Movie Prediction System

Justin A. Kolodziej

1. Introduction

A key feature of a movie streaming service such as NetFlix® is to recommend movies to users that they will rate highly based on their own ratings and the ratings of other users. The goal of the project is to develop a machine learning algorithm that will minimize root mean squared error (RMSE target <= 0.8649) of movie ratings on a scale of 0 to 5 stars, using the 10 million rating version of the MovieLens database. Multiple methods are used to identify patterns in a representative subsample of the data, and algorithms and tuning parameters selected that run in a reasonable time and give reasonably accurate results. In the end a hybrid approach is selected and run on the full dataset.

2. Methods/Analysis

The MovieLens data set contains the following columns:

```
##    userId movieId rating timestamp                         title
## 1:      1     122      5 838985046              Boomerang (1992)
## 2:      1     185      5 838983525               Net, The (1995)
## 3:      1     292      5 838983421               Outbreak (1995)
## 4:      1     316      5 838983392               Stargate (1994)
## 5:      1     329      5 838983392 Star Trek: Generations (1994)
## 6:      1     355      5 838984474       Flintstones, The (1994)
##                           genres
## 1:                Comedy|Romance
## 2:         Action|Crime|Thriller
## 3:  Action|Drama|Sci-Fi|Thriller
## 4:       Action|Adventure|Sci-Fi
## 5: Action|Adventure|Drama|Sci-Fi
## 6:       Children|Comedy|Fantasy
```

The "userid", "movieid", "rating", and "title" columns appear straightforward, while the "timestamp" and "genres" columns could use additional processing. There might be patterns in the year/month/day of rating at least, as well as whether a movie is or is not a drama, action, crime, etc. movie, and it may simplify analysis over having separate levels of a factor for "Action" vs. "Action|Adventure". The lack of movies in many genres, e.g. "Action|Adventure|Comedy|Horror|Crime|Western|Sci-Fi|Fantasy|IMAX" means that adding these columns adds to the storage requirements.

The "edx" dataset portion is split into a "train" and "test" set with 80% of the data in the "train" and 20% in the "test" dataset for analysis.

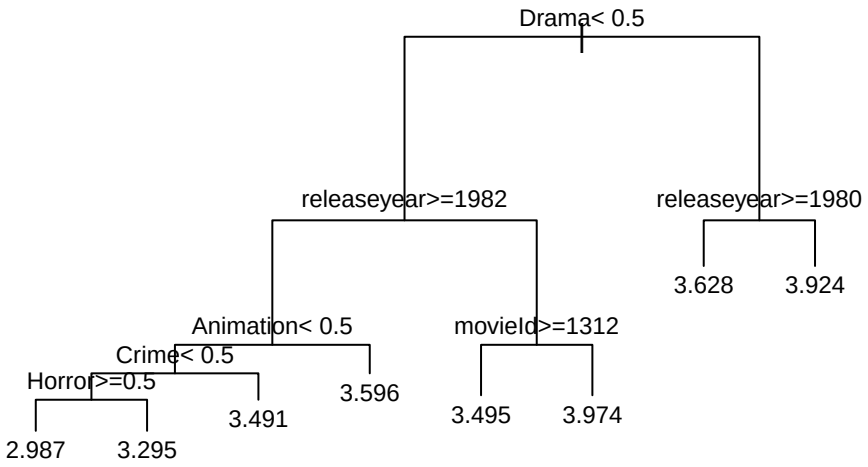
```
##    userId movieId rating releaseyear year month day      week Action Adventure
## 1:      1     185      5        1995 1996     8   2 839116800      1         0
```

```
## 2:      1     292      5        1995 1996     8   2 839116800      1         0
## 3:      1     316      5        1994 1996     8   2 839116800      1         1
## 4:      1     329      5        1994 1996     8   2 839116800      1         1
## 5:      1     355      5        1994 1996     8   2 839116800      0         0
## 6:      1     362      5        1994 1996     8   2 839116800      0         1
##    Animation Children Comedy Crime Documentary Drama Fantasy FilmNoir Horror
## 1:         0        0      0     1           0     0       0        0      0
## 2:         0        0      0     0           0     1       0        0      0
## 3:         0        0      0     0           0     0       0        0      0
## 4:         0        0      0     0           0     1       0        0      0
## 5:         0        1      1     0           0     0       1        0      0
## 6:         0        1      0     0           0     0       0        0      0
##    IMAX Musical Mystery Romance SciFi Thriller War Western
## 1:    0       0       0       0     0        1   0       0
## 2:    0       0       0       0     1        1   0       0
## 3:    0       0       0       0     1        0   0       0
## 4:    0       0       0       0     1        0   0       0
## 5:    0       0       0       0     0        0   0       0
## 6:    0       0       0       1     0        0   0       0

##    userId movieId rating releaseyear year month day      week Action Adventure
## 1:      1     122      5        1992 1996     8   2 839116800      0         0
## 2:      1     356      5        1994 1996     8   2 839116800      0         0
## 3:      1     377      5        1994 1996     8   2 839116800      1         0
## 4:      1     466      5        1993 1996     8   2 839116800      1         0
## 5:      1     594      5        1937 1996     8   2 839116800      0         0
## 6:      2     539      3        1993 1997     7   7 868147200      0         0
##    Animation Children Comedy Crime Documentary Drama Fantasy FilmNoir Horror
## 1:         0        0      1     0           0     0       0        0      0
## 2:         0        0      1     0           0     1       0        0      0
## 3:         0        0      0     0           0     0       0        0      0
## 4:         0        0      1     0           0     0       0        0      0
## 5:         1        1      0     0           0     1       1        0      0
## 6:         0        0      1     0           0     1       0        0      0
##    IMAX Musical Mystery Romance SciFi Thriller War Western
## 1:    0       0       0       1     0        0   0       0
## 2:    0       0       0       1     0        0   1       0
## 3:    0       0       0       1     0        1   0       0
## 4:    0       0       0       0     0        0   1       0
## 5:    0       1       0       0     0        0   0       0
## 6:    0       0       0       1     0        0   0       0
```

##2.1 Exploration with regression trees

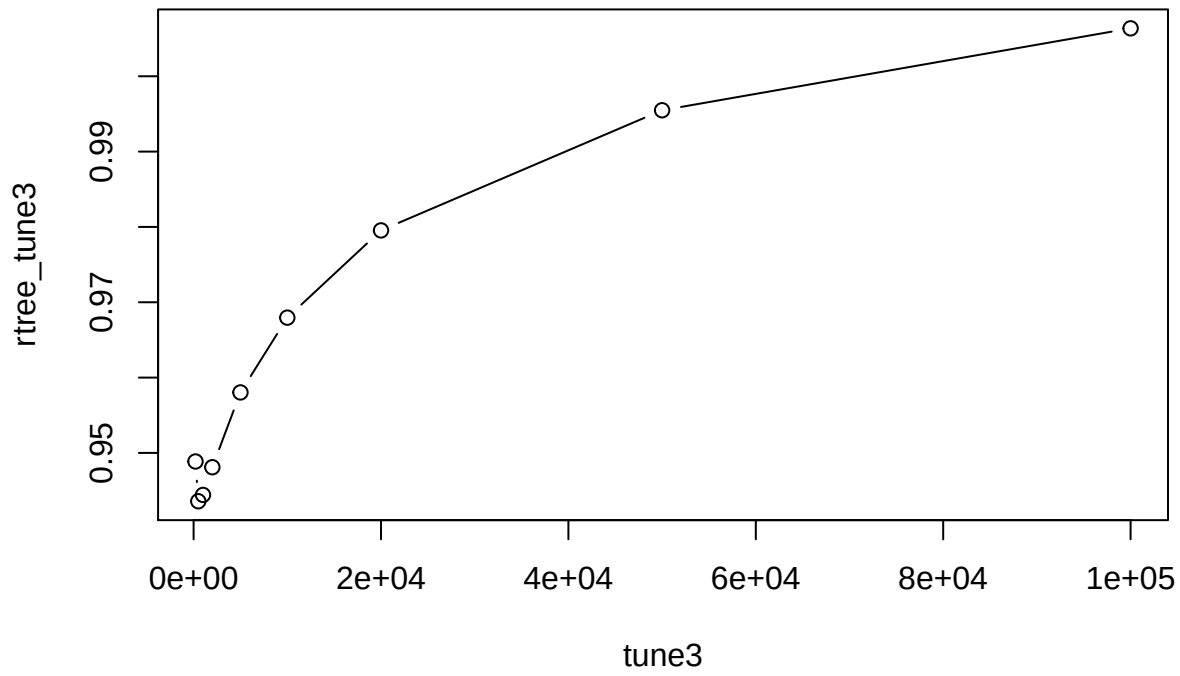
Since the wrangled dataset now has a large number of binary predictors, a regression tree would seem to be a natural choice to determine which predictors are more important in predicting movie ratings. By using a large complexity parameter (cp = .0025) the tree can be made simple enough to graph as a starting point.




```
##       Drama releaseyear     movieId      Horror   Animation       Crime
##   143056.63   132575.35    39316.91    24228.21    22224.84    20579.42
```

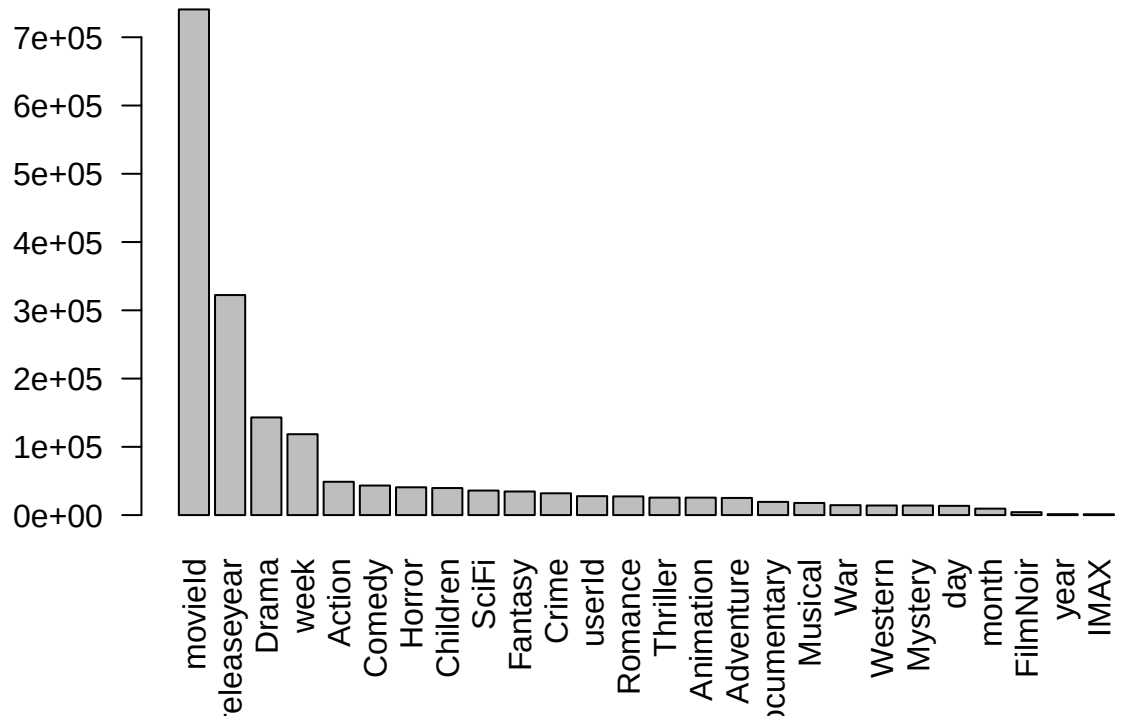
Most of the splits are by genre, so at a gross level using genre to predict rating appears to be a reasonable idea.

The regression tree can be tuned by minimum node size in addition to complexity parameter. It seems preferable, in fact, to avoid situations where some nodes have only a few observations and others have tens of thousands. Once an optimal node size is found, the variable importance can be graphed to see what predictors are



most useful.

```
## Best RMSE = 0.9435849
```



The best prediction appears to be a minsplit of 500 and the best predictors at that level appear to be movieId, releaseyear, Drama, and week. However the minimum RMSE is >0.94 so other approaches will be needed.

2.2 Least-squares regression

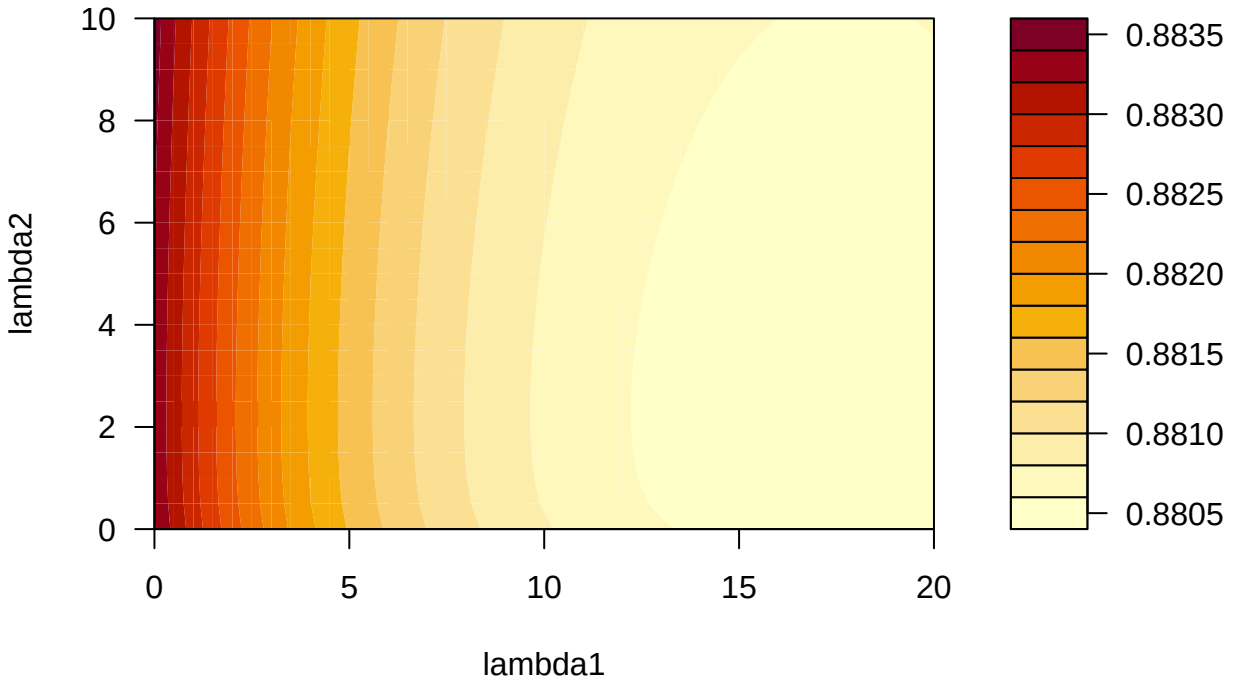
Perhaps other methods focusing on the most important variables found by the regression tree will give improved results without an extraordinary compute time or storage requirement. Also, another approach might be to use least-squares on the userId and movieId without any other knowledge of the dataset. All are compared to merely predicting the mean of all ratings for every observation.

```
##                            method      RMSE
## 1                            Mean 1.0602732
## 2                           Users 0.9794099
## 3                     Users+Movies 0.8834172
## 4                          Movies 0.9440821
## 5              Movies+ReleaseYear 0.9440821
## 6       Movies+ReleaseYear+Drama 0.9412334
## 7 Movies+ReleaseYear+Drama+Week 0.9409809
```

Note that MovieId + UserId is much better than any of the MovieId plus other options.

2.3 Penalized Least Squares and multiple lambdas

Penalized (or normalized) least squares reduces the influence of predictor means with few samples. Standard penalized least squares has one penalty parameter (lambda); there is no reason not to use one lambda per predictor (movieId and userId). These lambdas can be tuned to the training set and the results graphed as a contour plot:

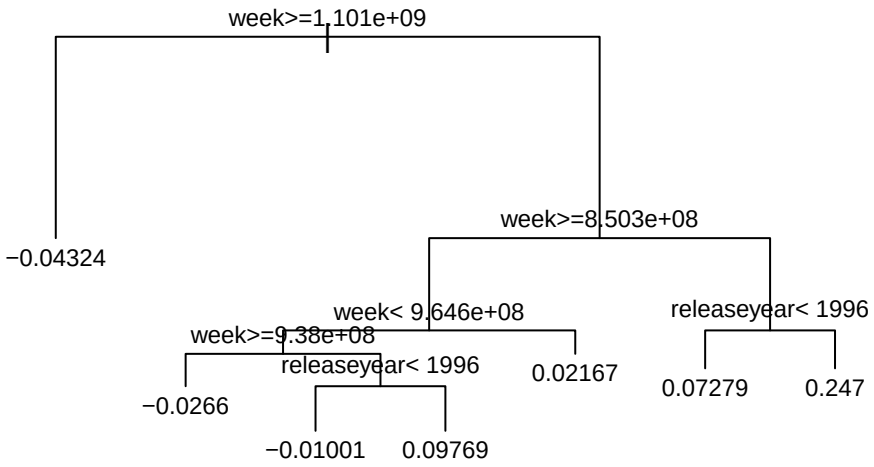


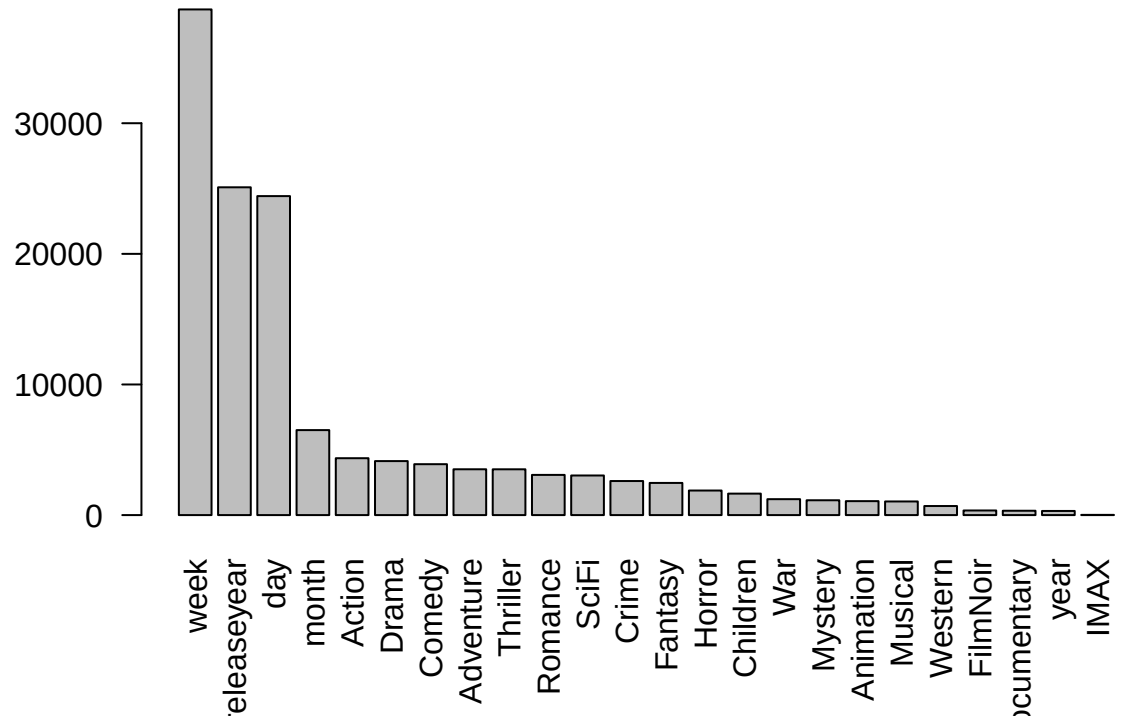
The optimal lambda1 is 17.5 and lambda2 is 2.5. The RMSE with those parameters is:

```
## penalized model RMSE =  0.88046
```

2.3 Hybrid approaches

After using the penalized least-squared prediction algorithm, there is still a residual left over. This can be subtracted out and the residuals predicted or analyzed using another algorithm. First the regression tree is tried.

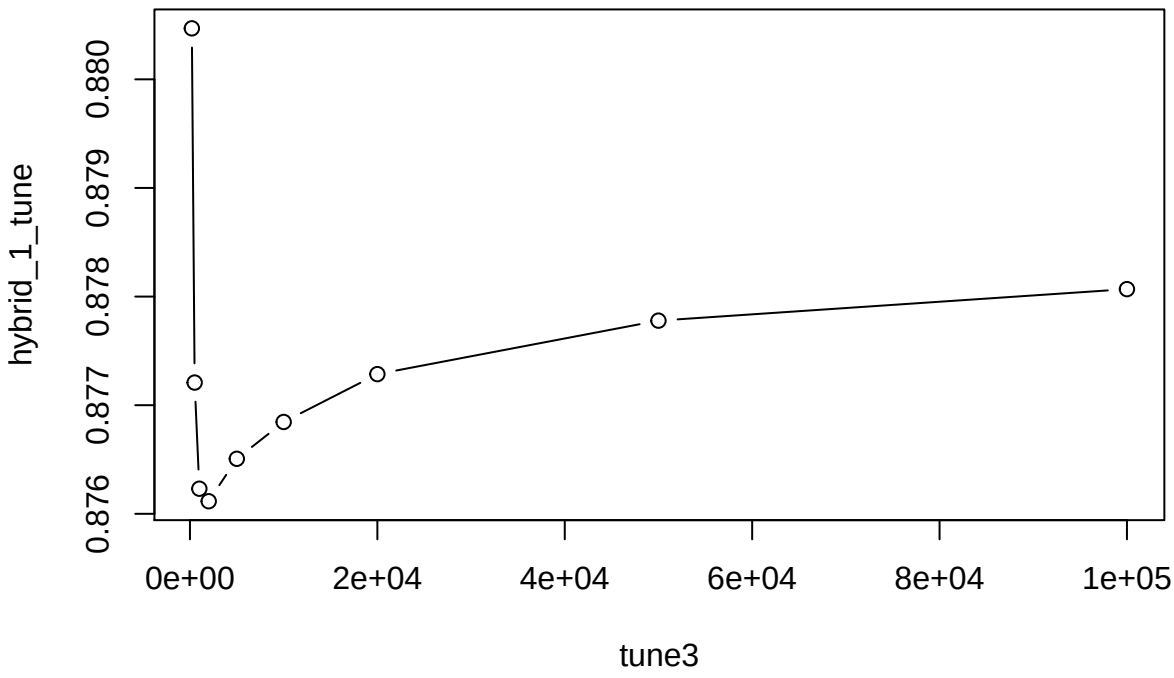

week>=1.101e+09
−0.04324
week>=8.503e+08
week< 9.646e+08
releaseyear< 1996
week>=9.38e+08
releaseyear< 1996
0.02167
0.07279
0.247
−0.0266
−0.01001
0.09769



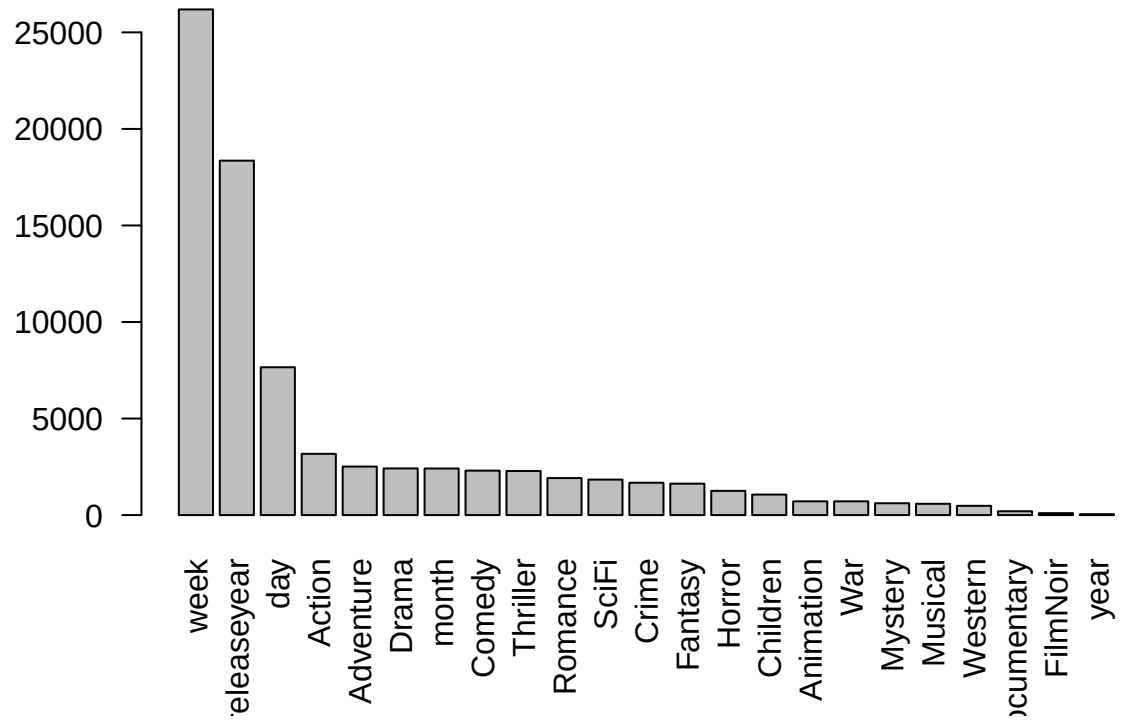
Then the RMSE for the train and test set using the original tuning is:

```
## hybrid model 1 RMSE =  0.8772083
```

The tuning for the minsplit parameter may be different for the residual vs. the original ratings, though.




```
## Model with penalized effects plus tree, RMSE= 0.8761157
```



2.4 Generalized Additive Model (GAM)

An alternative to the regression tree is a Generalized Additive Model. A GAM is an extension of a GLM (Generalized Linear Model) except smooth functions of predictors such as cubic splines are fitted to the data rather than linear functions, giving more flexibility. R has an efficient algorithm to train and predict GAM models so it can be used on the most important predictors from the regression tree results.

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## residual ~ s(as.numeric(week), bs = "cs") + s(releaseyear, bs = "cs") +
##     s(day, bs = "cs")
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0009402  0.0003251  -2.892  0.00383 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                        edf Ref.df      F p-value
## s(as.numeric(week)) 8.881      9 2110.3  <2e-16 ***
## s(releaseyear)      8.606      9  337.1  <2e-16 ***
```

```
## s(day)              5.142      9   19.5  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.00267   Deviance explained = 0.268%
## fREML = 9.22e+06  Scale est. = 0.7582    n = 7200080
```

```
## Model with penalized effects plus GAM, RMSE= 0.8793082
```

The GAM results for 3 individual predictors are not quite as good as the results for the regression tree with all remaining predictors (minus movieId and userId).

3 Results

The best results so far are from penalized regression on userId and movieId and a regression tree for the residual from the rest of the predictors. Training this model on the full edx set to predict the validation set gives a result:

```
## Final RMSE for full edx/validation sets =  0.8744894
```

This is not as good as the .859 target.

4 Conclusion

A hybrid approach with 2 levels of prediction was able to achieve a better result than either algorithm alone, but not to the level required by the parameters of the project. However, other approaches exist that were not implemented due to time and resource constraints. Based on the results, one called gradient boosting trees that iterates building regression trees on the residuals of the previous tree seems promising, though overtraining seems likely.

GAM has a large number of options for building a model; only a model of form f(x)+f(y)+f(z) was used when models such as f(x)+f(y)+f(z)+g(x,y,z), etc. are permissible. Exploring them may give the model additional predictive power.