NBA Project Final Draft

Briefly describe your data: The data that I collected is from kaggle.com and is csv file that describes every shot that was taken during the 2014-2015 NBA season. This dataset includes variables whether the shot was made, how far the player was away from the hoop, whether the shot was worth 2 points or 3 points as well as how much time was on the shot clock

My research questions are the following

1. Which variables will have the most impact on whether a shot will go in the basket?
2. is a random forest model a good model to use to predicting whether a shot will go in? Why or Why not? If not, explain what model you would use instead and why?

explain random forest model and demo it The random forest model is a supervised learning algorithm that randomly craetes and merges multiple decision trees into on forest. The idea is not to rely on a single learning model, but a collection of decision models to improve accuracy.

## Warning: package 'TH.data' was built under R version 3.6.3

## Loading required package: survival

## Loading required package: MASS

##   
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':  
##   
## geyser

## Warning: package 'rpart' was built under R version 3.6.3

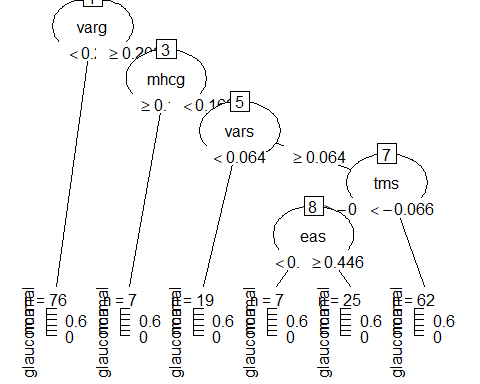
## Warning: package 'partykit' was built under R version 3.6.3

## Loading required package: grid

## Loading required package: libcoin

## Warning: package 'libcoin' was built under R version 3.6.3

## Loading required package: mvtnorm



##   
## Classification tree:  
## rpart(formula = Class ~ ., data = GlaucomaM)  
##   
## Variables actually used in tree construction:  
## [1] eas mhcg tms varg vars  
##   
## Root node error: 98/196 = 0.5  
##   
## n= 196   
##   
## CP nsplit rel error xerror xstd  
## 1 0.653061 0 1.00000 1.17347 0.070346  
## 2 0.071429 1 0.34694 0.37755 0.055904  
## 3 0.013605 2 0.27551 0.42857 0.058618  
## 4 0.010000 5 0.23469 0.43878 0.059119

## Warning: package 'randomForest' was built under R version 3.6.3

## randomForest 4.6-14

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

##   
## Call:  
## randomForest(formula = Class ~ ., data = GlaucomaM)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 7  
##   
## OOB estimate of error rate: 14.8%  
## Confusion matrix:  
## glaucoma normal class.error  
## glaucoma 80 18 0.1836735  
## normal 11 87 0.1122449

##   
## glaucoma normal  
## glaucoma 80 11  
## normal 18 87

## glaucoma   
## 82

## normal   
## 89

step 1 Part 1: exploatory data analysis cleaning data I noticed that there a lot of NA values for shot clock, considnering that this is bad data, I will filter out all null values in the dataset I cleaned the dataset to remove 6000 data points that were no good as they had an NA value for shot clock

Step 2: Visualizations See the bottom for explainations

## [1] "C:/Users/sdcha/Desktop/SDSU/Stats 410 R programming"

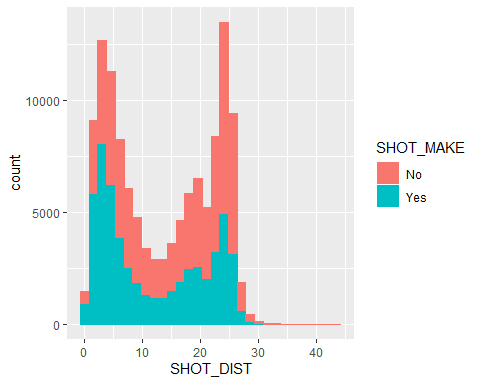
## -- Attaching packages ------------------------------------------------------ tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 1.0.2 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

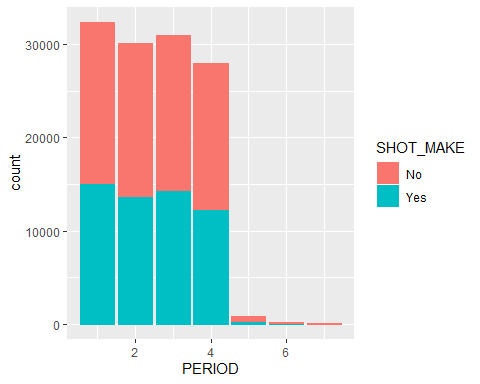
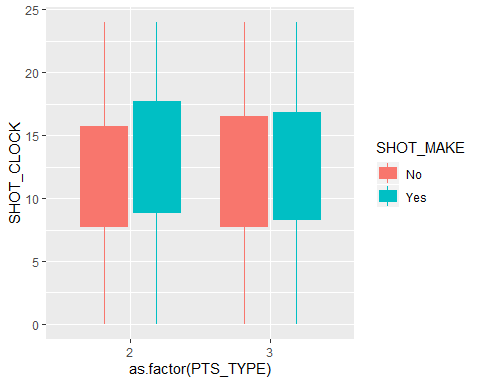
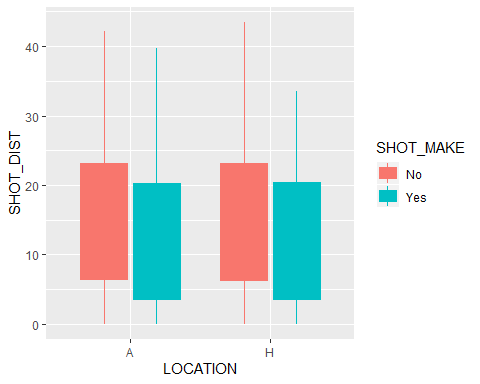
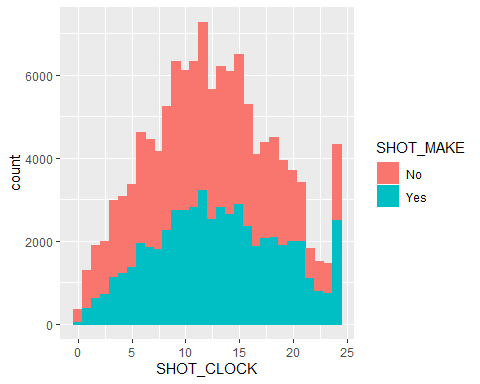
## -- Conflicts --------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::combine() masks randomForest::combine()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x ggplot2::margin() masks randomForest::margin()  
## x dplyr::select() masks MASS::select()

## GAME\_ID MATCHUP LOCATION W FINAL\_MARGIN SHOT\_NUMBER PERIOD  
## 1 21400899 MAR 04, 2015 - CHA @ BKN A W 24 1 1  
## 2 21400899 MAR 04, 2015 - CHA @ BKN A W 24 2 1  
## 4 21400899 MAR 04, 2015 - CHA @ BKN A W 24 4 2  
## 5 21400899 MAR 04, 2015 - CHA @ BKN A W 24 5 2  
## 6 21400899 MAR 04, 2015 - CHA @ BKN A W 24 6 2  
## 7 21400899 MAR 04, 2015 - CHA @ BKN A W 24 7 4  
## GAME\_CLOCK SHOT\_CLOCK DRIBBLES TOUCH\_TIME SHOT\_DIST PTS\_TYPE SHOT\_MAKE  
## 1 1:09 10.8 2 1.9 7.7 2 Yes  
## 2 0:14 3.4 0 0.8 28.2 3 No  
## 4 11:47 10.3 2 1.9 17.2 2 No  
## 5 10:34 10.9 2 2.7 3.7 2 No  
## 6 8:15 9.1 2 4.4 18.4 2 No  
## 7 10:15 14.5 11 9.0 20.7 2 No  
## CLOSEST\_DEFENDER CLOSEST\_DEFENDER\_PLAYER\_ID CLOSE\_DEF\_DIST FGM PTS  
## 1 Anderson, Alan 101187 1.3 1 2  
## 2 Bogdanovic, Bojan 202711 6.1 0 0  
## 4 Brown, Markel 203900 3.4 0 0  
## 5 Young, Thaddeus 201152 1.1 0 0  
## 6 Williams, Deron 101114 2.6 0 0  
## 7 Jack, Jarrett 101127 6.1 0 0  
## player\_name player\_id  
## 1 brian roberts 203148  
## 2 brian roberts 203148  
## 4 brian roberts 203148  
## 5 brian roberts 203148  
## 6 brian roberts 203148  
## 7 brian roberts 203148

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



shot\_distance: this is a histogram that compares whether a shot will go into the hoop or not based on how far they stood from the basket based on the results there are far more missed shots than made shots in particular the number of missed shots peaks at 25 feet from the basket The number of made shots peaks around 5 feet from the basket

shot\_clock: this is a histogram that compares whether a shot will go in based on the amount of time that is left on the shot clock the number of missed shots peaks around 12 to 12.5 feet away from the basket additionally the number of made shots peaks around 12 to 12.5 feet away from the basket

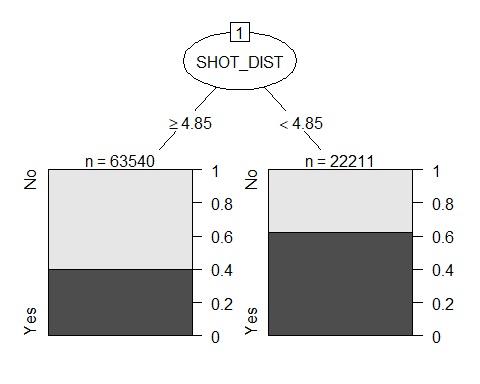
homevs away: i create a boxplot to compare the distance that players shoot from at home as opposed to away, this is complimented by a fill that filters for whether the shot is made or not visitng players who miss shots tend to take shots between 6 and 23 feet away from the basket vistting players who make shots tend to take shots between 4 and 21 feet away from the basket home players who miss shots tend to take shots between 6 and 23 feet away from the basket home players who make shots tend to take shots betwen 4 to 21 feet away from the basket

makeormiss: i created a boxplot that factors out each point type by whether the type of shot is a two pointer or a three pointer as well as the fact that the fill is based on whether the player made or missed the shot if a player missess a two point shot, they probably took the shot between 7.5 and 15.5 seconds if a player makes a two point shot, they probably took the shot between 9 and 22.5 seconds if a player misses a three point shot they probably took the shot between 7.5 seconds and 16 seconds if a player makes a three point shot they probably took the shot between 8 seconds and 16.5 seconds

makeormiss2: i created a barplot that compares that shots that are made and missed by the period of the game the players make the most shots in the first quarter and they miss the most shots in the fourth quarter. maybe the players are confident to start the game and crack under pressure when the game advances to the fourth quarter The players miss the most shots in the first quarter and they miss the least amount of shots in the fourth quarter. I think that the players miss most frequently in the first quarter as they are trying to be the most agreesive in the early going to help their team build a lead. I could see some players being tentitaive to shoot in the fourth quarter if they have missed a lot of shots earlier in the game or if the game is close and they dont want to hold the team back by missing Games rarely go to overtime so the data that is collected for periods 5 for 7 is insignificant compared to the data collected for periods 1 to 4

## Warning: package 'rpart.plot' was built under R version 3.6.3

## Call:  
## rpart(formula = SHOT\_MAKE ~ LOCATION + SHOT\_NUMBER + PERIOD +   
## SHOT\_CLOCK + DRIBBLES + SHOT\_DIST + PTS\_TYPE + CLOSE\_DEF\_DIST +   
## player\_name, data = train, method = "class")  
## n= 85751   
##   
## CP nsplit rel error xerror xstd  
## 1 0.1392525 0 1.0000000 1.0000000 0.003728715  
## 2 0.0100000 1 0.8607475 0.8610799 0.003656070  
##   
## Variable importance  
## SHOT\_DIST SHOT\_CLOCK CLOSE\_DEF\_DIST player\_name   
## 79 8 7 6   
##   
## Node number 1: 85751 observations, complexity param=0.1392525  
## predicted class=No expected loss=0.4561579 P(node) =1  
## class counts: 46635 39116  
## probabilities: 0.544 0.456   
## left son=2 (63540 obs) right son=3 (22211 obs)  
## Primary splits:  
## SHOT\_DIST < 4.85 to the right, improve=1661.1830, (0 missing)  
## PTS\_TYPE < 2.5 to the right, improve= 547.3777, (0 missing)  
## SHOT\_CLOCK < 19.15 to the left, improve= 351.8655, (0 missing)  
## player\_name splits as LRLRRLLRRRRRRLRRRLLRLRLLLLRRRLRLRLLLRRRLLLLLLRLLRRLLLRRRRLLLLRLLLRRRLLLRLLLLLLRLLRRLRLLLLLLLRRRLRRRLRLRLLRLLLRRRLLLLRLLLLLRLLRLLLLLRRLLLLRLRLLLLLRLLRLLLLRLLLRLRLLRLRLLLLRRRLLLRRLRRLLLLLLRLRLLLLLLLLLLLLRLLLLLLRLLRRLLLLRRLLLRLLLLLRLLLLLRRLLRRLLRLLLLRRLRLLRRLRRRLLLLRLRLLRLRLLLLLLLLRL, improve= 285.9683, (0 missing)  
## DRIBBLES < 1.5 to the right, improve= 75.9417, (0 missing)  
## Surrogate splits:  
## SHOT\_CLOCK < 20.65 to the left, agree=0.768, adj=0.104, (0 split)  
## CLOSE\_DEF\_DIST < 1.55 to the right, agree=0.765, adj=0.091, (0 split)  
## player\_name splits as LLLLLLLRLLRRRLLLRLLLLRLLLLLRLLLLLLLLLLLLLLLLLRLLLLLLLLLLLLLLLLLLLLRLLLLLLLLLLLLLLRRLRLLLLLLLLRLLLRRLLLLLLLLLLRLLLLLLLLLLLLRLLLRLRLLRRLLLLRLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLRLLLLLLLLLLLLLLLLLLLLLLRLLLLLLLLLLLLLLLLLLLLLLLRLLLLLLLLLLLLRLRLLLLLRLRLLLLLLRLRLLRLLLLLLLLLL, agree=0.761, adj=0.076, (0 split)  
##   
## Node number 2: 63540 observations  
## predicted class=No expected loss=0.3979698 P(node) =0.7409826  
## class counts: 38253 25287  
## probabilities: 0.602 0.398   
##   
## Node number 3: 22211 observations  
## predicted class=Yes expected loss=0.3773806 P(node) =0.2590174  
## class counts: 8382 13829  
## probabilities: 0.377 0.623



##   
## Classification tree:  
## rpart(formula = SHOT\_MAKE ~ LOCATION + SHOT\_NUMBER + PERIOD +   
## SHOT\_CLOCK + DRIBBLES + SHOT\_DIST + PTS\_TYPE + CLOSE\_DEF\_DIST +   
## player\_name, data = train, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] SHOT\_DIST  
##   
## Root node error: 39116/85751 = 0.45616  
##   
## n= 85751   
##   
## CP nsplit rel error xerror xstd  
## 1 0.13925 0 1.00000 1.00000 0.0037287  
## 2 0.01000 1 0.86075 0.86108 0.0036561

step 2: splits into train and test to develop tree

step 3 building the tree: according to the data set the three most important vairables are shot distance shot clock and close defender distance and palyer\_name the most important variables is shot\_distance

step 4 visualizating the tree: based on this tree if a shot is outside of 4.9 feet you have a 60% of missing and a 40% chance of making it based on this tree if a shot is inside 4.9 feet you have a 38% chance of missing it and a 62% chance of making it

Step 5 evaluation of the tree: the only variable that is used is shot\_distance the root error is 45.6%

##   
## No Yes  
## No 16345 10834  
## Yes 3642 5930

## No   
## 82

## Yes   
## 35

step 6: showcases the probablity of each shot going in the hoop by comparing the training set and test training set

step 7: write a function to assign whether a shot will go in or not based on its probabilty i wrote this function to return no if the probabilty exceeds 0.6020302 if the probability is anything else return yes i have a better idea of what kind of shot was taken based on this function the shots that are predicted to miss where more than likely further away from the hoop and the shots that made it in were more than likely closer to the hoop

step 8: this is used to assess the accuracy of table for test vs training data in predicting whether shots go in teh basket or outside of the basket  
predicts shots missed with 82% accuracy predicts shots made with 35% accuracy

##   
## Call:  
## randomForest(formula = SHOT\_MAKE ~ LOCATION + SHOT\_NUMBER + PERIOD + SHOT\_CLOCK + DRIBBLES + SHOT\_DIST + PTS\_TYPE + CLOSE\_DEF\_DIST, data = train, importance = TRUE)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 2  
##   
## OOB estimate of error rate: 38.94%  
## Confusion matrix:  
## No Yes class.error  
## No 38636 7999 0.1715235  
## Yes 25389 13727 0.6490694

##   
## rfpredict No Yes  
## No 16609 10898  
## Yes 3378 5866

## No   
## 83

## Yes   
## 35

step 9: create the random forest model the random forest model reduces the error slightly from 45.6% to 38.9%

step 10: create predictions and tables, evlaute accuracy of the random forest vs test the random forest model predicts shots that are missed with 83% accuracy the random forest model predidcts shots that are made with 35% accuracy

step 11: write conclusions

conclusion: based on the results of this analysis I have came up with a few conclusions

# the variables that are the most important in building the are shot distance shot clock and close defender distance. The only variable that is used in buliding the tree is the shot distance. The decision tree was built on whether a play takes a shot from 4.9 feet away and the likelihood as to whether or not they would make or miss the shotwould depend on how far they stood from the basket

1. The random forest model does a better job than the decision tree of indicating whether a shot is missed. There isn’t a difference between the decision tree and the random forest model of indicating whether a shot would be made as both models predict shots made with a 35% accuracy rate. I would recommend using the random forest model over the decision tree as it does a better job of reducing accuracy issues than the decision tree. Additionally, the error is reduced as well which makes the model reliable.