IST 707 Project

Data Degenerates Daniel Caley Michael Johnson

9/4/2021

Contents

Introduction

Recommender systems provide the backbone of systems designed to both enhance user workflow and to increase viewers for the overall platform. Using the Netflix data, successful models relying on various techniques help in building out a recommendation network that includes predicting the IMDb score and parental ratings for suggested viewing. When used in conjunction with Association Rule Mining techniques for genre, the recommender system will have robust capabilities that include a variety of models for people viewing everything from G to R ratings.

Four primary techniques make up the base for conducting analysis, exploring the data set, and building a predictive model that centers on the IMDb score and a parental rating classifier. Clustering and Classification on the description illuminate the parental rating and serve as a basis for comparing different descriptions. Association Rule Mining will help suggest what genre of movies or TV shows a viewer should watch next based on their most recently viewed. Identifying predictive solutions using Movie Length, Genre, Parental Rating, and TV Show or Movie as part of a Support Vector Machine (SVM), KNN, Random Forest, or Decision Tree model.

About the Data

The Netflix data set came from Satpreet Makhija (2021) on Kaggle, and it contains all the movies and TV shows from Netflix in 2001. The overall data is contained within the following columns: Description, Director, Genre, Cast, Rating, Duration and IMDB Score. Discretization of the parental ratings included combining ratings that aligned with certain age groups, like TV-MA and R. Although these have varying definitions for who should and should not be watching, the suggested age groups were close enough to combine together with a reasonable assumption that parents would not balk at the differences between G versus Y-7 or PG-13 versus TV-14.

Cleaning Data

Filtering the original data included removing some columns and limiting the data based on location and type. As a United States-focused recommender system, the data includes all production studios that shot scenes specifically within the United States at some point in the process. Additionally, the data contains no blank values wherever data was missing for director and cast records. Having complete data for those variables could be an important part of future modifications to the recommender system that rely on using specific directors or cast members. Additionally, the data only contains movies as TV shows have varying data for the cast, directors, and other key variables that went into predicting IMDb score like duration.

Discretization and cleaning of the data included the following focus areas:

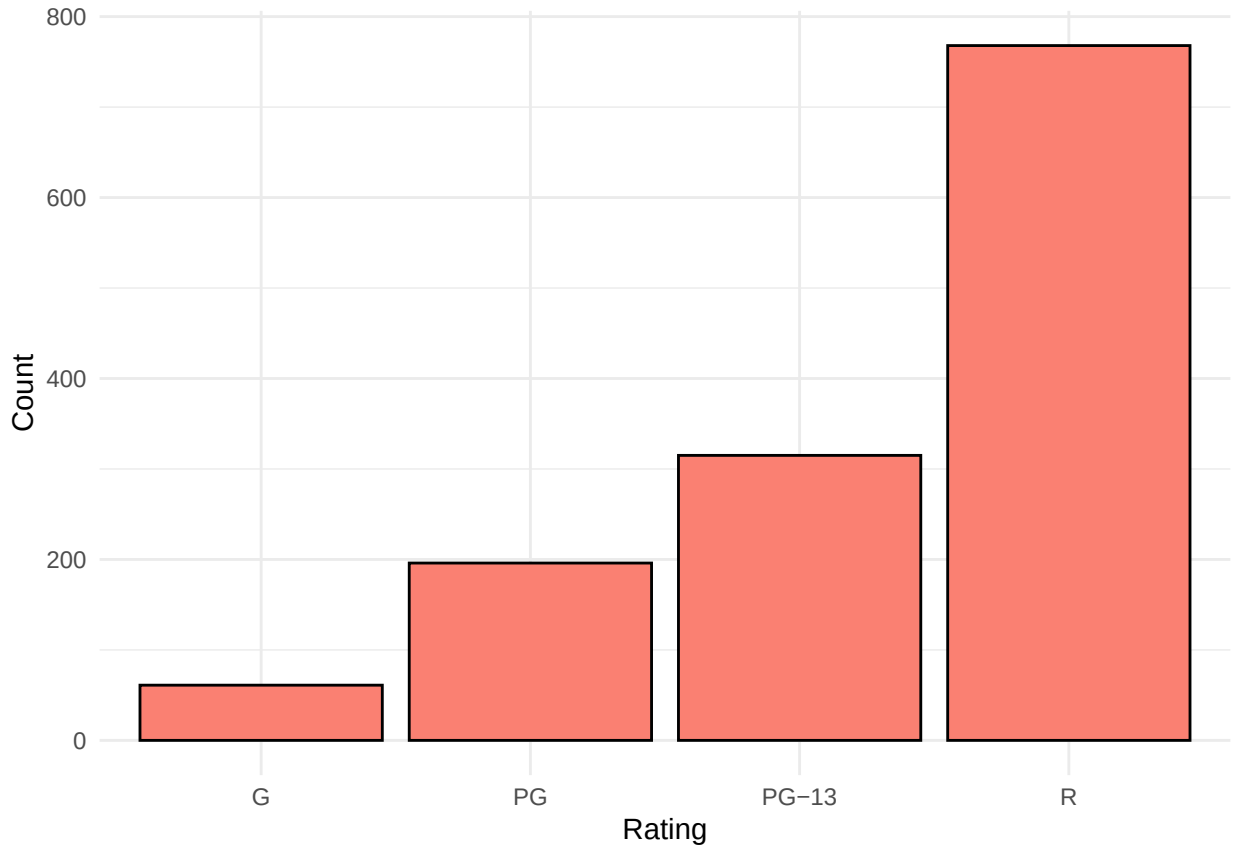
- Removing "/10" from the IMDB score column.
- Removing "min" from the duration column.
- Removing the Date Added column.
- Putting the length of movie in 30 minute bins.
- Creating an International flag field for production country that sets a value of 0 for movies that only include the United states.
- Cleaning up the rating by combining the TV ratings and movie ratings in the following format:
  - TV-Y transformed to G.
  - TV-Y7 transformed to PG.
  - TV-14 transformed to PG13.
  - TV-MA transformed to R.

The following output is the structure of the data, including variable types and field names:

```
str(CleanFlix)
```

```
## tibble [1,340 x 14] (S3: tbl_df/tbl/data.frame)
##  $ Show Id           : chr [1:1340] "c844460f-6178-4f87-929e-80816c74ca35" "0e5fc89e-be6a-44d1-9923-5
##  $ Title             : chr [1:1340] "#realityhigh" "1922" "1BR" "2 Hearts" ...
##  $ Description       : chr [1:1340] "When nerdy high schooler Dani finally attracts the interest of h
##  $ Director          : chr [1:1340] "Fernando Lebrija" "Zak Hilditch" "David Marmor" "Lance Hool" ..
##  $ Genres            : chr [1:1340] "Comedies" "Dramas, Thrillers" "Horror Movies, Independent Movies
##  $ Cast              : chr [1:1340] "Nesta Cooper, Kate Walsh, John Michael Higgins, Keith Powers, Al
##  $ Production Country: chr [1:1340] "United States" "United States" "United States" "United States"
##  $ Release Date      : num [1:1340] 2017 2017 2019 2020 2009 ...
##  $ Rating            : chr [1:1340] "PG-13" "R" "R" "PG-13" ...
##  $ Duration          : num [1:1340] 99 103 90 101 158 144 117 92 91 92 ...
##  $ Imdb Score        : num [1:1340] 5.1 6.4 5.7 5.9 6 6.6 6.4 5.8 4.7 6.1 ...
##  $ Content Type      : chr [1:1340] "Movie" "Movie" "Movie" "Movie" ...
##  $ Duration_bins     : Ord.factor w/ 7 levels "30"<"60"<"90"<..: 4 4 3 4 6 5 4 4 4 4 ...
##  $ internation_flag  : logi [1:1340] FALSE FALSE FALSE FALSE FALSE TRUE ...
```

Exploratory Data Analysis



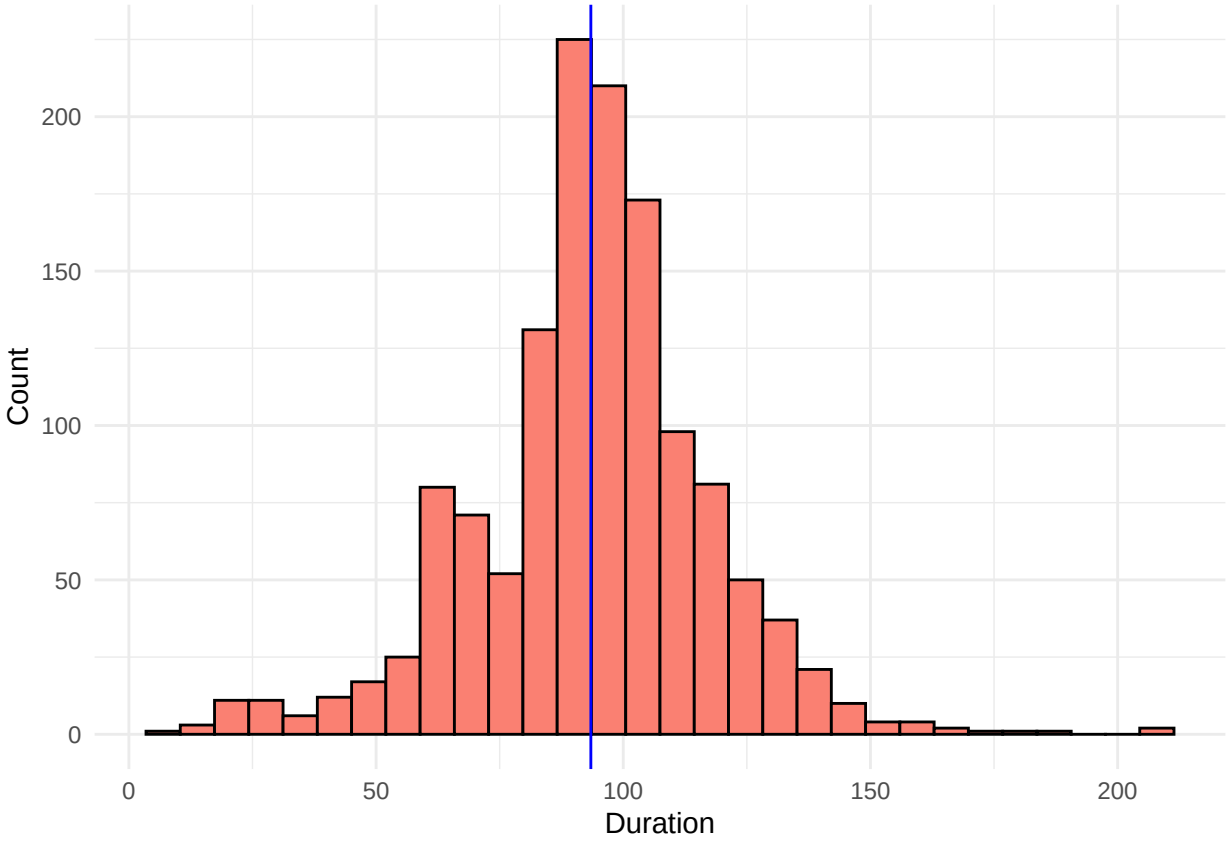

```
## [1] "Total Number of R Movies: 768"
```

```
## [1] "Total Number of PG-13 Movies: 315"
```

```
## [1] "Total Number of PG Movies: 196"
```

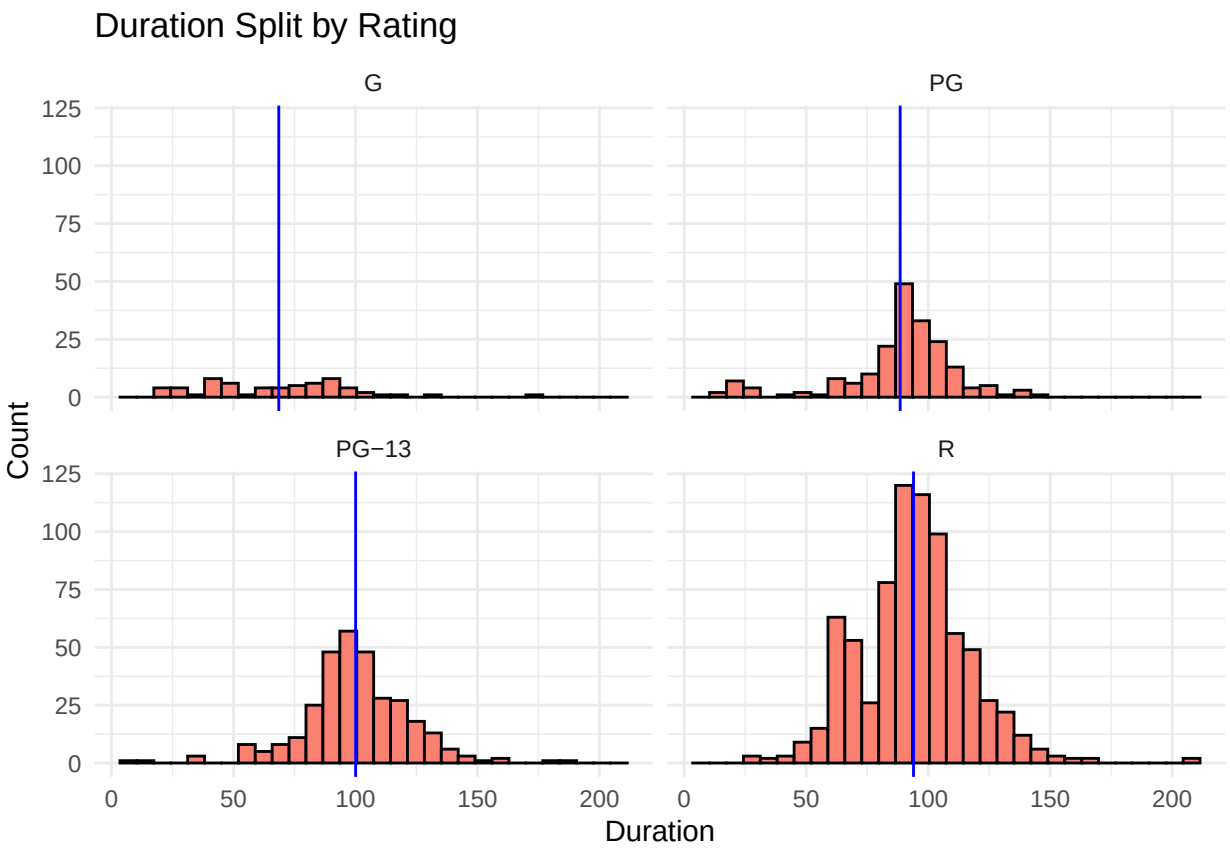
```
## [1] "Total Number of G Movies: 61"
```

Looking at the count of movies with each rating shows a disparaging difference. Movies that fall within the "R" category represent over 57 percent of all movies and could skew results in favor of that rating. PG-13 movies make up just over 23 percent of the total with PG lagging behind at just over 14 percent and G at just over 4 percent of the total movies.




```
## [1] "Average Duration in Minutes: 93.4440298507463"
```

The duration of movies has an interesting shape close to a normal distribution. The average time of movies just over 90 minutes shows a goal for most lengths at about the hour and a half mark with some outliers that run over 200 minutes in length.



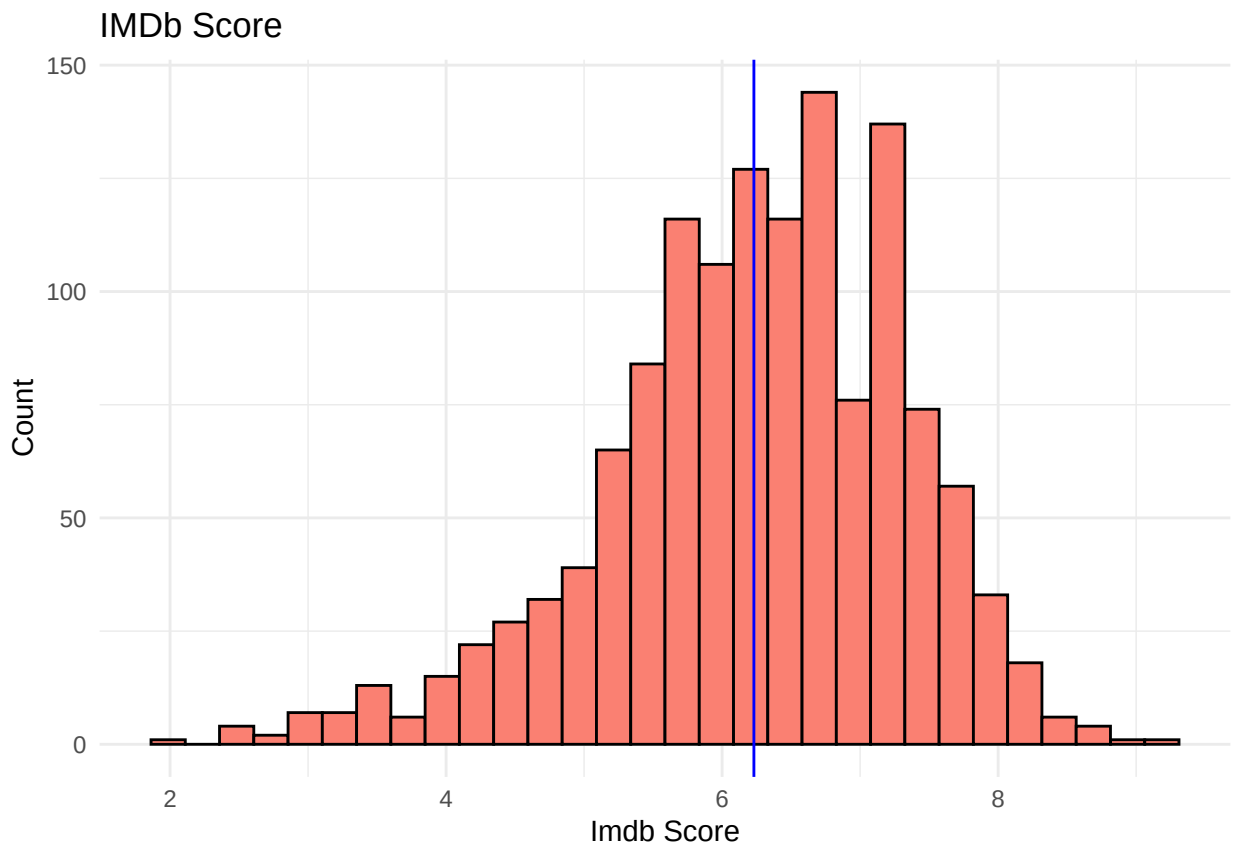

```
## [1] "Mean Duration for R Movies: 93.97265625"
```

```
## [1] "Mean Duration for PG-13 Movies: 100.047619047619"
```

```
## [1] "Mean Duration for PG Movies: 88.5051020408163"
```

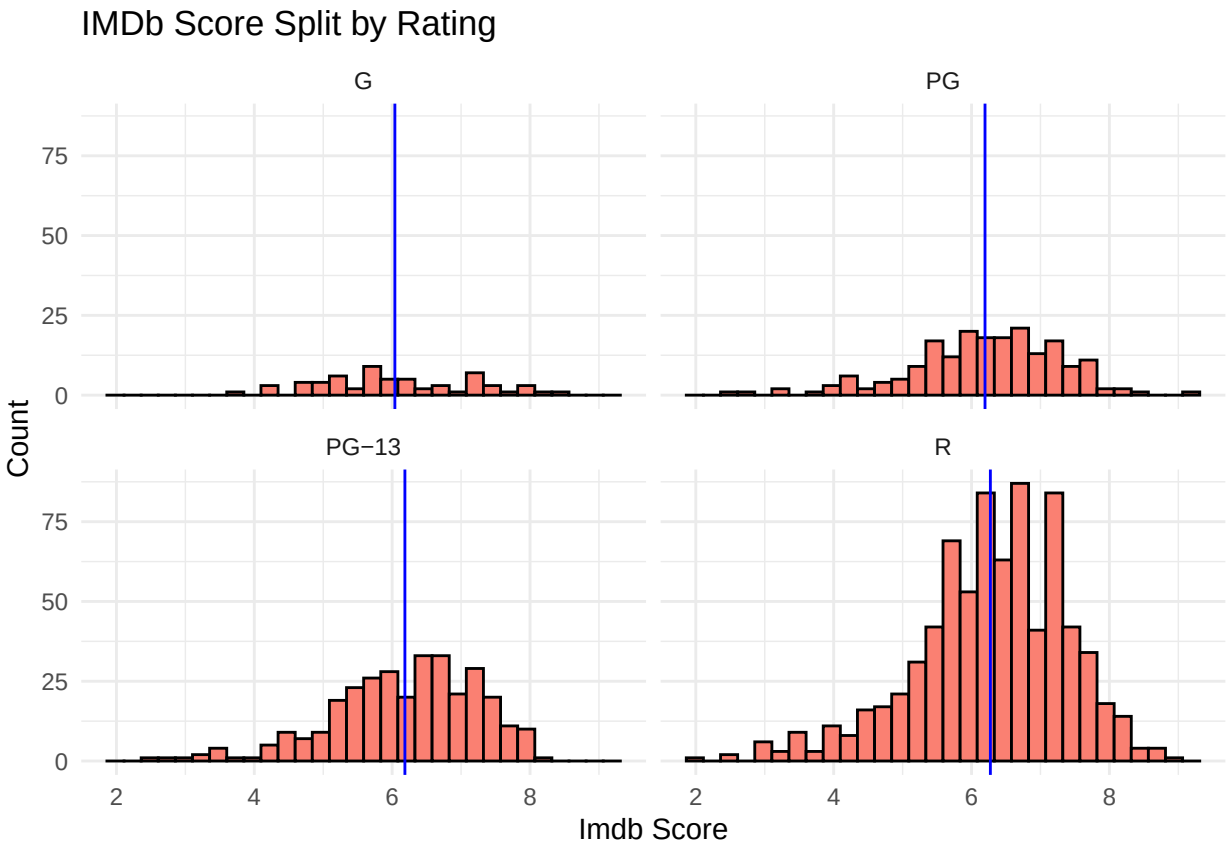
```
## [1] "Mean Duration for G Movies: 68.5573770491803"
```

Taking a closer look at the duration by parental rating indicates PG-13 movies have the longest run time, followed closely by R and PG. Movies with a G parental rating tend to be on the shorter side of the overall count, possibly due to the attention span of the audience.




```
## [1] "Average IMDb Score: 6.23044776119403"
```

The distribution of IMDb Scores has a negative skew with movies having more favorable scores with outliers approaching a score of 2 and 9 with a scale of 0 to 10. Based on the average IMDb around 6.2, the scale of IMDb scores indicates an imbalance where an expectation for movie averages would fall around the 5 mark.




```
## [1] "Mean IMDb Score for R Movies: 6.27330729166667"
```

```
## [1] "Mean IMDb Score for PG-13 Movies: 6.1847619047619"
```

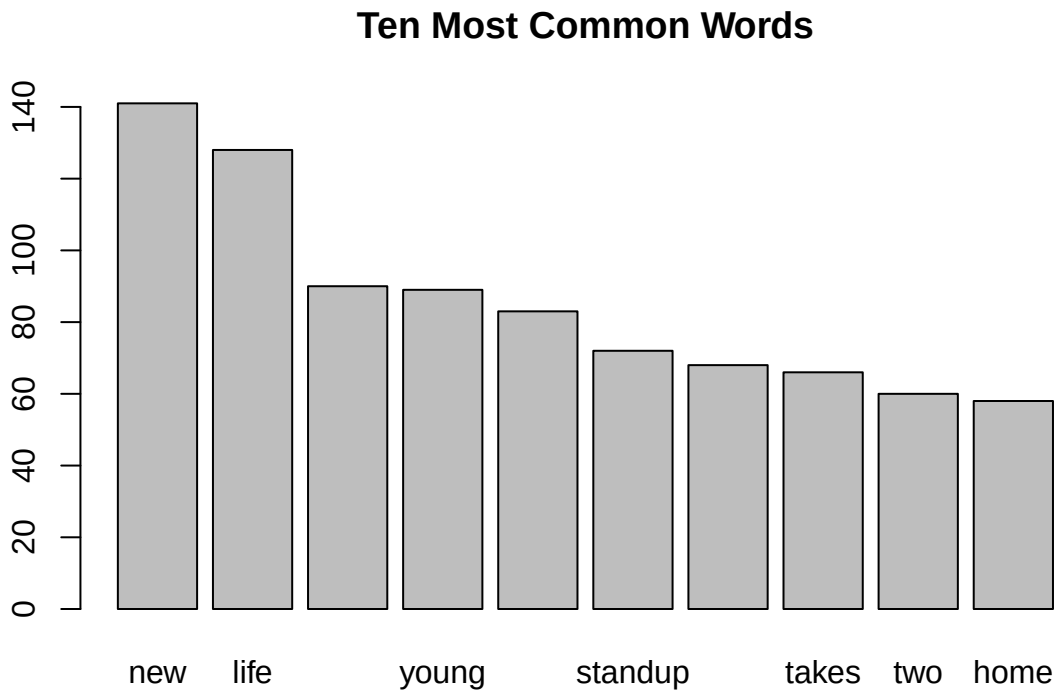
```
## [1] "Mean IMDb Score for PG Movies: 6.19540816326531"
```

```
## [1] "Mean IMDb Score for G Movies: 6.03934426229508"
```

The first indication of possibly favoring R movies due to the sheer number within the data shows when comparing the IMDb scores between ratings. R movies average an IMDb score of 6.27, the highest out of the other movie ratings.

Alongside with looking at the base counts, the description of each of the movies could help with identifying the parental rating for future movies entering the recommender system. The first step includes pulling out the description for each movie and vectorizing the words with normalizing the frequencies between each movie.

Clustering




```
## [1] "Avergae Number of Words Per Description: 14"
```

After removing all common words, stopwords, punctuation, numbers, and whitespace, the average number of words for each description comes out to 14 total words. This might limit the success of using clustering to identify trends in the data that could help with sorting movies between various categories. The goal is correctly identifying parental rating, but clustering will also show if there is anything within the data that could sort the movies. The ten most common words highlight the sentiment for overall movies on Netflix with "new" taking the top spot, followed closely by "life".

The next step in the clustering process identifies the best number of clusters to use based on measured values of cohesion within clusters and separation between clusters. Looking for values with higher separation and lower cohesion will highlight the best number of clusters for the data if any sorted groups form. Looking at clusters between 2 and 50 should allow for the best distribution of clusters without worrying about having too many clusters for the data.

The following is the distribution of cohesion and separation for the top five clusters:

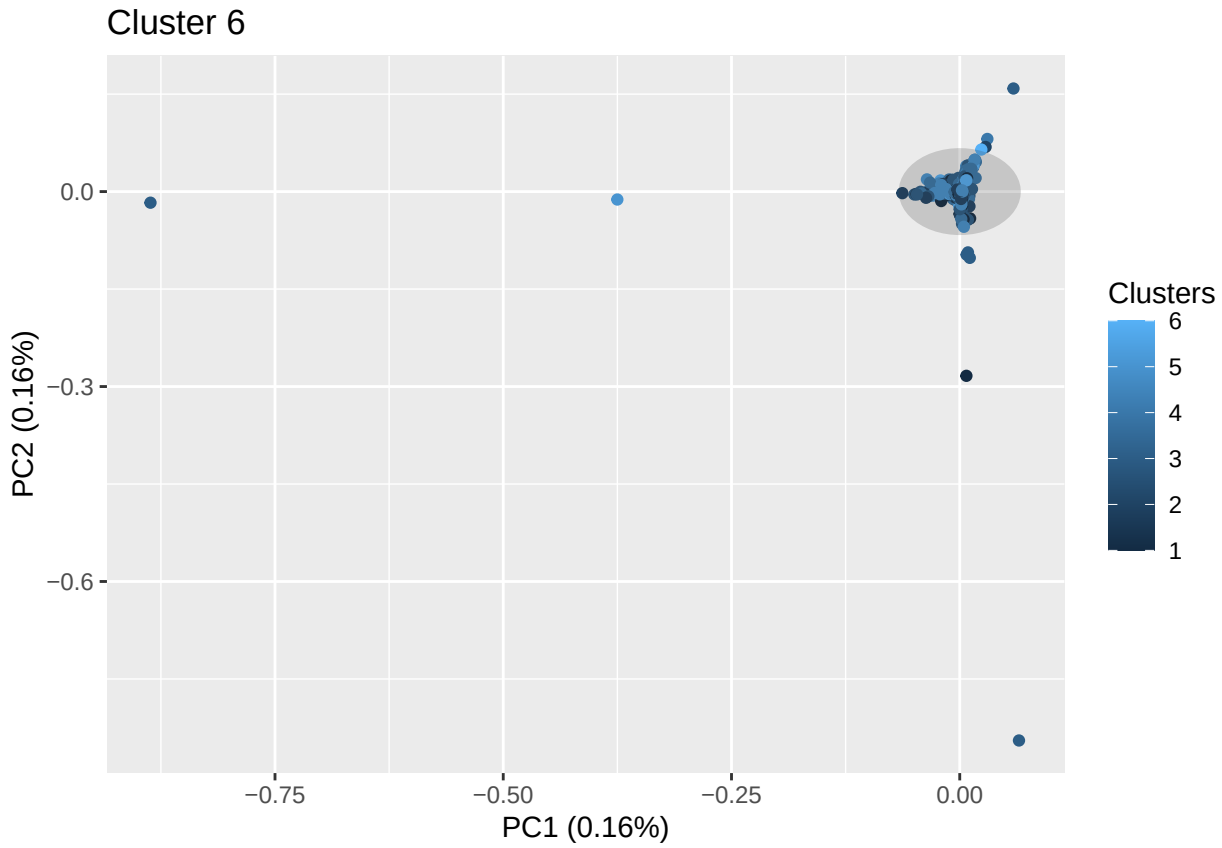
Best Cluster Results

```
##   Number     Cohesion   Separation ScaleCohesion ScaleSeparation     Combined
## 5      6 0.0003910025 7.190776e-06      1.753012       -1.753012 1.110223e-15
## 1      2 0.0003949957 3.197654e-06      2.122208       -2.122208 8.881784e-16
## 2      3 0.0003951626 3.030673e-06      2.137647       -2.137647 8.881784e-16
```

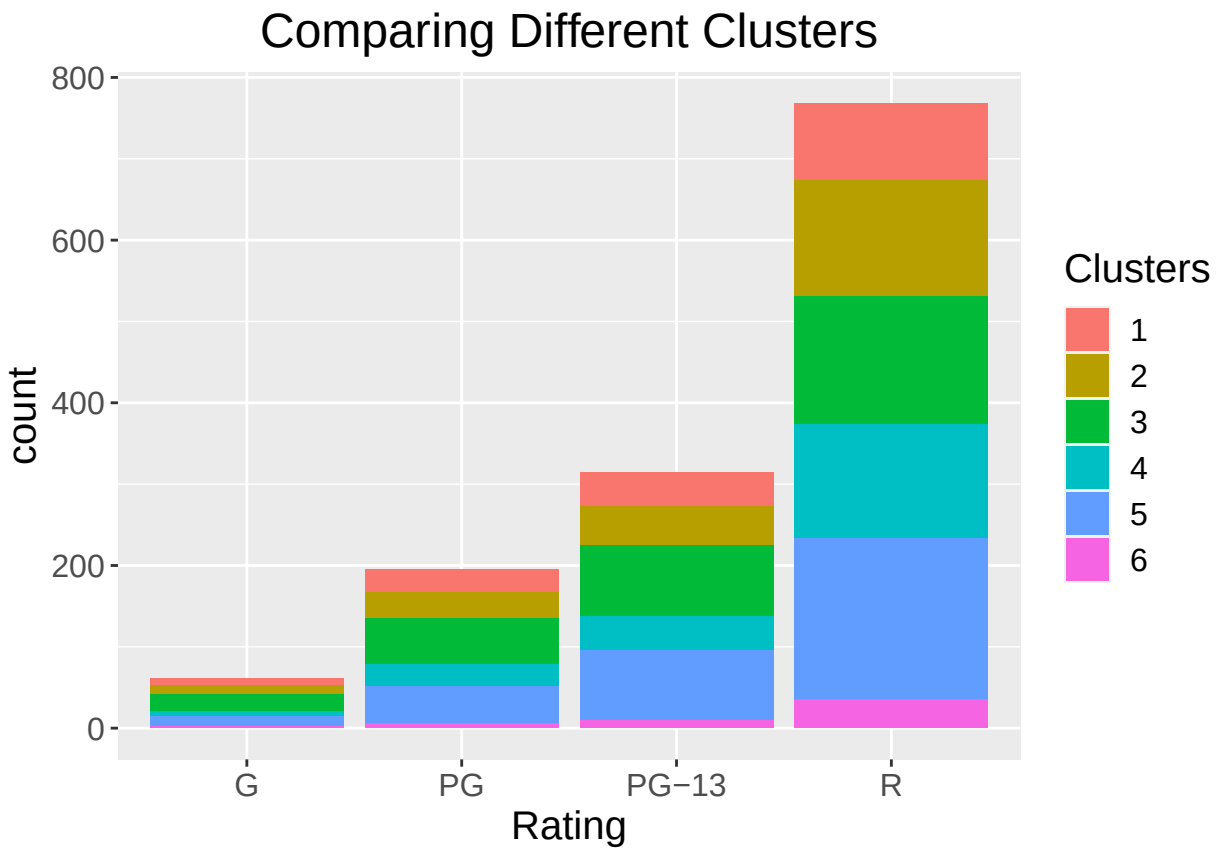
```
## 3      4 0.0003917749 6.418429e-06      1.824422      -1.824422 8.881784e-16
## 4      5 0.0003899077 8.285574e-06      1.651789      -1.651789 8.881784e-16
```

The best number of clusters after scaling the cohesion and separation for each of the clusters tested and adding them together came out to 6 clusters.

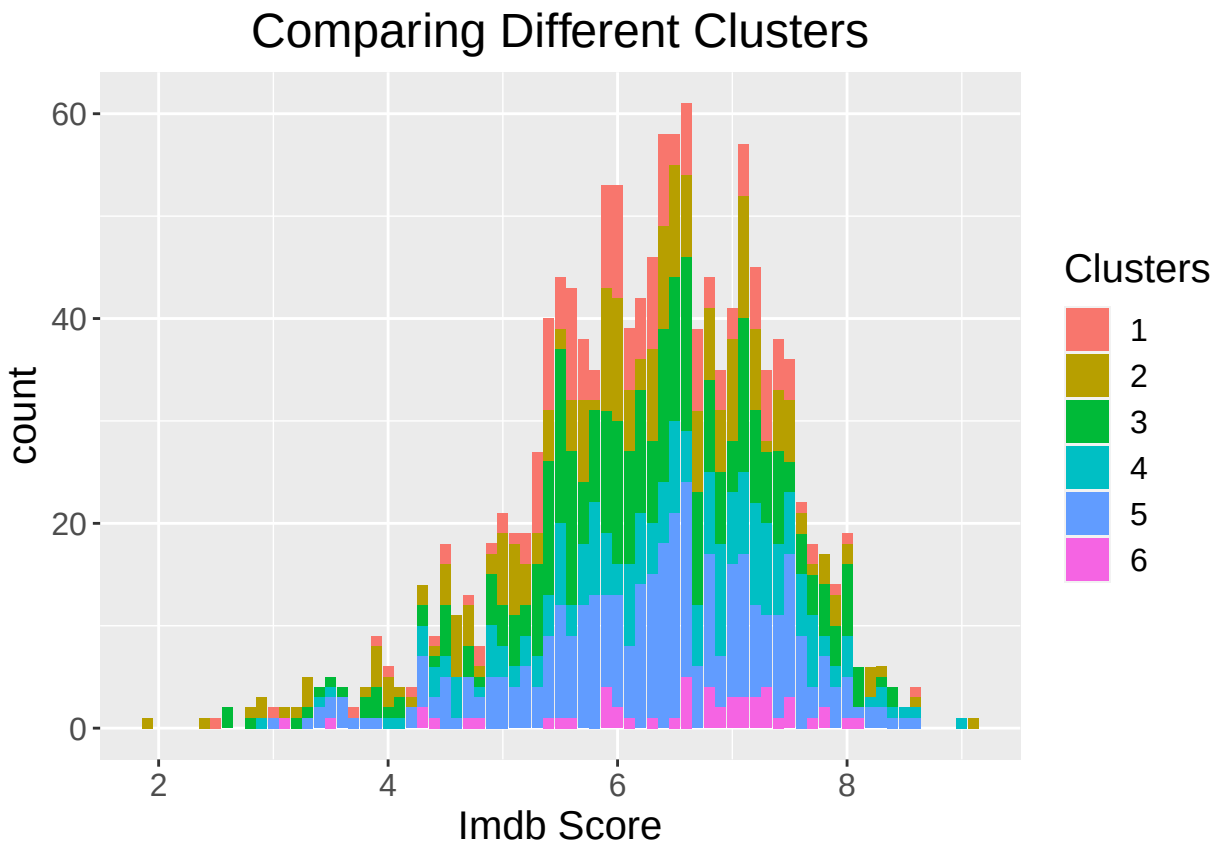
The next step is running through the data with 6 clusters using the kmeans function on the normalized word frequencies.



The graph of 6 clusters ends up looking pretty similar to the rest of the cluster amounts, even though it was the best in terms of cohesion and separation. The low frequency of words in the description with an average of 14 unique words per description highlights the limitations of relying on clustering to group this data.



Looking at the bar chart next specifically for the goal of grouping parental rating shows a spread of parental values for all ratings. Ideally, the clustering would show each cluster falling within a specific rating.



As a point of interest, clustering also shows difficulty in identifying movies with all IMDb scores. Although the results do not indicate clustering as a viable method, forcing the goal of finding parental rating from description with classification may be possible.

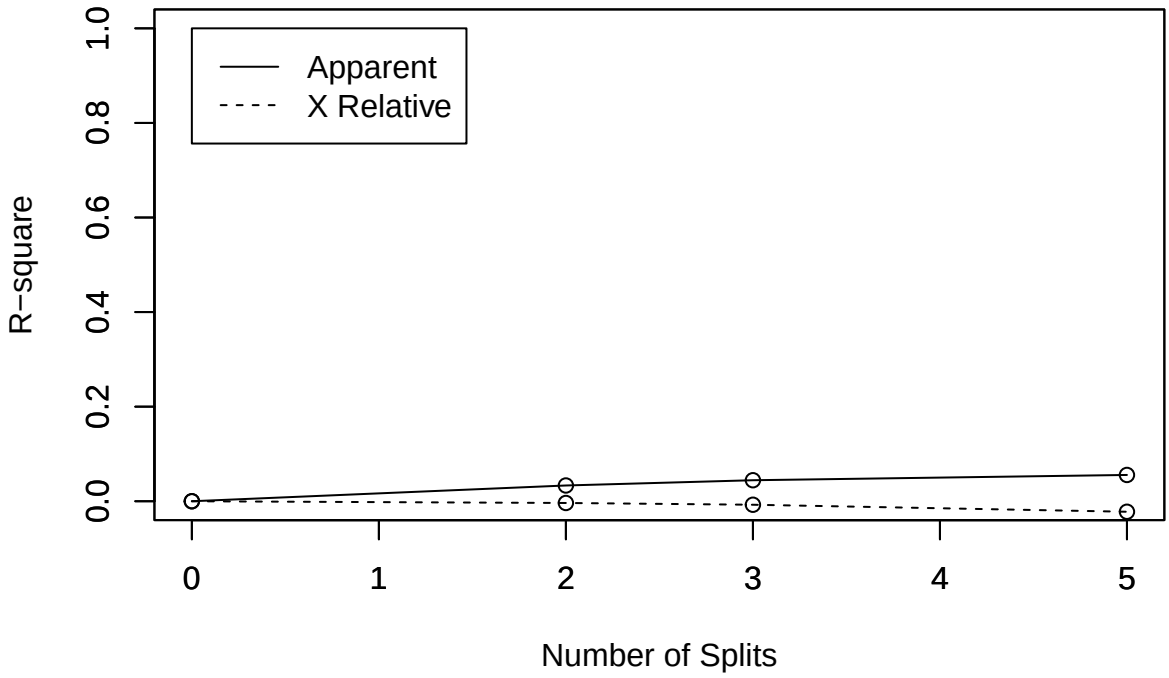
Classification

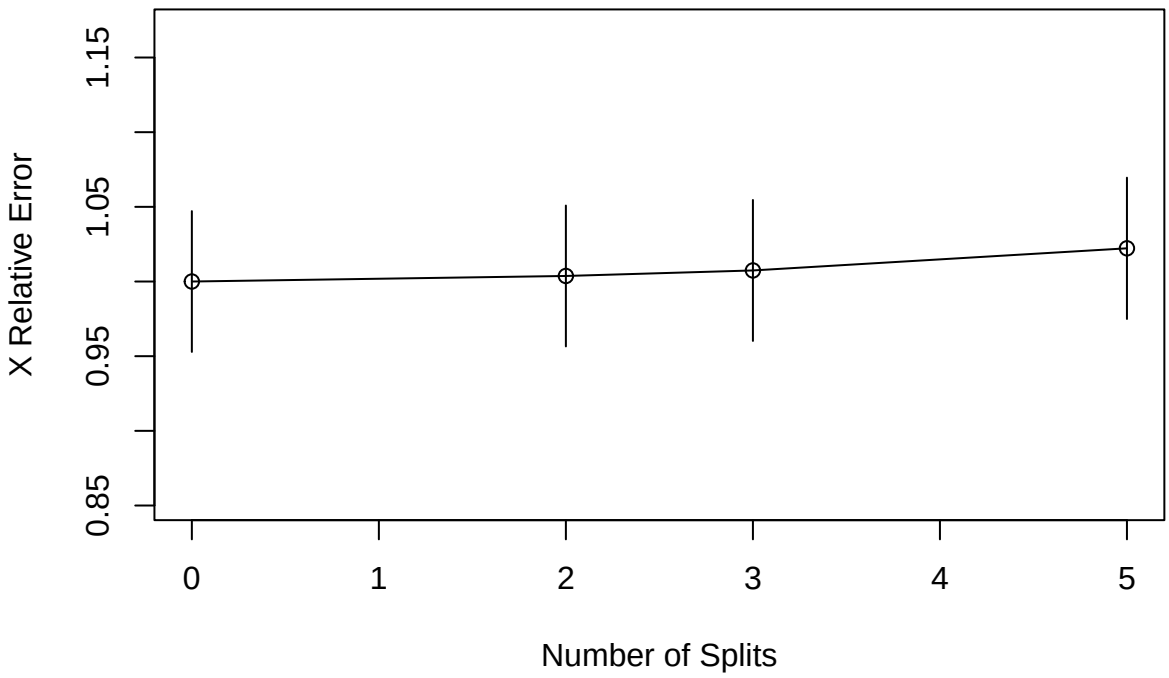
Next steps is to break the data into a Test and Train set to be about 50% for both groups.

```
## Train  Test
##   675   665
```

NO IDEA HELP, DAN!

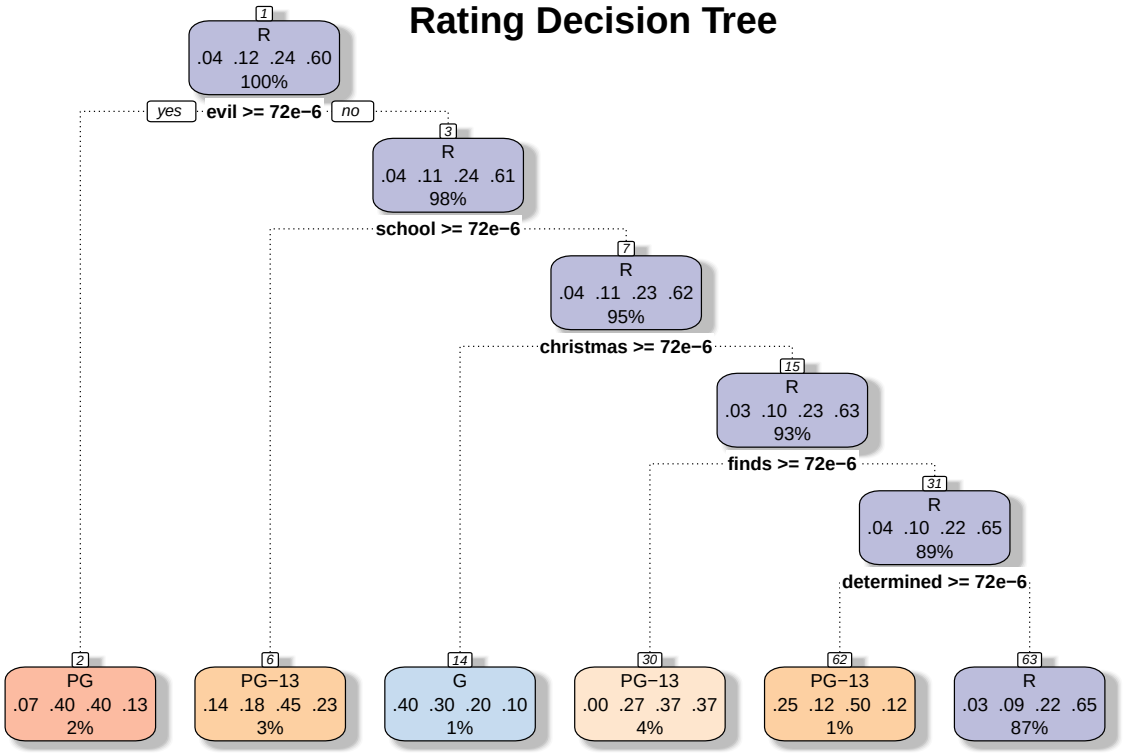
```
##
## Classification tree:
## rpart(formula = Rating ~ ., data = TrainClassFlix, method = "class",
##     control = rpart.control(cp = 0, maxdepth = 5))
##
## Variables actually used in tree construction:
## [1] christmas  determined evil       finds      school
##
## Root node error: 270/675 = 0.4
##
## n= 675
##
##          CP nsplit rel error xerror     xstd
## 1 0.0166667      0   1.00000 1.0000 0.047140
## 2 0.0111111      2   0.96667 1.0037 0.047169
## 3 0.0055556      3   0.95556 1.0074 0.047198
## 4 0.0000000      5   0.94444 1.0222 0.047307
```


Apparent
X Relative
R−square
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
Number of Splits



The R-Square chart shows a difference in the relationship between Apparent and X Relative showing as the number of splits increases, the separation between the two also increases.

The X Relative Error decreases to the lowest point at one split and includes more error at increased split levels.



Due to the low frequency of words throughout all the movies, the max depth set at 5 helps cut down the decision tree for use in classifying the movies based on parental rating. As the max depth increases, the accuracy of the model decreases.

Based on the chart, using the word "evil" is the first word to group within the entire decision tree. With only 2 percent coming off the total 100 percent, this decision chart shows the difficulties of not having a wide variety of movies between all ratings and a low word frequency for each description.

The following confusion matrix indicates the difficulties of using classification as a way to identify parental rating.

```
##        true
## Rating    G  PG PG-13   R
##   G       4   6     0   2
##   PG      0   4     7   3
##   PG-13   3  11    19  26
##   R      25  97   126 332
```

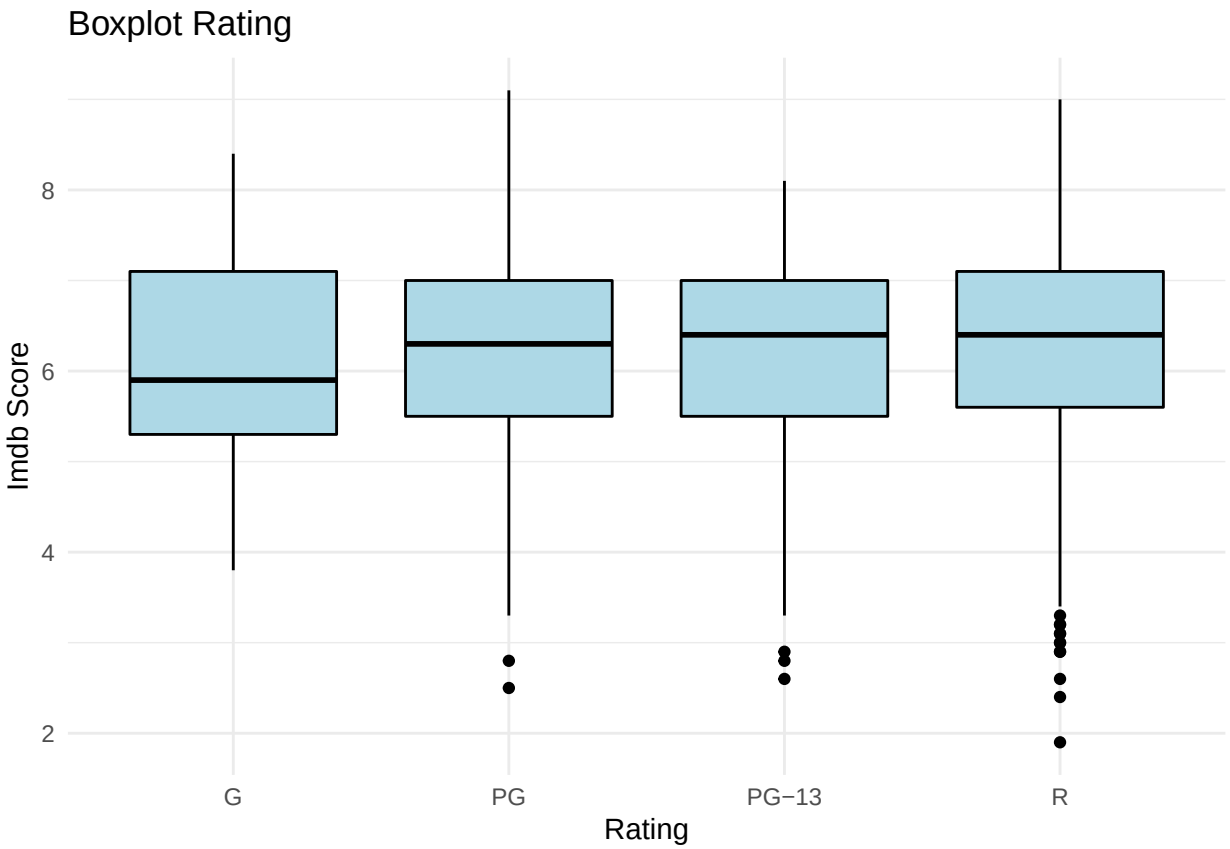
```
## [1] "Correct Ratings: 0.53984962406015"
```

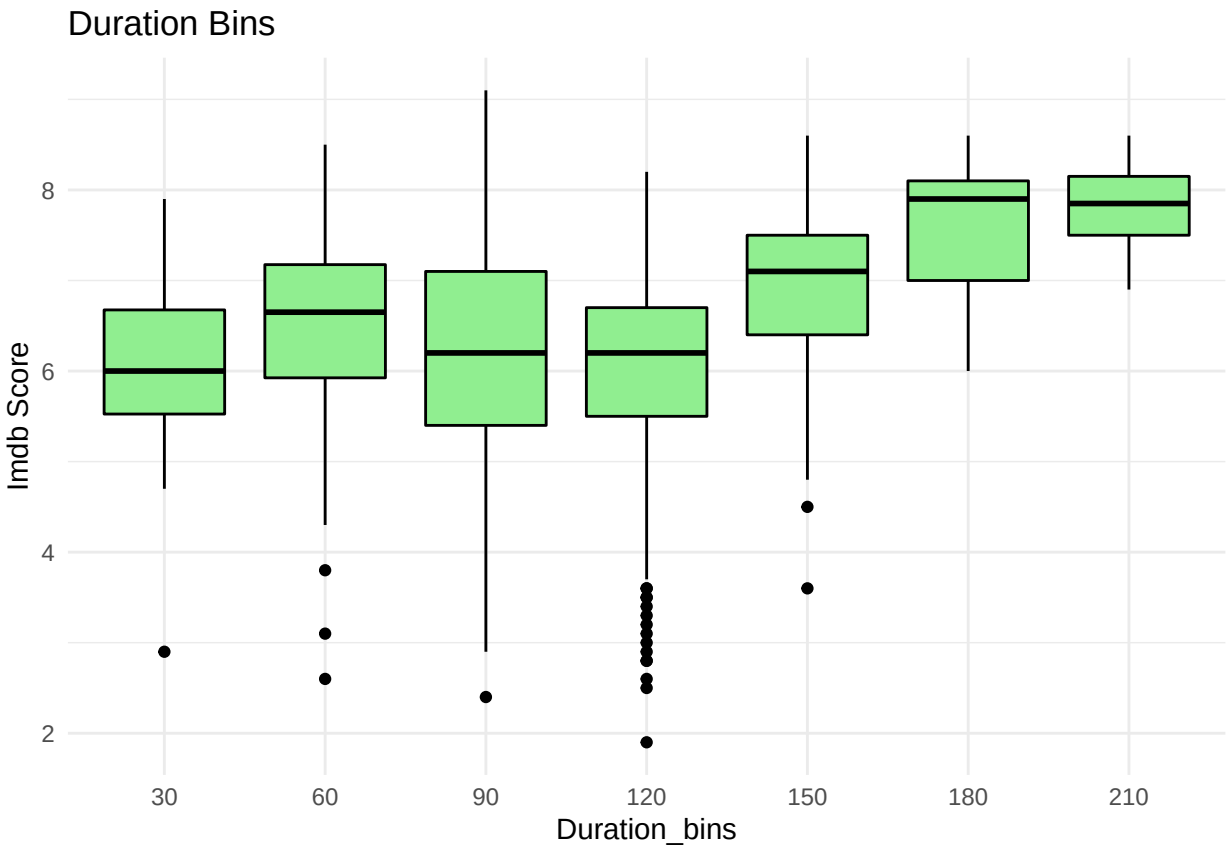
```
## [1] "Incorrect Ratings: 0.46015037593985"
```

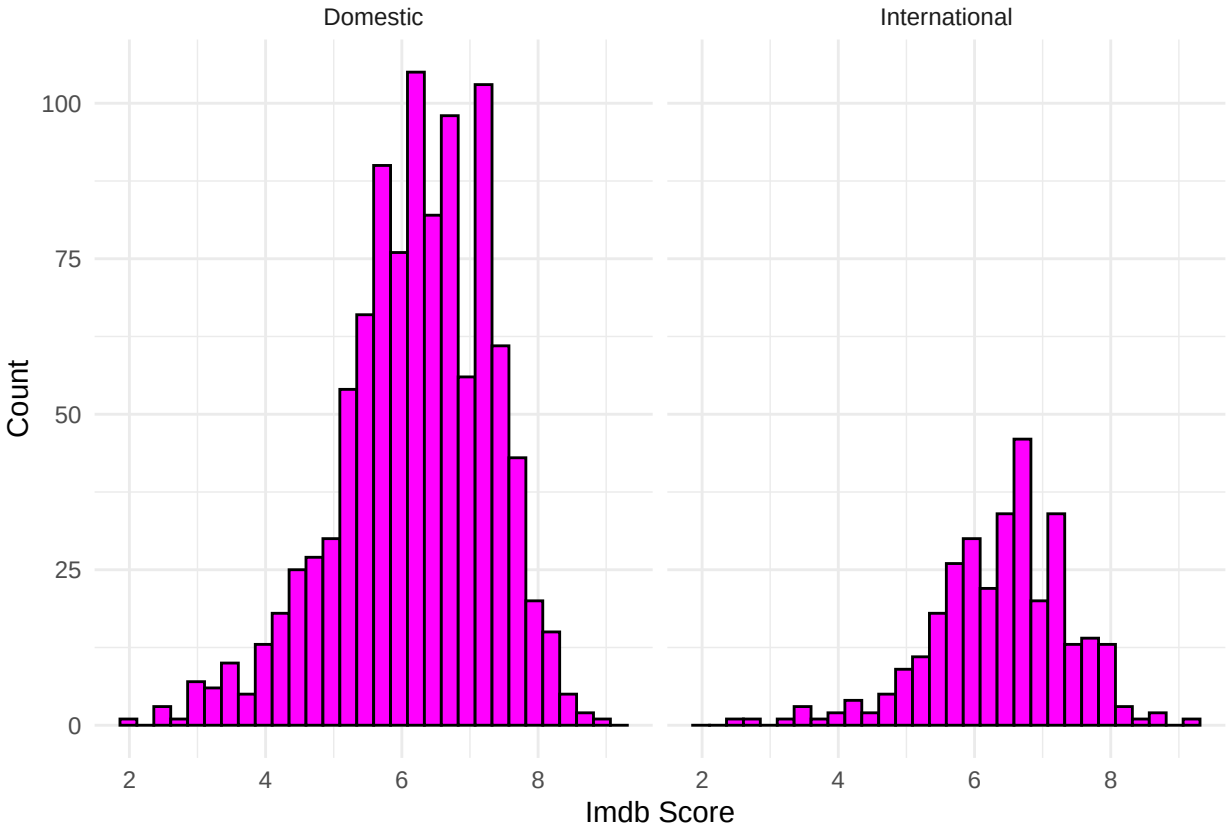
Exploratory Data Analysis on IMDB Scores

When building the model the target variable will be predicting IMDB score.
Before jumping into the Models performing some Exploratory Analysis will help in identify different attributes and the overall structure of the data.

- Boxplot - Rating
- Breaking IMDB score by rating shows that PG, PG-13, and R all have around the same average and are distributed similarly.
- The tails and outliers are different for PG, PG-13, and R, but not too different.
- When looking at the Rating score for G the average and median is different than the remaining cohorot of data.
- Boxplot - Duration Bins
- When looking at the boxplot the duration bin between 1.5 and 2.0 hours appear to have somewhat normally distributed data.
- The remaining duration bins either become smaller and/or are not as evenly distributed.
- Histogram - International
- Breaking the data out as International or domestic, there is quite more data in the domestic category.
- Both have a sizealbe amount of data and are left skewed.
- Domestic here means that the film exclusively released in the United States and no other country.
- Overall
- When breaking data into multiple dimensions the information can be less actionable due to not having enough data.
- The data here, as different dimensions are applied, shows that in most cases the data is not unevenly distributed.
- A caution will be as all of these dimension added into the model will the play in the ability to have predictive capabilities.


Boxplot Rating
Imdb Score
8
6
4
2
G
PG
PG−13
R
Rating


Duration Bins
Imdb Score
8
6
4
2
30
60
90
120
150
180
210
Duration_bins


Domestic
International
Count
100
75
50
25
0
2
4
6
8
Imdb Score

Creating Test/Train Data

Next steps is to break the data into a Test and Train set to be about 50% for both groups.

```
## Train  Test
##   676   664
```

Predicting IDMB Scores

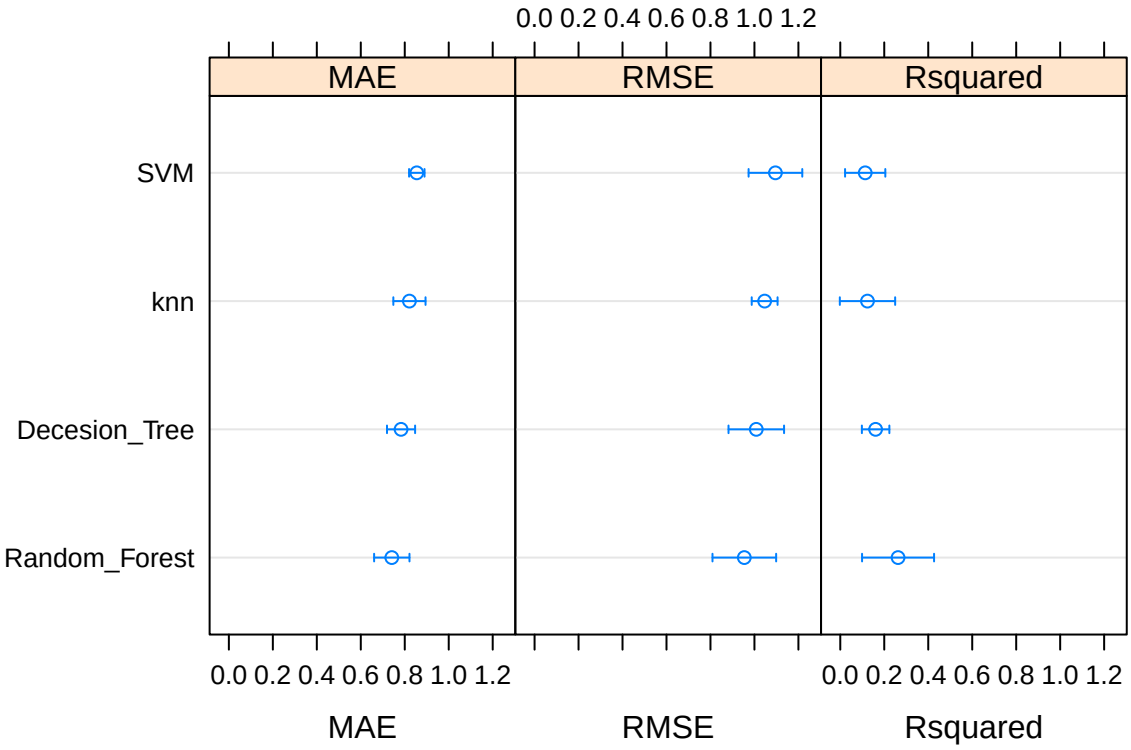
Model Results 1

Looking at Rating, Duration Bins, International Flag, Genres and Director

Interpreting the Results

- When using cross validation on the Train data set, Random Forest looks to be the best among SVM, knn, and Decesion Tree.
- The Rsquared is about 2.2 and the Root Mean Square Error is lower the rest.
- That being said overall the Rsquared are not at a favorable state to accept the model.
- Additionally when trying to test the model Director and Genre are too unique to be used and therefore the model errors out.

```
##
## Call:
## summary.resamples(object = results)
##
## Models: Decesion_Tree, knn, SVM, Random_Forest
## Number of resamples: 3
##
## MAE
##                     Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Decesion_Tree 0.7612581 0.7690432 0.7768283 0.7832051 0.7941786 0.8115289    0
## knn           0.7896707 0.8079498 0.8262290 0.8212933 0.8371045 0.8479801    0
## SVM           0.8446052 0.8467244 0.8488435 0.8547796 0.8598668 0.8708901    0
## Random_Forest 0.7040233 0.7295214 0.7550194 0.7410392 0.7595471 0.7640747    0
##
## RMSE
##                     Min.   1st Qu.    Median      Mean   3rd Qu.     Max. NA's
## Decesion_Tree 0.9615736 0.9813088 1.0010440 1.0083673 1.0317641 1.062484    0
## knn           1.0198361 1.0369283 1.0540204 1.0463624 1.0596255 1.065231    0
## SVM           1.0478487 1.0702159 1.0925830 1.0954701 1.1192808 1.145979    0
## Random_Forest 0.8887343 0.9304109 0.9720875 0.9539377 0.9865393 1.000991    0
##
## Rsquared
##                      Min.    1st Qu.    Median      Mean   3rd Qu.      Max.
## Decesion_Tree 0.13672021 0.14757147 0.1584227 0.1606770 0.1726554 0.1868882
## knn           0.06947011 0.10061544 0.1317608 0.1236772 0.1507808 0.1698007
## SVM           0.07308266 0.09729015 0.1214976 0.1133102 0.1334239 0.1453502
## Random_Forest 0.20382041 0.22715921 0.2504980 0.2627454 0.2922080 0.3339179
##               NA's
## Decesion_Tree    0
## knn              0
## SVM              0
## Random_Forest    0
```



Confidence Level: 0.95

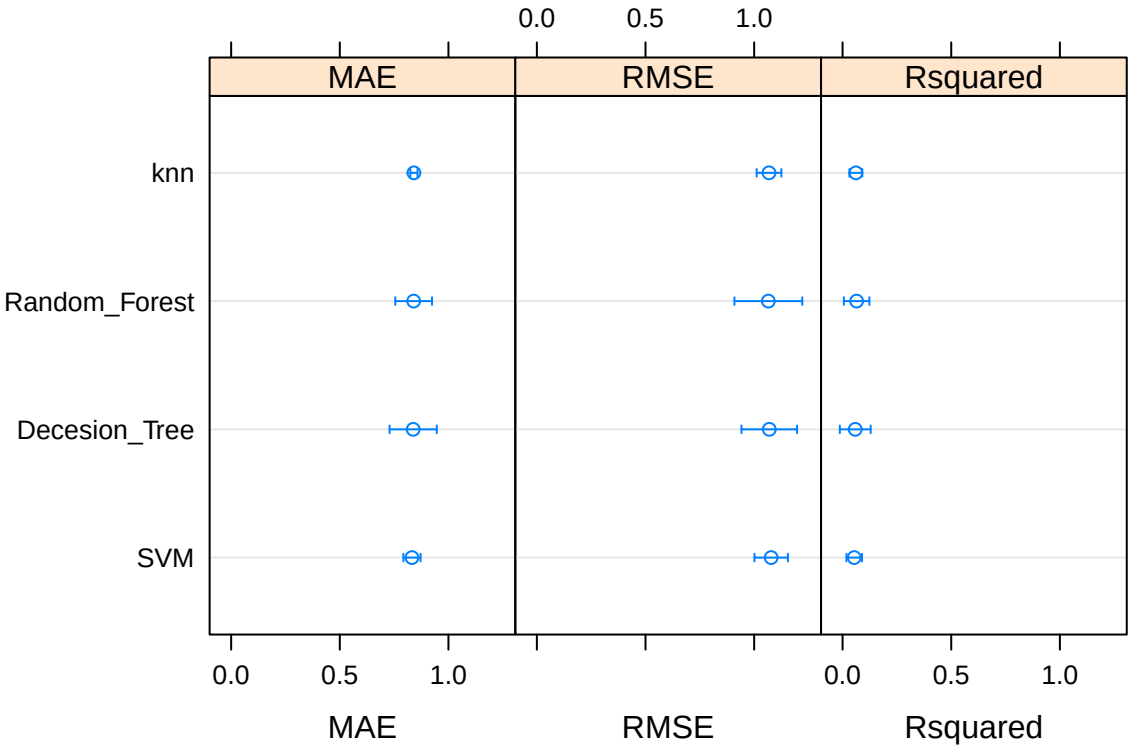
Model Results 2

Looking at Rating, Duration Bins, and International Flag

Interpreting the Results

- All of the models have an Rsquared just over 0.
- The RMSE's are also still too high.
- Overall the model would be hard to accept in applying to real world solution like predicting and improving the IMDB score based on these attributes.

```
##
## Call:
## summary.resamples(object = results)
##
## Models: Decesion_Tree, knn, SVM, Random_Forest
## Number of resamples: 3
##
## MAE
##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Decesion_Tree 0.8071586 0.8127176 0.8182765 0.8377500 0.8530456 0.8878147    0
## knn           0.8335939 0.8387641 0.8439344 0.8411651 0.8449507 0.8459670    0
## SVM           0.8157723 0.8245742 0.8333760 0.8322313 0.8404607 0.8475455    0
## Random_Forest 0.8016061 0.8267182 0.8518304 0.8399497 0.8591215 0.8664126    0
##
## RMSE
##                    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## Decesion_Tree 1.0234610 1.041895 1.060329 1.069676 1.092784 1.125239    0
## knn           1.0542846 1.055067 1.055850 1.068258 1.075244 1.094639    0
## SVM           1.0593274 1.060535 1.061743 1.078382 1.087909 1.114075    0
## Random_Forest 0.9971346 1.037642 1.078150 1.065359 1.099471 1.120792    0
##
## Rsquared
##                     Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## Decesion_Tree 0.02547729 0.04844762 0.07141795 0.05826309 0.07465600 0.07789405
## knn           0.05249723 0.05444881 0.05640039 0.06119403 0.06554243 0.07468447
## SVM           0.04221511 0.04546900 0.04872289 0.05355168 0.05921997 0.06971704
## Random_Forest 0.03728251 0.05761053 0.07793854 0.06457916 0.07822748 0.07851641
##               NA's
## Decesion_Tree    0
## knn              0
## SVM              0
## Random_Forest    0
```



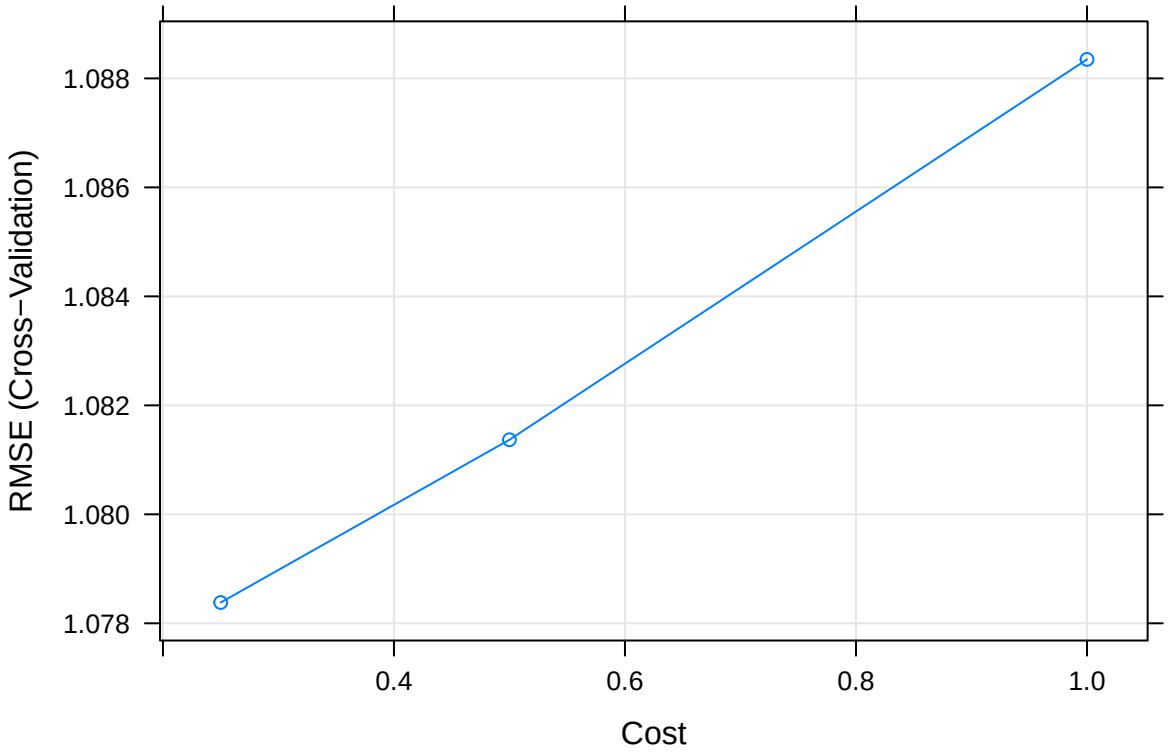
Confidence Level: 0.95

R Squared Results for Models

Reviewing SVM Model

- The computational cost goes up in order to identify the range of RMSE's that could potential occur by using this model.
- Said a different way the cost to cross validate 3 times can be much higher.

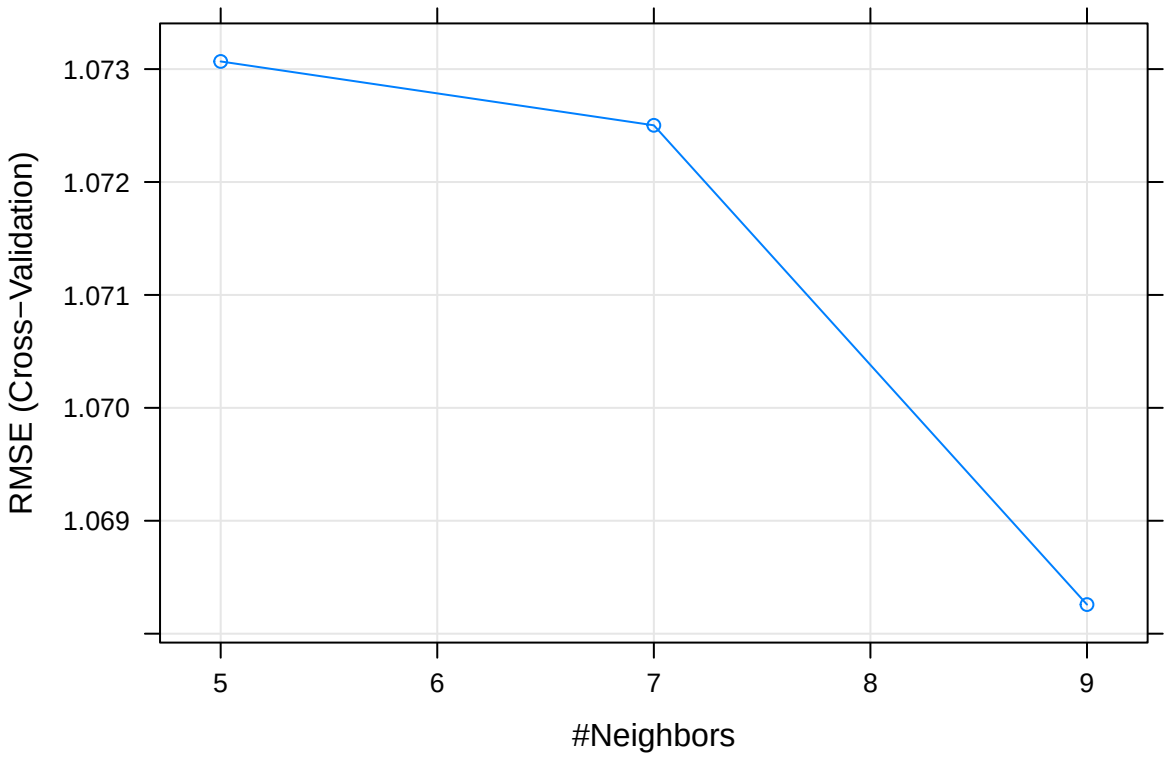
```
## Support Vector Machines with Radial Basis Function Kernel
##
## 676 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 452, 450, 450
## Resampling results across tuning parameters:
##
##   C     RMSE      Rsquared    MAE
##   0.25  1.078382  0.05355168  0.8322313
##   0.50  1.081368  0.05031733  0.8314339
##   1.00  1.088349  0.04645726  0.8357987
##
## Tuning parameter 'sigma' was held constant at a value of 0.06688856
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.06688856 and C = 0.25.
```


RMSE (Cross−Validation)
1.088
1.086
1.084
1.082
1.080
1.078
0.4
0.6
0.8
1.0
Cost

Reviewing KNN Model

- Looking at knn for cross validating 3 times the RMSE goes down as we introduce more number of neighbors.

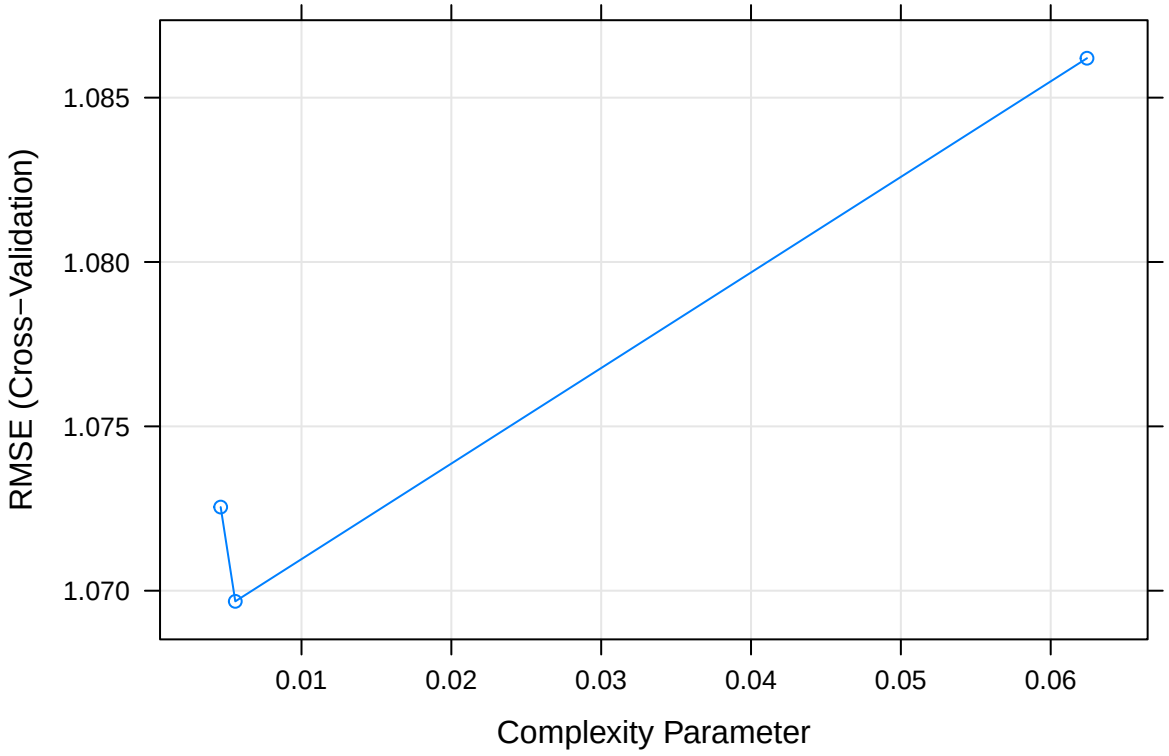
```
## k-Nearest Neighbors
##
## 676 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 451, 450, 451
## Resampling results across tuning parameters:
##
##   k  RMSE      Rsquared    MAE
##   5  1.073068  0.05694859  0.8441783
##   7  1.072502  0.05724067  0.8442520
##   9  1.068258  0.06119403  0.8411651
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```



Reviewing Decision Tree Model

- The complexity of the decision tree dramatically goes up during cross validation.
- From the first to the second the increase is only marginal.
- When going to the third cross validation the jump is more than 6 times.

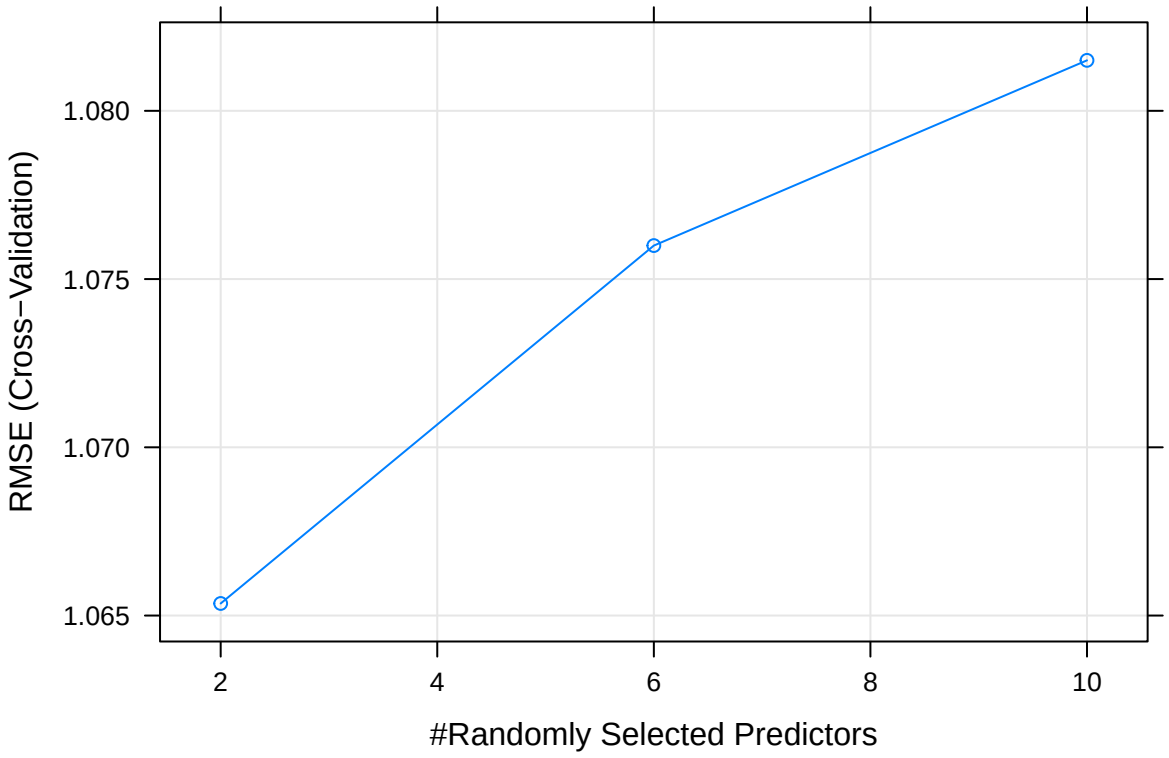
```
## CART
##
## 676 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 450, 451, 451
## Resampling results across tuning parameters:
##
##   cp           RMSE      Rsquared    MAE
##   0.004607214  1.072544  0.05716714  0.8373405
##   0.005582405  1.069676  0.05826309  0.8377500
##   0.062422683  1.086199  0.03998550  0.8614039
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.005582405.
```



Reviewing Random Forest

- This model takes the longest to run and in order to capture a range of 3 RMSE's 10 Randomly Selcted Predictor had to be introduced.

```
## Random Forest
##
## 676 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 451, 450, 451
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared    MAE
##    2    1.065359  0.06457916  0.8399497
##    6    1.075996  0.05756937  0.8404841
##   10    1.081501  0.05397236  0.8421437
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 2.
```



Overall R-Squared and RMSE

Before jumping into predicting IMDB scores the average Rsquares and RMSE's are as followed.

```
##      svm_rsq    knn_rsq   tree_rsq     rf_rsq
## 1 0.05355168 0.06119403 0.05826309 0.06457916
```

```
##   svm_rmse knn_rmse tree_rmse  rf_rmse
## 1 1.078382 1.068258  1.069676 1.065359
```

Pridicting IMDB Scores

- Something interesting happened here, regardless of how low all of the Rsquares were and how how the RMSE were knn was able to achieve a RMSE of 0.07 much lower than what the cross validation chose.
- The hypothesis to why K Nearest Neighbor is able to achieve these results is because most of the data is around an imdb score of 6. knn captures similiarity by looking at the distance or closeness to each data point. Meaning that the Rsquared is a great validation method to understand if the input variable can explain the change in the target variable (IMDB Score).
- Due to the data already being so close to 6, a user would be better off just guessing the score.

```
##   Imdb.Score     tree      svm      knn random_f
## 1        5.7 6.104967 6.389784 6.210769 6.178737
## 2        5.9 6.104967 6.009974 5.959016 6.037264
## 3        6.0 6.995833 6.859163 6.426531 7.179161
## 4        5.8 6.104967 6.009974 5.959016 6.037264
## 5        6.1 6.104967 6.009974 5.959016 6.037264
## 6        6.0 6.104967 6.389784 6.210769 6.178737
```

```
##   results svm_rmse   knn_rmse tree_rmse  rf_rmse
## 1   Train 1.078382 1.06825781  1.069676 1.065359
## 2    Test 1.046301 0.07241446  1.051801 1.044411
```

Summary of the Models

Overall the models were not able to provide value in predicting IMDB score. More attributes might help in having the precision to to predict a score but simply using Genre, Rating, if the movie was international and Duration Bins, does not help explain the score. K-Nearest Neighbors appeared to have an incredibly low Root Mean Square error but is overshadowed by the cross validation reducing the ability to adopt the model.

Association Rule Mining

Association rule mining require the data to be transaction type structure. Meaning that each movie will be treated as a separate transaction with at most 3 different genre's associated to them. The attempt here is to identify the next movie to watch by simply looking to Genre.

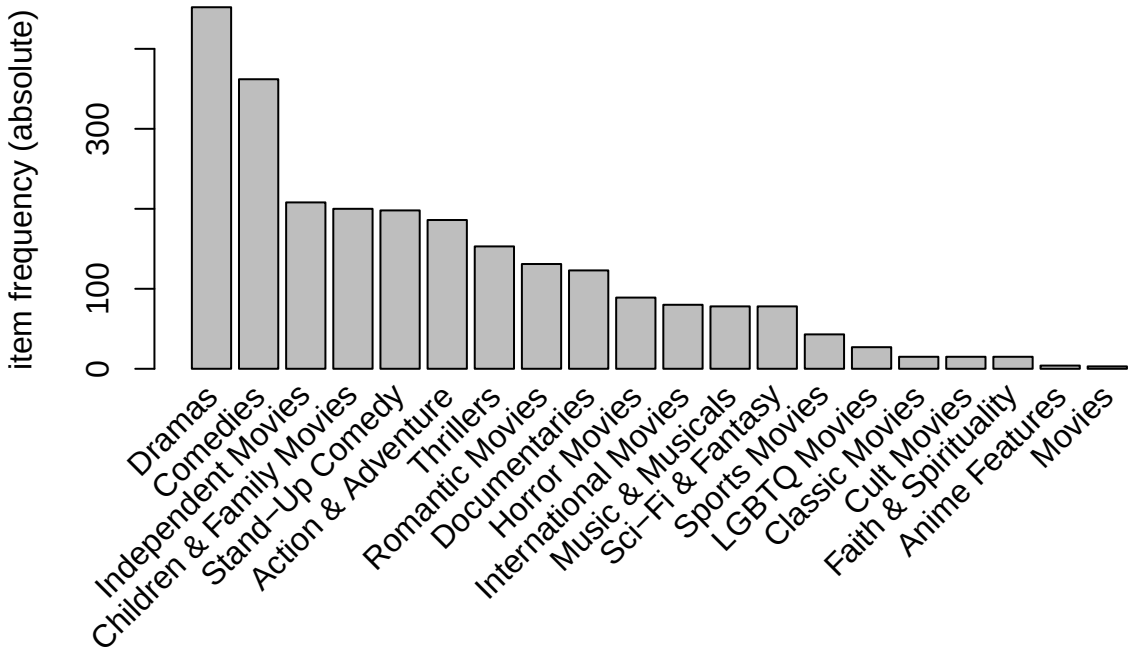
- The most watched type of movie is either Drama or Comedies.
- After those two genres the movies step down dramatically from around 350 to 200 titles.

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 1042 rows [1, 2,
## 5, 6, 7, 8, 11, 12, 13, 15, 16, 17, 23, 24, 25, 26, 27, 28, 29, 30, ...].
```

```
## Warning in asMethod(object): removing duplicated items in transactions
```

```
##      items
## [1]  {Comedies}
## [2]  {Dramas,Thrillers}
## [3]  {Horror Movies,Independent Movies,Thrillers}
## [4]  {Dramas,Faith & Spirituality,Romantic Movies}
```

```
## [5]  {Action & Adventure,Sci-Fi & Fantasy}
## [6]  {Dramas,Thrillers}
## [7]  {Action & Adventure}
## [8]  {Dramas,LGBTQ Movies}
## [9]  {Independent Movies,Sci-Fi & Fantasy,Thrillers}
## [10] {Comedies,Dramas,Independent Movies}
## [11] {Dramas,Independent Movies}
## [12] {Action & Adventure}
## [13] {Action & Adventure,Dramas}
## [14] {Dramas,Independent Movies,Romantic Movies}
## [15] {Documentaries}
## [16] {Children & Family Movies}
## [17] {Comedies,Romantic Movies}
## [18] {Children & Family Movies,Dramas,Romantic Movies}
## [19] {Children & Family Movies,Dramas,Romantic Movies}
## [20] {Children & Family Movies,Dramas,Romantic Movies}
```



The data is now in a transaction format. We will look at *support* and *confidence*. - *support* is an indication of how frequently an items appear in the data - *confidence* indicates the number of times the if-then statements are found true.

- Setting the minimum support to 0.001 pulls as many items into the dataset, but not all items to avoid bringing transactions that didn't have as many associations.
- The minimum confidence is set to 0.8 in order to bring in items that have a confidence over 0.8.

Looking to some summary info about the rules illuminates some interesting information such as:

- The number of rules generated: 8
- The distribution of rules by length: Most rules are 3 items long
- The summary of quality measures: interesting to see ranges of support, lift, and confidence.
- The information on the data mined: total data mined, and minimum parameters.

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[20 item(s), 1340 transaction(s)] done [0.00s].
## sorting and recoding items ... [20 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [8 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

## set of 8 rules
##
## rule length distribution (lhs + rhs):sizes
## 2 3
## 1 7
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    3.00    3.00    2.88    3.00    3.00
##
## summary of quality measures:
##     support         confidence      coverage           lift          count
##  Min.   :0.0015   Min.   :0.80   Min.   :0.0015   Min.   : 2.4   Min.   : 2.0
##  1st Qu.:0.0015   1st Qu.:0.82   1st Qu.:0.0015   1st Qu.: 2.4   1st Qu.: 2.0
##  Median :0.0026   Median :0.93   Median :0.0030   Median : 2.8   Median : 3.5
##  Mean   :0.0039   Mean   :0.91   Mean   :0.0046   Mean   : 4.4   Mean   : 5.2
##  3rd Qu.:0.0050   3rd Qu.:1.00   3rd Qu.:0.0060   3rd Qu.: 4.4   3rd Qu.: 6.8
##  Max.   :0.0090   Max.   :1.00   Max.   :0.0112   Max.   :10.9   Max.   :12.0
##
## mining info:
##               data ntransactions support confidence
##  GenreTransactions          1340   0.001        0.8
```

Exploring Metrics to Evaluate - Confidence

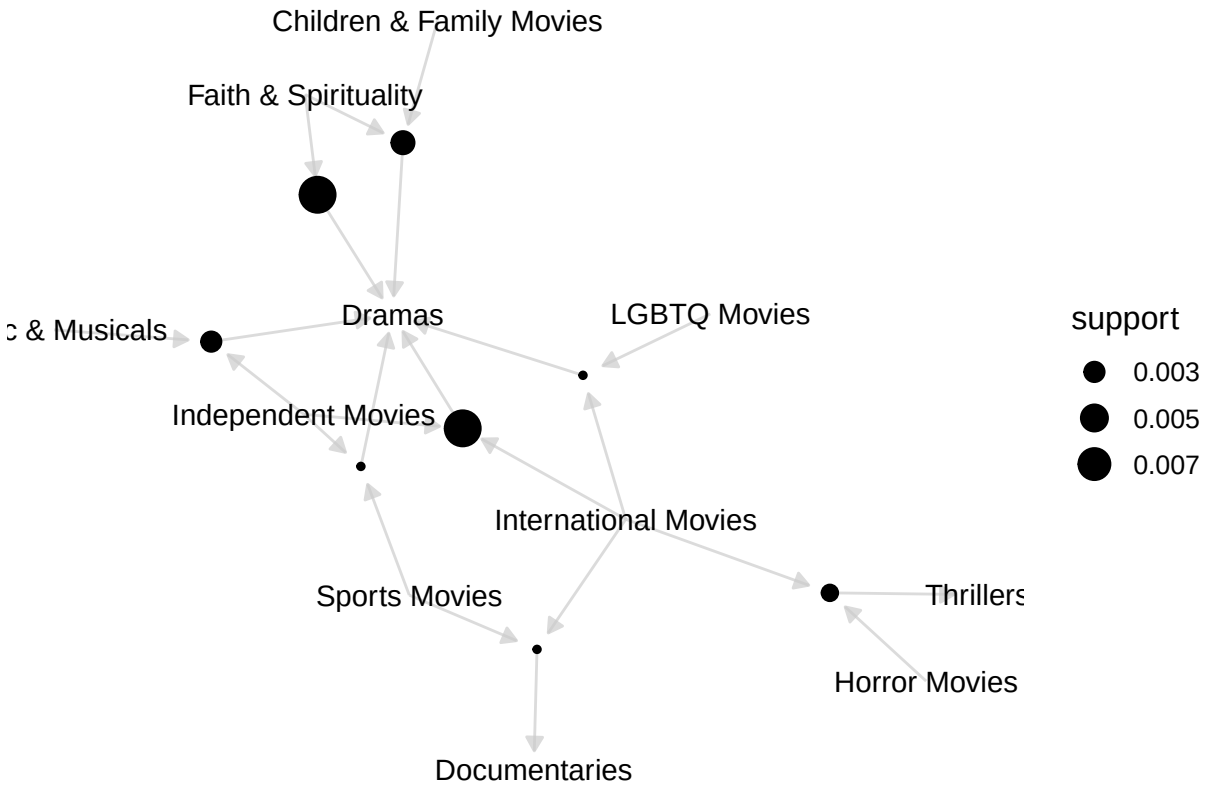
- The chart below shows a general sense of what the rules look like when sorting by confidence.

- The most confident predictors are when an individual watches an international or independent film that is either about Sports or LGBTQ has a confidence of 1 and in only one case 0.86. The movies suggested to watch next is a Drama.
- In the case of the movie being International and sports the next to watch would be a documentary.
- If a person watches a horror movies and is international the next movie suggestion should be a thriller.

```
##     lhs                           rhs             support confidence coverage lift count
## [1] {International Movies,
##      LGBTQ Movies}             => {Dramas}         0.0015       1.00   0.0015  3.0     2
## [2] {International Movies,
##      Sports Movies}            => {Documentaries}  0.0015       1.00   0.0015 10.9     2
## [3] {Independent Movies,
##      Sports Movies}            => {Dramas}         0.0015       1.00   0.0015  3.0     2
## [4] {Horror Movies,
##      International Movies}     => {Thrillers}      0.0022       1.00   0.0022  8.8     3
## [5] {Independent Movies,
##      International Movies}     => {Dramas}         0.0090       0.86   0.0104  2.5    12
## [6] {Children & Family Movies,
##      Faith & Spirituality}     => {Dramas}         0.0037       0.83   0.0045  2.5     5
## [7] {Faith & Spirituality}     => {Dramas}         0.0090       0.80   0.0112  2.4    12
## [8] {Independent Movies,
##      Music & Musicals}         => {Dramas}         0.0030       0.80   0.0037  2.4     4
```

Plotting the Association Rule

Most movies lead back to Dramas, Documentaries or Thrillers, with a high support for Drama.



Summary for Association Rule mining

This rule mining technique proved to be valuable in helping identify the next movie to watch. Although Drama was the most watched genre, what was interesting is the models technique in identifying the next movie to watch is Documentary after just watching a International Sports movie. This is outside the norm of what is expected in just normal behavior. In addition the lift was substantially higher than the other movies with confidence levels over 0.80. That suggestion could have a noticeable impact on helping Netflix watcher discover what they might like to watch next.

Conclusion

In terms of accurately grouping movies as they arrive in the recommender system, clustering and classification both struggled due primarily to the low frequency of words in the description with an average of 14 words per movie. Classification did end up slightly more successful than clustering with about a 54 percent success rate but still did not perform well with movies that had lower parental ratings. Pulling a summary of the movie from other websites with more information about the film might make classification and clustering viable methods for initially grouping the data in the future. Additionally, getting more movies with lower ratings into the system could help the reliability of clustering and classification as movies with a R rating greatly outnumbered the other movies.

References

Makhija, S. (2021, July). Netflix Movies and TV Shows 2021. Kaggle. https://www.kaggle.com/satpreetmakhija/netflix-movies-and-tv-shows-2021