TO PREDICT THE USER RATING BASED ON MOVIE INFORMATION.

STAT 52900 FINAL REPORT

-By Ravi Teja Seera

INTRODUCTION:

A film also called as a movie is a form of visual art that uses moving images to imitate experiences and convey ideas, tales, emotions, feelings, beauty, or atmosphere in different ways. Cinematography is shortened to "cinema," which is frequently used to refer to the process of making movies, the film business, and the art form that results from it.

Each movie watched and the amount of business made depends on how much the audience liked it. In this project I am going to predict the audience score which helps the OTT platforms like Netflix, prime to decide what to display in their feed and increase their amount of viewership.

It is very common in practice that multiple models provide adequate descriptions of the distributions generating the observed data. It is standard statistical practice that, in such situations, a better model must be selected according to some criteria, like model fit to the observed dataset, predictive capabilities or likelihood penalizations such as information criteria. After selection is performed, all inferences are made and conclusions drawn assuming the selected model as the true model.

However, there are downsides to this approach. The selection of one particular model may lead to overconfident inferences and riskier decision making as it ignores the existent model uncertainty in favor of very particular distributions and assumptions on the model of choice. Therefore, modeling this source of uncertainty to appropriately select or combine multiple models is very desirable.

Using Bayesian inference to this purpose has been suggested as a framework capable of achieving these goals (Leamer; 1978). Bayesian Model Averaging (BMA) is an extension of the usual Bayesian inference methods in which one does not only models parameter uncertainty through the prior distribution, but also model uncertainty obtaining posterior parameter and model posteriors using Bayes' theorem and therefore allowing for allow for direct model selection, combined estimation and prediction.

OBJECTIVE:

The main objective of this project is to develop a Bayesian Regression model to predict audience score.

Data:

Dataset consists of data on a variety of movie-related attributes, including how much audience and critics enjoy movies. The data collection includes information on 651 randomly selected movies that were produced only in United States and released in the period 1970-2016, from Rotten Tomatoes and IMDB.

In this project we are going to learn about what attributes make movie popular. Exploratory Data Analysis(EDA), modeling and prediction will be carried out.

Data Manipulation:

- Exploratory Data Analysis(EDA):
  EDA is a preliminary step done to obtain insights of the data like finding patterns etc.

For performing EDA I am going to create new variables.
I have performed all the analysis, modeling and prediction in R-studio

```
#Rcode for EDA

library(tidyverse)

library(ggplot2)

library(dplyr)

library(BAS)

library(ggpubr)

movies<-get(load("C:/Users/ravit/Downloads/movies.Rdata"))

movies=movies %>% mutate(feature_film=as.factor(if_else(title_type=="Feature Film","yes","no")))

movies=movies %>% mutate(drama=as.factor(if_else(genre=="Drama","yes","no")))

movies=movies %>% mutate(mpaa_rating_R=as.factor(if_else(mpaa_rating=="R","yes","no")))

movies=movies %>% mutate(oscar_season=as.factor(if_else(thtr_rel_month %in% c(10,11,12),"yes","no")))

movies=movies %>% mutate(summer_season=as.factor(if_else(thtr_rel_month %in% c(5,6,7,8),"yes","no")))

movies_eda=movies %>% select(audience_score,feature_film,drama,mpaa_rating_R,oscar_season,summer_season)

summary(movies_eda)
```

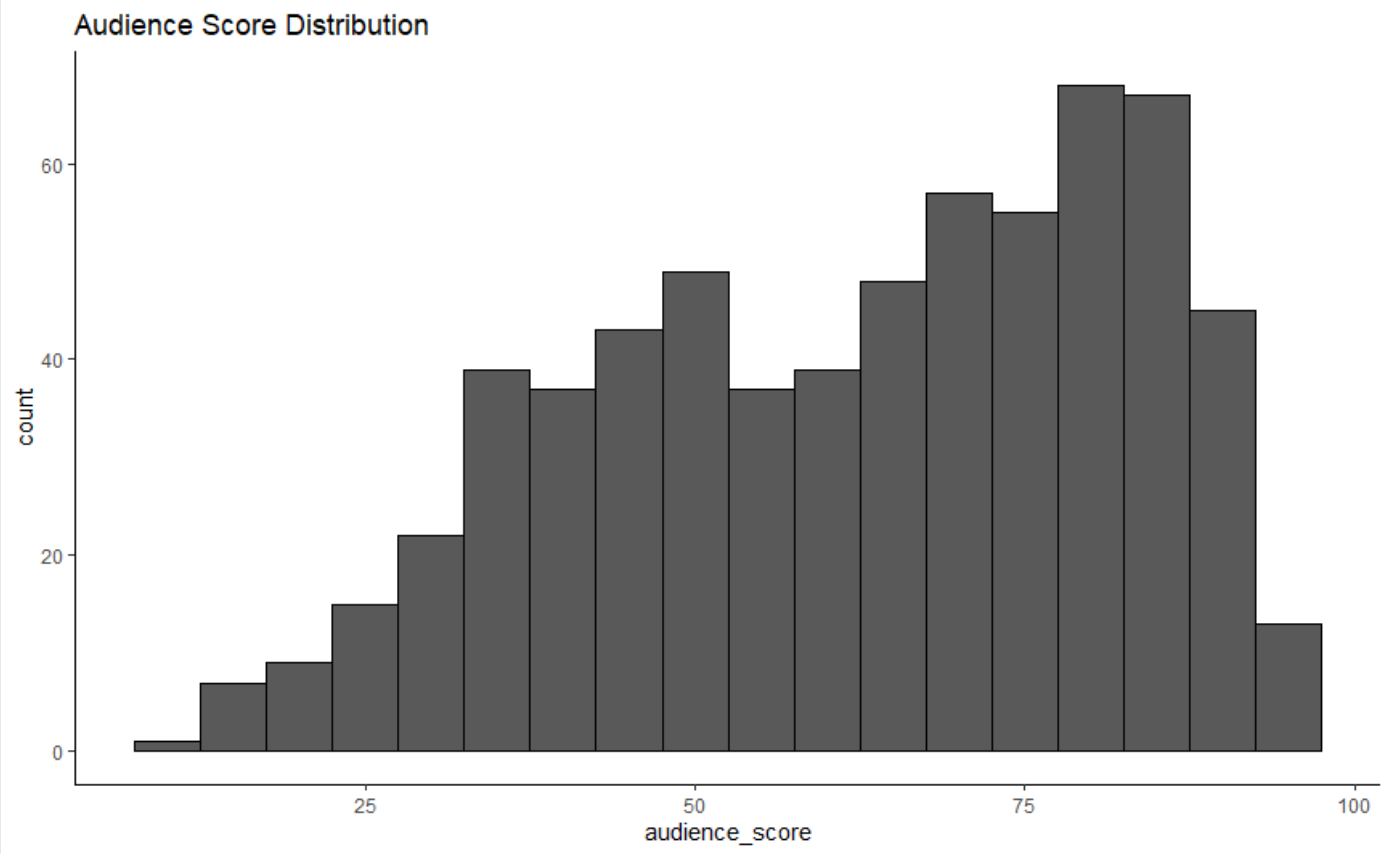
Output:

```
> summary(movies_eda)
 audience_score  feature_film drama     mpaa_rating_R
 Min.   :11.00   no : 60      no :346   no :322
 1st Qu.:46.00   yes:591      yes:305   yes:329
 Median :65.00
 Mean   :62.36
 3rd Qu.:80.00
 Max.   :97.00
 oscar_season summer_season
 no :460      no :443
 yes:191      yes:208
```

#Rcode for EDA

ggplot(movies_eda)+geom_histogram(aes(x=audience_score),binwidth=5,color="black")+labs(title="Audience Score Distribution")+theme_classic()

output:

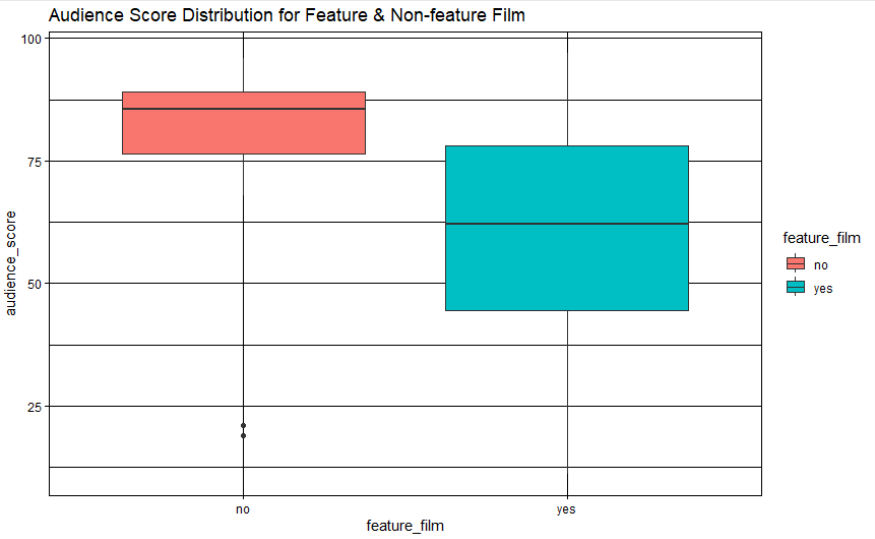


- The distribution of audience score is slightly left-skewed graph.

#Rcode for EDA

ggplot(movies_eda)+geom_boxplot(aes(x=feature_film,y=audience_score,fill=feature_film))+ labs(title="Audience Score Distribution for Feature & Non-feature Film")+theme_linedraw()
movies_eda %>% group_by(feature_film)%>% select(audience_score) %>% summarize(count=n(),mean=mean(audience_score),median=median(audience_score),min=min(audience_score),max=max(audience_score)

output:



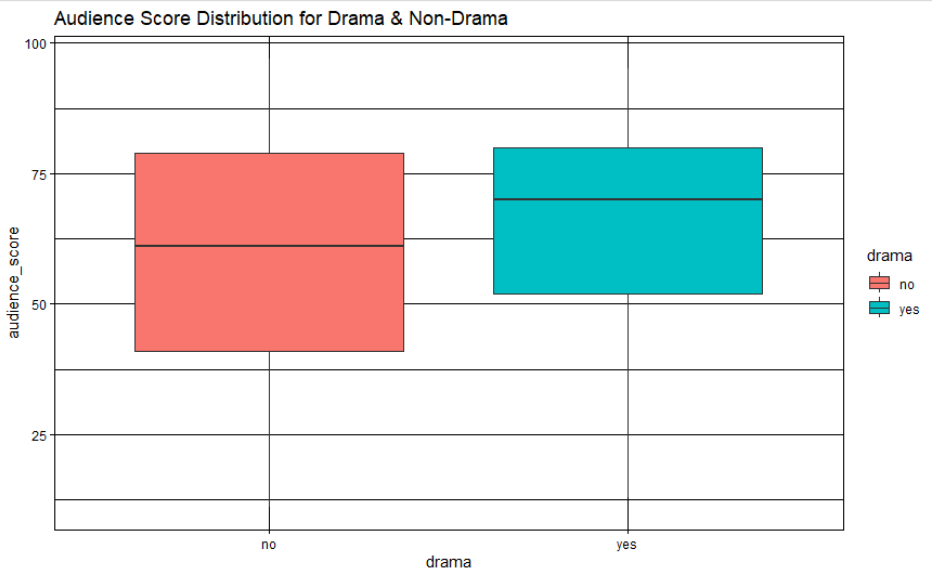
feature_film	count	mean	median	min	max
<fct>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
1 no	60	81.0	85.5	19	96
2 yes	591	60.5	62	11	97

>

- 60 non-feature films (9%) and 591 features (91%) make up the data set. The audience score for a feature film is often lower than that of a non-feature film (Mean:60, Median:62) (Mean:81, Median 85.5). If we take into account the box-plot as well, feature films have a far wider audience score range than non-feature films.

#Rcode for EDA

ggplot(movies_eda)+geom_boxplot(aes(x=drama,y=audience_score,fill=drama))+ labs(title="Audience Score Distribution for Drama & Non-Drama")+theme_linedraw()
movies_eda %>% group_by(drama)%>%select(audience_score) %>% summarize(count=n(),mean=mean(audience_score),median=median(audience_score),min=min(audience_score),max=max(audience_score))output:



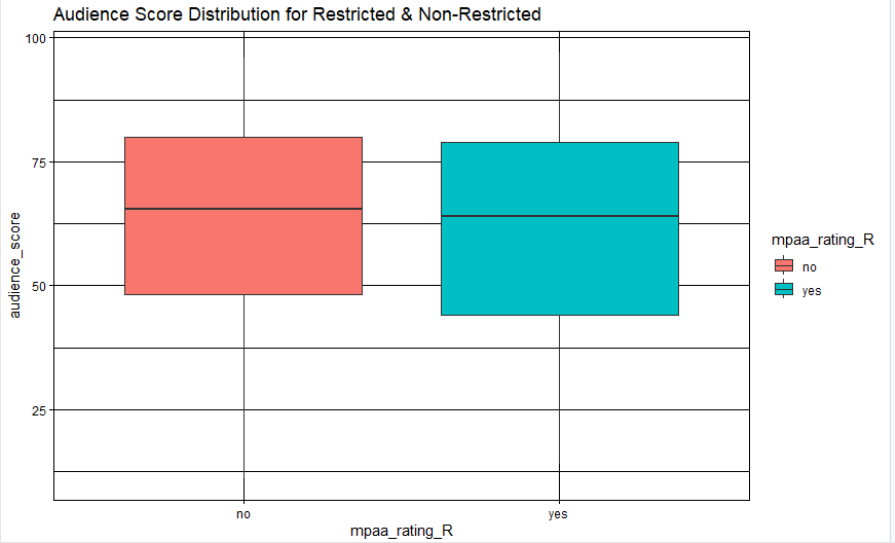
	drama	count	mean	median	min	max
	<fct>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
1	no	346	59.7	61	11	97
2	yes	305	65.3	70	13	95

>

- The data set contains 305 Drama (47%) and 346 non-Drama (53%). The audience score distribution between drama and non-drama appears to be relatively similar from the box-plot. They both fall within the same audience score range. The mean and median for the drama are marginally greater than those for non-drama (Mean:65.3, Median 70.0). (Mean:59.7, Median 61.0).

#Rcode for EDA

```
ggplot(movies_eda)+geom_boxplot(aes(x=mpaa_rating_R,y=audience_score,fill=mpaa_rating_R))+labs(title="Audience Score Distribution for Restricted & Non-Restricted")+theme_linedraw()
movies_eda %>% group_by(mpaa_rating_R)%>% select(audience_score) %>% summarize(count=n(),mean=mean(audience_score),median=median(audience_score),min=min(audience_score),max=max(audience_score))
```

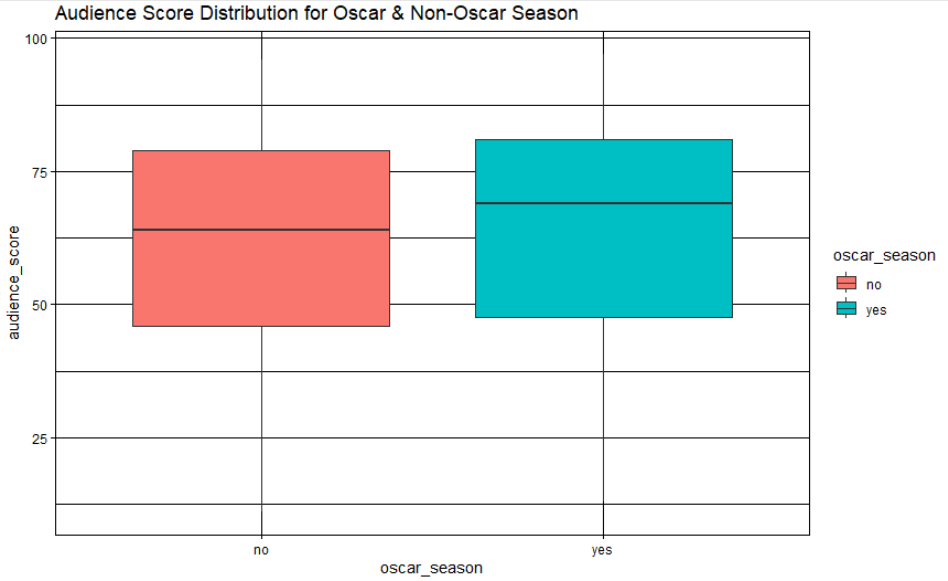



```
  mpaa_rating_R count  mean median   min   max
  <fct>         <int> <dbl>  <dbl> <dbl> <dbl>
1 no              322  62.7   65.5    11    96
2 yes             329  62.0   64      14    97
>
```

- The data collection contains 322 MPAA non-restricted and 329 MPAA restricted titles, respectively. The audience score distribution between MPAA restricted and MPAA non-restricted appears to be virtually comparable from the box plot. Their audience score ranges are comparable for the two of them. The mean and median scores for MPAA restricted (Mean:62, Median 64) and MPAA non-restricted (Mean:62.7, Median 65.5) are quite close.

#Rcode for EDA

```
ggplot(movies_eda)+geom_boxplot(aes(x=oscar_season,y=audience_score,fill=oscar_season))+labs(title="Audience Score Distribution for Oscar & Non-Oscar Season")+theme_linedraw()
movies_eda %>% group_by(oscar_season)%>% select(audience_score) %>% summarize(count=n(),mean=mean(audience_score),median=median(audience_score),min=min(audience_score),max=max(audience_score))
```



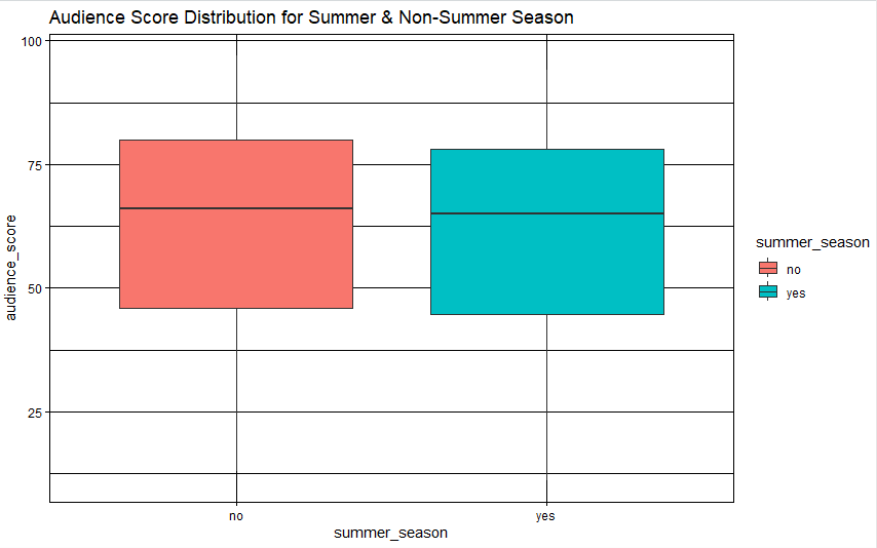
	oscar_season	count	mean	median	min	max
	<fct>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
1	no	460	61.8	64	11	96
2	yes	191	63.7	69	13	97

>

- The data collection includes 460 Non-Oscar season and 191 Oscar season episodes. The box-plot of the audience score distribution between Oscar and Non-Oscar is extremely similar to the previous MPAA rating and is very close to it. Their audience score ranges are fairly comparable for the two of them. The mean and median of the Oscar season film are marginally higher than those of the non-Oscar season film (Mean:63.7, Median 69). (Mean:61.8, Median 64).

#Rcode for EDA

```
ggplot(movies_eda)+geom_boxplot(aes(x=summer_season,y=audience_score,fill=summer_season))+labs(title="Audience Score Distribution for Summer & Non-Summer Season")+theme_classic()
movies_eda %>% group_by(summer_season)%>% select(audience_score) %>% summarize(count=n(),mean=mean(audience_score),median=median(audience_score),min=min(audience_score),max=max(audience_score))
```




```
  summer_season count  mean median   min   max
  <fct>          <int> <dbl>  <dbl> <dbl> <dbl>
1 no               443  62.6     66    13    97
2 yes              208  61.8     65    11    94
>
```

- There are 443 Non-Summer seasons and 208 Summer seasons in the data set, respectively (32% and 68%) Between Summer and Non-Summer, there is a striking similarity in the box-plot of audience score distribution. The audience score range for both of them is remarkably comparable. There isn't much of a difference between the means and medians for Summer season movies (Mean:61.8, Median 65) and Non-Summer season (Mean:62.6, Median 66.0).

Modeling :

#Rcode for Modeling

variable_list=c("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")

summary(movies %>% select(variable_list))

```
> summary(movies %>% select(variable_list))
Note: Using an external vector in selections is ambiguous.
i Use `all_of(variable_list)` instead of `variable_list` to silence this message.
i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
This message is displayed once per session.
 feature_film drama        runtime      mpaa_rating_R thtr_rel_year  oscar_season summer_season  imdb_rating
 no : 60      no :346   Min.   : 39.0   no :322       Min.   :1970   no :460      no :443       Min.   :1.900
 yes:591      yes:305   1st Qu.: 92.0   yes:329       1st Qu.:1990   yes:191      yes:208       1st Qu.:5.900
                        Median :103.0                 Median :2000                              Median :6.600
                        Mean   :105.8                 Mean   :1998                              Mean   :6.493
                        3rd Qu.:115.8                 3rd Qu.:2007                              3rd Qu.:7.300
                        Max.   :267.0                 Max.   :2014                              Max.   :9.000
                        NA's   :1
 imdb_num_votes   critics_score    best_pic_nom best_pic_win best_actor_win best_actress_win best_dir_win top200_box
 Min.   :   180   Min.   :  1.00   no :629      no :644      no :558        no :579          no :608      no :636
 1st Qu.:  4546   1st Qu.: 33.00   yes: 22      yes:  7      yes: 93        yes: 72          yes: 43      yes: 15
 Median : 15116   Median : 61.00
 Mean   : 57533   Mean   : 57.69
 3rd Qu.: 58301   3rd Qu.: 83.00
 Max.   :893008   Max.   :100.00
```

#Rcode for Modeling

movies_model=movies %>%
select(audience_score,variable_list) %>% drop_na()

As we can see there is only 1 missing data which is in run-time. So, we can simply remove this row as it very small proportion compared to the whole data-set.

BAYESIAN MODEL AVERAGING:

Let each model in consideration be denoted by M_l, l = 1,…..,K representing a set of probability distributions encompassing the likelihood function L($Y|\theta_l, M_l$)of the observed data **Y** in terms of model specific parameters θ_l and a set of prior probability densities for said parameters, denoted in general terms by $p(\theta_l|M_l)$ on which we omit eventual prior hyperparameters for the sake of clarity. Notice that both the likelihood and priors are conditional on a particular model.
Given a model, one then obtains the posterior distribution using Bayes' theorem, resulting in

$$p(\theta_l|Y, M_l)=\frac{\mathrm{L}(Y|\theta_l,M_l)\, p(\theta_l|M_l)}{\int \mathrm{L}(Y|\theta_l,M_l)p(\theta_l|M_l)d\theta_l} \qquad (1)$$

where the integral in the denominator is calculated over the support set for each prior distribution and represents the marginal distribution of the dataset over all parameter values specified in model M_l.

This quantity is essential for BMA applications as we will show momentarily and is called the model's marginal likelihood or model evidence and is denoted by

$$\mathrm{p}(\mathrm{Y}|M_l) = \int \mathrm{L}(Y|\theta_l, M_l)p(\theta_l|M_l)d\theta_l \qquad (2)$$

Bayesian model averaging then adds a layer to this hierarchical modeling present in Bayesian inference by assuming a prior distribution over the set of all considered models describing the prior uncertainty over each model's capability to accurately describe the data. If there is a probability mass function over all the models with values p(M_l) for l = 1, . . . , K, then
Bayes' theorem can be used to derive posterior model probabilities given the observed data by

$$p(M_l|Y)=\frac{\mathrm{p}(Y|M_l)\, p(M_l)}{\sum_{m=1}^{K} \mathrm{p}(Y|M_m)\, p(M_m)} \qquad (3)$$

resulting in a straightforward posterior model probability, representing the backing of each considered model by the observed data.
There is also a link between these posterior model probabilities and the use of Bayes Factors. Given two models l and m, the Bayes factor of model l against model m is given by

$$BF_{lm} = \frac{p(M_l|Y)}{p(M_m|Y)} \qquad (4)$$

thus quantifying the relative strength of the evidence in favor of model l against that of model m. Given a baseline model, which we arbitrarily fix as model 1, it is clear that equation (3) can be written in terms of Bayes Factors by simply dividing by the baseline model's evidence, resulting in

$$p(M_l|Y)=\frac{BF_{lm}\, p(M_l)}{\sum_{m=1}^{K} BF_{lm}p(M_m)} \qquad (5)$$

These model probabilities can mainly be used for two purposes. First, the posterior probabilities (3) can be used as a straightforward model selection criteria, with the most likely model being selected. Second, consider a quantity of interest Δ present in all models, such as a covariate or future observation, it follows that its marginal posterior distribution across all models is given by

$$p(\Delta|Y) = \sum_{l=1}^{K} p(\Delta|Y, M_l)p(M_l|Y)$$

which is an average of all posterior distributions weighted by each posterior model probability. Therefore, BMA allows for a direct combination of models to obtain combined parameter estimates or predictions "(Roberts; 1965)". This practice leads to predictions with lower risk under a logarithmic scoring rule "(Madigan and Raftery; 1994)" than using a single model. However, the implementation and application of BMA is not without difficulties. A prior distribution over the considered models must be specified, which is non trivial in most applications. Additionally, calculating each model evidence (equation 2) is non-trivial. Except in simple settings like in some generalized linear models with conjugate distributions, the evidence does not present a closed form and must be approximated, which presents plenty of challenges and is an active research field "(Friel and Wyse; 2012)".
Despite these difficulties, BMA was extensively applied in the last 20 years, mostly in combining multiple models for predictive purposes and selecting models, particularly covariate sets in regression models or network structure in Bayesian Network models. The latter application induces another pitfall in the form of large model spaces. For instance, consider a regression model with p covariates. The number of possible models without any interaction coefficients is 2^p , which represents a large number of models even for moderate values of p. This difficulty can be mostly addressed by prior filtering of all possible models or through stochastic search algorithms over the model space.

In this case we are using BIC to find the prior.

Let denote the maximum likelihood estimator for model l, then the Bayes Factor between two models l and m can be reasonably approximated by the Bayesian information criteria (BIC), given by

- $2\log B_{lm} \approx 2\log\left(L\left(Y\middle|\hat{\theta}_l, M_l\right)\right) - \log\left(L\left(Y\middle|\hat{\theta}_m, M_m\right)\right) - (P_l - P_m)logN$

Y-> data

$\hat{\theta}_l$-> maximum likelihood estimator of model L

$\hat{\theta}_m$-> maximum likelihood estimator of model m

P_l-> number of parameters in model L

P_m-> number of parameters in model and N-> sample size.

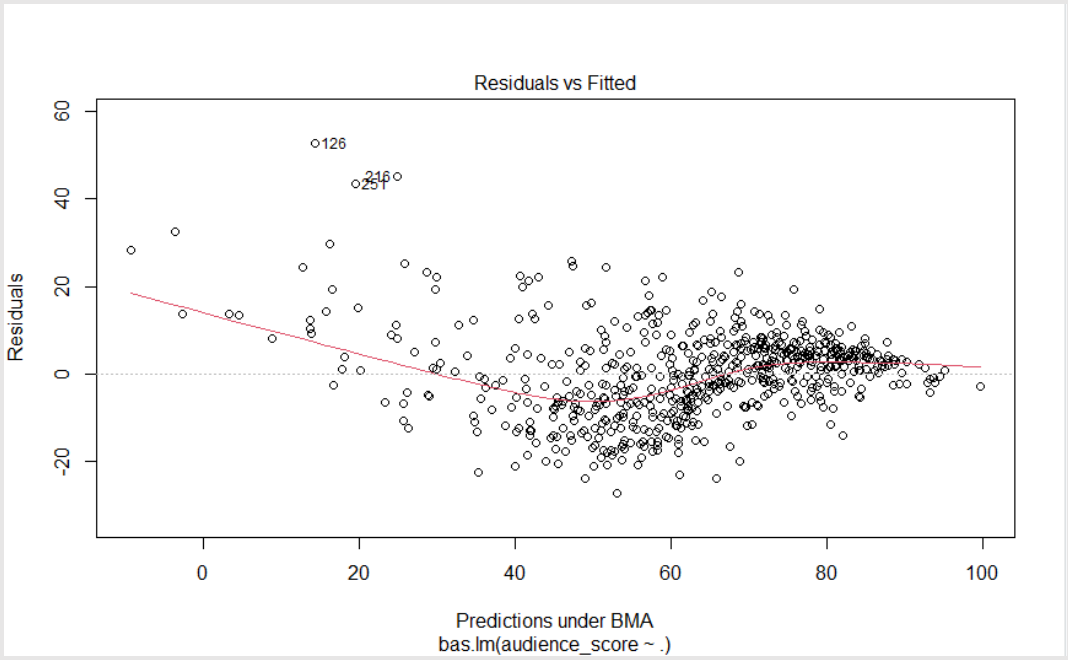
when both models are used to fit the same dataset of sample size N. The BIC provides a good approximation for many generalized linear models and enjoys widespread use, even with the larger approximation error of O(1). Both methods are very similar in spirit, and as such, were put into the same classification.

Diagnostics for model:

Residual variability :

#Rcode for Modeling

bma_model=bas.lm(audience_score~.,prior="BIC", modelprior= uniform(), data=movies_model)
plot(bma_model,1,add.smooth=TRUE)

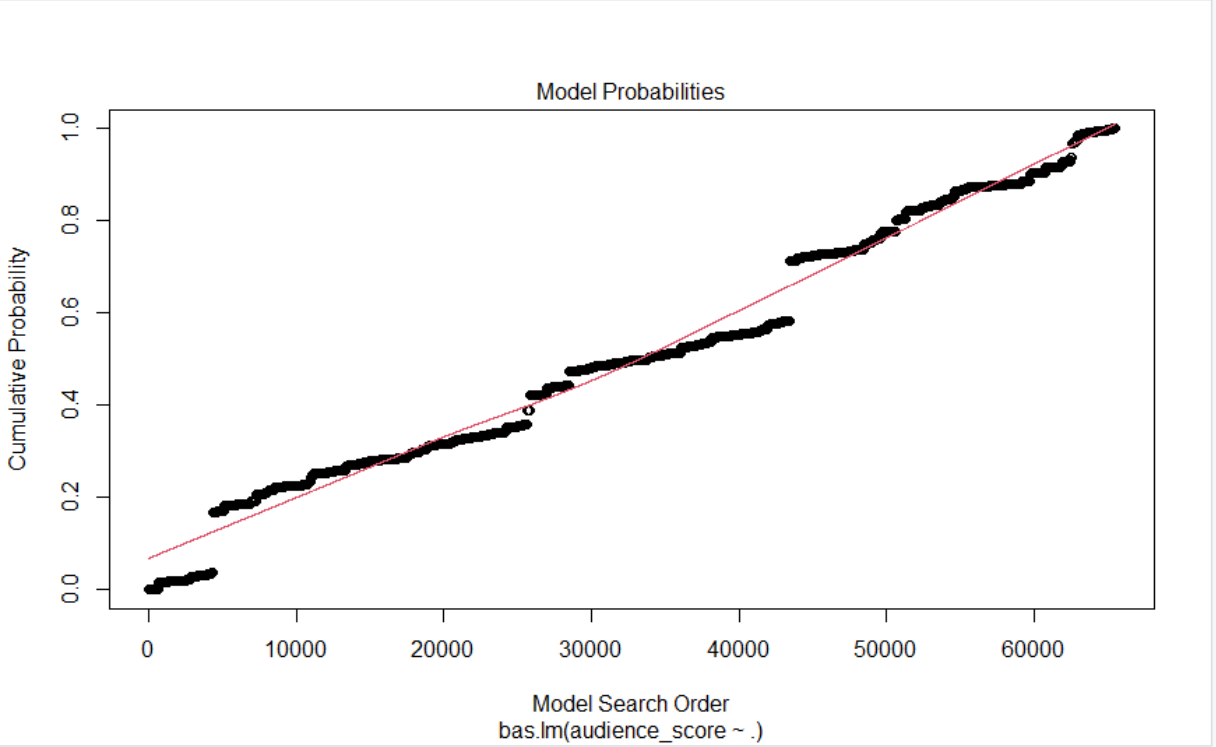


Residuals versus Fitted values, No random scatter is visible around the 0 line in the plot. It suggests that there is a non-constant variability (variance) in the data, particularly for the data with low audience score. As a result, with high scores, the model is probably going to make better predictions.

Model Probabilities:

#Rcode for Modeling

plot(bma_model,2,add.smooth=TRUE)

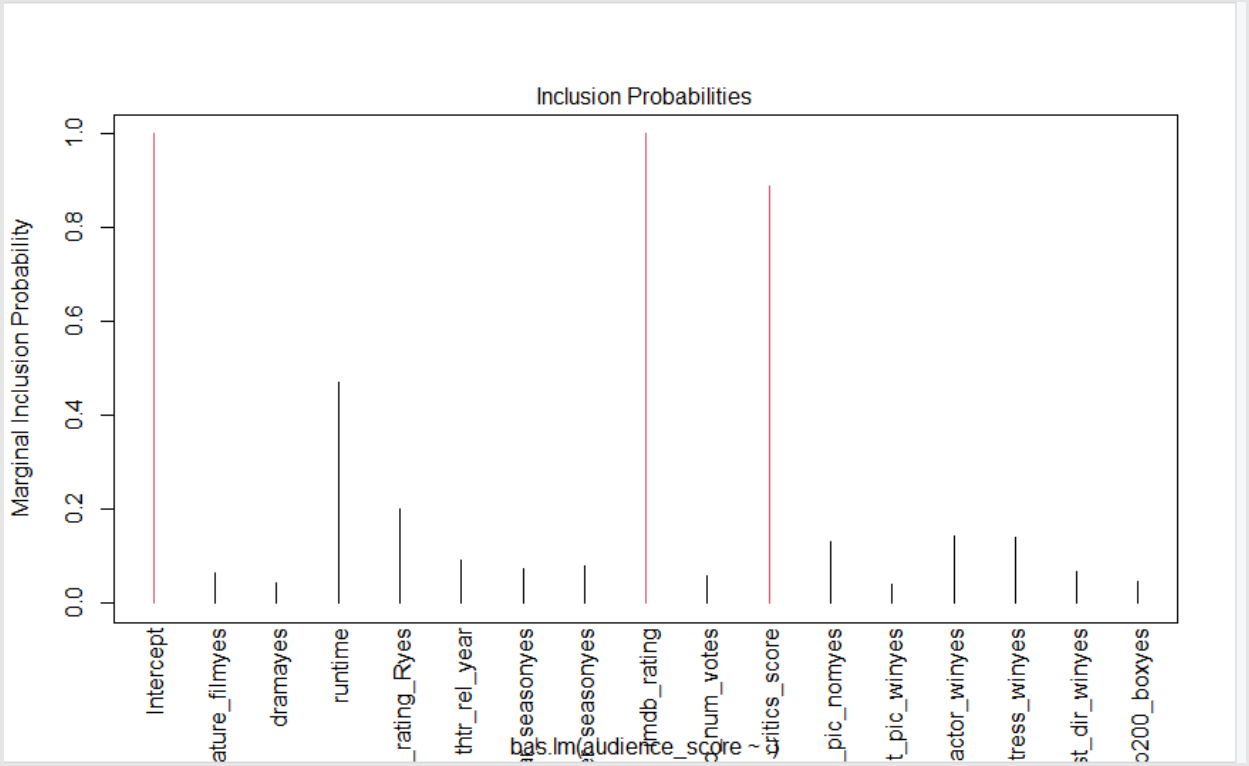


The model search is stopped at around 70000 number, instead of total 2^17 and also there is a linear regression line and also the probability is evenly distributed across each model.

Posterior Marginal Inclusion Probabilities:

#Rcode for Modeling

plot(bma_model,4)



The plot suggest that “imdb_rating” & “critics_score” have the posterior inclusion probability greater than 0.5, and are most likely to be included in the model.

Model Selection:

#Rcode for Model Selection

summary(bma_model)

```
> summary(bma_model)
                  P(B != 0 | Y)    model 1       model 2       model 3       model 4       model 5
Intercept            1.00000000     1.0000     1.0000000     1.0000000     1.0000000     1.0000000
feature_filmyes      0.06536947     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
dramayes             0.04319833     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
runtime              0.46971477     1.0000     0.0000000     0.0000000     0.0000000     1.0000000
mpaa_rating_Ryes     0.19984016     0.0000     0.0000000     0.0000000     1.0000000     1.0000000
thtr_rel_year        0.09068970     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
oscar_seasonyes      0.07505684     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
summer_seasonyes     0.08042023     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
imdb_rating          1.00000000     1.0000     1.0000000     1.0000000     1.0000000     1.0000000
imdb_num_votes       0.05773502     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
critics_score        0.88855056     1.0000     1.0000000     1.0000000     1.0000000     1.0000000
best_pic_nomyes      0.13119140     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
best_pic_winyes      0.03984766     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
best_actor_winyes    0.14434896     0.0000     0.0000000     1.0000000     0.0000000     0.0000000
best_actress_winyes  0.14128087     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
best_dir_winyes      0.06693898     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
top200_boxyes        0.04762234     0.0000     0.0000000     0.0000000     0.0000000     0.0000000
BF                           NA     1.0000     0.9968489     0.2543185     0.2521327     0.2391994
PostProbs                    NA     0.1297     0.1293000     0.0330000     0.0327000     0.0310000
R2                           NA     0.7549     0.7525000     0.7539000     0.7539000     0.7563000
dim                          NA     4.0000     3.0000000     4.0000000     4.0000000     5.0000000
logmarg                      NA -3615.2791 -3615.2822108 -3616.6482224 -3616.6568544 -3616.7095127
```

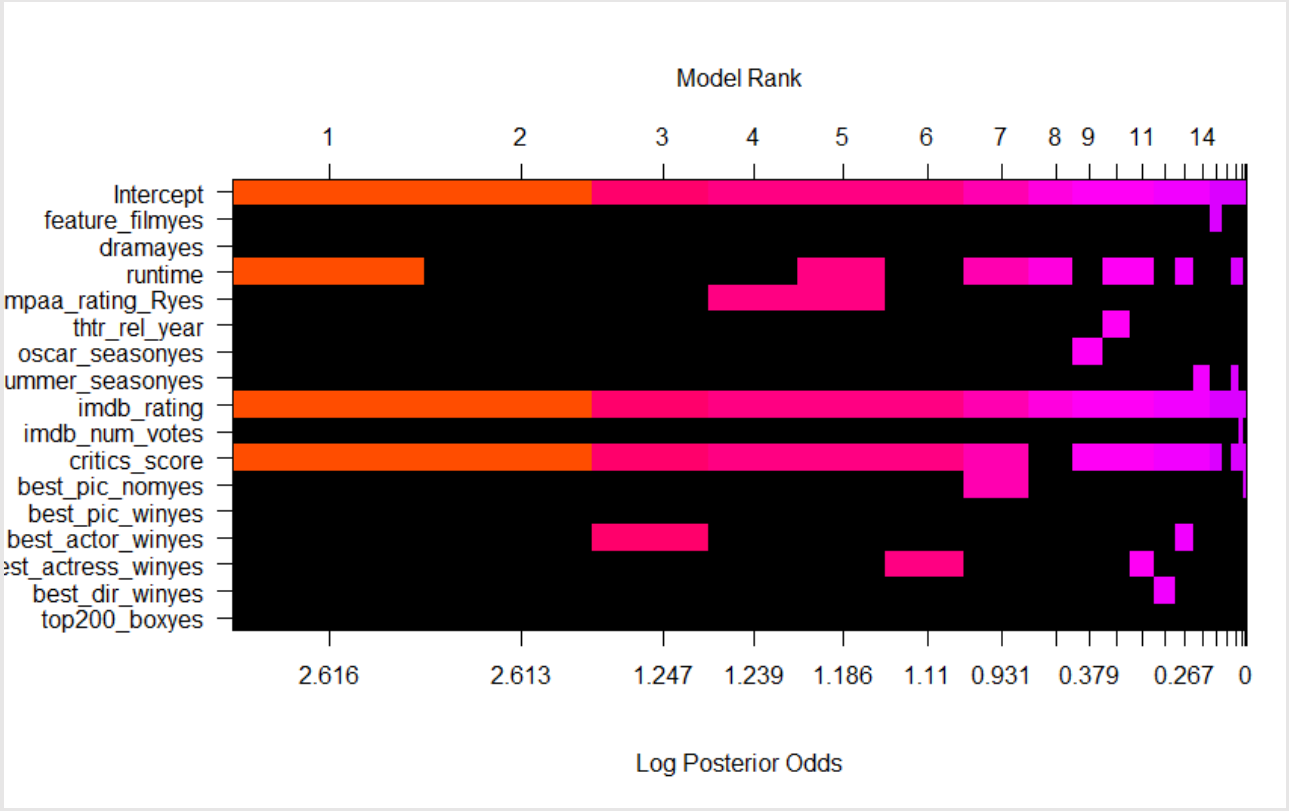
Here model 1 and model 2 has high Bayes factor and the predictor variables are very similar (imdb_rating , critics_score) except the model1 includes runtime only.

However, as the top 2 model only contribute to 26% and we still have 74% of model uncertainty. It is better to use a Bayesian Model Averaging (BMA) model to achieve an comprehensive prediction result.

Validating the model:

#Rcode for validating

image(bma_model, rotate=F)



From the visualization we can see that imdb_rating, critics_score and run time has the highest posterior probability of appearance.

This indicates that these variables should include in the audience score distribution.

Searching for correlation:

#Rcode for validating

```
data.frame("imdb_rating"=c(cor(movies_model$imdb_rating,movies_model$audience_score)),
           "critics_score"=c(cor(movies_model$critics_score,movies_model$audience_score)),
           "runtime"=c(cor(movies_model$runtime,movies_model$audience_score)))
```

```
  imdb_rating critics_score   runtime
1   0.8649091     0.7041573 0.1809629
>
```

From the Correlation coefficient, we can say that imdb_rating and critics_score are both strongly correlated with audience score and runtime is only slightly correlated.

#Rcode for validating

```
confint(coef(bma_model))
```

```
> confint(coef(bma_model))
                             2.5%        97.5%          beta
Intercept           61.565960606 6.313461e+01  6.234769e+01
feature_filmyes     -1.047390541 1.765179e-01 -1.046908e-01
dramayes             0.000000000 0.000000e+00  1.604413e-02
runtime             -0.082410653 0.000000e+00 -2.567772e-02
mpaa_rating_Ryes    -2.140081654 0.000000e+00 -3.036174e-01
thtr_rel_year       -0.053800192 0.000000e+00 -4.532635e-03
oscar_seasonyes     -0.933324179 0.000000e+00 -8.034940e-02
summer_seasonyes     0.000000000 1.031100e+00  8.704545e-02
imdb_rating         13.691690756 1.659472e+01  1.498203e+01
imdb_num_votes       0.000000000 2.351544e-06  2.080713e-07
critics_score        0.000000000 1.059417e-01  6.296648e-02
best_pic_nomyes     -0.004097843 5.040327e+00  5.068035e-01
best_pic_winyes      0.000000000 0.000000e+00 -8.502836e-03
best_actor_winyes   -2.626777605 0.000000e+00 -2.876695e-01
best_actress_winyes -2.875362005 0.000000e+00 -3.088382e-01
best_dir_winyes     -1.545115885 7.885410e-04 -1.195011e-01
top200_boxyes        0.000000000 0.000000e+00  8.648185e-02
attr(,"Probability")
[1] 0.95
attr(,"class")
[1] "confint.bas"
>
```

The Coefficients credible intervals show that there is evidence that imdb_rating , critics_score are positively associated with the audience score while longer films are scored less by the audience.

Prediction:

Now we are going to Predict the audience score for the movie Deadpool 2 which is released in the year 2018.

#Rcode for Predicting

```
dp2_df=data.frame(runtime =119, thtr_rel_year = 2018,  imdb_rating =7.7,imdb_num_votes =551,253,critics_score =84,best_pic_nom = "no", best_pic_win = "no", best_actor_win = "no",  best_actress_win = "no",best_dir_win = "no", top200_box = "no", feature_film = "yes", drama = "yes", mpaa_rating_R = "yes", oscar_season = "no", summer_season = "yes")

predict_result=predict(bma_model, newdata =dp2_df, estimator = "BMA",se.fit=TRUE)

confint(predict_result)
```

```
          2.5%    97.5%     pred
[1,] 62.28193 101.9299 81.66654
attr(,"Probability")
[1] 0.95
attr(,"class")
[1] "confint.bas"
>
```

The Predicted result is 81.7(with 95% credibal interval between 61.2 and 100.9). The actual audience score is 84 in rotten tomatoes. Therefore, the model can make a good prediction of audience score.

2. Predicting another movie audience score

Now, I have taken movie Batman Vs Superman : Dawn Of Justice which is released in the year 2016.

#Rcode for Predicting

```
bvs_df=data.frame(runtime =152, thtr_rel_year = 2016,imdb_rating =6.4,imdb_num_votes =685,249,critics_score =29,best_pic_nom = "no",best_pic_win = "no",best_actor_win = "no",best_actress_win = "no",best_dir_win = "no",top200_box = "no",feature_film = "yes",drama = "yes",mpaa_rating_R = "no",oscar_season = "no", summer_season = "no")


predict_result=predict(bma_model, newdata =bvs_df, estimator ="BMA",se.fit=TRUE)

confint(predict_result)
```

```
> confint(predict_result)
          2.5%    97.5%     pred
[1,] 38.33541 78.05844 58.10504
attr(,"Probability")
[1] 0.95
attr(,"class")
[1] "confint.bas"
>
```

The Predicted result is 58.1(with 95% credibal interval between 38.33 and 78.05). The actual audience score is 60 in rotten tomatoes. Therefore, the model can make a good prediction of audience score.

Summary:

The Project main aim is to build a Bayesian regression model(linear-model) which predicts audience score of movies based on different attributes.

In obtaining the Model I have used Bayesian Information criterion(BIC) which is used to explore model uncertainty of posterior probability.

Each model are assigned with different weights, corresponding to their posterior probability, and then compute to a Bayesian Model Averaging (BMA) model. Finally, we use the model to predict the audience score of 2 movies.

References:

1. Brown, P. J., Vannucci, M. and Fearn, T. (2002). Bayes model averaging with selection of regressors, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64(3): 519–536.

2. Conti, D. V., Cortessis, V., Molitor, J. and Thomas, D. C. (2003). Bayesian modeling of complex metabolic pathways, Human heredity 56(1-3): 83–93

3. Fernandez, C., Ley, E. and Steel, M. F. J. (2001a). Benchmark priors for bayesian model averaging, Journal of Econometrics 100(2): 381–427.

4. George, E. I. and McCulloch, R. E. (1997). Approaches for bayesian variable selection, Statistica sinica 7(2): 339–373.

5. Hoeting, J. A., Madigan, D., Raftery, A. E. and Volinsky, C. T. (1999). Bayesian model averaging: a tutorial, Statistical science pp. 382–401.

6. Madigan, D., York, J. and Allard, D. (1995). Bayesian graphical models for discrete data, International Statistical Review/Revue Internationale de Statistique pp. 215–232.