votes  1.073161  1        1.035935
genre  3.579767  6        1.112127
```

All VIF less than 5, indicating that there is no severe multicollinearity.

2.6 Check for assumption 6

There should be no significant outliers, leverage points, or influential points in the data.

Check for outliers:

```
# Calculate standardized residuals
model_residuals <- rstandard(model_1)

# Identify outliers with an absolute value greater than 3
outliers <- which(abs(model_residuals) > 3)

print(outliers)
```

```
 917 1870 2388
 917 1870 2388
```

We found that rows 917, 1870, and 2388 may contain outliers .Now, print these three rows for inspection.

```
print(df[outliers, ])
```

```
     film_id year length budget votes       genre rating high_rating
917    53015 1971     32   15.8     5 Documentary    7.0           0
1870   17630 1941     45   15.8    10      Comedy    4.2           0
2388   21503 1980    108   10.9   213      Action    8.4           1
     years_since
917           54
1870          84
2388          45
```

After inspection, these three values are considered normal data and will be retained for now.

Check for leverage points:

```
# Calculate leverage values
leverage <- hatvalues(model_1)

# Calculate the leverage threshold
threshold <- 2 * (length(coef(model_1)) / nrow(df))

# Identify points with high leverage
high_leverage <- which(leverage > threshold)
```

A large number of leverage values have been detected, requiring further investigation.

```
lev<-df[high_leverage, c("year", "length", "budget", "votes", "high_rating")]
```

Combine Cook's Distance to check which points are truly influential.

```
cooks_values <- cooks.distance(model_1)
influential_points <- which(cooks_values > 1)

print(intersect(high_leverage, influential_points))
```

```
integer(0)
```

2.7 Output

```
write.csv(x = df,file = "Cleaned_data.csv",row.names = FALSE)
```

No observation is both a high leverage point and a **highly influential point. Since the leverage points and Cook's Distance tests collectively suggest retaining the previously identified outliers, they will be kept. Now, the data supports this assumption.

The assumption that continuous independent variables have a linear relationship with the logit-transformed dependent variable will be further explored in later sections. This concludes the data preprocessing phase.

3 Explanatory data analysis

3.1 Distribution of IMDB rating

```
ggplot(df, aes(x=rating))+
  geom_histogram(binwidth=0.5,fill="blue",alpha=0.6)+
  theme_minimal()
```

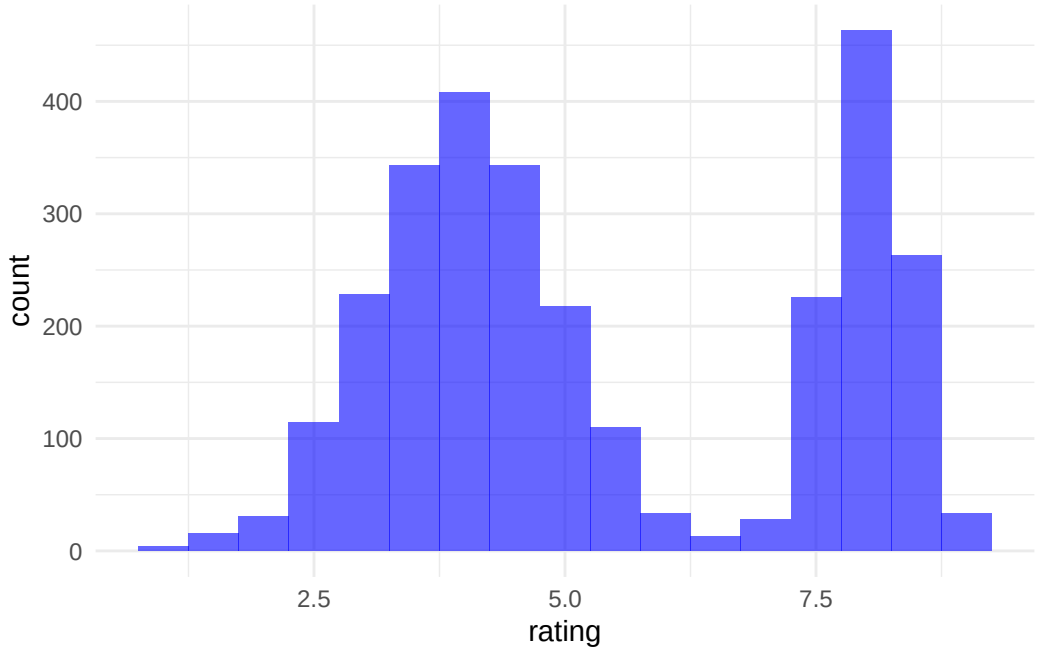

Figure 1: Distribution of IMDB Rating

Most of the IMDB scores are concentrated in 4-8 points, with a slightly right distribution. The dependent variable must be categorical, with at least one independent variable. Independent variables can be either continuous or categorical. We can see that the target response variable should be whether films are rated by IMDb as greater than 7 or not. Therefore, the response variable needs to be transformed.

3.2 key indicators influencing rating

```
# Rating vs Budget
p1<-ggplot(df,aes(x=budget,y=rating))+
  geom_point(alpha=0.5,color="blue")+
  geom_smooth(method="lm",color="red")+
  labs(title="Rating vs Budget",x="Budget",y="IMDB Rating")
# Rating vs Votes
p2<-ggplot(df,aes(x=votes,y=rating))+
  geom_point(alpha=0.5,color="purple")+
  geom_smooth(method="lm",color="red")+
  labs(title="Rating vs Votes",x="Votes",y="IMDB Rating")
# Rating by Genre
```

```
p3<-df %>%
  group_by(genre)%>%
  summarise(avg_rating=mean(rating,na.rm=TRUE))%>%
  ggplot(aes(x=reorder(genre,avg_rating),y=avg_rating,fill=genre))+
  geom_bar(stat="identity",show.legend = FALSE)+
  labs(title="Average Rating by Genre",x="Genre",y="Average Rating")
grid.arrange(p1,p2,p3,nrow=1)
```

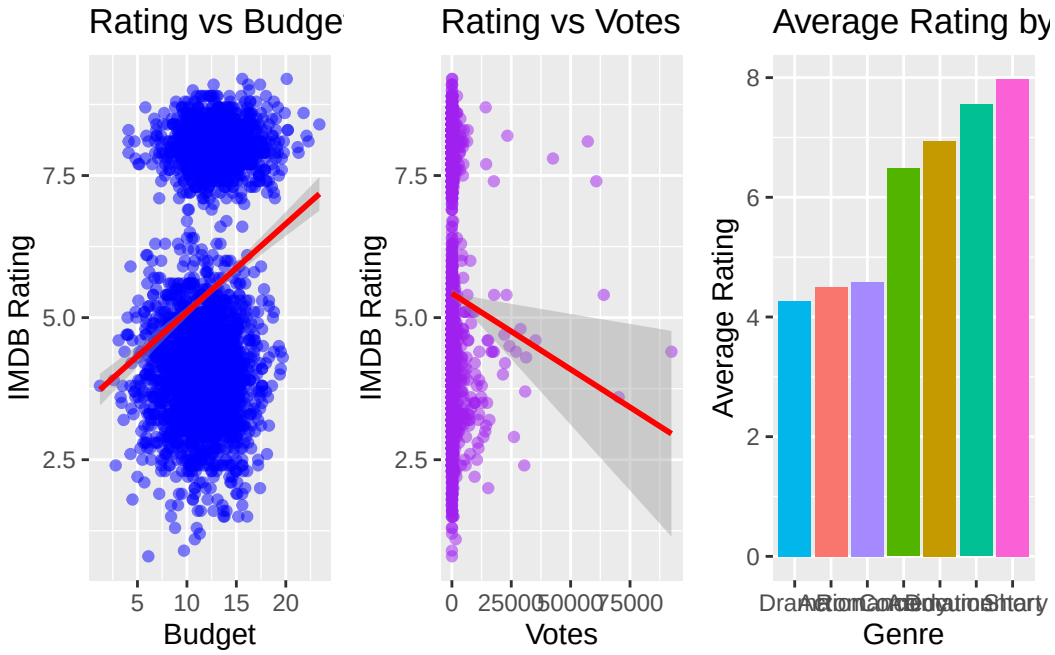

Figure 2: key indicators influencing rating

From the plots, we can see that movies with higher budges tend to have higher ratings, but the relevance is not obvious; There is a positive relationship trend in the score and the number of votes, indicating that the ratings of movies with more votes are usually higher; There are significant differences in the average scores of different film types, and some genres (documentaries and drama films) have higher scores.

```
ggpairs(df,c("length", "budget", "votes","rating"))
```

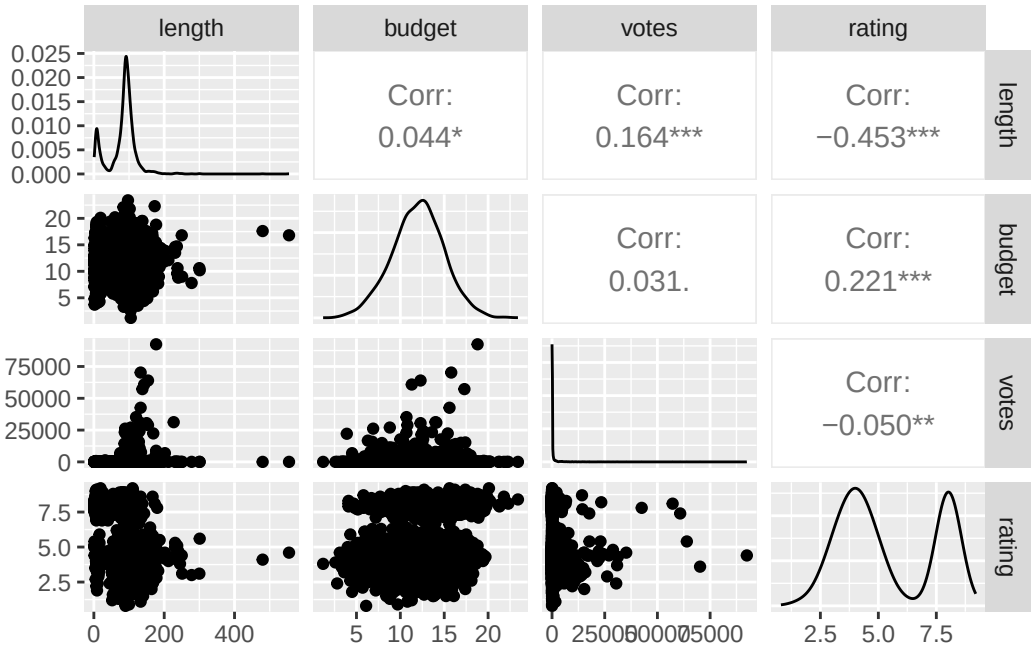

Figure 3: Relationships Between Movie Length, Budget, Votes, and Rating

From the correlation plots, we can see that the negative correlation of length and rating is the most obvious, indicating that long movies tend to have lower scores. In addition, the correlation between all variables is relatively weak, which means they can be used as independent variables in the logical regression model.

4 Formal analysis

4.1 fitting model

```
Call:
glm(formula = high_rating ~ years_since + length + budget + votes +
    genre, family = binomial(link = "logit"), data = df, method = "brglmFit")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9833  -0.3382  -0.1118   0.2010   2.9890

Coefficients:
```

```
                    Estimate  Std. Error z value             Pr(>|z|)
(Intercept)      -3.15209752  0.46616174  -6.762      0.0000000000136 ***
years_since      -0.00641196  0.00290247  -2.209              0.02717 *
length           -0.06142103  0.00367779 -16.701 < 0.0000000000000002 ***
budget            0.51458059  0.02950106  17.443 < 0.0000000000000002 ***
votes             0.00004551  0.00001501   3.031              0.00244 **
genreAnimation   -0.42465600  0.33380024  -1.272              0.20331
genreComedy       3.34577236  0.18017005  18.570 < 0.0000000000000002 ***
genreDocumentary  5.26635707  0.38326713  13.741 < 0.0000000000000002 ***
genreDrama       -1.39594005  0.22907248  -6.094      0.0000000011021 ***
genreRomance     -0.48564856  0.87144016  -0.557              0.57733
genreShort        3.90782772  0.88825448   4.399      0.0000108527515 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3722.9  on 2873  degrees of freedom
Residual deviance: 1469.2  on 2863  degrees of freedom
AIC:  1491.2

Type of estimator: AS_mixed (mixed bias-reducing adjusted score equations)
Number of Fisher Scoring iterations: 5
```

It is observed that in some cases the model produces predicted probabilities of exactly 0 or 1. This may be due to variables such as budget or votes having a very large range of values, which affects numerical computations and causes issues like quasi-complete separation in logistic regression.

```r
#Check which samples have predicted probabilities exactly equal to 0 or 1
pred_probs <- predict(model, type = "response")
summary(pred_probs)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.01688 0.12946 0.35040 0.75963 0.99997
```

```r
#Find and display the samples
df_extreme <- df[pred_probs %in% c(0, 1), ]
print(df_extreme)
```

```
[1] film_id     year        length      budget      votes       genre
```

```
[7] rating      high_rating years_since
<0 rows> (or 0-length row.names)
```

However, "df_extreme" is empty, indicating that there are no samples with predicted probabilities exactly equal to 0 or 1 ? they are merely close to 0 or 1. Therefore, no adjustment is necessary.

4.2 Likelihood Ratio Test

```r
# establish models
null_model <- glm(high_rating ~ 1, family = binomial(), data = df)


# likelihood ratio test
anova_result <- anova(null_model, model, test = "Chisq")
print(anova_result)
```

```
Analysis of Deviance Table

Model 1: high_rating ~ 1
Model 2: high_rating ~ years_since + length + budget + votes + genre
  Resid. Df Resid. Dev Df Deviance               Pr(>Chi)
1      2873     3722.9
2      2863     1469.2 10   2253.7 < 0.00000000000000022 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$\mathbf{p < 0.05}$ indicat that the compregensive model is significantly better than the benchmark model.

4.3 Wald Test

```r
# Wald test Type III test ?
wald_test <- Anova(model, type = "III", test = "Wald")
print(wald_test)
```

```
Analysis of Deviance Table (Type III tests)

Response: high_rating
            Df    Chisq            Pr(>Chisq)
(Intercept)  1  45.7221      0.00000000001363 ***
years_since  1   4.8803              0.027165 *
length       1 278.9080 < 0.00000000000000022 ***
budget       1 304.2506 < 0.00000000000000022 ***
votes        1   9.1887              0.002435 **
genre        6 549.5459 < 0.00000000000000022 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$\mathbf{p < 0.05}$ indicate that all variables are statistically significant.

4.4 Hosmer-Lemeshow test

```r
df$predicted_prob <- predict(model, type = "response")

# Hosmer-Lemeshow test
hl_test <- hoslem.test(as.numeric(df$high_rating), df$predicted_prob, g = 10)

# outcome
print(hl_test)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  as.numeric(df$high_rating), df$predicted_prob
X-squared = 371938, df = 8, p-value < 0.00000000000000022
```

$\mathbf{p > 0.05}$ indicate that a good fit of the model.

4.5 residual analysis

```r
# residuals vs fitted
df_residuals<-data.frame(
  Fitted=fitted(model),
  Residuals=residuals(model)
)
ggplot(df_residuals,aes(x=Fitted,y=Residuals))+
  geom_point(color="blue",alpha=0.6)+
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x="Fitted Values",y="Residuals")
```

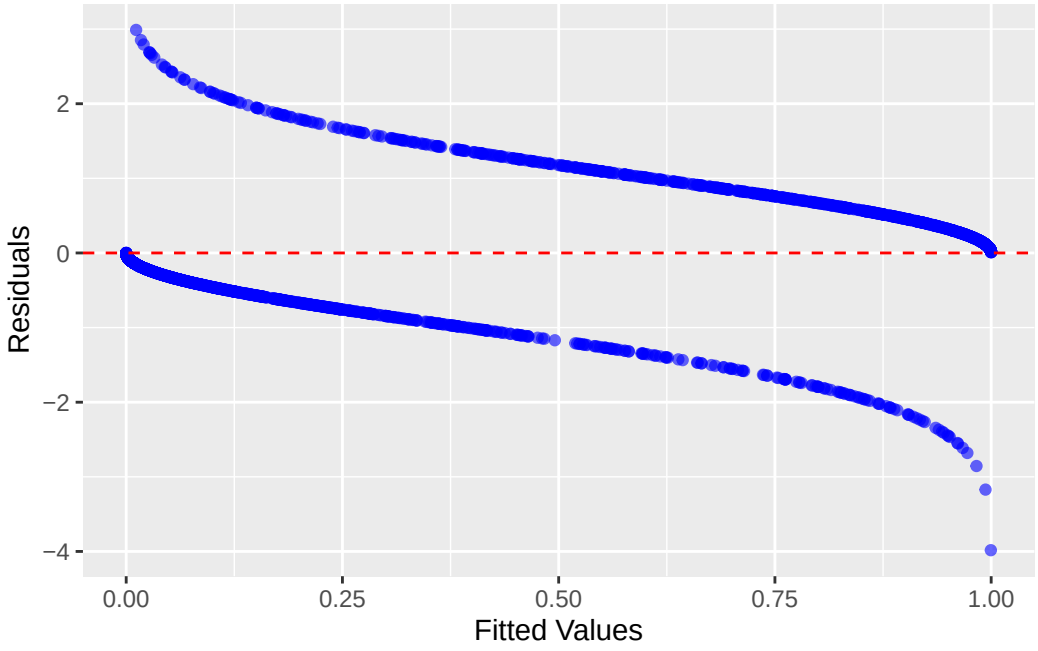

Figure 4: Residuals vs Fitted

From the Residuals vs Fitted plot, we can see that the model may have sysmetric errors, and the fitting effect is not good at extreme values (close to 0 or 1).

```r
# Q-Q plot
qq_data<-data.frame(
  sample=residuals(model),
  theoretical=qqnorm(residuals(model))
)
```

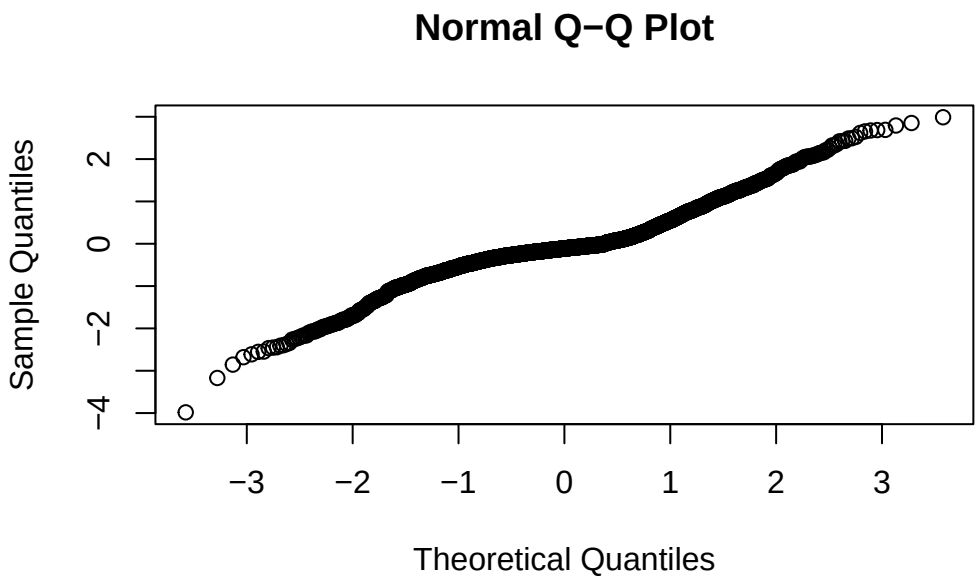

Figure 5: Normal QQ plot

```
ggplot(qq_data,aes(sample=sample))+
  stat_qq()+
  stat_qq_line(color="red")+
  labs(x="Theoretical Quantiles",
       y="Sample Quantiles")
```

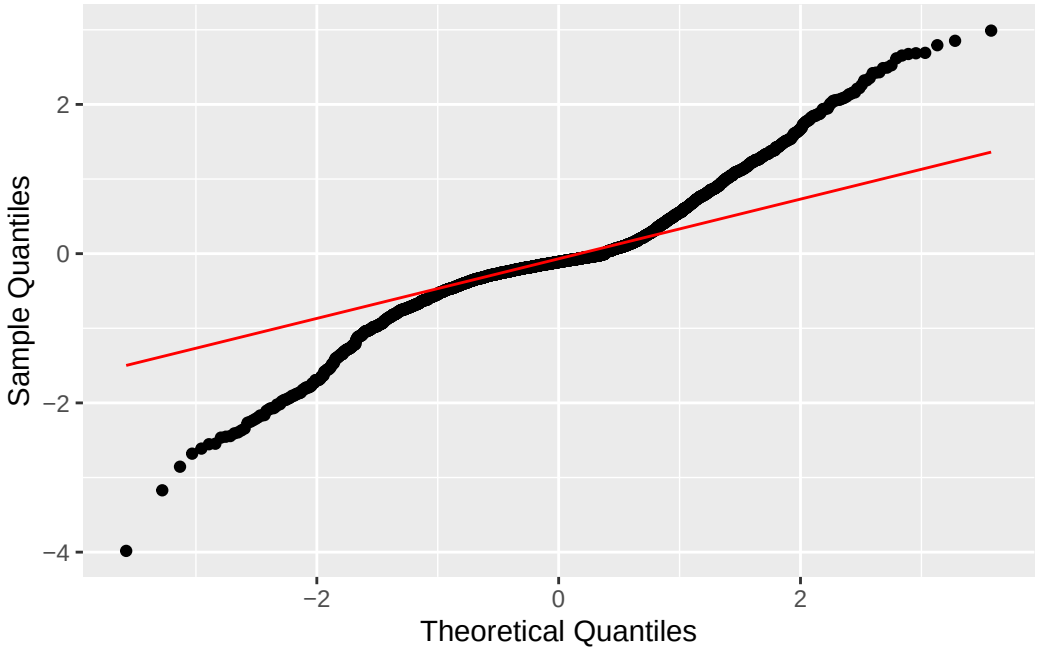

Figure 6: Q-Q Plot of Residuals

From the Q-Q plot, we can see that the distribution of residual deviate from the normal distribution, and there is a heavy tail distribution.

4.6 ROC curve and AUC value

```
# prediction probability
pred_prob <- predict(model, type = "response")

# ROC curve
roc_curve <- roc(response = df$high_rating, predictor = pred_prob)
par(mfrow=c(1,1))
par(pty="s")
plot(roc_curve, col="blue",lwd=2)
auc_value <- auc(roc_curve)
cat("AUC =", auc_value, "\n")
```

```
AUC = 0.956617
```

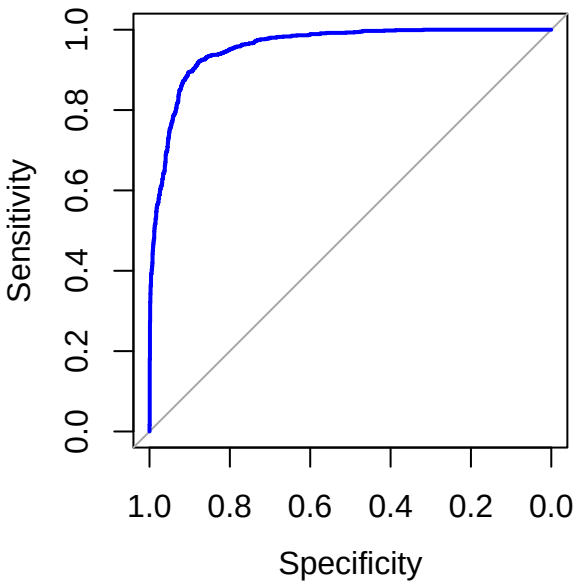

Figure 7: ROC Curve

the AUC value is 0.9566, indicating that the model performance is good, and it can accurately classify movies with IMDb scores above 7.

```
# prediction class
pred_class <- ifelse(pred_prob > 0.5, 1, 0)

# precision
metrics <- confusionMatrix(as.factor(pred_class), df$high_rating)
print(metrics)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1733  169
         1  134  838

               Accuracy : 0.8946
                 95% CI : (0.8828, 0.9056)
    No Information Rate : 0.6496
```

```
   P-Value [Acc > NIR] : < 0.0000000000000002

                  Kappa : 0.7665

 Mcnemar's Test P-Value : 0.05079

            Sensitivity : 0.9282
            Specificity : 0.8322
         Pos Pred Value : 0.9111
         Neg Pred Value : 0.8621
             Prevalence : 0.6496
         Detection Rate : 0.6030
   Detection Prevalence : 0.6618
      Balanced Accuracy : 0.8802

       'Positive' Class : 0
```

From the confusion matrix, we can see that the accuracy score is 0.8946, indicating that the model perform well in classification. What's more, the sensitivity score is 0.9282, indicating that the model perform well in the movies with high rating.

5 Conclusion

In conclusion, this study analyzed factors influencing whether a movie receives an IMDB rating above 7 using logistic regression. Movie length, budget and the number of votes are key indicators. The model achieved high accuracy (0.8946) and AUC (0.9566) demonstrating a strong classification performance. It effectively identified high-rated movies (0.9282 sensitivity) but showed moderate accuracy for low-rated movies (0.8322 specificity).

6 Future Work

In this analysis, we found that the ROC curve demonstrated a good model fit. However, the residual plot and QQ plot revealed some issues. During the model diagnostics, we also observed predicted probabilities that were very close to 0 and 1. We suspect that this may be due to the presence of certain categorical predictors with sparse categories, which could be affecting the model's stability. Also residual analysis suggested that possible improvements such as adding interaction term or log-transformation should be taken. These insights can help filmmakers produce movies and better marketing. Furthermore, applying machine learning techniques

such as using a Random Forest to handle this binary classification problem may yield better performance and more robust results.