PALLAV ANAND

A report on Hollywood Blockbuster Case Study

Contents

INTRODUCTION

Hollywood blockbuster data mining challenge

The challenge is to use the training data set to build a model to predict the target variable 'Category' in the training file. The prediction accuracy will be tested by applying the model to the scoring data set.

Data Exploration

I started by converting the files into csv and reading the files into R

```
> training=read.csv('/Users/04pallav/Downloads/Training\ Sheet1.csv',header=TRUE)
> dim(training)
[1] 1196   15
```

The training data has 1196 rows and 15 columns.
We have a multi-class classification problem at hand where the target variable "Category" has 9 levels.

The clues are in the summaries.

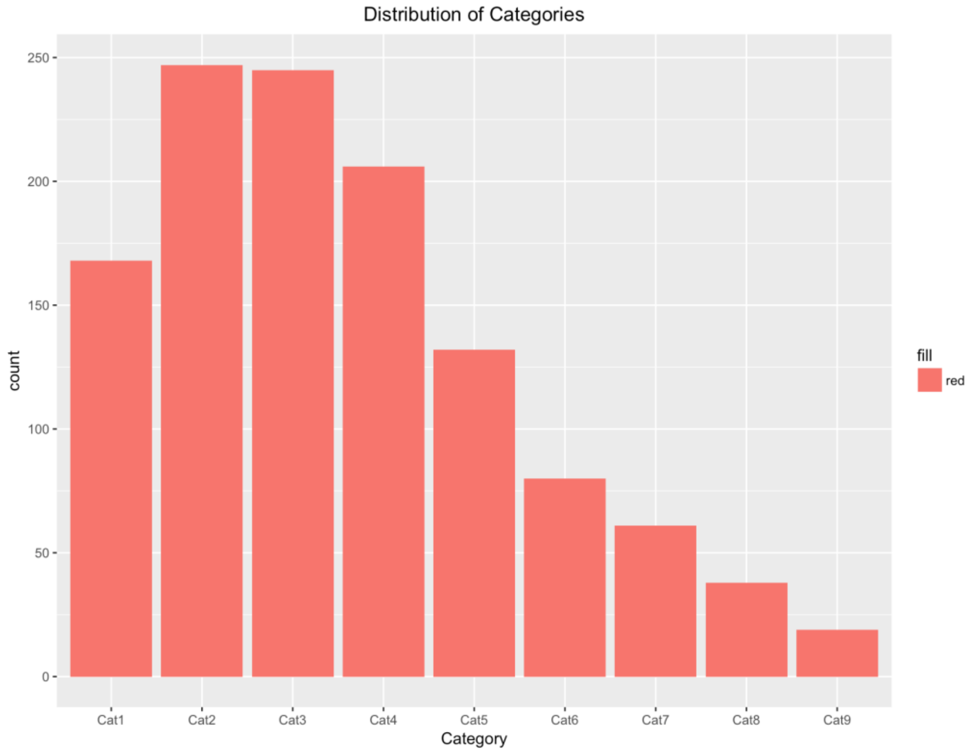
```
> summary(training)
       id                     name                          display_name  production_year  movie_sequel                  creative_type
 Min.   :    70115   10,000 B.C.       :   1   Death at a Funeral  :   2   Min.   :2007    Min.   :0.00000   Contemporary Fiction:638
 1st Qu.: 48080115   12 Rounds         :   1   (500) Days of Summer:   1   1st Qu.:2008    1st Qu.:0.00000   Fantasy             :131
 Median : 93910115   127 Hours         :   1   10,000 B.C.         :   1   Median :2009    Median :0.00000   Historical Fiction  :100
 Mean   : 89282030   1408              :   1   12 Rounds           :   1   Mean   :2009    Mean   :0.09783   Dramatization       : 95
 3rd Qu.:135432615   2010 Oscar Shorts:   1   127 Hours           :   1   3rd Qu.:2010    3rd Qu.:0.00000   Science Fiction     : 89
 Max.   :176970115   2011 Oscar Shorts:   1   1408                :   1   Max.   :2011    Max.   :1.00000   Factual             : 60
                     (Other)           :1190   (Other)             :1189                                     (Other)             : 83
                              source                  production_method               genre          language
 Original Screenplay              :629   Animation/Live Action      :  36   Drama            :321   English :1144
 Based on Fiction Book/Short Story:218   Digital Animation          :  53   Comedy           :260   Hindi   :  16
 Based on Real Life Events        :128   Hand Animation             :   6   Thriller/Suspense:131   French  :  14
 Remake                           : 65   Live Action                :1093   Action           :124   Spanish :   6
 Based on TV                      : 38   Multiple Production Methods:   3   Adventure        :107   German  :   3
 Based on Comic/Graphic Novel     : 36   Stop-Motion Animation      :   5   Romantic Comedy  : 80   Japanese:   2
 (Other)                          : 82                                      (Other)          :173   (Other) :  11
              board_rating_reason movie_board_rating_display_name   movie_release_pattern_display_name      total             Category
 International - to be excluded:  83   G        : 39               Exclusive           : 30              Min.   :   1.0   Min.   :1.000
 General                       :  34   NC-17    :  3               Expands Wide        : 21              1st Qu.:  11.0   1st Qu.:2.000
 for language                  :   9   Not Rated: 83               IMAX                :  3              Median :  40.5   Median :3.000
 for language.                 :   7   PG       :182               Limited             :342              Mean   : 104.7   Mean   :3.564
 for brief strong language     :   6   PG-13    :441               Oscar Qualifying Run:  3              3rd Qu.: 114.2   3rd Qu.:5.000
 for some language.            :   6   R        :448               Special Engagement  :  2              Max.   :2784.0   Max.   :9.000
 (Other)                       :1051                               Wide                :795
```

The scoring data has 91 rows.

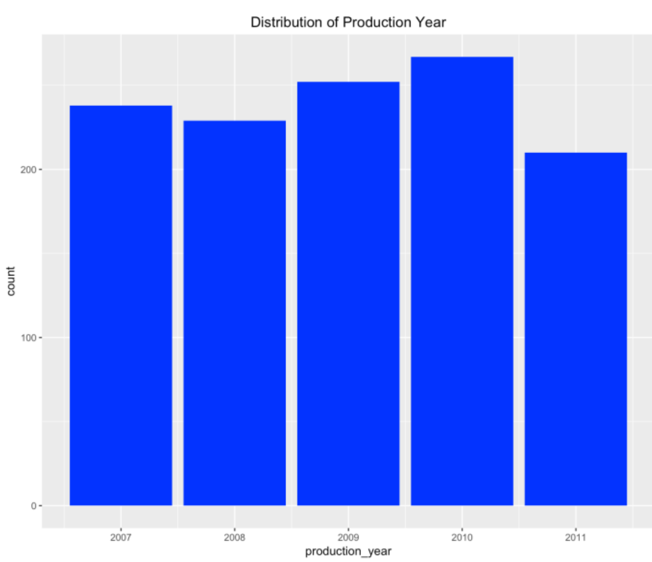
```
> scoring=read.csv('/Users/04pallav/Downloads/Scoring\ Sheet1.csv',header=TRUE)
> dim(scoring)
[1] 91 14
```

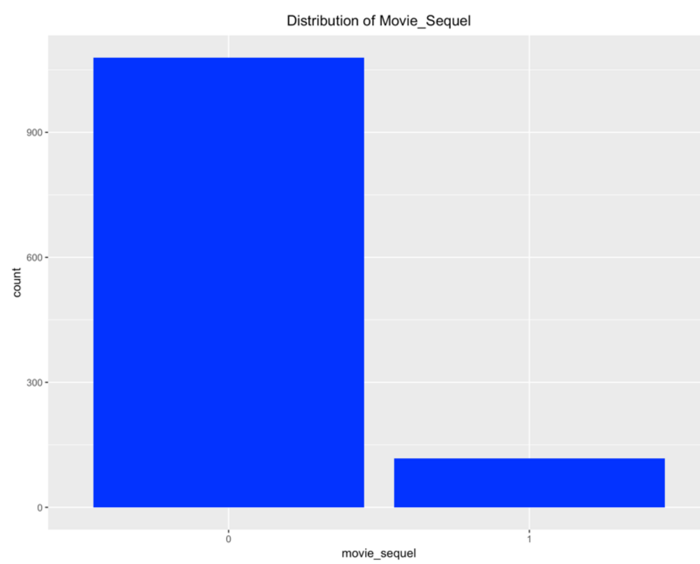
DATA EXPLORATION

Let us have a look at the distribution of the classes. We find that the distribution is right skewed and they are few movies belonging to the "Blockbuster" category.

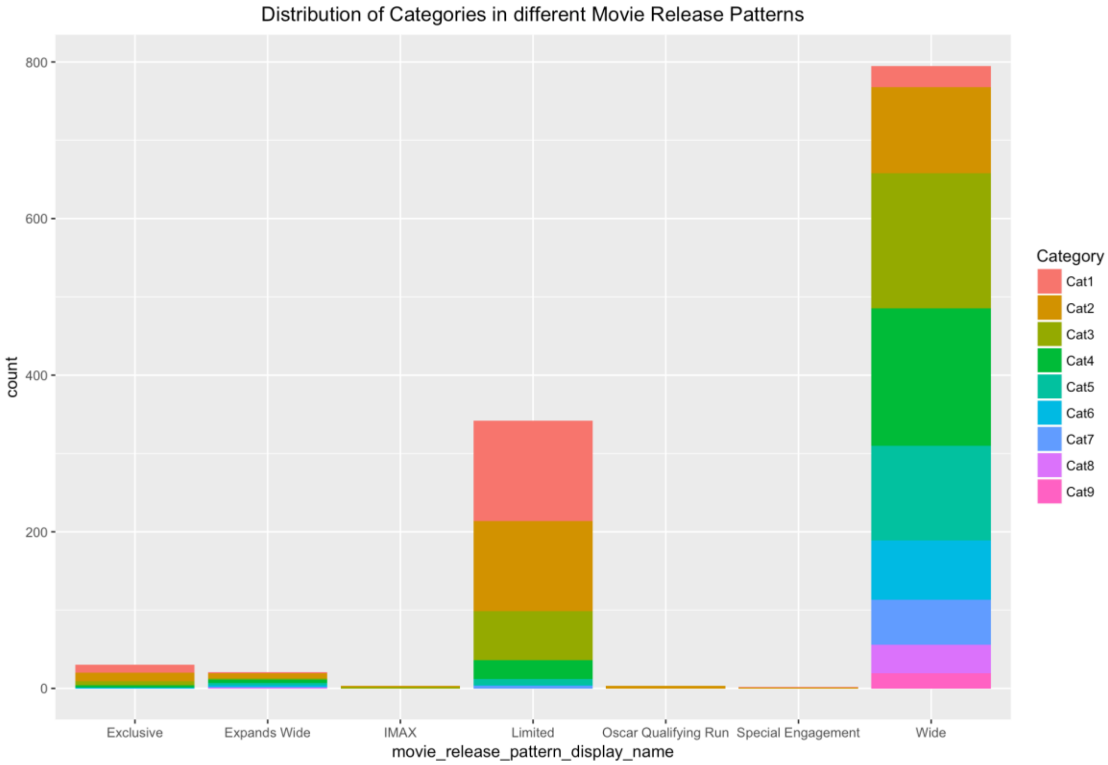


Exploration of features

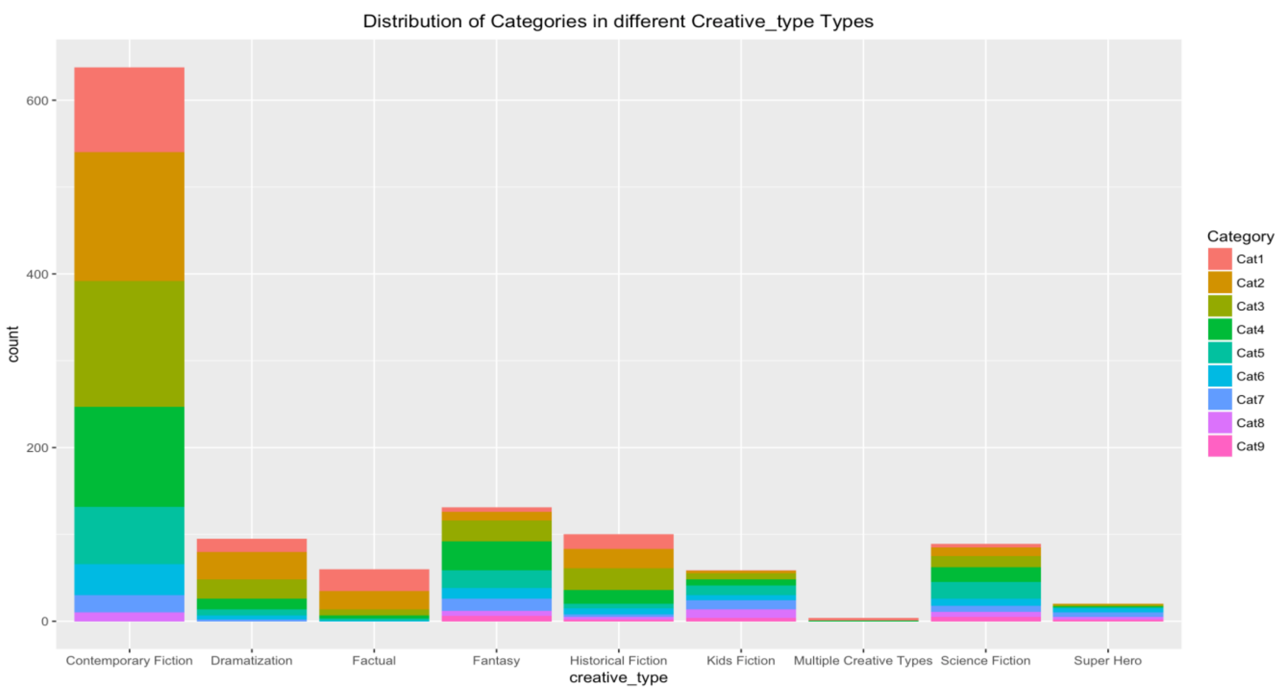




DATA EXPLORATION



We see that a lot of high Grossing movies are "Wide". It seems that this is an important predictor for our data. A lot of Category 1, Category2 and Category3 movies are "Limited"



We find that "Dramatization", "Factual" are some categories which don't have Cat8 and Cat9 movies.

CLASSIFICATION MODELS

Support Vector Machines

Using the "Caret" package, I train a SVM with a radial kernel. I am using **5-fold cross validation** And **tuning the parameters** to choose the best model.

Fitting the SVM

```
################################################### SVM Classification

ctrl <- trainControl(method = "repeatedcv",number=5,repeats =1,savePredictions = TRUE,classProbs = TRUE)
grid <- expand.grid(sigma = c(.01, .015, 0.2),
                    C = c(0.75, 0.9, 1, 1.1, 1.25))
mod_fitSVMRadial <- train(Category~production_year+movie_sequel+creative_type+source+production_method+genre+
                      language+movie_board_rating_display_name+movie_release_pattern_display_name,
                    data=train,method="svmRadial",trControl = ctrl,tuneGrid=grid)

mod_fitSVMRadial

pred = predict(mod_fitSVMRadial, newdata=test,type='raw')
```

Results

```
> mod_fitSVMRadial
Support Vector Machines with Radial Basis Function Kernel

960 samples
  9 predictor
  9 classes: 'Cat1', 'Cat2', 'Cat3', 'Cat4', 'Cat5', 'Cat6', 'Cat7', 'Cat8', 'Cat9'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 766, 769, 769, 768, 768
Resampling results across tuning parameters:

  sigma  C     Accuracy   Kappa
  0.010  0.75  0.2822358  0.1451166
  0.010  0.90  0.2926527  0.1558460
  0.010  1.00  0.2884267  0.1511340
  0.010  1.10  0.2832720  0.1435072
  0.010  1.25  0.2874443  0.1487639
  0.015  0.75  0.2843301  0.1451358
  0.015  0.90  0.2895277  0.1509106
  0.015  1.00  0.2811507  0.1389087
  0.015  1.10  0.2874227  0.1459888
  0.015  1.25  0.2853554  0.1411172
  0.200  0.75  0.2697624  0.1178955
  0.200  0.90  0.2791269  0.1290876
  0.200  1.00  0.2729469  0.1195801
  0.200  1.10  0.2677546  0.1133867
  0.200  1.25  0.2760288  0.1242038

Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.01 and C = 0.9.
```

Performance

```
> pred = predict(mod_fitSVMRadial, newdata=test,type='raw')
> confusionMatrix(pred, test$Category)$overall[1]
 Accuracy
0.2627119
> cm=table(pred,test$Category)
> d <- row(cm) - col(cm)
> away1error=sum(split(cm, d)$'0',split(cm, d)$'1',split(cm, d)$'-1')/sum(cm)
> away1error
[1] 0.6737288
```

Confusion Matrix SVM

Confusion Matrix and Statistics

	Reference								
Prediction	Cat1	Cat2	Cat3	Cat4	Cat5	Cat6	Cat7	Cat8	Cat9
Cat1	20	29	15	4	0	0	1	0	0
Cat2	5	2	2	1	0	0	1	0	0
Cat3	8	4	7	7	1	1	1	1	0
Cat4	0	12	23	27	22	13	5	1	0
Cat5	0	1	1	1	3	1	3	0	0
Cat6	0	0	1	1	0	1	0	2	1
Cat7	0	0	0	0	0	0	0	1	0
Cat8	0	0	0	0	0	0	0	0	0
Cat9	0	1	0	0	0	0	1	2	2

CLASSIFICATION MODELS

Random Forests

Next I fit random forest model for classification

Training the forest

```
############################################### Random Forests
set.seed(1)
 ctrl <- trainControl(method = "repeatedcv",number=5,repeats = 1,savePredictions = TRUE,classProbs = TRUE)
mod_fitRF <- train(Category~production_year+movie_sequel+creative_type+source+production_method+genre+
                     language1+movie_board_rating_display_name+movie_release_pattern_display_name,
                   data=train,method="rf",trControl = ctrl)
mod_fitRF
```

Results

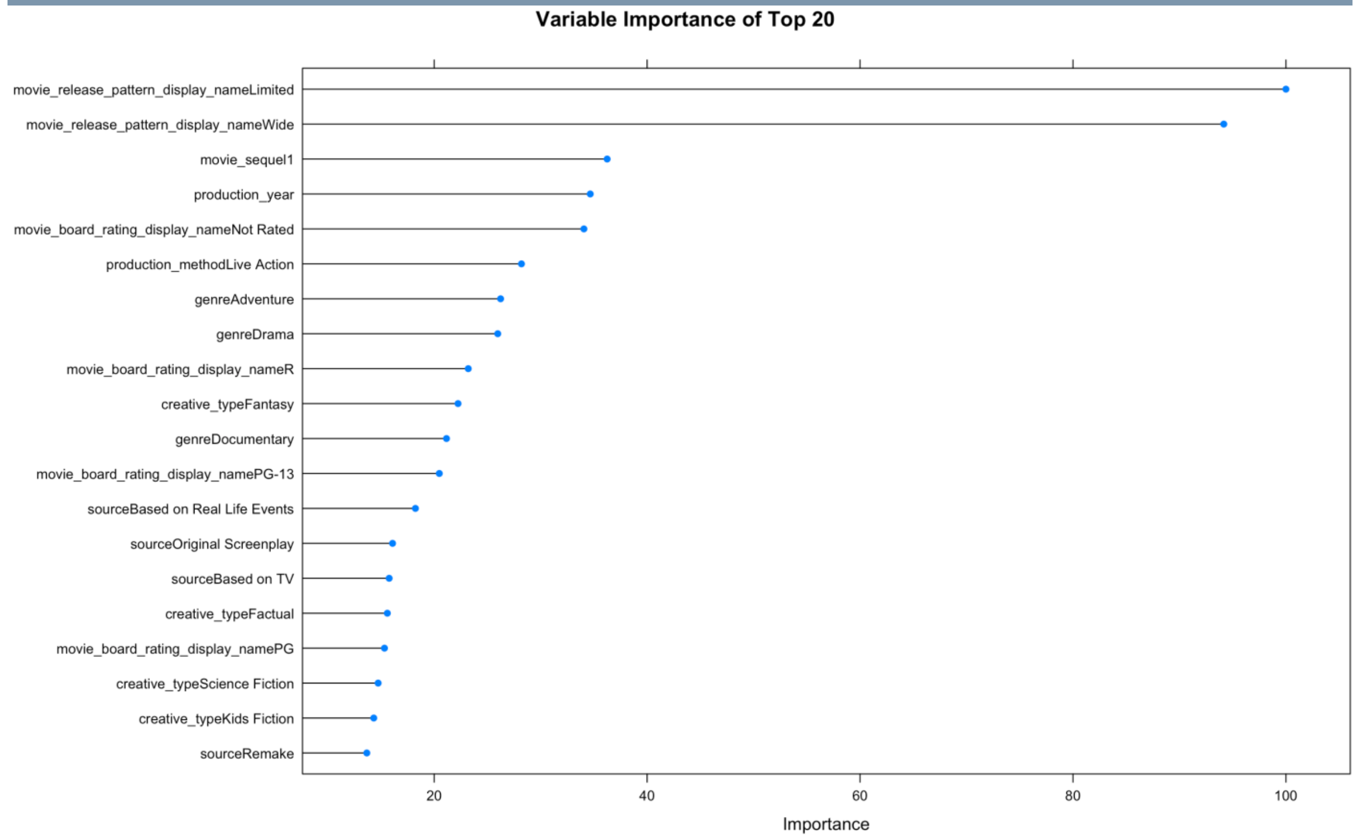
```
> mod_fitRF
Random Forest

960 samples
  9 predictor
  9 classes: 'Cat1', 'Cat2', 'Cat3', 'Cat4', 'Cat5', 'Cat6', 'Cat7', 'Cat8', 'Cat9'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 769, 768, 766, 769, 768
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.2781448  0.10724047
  28    0.2375078  0.09542081
  54    0.2468669  0.10754985

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was mtry = 2.
```

Performance

```
> confusionMatrix(pred, test$Category)$overall[1]
Accuracy
0.220339
```

1-Away Error

```
> away1error
[1] 0.6355932
```



As we had noticed earlier from exploratory analysis Movie_release_pattern_display_nameLimited and Wide are one of the most important variables!

Recursive Feature Elimination with Random Forests

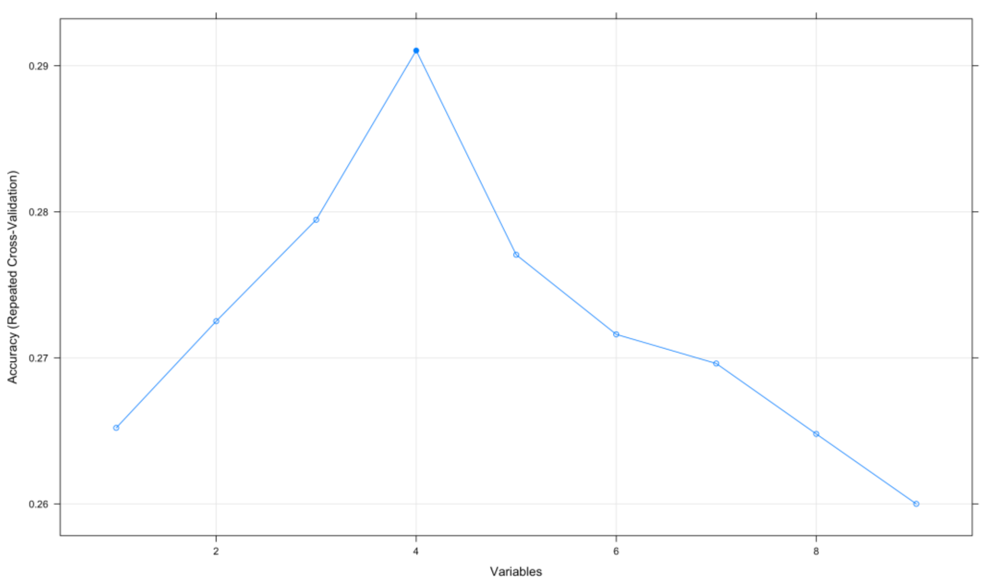
Recursive feature selection uses backward elimination to build models with different features

```
subsets <- c(1:20)
ctrl <- rfeControl(functions = rfFuncs,method = "repeatedcv",repeats =5,verbose = TRUE)
x=train[,c("production_year","movie_sequel","creative_type","source","production_method","genre",
           "language","movie_board_rating_display_name","movie_release_pattern_display_name")]
y=train[,"Category"]

rfProfile <- rfe(x, y,sizes = subsets, rfeControl = ctrl)

rfProfile
```

CLASSIFICATION MODELS

```
> rfProfile

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold, repeated 5 times)

Resampling performance over subset size:

 Variables Accuracy  Kappa AccuracySD KappaSD Selected
         1   0.2652 0.1013    0.02423 0.02773         
         2   0.2725 0.1205    0.02647 0.03213         
         3   0.2795 0.1311    0.02998 0.03646         
         4   0.2910 0.1505    0.03050 0.03693        *
         5   0.2771 0.1351    0.03122 0.03808         
         6   0.2716 0.1277    0.03518 0.04328         
         7   0.2696 0.1255    0.04013 0.04757         
         8   0.2648 0.1187    0.04168 0.05109         
         9   0.2600 0.1160    0.03593 0.04447         

The top 4 variables (out of 4):
   movie_release_pattern_display_name, movie_sequel, creative_type, production_method
```

We find that 4 variable model is giving us the best results on the training data



Test set results

```
> away1error
[1] 0.690678
> confusionMatrix(pred, test$Category)$overall[1]
 Accuracy
0.2754237
```

I get an Accuracy of 27.5% on the test data and an one-away error of 69.06%

Gradient Boosting Machine Classifier

We can use boosting methods to build a classifier. gbm package in R provides a way to do this.

Training XGB Classifier

```
ctrl <- trainControl(method = "repeatedcv",number=5,repeats = 3,savePredictions = TRUE,classProbs = TRUE,verboseIter=TRUE)
mod_fitGBM <- train(Category~production_year+movie_sequel+creative_type+source+production_method+genre+language+
                      movie_board_rating_display_name+movie_release_pattern_display_name,
                    data=train,method="gbm",trControl = ctrl,verbose=TRUE)

mod_fitGBM
```

Results

```
> mod_fitGBM
Stochastic Gradient Boosting

960 samples
  9 predictor
  9 classes: 'Cat1', 'Cat2', 'Cat3', 'Cat4', 'Cat5', 'Cat6', 'Cat7', 'Cat8', 'Cat9'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 769, 770, 767, 766, 768, 768, ...
Resampling results across tuning parameters:

  interaction.depth  n.trees  Accuracy   Kappa
  1                   50      0.2819746  0.1361414
  1                  100      0.2812603  0.1372830
  1                  150      0.2784773  0.1346027
  2                   50      0.2882195  0.1463592
  2                  100      0.2795259  0.1389240
  2                  150      0.2767480  0.1355305
  3                   50      0.2805929  0.1388398
  3                  100      0.2732265  0.1329410
  3                  150      0.2735792  0.1340820

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Tuning parameter 'n.minobsinnode' was held constant at a value of 10
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were n.trees = 50, interaction.depth = 2, shrinkage = 0.1 and n.minobsinnode = 10.
```

CLASSIFICATION MODELS

Performance

```
> confusionMatrix(pred, test$Category)$overall[1]
 Accuracy
0.2881356
> away1error
[1] 0.6949153
```

Accuracy is 28.8% and One-Away Accuracy is 69.49%

CLASSIFICATION MODELS

Neural Nets Classifier

Training the NN Classifier

```
ctrl <- trainControl(method = "repeatedcv",number=5,repeats = 3,savePredictions = TRUE,classProbs = TRUE,verboseIter=TRUE)
mod_fitNN <- train(Category~production_year+movie_sequel+creative_type+source+production_method+genre+language+
                      movie_board_rating_display_name+movie_release_pattern_display_name,
                    data=train,method="nnet",trControl = ctrl,verbose=TRUE)
```

Results

```
> mod_fitNN
Neural Network

960 samples
  9 predictor
  9 classes: 'Cat1', 'Cat2', 'Cat3', 'Cat4', 'Cat5', 'Cat6', 'Cat7', 'Cat8', 'Cat9'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 769, 768, 767, 769, 767, 768, ...
Resampling results across tuning parameters:

  size  decay  Accuracy   Kappa
  1     0e+00  0.2062544  0.00000000
  1     1e-04  0.2062544  0.00000000
  1     1e-01  0.2506346  0.07803467
  3     0e+00  0.2062544  0.00000000
  3     1e-04  0.2062544  0.00000000
  3     1e-01  0.2656275  0.10642617
  5     0e+00  0.2062544  0.00000000
  5     1e-04  0.2062544  0.00000000
  5     1e-01  0.2982903  0.15991664

Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were size = 5 and decay = 0.1.
```

Performance

```
> confusionMatrix(pred, test$Category)$overall[1]
 Accuracy
0.2754237
> away1error
[1] 0.690678
```

MODEL COMPARISION

Comparison of different Classifiers

Classifier	Bingo Accuracy	One Away Accuracy
Support Vector Machines	26.27%	67.37
Random Forests	27.5%	69.06%
Gradient Boosting	28.8%	69.49%
Neural Net	27.54%	69.06%

I see that boosting gives me the best results for classification.

Scoring data

I used the gradient boosting classifier to score the final data.

```
scoring$movie_sequel=as.factor(scoring$movie_sequel)
ctrl <- trainControl(method = "repeatedcv",number=5,repeats = 3,savePredictions = TRUE,classProbs = TRUE,verboseIter=TRUE)
mod_fitGBM2 <- train(Category~production_year+movie_sequel+creative_type+source+production_method+genre+
                       movie_board_rating_display_name+movie_release_pattern_display_name,
                     data=mydf,method="gbm",trControl = ctrl,verbose=TRUE)

mod_fitGBM2
summary(mod_fitGBM2)


pred = predict(mod_fitGBM2, newdata=scoring,type='raw')
```

LOOKING AHEAD

There could be some improvements which could be done.

1. Ensemble models combining all these different models could be created
2. Better predictors like "Star Value" and "Competition" which are used in the paper but not provided in the data can help us to increase our accuracy rates.
3. Natural Language Techniques could be used to extract features from the column "board_rating_reason"