An Analysis of Factors Influencing High IMDB Ratings

Group 8

1 Data Description

Source: IMDB film database

Description of variables:

- `film_id`: Unique identifier
- `year`: Year of release
- `length`: Duration (minutes)
- `budget`: Production budget (in $10 million)
- `votes`: Number of viewer votes
- `genre`: Genre of the film
- `rating`: IMDB score from 0–10

Total observations: 2,847 films

Objective of the analysis: To determine which factors of films are associated with an IMDB rating above 7 by using a Generalised Linear Model (GLM).

2 Data Preparing & Cleaning

2.1 Data Cleaning

```r
# Load dataset
raw_data <- read.csv("dataset08.csv")

# Preview the structure of the dataset
glimpse(raw_data)
```

```
Rows: 2,847
Columns: 7
$ film_id <int> 5993, 37190, 43646, 28476, 23975, 50170, 56142, 2287, 17822, 5~
$ year    <int> 1943, 1961, 1987, 1976, 1982, 1936, 1932, 1967, 1983, 2003, 19~
$ length  <int> 65, 87, 79, NA, 88, NA, 75, 100, 82, 15, 86, 96, 150, 86, 102,~
$ budget  <dbl> 15.5, 12.3, 16.4, 12.2, 12.5, 7.0, 12.0, 12.2, 13.4, 13.9, 11.~
$ votes   <int> 42, 6, 161, 5, 97, 146, 14, 8, 141, 20, 121, 119, 5, 14, 48, 1~
$ genre   <chr> "Action", "Drama", "Action", "Documentary", "Action", "Drama",~
$ rating  <dbl> 7.6, 6.0, 7.5, 8.0, 3.5, 4.4, 4.5, 8.4, 3.5, 7.8, 8.2, 2.9, 4.~
```

```r
# Remove rows that have missing values in 'length'variable
clean_data <- raw_data %>%
  filter(!is.na(length))

# Convert 'genre' to factor for categorical analysis
clean_data$genre <- as.factor(clean_data$genre)

# Define a function to create new binary response variable 'rating_above7'
rating_rank <- function(rating_column, threshold = 7){
  ifelse(rating_column > threshold, 1, 0)
}
#check the range of 'year' variable
range(clean_data$year) #we can see that range is between 1898 and 2005
```

```
[1] 1898 2005
```

```r
# Mutate new variables : binary outcome 'rating_above_7' & 'decade_group'
clean_data <- clean_data %>%
  mutate(
    rating_above_7 = rating_rank(rating),
    decade_group = cut(year,
                     breaks = c(1890, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, 2010),
                     labels = c("1890s-1920s", "1930s", "1940s", "1950s", "1960s", "1970s", "
                     right=FALSE)
  )
```

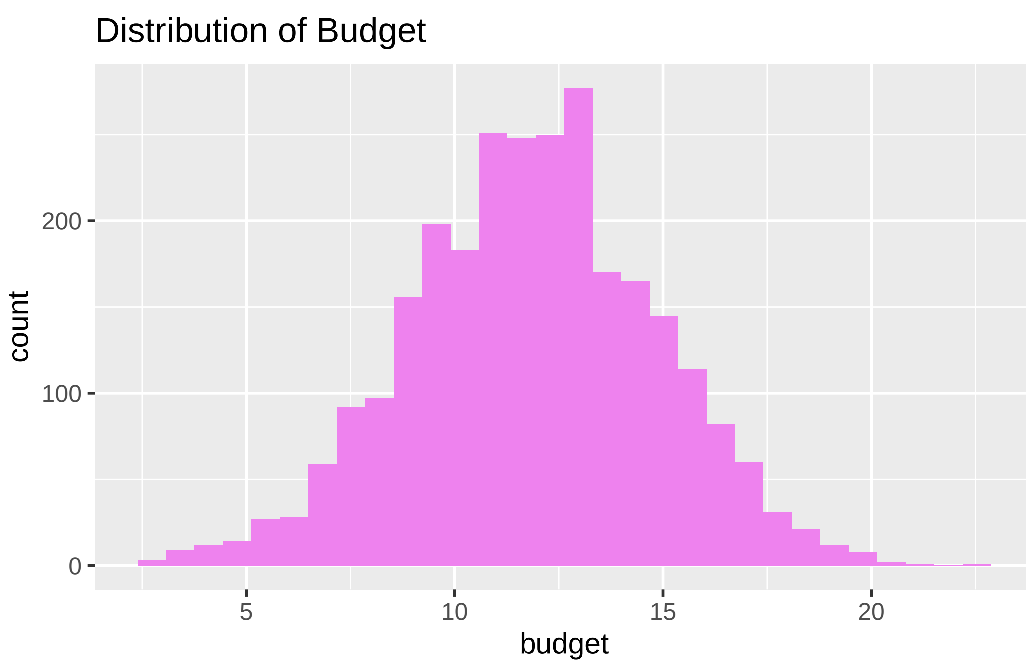
```
#check the missing values in 'budget' & 'votes' variables
sum(is.na(clean_data$budget)) # 0 missing values
```

```
[1] 0
```

```
sum(is.na(clean_data$votes)) # 0 missing values
```

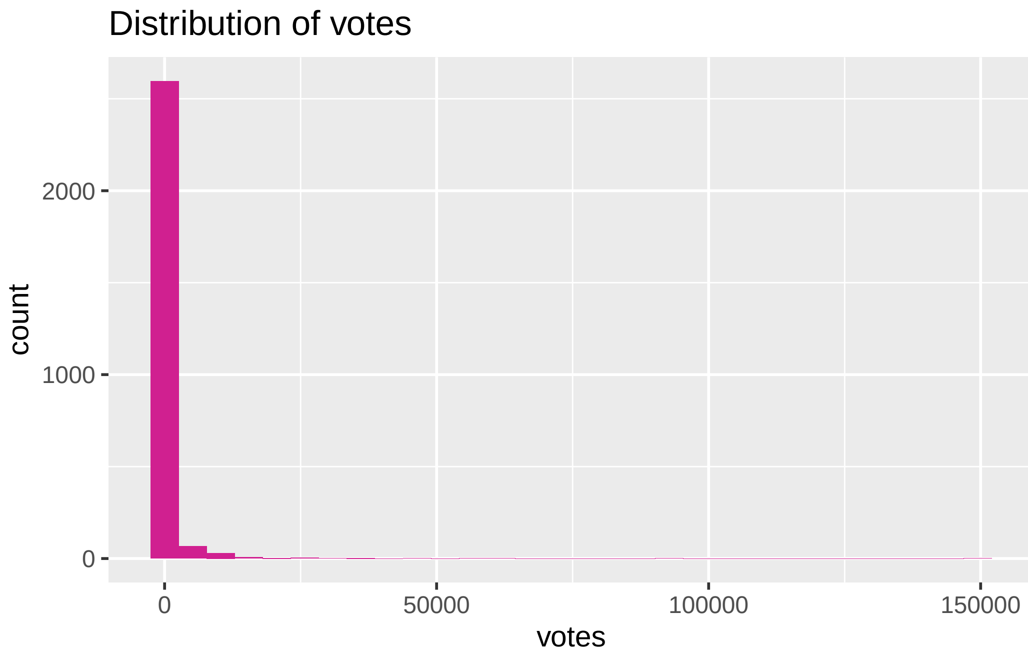
```
[1] 0
```

```
#Visualize the distribution of 'budget'
#If distribution is heavily skewed, log-transformation might be needed
ggplot(clean_data, aes(x = budget)) +
  geom_histogram(bins = 30, fill = "violet") +
  labs(title = "Distribution of Budget")
```




```
#Interpretation:
#The 'budget' variable appears approximately normally distributed.

#Visualize the distribution of 'votes'
ggplot(clean_data, aes(x = votes)) +
  geom_histogram(bins = 30, fill = "violetred") +
  labs(title = "Distribution of votes")
```




```
#Interpretation:
#The 'votes' variable is highly right-skewed.
#A log-transformation should be applied before using this variable in modelling.
```

2.2 Train-Test Splitting

```
set.seed(69)
# From this part, we split into 60/40
# A larger test set (40%) allows for more reliable model evaluation
train_data_index <- sample(seq_len(nrow(clean_data)), size = 0.6 * nrow(clean_data))

train_data <- clean_data[train_data_index, ]
test_data <- clean_data[-train_data_index, ]
```

3 Exploratory Data Analysis (EDA)

4 Statistical Modelling

In this section, we will perform the modelling of the generalised linear model.

From the visualisation results, the votes variables show a right-skewed (skewed distribution), so a log transformation is needed before modelling:

```
#Performs a log transformation on the votes variable
clean_data=clean_data%>%
  mutate(log_votes=log(votes+1)) #Avoiding the log(0) problem
```

Firstly, to test whether year should be put into the model as a continuous or grouped variable, we fitted a model for each and observed their AIC values:

```
#Fitting the GLM logistic regression model
glm_model=glm(rating_above_7~length+log_votes+budget+genre+year,
                 data=clean_data,
                 family=binomial(link="logit"))
summary(glm_model)
```

```

Call:
glm(formula = rating_above_7 ~ length + log_votes + budget +
    genre + year, family = binomial(link = "logit"), data = clean_data)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -22.410737   5.848178  -3.832 0.000127 ***
length            -0.058280   0.003691 -15.788  < 2e-16 ***
log_votes          0.060259   0.040508   1.488 0.136857
budget             0.510924   0.030160  16.941  < 2e-16 ***
genreAnimation    -0.168236   0.327364  -0.514 0.607314
genreComedy        3.069028   0.180969  16.959  < 2e-16 ***
genreDocumentary   5.648565   0.446796  12.642  < 2e-16 ***
genreDrama        -1.568914   0.239578  -6.549 5.81e-11 ***
genreRomance     -14.620723 390.700297  -0.037 0.970149
genreShort         3.978589   0.795084   5.004 5.62e-07 ***
year               0.009445   0.002987   3.162 0.001565 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3470.8  on 2715  degrees of freedom
Residual deviance: 1454.4  on 2705  degrees of freedom
AIC: 1476.4
```

```
Number of Fisher Scoring iterations: 15
```

```r
glm_model1=glm(rating_above_7~length+log_votes+budget+genre+decade_group,
                  data=clean_data,
                  family=binomial(link="logit"))
summary(glm_model1)
```

```
Call:
glm(formula = rating_above_7 ~ length + log_votes + budget +
    genre + decade_group, family = binomial(link = "logit"),
    data = clean_data)

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.377283   0.633780  -6.907 4.96e-12 ***
length             -0.059689   0.003833 -15.573  < 2e-16 ***
log_votes           0.074184   0.041124   1.804   0.0712 .
budget              0.514208   0.030408  16.910  < 2e-16 ***
genreAnimation     -0.231877   0.332119  -0.698   0.4851
genreComedy         3.104698   0.183114  16.955  < 2e-16 ***
genreDocumentary    5.701071   0.451668  12.622  < 2e-16 ***
genreDrama         -1.554340   0.241963  -6.424 1.33e-10 ***
genreRomance      -14.603838 389.730571  -0.037   0.9701
genreShort          4.019767   0.805525   4.990 6.03e-07 ***
decade_group1930s  -0.046639   0.543830  -0.086   0.9317
decade_group1940s   0.455167   0.561532   0.811   0.4176
decade_group1950s   0.475142   0.561143   0.847   0.3971
decade_group1960s   0.935925   0.562839   1.663   0.0963 .
decade_group1970s   0.877435   0.565057   1.553   0.1205
decade_group1980s   0.614581   0.553248   1.111   0.2666
decade_group1990s   0.564173   0.542689   1.040   0.2985
decade_group2000s   0.978918   0.539545   1.814   0.0696 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3470.8  on 2715  degrees of freedom
Residual deviance: 1444.0  on 2698  degrees of freedom
AIC: 1480
```

```
Number of Fisher Scoring iterations: 15
```

```r
AIC(glm_model,glm_model1)
```

```
           df      AIC
glm_model  11 1476.389
glm_model1 18 1479.967
```

From the results, the model with year as a continuous variable has lower AIC values and significant variables, so we will use this model for subsequent stepwise regressions.

```r
#Stepwise regression
best_model=stepAIC(glm_model,direction="both")
```

```
Start:  AIC=1476.39
rating_above_7 ~ length + log_votes + budget + genre + year

            Df Deviance    AIC
<none>           1454.4 1476.4
- log_votes  1   1456.6 1476.6
- year       1   1464.6 1484.6
- length     1   1834.4 1854.4
- budget     1   1878.2 1898.2
- genre      6   2431.2 2441.2
```

```r
summary(best_model)
```

```
Call:
glm(formula = rating_above_7 ~ length + log_votes + budget +
    genre + year, family = binomial(link = "logit"), data = clean_data)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -22.410737   5.848178  -3.832 0.000127 ***
length           -0.058280   0.003691 -15.788  < 2e-16 ***
log_votes         0.060259   0.040508   1.488 0.136857
budget            0.510924   0.030160  16.941  < 2e-16 ***
genreAnimation   -0.168236   0.327364  -0.514 0.607314
```

```
genreComedy         3.069028   0.180969  16.959  < 2e-16 ***
genreDocumentary    5.648565   0.446796  12.642  < 2e-16 ***
genreDrama         -1.568914   0.239578  -6.549 5.81e-11 ***
genreRomance      -14.620723 390.700297  -0.037 0.970149    
genreShort          3.978589   0.795084   5.004 5.62e-07 ***
year                0.009445   0.002987   3.162 0.001565 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3470.8  on 2715  degrees of freedom
Residual deviance: 1454.4  on 2705  degrees of freedom
AIC: 1476.4

Number of Fisher Scoring iterations: 15
```

```r
AIC(glm_model,best_model)
```

```
           df      AIC
glm_model  11 1476.389
best_model 11 1476.389
```

After the stepwise regression method, it is found that the AIC of the model is the same as the original model, but some of the variables of the original model are not significant, after that we will continue to search for the best model by eliminating the non-significant variables.

```r
#Model selection by removing insignificant variables
clean_data_selected=clean_data%>%
  filter(genre%in%c("Comedy","Documentary","Drama","Short"))
glm_model_reduced=glm(rating_above_7~length+log_votes+budget+genre+year,
                      family=binomial(link="logit"),data=clean_data_selected)
summary(glm_model_reduced)
```

```
Call:
glm(formula = rating_above_7 ~ length + log_votes + budget + 
    genre + year, family = binomial(link = "logit"), data = clean_data_selected)

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

```
(Intercept)      -23.792795   7.465561  -3.187  0.00144 **
length            -0.061252   0.004711 -13.001  < 2e-16 ***
log_votes         -0.065077   0.050232  -1.296  0.19513
budget             0.442562   0.037770  11.717  < 2e-16 ***
genreDocumentary   2.391256   0.426869   5.602 2.12e-08 ***
genreDrama        -4.704494   0.291035 -16.165  < 2e-16 ***
genreShort         0.213984   0.812499   0.263  0.79227
year               0.012525   0.003815   3.283  0.00103 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2275.92  on 1687  degrees of freedom
Residual deviance:  843.52  on 1680  degrees of freedom
AIC: 859.52

Number of Fisher Scoring iterations: 7
```

From the results, log_votes and genreShort are still not significant and we will continue with the culling.

```r
#Model selection by removing insignificant variables
clean_data_selected=clean_data%>%
  filter(genre%in%c("Comedy","Documentary","Drama"))
glm_model_reduced1=glm(rating_above_7~length+budget+genre+year,
                       family=binomial(link="logit"),data=clean_data_selected)
summary(glm_model_reduced1)
```

```
Call:
glm(formula = rating_above_7 ~ length + budget + genre + year,
    family = binomial(link = "logit"), data = clean_data_selected)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -22.958874   7.345806  -3.125  0.00178 **
length            -0.064203   0.004709 -13.635  < 2e-16 ***
budget             0.455413   0.038834  11.727  < 2e-16 ***
genreDocumentary   2.501551   0.428456   5.839 5.27e-09 ***
genreDrama        -4.733854   0.294997 -16.047  < 2e-16 ***
year               0.012016   0.003740   3.213  0.00131 **
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2023.54  on 1556  degrees of freedom
Residual deviance:  816.71  on 1551  degrees of freedom
AIC: 828.71

Number of Fisher Scoring iterations: 7
```

```
AIC(glm_model_reduced,glm_model_reduced1)
```

```
                   df      AIC
glm_model_reduced   8 859.5184
glm_model_reduced1  6 828.7069
```

After this exclusion, the resulting model variables were all significant and had the smallest AIC values, and we will use the model for subsequent evaluations.

5 Model Diagnostics

In this section, we will perform model diagnostics on the resulting model.

First we will look at the goodness-of-fit of the model by calculating the pseudo R^2:

```
#Evaluating the goodness-of-fit of the model
#Pseudo R²
pR2=1-(glm_model_reduced1$deviance/glm_model_reduced1$null.deviance)
print(pR2)
```

```
[1] 0.5963962
```

In GLM (logistic regression), the pseudo R^2 can be used to measure the explanatory power: as can be seen from the results, the pseudo R^2 is 0.60, which proves that the model has some explanatory power.

Next, we will perform a residual analysis:

```
#Residual Analysis
#Getting the residuals
residuals_data=data.frame(Index=1:length(residuals(glm_model_reduced1)),
                          Residuals=residuals(glm_model_reduced1,type="deviance"))
```

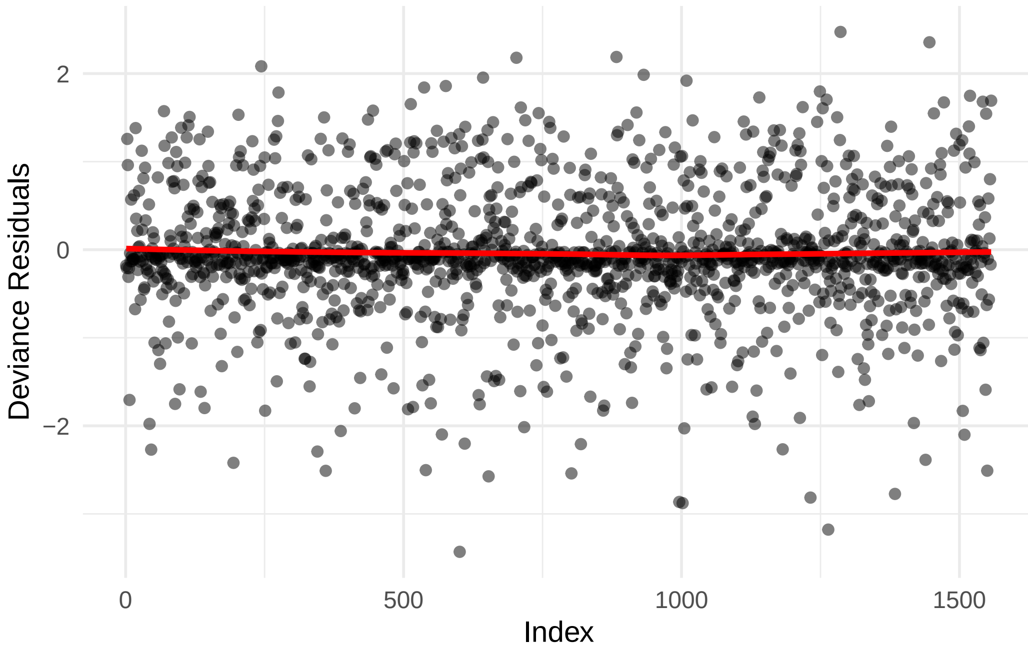


Figure 1: Residual Plot with LOESS Smoothing

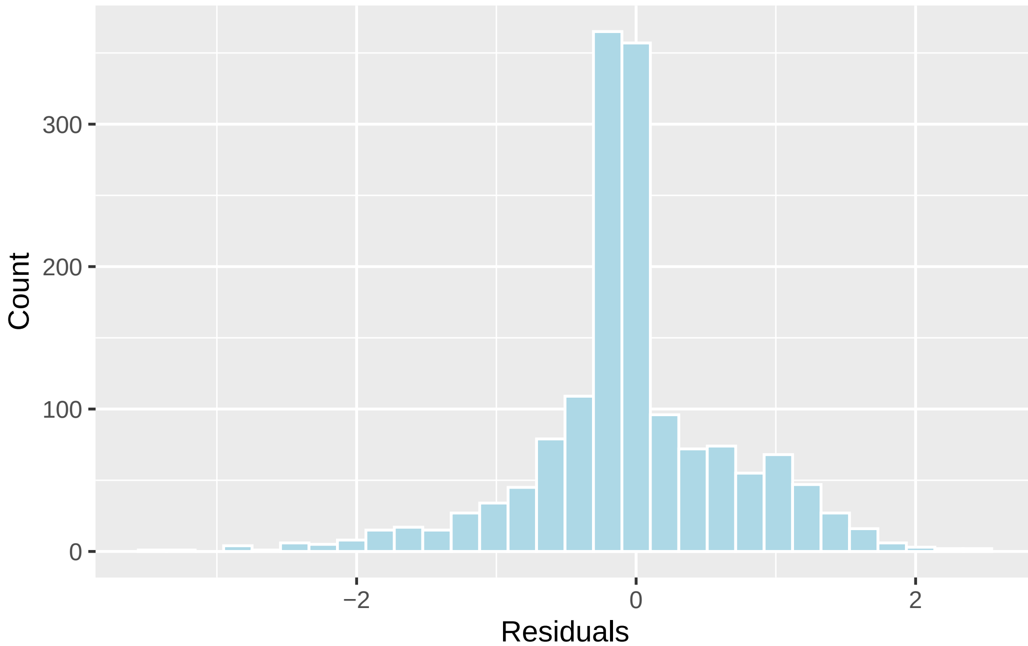


Figure 2: Histogram of Residuals

The two residual plots show that the model is overall good and acceptable.

Next we will calculate the ROC curve and AUC values to observe the predictive power of the model.

```
#Assessment of predictive capacity
#predictive probability
pred_probs=predict(glm_model_reduced1,type="response")
```

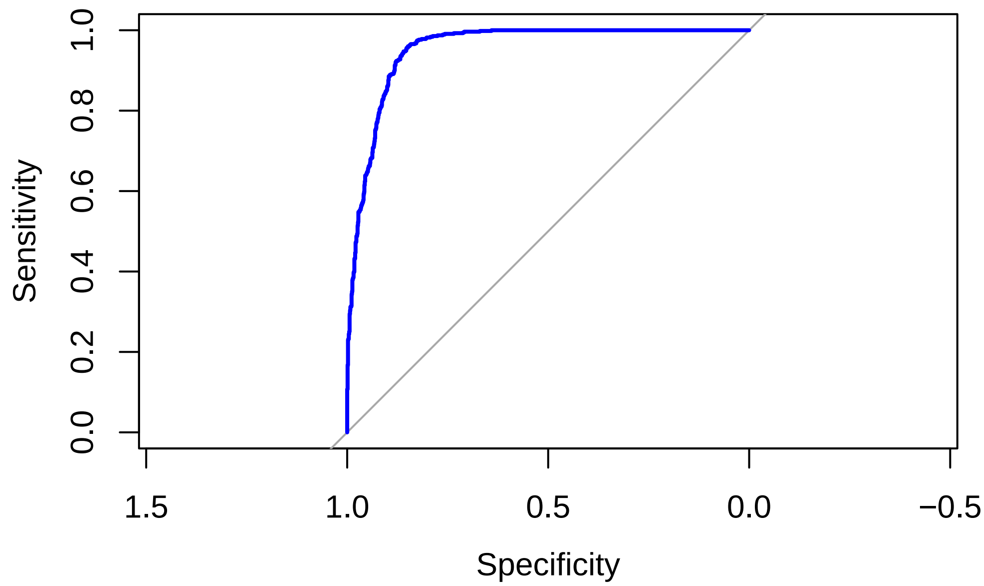


Figure 3: Plot of ROC

```
auc(roc_obj)  #View AUC values
```

```
Area under the curve: 0.9544
```

Area under the curve is 0.9544, which means ths model is good.

```
#Calculate the confusion matrix
pred_class=ifelse(pred_probs>0.5,1,0)
conf_matrix=confusionMatrix(as.factor(pred_class),as.factor(clean_data_selected$rating_above_
print(conf_matrix)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 915  93
         1  91 458

               Accuracy : 0.8818
```

```
                  95% CI : (0.8647, 0.8974)
    No Information Rate : 0.6461
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7414

 Mcnemar's Test P-Value : 0.9412

            Sensitivity : 0.9095
            Specificity : 0.8312
         Pos Pred Value : 0.9077
         Neg Pred Value : 0.8342
             Prevalence : 0.6461
         Detection Rate : 0.5877
   Detection Prevalence : 0.6474
      Balanced Accuracy : 0.8704

       'Positive' Class : 0
```

By calculating the confusion matrix, Accuracy = 88%, the model predicts more accurately overall and the model performs well and can be used for further analysis or optimisation.

```r
#Multicollinearity check
vif(glm_model_reduced1)
```

```
           GVIF Df GVIF^(1/(2*Df))
length 1.602052  1        1.265722
budget 1.303684  1        1.141790
genre  1.692097  2        1.140529
year   1.078064  1        1.038299
```

In the model, the VIF values of all the variables are close to 1, indicating that there is little or no covariance between these variables. Therefore, the model is stable with respect to multicollinearity and no further treatment of covariance is required.

6 Conclusions