# Predicting Movie Ratings using a Bayesian Regression Model

Dohyun Lee

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

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

```r
library(ggplot2)
library(dplyr)
library(BAS)

#Read in dataset
movie_data <- get(load("/Users/Dohyun/Downloads/movies.Rdata"))
movie_data
```

```
## # A tibble: 651 x 32
##    title title_type genre runtime mpaa_rating studio thtr_rel_year
##    <chr> <fct>      <fct>   <dbl> <fct>       <fct>          <dbl>
##  1 Fill~ Feature F~ Drama      80 R           Indom~          2013
##  2 The ~ Feature F~ Drama     101 PG-13       Warne~          2001
##  3 Wait~ Feature F~ Come~      84 R           Sony ~          1996
##  4 The ~ Feature F~ Drama     139 PG          Colum~          1993
##  5 Male~ Feature F~ Horr~      90 R           Ancho~          2004
##  6 Old ~ Documenta~ Docu~      78 Unrated     Shcal~          2009
##  7 Lady~ Feature F~ Drama     142 PG-13       Param~          1986
##  8 Mad ~ Feature F~ Drama      93 R           MGM/U~          1996
##  9 Beau~ Documenta~ Docu~      88 Unrated     Indep~          2012
## 10 The ~ Feature F~ Drama     119 Unrated     IFC F~          2012
## # ... with 641 more rows, and 25 more variables: thtr_rel_month <dbl>,
## #   thtr_rel_day <dbl>, dvd_rel_year <dbl>, dvd_rel_month <dbl>,
## #   dvd_rel_day <dbl>, imdb_rating <dbl>, imdb_num_votes <int>,
## #   critics_rating <fct>, critics_score <dbl>, audience_rating <fct>,
## #   audience_score <dbl>, best_pic_nom <fct>, best_pic_win <fct>,
## #   best_actor_win <fct>, best_actress_win <fct>, best_dir_win <fct>,
## #   top200_box <fct>, director <chr>, actor1 <chr>, actor2 <chr>, actor3 <chr>,
## #   actor4 <chr>, actor5 <chr>, imdb_url <chr>, rt_url <chr>
```

The dataset consists of 456 randomly sampled movies released between 1972 to 2014 from IMDB and Rotten Tomatoes. Since the samples were not randomly assigned, we cannot infer causality, making this an observational study rather than an experimental one.

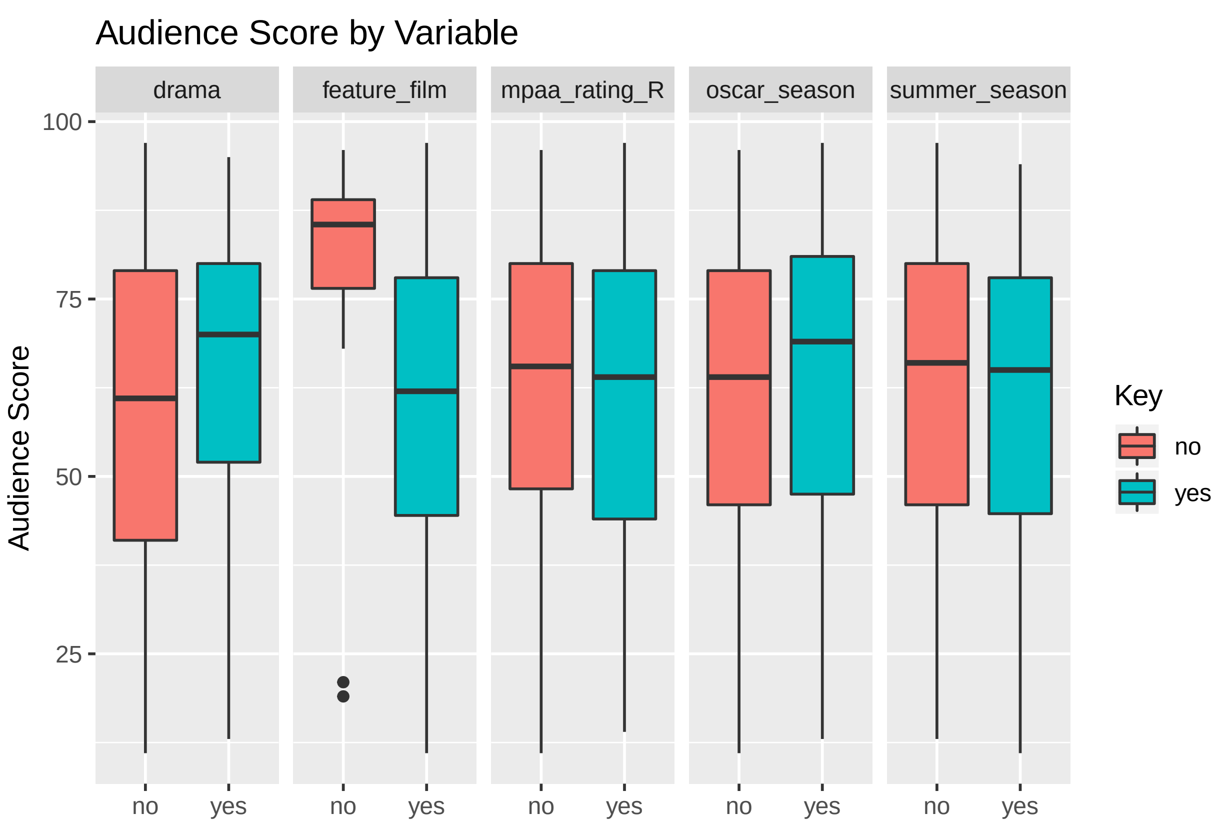
### Exploratory Data Analysis

```
movieDF = mutate(movie_data,
                    feature_film = ifelse(title_type == 'Feature Film', "yes", "no"),
                    drama = ifelse(genre == 'Drama', "yes", "no"),
                    mpaa_rating_R = ifelse(mpaa_rating == 'R', "yes", "no"),
                    oscar_season = ifelse(thtr_rel_month %in% 10:12 ,"yes","no"),
                    summer_season = ifelse(thtr_rel_month %in% 5:8 ,"yes","no"))

#visualize the relationship between audience scores and different variables of a film
eda <- movieDF %>% select(audience_score, feature_film, drama, mpaa_rating_R, oscar_season, summer_seas

#spreads out the data visuals
gatherDF <- gather(eda,key=varname,value=val,-audience_score)

#plot the boxplots
ggplot(data = gatherDF, aes(x=val,y=audience_score,fill=val)) +
  geom_boxplot() +
  facet_grid(~varname) +
  xlab("") + ylab("Audience Score") +
  labs(title="Audience Score by Variable",fill="Key")
```



Despite feature films being the most present type of film, it has a lower audience score rating of around 60 while non-features have a median score of around 85. Meanwhile, every other category seems to be fairly balanced.

### Bayesian Modeling

Here we build our Bayesian Regression Model:

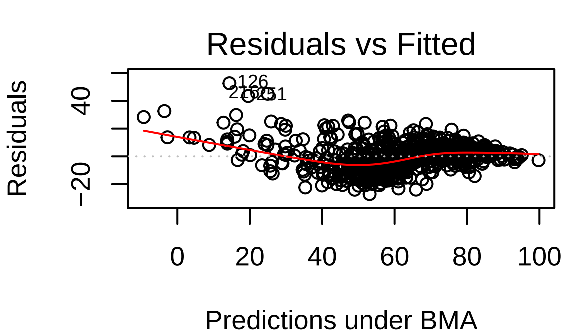
```
set.seed(123)
fit1 <- bas.lm(audience_score ~ feature_film + drama +
                runtime + mpaa_rating_R + thtr_rel_year +
                oscar_season + summer_season + imdb_rating +
                imdb_num_votes + critics_score +
                best_pic_nom + best_pic_win + best_actor_win +
                best_actress_win + best_dir_win + top200_box,
              data = movieDF,
              prior = "BIC",
              method = "MCMC",
              modelprior = uniform())
```

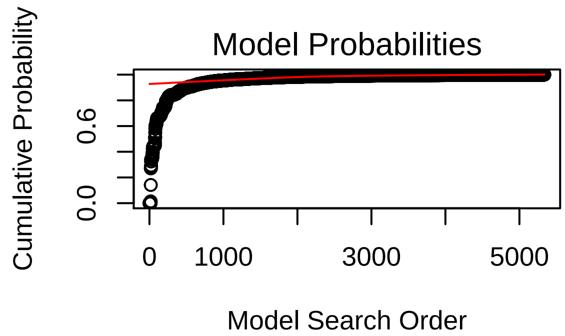
```
## Warning in bas.lm(audience_score ~ feature_film + drama + runtime +
## mpaa_rating_R + : dropping 1 rows due to missing data
```

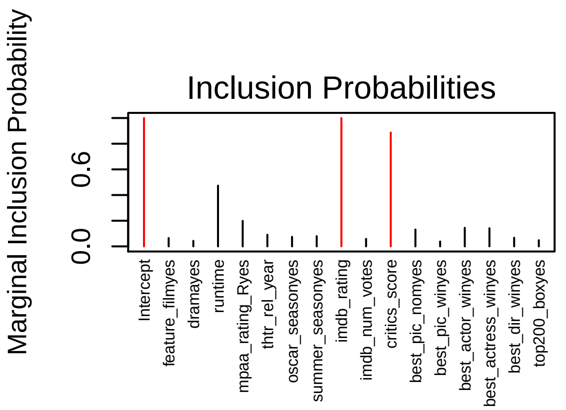
```
fit1
```

```
##
## Call:
## bas.lm(formula = audience_score ~ feature_film + drama + runtime +
##     mpaa_rating_R + thtr_rel_year + oscar_season + summer_season +
##     imdb_rating + imdb_num_votes + critics_score + best_pic_nom +
##     best_pic_win + best_actor_win + best_actress_win + best_dir_win +
##     top200_box, data = movieDF, prior = "BIC", modelprior = uniform(),
##     method = "MCMC")
##
##
##  Marginal Posterior Inclusion Probabilities:
##           Intercept      feature_filmyes             dramayes
##             1.00000              0.06493              0.04319
##             runtime     mpaa_rating_Ryes        thtr_rel_year
##             0.47324              0.19846              0.09108
##     oscar_seasonyes     summer_seasonyes          imdb_rating
##             0.07452              0.07937              0.99998
##      imdb_num_votes        critics_score      best_pic_nomyes
##             0.05856              0.88676              0.13147
##     best_pic_winyes    best_actor_winyes  best_actress_winyes
##             0.03848              0.14487              0.14152
##     best_dir_winyes        top200_boxyes
##             0.06749              0.04791
```

```
#Checking our assumptions
par(mfrow=c(2,2))
plot(fit1, which=c(1, 2), ask=FALSE)
plot(fit1, which=4, ask=FALSE, cex.lab=0.5)
```




The coefficients under each variable represent the likelihood (from 0 to 1) that it is included in the posterior model. For instance "imdb_rating" has a likelihood of 1, which means it will definitely be in included in the model. "critics_score" has a likelihood of 0.89, which tells us that it is an important variable that will play a huge part in predicting the movie rating.

As for our assumptions, our residuals vs fitted plot (top left) data points aren't randomly, evenly scattered, which might tell us that some predictor variables are unnecessary or need further evaluation for inclusion in the study. A Markov Chain Monte Carlo (MCMC) method was used for sampling models for fitting the data because there is a lot of predictor variables in the study. The top-right graph shows the posterior probability density leveling off at 1 after approximately 3000 model combinations. The bottom-left graph shows the likelihood of each predictor variable being included in the posterior model, which "imdb_rating" and "critics_score" yielding the highest likelihoods.

### Testing our Model

To test our posterior model, we will use three films that isn't listed in our dataset. We'll use all-time grossing movie "Avengers: Endgame", "Venom", and "The Last Airbender".

```
#Test Run 1
grep("Avengers: Endgame", movieDF$title)
```

```
## integer(0)
```

```
#run the movie through the model
testRun1 <- data.frame(feature_film ="yes",
                       drama ="yes",
                       runtime = 181,
                       mpaa_rating_R = "no",
```

```r
                        thtr_rel_year= 2019,
                        oscar_season ="yes",
                        summer_season = "no",
                        imdb_rating = 8.4,
                        imdb_num_votes= 734914,
                        critics_score= 94,
                        best_pic_nom ="no",
                        best_pic_win ="no",
                        best_actor_win ="no",
                        best_actress_win ="no",
                        best_dir_win ="no",
                        top200_box ="yes")

bma_predictor =  predict(newdata = testRun1, fit1, estimator = "BMA", se.fit = TRUE)
ci_bma = confint(bma_predictor, estimator = "BMA")
ci_bma
```

```
##          2.5%    97.5%     pred
## [1,] 70.80166 111.6011 91.55218
## attr(,"Probability")
## [1] 0.95
## attr(,"class")
## [1] "confint.bas"
```

For "Avengers: Endgame" we get a 95% confidence interval of 71 to 112, so we can expect our predicted Rotten Tomatoes audience score to fall in that range. We get a predicted score of 92 and the actual audience score is 90.

```r
#Test Run 2
grep("Venom", movieDF$title)
```

```
## integer(0)
```

```r
testRun2 <- data.frame(feature_film ="yes",
                        drama ="yes",
                        runtime = 112,
                        mpaa_rating_R = "no",
                        thtr_rel_year= 2018,
                        oscar_season ="yes",
                        summer_season = "no",
                        imdb_rating = 6.7,
                        imdb_num_votes= 335961,
                        critics_score= 30,
                        best_pic_nom ="no",
                        best_pic_win ="no",
                        best_actor_win ="no",
                        best_actress_win ="no",
                        best_dir_win ="no",
                        top200_box ="yes")

bma_predictor2 =  predict(newdata = testRun2, fit1, estimator = "BMA", se.fit = TRUE)
ci_bma2 = confint(bma_predictor2, estimator = "BMA")
ci_bma2
```

```
##          2.5%    97.5%     pred
## [1,] 43.80292 83.45415 63.76094
## attr(,"Probability")
## [1] 0.95
## attr(,"class")
## [1] "confint.bas"
```

"Venom" is a strange one because it was a movie that critics hated but fans loved. This movie has a Rotten Tomatoes critic score of 30, but has a predicted audience score of 64. The actual audience score is 80 and falls in between our confidence interval of 44 and 83, so our model performed well with this film.

```r
#Test Run 3
grep("The Last Airbender", movieDF$title)
```

```
## integer(0)
```

```r
testRun3 <- data.frame(feature_film ="yes",
                        drama ="no",
                        runtime = 103,
                        mpaa_rating_R = "no",
                        thtr_rel_year= 2010,
                        oscar_season ="no",
                        summer_season = "yes",
                        imdb_rating = 4.1,
                        imdb_num_votes= 147385,
                        critics_score= 5,
                        best_pic_nom ="no",
                        best_pic_win ="no",
                        best_actor_win ="no",
                        best_actress_win ="no",
                        best_dir_win ="no",
                        top200_box ="no")

bma_predictor3 =  predict(newdata = testRun3, fit1, estimator = "BMA", se.fit = TRUE)
ci_bma3 = confint(bma_predictor3, estimator = "BMA")
ci_bma3
```

```
##          2.5%    97.5%     pred
## [1,] 3.371239 43.03383 23.52024
## attr(,"Probability")
## [1] 0.95
## attr(,"class")
## [1] "confint.bas"
```

"The Last Airbender", notoriously one of the worst adaptations in film history, received a 5 on the RT critic score, and has an audience score of 30. The predicted value came out to be 24, which is still very close to the true value and since it fits into the CI of 3-43, it is fair to say that our model predicted this film's audience score quite well.

### Conclusion

Our Bayesian Model, under specific prior variables, predicted the Rotten Tomatoes audience score quite well – even films that had a disparity between critic and audience ratings. Although the films I tested on had overall good results, it doesn't necessarily mean the model I used was perfect. We could get more accurate results and narrow down the confidence intervals if we use more relevant variables with higher likelihoods, while cutting out less relevant variables to better fit the posterior model.