hw04-02

Weijun Zhu

December 5, 2019

Contents


```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

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

```
library(rjson)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
```

```
## The following objects are masked from 'package:rjson':
##
##     fromJSON, toJSON

## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(e1071)
set.seed(9)
```

2.

R/Python Project: We continue with the movie data set you used in the previous homework.

2a).

Read the original data into movieDat as you did in the previous homework. The column under the genres is in the JSON format (check Wikipedia to get familiar with this simple format.) Each movie may belong to several genres. You must parse this column for all movies, collect the set of all available genres, and for each one create a new binary feature whose name starts with genre_. So after this pre-processing you should have new features such as genre Action, genre Adventure etc. Since the format in which the genres are stored in this data set is JSON, you may wish to look into the relevant libraries in R and Pythos. In R you may wish to look at libraries rjson and litejson for utilities working with JSON format. In Python import json will load the necessary library items. See this page for more information on Python. Of course, you could ignore the JSON libraries and use direct string processing to extract genre names, but this may be more time-consuming. (For now ignore the other JSON features in the data.)

```
movieDat <- read.csv('C:/Users/zhuwe/Desktop/ML/ML_Homework/hw_code/data/tmdb_5000_movies.csv',
                     header = T)
head(movieDat)
```

```
##      budget
## 1 237000000
## 2 300000000
## 3 245000000
## 4 250000000
## 5 260000000
```

```
## 6 258000000
##
## 1 [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id
## 2                                        [{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fanta
## 3                                          [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adven
## 4              [{"id": 28, "name": "Action"}, {"id": 80, "name": "Crime"}, {"id": 18, "name": "Drama
## 5                                [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id
## 6                                          [{"id": 14, "name": "Fantasy"}, {"id": 28, "name": "Action
##                                       homepage     id
## 1                    http://www.avatarmovie.com/  19995
## 2   http://disney.go.com/disneypictures/pirates/    285
## 3    http://www.sonypictures.com/movies/spectre/ 206647
## 4             http://www.thedarkknightrises.com/  49026
## 5           http://movies.disney.com/john-carter  49529
## 6 http://www.sonypictures.com/movies/spider-man3/    559
##
## 1                           [{"id": 1463, "name": "culture clash"}, {"id": 2964, "name": "future"}
## 2
## 3
## 4 [{"id": 849, "name": "dc comics"}, {"id": 853, "name": "crime fighter"}, {"id": 949, "name": "terro
## 5
## 6
##   original_language                          original_title
## 1                en                                  Avatar
## 2                en Pirates of the Caribbean: At World's End
## 3                en                                 Spectre
## 4                en                  The Dark Knight Rises
## 5                en                            John Carter
## 6                en                          Spider-Man 3
##
## 1
## 2
## 3
## 4 Following the death of District Attorney Harvey Dent, Batman assumes responsibility for Dent's crim
## 5                                                                                    John Carter i
## 6
##   popularity
## 1  150.43758
## 2  139.08262
## 3  107.37679
## 4  112.31295
## 5   43.92699
## 6  115.69981
##
## 1 [{"name": "Ingenious Film Partners", "id": 289}, {"name": "Twentieth Century Fox Film Corporation"
## 2                                                                [{"name": "Walt Disney Pictures", "id
## 3
## 4                                              [{"name": "Legendary Pictures", "id": 923}, {"name"
## 5
## 6                                                                          [{"name": "Columbia Pictures"
##                                                                                          production_co
## 1 [{"iso_3166_1": "US", "name": "United States of America"}, {"iso_3166_1": "GB", "name": "United Kin
## 2                                                [{"iso_3166_1": "US", "name": "United States of Ame
## 3 [{"iso_3166_1": "GB", "name": "United Kingdom"}, {"iso_3166_1": "US", "name": "United States of Ame
```

```
## 4                                                    [{"iso_3166_1": "US", "name": "United States of Am
## 5                                                    [{"iso_3166_1": "US", "name": "United States of Am
## 6                                                    [{"iso_3166_1": "US", "name": "United States of Am
##   release_date    revenue runtime
## 1   2009-12-10 2787965087     162
## 2   2007-05-19  961000000     169
## 3   2015-10-26  880674609     148
## 4   2012-07-16 1084939099     165
## 5   2012-03-07  284139100     132
## 6   2007-05-01  890871626     139
##
## 1
## 2
## 3 [{"iso_639_1": "fr", "name": "Fran\\u00e7ais"}, {"iso_639_1": "en", "name": "English"}, {"iso_639_
## 4
## 5
## 6
##     status                                          tagline
## 1 Released                     Enter the World of Pandora.
## 2 Released At the end of the world, the adventure begins.
## 3 Released                           A Plan No One Escapes
## 4 Released                                The Legend Ends
## 5 Released           Lost in our world, found in another.
## 6 Released                              The battle within.
##                                    title vote_average vote_count
## 1                                 Avatar          7.2      11800
## 2 Pirates of the Caribbean: At World's End          6.9       4500
## 3                                Spectre          6.3       4466
## 4                    The Dark Knight Rises          7.6       9106
## 5                             John Carter          6.1       2124
## 6                            Spider-Man 3          5.9       3576
```

```
glimpse(movieDat)
```

```
## Observations: 4,803
## Variables: 20
## $ budget               <int> 237000000, 300000000, 245000000, 25000000...
## $ genres               <fct> "[{\"id\": 28, \"name\": \"Action\"}, {\"...
## $ homepage             <fct> http://www.avatarmovie.com/, http://disne...
## $ id                   <int> 19995, 285, 206647, 49026, 49529, 559, 38...
## $ keywords             <fct> "[{\"id\": 1463, \"name\": \"culture clas...
## $ original_language    <fct> en, en, en, en, en, en, en, en, en, en, e...
## $ original_title       <fct> Avatar, Pirates of the Caribbean: At Worl...
## $ overview             <fct> "In the 22nd century, a paraplegic Marine...
## $ popularity           <dbl> 150.43758, 139.08262, 107.37679, 112.3129...
## $ production_companies <fct> "[{\"name\": \"Ingenious Film Partners\",...
## $ production_countries <fct> "[{\"iso_3166_1\": \"US\", \"name\": \"Un...
## $ release_date         <fct> 2009-12-10, 2007-05-19, 2015-10-26, 2012-...
## $ revenue              <dbl> 2787965087, 961000000, 880674609, 1084939...
## $ runtime              <dbl> 162, 169, 148, 165, 132, 139, 100, 141, 1...
## $ spoken_languages     <fct> "[{\"iso_639_1\": \"en\", \"name\": \"Eng...
## $ status               <fct> Released, Released, Released, Released, R...
## $ tagline              <fct> "Enter the World of Pandora.", "At the en...
## $ title                <fct> Avatar, Pirates of the Caribbean: At Worl...
```

```
## $ vote_average         <dbl> 7.2, 6.9, 6.3, 7.6, 6.1, 5.9, 7.4, 7.3, 7...
## $ vote_count           <int> 11800, 4500, 4466, 9106, 2124, 3576, 3330...
```

Create new binary new features, but the new features start with "genre_" are hard to create, so I just keep the genre names. Then I set the genres for each movie.

```
k = 1
for (i in movieDat$genres) {
  json_file <- as.character(movieDat$genres[i])  # change "genres" to string
  temp <- jsonlite::fromJSON(i)
  for (j in temp["name"]) {
    movieDat[k,j] <- 1
  }
  k = k + 1
}
head(movieDat)
```

```
##      budget
## 1 237000000
## 2 300000000
## 3 245000000
## 4 250000000
## 5 260000000
## 6 258000000
##
## 1 [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"i
## 2                                          [{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fanta
## 3                                            [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adven
## 4              [{"id": 28, "name": "Action"}, {"id": 80, "name": "Crime"}, {"id": 18, "name": "Drama
## 5                                  [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"i
## 6                                           [{"id": 14, "name": "Fantasy"}, {"id": 28, "name": "Action
##                                        homepage     id
## 1                     http://www.avatarmovie.com/  19995
## 2    http://disney.go.com/disneypictures/pirates/    285
## 3     http://www.sonypictures.com/movies/spectre/ 206647
## 4              http://www.thedarkknightrises.com/  49026
## 5            http://movies.disney.com/john-carter  49529
## 6 http://www.sonypictures.com/movies/spider-man3/    559
##
## 1                            [{"id": 1463, "name": "culture clash"}, {"id": 2964, "name": "future"}
## 2
## 3
## 4 [{"id": 849, "name": "dc comics"}, {"id": 853, "name": "crime fighter"}, {"id": 949, "name": "terr
## 5
## 6
##   original_language                          original_title
## 1                en                                  Avatar
## 2                en Pirates of the Caribbean: At World's End
## 3                en                                 Spectre
## 4                en                  The Dark Knight Rises
## 5                en                             John Carter
## 6                en                           Spider-Man 3
##
## 1
```

```
## 2
## 3
## 4 Following the death of District Attorney Harvey Dent, Batman assumes responsibility for Dent's crim
## 5                                                                                          John Carter i
## 6
##   popularity
## 1  150.43758
## 2  139.08262
## 3  107.37679
## 4  112.31295
## 5   43.92699
## 6  115.69981
##
## 1 [{"name": "Ingenious Film Partners", "id": 289}, {"name": "Twentieth Century Fox Film Corporation"
## 2                                                                  [{"name": "Walt Disney Pictures", "id
## 3
## 4                                                [{"name": "Legendary Pictures", "id": 923}, {"name"
## 5
## 6                                                                       [{"name": "Columbia Pictures"
##                                                                                                production_co
## 1 [{"iso_3166_1": "US", "name": "United States of America"}, {"iso_3166_1": "GB", "name": "United Kin
## 2                                                 [{"iso_3166_1": "US", "name": "United States of Ame
## 3 [{"iso_3166_1": "GB", "name": "United Kingdom"}, {"iso_3166_1": "US", "name": "United States of Ame
## 4                                                 [{"iso_3166_1": "US", "name": "United States of Ame
## 5                                                 [{"iso_3166_1": "US", "name": "United States of Ame
## 6                                                 [{"iso_3166_1": "US", "name": "United States of Ame
##   release_date    revenue runtime
## 1   2009-12-10 2787965087     162
## 2   2007-05-19  961000000     169
## 3   2015-10-26  880674609     148
## 4   2012-07-16 1084939099     165
## 5   2012-03-07  284139100     132
## 6   2007-05-01  890871626     139
##
## 1
## 2
## 3 [{"iso_639_1": "fr", "name": "Fran\\u00e7ais"}, {"iso_639_1": "en", "name": "English"}, {"iso_639_1
## 4
## 5
## 6
##     status                                          tagline
## 1 Released                     Enter the World of Pandora.
## 2 Released At the end of the world, the adventure begins.
## 3 Released                          A Plan No One Escapes
## 4 Released                               The Legend Ends
## 5 Released           Lost in our world, found in another.
## 6 Released                             The battle within.
##                                  title vote_average vote_count Action
## 1                               Avatar          7.2      11800      1
## 2 Pirates of the Caribbean: At World's End          6.9       4500      1
## 3                              Spectre          6.3       4466      1
## 4                The Dark Knight Rises          7.6       9106      1
## 5                          John Carter          6.1       2124      1
## 6                         Spider-Man 3          5.9       3576      1
```

```
##   Adventure Fantasy Science Fiction Crime Drama Thriller Animation Family
## 1         1       1               1    NA    NA       NA        NA     NA
## 2         1       1              NA    NA    NA       NA        NA     NA
## 3         1      NA              NA     1    NA       NA        NA     NA
## 4        NA      NA              NA     1     1        1        NA     NA
## 5         1      NA               1    NA    NA       NA        NA     NA
## 6         1       1              NA    NA    NA       NA        NA     NA
##   Western Comedy Romance Horror Mystery History War Music Documentary
## 1      NA     NA      NA     NA      NA      NA  NA    NA           NA
## 2      NA     NA      NA     NA      NA      NA  NA    NA           NA
## 3      NA     NA      NA     NA      NA      NA  NA    NA           NA
## 4      NA     NA      NA     NA      NA      NA  NA    NA           NA
## 5      NA     NA      NA     NA      NA      NA  NA    NA           NA
## 6      NA     NA      NA     NA      NA      NA  NA    NA           NA
##   Foreign TV Movie
## 1      NA       NA
## 2      NA       NA
## 3      NA       NA
## 4      NA       NA
## 5      NA       NA
## 6      NA       NA
```

```
rm(k)
```

"1" represents that this movie belonges to the genre, and "0" represents that this movie doesn't belong to the genre.

```
movieDat[is.na(movieDat)] <- 0
head(movieDat)
```

```
##      budget
## 1 237000000
## 2 300000000
## 3 245000000
## 4 250000000
## 5 260000000
## 6 258000000
##
## 1 [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"i
## 2                                          [{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fanta
## 3                                            [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adven
## 4               [{"id": 28, "name": "Action"}, {"id": 80, "name": "Crime"}, {"id": 18, "name": "Drama
## 5                                  [{"id": 28, "name": "Action"}, {"id": 12, "name": "Adventure"}, {"i
## 6                                           [{"id": 14, "name": "Fantasy"}, {"id": 28, "name": "Action
##                                        homepage     id
## 1                     http://www.avatarmovie.com/  19995
## 2    http://disney.go.com/disneypictures/pirates/    285
## 3     http://www.sonypictures.com/movies/spectre/ 206647
## 4              http://www.thedarkknightrises.com/  49026
## 5            http://movies.disney.com/john-carter  49529
## 6 http://www.sonypictures.com/movies/spider-man3/    559
##
## 1                            [{"id": 1463, "name": "culture clash"}, {"id": 2964, "name": "future"}
```

```
## 2
## 3
## 4 [{"id": 849, "name": "dc comics"}, {"id": 853, "name": "crime fighter"}, {"id": 949, "name": "terro
## 5
## 6
##   original_language                             original_title
## 1                en                                     Avatar
## 2                en Pirates of the Caribbean: At World's End
## 3                en                                    Spectre
## 4                en                      The Dark Knight Rises
## 5                en                                John Carter
## 6                en                              Spider-Man 3
##
## 1
## 2
## 3
## 4 Following the death of District Attorney Harvey Dent, Batman assumes responsibility for Dent's crir
## 5                                                                                        John Carter i
## 6
##   popularity
## 1  150.43758
## 2  139.08262
## 3  107.37679
## 4  112.31295
## 5   43.92699
## 6  115.69981
##
## 1 [{"name": "Ingenious Film Partners", "id": 289}, {"name": "Twentieth Century Fox Film Corporation"
## 2                                                                  [{"name": "Walt Disney Pictures", "id
## 3
## 4                                                [{"name": "Legendary Pictures", "id": 923}, {"name"
## 5
## 6                                                                                [{"name": "Columbia Pictures"
##                                                                                                 production_co
## 1 [{"iso_3166_1": "US", "name": "United States of America"}, {"iso_3166_1": "GB", "name": "United Kin
## 2                                                  [{"iso_3166_1": "US", "name": "United States of Ame
## 3 [{"iso_3166_1": "GB", "name": "United Kingdom"}, {"iso_3166_1": "US", "name": "United States of Ame
## 4                                                  [{"iso_3166_1": "US", "name": "United States of Ame
## 5                                                  [{"iso_3166_1": "US", "name": "United States of Ame
## 6                                                  [{"iso_3166_1": "US", "name": "United States of Ame
##   release_date    revenue runtime
## 1   2009-12-10 2787965087     162
## 2   2007-05-19  961000000     169
## 3   2015-10-26  880674609     148
## 4   2012-07-16 1084939099     165
## 5   2012-03-07  284139100     132
## 6   2007-05-01  890871626     139
##
## 1
## 2
## 3 [{"iso_639_1": "fr", "name": "Fran\\u00e7ais"}, {"iso_639_1": "en", "name": "English"}, {"iso_639_1
## 4
## 5
## 6
```

```
##     status                                          tagline
## 1 Released                    Enter the World of Pandora.
## 2 Released At the end of the world, the adventure begins.
## 3 Released                           A Plan No One Escapes
## 4 Released                                The Legend Ends
## 5 Released          Lost in our world, found in another.
## 6 Released                             The battle within.
##                                    title vote_average vote_count Action
## 1                                 Avatar          7.2      11800      1
## 2 Pirates of the Caribbean: At World's End          6.9       4500      1
## 3                                Spectre          6.3       4466      1
## 4                    The Dark Knight Rises          7.6       9106      1
## 5                             John Carter          6.1       2124      1
## 6                            Spider-Man 3          5.9       3576      1
##   Adventure Fantasy Science Fiction Crime Drama Thriller Animation Family
## 1         1       1               1     0     0        0         0      0
## 2         1       1               0     0     0        0         0      0
## 3         1       0               0     1     0        0         0      0
## 4         0       0               0     1     1        1         0      0
## 5         1       0               1     0     0        0         0      0
## 6         1       1               0     0     0        0         0      0
##   Western Comedy Romance Horror Mystery History War Music Documentary
## 1       0      0       0      0       0       0   0     0           0
## 2       0      0       0      0       0       0   0     0           0
## 3       0      0       0      0       0       0   0     0           0
## 4       0      0       0      0       0       0   0     0           0
## 5       0      0       0      0       0       0   0     0           0
## 6       0      0       0      0       0       0   0     0           0
##   Foreign TV Movie
## 1       0        0
## 2       0        0
## 3       0        0
## 4       0        0
## 5       0        0
## 6       0        0
```

2b).

As in the previous homework, extract only the numerical features and save it in the data frame nmovieDat. However, add all columns you generated in part a) for genres to nmovieDat. Finally, create a new column called profit which is revenue minus budget. Compute this column and add it to the nmovieDat.

```
nmovieDat <- movieDat %>%
  dplyr::select_if(is.numeric) %>%  # select the numeric columns
  dplyr::select(-id)  # Drop the the column of "id"
head(nmovieDat)
```

```
##      budget popularity    revenue runtime vote_average vote_count Action
## 1 237000000  150.43758 2787965087     162          7.2      11800      1
## 2 300000000  139.08262  961000000     169          6.9       4500      1
## 3 245000000  107.37679  880674609     148          6.3       4466      1
## 4 250000000  112.31295 1084939099     165          7.6       9106      1
## 5 260000000   43.92699  284139100     132          6.1       2124      1
```

```
## 6 258000000  115.69981  890871626     139          5.9       3576      1
##   Adventure Fantasy Science Fiction Crime Drama Thriller Animation Family
## 1         1       1               1     0     0        0         0      0
## 2         1       1               0     0     0        0         0      0
## 3         1       0               0     1     0        0         0      0
## 4         0       0               0     1     1        1         0      0
## 5         1       0               1     0     0        0         0      0
## 6         1       1               0     0     0        0         0      0
##   Western Comedy Romance Horror Mystery History War Music Documentary
## 1       0      0       0      0       0       0   0     0           0
## 2       0      0       0      0       0       0   0     0           0
## 3       0      0       0      0       0       0   0     0           0
## 4       0      0       0      0       0       0   0     0           0
## 5       0      0       0      0       0       0   0     0           0
## 6       0      0       0      0       0       0   0     0           0
##   Foreign TV Movie
## 1       0        0
## 2       0        0
## 3       0        0
## 4       0        0
## 5       0        0
## 6       0        0
```

2c).

Once you create the nmovieDat data frame, divide the data into two groups of training and test sets. Choose, randomly 80% of the data and put them in a data frame called nmovieDatTrain. Put the remainder in a data frame called nmovieDatTest.

```
trainIndex <- sample(nrow(nmovieDat), 0.8*nrow(nmovieDat))
nmovieDatTrain <- nmovieDat[trainIndex,]  # Traning Set
nmovieDatTest <- nmovieDat[-trainIndex,]  # Test Set
```

2d).

Build a linear regression model called lmmodel1 relating profit to only the numerical features (except budget and revenue, of course.) What is the percentage of variation in profit explained by lmmodel1?

```
nmovieDatModel <- nmovieDat %>%
  mutate(profit = revenue - budget) %>%  # create a new feature named "profit"
  select(-budget,-revenue)

lmmodel1 <- lm(profit ~., data=nmovieDatModel)
summary(lmmodel1)
```

```
##
## Call:
## lm(formula = profit ~ ., data = nmovieDatModel)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
```

```
## -702159078  -27007395   -1182767   19043529 1597999344
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -17608065    8940469  -1.969 0.048956 *
## popularity           265761      62276   4.267 2.02e-05 ***
## runtime              410315      66597   6.161 7.81e-10 ***
## vote_average       -4588417    1200387  -3.822 0.000134 ***
## vote_count            75724       1671  45.308  < 2e-16 ***
## Action              -529973    3532615  -0.150 0.880753
## Adventure          24294607    3975635   6.111 1.07e-09 ***
## Fantasy             5997821    4654201   1.289 0.197568
## `Science Fiction` -26146984    4272880  -6.119 1.02e-09 ***
## Crime             -15266528    3887732  -3.927 8.73e-05 ***
## Drama             -13121643    3136719  -4.183 2.93e-05 ***
## Thriller           -2926998    3454955  -0.847 0.396933
## Animation          38268269    7015228   5.455 5.14e-08 ***
## Family             19893364    5029848   3.955 7.76e-05 ***
## Western           -34709198    9651516  -3.596 0.000326 ***
## Comedy               627741    3112081   0.202 0.840151
## Romance             9799635    3445106   2.845 0.004467 **
## Horror              -604688    4579361  -0.132 0.894953
## Mystery            -5253615    5044374  -1.041 0.297705
## History            -8769630    6878790  -1.275 0.202414
## War               -11102734    7784380  -1.426 0.153852
## Music               6100603    6530978   0.934 0.350298
## Documentary         8072010    8852078   0.912 0.361879
## Foreign             6186673   14780065   0.419 0.675540
## `TV Movie`        -16652178   30312306  -0.549 0.582788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85460000 on 4778 degrees of freedom
## Multiple R-squared:  0.6069, Adjusted R-squared:  0.6049
## F-statistic: 307.4 on 24 and 4778 DF,  p-value: < 2.2e-16
```

Conclusion: R-square equals 0.6069. That means 60.49 percent of variance in profit explained by lmmodel1.

2e).

Now build a linear regression model called lmmodel2 relating profit to all features of nmovieDatTrain. What percentage of the variation in profit is described by lmmodel2?

```
nmovieDatTrain <- nmovieDatTrain %>%
  mutate(profit = revenue - budget) %>%  # create a new feature named "profit"
  select(-budget,-revenue)

lmmodel2 <- lm(profit ~., data=nmovieDatTrain)
summary(lmmodel2)
```

```
##
## Call:
```

```
## lm(formula = profit ~ ., data = nmovieDatTrain)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -638465473  -27886992   -1474261   19614042 1584678859
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -16466577   10277540  -1.602 0.109196
## popularity           324626      67268   4.826 1.45e-06 ***
## runtime              436591      76571   5.702 1.28e-08 ***
## vote_average       -4960307    1372038  -3.615 0.000304 ***
## vote_count            76395       1859  41.089  < 2e-16 ***
## Action             -3184325    4060165  -0.784 0.432922
## Adventure          25502050    4565928   5.585 2.50e-08 ***
## Fantasy             -393714    5265040  -0.075 0.940395
## `Science Fiction` -24480880    4844451  -5.053 4.54e-07 ***
## Crime             -17498812    4398119  -3.979 7.06e-05 ***
## Drama             -15273240    3606071  -4.235 2.34e-05 ***
## Thriller           -1829813    3947692  -0.464 0.643022
## Animation          34406434    8043607   4.277 1.94e-05 ***
## Family             20063882    5732463   3.500 0.000470 ***
## Western           -39588692   11203200  -3.534 0.000415 ***
## Comedy              -669043    3556240  -0.188 0.850783
## Romance            10176318    3915401   2.599 0.009384 **
## Horror             -1230748    5215055  -0.236 0.813446
## Mystery            -9249409    5774169  -1.602 0.109269
## History            -8972554    7724573  -1.162 0.245487
## War               -10588105    8409419  -1.259 0.208080
## Music               5239940    7292434   0.719 0.472465
## Documentary         8407851   10031998   0.838 0.402025
## Foreign             6887796   16597335   0.415 0.678170
## `TV Movie`        -25738676   35653350  -0.722 0.470391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86990000 on 3817 degrees of freedom
## Multiple R-squared:  0.6049, Adjusted R-squared:  0.6024
## F-statistic: 243.4 on 24 and 3817 DF,  p-value: < 2.2e-16
```

Conclusion: R-square equals 0.6049. That means 60.49 percent of variance in profit described by lmmodel2.

3.

For this assignment we are going to test the binary classifications using SVM (for various kernels) and the logistic regression.

3a).

Create a new feature called incomeGenre. The idea is we are going to lump together genres that tend to generate more revenue into one group and the rest into another. Under incomeGenre column put a “1” if the

movie belongs to two or more of genres from the set: Action, Adventure, Fantasy, Science Fiction, Crime, Drama, Thriller, Horror. Make sure to make this new feature a categorical variable.

```
# Then I choose the specified columns, and create a temp dataframe to store these columns.
genreDf <- nmovieDat %>%
  select(Action, Adventure, Fantasy, 'Science Fiction', Crime, Drama, Thriller, Horror)

# Create the new column 'incomeGenre' with values
for (i in seq(dim(genreDf)[1])) {
  if (rowSums(genreDf)[i] >= 2) {
    nmovieDat[i,'incomeGenre'] = 1
  } else {
    nmovieDat[i,'incomeGenre'] = 0
  }
}
df <- nmovieDat %>%
  select(budget,popularity,revenue,runtime,vote_average,vote_count,incomeGenre)
```

3b).

Run the logistic regression modeling incomeGenre as a function of all numerical features (six in total). Use only the training data for this purpose, that is use only the same rows in the nmovieDat data frame. Name this model logitModel1. Print a summary of the model.

```
# Splict the training data and test data
trainIndex <- sample(nrow(df), 0.8*nrow(df))
nmovieDatTrain <- df[trainIndex,]  # Traning Set
nmovieDatTest <- df[-trainIndex,]  # Test Set
```

```
logitModel1 <- glm(incomeGenre ~., data = nmovieDatTrain, family=binomial(link='logit'))
summary(logitModel1)
```

```
##
## Call:
## glm(formula = incomeGenre ~ ., family = binomial(link = "logit"),
##     data = nmovieDatTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7845  -1.0206  -0.8671   1.2181   1.7534
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.593e-01  2.044e-01  -1.268  0.20466
## budget        1.199e-08  1.417e-09   8.466  < 2e-16 ***
## popularity    1.433e-02  3.315e-03   4.323 1.54e-05 ***
## revenue      -2.707e-09  4.266e-10  -6.346 2.22e-10 ***
## runtime       3.231e-03  1.656e-03   1.951  0.05108 .
## vote_average -1.281e-01  3.168e-02  -4.045 5.24e-05 ***
## vote_count    2.277e-04  7.691e-05   2.961  0.00307 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5308.0  on 3841  degrees of freedom
## Residual deviance: 4990.2  on 3835  degrees of freedom
## AIC: 5004.2
##
## Number of Fisher Scoring iterations: 4
```

3c).

Now predict this model on the test set, that is the data in nmovieDatTest. Compute and print the confusion table and classification error rate.

```
logitModel1_pred <- predict(logitModel1,nmovieDatTest)

# Confusion Matrix
confusionTable <- table(ifelse(logitModel1_pred > 0,1,0),nmovieDatTest$incomeGenre)
confusionTable
```

```
##
##       0   1
##   0 416 238
##   1  93 214
```

```
# Classification Error Rate
cat('The ERROR RATE is: ')
```

```
## The ERROR RATE is:
```

```
1 - sum(diag(confusionTable))/sum(confusionTable)
```

```
## [1] 0.3444329
```

3d).

Compute the BIC value of this model. In R you may use the AIC() function. You must supply the model and k = ln(N) where N is the number of data points.

```
# BIC
cat("BIC Index: ")
```

```
## BIC Index:
```

```
BIC(logitModel1)
```

```
## [1] 5047.951
```

```
# AIC
cat("AIC Index: ")
```

```
## AIC Index:
```

```
AIC(logitModel1, k=log(dim(nmovieDatTrain)[1]))
```

```
## [1] 5047.951
```

3e).

Repeat parts 3b-3d, but this time use cross product B-Splines with df=6. Based on the BIC value, does the cross product model improve over the linear model?

```
library(splines)
# B-Splines with df = 6
nlogitModel <- glm(incomeGenre ~ bs(budget,df=6)+bs(popularity,df=6)+bs(revenue,df=6)+bs(runtime,df=6)+
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(nlogitModel)
```

```
##
## Call:
## glm(formula = incomeGenre ~ bs(budget, df = 6) + bs(popularity,
##     df = 6) + bs(revenue, df = 6) + bs(runtime, df = 6) + bs(vote_average,
##     df = 6) + bs(vote_count, df = 6), family = binomial, data = nmovieDatTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1938  -1.0348  -0.6719   1.1204   2.2411
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -4.69696    3.88276  -1.210 0.226396
## bs(budget, df = 6)1         0.22040    0.19151   1.151 0.249805
## bs(budget, df = 6)2         0.04900    0.18533   0.264 0.791492
## bs(budget, df = 6)3         0.07385    0.17160   0.430 0.666927
## bs(budget, df = 6)4         2.16354    0.73127   2.959 0.003090 **
## bs(budget, df = 6)5        -1.78779    2.84693  -0.628 0.530023
## bs(budget, df = 6)6        13.87819    9.66392   1.436 0.150979
## bs(popularity, df = 6)1     0.46191    0.42151   1.096 0.273148
## bs(popularity, df = 6)2    -0.04263    0.37917  -0.112 0.910484
## bs(popularity, df = 6)3    -0.08198    0.43358  -0.189 0.850026
## bs(popularity, df = 6)4     4.37806    1.82242   2.402 0.016291 *
## bs(popularity, df = 6)5   -15.07506   11.28749  -1.336 0.181695
## bs(popularity, df = 6)6    68.57258   63.13413   1.086 0.277417
## bs(revenue, df = 6)1        3.29078    3.84921   0.855 0.392594
## bs(revenue, df = 6)2        3.12792    3.84857   0.813 0.416361
## bs(revenue, df = 6)3        2.73295    3.85143   0.710 0.477956
```

```
## bs(revenue, df = 6)4       -2.14337    3.87424  -0.553 0.580101
## bs(revenue, df = 6)5        5.71787    5.27699   1.084 0.278565
## bs(revenue, df = 6)6             NA         NA      NA       NA
## bs(runtime, df = 6)1       -0.60171    2.05415  -0.293 0.769582
## bs(runtime, df = 6)2       -0.28980    0.57595  -0.503 0.614852
## bs(runtime, df = 6)3        0.29531    0.48289   0.612 0.540843
## bs(runtime, df = 6)4        1.57844    0.69849   2.260 0.023836 *
## bs(runtime, df = 6)5       -2.64555    1.81113  -1.461 0.144094
## bs(runtime, df = 6)6        3.19179    3.11442   1.025 0.305438
## bs(vote_average, df = 6)1  -0.38310    0.94363  -0.406 0.684753
## bs(vote_average, df = 6)2   0.82935    0.50748   1.634 0.102208
## bs(vote_average, df = 6)3  -0.28192    0.40142  -0.702 0.482489
## bs(vote_average, df = 6)4  -1.09049    0.48980  -2.226 0.025987 *
## bs(vote_average, df = 6)5  -2.96475    1.00005  -2.965 0.003031 **
## bs(vote_average, df = 6)6  -0.07333    1.66682  -0.044 0.964910
## bs(vote_count, df = 6)1     0.28998    0.41339   0.701 0.483003
## bs(vote_count, df = 6)2     1.43288    0.38810   3.692 0.000222 ***
## bs(vote_count, df = 6)3     1.62664    0.44546   3.652 0.000261 ***
## bs(vote_count, df = 6)4     4.02873    1.01075   3.986 6.72e-05 ***
## bs(vote_count, df = 6)5     5.09865    1.85848   2.743 0.006080 **
## bs(vote_count, df = 6)6     3.91857    2.18775   1.791 0.073271 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5308.0  on 3841  degrees of freedom
## Residual deviance: 4831.4  on 3806  degrees of freedom
## AIC: 4903.4
##
## Number of Fisher Scoring iterations: 9
```

```
nlogitModel_pred <- predict(nlogitModel,nmovieDatTest)
```

```
## Warning in bs(popularity, degree = 3L, knots = c(`25%` = 4.5751015, `50%`
## = 12.8351815, : some 'x' values beyond boundary knots may cause ill-
## conditioned bases
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
# Confusion Matrix
confusionTable <- table(ifelse(nlogitModel_pred > 0,1,0),nmovieDatTest$incomeGenre)
confusionTable
```

```
##
##       0   1
##   0 365 180
##   1 144 272
```

```
# Classification Error Rate
cat('The ERROR RATE is: ')
```

```
## The ERROR RATE is:
```

```
1 - sum(diag(confusionTable))/sum(confusionTable)
```

```
## [1] 0.3371488
```

```
# BIC
cat("BIC Index: ")
```

```
## BIC Index:
```

```
BIC(nlogitModel)
```

```
## [1] 5128.516
```

Conclusion: No, it doesn't. The BIC becomes larger than former one. Some former one is better.

3f).

Repeat parts 3b-3c, but this time use an SVM model with polynomial kernel and degree 4.

```
library(e1071)
svmModel <- svm(nmovieDatTrain$incomeGenre ~., data=nmovieDatTrain, kernel='polynomial',
                degree=4)
summary(svmModel)
```

```
##
## Call:
## svm(formula = nmovieDatTrain$incomeGenre ~ ., data = nmovieDatTrain,
##     kernel = "polynomial", degree = 4)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  4
##       gamma:  0.1666667
##      coef.0:  0
##     epsilon:  0.1
##
##
## Number of Support Vectors:  3529
```

```
svmModel_pred <- predict(svmModel,nmovieDatTest)

# Confusion Matrix
confusionTable <- table(ifelse(svmModel_pred > 0,1,0),nmovieDatTest$incomeGenre)
confusionTable
```

```
##
##       0   1
##   0   7   4
##   1 502 448
```

```
# Classification Error Rate
cat('The ERROR RATE is: ')
```

```
## The ERROR RATE is:
```

```
1 - sum(diag(confusionTable))/sum(confusionTable)
```

```
## [1] 0.5265349
```

3g).

Repeat parts 3b-3c, but this time use an SVM model with radial basis kernel.

```
svmModel1 <- svm(nmovieDatTrain$incomeGenre ~., data=nmovieDatTrain, kernel='radial')
summary(svmModel)
```

```
##
## Call:
## svm(formula = nmovieDatTrain$incomeGenre ~ ., data = nmovieDatTrain,
##     kernel = "polynomial", degree = 4)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  polynomial
##        cost:  1
##      degree:  4
##       gamma:  0.1666667
##      coef.0:  0
##     epsilon:  0.1
##
##
## Number of Support Vectors:  3529
```

```
svmModel_pred1 <- predict(svmModel1,nmovieDatTest)

# Confusion Matrix
confusionTable <- table(ifelse(svmModel_pred1 > 0,1,0),nmovieDatTest$incomeGenre)
confusionTable
```

```
##
##       0   1
##   0  16   9
##   1 493 443
```

```r
# Classification Error Rate
cat('The ERROR RATE is: ')
```

```
## The ERROR RATE is:
```

```r
1 - sum(diag(confusionTable))/sum(confusionTable)
```

```
## [1] 0.5223725
```