STAT 542, homework 5

Shuhui Guo

Question 1

$$\begin{aligned}
(A - bb^T)(A^{-1} + \frac{A^{-1}bb^TA^{-1}}{1 - b^TA^{-1}b}) &= AA^{-1} - bb^TA^{-1} + \frac{AA^{-1}bb^TA^{-1} - bb^TA^{-1}bb^TA^{-1}}{1 - b^TA^{-1}b} \\
&= I - bb^TA^{-1} + \frac{bb^TA^{-1} - bb^TA^{-1}bb^TA^{-1}}{1 - b^TA^{-1}b} \\
&= I - bb^TA^{-1} + \frac{(b - bb^TA^{-1}b)b^TA^{-1}}{1 - b^TA^{-1}b} \\
&= I - bb^TA^{-1} + \frac{b(1 - b^TA^{-1}b)b^TA^{-1}}{1 - b^TA^{-1}b} \\
&= I - bb^TA^{-1} + bb^TA^{-1} \\
&= I
\end{aligned}$$

Therefore, $(A - bb^T)^{-1} = A^{-1} + \frac{A^{-1}bb^TA^{-1}}{1-b^TA^{-1}b}$

Question 2

First I wrote my own code to fit the sliced inverse regression. To validate my code, I then generated X with 1000 observations and 10 variables from standard normal distribution. That is, the number of rows $n = 1000$ and the number of columns $p = 10$. Y is generated by:

$$y = 0.125(x\beta)^3 + 0.5\epsilon, \quad \epsilon \sim N(0, 1)$$

where $\beta^T = (1, 1, 0, 0, 0, 0, 0, 0, 0, 0)$.

Let scilces=10, the first 10 eigenvectors derived by my code is:

```
##              [,1]        [,2]        [,3]       [,4]       [,5]
##  [1,] -0.731116714  0.06390168  0.11638999  0.2407000  0.2490442
##  [2,] -0.675336375 -0.05338859 -0.08978788 -0.2828972 -0.2516881
##  [3,]  0.022874624 -0.22041797 -0.62118323  0.4000343 -0.1696594
##  [4,]  0.002419200 -0.66363611 -0.32226807 -0.2898965  0.4535309
##  [5,] -0.002852599 -0.41043252  0.18974075 -0.1845853 -0.4939976
##  [6,]  0.048720637 -0.17571497 -0.28471468  0.1471128 -0.3608132
```

```
##  [7,] -0.043418239 -0.33197455  0.19168006  0.4064990 -0.3138105
##  [8,]  0.049442112 -0.28862029  0.22987758 -0.1738746  0.1254729
##  [9,] -0.041369692  0.26953484 -0.42675453 -0.5823210 -0.3167355
## [10,]  0.020906347 -0.19642011  0.31606616 -0.1663724 -0.2267019
##              [,6]        [,7]        [,8]         [,9]        [,10]
##  [1,] -0.10066166  0.51524326  0.16827746  0.071367837  0.003410259
##  [2,]  0.06304002 -0.52820317 -0.24199138  0.001660962 -0.115615829
##  [3,]  0.21940432 -0.08420844  0.41215810 -0.096173704 -0.344893124
##  [4,]  0.03645524  0.05320727 -0.08241458  0.034737885  0.323072186
##  [5,]  0.37309307  0.54164748 -0.15278924 -0.162676062 -0.266610877
##  [6,] -0.38485534  0.21479466 -0.29779816  0.705908212  0.135288632
##  [7,] -0.40861658 -0.11342861  0.06367421 -0.452354881  0.366630525
##  [8,] -0.58184673 -0.09450686  0.14210557  0.047998067 -0.663398897
##  [9,] -0.30696173  0.23941708  0.32592899 -0.222404038  0.179971812
## [10,]  0.22220540 -0.16421587  0.70281577  0.450871997  0.258851658
```

The first 10 eigenvectors derived by the "dr" package is:

```
##             Dir1        Dir2        Dir3       Dir4       Dir5        Dir6
## X1   0.731116714  0.06390168  0.11638999 -0.2407000  0.2490442  0.10066166
## X2   0.675336375 -0.05338859 -0.08978788  0.2828972 -0.2516881 -0.06304002
## X3  -0.022874624 -0.22041797 -0.62118323 -0.4000343 -0.1696594 -0.21940432
## X4  -0.002419200 -0.66363611 -0.32226807  0.2898965  0.4535309 -0.03645524
## X5   0.002852599 -0.41043252  0.18974075  0.1845853 -0.4939976 -0.37309307
## X6  -0.048720637 -0.17571497 -0.28471468 -0.1471128 -0.3608132  0.38485534
## X7   0.043418239 -0.33197455  0.19168006 -0.4064990 -0.3138105  0.40861658
## X8  -0.049442112 -0.28862029  0.22987758  0.1738746  0.1254729  0.58184673
## X9   0.041369692  0.26953484 -0.42675453  0.5823210 -0.3167355  0.30696173
## X10 -0.020906347 -0.19642011  0.31606616  0.1663724 -0.2267019 -0.22220540
##            Dir7        Dir8         Dir9        Dir10
## X1  -0.51524326 -0.16827746  0.071367837 -0.003410259
## X2   0.52820317  0.24199138  0.001660962  0.115615829
## X3   0.08420844 -0.41215810 -0.096173704  0.344893124
## X4  -0.05320727  0.08241458  0.034737885 -0.323072186
## X5  -0.54164748  0.15278924 -0.162676062  0.266610877
## X6  -0.21479466  0.29779816  0.705908212 -0.135288632
## X7   0.11342861 -0.06367421 -0.452354881 -0.366630525
## X8   0.09450686 -0.14210557  0.047998067  0.663398897
## X9  -0.23941708 -0.32592899 -0.222404038 -0.179971812
## X10  0.16421587 -0.70281577  0.450871997 -0.258851658
```

Based on the above results, the directions derived by my code and "dr" package are all the same. Although some results have opposite sign, the directions defined by eigenvectors are the same. So my code is validated to be correct.

a)

In this question, the data X I generated using an underlying model that can be detected by SIR has 1000 observations and 10 variables. That is, the number of rows $n = 1000$ and the number of columns $p = 10$. X is generated by standard normal distribution. Y is generated by:

$$y = 5sin(x\beta) + \epsilon, \quad \epsilon \sim N(0, 1)$$

where $\beta^T = (1, 0, 0, 1, 0, 0, 0, 0, 0, 0)$.

My estimated direction is:

$$\begin{array}{lll} \frac{\hat{\beta}_2}{\hat{\beta}_1} = -0.05 & \frac{\hat{\beta}_3}{\hat{\beta}_1} = 0.09 & \frac{\hat{\beta}_4}{\hat{\beta}_1} = 0.94 \\ \frac{\hat{\beta}_5}{\hat{\beta}_1} = 0.06 & \frac{\hat{\beta}_6}{\hat{\beta}_1} = -0.03 & \frac{\hat{\beta}_7}{\hat{\beta}_1} = 0.08 \\ \frac{\hat{\beta}_8}{\hat{\beta}_1} = -0.05 & \frac{\hat{\beta}_9}{\hat{\beta}_1} = -0.01 & \frac{\hat{\beta}_{10}}{\hat{\beta}_1} = -0.09 \end{array}$$

The true direction is:

$$\begin{array}{lll} \frac{\beta_2}{\beta_1} = 0 & \frac{\beta_3}{\beta_1} = 0 & \frac{\beta_4}{\beta_1} = 1 \\ \frac{\beta_5}{\beta_1} = 0 & \frac{\beta_6}{\beta_1} = 0 & \frac{\beta_7}{\beta_1} = 0 \\ \frac{\beta_8}{\beta_1} = 0 & \frac{\beta_9}{\beta_1} = 0 & \frac{\beta_{10}}{\beta_1} = 0 \end{array}$$

Based on the above results, my estimated direction are very close to true direction. So the SIR could detect the underlying model I generated before. It is because that the generating function $y = 5sin(x\beta)$ is a linear function. Therefore, SIR can detect the underlying model by its first moment information.

b)

In this question, the data X I generated using an underlying model that cannot be detected by SIR has 1000 observations and 10 variables. That is, the number of rows $n = 1000$ and the

number of columns $p = 10$. X is generated by standard normal distribution. Y is generated by:

$$y = 3(cos(x\beta))^2 + \epsilon, \quad \epsilon \sim N(0, 1)$$

where $\beta^T = (1, 1, 1, 0, 0, 0, 0, 0, 0, 0)$.

My estimated direction is:

$$\begin{array}{lll} \frac{\hat{\beta}_2}{\hat{\beta}_1} = -0.55 & \frac{\hat{\beta}_3}{\hat{\beta}_1} = 0.86 & \frac{\hat{\beta}_4}{\hat{\beta}_1} = 0.28 \\ \frac{\hat{\beta}_5}{\hat{\beta}_1} = 3.25 & \frac{\hat{\beta}_6}{\hat{\beta}_1} = 1.28 & \frac{\hat{\beta}_7}{\hat{\beta}_1} = -1.11 \\ \frac{\hat{\beta}_8}{\hat{\beta}_1} = -0.18 & \frac{\hat{\beta}_9}{\hat{\beta}_1} = 1.28 & \frac{\hat{\beta}_{10}}{\hat{\beta}_1} = 1.21 \end{array}$$

The true direction is:

$$\begin{array}{lll} \frac{\beta_2}{\beta_1} = 1 & \frac{\beta_3}{\beta_1} = 1 & \frac{\beta_4}{\beta_1} = 1 \\ \frac{\beta_5}{\beta_1} = 0 & \frac{\beta_6}{\beta_1} = 0 & \frac{\beta_7}{\beta_1} = 0 \\ \frac{\beta_8}{\beta_1} = 0 & \frac{\beta_9}{\beta_1} = 0 & \frac{\beta_{10}}{\beta_1} = 0 \end{array}$$

Based on the above results, my estimated direction are quitely different from true direction. So the SIR could not detect this underlying model. It is because that the generating function $y = 3(cos(x\beta))^2$ is not a linear function. SIR can not detect the underlying model by its first moment information, which is the way that SIR applies. Therefore, SIR cannot detect this underlying model.

Question 3

In this part, there are two questions to be settled. One is to solve a regression problem to predict the variable **revenue**. The other one is to slove a classification problem to predict whether the **vote_average** is greater than 7.

To deal with these problems, first I conduct variable selections, including the following steps:

i) Remove the variables **vote_count** and **popularity**. The information in these variables is after the release date so they should not be used to do prediction.

ii) Remove the variables **homepage**, **original_language**, **original_title**, **overview**, **release_date**, **spoken_language**, **status**, **tagline**, **title** because they cannot provide much information to solve the two questions.

So the variables I will use for prediction are: **budget**, **genres**, **keywords**, **production_companies**, **production_countries**, **runtime**.

Then I preprocess the data in the following steps:

i) Expand the categorical variables **genres**, **keywords**, **production_companies**, **production_countries** into dummy variables which only contain 0 and 1 levels in each variable. For example, for the variable **genres**, there are 20 types such as "Action", "Adventure", "Fantasy". Therefore **genres** could be transformed into 20 dummy variables. If one type is included in genres of a movie, the level of this type would be 1, otherwise the level would be 0.

ii) Remove the dummy variables containing all level 0 or only one level 1. Since these variables cannot provide much infromation to prediction.

iii) Select the variables whose correlation with **revenue**, **vote_average** is not small. For the first question to predict **revenue**, the threshold of correlation is 0.1. The variables whose correlation with **revenue** not less than 0.1 are selected. For the second question to predict **vote_average**, the threshold is 0.05. The variables whose correlation with **vote_average** not less than 0.05 are selected.

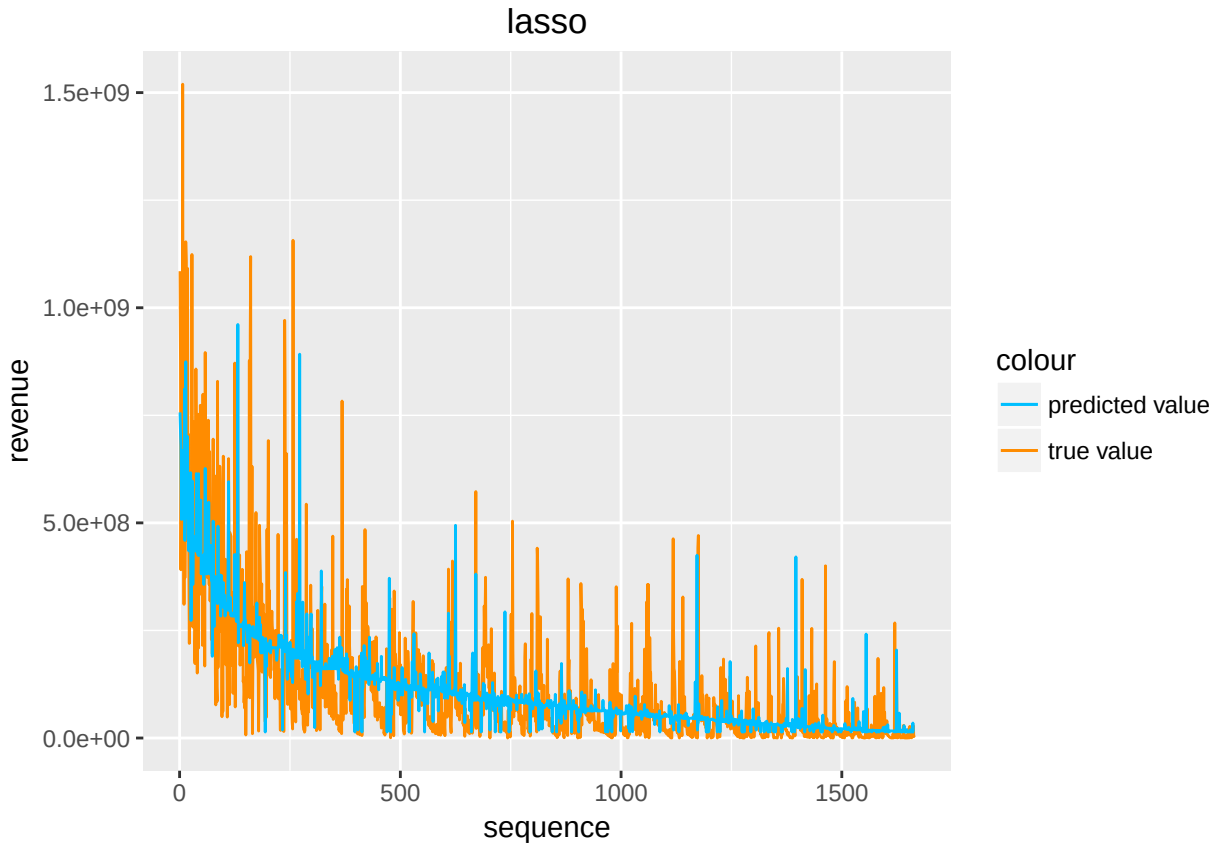
After above steps, I will fit models to each question.

Part I. Predict Revenue

In this question, first I will remove the observations whose revenue equals 0 since these are missing values and should not be included in the prediction. Then I will try two models, lasso and random forest to do prediction. Finally I will use the mean absolute error to compare their prediction effect and select the model with the smallest error to make conclusions.

Lasso

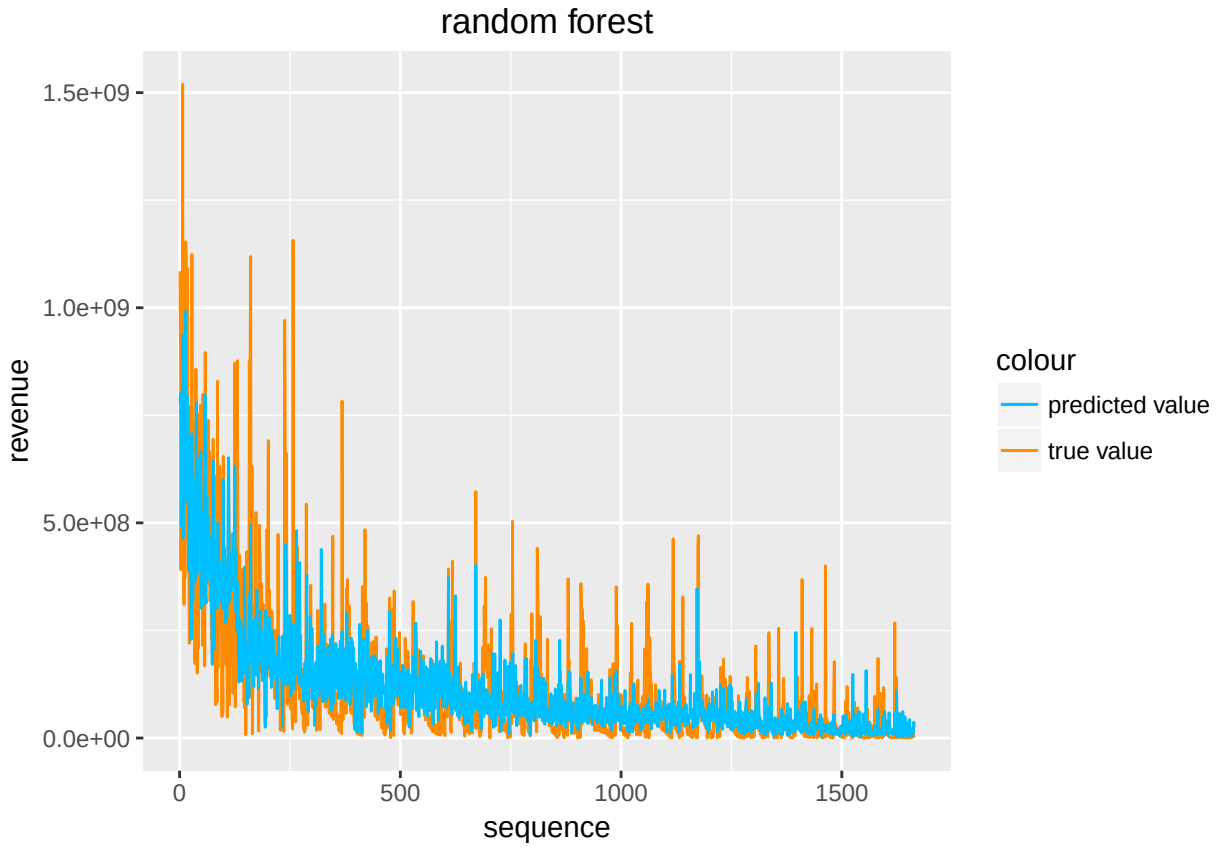
For the lasso model, I first use training data to do 10 folds cross validation to find the best λ, which is $log(\lambda) = 16$. Then I use the model with this λ to fit testing data and get prediction values. The comparison of testing data and prediction values is plotted as below:



In this plot, the trend of testing data is well fitted by prediction values. The prediction error is 6.8951974×10^7.

Random Forest

For random forest model, I set the parameter $ntree = 1000$, $mtry = p/3 = 139/3$, $nodesize = 5$. Then I use this model to fit testing data and get prediction values. The comparison of testing data and prediction values is plotted as below:

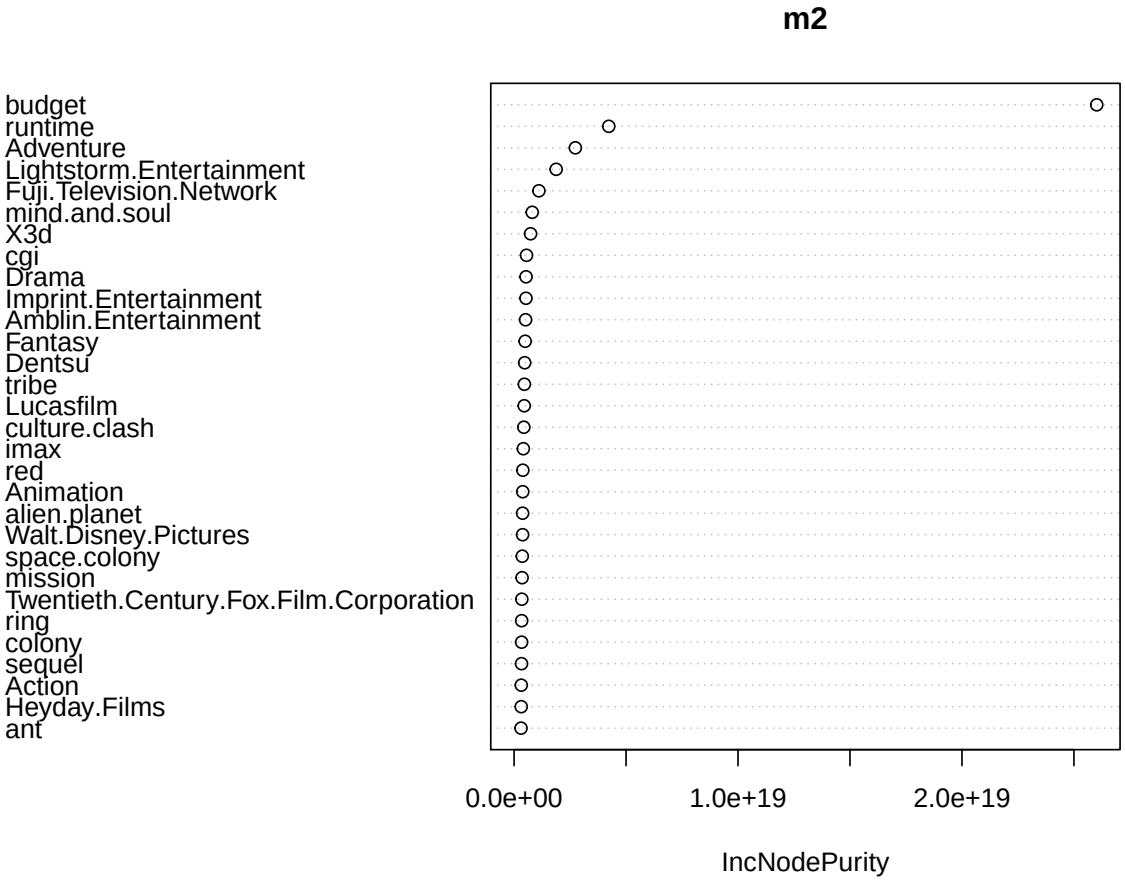


In this plot, the trend of testing data is well fitted by prediction values. The prediction error is 6.5523848×10^7.

Model Selection

Based on the above results, the prediction error of random forest is much smaller, which is 6.5523848×10^7. So random forest is chosen to be the final model.

The importance of predictors given by random forest can be shown as below:



Based on this plot, there are 6 predictor that are relatively more important in this model. The most important predictor is **budget**. It matches the real condition because whether the movie production is excellent is mostly decided by the budget. And the excellence of movie production is closely related to movie's revenue. The second important predictor is **runtime**. The runtime of a movie is related to the complexity and richness of content so that it is important for revenue. The third important predictor is **Adventure** in **genres**. The fourth important predictor **Lightstorm.Entertainment** and fifth important predictor **Fuji.Television.Network** are names of production companies. And the sixth important predictor **mind.and.soul** is a type of key words of movies.

Part II. Predict vote_average

In this question, first I will classify expand **vote_average** to a dummy variable that 1 is value of **vote_average** greater than 7 and otherwise is 0. However, the rate of level 1 is around 0.174 while the rate of level 0 is around 0.83, which are unbalanced. So I will use all level 1 data and randomly select the same amount of level 0 data to be training data. Thus they are balanced to do prediction. Then I will try three models, lasso, SVM and random forest to do prediction. Finally I will use the misclassification error to compare their

prediction effect and select the model with the smallest error to make conclusions.

Lasso

For the lasso model, I first use training data to do 10 folds cross validation to find the best λ, which is 0. Then I use the model with this λ to fit testing data and get prediction values. The misclassification error is 0.1894515.

SVM

For SVM, I first try the cost value from 1 to 31 by 5 and compare their misclassification error to find the best model. Then I use this model to fit testing data and get prediction values. The misclassification error is 0.2012658. And the cost for selected model is 1.

Random Forest

For random forest model, I set the parameter $ntree = 1000$, $mtry = p/3 = 351/3$, $nodesize = 5$. Then I use this model to fit testing data and get prediction values. The misclassification error is 0.2054852.

Model Selection

Based on the above results, the misclassification error of lasso is the smallest, which is 0.1894515. So lasso is chosen to be the final model.

Since the larger the coefficient of a variable, the more influence this variable could exert on the prediction values. Hence I find the variables with the largest 20 coefficients, which are relatively important predictors:

```
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient
```

```
##  [1] "Thriller"                "United.States.of.America"
##  [3] "Comedy"                  "Horror"
##  [5] "Action"                  "cult"
##  [7] "music"                   "rage.and.hate"
##  [9] "country"                 "normandy"
## [11] "district.attorney"       "vigilante"
## [13] "dealer"                  "persecution"
## [15] "extraterrestrial"        "love"
## [17] "swastika"                "crooked.lawyer"
## [19] "trauma"                  "rehab"
```

From the results, we could know that the top 20 important predictors mostly belong to **genres** and **keywords**. Also there is a predictor **United.States.of.America** belonging

to **production_countries**. However, there are no **budget** and **runtime** like part I. The reason could be that the influence of these two variables on average rating is not much. On the other hand, audiences focus more on movies' genres and contents while they are rating movies.

Predict "Star Wars: The Last Jedi"

For the revenue part, I use random forest selected in Part I. This movie's information included in the important predictors selected by random forest are **budget** $= 2.45 \times 10^8$, **runtime** $= 152$, and **Adventure**, **Action**, **Fantasy** type in **genres**. Hence, I use these five predictors to do the prediction. The predicted revenue is 7.7650499×10^8.

For the rating part, since the information released is not included in the important predictors selected by lasso in Part II, it is not easy to do prediction using existed models. Nevertheless, the rating could still be predicted as greater than 7. Since the rating of previous "Star Wars" series are all greater than 7 and the budget of this "Star Wars" movie is so high that the movie production could be excellent. Therefore, the rating could be greater than 7.