R Notebook

# The following project was completed to determine the likelihood of a home run every instance of a ball in play given a number of metrics. I chose to use a Random Forest model as it creates a ‘forest’ of decision trees where the output is the class selecting the most trees which removes bias within the data. Random Forest is one of the strongest predictive models that one can create and given the amount of data within the data set, I deemed it too great for that of a single decsion tree to handle.

# Below are the packages I used to create this model.

require(dplyr)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

require(randomForest)

## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

require(ggplot2)

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':  
##   
## margin

require(ROCR)

## Loading required package: ROCR

require(corrplot)

## Loading required package: corrplot

## corrplot 0.84 loaded

require(pROC)

## Loading required package: pROC

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(dplyr)  
library(randomForest)  
library(ggplot2)  
library(ROCR)  
library(corrplot)  
library(pROC)

# I first uploaded the data set in RStudio and checked its summary and structure to gain a better understanding of the data. I noticed that there were 98 NA values within the data set. This consitituted 0.1% of the data set so I deemed it prudent to omit them. I noticed two character attributes that I wanted to obtain a better understanding of the data within them. I decided to see the unique values within each and noticed they could definitely play a factor in predicting a home run.

url <- "https://raw.githubusercontent.com/Chrisboatto/Likelihood-of-a-Homerun/main/Balls%20in%20play.csv"

BallsInPlay <- read.csv(url)

str(BallsInPlay)

## 'data.frame': 61413 obs. of 40 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ GAME\_ID : int 310 375 17 541 17 1175 569 446 30 462 ...  
## $ DAY\_OR\_NIGHT : chr "N" "N" "N" "N" ...  
## $ TEMPERATURE : int 66 76 71 72 71 70 81 69 65 71 ...  
## $ WEATHER : chr "Cloudy" "Cloudy" "Clear" "Clear" ...  
## $ WIND\_SPEED : int 8 8 10 6 10 6 3 7 10 8 ...  
## $ WIND\_DIRECTION : chr "Out To CF" "R To L" "Out To CF" "Out To CF" ...  
## $ PLAY\_ID : int 14861 18009 792 26000 793 57702 27306 21297 1380 22106 ...  
## $ PITCHER\_ID : int 97 610 38 86 38 75 651 623 836 333 ...  
## $ PITCHER\_SIDE : chr "R" "L" "R" "R" ...  
## $ BATTER\_ID : int 4 8 8 8 20 30 35 37 37 37 ...  
## $ BATTER\_SIDE : chr "R" "R" "R" "R" ...  
## $ STADIUM\_ID : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ INNING : int 6 8 4 2 4 3 4 5 2 7 ...  
## $ INNING\_HALF : chr "Y" "N" "N" "N" ...  
## $ OUTS : int 2 0 0 2 2 2 1 1 0 0 ...  
## $ BALLS : int 1 1 2 3 3 3 0 1 0 1 ...  
## $ STRIKES : int 2 0 1 1 2 2 1 2 2 1 ...  
## $ PITCH\_TYPE : chr "FB" "SI" "CF" "SI" ...  
## $ PITCH\_VELOCITY : num 89.3 92.7 89.2 94.1 94.1 ...  
## $ HORIZONTAL\_BREAK : num -9.58 13.3 4.53 -13.57 -3.4 ...  
## $ INDUCED\_VERTICAL\_BREAK : num 12.59 13.17 7.01 19.38 13.72 ...  
## $ RELEASE\_SIDE : num -3.43 1.6 -3.24 -1.22 -3.09 ...  
## $ RELEASE\_HEIGHT : num 4.22 6.64 5.71 6.64 5.63 ...  
## $ RELEASE\_EXTENSION : num 7.47 5.94 6.12 5.81 6.43 ...  
## $ PITCH\_LOCATION\_SIDE : num -0.1474 0.0378 0.2738 -0.5043 0.7115 ...  
## $ PITCH\_LOCATION\_HEIGHT : num 2.31 1.87 2.9 1.83 2.33 ...  
## $ STRIKE\_ZONE\_BOTTOM : num 1.49 1.51 1.51 1.51 1.67 ...  
## $ STRIKE\_ZONE\_TOP : num 3.27 3.32 3.32 3.32 3.61 ...  
## $ EXIT\_VELOCITY : num 76.7 100.8 65.9 80.6 80.1 ...  
## $ LAUNCH\_ANGLE : num 58.91 -9.28 -24.91 38.36 2.18 ...  
## $ LAUNCH\_DIRECTION : num 33.9 -29.2 -34.7 -22.1 38 ...  
## $ HORIZONTAL\_APPROACH\_ANGLE: num 2.705 -0.429 4.207 -0.526 3.778 ...  
## $ VERTICAL\_APPROACH\_ANGLE : num -4.2 -6.99 -5.67 -6.4 -5.25 ...  
## $ PITCH\_RESULT : chr "InPlay" "InPlay" "InPlay" "InPlay" ...  
## $ EVENT\_RESULT : chr "field\_out" "single" "grounded\_into\_double\_play" "field\_out" ...  
## $ SINGLE\_YES\_OR\_NO : int 0 1 0 0 0 0 1 0 0 0 ...  
## $ DOUBLE\_YES\_OR\_NO : int 0 0 0 0 0 0 0 0 0 1 ...  
## $ TRIPLE\_YES\_OR\_NO : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ HOME\_RUN\_YES\_OR\_NO : int 0 0 0 0 0 0 0 0 0 0 ...

summary(BallsInPlay)

## X GAME\_ID DAY\_OR\_NIGHT TEMPERATURE   
## Min. : 1 Min. : 1 Length:61413 Min. : 32.00   
## 1st Qu.:15354 1st Qu.: 321 Class :character 1st Qu.: 68.00   
## Median :30707 Median : 638 Mode :character Median : 73.00   
## Mean :30707 Mean : 633 Mean : 73.16   
## 3rd Qu.:46060 3rd Qu.: 946 3rd Qu.: 81.00   
## Max. :61413 Max. :1248 Max. :102.00   
##   
## WEATHER WIND\_SPEED WIND\_DIRECTION PLAY\_ID   
## Length:61413 Min. : 0.000 Length:61413 Min. : 1   
## Class :character 1st Qu.: 4.000 Class :character 1st Qu.:15354   
## Mode :character Median : 7.000 Mode :character Median :30707   
## Mean : 7.042 Mean :30707   
## 3rd Qu.:10.000 3rd Qu.:46060   
## Max. :25.000 Max. :61413   
##   
## PITCHER\_ID PITCHER\_SIDE BATTER\_ID BATTER\_SIDE   
## Min. : 1.0 Length:61413 Min. : 1.0 Length:61413   
## 1st Qu.:179.0 Class :character 1st Qu.:183.0 Class :character   
## Median :382.0 Mode :character Median :381.0 Mode :character   
## Mean :389.6 Mean :393.4   
## 3rd Qu.:586.0 3rd Qu.:587.0   
## Max. :848.0 Max. :843.0   
##   
## STADIUM\_ID INNING INNING\_HALF OUTS   
## Min. : 1.00 Min. : 1.000 Length:61413 Min. :0.0000   
## 1st Qu.: 8.00 1st Qu.: 3.000 Class :character 1st Qu.:0.0000   
## Median :16.00 Median : 5.000 Mode :character Median :1.0000   
## Mean :16.43 Mean : 4.886 Mean :0.9628   
## 3rd Qu.:25.00 3rd Qu.: 7.000 3rd Qu.:2.0000   
## Max. :32.00 Max. :16.000 Max. :2.0000   
##   
## BALLS STRIKES PITCH\_TYPE PITCH\_VELOCITY   
## Min. :0.000 Min. :0.000 Length:61413 Min. : 32.97   
## 1st Qu.:0.000 1st Qu.:0.000 Class :character 1st Qu.: 84.70   
## Median :1.000 Median :1.000 Mode :character Median : 89.74   
## Mean :1.093 Mean :1.103 Mean : 88.76   
## 3rd Qu.:2.000 3rd Qu.:2.000 3rd Qu.: 93.52   
## Max. :3.000 Max. :2.000 Max. :101.99   
##   
## HORIZONTAL\_BREAK INDUCED\_VERTICAL\_BREAK RELEASE\_SIDE RELEASE\_HEIGHT   
## Min. :-27.358 Min. :-25.907 Min. :-4.5116 Min. :1.048   
## 1st Qu.:-11.148 1st Qu.: 2.292 1st Qu.:-2.0694 1st Qu.:5.632   
## Median : -2.543 Median : 8.894 Median :-1.4369 Median :5.916   
## Mean : -1.552 Mean : 7.569 Mean :-0.7172 Mean :5.877   
## 3rd Qu.: 7.328 3rd Qu.: 14.925 3rd Qu.: 1.2371 3rd Qu.:6.203   
## Max. : 55.700 Max. : 46.057 Max. : 4.4699 Max. :7.567   
##   
## RELEASE\_EXTENSION PITCH\_LOCATION\_SIDE PITCH\_LOCATION\_HEIGHT  
## Min. :3.353 Min. :-2.1004701 Min. :0.02045   
## 1st Qu.:5.858 1st Qu.:-0.3878030 1st Qu.:1.94587   
## Median :6.147 Median :-0.0019161 Median :2.33972   
## Mean :6.156 Mean : 0.0000245 Mean :2.34947   
## 3rd Qu.:6.446 3rd Qu.: 0.3848440 3rd Qu.:2.74306   
## Max. :8.208 Max. : 2.9690700 Max. :6.21727   
## NA's :68   
## STRIKE\_ZONE\_BOTTOM STRIKE\_ZONE\_TOP EXIT\_VELOCITY LAUNCH\_ANGLE   
## Min. :1.387 Min. :3.075 Min. : 9.333 Min. :-88.067   
## 1st Qu.:1.514 1st Qu.:3.318 1st Qu.: 80.207 1st Qu.: -5.182   
## Median :1.565 Median :3.415 Median : 91.369 Median : 12.932   
## Mean :1.562 Mean :3.410 Mean : 88.800 Mean : 12.493   
## 3rd Qu.:1.591 3rd Qu.:3.464 3rd Qu.: 99.498 3rd Qu.: 30.674   
## Max. :1.768 Max. :3.804 Max. :122.214 Max. : 89.113   
## NA's :15 NA's :15   
## LAUNCH\_DIRECTION HORIZONTAL\_APPROACH\_ANGLE VERTICAL\_APPROACH\_ANGLE  
## Min. :-179.810 Min. :-7.1321 Min. :-27.9678   
## 1st Qu.: -18.590 1st Qu.:-0.5944 1st Qu.: -7.4518   
## Median : -1.674 Median : 0.6888 Median : -6.2681   
## Mean : -1.207 Mean : 0.6290 Mean : -6.4744   
## 3rd Qu.: 15.526 3rd Qu.: 2.0050 3rd Qu.: -5.3037   
## Max. : 179.157 Max. : 7.4695 Max. : -0.2186   
##   
## PITCH\_RESULT EVENT\_RESULT SINGLE\_YES\_OR\_NO DOUBLE\_YES\_OR\_NO   
## Length:61413 Length:61413 Min. :0.0000 Min. :0.00000   
## Class :character Class :character 1st Qu.:0.0000 1st Qu.:0.00000   
## Mode :character Mode :character Median :0.0000 Median :0.00000   
## Mean :0.2049 Mean :0.06549   
## 3rd Qu.:0.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.00000   
##   
## TRIPLE\_YES\_OR\_NO HOME\_RUN\_YES\_OR\_NO  
## Min. :0.000000 Min. :0.00000   
## 1st Qu.:0.000000 1st Qu.:0.00000   
## Median :0.000000 Median :0.00000   
## Mean :0.005357 Mean :0.04975   
## 3rd Qu.:0.000000 3rd Qu.:0.00000   
## Max. :1.000000 Max. :1.00000   
##

sum(is.na(BallsInPlay))

## [1] 98

BallsInPlay <- na.omit(BallsInPlay)

sum(is.na(BallsInPlay))

## [1] 0

unique(BallsInPlay$WEATHER)

## [1] "Cloudy" "Clear" "Sunny" "Partly Cloudy"  
## [5] "Overcast" "Drizzle" "Rain" "Dome"   
## [9] "Roof Closed" "Snow"

unique(BallsInPlay$WIND\_DIRECTION)

## [1] "Out To CF" "R To L" "Out To LF" "Out To RF" "In From RF"  
## [6] "Varies" "In From CF" "Calm" "In From LF" "L To R"   
## [11] "None"

# I created the lolipop box plots below to gain a better understanding of the interquartile ranges amongst the numeric metrics and where the outliers are. Notice the two outliers amongst the break attributes. Both those outliers were on one single pitch, a fastball. This does not make sense as fastball are supposed to be almost as straight as an arrow. Having a break on your fastball over 40" is not normal and should be seen as an outlier. I used these box plots as a means to subset out any outliers within the data. I removed the two large breaks and and velocity seen under 65mph. Even though the first interquartile range starts at 84.7mph, I deemed it prudent to keep any pitches over 65mph within the data set simply because some pitchers have curveballs that are in the low 70’s. Therefore, setting the subset to 65mph would ensure all curveballs were included. Any pitch recorded below that could be deemed a ‘lob’ from a position player.

boxplot(BallsInPlay[20:25], col = rainbow(14), main = "Box Plot of Pitch Type Metrics", xlab = "Pitch Type Metrics", ylab = "Scores")

Chart, box and whisker chart

Description automatically generated

boxplot(BallsInPlay[26:29], col = rainbow(14), main = "Box Plot of Pitch Location Metrics", xlab = "Pitch Location Metrics", ylab = "Scores")

Chart, box and whisker chart

Description automatically generated

boxplot(BallsInPlay[30:34], col = rainbow(14), main = "Box Plot of Hitting Metrics", xlab = "Categories", ylab = "Scores")

Chart, box and whisker chart

Description automatically generated

BallsInPlay <- filter(BallsInPlay, PITCH\_VELOCITY > 65)  
BallsInPlay <- filter(BallsInPlay, HORIZONTAL\_BREAK < 40)  
BallsInPlay <- filter(BallsInPlay, INDUCED\_VERTICAL\_BREAK < 40)

# Below is a correlation plot depicting how well each metric is correlated to all others within the data set. The deeper the blue, the more positively correlated the metrics are and the deeper the red, the more negatively correlated.

BallsInPlay\_Cor <- cor(BallsInPlay[20:34])  
corrplot(BallsInPlay\_Cor, type = "upper", order = 'hclust', tl.col = "blue")

Chart, scatter chart

Description automatically generated

# I then trained the data on an 80:20 train to test ratio based on random samples to avoid any bias. Making the ratios split at random is a key factor as I wanted to remove as much bias as possible throughout my entire model creation. I wanted to produce the most fair and unbiased model possible that way I can ensure the highest level of accuracy that way I know my model is correct.

set.seed(49838)  
  
train <- sample(nrow(BallsInPlay), 0.80\*nrow(BallsInPlay), replace = FALSE)  
  
TrainSet <- BallsInPlay[train,]  
TestSet <- BallsInPlay[-train,]

# I then created a base model using the ‘HOME\_RUN\_YES\_OR\_NO’ variable as my dependent and using all others as my independents. This base model will allow me to gain an understanding of what metrics have the highest importance within the model. The importance metric shows how much of an influence the attribute has on the output. If the metric is too high then it could lead to domination thus resulting in an overfit. As you can see below, ‘EVENT\_RESULT’ had the most influence on the dependent variable by a wide margin which is understandable as it shows results of the ball in play much like that of the binary event attributes do. This therefore shold be removed as it is redundant.

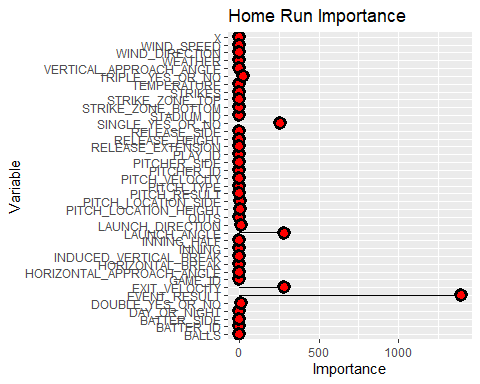
baseModel <- randomForest(HOME\_RUN\_YES\_OR\_NO ~., data = TrainSet, importance = TRUE, ntrees = 50)

## Warning in randomForest.default(m, y, ...): The response has five or fewer  
## unique values. Are you sure you want to do regression?

baseImp <- importance(baseModel)  
baseImp

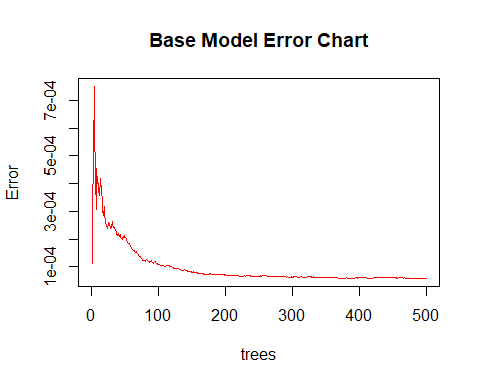
## %IncMSE IncNodePurity  
## X 5.0099746 1.001653e+00  
## GAME\_ID 2.2426579 6.844384e-01  
## DAY\_OR\_NIGHT -0.1894442 1.158529e-01  
## TEMPERATURE 4.3499565 1.074667e+00  
## WEATHER 4.0348361 2.813192e-01  
## WIND\_SPEED 10.7991493 5.452112e-01  
## WIND\_DIRECTION 5.6235893 5.109592e-01  
## PLAY\_ID 2.3282199 8.890891e-01  
## PITCHER\_ID 1.3599197 9.242732e-01  
## PITCHER\_SIDE 2.7185961 2.612332e-02  
## BATTER\_ID 0.5402919 1.200263e+00  
## BATTER\_SIDE 3.3564378 1.089851e+00  
## STADIUM\_ID 3.0980300 5.499674e-01  
## INNING 0.3389796 5.443639e-01  
## INNING\_HALF 1.0987845 6.170041e-02  
## OUTS 7.5286510 1.989748e+00  
## BALLS 2.4009976 1.903504e-01  
## STRIKES 1.6635361 1.422362e-01  
## PITCH\_TYPE 5.0986062 3.517097e-01  
## PITCH\_VELOCITY 7.2644912 1.213967e+00  
## HORIZONTAL\_BREAK 7.7761436 1.182527e+00  
## INDUCED\_VERTICAL\_BREAK 13.6051647 1.309952e+00  
## RELEASE\_SIDE 11.7257556 1.143417e+00  
## RELEASE\_HEIGHT 2.9526899 1.030018e+00  
## RELEASE\_EXTENSION 2.7237389 1.251302e+00  
## PITCH\_LOCATION\_SIDE 7.0784357 2.754237e+00  
## PITCH\_LOCATION\_HEIGHT 4.4519758 3.943158e+00  
## STRIKE\_ZONE\_BOTTOM 1.2898976 5.145010e-01  
## STRIKE\_ZONE\_TOP 0.5228000 5.015621e-01  
## EXIT\_VELOCITY 12.6758795 2.832289e+02  
## LAUNCH\_ANGLE 21.0280766 2.820337e+02  
## LAUNCH\_DIRECTION 11.1615282 1.140947e+01  
## HORIZONTAL\_APPROACH\_ANGLE 12.7383092 1.202245e+00  
## VERTICAL\_APPROACH\_ANGLE 5.9354942 1.130160e+00  
## PITCH\_RESULT 0.0000000 0.000000e+00  
## EVENT\_RESULT 127.3750794 1.394652e+03  
## SINGLE\_YES\_OR\_NO 19.0088381 2.552967e+02  
## DOUBLE\_YES\_OR\_NO 12.2993650 1.385281e+01  
## TRIPLE\_YES\_OR\_NO 12.4898696 2.548329e+01

baseImp <- as.data.frame(baseImp)  
ggplot(baseImp, aes(IncNodePurity, row.names(baseImp))) +   
 geom\_bar(stat = "identity", width = 0.1, fill = "black") +   
 geom\_point(shape = 21, size = 3, colour = "black", fill = "red", stroke = 2) +   
 labs(title = "Home Run Importance", x = "Importance", y = "Variable")



# I then created an error chart to view how well the model removed bias from its output. Judging by the chart below, the base model worked well at reducing the chance of errors.

plot(baseModel, col = "Red", main = "Base Model Error Chart")



# I then created a new model on the same dependent variable but by removing the aforementioned ‘EVENT\_RESULT’ column as well others that were not deemed to have an affect on the outcome. I stated the number of nodes to 5 and the number of trees at 100 to obtain an Area Under the Curve (AUC) accuracy metric of over 0.9 which can be found later in the assignment. After many attempts at tuning and pruning, those were the two amounts that I found gave me the best accuracy ratings.

# Notice for this model ‘EXIT\_VELOCITY’ and ‘LAUNCH\_ANGLE’ were the main contributors to home runs. The harder you hit the ball and the higher it’s initial trajectory is, the further the baseball will travel. Players that are able to do both at a high rate will make them highly likely to hit home runs.

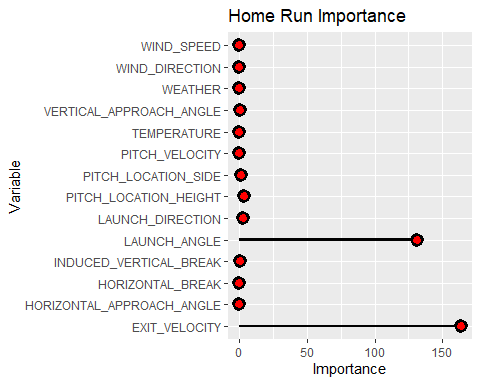
newModel <- randomForest(HOME\_RUN\_YES\_OR\_NO ~ WIND\_DIRECTION + WIND\_SPEED + WEATHER + PITCH\_LOCATION\_HEIGHT + PITCH\_LOCATION\_SIDE + PITCH\_VELOCITY + EXIT\_VELOCITY + LAUNCH\_ANGLE + LAUNCH\_DIRECTION + TEMPERATURE + VERTICAL\_APPROACH\_ANGLE + HORIZONTAL\_APPROACH\_ANGLE + HORIZONTAL\_BREAK + INDUCED\_VERTICAL\_BREAK, data = TrainSet, importance = TRUE, ntrees = 100, maxnodes = 5)

## Warning in randomForest.default(m, y, ...): The response has five or fewer  
## unique values. Are you sure you want to do regression?

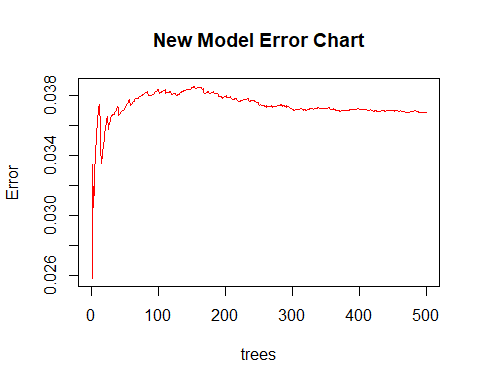
newImp <- importance(newModel)  
newImp

## %IncMSE IncNodePurity  
## WIND\_DIRECTION -0.9470921 3.520130e-03  
## WIND\_SPEED -0.7289524 1.625344e-02  
## WEATHER -1.0444884 1.566758e-04  
## PITCH\_LOCATION\_HEIGHT 9.4186393 3.262207e+00  
## PITCH\_LOCATION\_SIDE 7.7215793 1.261528e+00  
## PITCH\_VELOCITY 3.4522172 1.561907e-01  
## EXIT\_VELOCITY 15.9453040 1.639369e+02  
## LAUNCH\_ANGLE 13.9572448 1.313382e+02  
## LAUNCH\_DIRECTION 11.7503018 3.015155e+00  
## TEMPERATURE -0.5284109 1.386636e-02  
## VERTICAL\_APPROACH\_ANGLE 1.2558494 2.762046e-01  
## HORIZONTAL\_APPROACH\_ANGLE 2.9779239 1.274030e-01  
## HORIZONTAL\_BREAK 1.7137280 1.689273e-01  
## INDUCED\_VERTICAL\_BREAK 2.9141044 4.152442e-01

newImp <- as.data.frame(newImp)  
ggplot(newImp, aes(IncNodePurity, row.names(newImp))) +   
 geom\_bar(stat = "identity", width = 0.1, fill = "black") +   
 geom\_point(shape = 21, size = 3, colour = "black", fill = "red", stroke = 2) +   
 labs(title = "Home Run Importance", x = "Importance", y = "Variable")



plot(newModel, col = "red", main = "New Model Error Chart")



# I used the ‘response’ method to predict the probability of each outcome being a home run. I wanted a numerical result for each observation rather than a value being assigned by using ‘class’. I then bound the scores onto their respective sets that I predicted them from and changed the predictive column names to match each other that way I am then able to bind both the Train and Test sets back together to create one full data set. This will allow me to see all the predictions within one data set.

predTrainSet <- predict(newModel, TrainSet, type = "response")  
predTestSet <- predict(newModel, TestSet, type = "response")

TrainSet <- cbind(TrainSet, predTrainSet)  
TestSet <- cbind(TestSet, predTestSet)

names(TrainSet)[names(TrainSet) == "predTrainSet"] <- "HR\_Pred"  
names(TestSet)[names(TestSet) == "predTestSet"] <- "HR\_Pred"

HR\_Full <- rbind(TrainSet, TestSet)

# I created a Receiver Operator Characteristic (ROC) Curve and calculated the AUC below to show how well the model performed. Judging by the findings, the model performed excellently as the AUC was tabulated at 0.9555 giving a 95% accuracy rating while the ROC had an almost perfect curve to the top left corner showing that the model’s supervised learning worked well.

roc\_test <- roc(ifelse(TestSet$HOME\_RUN\_YES\_OR\_NO == "1", "1", "0"), as.numeric(TestSet$HR\_Pred))

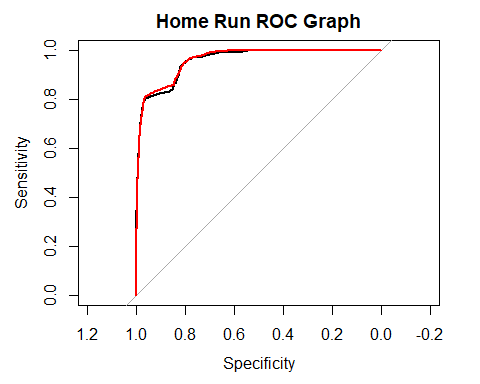
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

roc\_train <- roc(ifelse(TrainSet$HOME\_RUN\_YES\_OR\_NO == "1", "1", "0"), as.numeric(TrainSet$HR\_Pred))

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

plot(roc\_test, col = "black", main = "Home Run ROC Graph")  
lines(roc\_train, col = "red")



auc(HR\_Full$HOME\_RUN\_YES\_OR\_NO, HR\_Full$HR\_Pred)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Area under the curve: 0.9593

HR\_Full <- select(HR\_Full, c(X, HOME\_RUN\_YES\_OR\_NO, HR\_Pred, EXIT\_VELOCITY, LAUNCH\_ANGLE))  
Least\_Likely\_HR <- filter(HR\_Full, HOME\_RUN\_YES\_OR\_NO == 1)  
Least\_Likely\_HR <- as.data.frame(Least\_Likely\_HR[order(Least\_Likely\_HR$HR\_Pred),])  
Least\_Likely\_HR

Most\_Likely\_HR <- filter(HR\_Full, HOME\_RUN\_YES\_OR\_NO == 0)  
Most\_Likely\_HR <- as.data.frame(Most\_Likely\_HR[order(Most\_Likely\_HR$HR\_Pred),])  
Most\_Likely\_HR