STA 360 Final Project

Glen Morgenstern and Jason Gerber

4/24/2021

```r
library(tidyverse)
```

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(broom)
```

```r
oscars <- read_csv("data/oscars.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   Name = col_character(),
##   Movie = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```r
director_noms <- oscars %>%
  filter(DD == 1, Year != 2006) %>%
  rename(win = Ch) %>%
  mutate(win = ifelse(win==1, 1, 0))
# win is 1 if a win, 0 if loss
```

trouble with viewing win column??

Introduction: A few paragraphs which (i) motivate problem importance & relevance

(supported by any pertinent literature), (ii) describe project goals and how such goals address the problem, (iii) a high-level roadmap of the proposed Bayesian modeling framework, and (iv) other relevant information for the reader. See project rubric for details.

Billions of people around the globe watch movies, and millions tune in to watch the Academy Awards every year(citation).

The goal of this project is to create a model which will predict with as much accuracy as possible the four major category winners at the Oscars. The Academy Awards is heavily publicized, promoted, and talked about every year, where millions of fans tune in to watch the ceremony and tens of millions(perhaps more) read about it later. Receiving an Oscar is the highest honor that one can procure in the business of movie-making, and since movies are such a large industry and many of our country's most influential celebrities are movie stars, the Oscars is one of the biggest events there is each year in the United States.

All this attention inevitably leads to general curiosity and predictions about who will take home the most prized awards in all of filmmaking, which is where the reason for this project arises.

The goal of this project is to create a model which will predict with as much accuracy as possible the four major category winners at the Oscars, which include Best Lead Actor, Best Lead Actress, Best Picture, and Best Director. Being able to predict Oscar winners is an analysis which many across the globe will have an interest in, whether they be data-lovers, typical cinephiles, or even gamblers who feel that they need an edge.

# high level roadmap of proposed modeling framework.

# reader info

# fix up wording here

• Data sources are clearly outlined & fully described. If needed, a clear procedure is given on how to get from raw data to usable data. • All variables clearly identified and described for the study. Any relevant features of these variables (e.g., data type, interpretation, etc.) should be fully discussed. • The conclusions obtained from the proposed data can fully address project goals. If not, potential limitations in the data should be discussed fully. • Project goals are addressed in a comprehensive and nuanced way. • Exploratory data analysis should support project goals and help guide specification of model.

Data: A couple of paragraphs describing the data to be used. You may wish to

discuss: (i) data sources – where are you getting the data? (ii) data description – what data / variables will be used for modeling? (iii) data type – ordinal discrete, nominal discrete, continuous, etc., and (iv) data scraping / wrangling – how to extract and clean data for modeling? See project rubric for details.

The data set that we are using for this analysis is one that contains information about all Oscar nominations from 1928-2006 for the four major academy awards: Best Picture, Best Director, Best Leading Actor, and Best Leading Actress. The data has been taken from the personal website of Iain Pardoe, an online educator in mathematics and statistics.

In our model, we will attempt to use all relevant variables that may be significant predictors of whether a candidate wins their award type or not. The data set given has 62 variables, of which 4 are indicator variables indicating if the nominee is up for that award. For example, if Kate Winslet is nominated for best leading actress, the variable FF(lead actress indicator) will equal 1, while the indicator variables for lead actor, best picture, and best director will equal 0. These variables will help us separate to make four separate models, one for predicting each award type,

Our explanatory variables include the year the movie was produced, as well as several statistics involving other people who worked on the movie that the award nominee is from. For example, there is an indicator variable that specifies if the editing in the movie was nominated for an Oscar, a number of times the lead actor for the movie was previously nominated, and more similar variables. The basic idea is to predict if the nominee will win the award based on the strength of the rest of the movie and its cast.

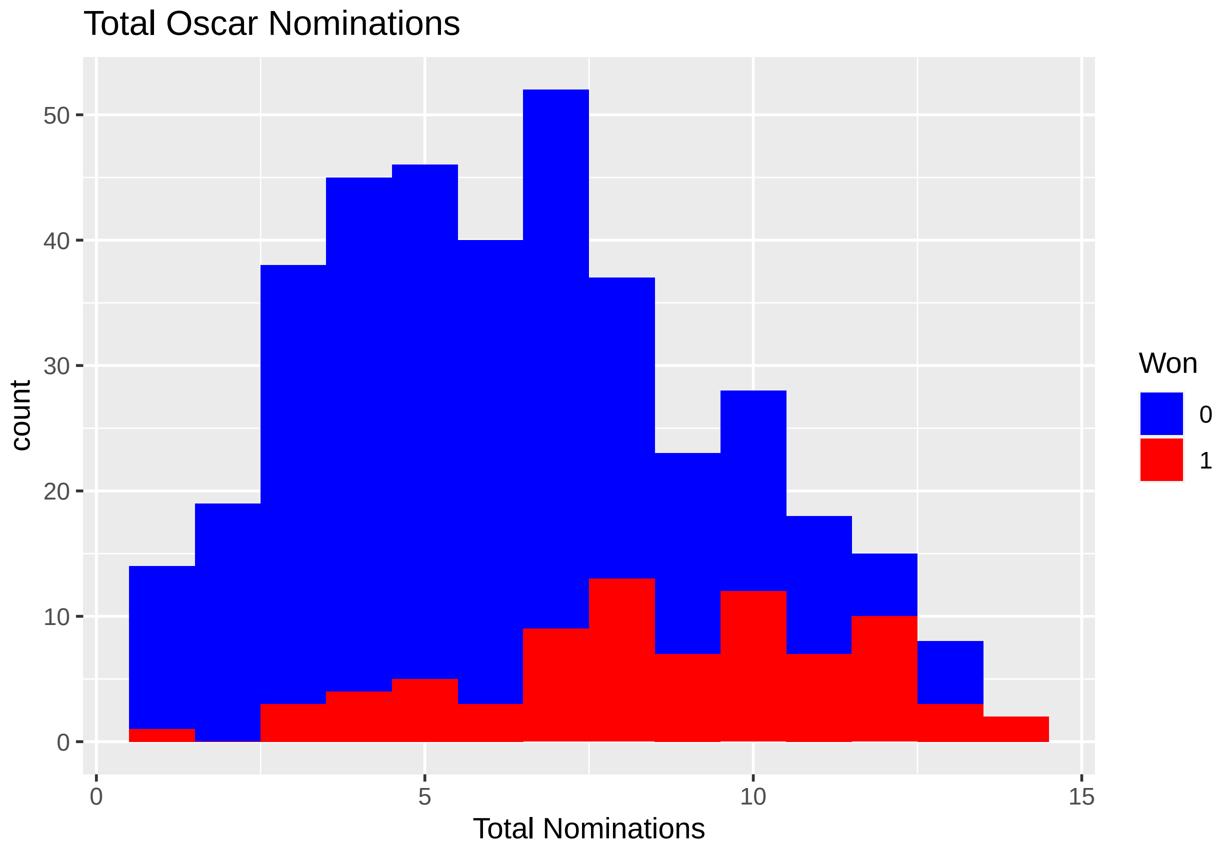
Additionally, there are several interaction variables already built into the data ### Glen, idk if we want to add MORE interactions?

## Limitations

Project Goals

EDA

```r
ggplot(data = director_noms, aes(x = Nom, fill = as.character(win))) + geom_histogram(binwidth = 1) +
  labs(title = "Total Oscar Nominations", x = "Total Nominations")  +
  scale_fill_manual(name = "Won", values = c("blue", "red"))
```



```r
#Total nominations matter a lot RELEVANT
director_noms %>%
  group_by(win) %>%
  summarise(mean_noms = mean(Nom),
            Q1_noms = quantile(Nom, c(0.25)),
            Q3_noms = quantile(Nom, c(0.75)))
```

```
## # A tibble: 2 x 4
##     win mean_noms Q1_noms Q3_noms
##   <dbl>     <dbl>   <dbl>   <dbl>
## 1     0      5.85       4    7.75
## 2     1      8.65       7   11
```

```r
# Best picture matters a lot RELEVANT
director_noms %>%
  group_by(win) %>%
  summarise(total = n(), total_best_picture = length(Pic[Pic == 1]),
            prop_picture = total_best_picture/total)
```

```
## # A tibble: 2 x 4
##     win total total_best_picture prop_picture
##   <dbl> <int>              <int>        <dbl>
## 1     0   306                204        0.667
## 2     1    79                 77        0.975
```

```r
# Male leads matter a lot RELEVANT
director_noms %>%
  group_by(win) %>%
  summarise(total = n(), prop_at_least_1_male_lead = length(Aml[Aml>=1])/n(),
            std_dev_male_lead = sd(Aml))
```

```
## # A tibble: 2 x 4
##     win total prop_at_least_1_male_lead std_dev_male_lead
##   <dbl> <int>                     <dbl>             <dbl>
## 1     0   306                     0.395             0.551
## 2     1    79                     0.671             0.571
```

```r
# Female leads matter less NOT RELEVANT
director_noms %>%
  group_by(win) %>%
  summarise(total = n(), prop_at_least_1_female_lead = length(Afl[Afl>=1])/n(),
            std_dev_male_lead = sd(Afl))
```

```
## # A tibble: 2 x 4
##     win total prop_at_least_1_female_lead std_dev_male_lead
##   <dbl> <int>                       <dbl>             <dbl>
## 1     0   306                       0.310             0.480
## 2     1    79                       0.342             0.535
```

```r
# Difference in both male and female supporting actors RELEVANT
director_noms %>%
  group_by(win) %>%
  summarise(prop_at_least_1_male_support = length(Ams[Ams>=1])/n(),
            std_dev_male_support= sd(Ams))
```

```
## # A tibble: 2 x 3
##     win prop_at_least_1_male_support std_dev_male_support
##   <dbl>                        <dbl>                <dbl>
## 1     0                        0.304                0.544
## 2     1                        0.506                0.706
```

```r
director_noms %>% # NOT RELEVANT
  group_by(win) %>%
  summarise(prop_at_least_1_female_support = length(Afs[Afs>=1])/n(),
            std_dev_male_lead = sd(Afs))
```

```
## # A tibble: 2 x 3
##     win prop_at_least_1_female_support std_dev_male_lead
##   <dbl>                          <dbl>             <dbl>
## 1     0                          0.284             0.539
## 2     1                          0.342             0.673
```

More EDA

```
director_noms %>%
  group_by(win) %>%
  summarise(prop_screenplay_nom = length(Scr[Scr==1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_screenplay_nom
##   <dbl>               <dbl>
## 1     0               0.732
## 2     1               0.924
```

```
director_noms %>%
  group_by(win) %>%
  summarise(prop_cinematography_nom = length(Cin[Cin==1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_cinematography_nom
##   <dbl>                   <dbl>
## 1     0                   0.399
## 2     1                   0.658
```

```
director_noms %>%
  group_by(win) %>%
  summarise(prop_art_nom = length(Art[Art==1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_art_nom
##   <dbl>        <dbl>
## 1     0        0.320
## 2     1        0.468
```

```
director_noms %>%
  group_by(win) %>%
  summarise(prop_costum_nom = length(Cos[Cos==1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_costum_nom
##   <dbl>           <dbl>
## 1     0           0.209
## 2     1           0.316
```

```
director_noms %>%
  group_by(win) %>%
  summarise(prop_score_nom = length(Sco[Sco==1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_score_nom
##   <dbl>          <dbl>
## 1     0          0.281
## 2     1          0.532
```

```
# All appear to be somewhat relevant??
# double the chance
director_noms %>%
  group_by(win) %>%
  summarise(prop_score_nom = length(Edi[Edi==1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_score_nom
##   <dbl>          <dbl>
## 1     0          0.389
## 2     1          0.747
```

```
# 2.5x the chance
director_noms %>%
  group_by(win) %>%
  summarise(prop_score_nom = length(Sou[Sou==1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_score_nom
##   <dbl>          <dbl>
## 1     0          0.219
## 2     1          0.519
```

```
# double chance, but not too many are nominated
director_noms %>%
  group_by(win) %>%
  summarise(prop_score_nom = length(Eff[Eff==1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_score_nom
##   <dbl>          <dbl>
## 1     0         0.0359
## 2     1         0.0886
```

```
# Slight Advantage to Nominees
director_noms %>%
  group_by(win) %>%
  summarise(prop_score_nom = mean(PrN))
```

```
## # A tibble: 2 x 2
##     win prop_score_nom
##   <dbl>          <dbl>
## 1     0           1.08
## 2     1           1.33
```

```
# Slight Advantage to Winners
director_noms %>%
  group_by(win) %>%
  summarise(prop_score_nom = mean(PrW))
```

```
## # A tibble: 2 x 2
##     win prop_score_nom
##   <dbl>          <dbl>
## 1     0          0.252
## 2     1          0.342
```

```
# previous Wins
```

```r
# Extreme Correlation
director_noms %>%
  group_by(win) %>%
  summarise(prop_gd_win = length(Gd[Gd == 1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_gd_win
##   <dbl>       <dbl>
## 1     0      0.0719
## 2     1      0.430
```

```r
# Extreme Correlation, if they win the DGA, choose them to win the Oscar
director_noms %>%
  group_by(win) %>%
  summarise(prop_DGA = length(DGA[DGA == 1])/n())
```

```
## # A tibble: 2 x 2
##     win prop_DGA
##   <dbl>    <dbl>
## 1     0   0.0163
## 2     1   0.646
```

Check for real interactions!!

```r
# High correlation interaction variables!!

# year interactions

director_wins <- director_noms %>%
  filter(win == 1)

director_wins %>%
  summarise(mean(Art*Sco))
```

```
## # A tibble: 1 x 1
##   ‘mean(Art * Sco)‘
##               <dbl>
## 1             0.367
```

```r
# other variables that I tested but were not significant at all: PrN, PrW, Sou, Sco, Cos, Art, Aml, Afl

log.model <- glm(win ~ Nom + Pic + Scr + Cin + Edi + Eff +
                   Gd + DGA + Year
                 , data = director_noms, family = binomial)

tidy(log.model, exponentiate = FALSE, conf.int = TRUE)
```

```
## # A tibble: 10 x 7
##    term        estimate std.error statistic  p.value conf.low conf.high
##    <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1 (Intercept)  92.1      23.1         3.99 6.53e- 5  49.9     141.
##  2 Nom          -0.172     0.124      -1.39 1.65e- 1  -0.422     0.0669
##  3 Pic           1.52      0.824       1.84 6.52e- 2   0.0747    3.46
##  4 Scr           0.814     0.489       1.67 9.59e- 2  -0.129     1.80
##  5 Cin           0.605     0.462       1.31 1.91e- 1  -0.306     1.52
##  6 Edi           1.32      0.537       2.46 1.40e- 2   0.282     2.40
##  7 Eff           0.766     0.567       1.35 1.77e- 1  -0.402     1.83
##  8 Gd            0.612     0.554       1.10 2.69e- 1  -0.519     1.68
##  9 DGA           5.30      0.711       7.46 8.52e-14   4.03      6.84
## 10 Year         -0.0493    0.0119     -4.16 3.21e- 5  -0.0745   -0.0276
```

How do I check if a logistic model fits??

Data sources clearly described

Model: Discussion & justification of the proposed Bayesian model framework

(prior and sampling model). This discussion should elicit prior information on the problem, the data sources available, and relevant project goals. Any "downstream" uses of the model (e.g., for prediction, optimization, ranking) should be discussed in detail here. See project rubric for details.

Results: Posterior analyses from the fitted Bayesian model, and a translation of

such findings into meaningful & understandable conclusions for the target audience (e.g., engineers, business managers, policy-makers, etc). See project rubric for details.

Conclusion: A summary of key findings and potential impacts of your project.