King's College London
Department of Mathematics
Submission Cover Sheet for Coursework

The following cover sheet must be completed and submitted with any dissertation, project, coursework essay or report submitted as a part of formal assessment for degree within the Mathematics Department.

You are not required to write your name on your work

Candidate Number (this is found on your student record account):	AF55416
Module Code and Title:	5CCM242A: Statistical Modelling
Title of Project/Coursework:	The Oscars Challenge

Declaration

By submitting this assignment I agree to the following statements:
I have read and understand the King's College London Academic Honesty and Integrity Statement that I signed upon entry to this programme of study.
I declare that the content of this submission is my own work.
I understand that plagiarism is a serious examination offence, an allegation of which can result in action being taken under the College's Misconduct regulations.

Your work may be used as an example of good practice for future students to refer to. If chosen, your work will be made available either via KEATS or by paper copy. Your work will remain anonymous; neither the specific mark nor any individual feedback will be shared. Participation is entirely optional and will not affect your mark. If you consent to your submission being used in this way, please add an X in the box.	X

1. First, I created the full model:

```
> setwd("C:/Users/caitlyn jones/OneDrive/classes/math/statistical modelling/Coursework")
> oscars <- read.csv("oscars.csv", header = TRUE)
> oscars <- oscars[oscars$Ch != 0, ]
> oscars$Ch <- ifelse(oscars$Ch == 1, 1, 0)
> full_model <- glm(Ch ~ Nom + Pic + Dir + Aml + Afl + Ams + Afs + Scr + Cin + Art + Cos + Sco + So
n + Edi + Sou + For + Anf + Eff + Mak + Dan + AD + Gdr + Gmc + Gd + Gm1 + Gm2 + Gf1 + Gf2 + PGA + D
GA + Action + Adventure + Animation + Biography + Comedy + Crime + Docu + Drama + Family + Fantasy
+ Film.noir + History + Horror + Music + Musical + Mystery + Romance + SciFi + Sport + Thriller + W
ar + Western + Length + Days + G + PG + PG13 + R + U + Ebert + NYFCC + LAFCA + NSFC + NBR + WR, dat
a = oscars, family = binomial)
> summary(full_model)
```

   Then, determined the significant coefficient, which was:

```
Dir           1.780e+00  6.959e-01    2.557  0.01055 *
```

   From there, I determined that winning best director allotted for around 5% better chance at winning.

```
> exp(coef(full_model)["Dir"])
    Dir
5.9277
> confint(full_model, "Dir")
Waiting for profiling to be done...
    2.5 %    97.5 %
0.4853807 3.2454707
```

2. I chose to use a step model to create the best model, as it will iterate over the model fitting until it finds the best one.

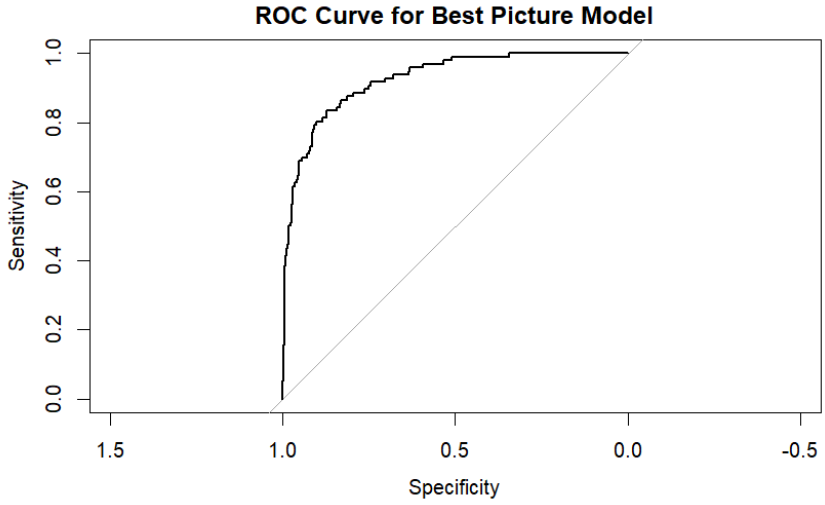
```
> library(MASS)
> best_model <- step(full_model, direction = "both")
```

   The AIC started out at 388.01 and ended at 313.92 with the following predictors:

```
Step:  AIC=313.92
Ch ~ Dir + Edi + Dan + Gdr + Gd + PGA + DGA + Romance + SciFi +
    Days + PG + PG13 + R + NSFC + WR
```

   The variables that are included are based on the categories which have the most pull on the AIC. In other words, the variables that were excluded are not as significant to the final AIC as the ones that are included are.

3. Next, I calculated the area under the curve:



```
> predicted_probs <- predict(best_model, type ="response")
> roc_curve <- roc(oscars$Ch, predicted_probs)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> plot(roc_curve, main = "ROC Curve for Best Picture Model")
> auc_value <- auc(roc_curve)
> print(auc_value)
Area under the curve: 0.9257
```

Then, determined the optimal threshold based on the ROC curve:

```
> opt_thresh <- coords(roc_curve, "best", ret = "threshold")
> print(opt_thresh)
  threshold
1  0.179937
> str(predicted_probs)
 Named num [1:604] 0.731998 0.43912 0.000417 0.172277 0.006449 ...
 - attr(*, "names")= chr [1:604] "11" "12" "13" "14" ...
> str(oscars$Ch)
 num [1:604] 0 0 0 0 0 1 0 0 0 0 ...
> predicted_probs <- as.numeric(predicted_probs)
> pred_class <- ifelse(predicted_probs >= opt_thresh, 1, 0)
> opt_thresh <- as.numeric(opt_thresh)
```

And finally, calculated the sensitivity:

```
> pred_class <- ifelse(predicted_probs >= opt_thresh, 1, 0)
> con_mat <- table(Predicted = pred_class, Actual = oscars$Ch)
> sensitivity <- con_mat[2,2] / (con_mat[2,2] + con_mat[1,2])
> print(sensitivity)
[1] 0.8020833
```

4. Then, I used my model to predict the best picture winner of this year's Oscars.

```
> oscars <- read.csv("oscars.csv", header = TRUE)
> curr_noms <- oscars[oscars$Ch == 0,]
> curr_noms$Predicted_Prob <- predict(best_model, newdata = curr_noms, type = "response")
> curr_noms$Normalized_Prob <- curr_noms$Predicted_Prob / sum(curr_noms$Predicted_Prob)
> results <- curr_noms[, c("Name", "Normalized_Prob")]
> print(results)
```

The winner Anora was correctly predicted.

```
                Name Normalized_Prob
1  A Complete Unkown    0.0130095429
2              Anora    0.8684878987
3           Conclave    0.0105519067
4     Dune: Part Two    0.0008996693
5       Emilia Perez    0.0197093734
6     I'm Still Here    0.0069555885
7        Nickel Boys    0.0474136578
8      The Brutalist    0.0126448182
9      The Substance    0.0012700444
10            Wicked    0.0190575002
```

5. Another model that could work would be Bayesian Networks, which deal with dependencies on other variables. For example, it will predict based on the relationship between number of nominations and winning best director, which will be used to predict the best picture win. It is very important for the predicted probabilities to sum up to 1 because we get a more accurate depiction of probability in other places, in this case, Oscar win predictions.