Statistical Modelling Assignment: The Oscars Challenge

Introduction

This report used a logistic regression model to predict the winner of the Best Picture award at the 2025 Oscars. The model was built using historical data from 1928 to 2024, with various film attributes including genre and other category nominations as predictors. The dataset was preprocessed in Python by separating 2024 films and converting the dependent variable into binary format.

Task One: Fitting The Model

The logistic regression output containing a few significant predictors is summarised below, the full model can be found in Appendix .1:

Predictor	Estimate	Std. Error	z-value	p-value
Intercept	-9.459	3.063	-3.088	0.002
PGA Award (PGA)	3.571	0.499	7.150	8.69×10^{-13}
Best Director Nomination (Dir)	1.780	0.696	2.557	0.0106
Drama Genre (Gdr)	1.305	0.461	2.830	0.00456

The PGA variable (winning the Producer's Guild of America Award) was interpreted due to its strong correlation with Best Picture win. The PGA coefficient had a highly significant p-value (8.69×10^{-13}) indicating strong predictive power, potentially reflecting the overlap in voting bodies and criteria between the two awards.

The **odds ratio** was calculated: $e^{3.571} = 35.54$, meaning a PGA-winning film is 35 times more likely to win Best Picture, the derivation can be found Appendix .1.1.

The **confidence intervals** were calculated: $CI_{95\%} = [14.00, 100.3]$, since the interval does not contain 0, the effect of the PGA predictor is statistically significant and positive, shown in Appendix .1.2.

Task Two: Model Selection

The full model contained 64 predictors, many with high p-values (> 0.05), suggesting they do not meaningfully contribute to Best Picture prediction. Including all variables risks overfitting, so model selection strategies were used to remove those that don't improve predictive power.

Backwards Elimination with Likelihood Ratio Test (LRT)

Starting with the full model, this method iteratively removes predictors with the highest p-value until the model cannot be simplified further without compromising model fit. Using the drop1 function, single-term deletions were made until no further improvements could be made to the AIC score. The final model contained 15 predictors and had an AIC of 313.9 and a BIC of 384.3, as shown in Appendix .2.1 and Appendix .2.5.

Stepwise Search with Bayesian Information Criterion (BIC)

This method combines forward and backward steps, adding or removing predictors based on BIC to find the best model. Using R's step function, the final model had a BIC of 350.1 and included 3 predictors, all with p-values below 0.05. However, BIC applies a heavy penalty for complexity, resulting in an AIC score of 332.5, higher than the model selected by backward elimination, this can be seen in Appendix .2.2 and Appendix .2.5.

Stepwise Search with Akaike Information Criterion (AIC)

This strategy is similar to the previous, except it uses AIC rather than BIC to assess model fit. AIC applies a lower penalty for complexity, allowing for the inclusion of more predictors. Using the step function, the final model was the same as the one selected by backwards elimination indicating that the model is robust, Appendix .2.3.

Model Comparison

The models from each selection strategy were compared to the full model using the ANOVA function to determine if the simplified models caused a statistically significant reduction in fit. In these tests the null hypothesis states that the additional predictors in the full model improve model fit. The results are summarised below:

Model	Residual DF	Residual Dev	Df Deviance	p-value
AIC	588	281.92		
Full	539	258.01	49	0.999

Model	Residual DF	Residual Dev	Df Deviance	p-value
BIC	600	324.47		
Full	539	258.01	66.46	0.295

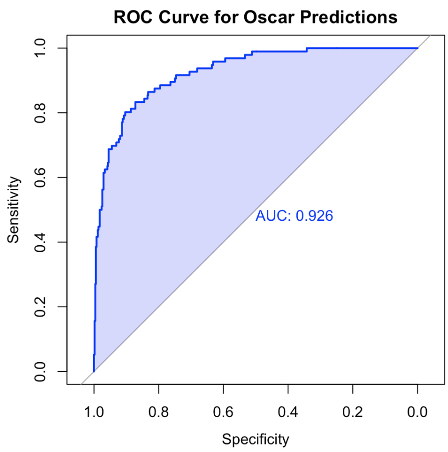
Both p-values are large, suggesting the additional predictors in the full model do not improve model fit, so the simplified models do not reduce the model's ability to fit the data. Next, the models from the selection strategies were compared, to evaluate whether the additional predictors in the more complex model improved model fit. The results are as follows:

Model	Residual DF	Residual Dev	Df Deviance	p-value
BIC	600	324.47		
AIC	588	281.92	42.55	2.69×10^{-5}

Using this information, it can be inferred that the additional predictors included in the model selected by backward elimination and AIC comparison significantly improves model fit. Additionally, the simpler model reduces the deviance by 42.55, indicating that it fits the data substantially better. For these reasons, it is chosen as the final model. The derivation of these tables is shown in Appendix .2.4.

Task Three: Model Performance Evaluation

The final model's performance was evaluated using the ROC curve, which plots Sensitivity vs. False Positive Rate. The Area Under the Curve (AUC) was calculated along with the optimal threshold and model sensitivity.



The high **AUC** (0.926) indicates that the model has excellent discriminatory power, effectively distinguishing between positive and negative classes.

$$\textbf{Optimal Threshold} = 0.1799 \quad (1)$$

This threshold is chosen to maximise sensitivity while maintaining a reasonable balance with specificity, the calculations can be seen in more detail in Appendix .3.1 and Appendix .3.2.

The confusion matrix for the final model at the optimal threshold is:

	Actual Negative	Actual Positive
Predicted Negative	456	19
Predicted Positive	49	77

$$\textbf{Sensitivity} = \frac{TP}{TP + FN} = \frac{77}{77 + 19} = 0.802083 \tag{2}$$

The model has a sensitivity of approximately 0.8021 indicating that it correctly identifies **80.21%** of actual positive cases, as shown in Appendix .3.2.

Task Four: Predicted Probability of Winning

Based on my final model, the predicted probabilities of winning for each nominee for this year's Best Picture category are as follows:

Nominee	Predicted Probability
A Complete Unknown	0.01301
Anora	0.8685
Conclave	0.01055
Dune: Part Two	8.997×10^{-4}
Emilia Perez	0.01971
I'm Still here	6.956×10^{-3}
Nickle Boys	0.04741
The Brutalist	0.01264
The Substance	1.270×10^{-3}
Wicked	0.01906

The sum of these probabilities is equal to 1, ensuring that the model's predictions are correctly normalised, further information on this calculation can be found in Appendix .4.

With a predicted probability of 0.8685 (86.85%), Anora was the overwhelming favourite to win Best Picture, significantly performing all other nominees. This prediction was validated as Anora won Best Picture at the 2025 Academy Awards.

Task Five: Alternative Models

Logistic regression is an unsuitable method for predicting Best Picture winner as it is inherently a binary classification model and treats each nominee as an independent observation. It fails to capture the constraint that exactly one film wins per year and does not account for the comparative nature of the selection process. Additionally, logistic regression estimates individual probabilities without enforcing that the sum of probabilities in a given year is 1.

A more appropriate model is multinomial regression, which extends logistic regression to multi-class settings by jointly modelling all nominees within a given year. The model works by assigning score s_i to each nominee and transforms these score into probabilities p_i using the softmax function:

$$p_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)} \tag{3}$$

where the summation is taken over all nominees j in the same year. This normalisation ensures that the predicted probabilities sum to 1, aligning with the constraint that exactly one nominee wins. By modelling the probability distribution over all nominees, multinomial regression captures the relative nature of the Oscar selection process, making it a more suitable choice.

R Code and Model Output

.1 Task One

```
> oscars<-read.csv("OscarsDataCleaned.csv", heade r= TRUE)
> oscarsCh$<-as.factor(oscars$Ch)
> oscars<-subset(oscars, select = -c(Year, Name))


> full_model<-glm(Ch ~ ., data=oscars, family = binomial)
> summary(full_model)

Call:
glm(formula = Ch ~ ., family = binomial, data = oscars)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.459e+00  3.063e+00  -3.088  0.00202 **
Nom         -1.143e-02  3.910e-01  -0.029  0.97668
Dir          1.780e+00  6.959e-01   2.557  0.01055 *
Aml         -1.211e-02  4.934e-01  -0.025  0.98041
Afl          2.905e-01  5.629e-01   0.516  0.60578
Ams          6.048e-01  4.765e-01   1.269  0.20438
Afs          1.481e-01  4.990e-01   0.297  0.76660
Scr          3.845e-01  6.442e-01   0.597  0.55063
Cin         -1.487e-01  5.846e-01  -0.254  0.79921
Art          2.350e-01  5.889e-01   0.399  0.68985
Cos         -5.047e-01  6.502e-01  -0.776  0.43754
Sco          2.931e-01  5.493e-01   0.534  0.59357
Son         -1.610e-01  8.498e-01  -0.190  0.84970
Edi          1.207e+00  5.636e-01   2.142  0.03223 *
Sou         -5.270e-01  6.362e-01  -0.828  0.40744
For          4.318e-01  1.465e+00   0.295  0.76824
Anf          1.065e+00  4.817e+03   0.000  0.99982
Eff          2.175e-01  6.460e-01   0.337  0.73630
Mak          1.457e+00  9.438e-01   1.544  0.12257
Dan          2.769e+00  1.628e+00   1.701  0.08895 .
AD           1.683e+00  1.314e+00   1.281  0.20003
Gdr          1.305e+00  4.610e-01   2.830  0.00465 **
Gmc         -1.290e-01  7.790e-01  -0.166  0.86850
Gd          -2.950e+00  1.770e+00  -1.667  0.09556 .
Gm1          1.467e+00  1.269e+00   1.157  0.24740
Gm2         -1.193e+00  3.497e+00  -0.341  0.73293
Gf1         -1.558e+00  1.807e+00  -0.862  0.38856
Gf2         -1.435e+01  1.772e+03  -0.008  0.99354
PGA          3.571e+00  4.994e-01   7.150 8.69e-13 ***
DGA          1.905e+00  1.331e+00   1.431  0.15240
Action      -6.718e-01  7.969e-01  -0.843  0.39925
Adventure   -2.477e-01  7.548e-01  -0.328  0.74279
Animation   -1.172e+01  3.956e+03  -0.003  0.99764
Biography   -5.795e-01  6.509e-01  -0.890  0.37334
Comedy      -1.364e-01  5.373e-01  -0.254  0.79953
Crime        1.021e+00  6.313e-01   1.617  0.10594
Docu        -1.333e+01  3.956e+03  -0.003  0.99731
Drama       -9.871e-01  6.238e-01  -1.582  0.11359
Family       1.222e+00  8.718e-01   1.402  0.16102
Fantasy     -1.039e+00  1.169e+00  -0.888  0.37443
Film.noir   -6.244e-01  1.425e+00  -0.438  0.66130
History      3.740e-01  7.037e-01   0.531  0.59508
Horror      -8.303e-01  2.053e+00  -0.404  0.68593
Music        7.516e-01  9.649e-01   0.779  0.43604
Musical      8.186e-01  8.582e-01   0.954  0.34012
```

```
Mystery       6.285e-01  7.954e-01   0.790  0.42944
Romance       4.362e-01  4.183e-01   1.043  0.29704
SciFi        -1.084e+00  1.637e+00  -0.662  0.50792
Sport         5.060e-01  1.178e+00   0.430  0.66745
Thriller     -8.074e-01  7.014e-01  -1.151  0.24973
War           9.575e-01  6.376e-01   1.502  0.13317
Western      -3.627e-02  9.434e-01  -0.038  0.96933
Length        1.789e-03  8.491e-03   0.211  0.83311
Days          2.381e-03  1.621e-03   1.469  0.14174
G            -1.985e+00  2.029e+00  -0.978  0.32803
PG           -1.135e+00  6.883e-01  -1.649  0.09922 .
PG13         -1.424e+00  8.666e-01  -1.643  0.10038
R            -1.438e+00  7.088e-01  -2.030  0.04241 *
U            -1.384e+01  1.989e+03  -0.007  0.99445
Ebert         1.325e-01  1.683e-01   0.787  0.43115
NYFCC         1.660e-01  4.684e-01   0.354  0.72304
LAFCA        -8.886e-01  7.154e-01  -1.242  0.21423
NSFC          1.880e+00  7.502e-01   2.506  0.01221 *
NBR          -1.391e-01  4.917e-01  -0.283  0.77719
WR            5.265e-01  3.997e-01   1.317  0.18771

Null deviance: 528.99  on 603  degrees of freedom
Residual deviance: 258.01  on 539  degrees of freedom
AIC: 388.01
Number of Fisher Scoring iterations: 16
```

.1.1 Odds Ratio

```
> exp(coef(full_model)["PGA"])

PGA
35.53586
```

.1.2 Confidence Intervals

```
> exp(confint(full_model, parm = "PGA")

   2.5 %    97.5 %
13.99565  100.27270
```

.2 Task Two

.2.1 Backward Elimination with LRT

```
>drop1(full_model,test = "LRT")

>fit1 <-update(full_model, .~. - Anf)
>drop1(fit1, test = "LRT")
>fit2 <-update(fit1, .~. - Aml)
>drop1(fit2, test = "LRT")
>fit3 <-update(fit2, .~. - Western)
>drop1(fit3, test = "LRT")
>fit4 <-update(fit3, .~. - Nom)
>drop1(fit4, test = "LRT")
>fit5 <-update(fit4, .~. - Animation)
>drop1(fit5, test = "LRT")
>fit6 <-update(fit5, .~. - Gmc)
>drop1(fit6, test = "LRT")
>fit7 <-update(fit6, .~. - Docu)
```

```
>drop1(fit7, test = "LRT")
>fit8 <-update(fit7, .~. - Length)
>drop1(fit8, test = "LRT")
>fit9 <-update(fit8, .~. - Son)
>drop1(fit9, test = "LRT")
>fit10 <-update(fit9, .~. - NBR)
>drop1(fit10, test = "LRT")
?fit11 <-update(fit10, .~. - NYFCC)
>drop1(fit11, test = "LRT")
>fit12 <-update(fit11, .~. - Adventure)
>drop1(fit12, test = "LRT")
>fit13 <-update(fit12, .~. - For)
>drop1(fit13, test = "LRT")
>fit14 <-update(fit13, .~. - Gm2)
>drop1(fit15, test = "LRT")
>fit15 <-update(fit14, .~. - Sport)
>drop1(fit14, tes = "LRT")
>fit16 <-update(fit15, .~. - Afs)
>drop1(fit16, test = "LRT")
>fit17 <-update(fit16, .~. - Horror)
>drop1(fit17, test = "LRT")
>fit18 <-update(fit17, .~. - U)
>drop1(fit18, test = "LRT")
>fit19 <-update(fit18, .~. - Cin)
>drop1(fit19, test = "LRT")
>fit20 <-update(fit19, .~. - Art)
>drop1(fit20, test = "LRT")
>fit21 <-update(fit20, .~. - Comedy)
>drop1(fit21, test = "LRT")
>fit22 <-update(fit21, .~. - Eff)
>drop1(fit22, test = "LRT")
>fit23 <-update(fit22, .~. - Film.noir)
>drop1(fit23, test = "LRT")
>fit24 <-update(fit23, .~. - Ebert)
>drop1(fit24, test = "LRT")
>fit25 <-update(fit24, .~. - History)
>drop1(fit25, test = "LRT")
>fit26 <-update(fit25, .~. - Music)
>drop1(fit26, test = "LRT")
>fit27 <-update(fit26, .~. - Biography)
>drop1(fit27, test = "LRT")
>fit28 <-update(fit27, .~. - Gf2)
>drop1(fit28, test = "LRT")
>fit29 <-update(fit28, .~. - Afl)
>drop1(fit29, test = "LRT")
>fit30 <-update(fit29, .~. - Scr)
>drop1(fit30, test = "LRT")
>fit31 <-update(fit30, .~. - Sco)
>drop1(fit31, test = "LRT")
>fit32 <-update(fit31, .~. - Mystery)
>drop1(fit32, test = "LRT")
>fit33 <-update(fit32, .~. - Musical)
>drop1(fit33, test = "LRT")
>fit34 <-update(fit33, .~. - Cos)
>drop1(fit34, test = "LRT")
>fit35 <-update(fit34, .~. - Thriller)
>drop1(fit35, test = "LRT")
>fit36 <-update(fit35, .~. - LAFCA)
>drop1(fit36, test = "LRT")
>fit37 <-update(fit36, .~. - Gm1)
>drop1(fit37, test = "LRT")
>fit38 <-update(fit37, .~. - G)
>drop1(fit38, test = "LRT")
```

```
>fit39 <-update(fit38, .~. - Sou)
>drop1(fit39, test = "LRT")
>fit40 <-update(fit39, .~. - Crime)
>drop1(fit40, test = "LRT")
>fit41 <-update(fit40, .~. - AD)
>drop1(fit41, test = "LRT")
>fit42 <-update(fit41, .~. - Family)
>drop1(fit42, test = "LRT")
>fit43 <-update(fit42, .~. - Fantasy)
>drop1(fit43, test = "LRT")
>fit44 <-update(fit43, .~. - War)
>drop1(fit44, test = "LRT")
>fit45 <-update(fit44, .~. - Drama)
>drop1(fit45, test = "LRT")
>fit46 <-update(fit45, .~. - Action)
>drop1(fit46, test = "LRT")
>fit47 <-update(fit46, .~. - Mak)
>drop1(fit47, test = "LRT")
>fit48 <-update(fit47, .~. - Ams)
>drop1(fit48, test = "LRT")
>fit49 <-update(fit48, .~. - Romance)
>drop1(fit49, test = "LRT")
>fit50 <-update(fit49, .~. - WR)
>drop1(fit50, test = "LRT")
>fit51 <-update(fit50, .~. - PG13)
>drop1(fit51, test = "LRT")
>fit52 <-update(fit51, .~. - PG)
>drop1(fit52, test = "LRT")
>fit53 <-update(fit52, .~. - R)
>drop1(fit53, test = "LRT")
>fit54 <-update(fit53, .~. - DGA)
>drop1(fit54, test = "LRT")
>fit55 <-update(fit54, .~. - Gd)
>drop1(fit55, test = "LRT")
>fit56 <-update(fit55, .~. - NSFC)
>drop1(fit56, test = "LRT")

> AIC(fit47, fit48, fit49, fit50, fit51, fit52, fit53, fit54,
   fit55, fit56)

      df      AIC
fit47 17 314.3500
fit48 16 313.9184
fit49 15 314.1598
fit50 14 314.8799
fit51 13 315.2683
fit52 12 315.2178
fit53 11 316.0977
fit54 10 317.9327
fit55  9 317.8266
fit56  8 318.9511

> summary(fit48)

Call:
glm(formula = Ch ~ Dir + Edi + Dan + Gdr + Gd + PGA + DGA +
   Romance +
    SciFi + Days + PG + PG13 + R + NSFC + WR, family = binomial,
    data = oscars)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.428192   2.462220  -3.829 0.000129 ***
```

```
Dir          1.574330   0.494778   3.182 0.001463 **
Edi          1.169340   0.373104   3.134 0.001724 **
Dan          3.131095   1.300739   2.407 0.016077 *
Gdr          1.170446   0.393905   2.971 0.002965 **
Gd          -2.476237   1.353463  -1.830 0.067316 .
PGA          3.320186   0.408168   8.134 4.14e-16 ***
DGA          1.733160   0.996429   1.739 0.081969 .
Romance      0.545207   0.363354   1.500 0.133490
SciFi       -2.375553   1.384500  -1.716 0.086195 .
Days         0.002247   0.001310   1.715 0.086365 .
PG          -0.865544   0.546307  -1.584 0.113113
PG13        -0.980035   0.580730  -1.688 0.091489 .
R           -1.041853   0.446238  -2.335 0.019557 *
NSFC         1.348854   0.523822   2.575 0.010023 *
WR           0.590864   0.320739   1.842 0.065447 .
---
Signif. codes:  0   ***   0.001   **   0.01   *   0.05   .
      0.1        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 528.99  on 603  degrees of freedom
Residual deviance: 281.92  on 588  degrees of freedom
AIC: 313.92

Number of Fisher Scoring iterations: 6
```

.2.2 Stepwise Search with BIC

```
> step_bic <- step(full_model, direction = "both", k = log(nrow(
   oscars)))
> summary(step_bic)
Call:
glm(formula = Ch ~ Dir + Edi + PGA, family = binomial, data =
   oscars)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -4.6961     0.4999  -9.394  < 2e-16 ***
Dir           1.9363     0.4712   4.109 3.97e-05 ***
Edi           1.1918     0.3303   3.608 0.000309 ***
PGA           3.1721     0.3320   9.554  < 2e-16 ***
---
Signif. codes:  0   ***   0.001   **   0.01   *   0.05   .
      0.1        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 528.99  on 603  degrees of freedom
Residual deviance: 324.47  on 600  degrees of freedom
AIC: 332.47

Number of Fisher Scoring iterations: 6
```

.2.3 2.3. Stepwise Search with AIC

```
> step_aic <- step(full_model, direction = "both", k = 2)
> summary(step_aic)

Call:
```

```
glm(formula = Ch ~ Dir + Edi + Dan + Gdr + Gd + PGA + DGA +
    Romance +
     SciFi + Days + PG + PG13 + R + NSFC + WR, family = binomial,
     data = oscars)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -9.428192   2.462220  -3.829 0.000129 ***
Dir          1.574330   0.494778   3.182 0.001463 **
Edi          1.169340   0.373104   3.134 0.001724 **
Dan          3.131095   1.300739   2.407 0.016077 *
Gdr          1.170446   0.393905   2.971 0.002965 **
Gd          -2.476237   1.353463  -1.830 0.067316 .
PGA          3.320186   0.408168   8.134 4.14e-16 ***
DGA          1.733160   0.996429   1.739 0.081969 .
Romance      0.545207   0.363354   1.500 0.133490
SciFi       -2.375553   1.384500  -1.716 0.086195 .
Days         0.002247   0.001310   1.715 0.086365 .
PG          -0.865544   0.546307  -1.584 0.113113
PG13        -0.980035   0.580730  -1.688 0.091489 .
R           -1.041853   0.446238  -2.335 0.019557 *
NSFC         1.348854   0.523822   2.575 0.010023 *
WR           0.590864   0.320739   1.842 0.065447 .
---
Signif. codes:  0   ***   0.001   **   0.01   *   0.05   .
       0.1        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 528.99  on 603  degrees of freedom
Residual deviance: 281.92  on 588  degrees of freedom
AIC: 313.92

Number of Fisher Scoring iterations: 6
```

.2.4 2.4. ANOVA Tables

```
> anova(step_aic, full_model, test = "LRT")
Analysis of Deviance Table

Model 1: Ch ~ Dir + Edi + Dan + Gdr + Gd + PGA + DGA + Romance +
    SciFi +
     Days + PG + PG13 + R + NSFC + WR
Model 2: Ch ~ Nom + Dir + Aml + Afl + Ams + Afs + Scr + Cin + Art
    + Cos +
     Sco + Son + Edi + Sou + For + Anf + Eff + Mak + Dan + AD +
     Gdr + Gmc + Gd + Gm1 + Gm2 + Gf1 + Gf2 + PGA + DGA + Action +
     Adventure + Animation + Biography + Comedy + Crime + Docu +
     Drama + Family + Fantasy + Film.noir + History + Horror +
     Music + Musical + Mystery + Romance + SciFi + Sport + Thriller
     +
     War + Western + Length + Days + G + PG + PG13 + R + U + Ebert
    +
     NYFCC + LAFCA + NSFC + NBR + WR
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       588     281.92
2       539     258.01 49    23.91    0.999


> anova(step_bic, full_model, test = "LRT")
Analysis of Deviance Table
```

```
Model 1: Ch ~ Dir + Edi + PGA
Model 2: Ch ~ Nom + Dir + Aml + Afl + Ams + Afs + Scr + Cin + Art
    + Cos +
     Sco + Son + Edi + Sou + For + Anf + Eff + Mak + Dan + AD +
     Gdr + Gmc + Gd + Gm1 + Gm2 + Gf1 + Gf2 + PGA + DGA + Action +
     Adventure + Animation + Biography + Comedy + Crime + Docu +
     Drama + Family + Fantasy + Film.noir + History + Horror +
     Music + Musical + Mystery + Romance + SciFi + Sport + Thriller
     +
     War + Western + Length + Days + G + PG + PG13 + R + U + Ebert
    +
     NYFCC + LAFCA + NSFC + NBR + WR
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       600     324.47
2       539     258.01 61    66.46   0.2945

> anova(step_bic, step_aic, test = "LRT")
Analysis of Deviance Table

Model 1: Ch ~ Dir + Edi + PGA
Model 2: Ch ~ Dir + Edi + Dan + Gdr + Gd + PGA + DGA + Romance +
    SciFi +
     Days + PG + PG13 + R + NSFC + WR
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       600     324.47
2       588     281.92 12    42.55 2.691e-05 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .
        0.1         1
```

.2.5 2.5. AIC and BIC comparison tables

```
> BIC(full_model, step_aic, step_bic)
           df      BIC
full_model 65 674.2406
step_aic   16 384.3756
step_bic    4 350.0823

> AIC(full_model, step_aic, step_bic)
           df      AIC
full_model 65 388.0083
step_aic   16 313.9184
step_bic    4 332.4680
```

.3 3. Task Three

.3.1 Plotting ROC Curve and Computing AUC

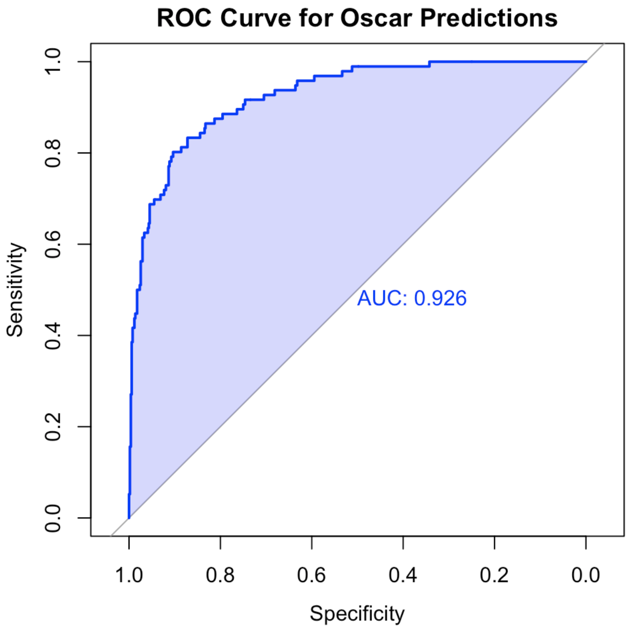
```
> library(pROC)

> probabilities<-predict(step_aic, type = "response")
> roc_curve<-roc(oscar$Ch, probabilities)
> auc_value<-auc(roc_curve)
> print(auc)

Area under curve: 0.9257


> plot(roc_curve, col = "blue", lwd = 2, main = "ROC curve for
    Oscar Predictions", print.auc = TRUE)
```

```
> polygon(c(roc_curve$specificities, 1), c(roc_curve$sensitivities
   , 0),col = rgb(0,0,1,0.2), border = NA)
```



.3.2 Calculating Optimal Threshold and Sensitivity

```
> optimal_coords<-coords(roc_curve, "best", ret = "threshold")
> optimal_threshold<-optimal_coords[1]
> print(optimal_threshold)

  threshold
1 0.179937


> predicted_classes<- ifelse(probabilities >= 0.179937, 1, 0)
> predicted_classed<-factor(predicted_classes, levels = c(0,1))

> conf_matrix<- table(Predicted = predicted_classes, Actual=oscars
   $Ch)
> print(conf_matrix)

           Actual
Predicted    0   1
          0 459  19
          1  49  77

> TP<-conf_matrix[2,2]
> FN<-conf_matrix[1,2]
> sensitivity<-TP/(TP+FN)
> print(sensitivity)

[1] 0.8020833
```

.4 4. Task Four

```
> oscars_2024<-read.csv("Oscars2024.csv")
```

```
> probabilities_2024<-predict(step_aic, newdata = oscars_2024,
   type = "response")

> probabilities_2024<-probabilities / sum(probabilities_2024)

> oscars_2024$Predicted_Probability<-probabilities_2024

> print(oscars_2024[, c("Predicted_Probabilities")])

 [1] 0.0130095429 0.8684878987 0.0105519067 0.0008996693
   0.0197093734
 [6] 0.0069555885 0.0474136578 0.0126448182 0.0012700444
   0.0190575002
```