2021 NBA all-star 24-man roster prediction

Final Project for DSA 5103

Group members:

Haixiao Lu, Runsheng Zhou, Tengnan Yao

2020

Executive Summary

The National Basketball Association All-Star Game has been an interesting weekend for all the basketball fans all over the world. This project is based on personal interest to predict the top 12 players from both East and West Conference. It is compared with previously selected NBA All-Star players in the previous seasons. The goal of this project is to understand the impact feature selection had on the overall accuracy of the analysis when comparing different models. This model also will predict which team will win between the West and East Conference Team.

The training dataset used for this project contains 5 seasons data which was from 2015 to 2019. 182812 observations and 90 variables are collected from this dataset. For the purpose of creating a higher performance model, the data was split by each season. To reduce the 90 features, a wrapper feature selection algorithm known as Boruta was used in each dataset to extract the important features. Some of the new features also added to each dataset, such as sum, average and rank all player’s performance in that particular season. Then the entire data set is categorized by each individual player’s name. In addition, the players are also divided into West and East since we are ultimately choosing players from East and West conferences separately. The final training dataset has 202 features for each season dataset. Therefore, there are 10 dataframes(5 seasons times 2 conference teams each) to train the models as training data. The rest of 2 dataframes which is the data from the 2020-2021 season are used to predict the final result.

The modeling process consisted of creating both logistic regression models and decision trees.

Backgrounds

The National Basketball Association All-Star Game is a basketball exhibition game hosted every February by the National Basketball Association (NBA) and showcases 24 of the league's star players. It is the featured event of NBA All-Star Weekend, a three-day event which goes from Friday to Sunday.

The starting lineup for each squad is selected by a combination of fan, player, and media voting, while head coaches choose the reserves, seven players from their respective conferences, so each side has a 12-man roster. Coaches are not allowed to vote for their own players. If a selected player cannot participate because of injury, the NBA commissioner selects a replacement.

Starting in 2018, the leading vote-getters for each conference are designated as team captains and can choose from the pool of All-Star reserves to form their teams regardless of conference. LeBron James and Stephen Curry became the first players to choose teams through the new format, selecting players for the 2018 NBA All-Star Game in a non-televised draft on January 25. Likely due to fan interest in the draft process, captains for the 2019 All-Star Game, James and Giannis Antetokounmpo, drafted their teams live on TNT. The teams also play for a charity of their choice to help the games remain competitive.

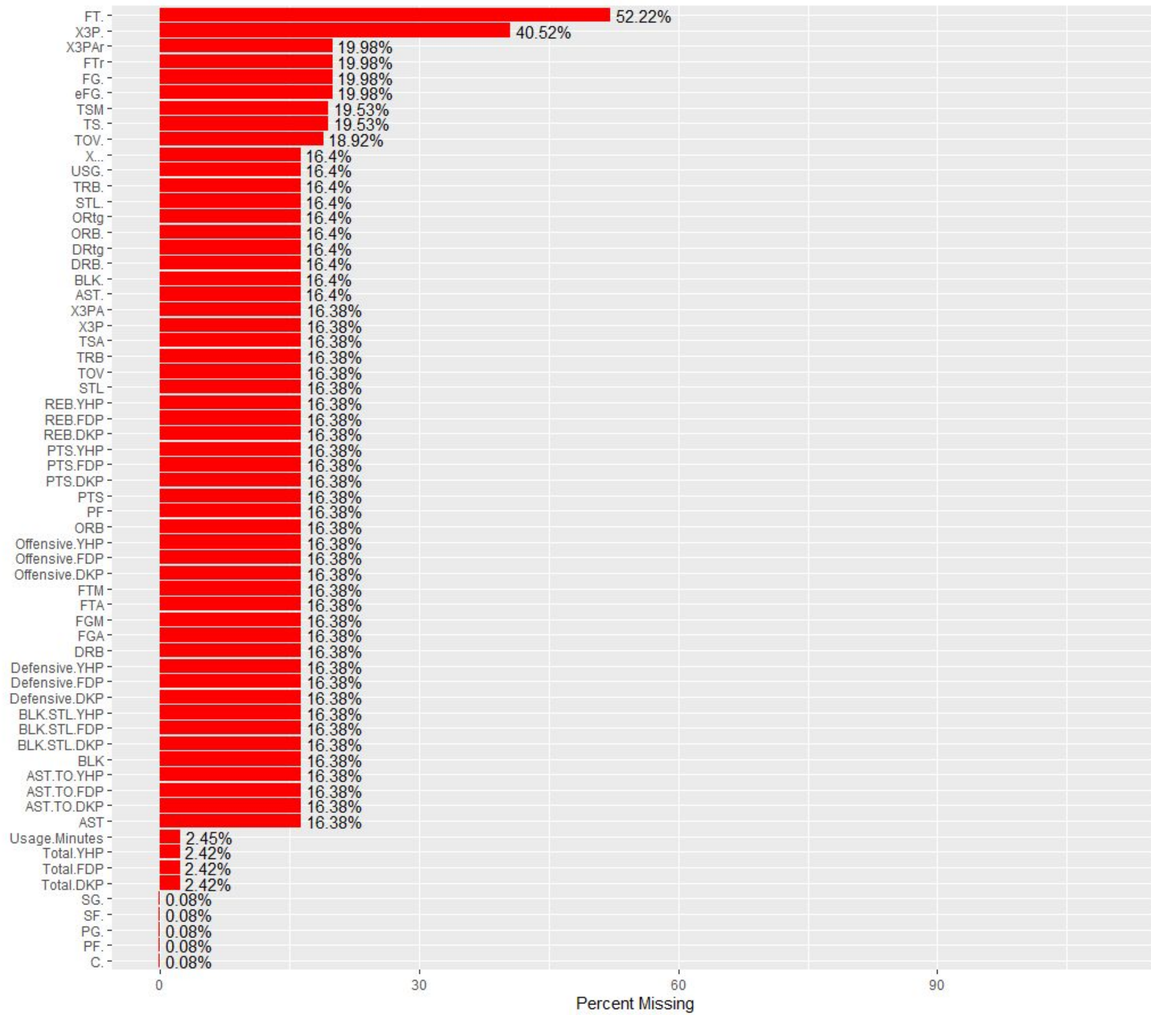
The starting five from each conference consists of three frontcourt players and two guards, selected by a combination of fan, player, and media voting. In 2017, the NBA moved from a pure fan vote to a weighted process wherein fan voting accounts for 50%, with player and media voting account for 25% each. Prior to 2013, fans selected two forwards and one center instead of generic frontcourt players. The NBA in 2003 began offering All-Star ballots in three languages—English, Spanish and Chinese—for fan voting of the starters.

Data Description

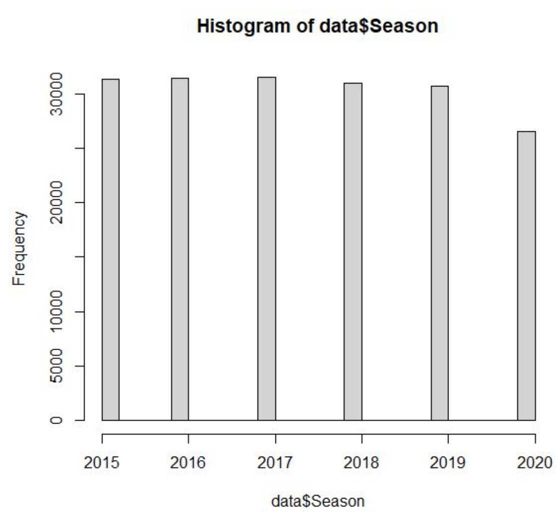
The Dataset includes NBA team's data with 90 attributes, such as FGM(Field Goal Made by Team), FGA(Field Goal Attempts by Team), Pts(Points by Team) and 87 other variables. The dimension of the data is 182811 observations as presented in Fig-1. There are 12 categorical variables, and the rest of all others are continuous variables.   Each row represents the data of a specific player, measuring how well he performed in that game.

```
> dim(data)
[1] 182811     90
```

We can see there are so many NAs in the data, most of which are easy to deal with because they usually are caused by the player not playing in a game.



The data is about all records in the past 6 seasons.



Those are the 30 teams (15 on each side), from which we need to know if a team is from the East side or West.

```
> levels(as.factor(data$Team))
 [1] "Atlanta Hawks"          "Boston Celtics"         "Brooklyn Nets"          "Charlotte Hornets"
 [5] "Chicago Bulls"          "Cleveland Cavaliers"    "Dallas Mavericks"       "Denver Nuggets"
 [9] "Detroit Pistons"        "Golden State Warriors"  "Houston Rockets"        "Indiana Pacers"
[13] "Los Angeles Clippers"   "Los Angeles Lakers"     "Memphis Grizzlies"      "Miami Heat"
[17] "Milwaukee Bucks"        "Minnesota Timberwolves" "New Orleans Pelicans"   "New York Knicks"
[21] "Oklahoma City Thunder"  "Orlando Magic"          "Philadelphia 76ers"     "Phoenix Suns"
[25] "Portland Trail Blazers" "Sacramento Kings"       "San Antonio Spurs"      "Toronto Raptors"
[29] "Utah Jazz"              "Washington Wizards"
```

```
#      East:                                     West:
#        - Boston Celtics                          - Utah Jazz
#        - Toronto Raptors                         - Oklahoma City Thunder
#        - New York Knicks                         - Portland Trail Blazers
#        - Philadelphia 76ers                      - Denver Nuggets
#        - Brooklyn Nets                           - Minnesota Timberwolves
#        - Cleveland Cavaliers                     - Golden State Warriors
#        - Detroit Pistons                         - Los Angeles Clippers
#        - Milwaukee Bucks                         - Sacramento Kings
#        - Indiana Pacers                          - Los Angeles Lakers
#        - Chicago Bulls                           - Phoenix Suns
#        - Washington Wizards                      - San Antonio Spurs
#        - Atlanta Hawks                           - Houston Rockets
#        - Miami Heat                              - Memphis Grizzlies
#        - Charlotte Hornets                       - New Orleans Pelicans
#        - Orlando Magic                           - Dallas Mavericks
#
```

Data preparation

Separate data into seasons.

```
data2015 <- data[data$Season == 2015, ]
data2016 <- data[data$Season == 2016, ]
data2017 <- data[data$Season == 2017, ]
data2018 <- data[data$Season == 2018, ]
data2019 <- data[data$Season == 2019, ]
data2020 <- data[data$Season == 2020, ]
```

Create a function to batch clean the data. The next following steps are all in the function.

```
# data cleaning
massage = function(dataFrame) {
```

Deal with the NAs.

```
# player did not play in a specific game
newData <- dataFrame[dataFrame$Minutes != "Did Not Play", ]

# remove rows that have too many NAs
newData <- newData[rowSums(is.na(newData)) < 10,]
```

Select some attributes that we think are useful. Actually we selected 52 attributes in the end.

```
cols <- c(
  "Player",
  "Team",

  "FGM",   # Field Goals Made
  "FGA",   # Field Goals Attempt
  # "FG.",   # Field Goals Percentage
  "X3P",   # 3-Points
  "X3PA",  # 3-Points Attempt
  # "X3P.",  # 3-Points Percentage
  "FTM",   # Free Throws Made
```

Extract the clean data.

```
cleanData2015 <- massage(data2015)
cleanData2016 <- massage(data2016)
cleanData2017 <- massage(data2017)
cleanData2018 <- massage(data2018)
cleanData2019 <- massage(data2019)
cleanData2020 <- massage(data2020)
```

Separate East from West like this:

```
West <- c("Golden State Warriors", "Houston Rockets", "Utah Jazz", "Phoenix Suns",
          "Denver Nuggets", "Oklahoma City Thunder", "San Antonio Spurs",
          "Dallas Mavericks", "Minnesota Timberwolves", "Los Angeles Lakers",
          "Los Angeles Clippers", "Memphis Grizzlies", "Portland Trail Blazers",
          "Sacramento Kings", "New Orleans Pelicans")

data2015_West <- cleanData2015[cleanData2015$Team %in% West, ]
data2015_West$Team <- NULL
data2015_East <- cleanData2015[!(cleanData2015$Team %in% West), ]
data2015_East$Team <- NULL
```

We need to aggregate the data grouped by player. We chose 4 type of aggregations:

1. Sum: for example that we think a player will be selected or not may rely on the total score he made throughout the season.
2. Mean: we also think that the mean score might be useful.
3. Rank of sum: how a player performed compared with all other players on sum
4. Rank of mean: how a player performed compared with all other players on mean

So we created an aggregate function.

```
# aggregations(sum, mean, rank sum, rank mean)
aggregateFunc <- function(data) {

  groupedData <- data %>% group_by(Player)

  sumData <- groupedData %>% summarise_all(sum)
  colnames(sumData) <- paste("sum", colnames(sumData), sep = "_")
  names(sumData)[1] <- "Player"

  meanData <- groupedData %>% summarise_all(mean)
  colnames(meanData) <- paste("mean", colnames(meanData), sep = "_")

  sumDataRank <- (-sumData[, -1]) %>% apply(2, rank)
  glimpse(sumDataRank)
  colnames(sumDataRank) <- paste("rank", colnames(sumDataRank), sep = "_")

  meanDataRank <- (-meanData[, -1]) %>% apply(2, rank)
  colnames(meanDataRank) <- paste("rank", colnames(meanDataRank), sep = "_")

  return (sumData %>%
            cbind(meanData[, -1]) %>%
            cbind(sumDataRank) %>%
            cbind(meanDataRank)
          )

}
```

Apply to all seasons like this:

```r
# season 2015
data2015_East_Aggr <- aggregateFunc(data2015_East)
data2015_West_Aggr <- aggregateFunc(data2015_West)
```

We need to know if a player was in the roster or not in previous seasons. For example:

```r
# 2015-2016
#   Eastern:                          Western:
#     John Wall                         Stephen Curry
#     Kyle Lowry                        Kobe Bryant
#     LeBron James                      Anthony Davis
#     Pau Gasol                         Marc Gasol
#     Carmelo Anthony                   Blake Griffin
#     Al Horford                        LaMarcus Aldridge
#     Chris Bosh                        Tim Duncan
#     Paul Millsap                      Kevin Durant
#     Jimmy Butler                      Klay Thompson
#     Dwyane Wade                       Russell Westbrook
#     Jeff Teague                       James Harden
#     Kyrie Irving                      Chris Paul
#     Kyle Korver                       DeMarcus Cousins
#                                       Damian Lillard
#                                       Dirk Nowitzki
```

```r
allstar2015east <- c('John Wall', 'Kyle Lowry', 'LeBron James', 'Pau Gasol',
                     'Carmelo Anthony', 'Al Horford', 'Chris Bosh', 'Paul Millsap',
                     'Jimmy Butler', 'Dwyane Wade', 'Jeff Teague', 'Kyrie Irving',
                     'Kyle Korver')

allstar2015west <- c('Stephen Curry', 'Kobe Bryant', 'Anthony Davis', 'Marc Gasol',
                     'Blake Griffin', 'LaMarcus Aldridge', 'Tim Duncan',
                     'Kevin Durant', 'Klay Thompson', 'Russell Westbrook',
                     'James Harden', 'Chris Paul', 'DeMarcus Cousins',
                     'Damian Lillard', 'Dirk Nowitzki')
```

Then we add another column “select” to indicate if a player was in the roster(1) or not(0). The 2020-2021 season data shall not have that column of course.

```r
# season 2015
d2015east <- data2015_East_Aggr
d2015east$select <- 0
d2015east$select[d2015east$Player %in% allstar2015east] <- 1
```

Modeling Approach

Our modeling approaches were straight forward. Since we were predicting whether the players who were selected in the All Star Team or not, it was a classification problem. It was broken down to two main modeling approaches, logistic modeling and classification trees.

The analysis for classification of players selection was done using numerous predictive algorithms for the project. The predictive classification models used are: Logistic Regression, Decision Tree, Penalized Logistic Regress (Ridge), MARS, Support Vector Machine, and Random Forest. Four different logistic models were created based on the 202 features. First model used the whole dataset to build a basic model. Then, the second model from the tree's variable importance was used. Third one we used all the important features from Boruta algorithm feature selection. The last one we created a penalized logistic regression model using important features that were selected from the Boruta algorithm. The model then went through stepwise optimization for the Akaike Information Criterion (AIC). From these optimal models, the team continued to iterate as well, working to remove variables with lesser significance without leading to large reductions in the overall performance of the model.

A classification tree was also created based on all 202 features. Both decision tree and random forest used during classification tree creation. Both tree's variable importance were then used in the creation of several logistic models.

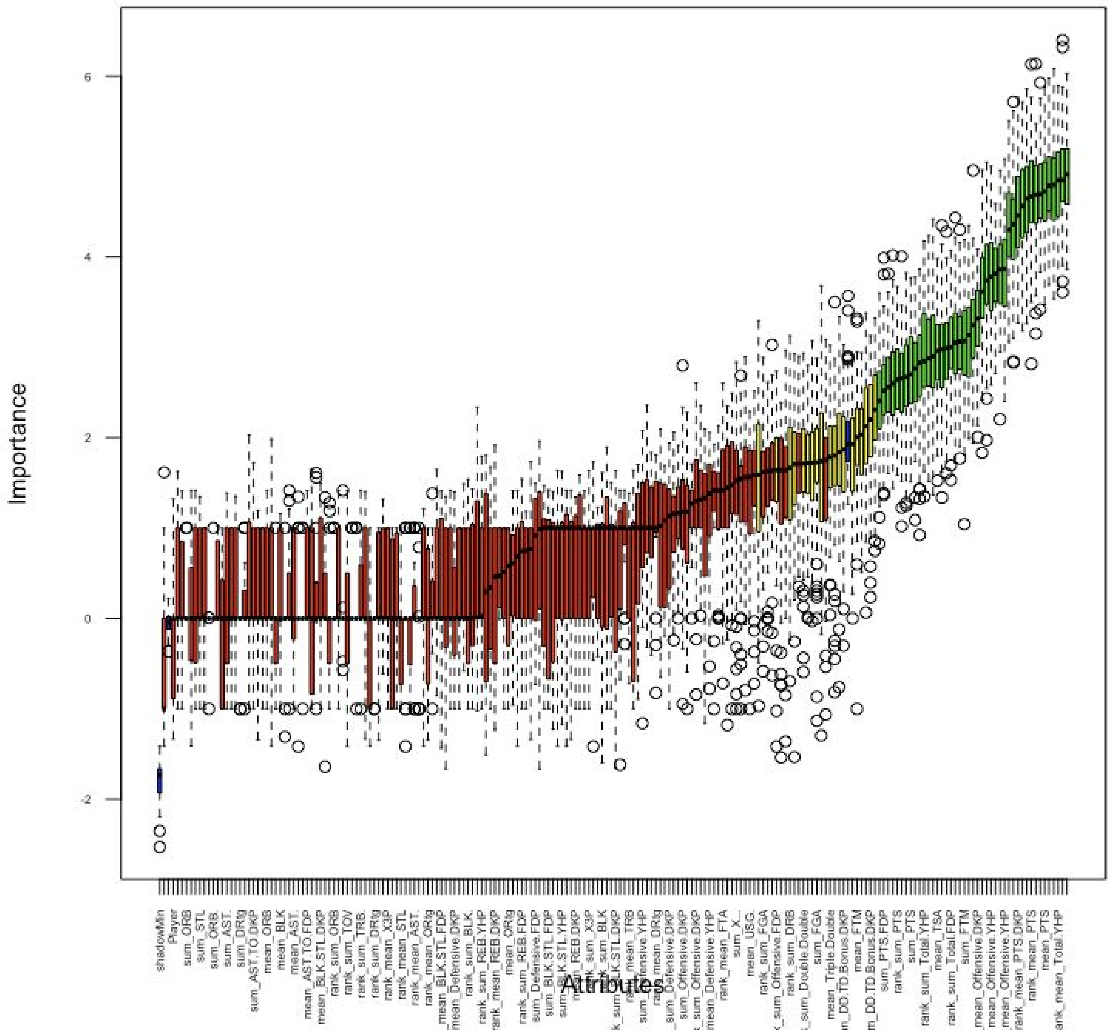
Ideally, all the players who were selected previously in the season 2015 - 2019 should be showing in our predictions. We used several measure metrics to evaluate our model performance, such as accuracy, kappa, sensitivity and specificity.

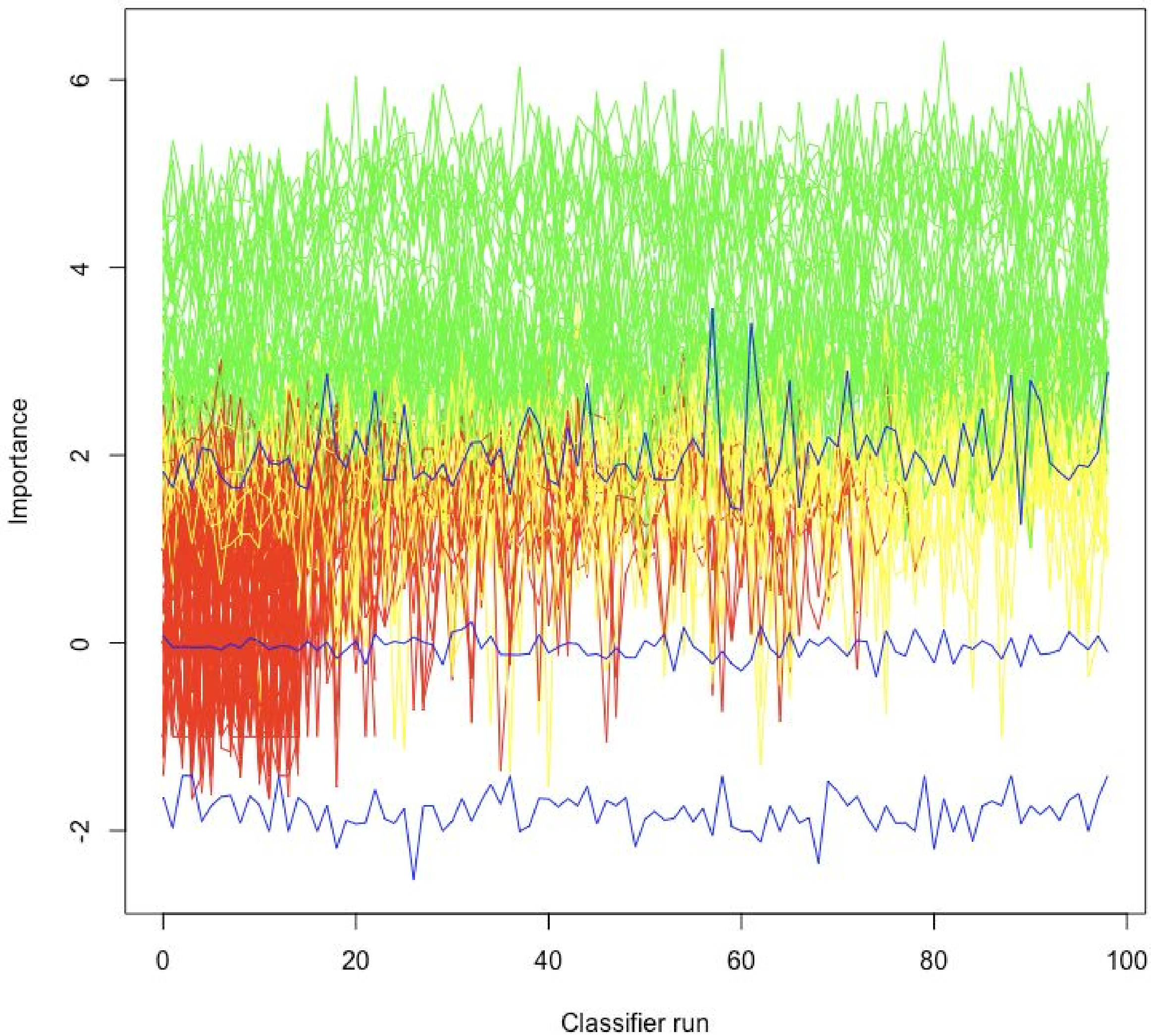
As we tested, the accuracy was all around 95% or above. So we decide to choose all models that we think are doing good and sum up the prediction values from each model.

Finally, we got our result.

Results & Validation of Analysis

To begin our modeling creation and analysis, the team used Boruta Algorithm to extract all the important features from the dataset. Then we used these features to create our logistic model and classification trees. As the plot shows below, the red and green color shows whether the variable is important or not, yellow indicates tentative features which can not decide.


Importance
Attributes
6
4
2
0
-2


Importance
6
4
2
0
-2
0
20
40
60
80
100
Classifier run

Logistic Model with all variables (lgMod)

The first natural step in creating a logistic model from scratch was to create a model that incorporated all of the 202 feature variables. This model was created to be used as a stepping stone to other models. The model was known as “lgMod”, and was created using the code shown below.

```
# 1. build logistic regression model with whole dataset

lgMod <- glm(select~., data = d2015west, family =  "binomial")
```

Logistic Model with Boruta Algorithm Features Selection

There were 49 variables selected from Boruta Algorithm. These 49 variables were used to create a logistic model. We also used each season’s dataset which from 2015 - 2019 to create an individual model, then we used these different individual models to predict season 2020 All Star players.

Results of Boruta Algorithm

```
Boruta performed 99 iterations in 9.056326 secs.
Tentatives roughfixed over the last 99 iterations.
 49 attributes confirmed important: mean_FGA, mean_FGM, mean_Offensive.DKP,
mean_Offensive.FDP, mean_Offensive.YHP and 44 more;
 152 attributes confirmed unimportant: mean_AST, mean_AST., mean_AST.TO.DKP,
mean_AST.TO.FDP, mean_AST.TO.YHP and 147 more;
```

Logistic Model with Boruta Selected Features

```
lgMod2 <- glm(data =d2015west, select ~ sum_FGM +
                    sum_FTM + sum_PTS + sum_PTS.FDP+ sum_Total.FDP + sum_PTS.DKP+
                    sum_Total.DKP  + sum_PTS.YHP  + sum_Total.YHP + mean_FGM   +
                    mean_FGA + mean_PTS  + mean_TSA  + mean_PTS.FDP + mean_Offensive.FDP  +
                    mean_Total.FDP   + mean_PTS.DKP  + mean_Offensive.DKP  + mean_Total.DKP   +
                    mean_PTS.YHP +  mean_Offensive.YHP + mean_Total.YHP  + rank_sum_FGM    +
                    rank_sum_FTM + rank_sum_PTS   + rank_sum_PTS.FDP  + rank_sum_Total.FDP   +
                    rank_sum_PTS.DKP + rank_sum_Total.DKP  + rank_sum_PTS.YHP  + rank_sum_Total.YHP   +
                    rank_mean_FGM + rank_mean_FGA  + rank_mean_PTS  + rank_mean_TSA   +
                    rank_mean_PTS.FDP  + rank_mean_Offensive.FDP   + rank_mean_Total.FDP  +
                    rank_mean_PTS.DKP  + rank_mean_Offensive.DKP  + rank_mean_Total.DKP   +
                    rank_mean_PTS.YHP  + rank_mean_Offensive.YHP + rank_mean_Total.YHP, family = "binomial")
```

Logistic Model with Classification Tree Variables (lgMod_2015-2019)

There were thirteen variables used from the classification tree. These thirteen variables were used to create logistic models from dataset 2015 - 2019 with the code shown below.

```
# d2015_east
lgMod_2015 <- glm(select~ mean_Total.FDP + mean_Total.YHP + rank_mean_Total.FDP +
                   rank_mean_Total.YHP + mean_Total.DKP + rank_mean_Total.DKP +
                   rank_sum_FGM + sum_FGM + sum_PTS + sum_PTS.DKP + sum_PTS.FDP +
                   sum_PTS.YHP, data = d2015east, family = "binomial")
```

Logistic Model with Stepwise Method (AIC)

To find out the most significant variables for prediction of selecting players of All NBA Teams, stepwise Alkaline Information Criteria(AIC) was used for feature selection. The number of variables (subsets) with the lowest AIC value are selected to fit into the logistic regression model. After a little bit of time to run the model, the lowest AIC was 10 which included 4 features.

```r
model.both <- stepAIC(lgMod) # this may take a long time
model.both$formula
```

```
> model.both <- stepAIC(lgMod) # this may take a long time
Start:  AIC=260
select ~ sum_FGM + sum_FGA + sum_X3P + sum_X3PA + sum_FTM + sum_FTA +
    sum_ORB + sum_DRB + sum_TRB + sum_AST + sum_STL + sum_BLK +
    sum_TOV + sum_PF + sum_PTS + sum_X... + sum_ORB. + sum_DRB. +
    sum_TRB. + sum_AST. + sum_STL. + sum_BLK. + sum_USG. + sum_ORtg +
    sum_DRtg + sum_TSA + sum_PTS.FDP + sum_AST.TO.FDP + sum_REB.FDP +
    sum_BLK.STL.FDP + sum_Offensive.FDP + sum_Defensive.FDP +
    sum_Total.FDP + sum_PTS.DKP + sum_AST.TO.DKP + sum_REB.DKP +
    sum_BLK.STL.DKP + sum_DD.TD.Bonus.DKP + sum_Offensive.DKP +
```

```
Step:  AIC=10
select ~ sum_AST + sum_TOV + sum_DRB. + mean_FGA

           Df Deviance    AIC
<none>           0.000 10.000
- sum_TOV   1   19.101 27.101
- sum_AST   1   20.544 28.544
- sum_DRB.  1   34.153 42.153
- mean_FGA  1   67.630 75.630
```

Penalized Logistic Model with All Variables (ridge)

Since Penalized logistic regression works around with low bias and high variance, by applying penalties to fluctuations of given parameters. It reduces the coefficients of less contributing variables to zero. We used ridge as one type of penalized logistic regression in our modeling process. One of the example as following:

```r
trctrl <- trainControl (method = "cv", number = 10)
fit2015_east <- train(select~., data = d2015east[,-1],
               method = "glmnet",
               trControl = trctrl,
               tuneGrid = data.frame(alpha = 0, lambda = seq(0, 0.5, 0.8)))
```

MARS

Multivariate adaptive regression splines (MARS) provide a convenient approach to capture the nonlinearity aspect of polynomial regression. We used the R package "earth" with all the variables in the dataset to create our model. An example code as below,

```r
# --2016 east--
mars2016_east <- earth(select~., data = d2016east[,-1])
```

Classification Trees (DT)

To begin our analysis, the team created a decision tree based on all variables predicting the ground truth for player selection and team winning. The output of the decision tree is a factor that indicates the predicted value of whether or not a player is selected.

```r
# 2015east data
DT2015_east <- rpart(select~., data = d2015east[,-1])
```

Classification Tree with Cross-Validation

Since decision trees can divide large groups of data set into smaller subsets of groups. We also used some Cross-Validation to run the decision tree model.

```r
fitControl <- trainControl(method = "repeatedcv", number = 5, repeats = 10)

# 2015 east
tr2015_east <- train(select~., data = d2015east, method = "rpart", trControl = fitControl)
east2015_dt <- rpart(select~., data = d2015east[,-1], cp = tr2015_east$bestTune)
plot(tr2015_east)
```

Random Forest Model

Random Forest is similar to a decision tree which is used to classify by splitting feature values into branches. mtry is the number of variables randomly sampled at each split. We set maxnodes to 25 for each tree and 800 ntree in our model

```
# 2015 east
RF2015_east <- randomForest(select~., data = d2015east[,-1],
                            mtry = 4, ntree = 800, maxnodes = 25)
```

Support Vector Machine (SVM)

The basic idea of a support vector machine can be considered as separating two classes in feature space using a hyperplane separator. The optimal separator is determined by maximizing the distance to the nearest neighbor of each class. We used package e1071 library in r to use the svm function as display below:

```
# 2015 west
svm2015_west <- svm(select~., data = d2015west[,-1], type = "C-classification",
                    kernel = 'linear', probability = TRUE)
```

Prediction

We use all our models trained from the data of previous seasons and do predictions on the 2020-2021 season. Then we put all prediction values to a dataframe like this:

```
#---------Season 2015-------

# use 2015 east dataset to build model and predict
result_east["rf2015e"] <- rfModel(d2015east)[1]
result_west["rf2015e"] <- rfModel(d2015east)[2]

# use 2015 west dataset to build model and predict
result_east["rf2015w"] <- rfModel(d2015west)[1]
result_west["rf2015w"] <- rfModel(d2015west)[2]
```

Then we got a dataframe where all scores are predicted for each player who ever played in 2015-2019 seasons. We have 279 players from the west and 280 players from the east with 61 scores from different models.

```
> dim(result_west)
[1] 279  61
```

```
> dim(result_east)
[1] 280  61
```

We normalized those scores to range (0-1) and summed up all of them. Then we sorted the result of their total score.

```
# east
norm_east <- preProcess(result_east, method = c("range"))
final_result_east <- predict(norm_east, result_east)
final_result_east$sum <- rowSums(final_result_east[ , -1])

players_east <- data.frame(
  final_result_east$Player,
  final_result_east$sum
)[order(-final_result_east$sum), ]

# west
norm_west <- preProcess(result_west, method = c("range"))
final_result_west <- predict(norm_west, result_west)
final_result_west$sum <- rowSums(final_result_west[ , -1])

players_west <- data.frame(
  final_result_west$Player,
  final_result_west$sum
)[order(-final_result_west$sum), ]
```

Our final result dataframe looks like this:

West

	final_result_west.Player	final_result_west.sum
113	James Harden	58.597080
183	Luka Doncic	52.946702
177	LeBron James	50.849355
159	Kawhi Leonard	50.759079
11	Anthony Davis	50.368881
42	Damian Lillard	50.042625
238	Russell Westbrook	44.585085
215	Nikola Jokic	41.082220
63	Devin Booker	38.654754
158	Karl-Anthony Towns	37.385597
69	Donovan Mitchell	30.826047
23	Brandon Ingram	28.376649
170	Kristaps Porzingis	28.303039
36	CJ McCollum	27.584373
56	DeMar DeRozan	27.339637
96	Hassan Whiteside	25.086533
237	Rudy Gobert	24.215628
41	D'Angelo Russell	22.063419
156	Jusuf Nurkic	20.908052
222	Paul George	20.377155
51	De'Aaron Fox	18.305177
35	Chris Paul	17.987385
243	Shai Gilgeous-Alexander	16.718997
9	Andrew Wiggins	16.395005
53	Deandre Ayton	15.886684
149	Jrue Holiday	15.588289

East

	final_result_east.Player	final_result_east.sum
99	Giannis Antetokounmpo	59.485680
258	Trae Young	49.128231
25	Bradley Beal	47.701291
124	Jayson Tatum	46.533057
133	Joel Embiid	40.627940
212	Pascal Siakam	39.409793
205	Nikola Vucevic	37.124358
279	Zach LaVine	36.724771
165	Kyrie Irving	36.558775
130	Jimmy Butler	34.492242
73	Domantas Sabonis	32.924626
14	Andre Drummond	32.302587
20	Ben Simmons	28.767060
19	Bam Adebayo	28.367999
163	Kyle Lowry	28.004896
255	Tobias Harris	26.878375
235	Spencer Dinwiddie	26.495683
158	Khris Middleton	25.330689
134	John Collins	20.445874
151	Kemba Walker	19.878479
144	Julius Randle	19.305934
91	Fred VanVleet	19.161636
69	Devonte' Graham	17.808578
155	Kevin Love	15.461930
48	Collin Sexton	12.899505
103	Gordon Hayward	12.772484

Conclusion

As we predict, the top scored 12 players that will be selected in the roster of West are:

- James Harden
- Luka Doncic
- LeBron James
- Kawhi Leonard
- Anthony Davis
- Damian Lillard
- Russell Westbrook
- Nikola Jokic
- Devin Booker
- Karl-Anthony Towns
- Donovan Mitchell
- Brandon Ingram

12 players that will be selected in the roster of East are:

- Giannis Antetokounmpo
- Trae Young
- Bradley Beal
- Jayson Tatum
- Joel Embiid
- Pascal Siakam
- Nikola Vucevic
- Zach LaVine
- Kyrie Irving
- Jimmy Butler
- Domantas Sabonis
- Andre Drummond

By summing up all the 12 players' scores on each side, we got:

East	West
493.0134	534.4741

So west will win!

Appendix 1: Terminology

Pts = Points By Team

FGM = Field Goals Made By Team

FGA = Field Goal Attempts By Team

FG% = Field Goal Percentage By Team

2FGM = Two-Point Field Goals Made By Team

2FGA = Two-Point Field Goal Attempts By Team

2FG% = Two-Point Field Goal Percentage By Team

3FGM = Three-Point Field Goals Made By Team

3FGA = Three-Point Field Goal Attempts By Team

3FG% = Three-Point Field Goal Percentage By Team

eFG% = Effective Field Goal Percentage By Team

TS% = Total Shot Percentage By Team

ORB = Offensive Rebounds By Team

DRB = Defensive Rebounds By Team

TRB = Total Rebounds By Team

ORB% = Offensive Rebound Percentage By Team

DRB% = Defensive Rebound Percentage By Team

Fouls = Total Fouls By Team

AST = Total Assists By Team

AST/TO = Assist to Turnover Ratio By Team

TOV% = Turnover Percentage By Team

TO Ratio = Turnover Ratio By Team

STL = Steals By Team

BLK = Blocks By Team

Pos = Time Of Possession By Team

Avg Pos = Average Possession By Team

Set Play = Set Play Possession By Team

OOBB = Baseline Inbound (Out Of Bounds - Baseline) Possessions By Team

OOBS = Sideline Inbound (Out Of Bounds - Sideline) Possessions By Team

Trans = Transition Possessions By Team

2nd Ch = 2nd Chance Possession By Team

Total = Pace By Team

PPP = Points Per Possession By Team

Set Play = Set Play Points By Team

OOBB = Baseline Inbound (Out Of Bounds - Baseline) Points By Team

OOBS = Sideline Inbound (Out Of Bounds - Sideline) Points By Team

Trans = Transition Points By Team

2nd Ch = 2nd Chance Points By Team

PITP = Points In The Paint By Team

Plays = Play Types By Team

Pts = Plays Total Points

Fouls = Fouls By Team

Off = Offensive Fouls By Team

Pers = Personal Fouls By Team

Tech = Technical Fouls By Team

Unsp = Unsportsmanlike Fouls By Team

DQ = Disqualifying Fouls By Team

Appendix 2: Roster from previous seasons

2015-2016

Eastern:	Western:
John Wall	Stephen Curry
Kyle Lowry	Kobe Bryant
LeBron James	Anthony Davis
Pau Gasol	Marc Gasol
Carmelo Anthony	Blake Griffin
Al Horford	LaMarcus Aldridge
Chris Bosh	Tim Duncan
Paul Millsap	Kevin Durant
Jimmy Butler	Klay Thompson
Dwyane Wade	Russell Westbrook
Jeff Teague	James Harden
Kyrie Irving	Chris Paul
Kyle Korver	DeMarcus Cousins
	Damian Lillard
	Dirk Nowitzki

2016-2017

Eastern:	Western:
Dwyane Wade	Stephen Curry
Kyle Lowry	Russell Westbrook
LeBron James	Kobe Bryant
Paul George	Kevin Durant
Carmelo Anthony	Kawhi Leonard
Jimmy Butler	Chris Paul
Chris Bosh	LaMarcus Aldridge
John Wall	James Harden
Paul Millsap	Anthony Davis
DeMar DeRozan	DeMarcus Cousins
Andre Drummond	Klay Thompson
Isaiah Thomas	Draymond Green
Pau Gasol	
Al Horford	

2017-2018

Eastern:	Western:
Kyrie Irving	Stephen Curry
DeMar DeRozan	James Harden
LeBron James	Kevin Durant
Jimmy Butler	Kawhi Leonard

Giannis Antetokounmpo	Anthony Davis
Isaiah Thomas	Russell Westbrook
John Wall	Klay Thompson
Kevin Love	Draymond Green
Carmelo Anthony	DeMarcus Cousins
Kyle Lowry	Marc Gasol
Paul George	DeAndre Jordan
Kemba Walker	Gordon Hayward
Paul Millsap	

2018-2019

Eastern:	Western:
Kyrie Irving	Stephen Curry
DeMar DeRozan	James Harden
LeBron James	Kevin Durant
Joel Embiid	DeMarcus Cousins
Giannis Antetokounmpo	Anthony Davis
Bradley Beal	Russell Westbrook
Goran Dragic	Damian Lillard
Al Horford	Draymond Green
Kevin Love	Karl-Anthony Towns
Kyle Lowry	LaMarcus Aldridge
Victor Oladipo	Klay Thompson
Kristaps Porzingis	Jimmy Butler

John Wall	Paul George
Andre Drummond	
Kemba Walker	

2019-2020

Eastern:	Western:
Kemba Walker	Stephen Curry
Kyrie Irving	James Harden
Kawhi Leonard	Kevin Durant
Giannis Antetokounmpo	Paul George
Joel Embiid	LeBron James
Kyle Lowry	Russell Westbrook
Victor Oladipo	Damian Lillard
Khris Middleton	Klay Thompson
Bradley Beal	Anthony Davis
Ben Simmons	LaMarcus Aldridge
Blake Griffin	Nikola Jokic
Nikola Vucevic	Karl-Anthony Towns
Dwyane Wade	Dirk Nowitzki
D'Angelo Russell	