

Updated ID11 Player Valuation Model 2023

Nicholas Kondo, Darron Kotoyan, Nic Vamis

- Introduction
- Joining Tables
- Filtering By Position
- Random Forest Model
 - Attacking Midfield
 - Center Midfield
 - Center Back
 - Attacking Midfield
 - Center Back
 - Center Forward/Second Striker
- Variable Importance Plot
 - Attacking Midfield
 - Winger
- Important Variables
 - Attacking Midfield
 - Center Midfield
 - Centre Back
 - Center Forward/Second Striker
 - Right/Left Back
 - Right/Left Midfield
 - Right/Left Winger
- Linear Regression (Coming Soon)
 - Attacking Midfield
 - Center Midfield
 - Center Back
 - Centre Forward and Second Striker
 - Right/Left Back
 - Right/Left Midfield
 - Right/Left Winger
- Appendix A: Important Variables
- Appendix B: Tables:
 - Standard

- Goalkeeping
- Advanced Goalkeeping
- Shooting
- Passing
- Pass Types
- Goal and Shot Creation
- Defensive Actions
- Possessions
- Playing Time
- Misc
- Market Values Table

Here are all the libraries we are using.

```
# loading all libraries
library(tidyverse)
library(dplyr)
library(tidyr)
library(worldfootballR)
library(utis)
library(rsample)
library(glmnet)
library(glmnetUtils)
library(forcats)
library(rsample)
library(ggplot2)
library(sjPlot)
library(partykit)
library(titanic)
library(PerformanceAnalytics)
library(rpart)
library(rpart.plot)
library(randomForest)

# note, do not run install.packages() inside a code chunk. install them in the console outside of a code chunk.
```

Introduction

For this study, data was collected from FBref.com on the “Big 5” Leagues. Our data includes statistics in categories such as standard , shooting , passing , goal and shot creation , defense , miscellaneous , playing time , and market values . We compare a player’s statistics with their market values listed on TransferMarkt.com and find the statistics that are most correlated with a player’s market values. In other words, we are aiming to find the certain statistics that makes a player valuable.

Here we are loading the tables by calling the function `fb_big5_advanced_season_stats` from our `worldfootballR` package. We are taking data from 2018 to 2020. The chunks are split up because for computational purposes.

#Loading the data sets

```
standard <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "standard", team_or_player = "player")
```

```
shooting <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "shooting", team_or_player = "player")
```

```
passing <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "passing", team_or_player = "player")
```

```
passingtypes <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "passing_types", team_or_player = "player")
```

```
gca <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "gca", team_or_player = "player")
```

```
defense <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "defense", team_or_player = "player")
```

```
possession <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "possession", team_or_player = "player")
```

```
misc <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "misc", team_or_player = "player")
playingtime <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "playing_time", team_or_player = "player")
keepers <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "keepers", team_or_player = "player")
keepers_adv <- fb_big5_advanced_season_stats(season_end_year = c(2019:2021), stat_type = "keepers_adv", team_or_player = "player")
```

Here we are loading the market values table from transfermarkt.com by calling the `get_player_market_values` function.

```
market_values18 <- tm_player_market_values(country_name = c("England", "Spain", "France", "Italy", "Germany"), start_year = 2018)
```

```
market_values19 <- tm_player_market_values(country_name = c("England", "Spain", "France", "Italy", "Germany"), start_year = 2019)
```

```
market_values20 <- tm_player_market_values(country_name = c("England", "Spain", "France", "Italy", "Germany"), start_year = 2020)
```

```
market_values <- rbind(market_values18, market_values19, market_values20)
```

Renaming variables.

```
# Renaming the variables
```

```
standard <- standard %>%  
  rename(Matches_Played = MP_Playing,  
         Starts = Starts_Playing,  
         Min = Min_Playing,  
         Min_per_90 = Mins_Per_90_Playing,  
         xG = xG_Expected,  
         npxG = npxG_Expected,  
         xAG = xAG_Expected,  
         'npxG+xAG' = 'npxG+xAG_Expected')  
  
shooting <- shooting %>%  
  rename(Gls = Gls_Standard,  
         Sh = Sh_Standard,  
         SoT = SoT_Standard,  
         SoT_percent = SoT_percent_Standard,  
         SoT_per_90 = SoT_per_90_Standard,  
         Sh_per_90_Standard = Sh_per_90_Standard,  
         SoT_per_90 = SoT_per_90_Standard,  
         G_per_Sh = G_per_Sh_Standard,  
         G_per_SoT = G_per_SoT_Standard,  
         Dist = Dist_Standard,  
         FK = FK_Standard,  
         PK = PK_Standard,  
         PKatt = PKatt_Standard,  
         xG = xG_Expected,  
         npxG = npxG_Expected,  
         npxG_per_Sh = npxG_per_Sh_Expected,  
         G_minus_xG = G_minus_xG_Expected,  
         npG_minus_xG = 'np:G_minus_xG_Expected')
```

```
defense <- defense %>%  
  rename(  
    Tkl = Tkl_Tackles,  
    TklW = TklW_Tackles,  
    Def_3rd = 'Def 3rd_Tackles',  
    Mid_3rd = 'Mid 3rd_Tackles',  
    Att_3rd = 'Att 3rd_Tackles',
```

```
Tkl_vs_dribble = Tkl_Vs,  
Att_vs_dribble = Att_Vs,  
Successful_Pressure_Percent = Tkl_percent_Vs,  
#Def_3rd_Pressures = 'Def 3rd_Pressures',  
#Mid_3rd_Pressures = 'Mid 3rd_Pressures',  
#Att_3rd_Pressures = 'Att 3rd_Pressures',  
Blocks = Blocks_Blocks)  
  
possession <- possession %>%  
  rename(  
    Touches = Touches_Touches,  
    Def_3rd_Touches = 'Def 3rd_Touches',  
    Mid_3rd_Touches = 'Mid 3rd_Touches',  
    Att_3rd_Touches = 'Att 3rd_Touches',  
    Att_Pen_Touches = 'Att Pen_Touches')  
    #Carries = Carries_Carries  
  
keepers <- keepers %>%  
  rename(Matches_Played = MP_Playing,  
    Starts = Starts_Playing,  
    Min = Min_Playing,  
    Min_per_90 = Mins_Per_90_Playing)  
  
keepers_adv <- keepers_adv %>%  
  rename(GA = GA_Goals,  
    PKA = PKA_Goals,  
    FK = FK_Goals,  
    CK = CK_Goals,  
    OG = OG_Goals,  
    PSxG = PSxG_Expected,  
    PSxG_per_SoT = PSxG_per_SoT_Expected,  
    PSxG_plus_minus = `PSxG+_per__minus__Expected`,  
    per_90 = `_per_90_Expected`)  
  
passing <- passing %>%  
  rename(Cmp = Cmp_Total,  
    Cmp_percent = Cmp_percent_Total,  
    TotDist = TotDist_Total,  
    PrgDist = PrgDist_Total)
```

```

playingtime <- playingtime %>%
  rename(Matches_Played = MP_Playing.Time,
         Min = Min_Playing.Time,
         Min_per_90 = Mins_Per_90_Playing.Time,
         Starts = Starts_Starts,
         Min_per_Match_Start = Mn_per_Start_Starts,
         Compl = Compl_Starts,
         Subs = Subs_Subs,
         Mn_per_Sub = Mn_per_Sub_Subs,
         unSub = unSub_Subs,
         PPM = PPM_Team.Success,
         onG = onG_Team.Success,
         onGA = onGA_Team.Success,
         plus_per_minus = plus_per__minus__Team.Success,
         plus_per_minus_per_90 = plus_per__minus_90_Team.Success,
         On_minus_Off = On_minus_Off_Team.Success,
         onxG = onxG_Team.Success..xG.,
         onxGA = onxGA_Team.Success..xG,
         xGplus_per_minus = xGplus_per__minus__Team.Success..xG,
         xGplus_per_minus_per_90 = xGplus_per__minus_90_Team.Success..xG,
         xG_On_minus_Off = On_minus_Off_Team.Success..xG)

```

Here we are unclassing all the variables and then setting the variables with string values as factors. This stores less data and allows us to run models on the data.

```

# Storing the character data types as factors to save more storage and call objects more efficiently
market_values <- as.data.frame(unclass(market_values),stringsAsFactors = TRUE)
defense <- as.data.frame(unclass(defense),stringsAsFactors = TRUE)
gca <- as.data.frame(unclass(gca),stringsAsFactors = TRUE)
keepers <- as.data.frame(unclass(keepers),stringsAsFactors = TRUE)
keepers_adv <- as.data.frame(unclass(keepers_adv),stringsAsFactors = TRUE)
misc <- as.data.frame(unclass(misc),stringsAsFactors = TRUE)
passing <- as.data.frame(unclass(passing),stringsAsFactors = TRUE)
passingtypes <- as.data.frame(unclass(passingtypes),stringsAsFactors = TRUE)
playingtime <- as.data.frame(unclass(playingtime),stringsAsFactors = TRUE)
possession <- as.data.frame(unclass(possession), stringsAsFactors = TRUE)
shooting <- as.data.frame(unclass(shooting), stringsAsFactors = TRUE)
standard <- as.data.frame(unclass(standard), stringsAsFactors = TRUE)

```

We are going to filter the players by position. Here are the different positions listed on Transfermarkt and the number of players at the respective position.

```
table(market_values$player_position)
##
##           Attack Attacking Midfield   Central Midfield
##                2                656                1562
##           Centre-Back   Centre-Forward           Defence
##                2003                1530                3
## Defensive Midfield           Goalkeeper   Left Midfield
##                861                1284                87
##           Left Winger           Left-Back           Midfield
##                676                833                4
##           Right Midfield   Right Winger           Right-Back
##                59                722                897
##           Second Striker
##                102
```

Notice that there are some 'strange' values at some positions. Specifically:

attack : 1 player

midfield : 5 players

defence : 2 players

```
strange <- market_values %>%
  select(player_name, comp_name, player_position, current_club, player_market_value_euro, player_age) %>%
  filter(player_position == 'midfield' | player_position == 'attack' | player_position == 'defence')

head(strange, 8)
## [1] player_name           comp_name
## [3] player_position       current_club
## [5] player_market_value_euro player_age
## <0 rows> (or 0-length row.names)
```

Are these players we can remove? Non-relevant players? Most players we don't even have a valuation. We will not include these players.

We are creating a unique identifier for each table here. This is so that we can use the identifier to merge the tables and join them together later on.


```
# Here we are creating a unique identifier for each table so they can be joined

market_values <- market_values %>%
  mutate(Season_End_Year = season_start_year + 1,
         PlayerYearComp_id = paste(player_name, Season_End_Year, comp_name)) %>% drop_na(player_market_value_euro)

euro_val <- market_values %>% select(PlayerYearComp_id, player_market_value_euro, player_position)

standard <- standard %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

shooting <- shooting %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

passing <- passing %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

passingtypes <- passingtypes %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

gca <- gca %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

defense <- defense %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

possession <- possession %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

playingtime <- playingtime %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

misc <- misc %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))

keepers <- keepers %>%
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))
```

```
keepers_adv <- keepers_adv %>%  
  mutate(PlayerYearComp_id = paste(Player, Season_End_Year, Comp))  
  
#making new dataframe to import market values euro
```

Joining Tables

Here we are joining and merging all the data sets together using the `inner join` function. We named this table master, and for duplicate values, we added a suffix: `REMOVEDUPLICATE`, which will appear in all of the duplicate variables. Then, a code is executed to remove anything containing those string of letters. This merges all our tables together and removes duplicate variables.

```
# Here we are merging the relevant tables to goal keepers
# For every duplicate variable we are adding a suffix on the end ".REMOVEDUPLICATE"
# We will write a code later that removes any variable with this suffix, thus removing all the variables

# Standard
# Shooting
euro_val2 <- inner_join(x = euro_val,
                        y = shooting,
                        by = "PlayerYearComp_id",
                        suffix = c("", ".REMOVEDUPLICATE"))

# Passing
euro_val3 <- inner_join(x = euro_val2,
                        y = passing,
                        by = "PlayerYearComp_id",
                        suffix = c("", ".REMOVEDUPLICATE"))

# Passing Types
euro_val4 <- inner_join(x = euro_val3,
                        y = passingtypes,
                        by = "PlayerYearComp_id",
                        suffix = c("", ".REMOVEDUPLICATE"))

# GCA
euro_val5 <- inner_join(x = euro_val4,
                        y = gca,
                        by = "PlayerYearComp_id",
                        suffix = c("", ".REMOVEDUPLICATE"))

# Defense
euro_val6 <- inner_join(x = euro_val5,
                        y = defense,
                        by = "PlayerYearComp_id",
                        suffix = c("", ".REMOVEDUPLICATE"))

# Possession
euro_val7 <- inner_join(x = euro_val6,
                        y = possession,
                        by = "PlayerYearComp_id",
```

```
suffix = c("", ".REMOVEDUPLICATE"))

# Playing Time
master <- inner_join(x = euro_val7,
                    y = playingtime,
                    by = "PlayerYearComp_id",
                    suffix = c("", ".REMOVEDUPLICATE"))

# Market Values
#master <- inner_join(x = master,
                    #y = market_values,
                    #by = "PlayerYearComp_id",
                    #suffix = c("", ".REMOVEDUPLICATE"))

#Euro Values

master <- inner_join(x = master,
                    y = euro_val,
                    by = "PlayerYearComp_id",
                    suffix = c("", ".REMOVEDUPLICATE"))

# Removing duplicate variables
master <- master %>% select(-contains("REMOVEDUPLICATE"))
```

Filtering By Position

Let's separate each player by position.

```
attacking_midfield <- master %>%  
  filter(player_position == 'Attacking Midfield')  
attacking_midfield <- attacking_midfield %>% select(-contains("REMOVEDUPLICATE"))  
  
central_midfield <- master %>%  
  filter(player_position == 'Central Midfield')  
central_midfield <- central_midfield %>% select(-contains("REMOVEDUPLICATE"))  
  
centre_back <- master %>%  
  filter(player_position == 'Centre-Back')  
centre_back <- centre_back %>% select(-contains("REMOVEDUPLICATE"))  
  
centreForward_secondStriker <- master %>%  
  filter(player_position == 'Centre-Forward' | player_position == 'Second Striker')  
centreForward_secondStriker <- centreForward_secondStriker %>% select(-contains("REMOVEDUPLICATE"))  
  
right_left_back <- master %>%  
  filter(player_position == 'Right-Back' | player_position == 'Left-Back')  
right_left_back <- right_left_back %>% select(-contains("REMOVEDUPLICATE"))  
  
right_left_midfielder <- master %>%  
  filter(player_position == 'Right Midfield' | player_position == 'Left Midfield')  
right_left_midfielder <- right_left_midfielder %>% select(-contains("REMOVEDUPLICATE"))  
  
winger <- master %>%  
  filter(player_position == 'Right Winger' | player_position == 'Left Winger')  
winger <- winger %>% select(-contains("REMOVEDUPLICATE"))
```

Random Forest Model

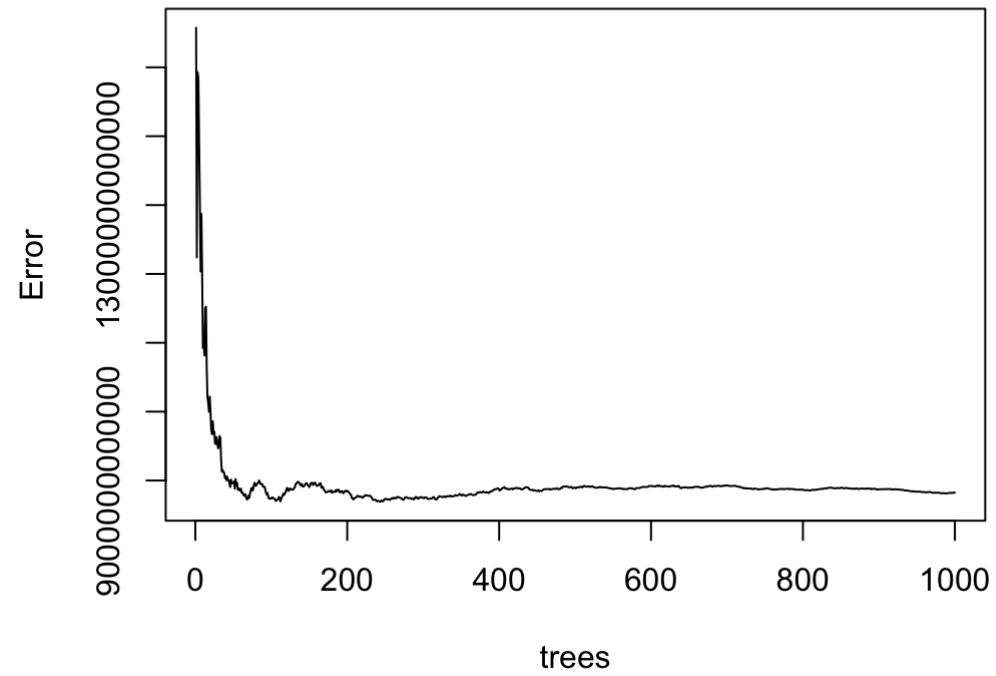
Attacking Midfield

```
# Removing unnecessary variables
attacking_midfield1 <- attacking_midfield %>%
  select(-PlayerYearComp_id, -Url, -Season_End_Year, -Squad,
        -Player, -Nation, -Pos, -Age, -Born, -Comp)
#attacking_midfield1 <- attacking_midfield1 %>% na.omit()

# Random Forest
RF_attacking_midfield <- randomForest(player_market_value_euro ~ .,
  data = attacking_midfield1,
  type = regression,
  mtry = 5,
  na.action = na.roughfix,
  ntree = 1000)

# Plot
print(RF_attacking_midfield)
##
## Call:
## randomForest(formula = player_market_value_euro ~ ., data = attacking_midfield1, type = regression, mtry
= 5, ntree = 1000, na.action = na.roughfix)
##           Type of random forest: regression
##           Number of trees: 1000
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 8823358056701
##           % Var explained: 86.72
plot(RF_attacking_midfield)
```

RF_attacking_midfield



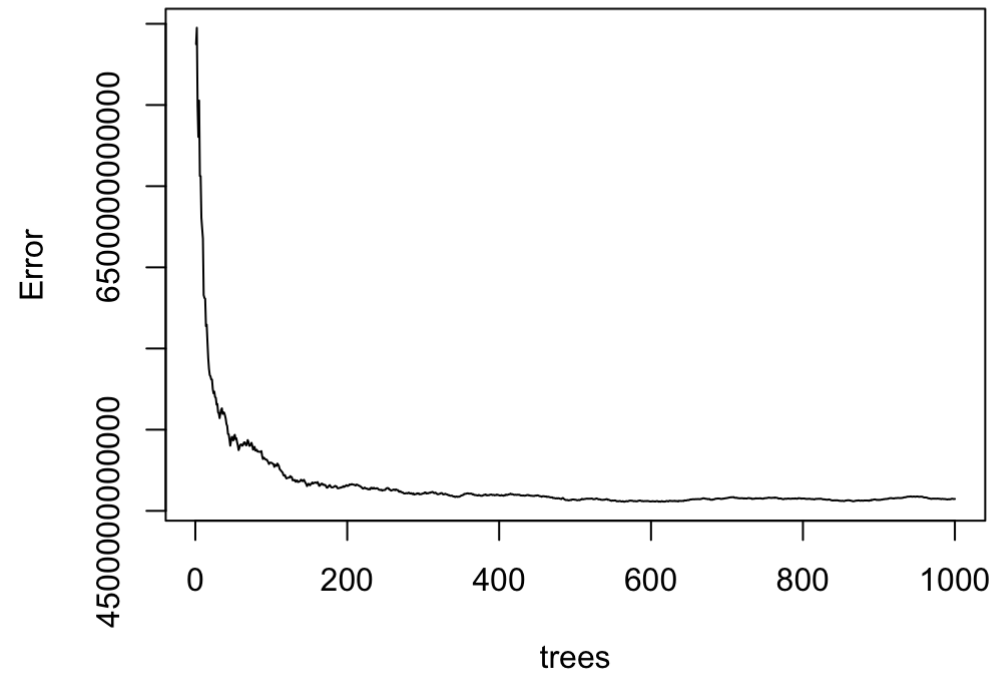
Center Midfield

```
# Removing unnecessary variables
central_midfield1 <- central_midfield %>%
  select(-PlayerYearComp_id, -Url, -Season_End_Year, -Squad,
         -Player, -Nation, -Pos, -Age, -Born, -Comp)

# Random Forest
RF_central_midfield <- randomForest(player_market_value_euro ~ .,
                                     data = central_midfield1,
                                     type = regression,
                                     mtry = 5,
                                     na.action = na.roughfix,
                                     ntree = 1000)

# Plot
print(RF_central_midfield)
##
## Call:
## randomForest(formula = player_market_value_euro ~ ., data = central_midfield1, type = regression, mtry =
5, ntree = 1000, na.action = na.roughfix)
##              Type of random forest: regression
##              Number of trees: 1000
## No. of variables tried at each split: 5
##
##              Mean of squared residuals: 4573399549449
##              % Var explained: 87.89
plot(RF_central_midfield)
```


RF_central_midfield



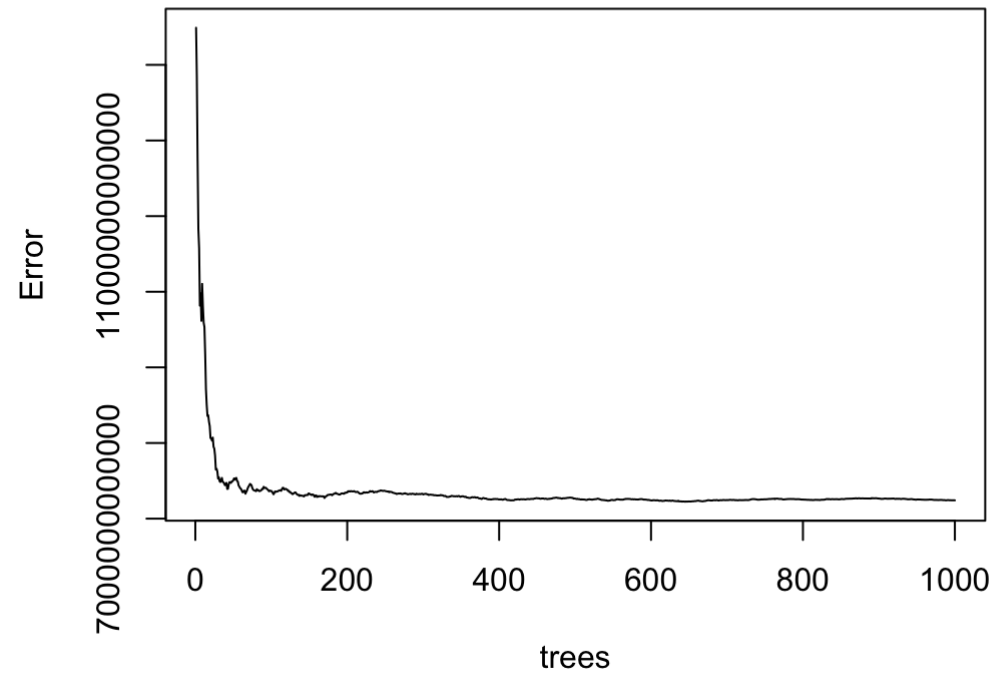
Center Back

```
# Removing unnecessary variables
centre_back1 <- centre_back %>%
  select(-PlayerYearComp_id, -Url, -Season_End_Year, -Squad,
        -Player, -Nation, -Pos, -Age, -Born, -Comp)

# Random Forest
RF_centre_back <- randomForest(player_market_value_euro ~ .,
                              data = centre_back1,
                              type = regression,
                              mtry = 5,
                              na.action = na.roughfix,
                              ntree = 1000)

# Plot
print(RF_centre_back)
##
## Call:
## randomForest(formula = player_market_value_euro ~ ., data = centre_back1, type = regression, mtry = 5, n
## tree = 1000, na.action = na.roughfix)
##              Type of random forest: regression
##              Number of trees: 1000
## No. of variables tried at each split: 5
##
##              Mean of squared residuals: 7241728138201
##              % Var explained: 73.46
plot(RF_centre_back)
```

RF_centre_back



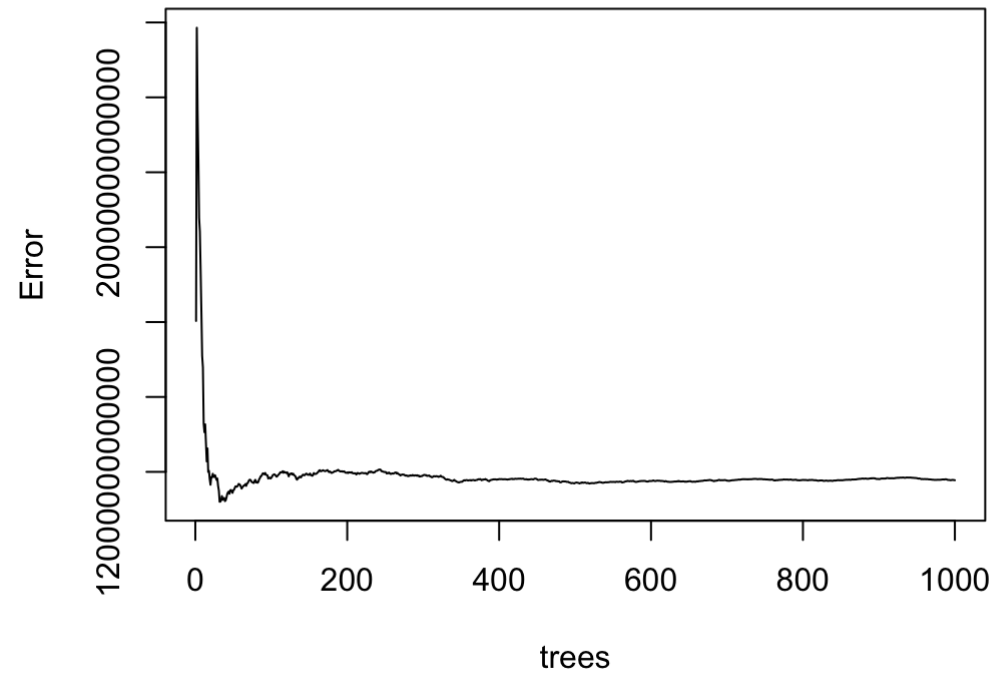
Center Forward/Second Striker

```
# Removing unnecessary variables
centreForward_secondStriker1 <- centreForward_secondStriker %>%
  select(-PlayerYearComp_id, -Url, -Season_End_Year, -Squad,
        -Player, -Nation, -Pos, -Age, -Born, -Comp)

# Random Forest
RF_centreForward_secondStriker <- randomForest(player_market_value_euro ~ .,
  data = centreForward_secondStriker1,
  type = regression,
  mtry = 5,
  na.action = na.roughfix,
  ntree = 1000)

# Plot
print(RF_centreForward_secondStriker)
##
## Call:
## randomForest(formula = player_market_value_euro ~ ., data = centreForward_secondStriker1,      type = regress
ion, mtry = 5, ntree = 1000, na.action = na.roughfix)
##              Type of random forest: regression
##              Number of trees: 1000
## No. of variables tried at each split: 5
##
##              Mean of squared residuals: 11776597147994
##              % Var explained: 78.44
plot(RF_centreForward_secondStriker)
```

RF_centreForward_secondStriker



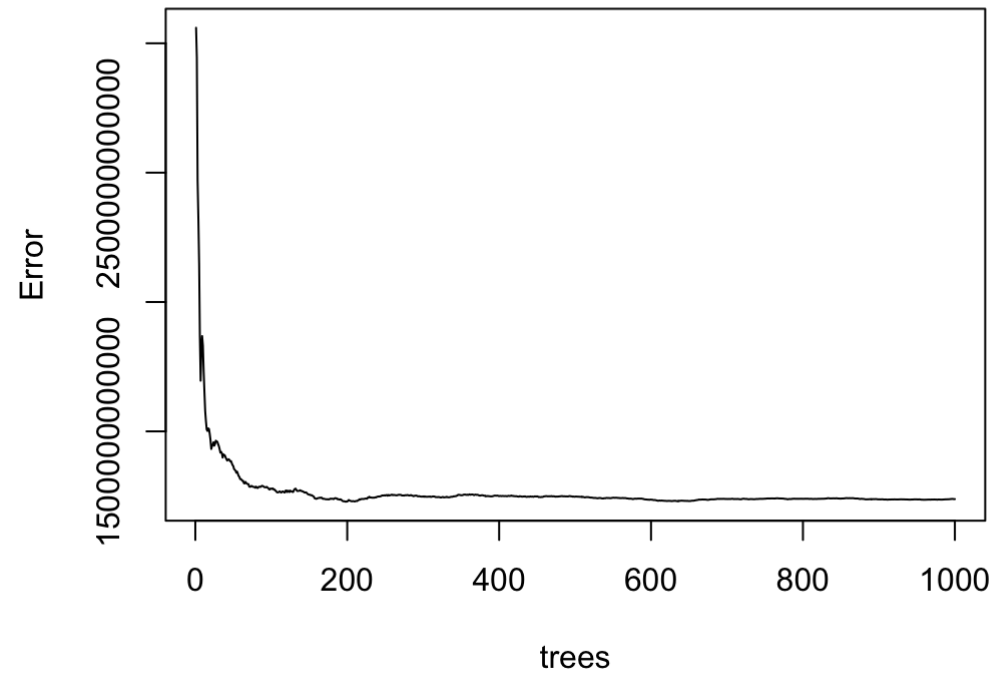
Right/Left Back

```
# Removing unnecessary variables
right_left_back1 <- right_left_back %>%
  select(-PlayerYearComp_id, -Url, -Season_End_Year, -Squad,
         -Player, -Nation, -Pos, -Age, -Born, -Comp)

# Random Forest
RF_right_left_back <- randomForest(player_market_value_euro ~ .,
                                   data = right_left_back1,
                                   type = regression,
                                   mtry = 5,
                                   na.action = na.roughfix,
                                   ntree = 1000)

# Plot
print(RF_right_left_back)
##
## Call:
## randomForest(formula = player_market_value_euro ~ ., data = right_left_back1, type = regression, mtry =
5, ntree = 1000, na.action = na.roughfix)
##              Type of random forest: regression
##              Number of trees: 1000
## No. of variables tried at each split: 5
##
##              Mean of squared residuals: 1238269101480
##              % Var explained: 92.41
plot(RF_right_left_back)
```

RF_right_left_back

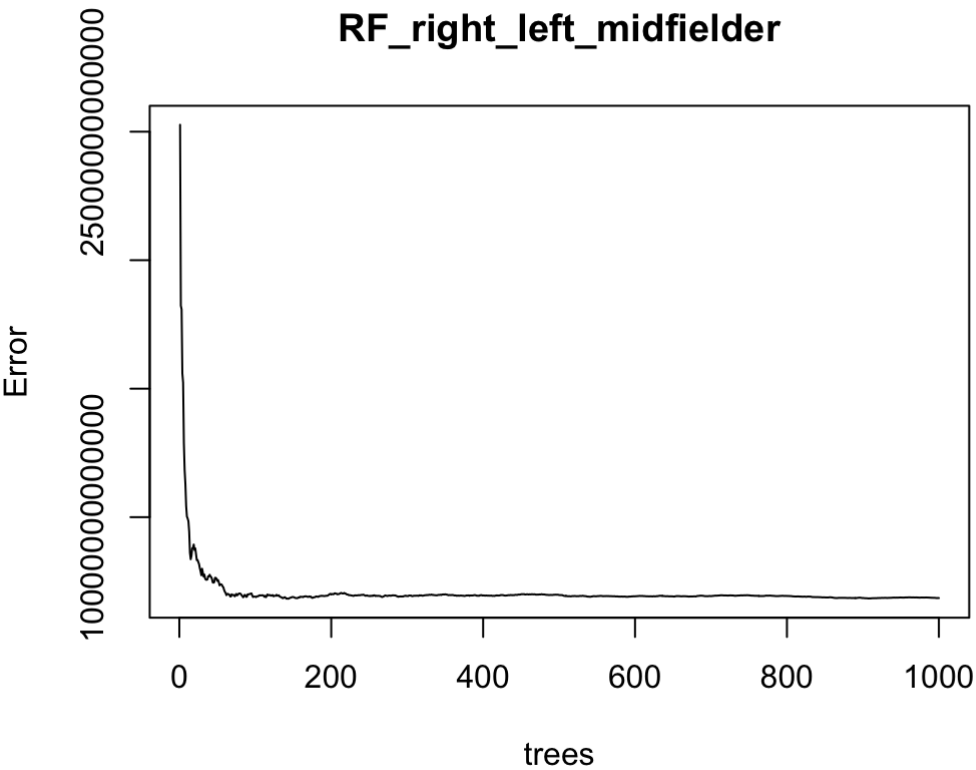


Right/Left Midfielders

```
# Removing unnecessary variables
right_left_midfielder1 <- right_left_midfielder %>%
  select(-PlayerYearComp_id, -Url, -Season_End_Year, -Squad,
         -Player, -Nation, -Pos, -Age, -Born, -Comp)

# Random Forest
RF_right_left_midfielder <- randomForest(player_market_value_euro ~ .,
                                         data = right_left_midfielder1,
                                         type = regression,
                                         mtry = 5,
                                         na.action = na.roughfix,
                                         ntree = 1000)

# Plot
print(RF_right_left_midfielder)
##
## Call:
## randomForest(formula = player_market_value_euro ~ ., data = right_left_midfielder1,      type = regression, m
try = 5, ntree = 1000, na.action = na.roughfix)
##              Type of random forest: regression
##              Number of trees: 1000
## No. of variables tried at each split: 5
##
##              Mean of squared residuals: 6852294406044
##              % Var explained: 54.65
plot(RF_right_left_midfielder)
```

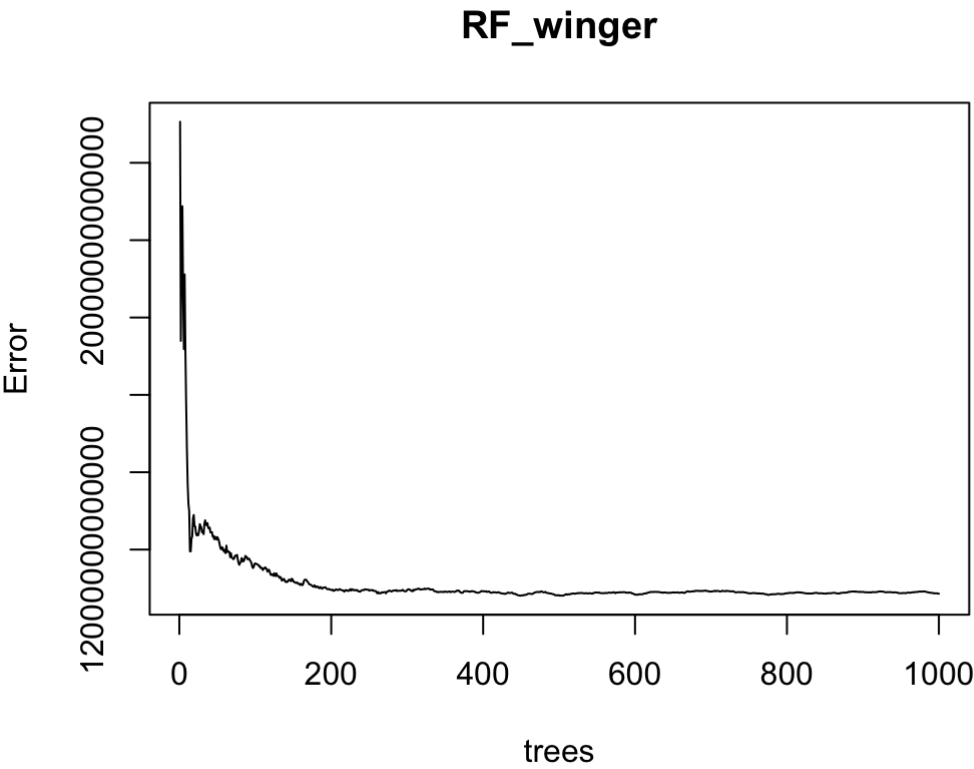



Right/Left Wingers

```
# Removing unnecessary variables
winger1 <- winger %>%
  select(-PlayerYearComp_id, -Url, -Season_End_Year, -Squad,
         -Player, -Nation, -Pos, -Age, -Born, -Comp)

# Random Forest
RF_winger <- randomForest(player_market_value_euro ~ .,
                          data = winger1,
                          type = regression,
                          mtry = 5,
                          na.action = na.roughfix,
                          ntree = 1000)

# Plot
print(RF_winger)
##
## Call:
## randomForest(formula = player_market_value_euro ~ ., data = winger1,      type = regression, mtry = 5, ntree
= 1000, na.action = na.roughfix)
##              Type of random forest: regression
##              Number of trees: 1000
## No. of variables tried at each split: 5
##
##              Mean of squared residuals: 10857382672546
##              % Var explained: 94.62
plot(RF_winger)
```

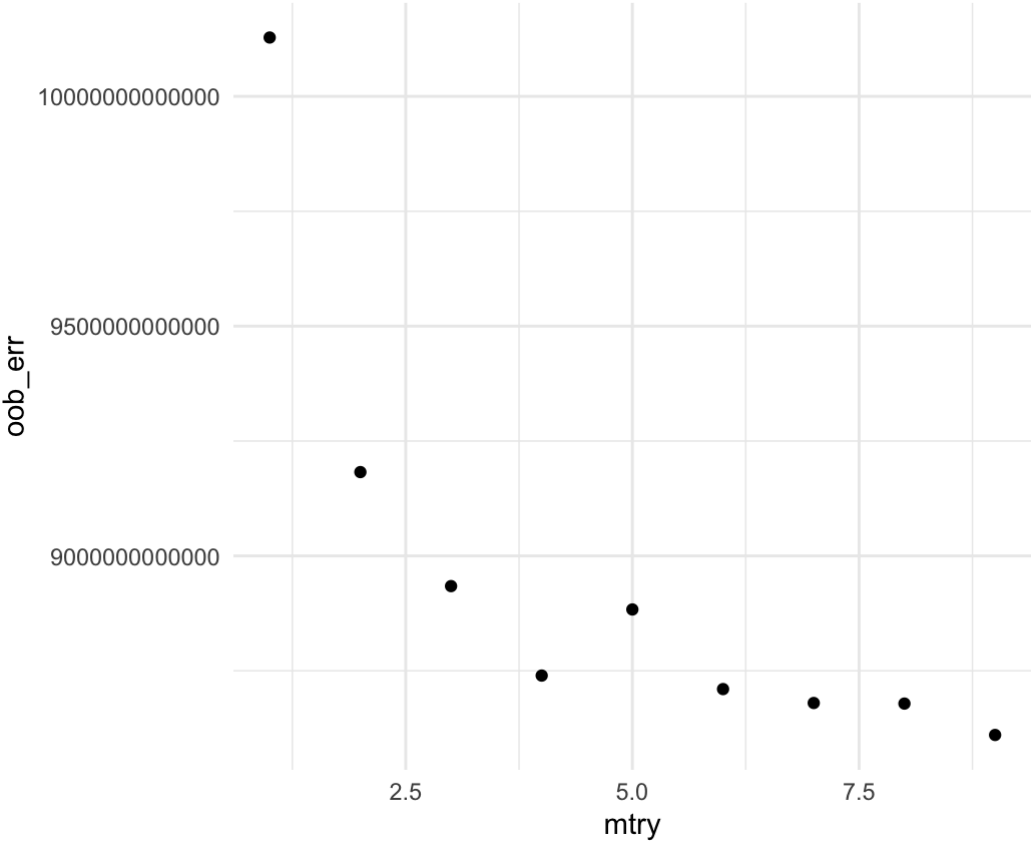


Model Tuning

Attacking Midfield

```
# All Variables
rf_mods <- list()
oob_err <- NULL
test_err <- NULL
for(mtry in 1:9){
  RFfit_attacking_midfield <- randomForest(player_market_value_euro ~
    .,
    data = attacking_midfield1,
    mtry = mtry,
    na.action = na.roughfix,
    ntree = 1000)
  oob_err[mtry] <- RFfit_attacking_midfield$mse[1000]

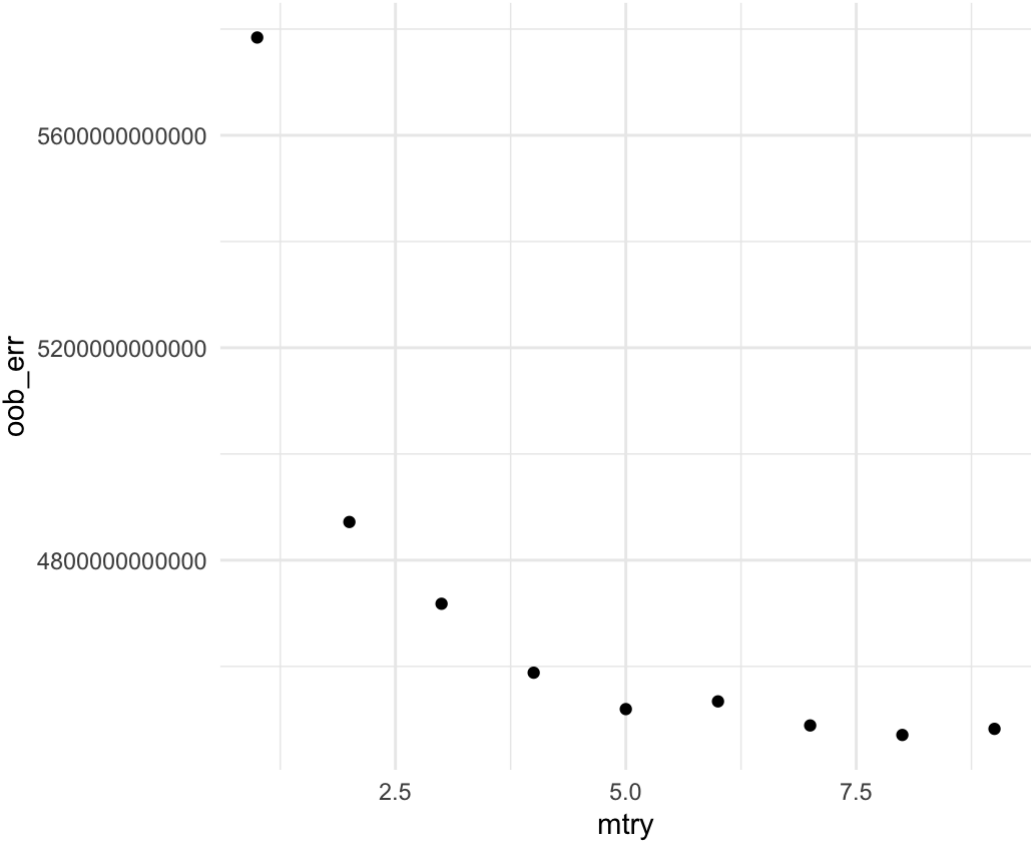
  cat(mtry, " ")
}
## 1  2  3  4  5  6  7  8  9
## 1  2  3  4  5  6  7  8  9
results_DF <- data.frame(mtry = 1:9, oob_err)
ggplot(results_DF, aes(x = mtry, y = oob_err)) + geom_point() + theme_minimal() + xlim(1,9)
```



Center Midfield

```
# All Variables
rf_mods <- list()
oob_err <- NULL
test_err <- NULL
for(mtry in 1:9){
  RFfit_central_midfield <- randomForest(player_market_value_euro ~
    .,
    data = central_midfield1,
    mtry = mtry,
    na.action = na.roughfix,
    ntree = 1000)
  oob_err[mtry] <- RFfit_central_midfield$mse[1000]

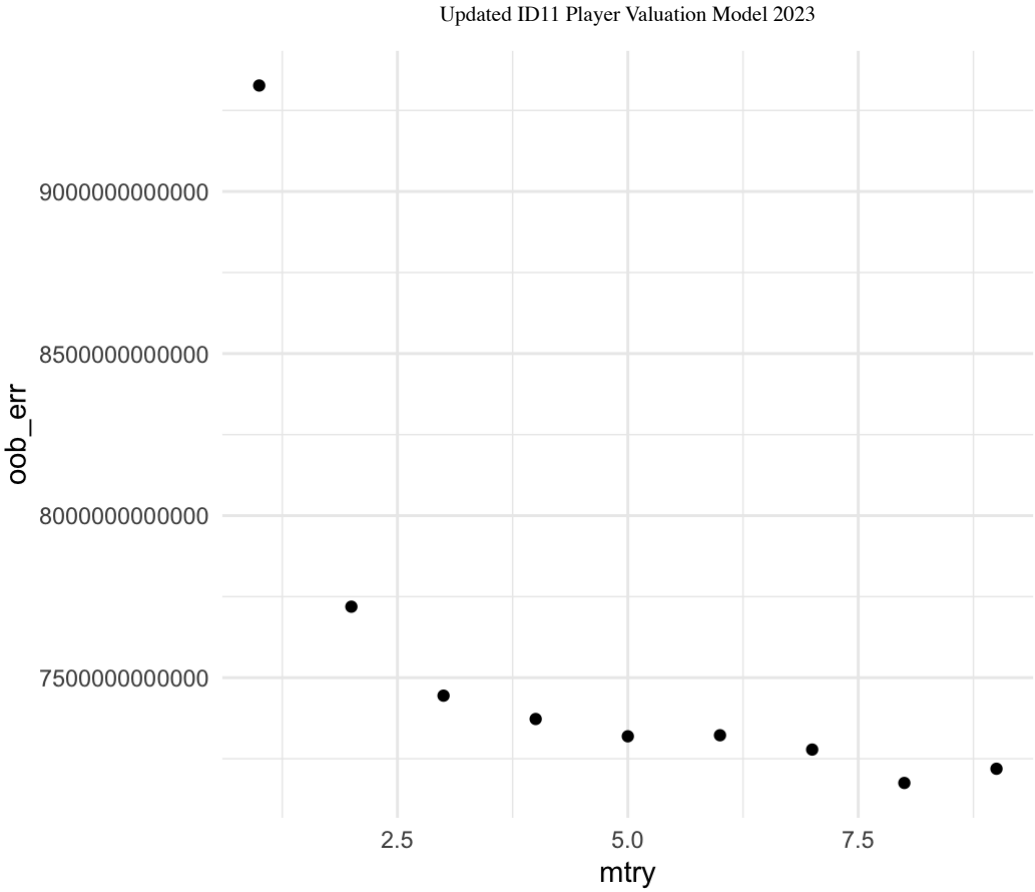
  cat(mtry, " ")
}
## 1 2 3 4 5 6 7 8 9
## 1 2 3 4 5 6 7 8 9
results_DF <- data.frame(mtry = 1:9, oob_err)
ggplot(results_DF, aes(x = mtry, y = oob_err)) + geom_point() + theme_minimal() + xlim(1,9)
```



Center Back

```
# All Variables
rf_mods <- list()
oob_err <- NULL
test_err <- NULL
for(mtry in 1:9){
  RFfit_centre_back <- randomForest(player_market_value_euro ~
    .,
    data = centre_back1,
    mtry = mtry,
    na.action = na.roughfix,
    ntree = 1000)
  oob_err[mtry] <- RFfit_centre_back$mse[1000]

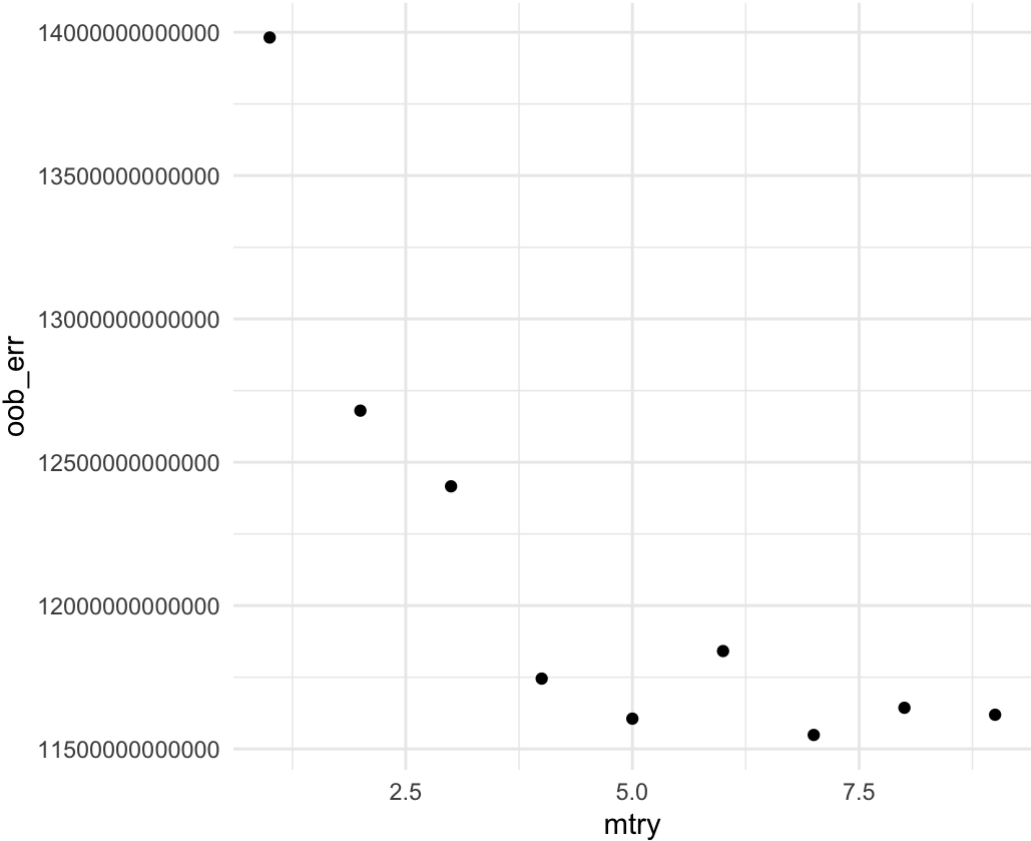
  cat(mtry, " ")
}
## 1  2  3  4  5  6  7  8  9
## 1  2  3  4  5  6  7  8  9
results_DF <- data.frame(mtry = 1:9, oob_err)
ggplot(results_DF, aes(x = mtry, y = oob_err)) + geom_point() + theme_minimal() + xlim(1,9)
```

Center Forward/Second Striker

```
# All Variables
rf_mods <- list()
oob_err <- NULL
test_err <- NULL
for(mtry in 1:9){
  RFfit_centreForward_secondStriker <- randomForest(player_market_value_euro ~
    .,
    data = centreForward_secondStriker1,
    mtry = mtry,
    na.action = na.roughfix,
    ntree = 1000)
  oob_err[mtry] <- RFfit_centreForward_secondStriker$mse[1000]

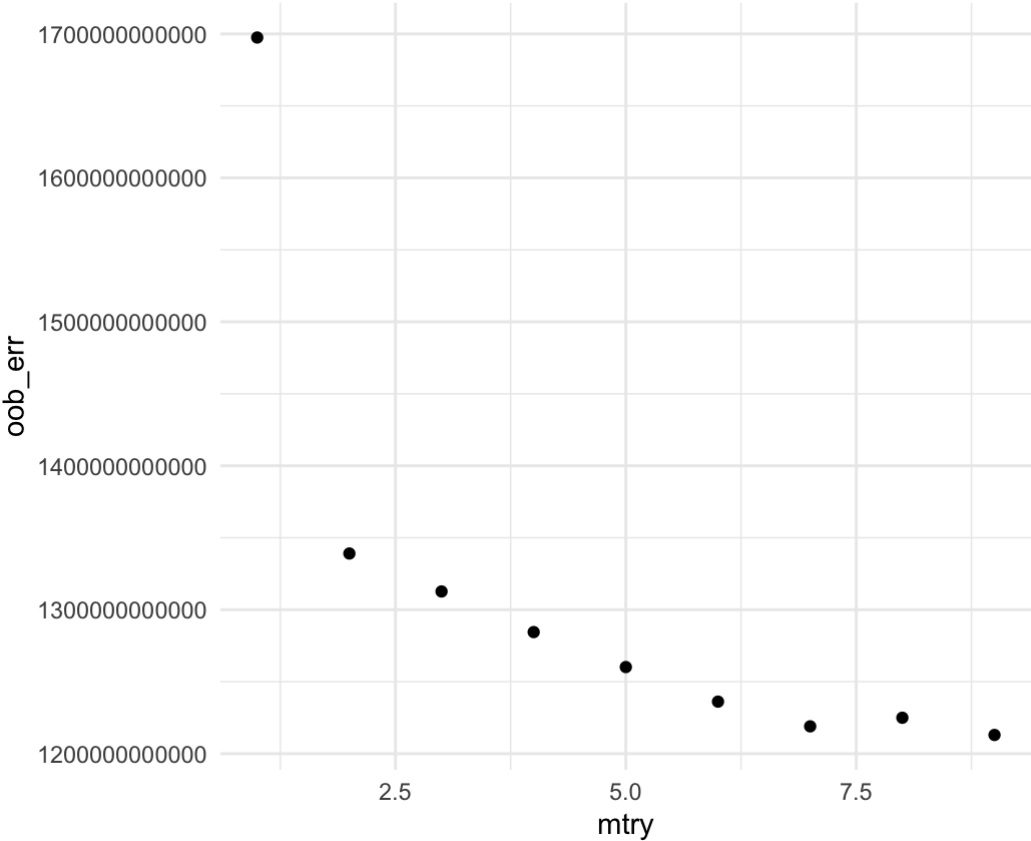
  cat(mtry, " ")
}
## 1 2 3 4 5 6 7 8 9
## 1 2 3 4 5 6 7 8 9
results_DF <- data.frame(mtry = 1:9, oob_err)
ggplot(results_DF, aes(x = mtry, y = oob_err)) + geom_point() + theme_minimal() + xlim(1,9)
```



Right/Left Back

```
# All Variables
rf_mods <- list()
oob_err <- NULL
test_err <- NULL
for(mtry in 1:9){
  RFfit_right_left_back <- randomForest(player_market_value_euro ~
    .,
    data = right_left_back1,
    mtry = mtry,
    na.action = na.roughfix,
    ntree = 1000)
  oob_err[mtry] <- RFfit_right_left_back$mse[1000]

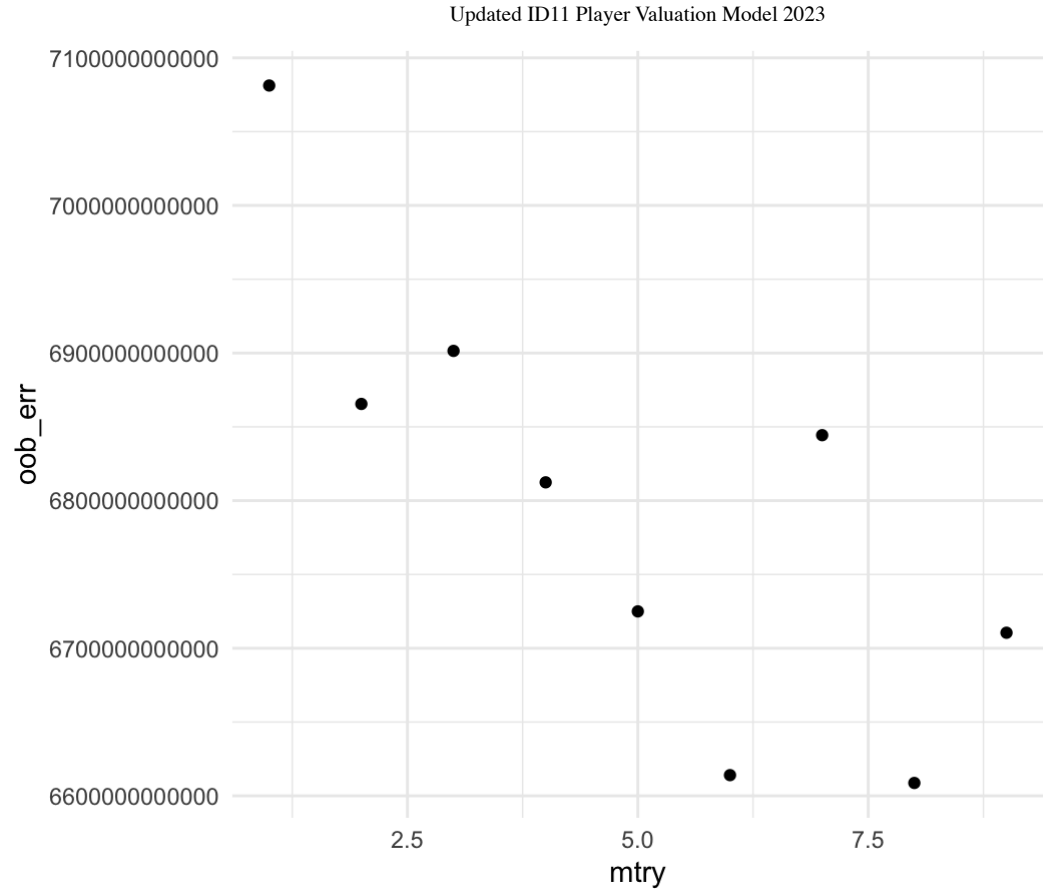
  cat(mtry, " ")
}
## 1 2 3 4 5 6 7 8 9
## 1 2 3 4 5 6 7 8 9
results_DF <- data.frame(mtry = 1:9, oob_err)
ggplot(results_DF, aes(x = mtry, y = oob_err)) + geom_point() + theme_minimal() + xlim(1,9)
```



Right/Left Midfield

```
# All Variables
rf_mods <- list()
oob_err <- NULL
test_err <- NULL
for(mtry in 1:9){
  RFfit_right_left_midfielder <- randomForest(player_market_value_euro ~
      .,
      data = right_left_midfielder1,
      mtry = mtry,
      na.action = na.roughfix,
      ntree = 1000)
  oob_err[mtry] <- RFfit_right_left_midfielder$mse[1000]

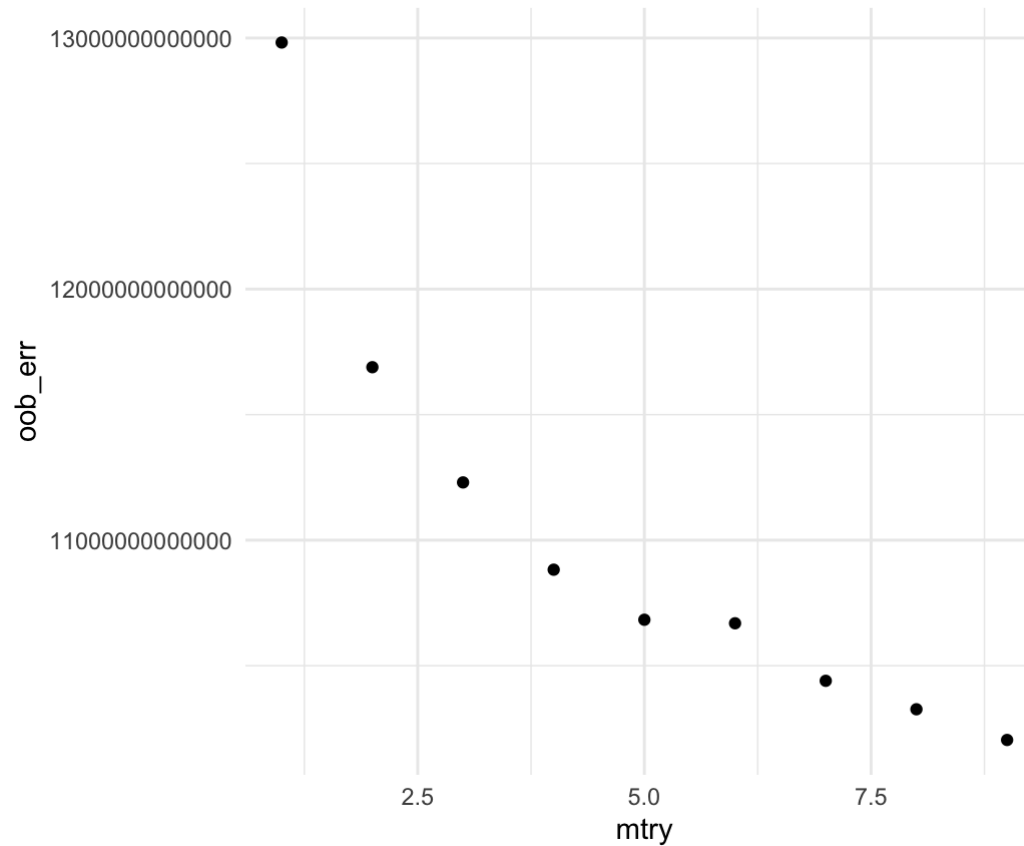
  cat(mtry, " ")
}
## 1 2 3 4 5 6 7 8 9
## 1 2 3 4 5 6 7 8 9
results_DF <- data.frame(mtry = 1:9, oob_err)
ggplot(results_DF, aes(x = mtry, y = oob_err)) + geom_point() + theme_minimal() + xlim(1,9)
```



Right/Left Winger

```
# All Variables
rf_mods <- list()
oob_err <- NULL
test_err <- NULL
for(mtry in 1:9){
  RFfit_winger <- randomForest(player_market_value_euro ~
                                .,
                                data = winger1,
                                mtry = mtry,
                                na.action = na.roughfix,
                                ntree = 1000)
  oob_err[mtry] <- RFfit_winger$mse[1000]

  cat(mtry, " ")
}
## 1 2 3 4 5 6 7 8 9
## 1 2 3 4 5 6 7 8 9
results_DF <- data.frame(mtry = 1:9, oob_err)
ggplot(results_DF, aes(x = mtry, y = oob_err)) + geom_point() + theme_minimal() + xlim(1,9)
```

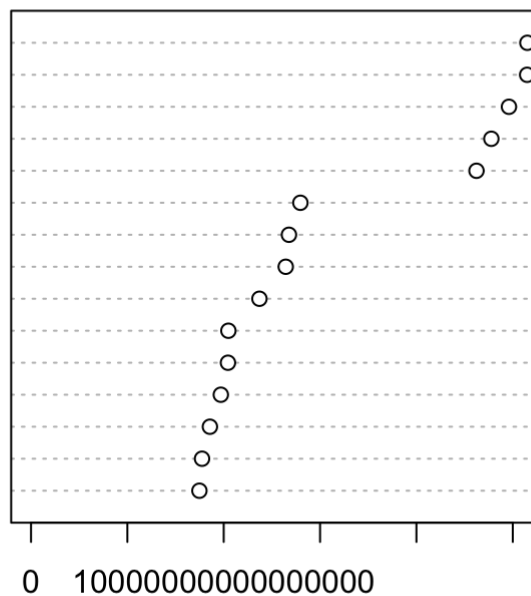
Variable Importance Plot

Attacking Midfield

```
varImpPlot(RFfit_attacking_midfield, n.var = 15)
```

RFfit_attacking_midfield

unSub
Att_Dribbles
Dist
Succ_Dribbles
npxG_per_Sh
Cmp_percent_Medium
Final_Third
Cmp_Medium
xGplus_per_minus
Cmp_Long
SoT_percent
GCA90_GCA
Drib_SCA
plus_per_minus
Cmp_percent

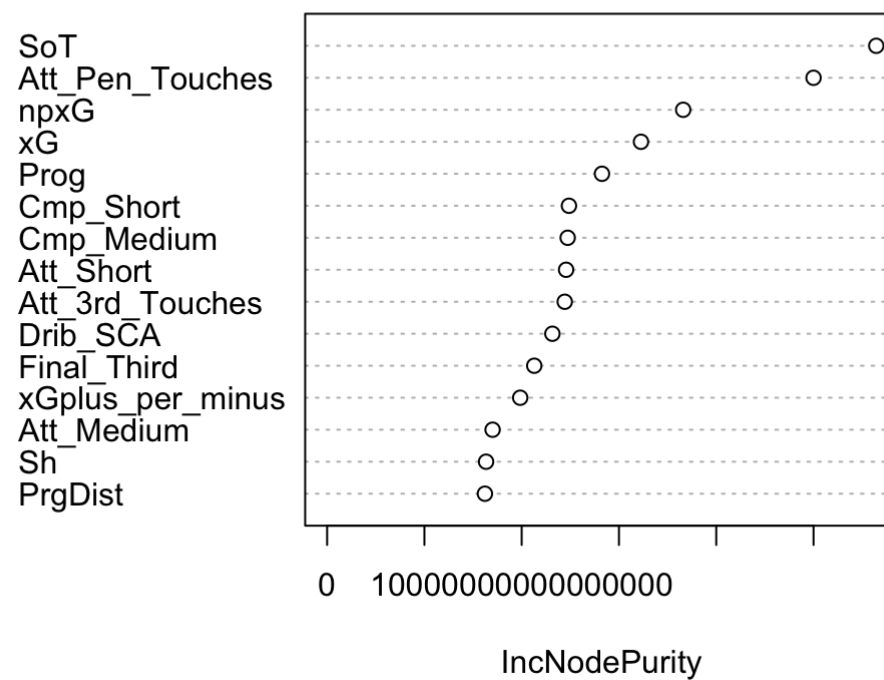


IncNodePurity

Center Midfield

```
varImpPlot(RFfit_central_midfield, n.var = 15)
```

RFfit_central_midfield

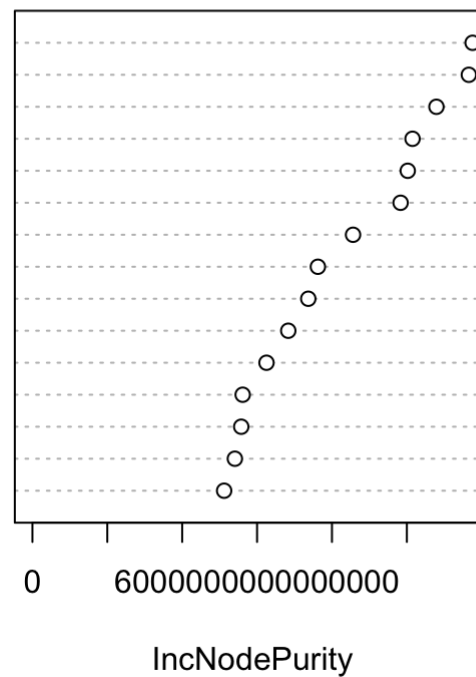


Center Back

```
varImpPlot(RFfit_centre_back, n.var = 15)
```

RFfit_centre_back

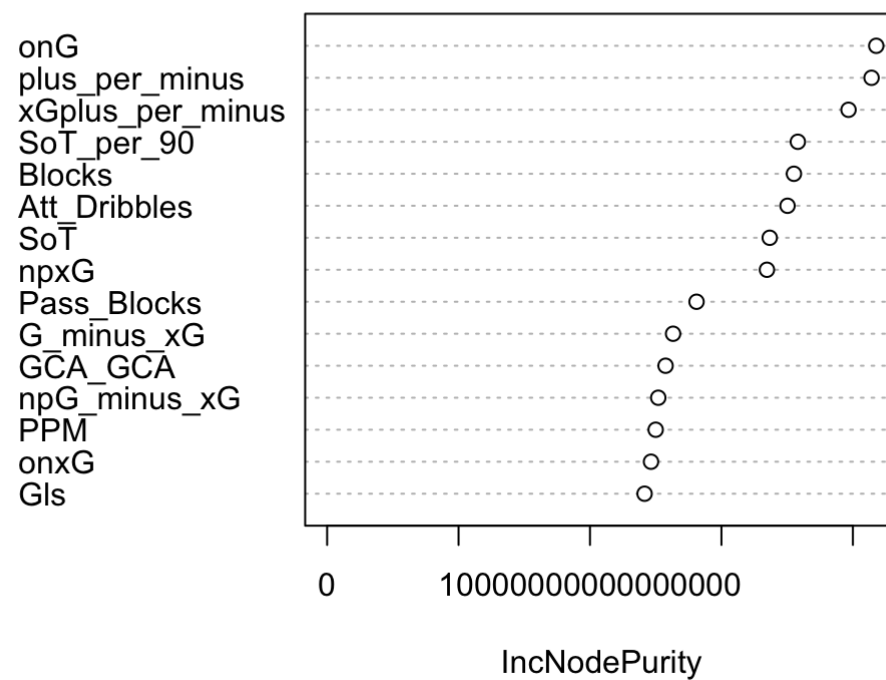
xGplus_per_minus
 onxG
 Tkl
 TklW
 plus_per_minus
 Mid_3rd
 TB_Pass
 Pass_Blocks
 onG
 Mid_3rd_Touches
 Prog
 Cmp_Short
 PrgDist
 xGplus_per_minus_per_90
 Cmp



Center Forward/Second Striker

```
varImpPlot(RFfit_centreForward_secondStriker, n.var = 15)
```

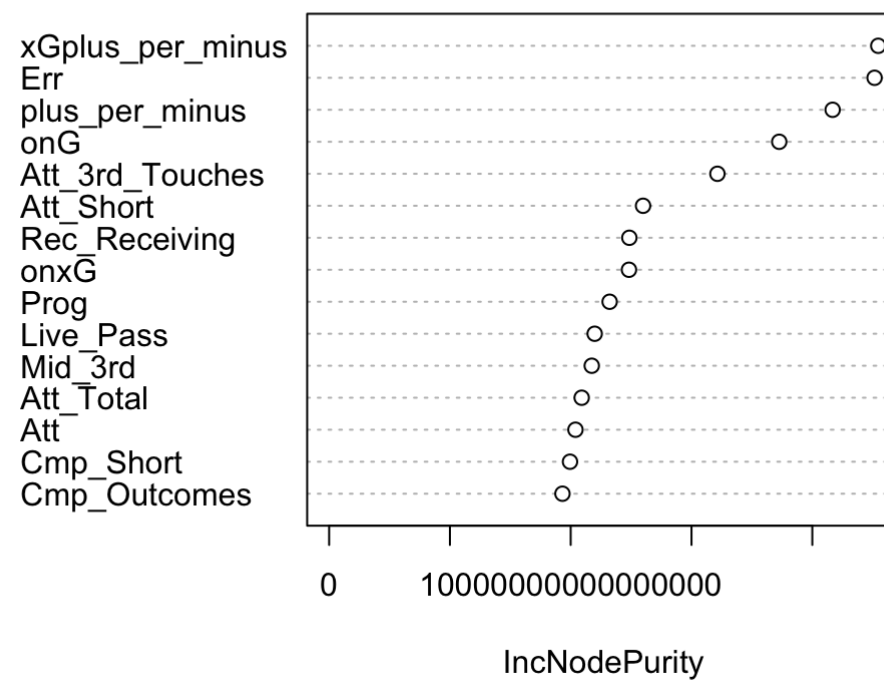
RFfit_centreForward_secondStriker



Right/Left Back

```
varImpPlot(RFfit_right_left_back, n.var = 15)
```

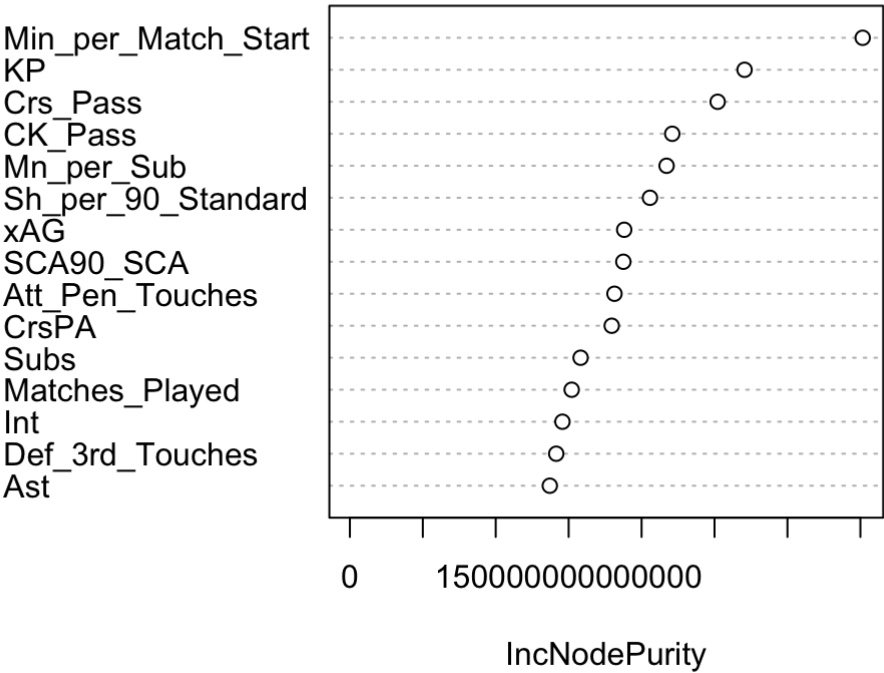
RFfit_right_left_back



Right/Left Midfield

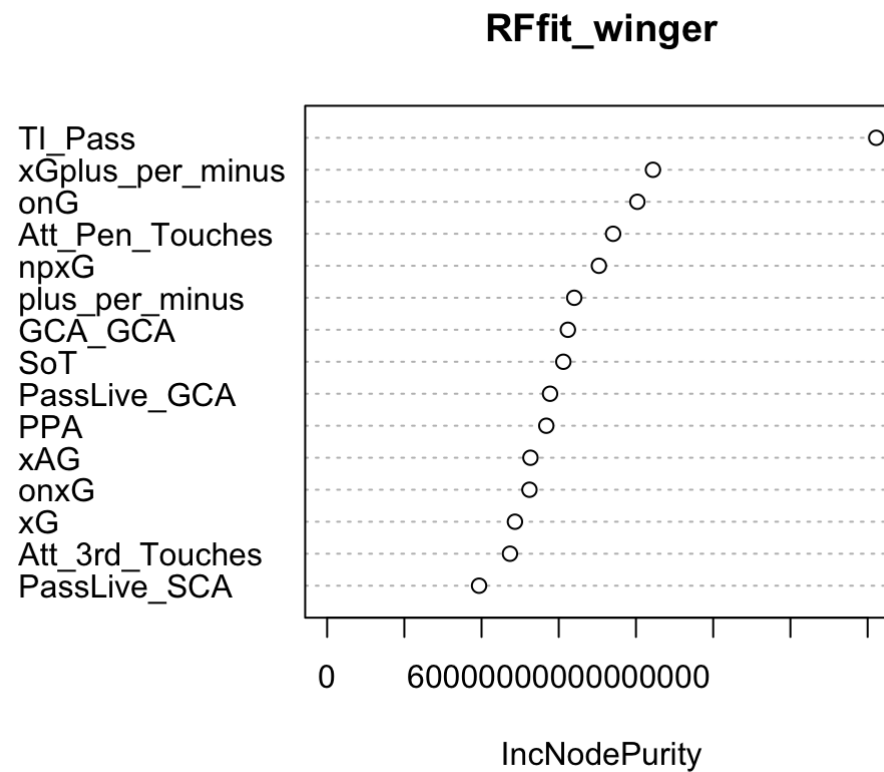
```
varImpPlot(RFfit_right_left_midfielder, n.var = 15)
```

RFfit_right_left_midfielder



Winger

```
varImpPlot(RFfit_winger, n.var = 15)
```



Important Variables

Attacking Midfield

xG +/- : Expected Goals Scored minus Expected Goals Allowed

GCA90_GCA : Goal Creating Actions per 90

Ast_Per :

Center Midfield

xG +/- : Expected Goals Scored minus Expected Goals Allowed

GCA90_GCA : Goal Creating Actions per 90

player_foot :

Centre Back

xG +/- : Expected Goals Scored minus Expected Goals Allowed

int :

Cmp_percent_Medium :

Mid_3rd_Touches

Center Forward/Second Striker

xG +/- : Expected Goals Scored minus Expected Goals Allowed

PPA :

GCA_GCA :

Right/Left Back

Comp

Mid_3rd

Low_Height

Right/Left Midfield

SoT_percent :

player_age :

TB_pass :

Comp :

Right/Left Winger

xG +/- : Expected Goals Scored minus Expected Goals Allowed

Gls :

Rec_receiving :

Linear Regression (Coming Soon)

Attacking Midfield

```

attackingMidfield_linReg <- lm(player_market_value_euro ~ GCA90_GCA + xGplus_per_minus,
                                data = attacking_midfield1)

summary(attackingMidfield_linReg)
##
## Call:
## lm(formula = player_market_value_euro ~ GCA90_GCA + xGplus_per_minus,
##     data = attacking_midfield1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14884716  -4072969  -2382343   -98114  115002443
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept)    4789542    124586   38.44 <0.0000000000000002 ***
## GCA90_GCA      4789298    260549   18.38 <0.0000000000000002 ***
## xGplus_per_minus  407199     19648   20.73 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7637000 on 7444 degrees of freedom
## (12 observations deleted due to missingness)
## Multiple R-squared:  0.1215, Adjusted R-squared:  0.1213
## F-statistic: 514.9 on 2 and 7444 DF, p-value: < 0.00000000000000022

```

Center Midfield

```
centralMidfield_linReg <- lm(player_market_value_euro ~ SoT_per_90 + PPA + xGplus_per_minus,
                             data = central_midfield1)

summary(centralMidfield_linReg)
##
## Call:
## lm(formula = player_market_value_euro ~ SoT_per_90 + PPA + xGplus_per_minus,
##     data = central_midfield1)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-18479514	-3465209	-982597	2537907	72817846

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5147428	63662	80.86	<0.0000000000000002 ***
SoT_per_90	1742102	89876	19.38	<0.0000000000000002 ***
PPA	323347	6619	48.85	<0.0000000000000002 ***
xGplus_per_minus	441884	11228	39.35	<0.0000000000000002 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5336000 on 15480 degrees of freedom
## (14 observations deleted due to missingness)
## Multiple R-squared:  0.2468, Adjusted R-squared:  0.2466
## F-statistic: 1690 on 3 and 15480 DF, p-value: < 0.00000000000000022
```

Center Back

```

centreBack_linReg <- lm(player_market_value_euro ~ Int + Cmp + xGplus_per_minus,
                        data = centre_back1)

summary(centreBack_linReg)
##
## Call:
## lm(formula = player_market_value_euro ~ Int + Cmp + xGplus_per_minus,
##     data = centre_back1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21828623 -2273442  -567071  1413674  54501893
##
## Coefficients:
##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept)  2753928.2    75007.4   36.715 <0.0000000000000002 ***
## Int           8034.5      4062.0    1.978      0.048 *
## Cmp           6983.1      168.9   41.334 <0.0000000000000002 ***
## xGplus_per_minus 298188.4    8206.0   36.338 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4516000 on 10757 degrees of freedom
## (622 observations deleted due to missingness)
## Multiple R-squared:  0.2874, Adjusted R-squared:  0.2872
## F-statistic: 1446 on 3 and 10757 DF, p-value: < 0.00000000000000022

```

Centre Forward and Second Striker

```

centreForward_linReg <- lm(player_market_value_euro ~ xA + Att_3rd_Touches + PPA + xGplus_per_minus,
                           data = centreForward_secondStriker1)

summary(centreForward_linReg)
##
## Call:
## lm(formula = player_market_value_euro ~ xA + Att_3rd_Touches +
##      PPA + xGplus_per_minus, data = centreForward_secondStriker1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25786367 -3714527 -1437942  2552204 154587996
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5225640.6    85736.3   60.95 <0.0000000000000002
## xA            -2296878.6   132176.0  -17.38 <0.0000000000000002
## Att_3rd_Touches  11236.2     509.1   22.07 <0.0000000000000002
## PPA            586681.8    20758.1   28.26 <0.0000000000000002
## xGplus_per_minus 460234.0    13972.4   32.94 <0.0000000000000002
##
## (Intercept)      ***
## xA                ***
## Att_3rd_Touches  ***
## PPA               ***
## xGplus_per_minus ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6347000 on 12057 degrees of freedom
## (18 observations deleted due to missingness)
## Multiple R-squared:  0.2631, Adjusted R-squared:  0.2629
## F-statistic: 1076 on 4 and 12057 DF, p-value: < 0.00000000000000022

```

Right/Left Back

```
rightLeftBack_linReg <- lm(player_market_value_euro ~ Cmp + Mid_3rd,
                           data = right_left_back1)

summary(rightLeftBack_linReg)
##
## Call:
## lm(formula = player_market_value_euro ~ Cmp + Mid_3rd, data = right_left_back1)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
##	-19466590	-1233371	-382900	688050	76820947

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	2505603.46	26119.83	95.93	<0.0000000000000002 ***
## Cmp	6862.22	98.94	69.36	<0.0000000000000002 ***
## Mid_3rd	237724.36	4395.55	54.08	<0.0000000000000002 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3373000 on 31756 degrees of freedom
## Multiple R-squared:  0.3032, Adjusted R-squared:  0.3031
## F-statistic: 6909 on 2 and 31756 DF, p-value: < 0.00000000000000022
```

Right/Left Midfield

```
rightLeftMidfielder_linReg <- lm(player_market_value_euro ~ SoT_percent + Tkl + Cmp,
                                data = right_left_midfielder1)

summary(rightLeftMidfielder_linReg)
##
## Call:
## lm(formula = player_market_value_euro ~ SoT_percent + Tkl + Cmp,
##     data = right_left_midfielder1)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
##	-15693061	15076	250018	1391220	22341717

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	13375838	688325	19.432	< 0.0000000000000002 ***
## SoT_percent	68807	14416	4.773	0.00000229 ***
## Tkl	-21358	21633	-0.987	0.3239
## Cmp	-1905	1066	-1.787	0.0744 .

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3734000 on 590 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.06536,    Adjusted R-squared:  0.06061
## F-statistic: 13.75 on 3 and 590 DF,  p-value: 0.0000000111
```

Right/Left Winger

```
rightLeftWinger_linReg <- lm(player_market_value_euro ~ PPM + xGplus_per_minus,
                             data = winger1)

summary(rightLeftWinger_linReg)
##
## Call:
## lm(formula = player_market_value_euro ~ PPM + xGplus_per_minus,
##     data = winger1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -39328425  -7217224  -3247564   3537297 143176456
##
## Coefficients:
##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept)    8381321     288670  29.034 < 0.0000000000000002 ***
## PPM             1695856     213385   7.947  0.00000000000000212 ***
## xGplus_per_minus 1220810       20949  58.274 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11660000 on 9455 degrees of freedom
## (16 observations deleted due to missingness)
## Multiple R-squared:  0.3256, Adjusted R-squared:  0.3254
## F-statistic: 2282 on 2 and 9455 DF, p-value: < 0.00000000000000022
```

Appendix A: Important Variables

Appendix B: Tables:

Standard

Season_End_Year : The year at the end of the season.

Squad : The team the player plays on.

Comp : The league the player plays in.

Player : The name of the player.

Nation : Nationality of the Player.

Pos : The player's position

Age : Current Age

Born : Year of Birth

Matches_Played : Matches Played

Starts : Games Started

Min : Minutes Played

Min_per_90 : Minutes Played divided by 90

Gls : Goals

Ast : Assists

G_minus_PK : Goals minus Penalty Kicks

PK : Penalty Kicks Made

PKatt : Penalty Kicks Attempted

CrdY : Yellow cards

CrdR : Red Cards

Gls_Per : Goals Scored per 90 minutes

Ast_Per : Assists per 90 minutes

G+A_Per : Goals and Assists per 90 minutes

G_minus_PK_Per : Goals minus Penalty Kicks per 90 minutes

G+A_minus_PK_Per : Goals and Assists minus Penalty Kicks per 90 minutes

xG : Expected Goals

$np\bar{x}G$: Non-Penalty Expected Goals

$x\bar{A}$: xG Assisted

$np\bar{x}G + x\bar{A}$: Non Penalty Expected Goals plus xG Assisted

xG_{Per} : Expected Goals per 90 minutes

$x\bar{A}_{Per}$: xG Assisted per 90 minutes

$xG + x\bar{A}_{Per}$: Expected Goals plus Assist per 90 minutes

$np\bar{x}G_{Per}$: Non-Penalty Expected Goals per 90 minutes

$np\bar{x}G + x\bar{A}_{Per}$: Non-Penalty Expected Goals plus xG Assisted per 90 minutes

Goalkeeping

$Season_End_Year$: The year at the end of the season.

$Squad$: The team the player plays on.

$Comp$: The league the player plays in.

$Player$: The name of the player.

$Nation$: Nationality of the Player.

Pos : The player's position

Age : Current Age

$Born$: Year of Birth

$Matches_Played$: Matches Played

$Starts$: Games Started

Min : Minutes Played

Min_per_90 : Minutes Played divided by 90 (Number of games)

GA : Goals scored against

$GA90$: Goals scored against per 90 minutes

SoTA : Shots on Target Against

Saves : Number of saves

save_percent : (Shots on Target Against - Goals Against) / Shots on Target Against *Note that not all shots on target are stopped by the keeper, many will be stopped by defenders. Does not include penalty kicks*

W : Wins

D : Draws

L : Losses

CS : Clean sheets (When no goals are scored against)

CS_percent : Percentage of matches resulting in clean sheets

PKatt_Penalty : Penalty kicks attempted

PKA_Penalty : Penalty kicks allowed in

PKsv_Penalty : Penalty kicks saved

PKm_Penalty : Penalty kicks missed by opposing team

Save_percent_Penalty : Penalty Kick Goals Against/Penalty Kick Attempts *Penalty shots that miss the target are not included*

Advanced Goalkeeping

Season_End_Year : The year at the end of the season.

Squad : The team the player plays on.

Comp : The league the player plays in.

Player : The name of the player.

Nation : Nationality of the Player.

Pos : The player's position

Age : Current Age

Born : Year of Birth

Min_per_90 : Minutes Played divided by 90 (Number of games)

GA : Goals scored against

PKA : Penalty Kicks Allowed

FK : Free Kick Goals Against

CK : Corner Kick Goals Against

OG : Own Goals Scored Against Keeper

PSxG : Post-shot expected goal; based on how likely the goalkeeper is to save the shot. (In-game penalty kicks included)

PSxG_per_sOT : Post-shot expected goals per shot on target. (Penalty kicks not included)

PSxG_plus_minus : Post-Shot Expected Goals minus Goals Allowed (Positive numbers suggest better luck or an above average ability to stop shots)

per_90 : Post-Shot Expected Goals minus Goals Allowed per 90 minutes

Cmp_Launched : Passes longer than 40 yards completed

Att_Launched : Passes longer than 40 yards attempted

Cmp_percent_Launched : 40+ yard pass completion rate

Att_Passes : Attempted passes excluding goal kicks

Thr_Passes : Throws attempted

Launch_percent_Passes : Percentage of Passes that were Launched and not Goal Kicks

AvgLen_Passes : Average Length of Passes in Yards

Att_Goal : Goal Kicks Attempted

Launch_percent_Goal : Percentage of Goal Kicks sent more than 40 yards

AvgLen_Goal : Average length of goal kicks

Opp_Crosses : Opponents attempted crosses into penalty area

Stp_percent_Crosses : Percent of crosses into penalty area which were successfully stopped by the goalkeeper

#OPA_Sweeper : Number of defensive actions outside of penalty area

#OPA_per_90_Sweeper : Defensive actions outside of penalty area per 90 minutes

AvgDist_Sweeper : Average distance from goal (in yards) of all defensive actions

Shooting

Season_End_Year : The year at the end of the season.

Squad : The team the player plays on.

Player : The name of the player.

Nation : Nationality of the Player.

Pos : The player's position

Age : Current Age

Born : Year of Birth

Mins_per_90 : Minutes played divided by 90

Gls : Goals scored or allowed

SH : Shots Total

SoT : Shots on target

SoT_percent : Shots on target percentage, percentage of shots that are on target

Sh_per_90 : Shots total per 90 minutes

Sot_per_90 : Shots on target per 90 minutes

G_per_Sh : Goals per shot

G_per_SoT : Goals per shot on target

Dist : Average distance, in yards, from goal of all shots taken

FK : Shots from free kicks

PK : Penalty Kicks Made

PKatt : Penalty Kicks Attempted

xG : Expected Goals

npG : Non-Penalty Expected Goals

npG_per_sh : Non-Penalty Expected Goals per shot

G_minus_xG : Goals minus Expected Goals

npG_minus_npG : Non-Penalty Goals minus Non-Penalty Expected Goals

Passing

Season_End_Year : The year at the end of the season

Squad : The team the player plays on

Comp : The league the player plays in

Player : The name of the player

Nation : Nationality of the Player

Pos : The player's position

Age : Current Age

Born : Year of Birth

Mins_per_90 : Minutes played divided by 90

Cmp : Passes Completed

Cmp_percent : Pass completion percentage

TotDist : Total distance of completed passes in yards

PrgDist : Total distance, in yards, that completed passes have traveled towards the opponent's goal

Cmp_short : Completed passes within 5 and 15 yards

Att_short : Attempted passes within 5 to 15 yards

Cmp_percent_short : Completion percentage of passes within 5 to 15 yards

Cmp_Medium : Completed passes within 15 to 30 yards

Att_Medium : Attempted passes within 15 to 30 yards

Cmp_percent_Medium : Completion percentage of passes within 15 to 30 yards

Cmp_Long : Completed passes over 30 yards

Att_Long : Attempted passes over 30 yards

Cmp_percent_Long : Completion percentage of passes over 30 yards

Ast : Assists

xA : Expected Assists

A_minus_xA : Assists minus xA

KP : Passes directly leading to a shot

Final_Third : Completed passes that enter the 1/3 of the pitch closest to the goal

PPA : Completed passes into the 18-yard box

CrsPA : Completed crosses into the 18-yard box

Prog : Completed passes moving the ball towards the opponent's goal at least 10 yards from its furthest point in the last six passes, or any completed pass into the penalty area

Pass Types

Season_End_Year : The year at the end of the season.

Squad : The team the player plays on.

Comp : The league the player plays in.

Player : The name of the player.

Nation : Nationality of the Player.

Pos : The player's position

Age : Current Age

Born : Year of Birth

Mins_Per_90 : Minutes played divided by 90

Att : Passes Attempted

Live_Pass : Live-ball passes

Dead_Pass : Dead-ball passes such as a free kick or corner kick

FK_Pass : Passes attempted from freekicks

TB_Pass : Completed pass sent between back defenders into open space

Press_Pass : Passes made while under pressure from opponent

Sw_Pass : Passes that travel more than 40 yards of the width of the pitch

Crs_Pass : Crosses

CK_Pass : Corner Kicks

In_Corner : Inswinging Corner Kicks

Out_Corner : Outswinging Corner Kicks

Str_Corner : Straight Corner Kicks

Ground_Height : Ground passes

Low_Height : Passes below shoulder-level

High_Height : Passes above shoulder-level

Left_Body : Passes attempted using left foot

Right_Body : Passes attempted using right foot

Head_Body : Passes attempted using head

TI_Body : Throw-ins Taken

Other_Body : Passes attempted using body parts other than the player's head or feet

Cmp_Outcomes : Passes completed

Off_Outcomes : Offsides

Out_Outcomes : Out of bounds

In_Outcomes : Intercepted

Blocks_Outcomes : Blocked by an opponent

Goal and Shot Creation

Season_End_Year : The year at the end of the season.

Squad : The team the player plays on.

Comp : The league the player plays in.

Player : The name of the player.

Nation : Nationality of the Player.

Pos : The player's position

Age : Current Age

Born : Year of Birth

Mins_Per_90 : Minutes played divided by 90

SCA_SCA : Shot-creating Actions

SCA90__SCA : Shot-creating Actions per 90 minutes

PassLive_SCA : Completed live-ball passes that lead to a shot attempt

PassDead_SCA : Completed dead-ball passes that lead to a shot attempt (Includes free kicks, corner kicks, kick offs, throw-ins and goal kicks)

Drib_SCA : Successful dribbles that lead to a shot attempt

Sh_SCA : Shots that lead to another shot attempt

Fld_SCA : Fouls drawn that lead to a shot attempt

Def_SCA : Defensive actions that lead to a shot attempt

GCA_GCA : The offensive actions directly leading to a goal, such as passes, dribbles and drawing fouls

GCA90_GCA : Goal-Creating Actions per 90 minutes

PassLive_GCA : Completed live-ball passes that lead to a goal

PassDead_GCA : Completed dead-ball passes that lead to a goal (Includes free kicks, corner kicks, kick offs, throw-ins and goal kicks)

Drib_GCA : Successful dribbles that lead to a goal

Sh_GCA : Shots that lead to another goal-scoring shot

Fld_GCA : Fouls drawn that lead to a goal

Def_GCA : Defensive actions that lead to a goal

Defensive Actions

Season_End_Year : The year at the end of the season.

Squad : Team

Comp : Completions

Player : Players name

Nation : Players Nationality

Pos : Players position

Age : Current age

Born : Year of Birth

Mins_Per_90 : Minutes played divided by 90

Tkl : Number of players tackled

TklW : Tackles in which the tackler's team won possession of the ball

Def_3rd : Tackles in defensive 1/3

Mid_3rd : Tackles in middle 1/3

Att_3rd : Tackles in attacking 1/3

Tkl_vs_dribble : Number of dribblers tackled

Att_vs_dribble : Number of times dribbled past plus number of tackles

Press_Pressures : Number of times applying pressure to opposing player who is receiving, carrying or releasing the ball

Succ_Pressures : Number of times the squad gained possession withing five seconds of applying pressure

Successful_Pressure_Percent : Percentage of time the squad gained possession withing five seconds of applying pressure

Def_3rd_Pressures : Number of times applying pressure to opposing player who is receiving, carrying or releasing the ball, in the defensive 1/3

Mid_3rd_Pressures : Number of times applying pressure to opposing player who is receiving, carrying or releasing the ball, in the middle 1/3

Att_3rd_Pressures : Number of times applying pressure to opposing player who is receiving, carrying or releasing the ball, in the attacking 1/3

Blocks : Number of times blocking the ball by standing in its path

Sh_Blocks : Number of times blocking a shot by standing in its path

ShSv_Blocks : Number of times blocking a shot that was on target, by standing in its path

Pass_Blocks : Number of times blocking a pass by standing in its path

Int : Interceptions

Tkl+Int : Number of players tackled plus number of interceptions

Clr : Clearances

Err : Mistakes leading to an opponent's shot

Possessions

Season_End_Year : The year at the end of the season.

Squad : Team

Comp : Completions

Player : Players name

Nation : Players Nationality

Pos : Players position

Age : Current age

Born : Year of Birth

Touches : Number of times a player touched the ball

Def_Pen : Touches in defensive penalty area

Def_3rd_Touches : Touches in defensive 1/3

Mid_3rd_Touches : Touches in middle 1/3

Att_3rd_Touches : Touches in attacking 1/3

Att_Pen_Touches : Touches in attacking penalty area

Live_Touches : Live-ball touches

Succ_Dribbles : Dribbles Completed Successfully

Att_Dribbles : Dribbles Attempted

Succ_percent_Dribbles : Percentage of Dribbles Completed Successfully

#Pl_Dribbles : Number of Players Dribbled Past

Megs_Dribbles : Number of times a player dribbled the ball through an opposing player's legs

Carries : Number of times the player controlled the ball with their feet

TotDist_Carries : Total distance, in yards, a player moved the ball while controlling it with their feet, in any direction

PrgDist_Carries : Progressive Distance; Total distance, in yards, a player moved the ball while controlling it with their feet towards the opponent's goal

Prog_Carries : Carries that move the ball towards the opponent's goal at least 5 yards, or any carry into the penalty area

Final_Third_Carries : Carries that enter the 1/3 of the pitch closest to the goal

CPA_Carries : Carries into the 18-yard box

Mis_Carries : Number of times a player failed when attempting to gain control of a ball

Dis_Carries : Number of times a player loses control of the ball after being tackled by an opposing player

Targ_Receiving : Number of times a player was the target of an attempted pass

Rec_Receiving : Number of times a player successfully received a pass

Rec_percent_Receiving : Passes Received Percentage

Prog_Receiving : Progressive Passes Received

Playing Time

Season_End_Year : The year at the end of the season.

Squad : The team the player plays on.

Comp : The league the player plays in.

Player : The name of the player.

Nation : Nationality of the Player.

Pos : The player's position

Age : Current Age

Born : Year of Birth

MP : Matches Played

Min : Minutes

Min_per_MP : Minutes per Match Playerd

Min_percent_Playing.Time : Percentage of team's total minutes in which player was on the pitch

Mins_per_90 : Minutes played divided by 90

Starts : Game or games started by player

Mn_per_Start : Minutes Per Match Started

Comp1 : Complete matches played

Subs : Games as sub

Mn_per_Sub : Minutes Per Substitution

unSub : Games as an unused substitute

PPM : Points per Match

onG : Goals scored by team while on pitch

onGA : Goals allowed by team while on pitch

plus_per_minus : Goals scored minus goals allowed by the team while the player was on the pitch

plus_per_minus_per_90 : Goals scored minus goals allowed by the team while the player was on the pitch per 90 minutes played

On_minus_Off :

onxG : Expected goals by team while on pitch

onxGA : Expected goals allowed by team while on pitch

xGplus_per_minus : Expected goals scored minus expected goals allowed by the team while the player was on the pitch

xGplus_per_minus_per_90 : Expected goals scored minus expected goals allowed by the team while the player was on the pitch per 90 minutes played

Misc

Season_End_Year : The year at the end of the season.

Squad : The team the player plays on.

Comp : The league the player plays in.

Player : The name of the player.

Nation : Nationality of the Player.

Pos : The player's position

Age : Current Age

Mins_Per_90 : Minutes played divided by 90

CrdY : Yellow cards

2CrdY : Second Yellow Card

Fls : Fouls committed

Fld : Fouls drawn

Off : Off-sides

Crs : Crosses

Int : Interceptions

Tk1W : Tackles in which the tackler's team won possession of the ball

PKwon : Penalty Kicks Won

PKcon : Penalty Kicks Conceded

OG : Own Goals

Recov : Number of loose balls recovered

Won_Aerial : Aerials won

Lost_Aerial : Aerials lost

Won_percent_Aerial : Minimum .97 aerial duels per squad game to qualify as a leader

Market Values Table

squad : The team the player plays on.

player_name : The player's name

player_url : The player's URL on transfermarkt

player_position : The player's position

player_age : The player's age

player_name : The player's name

player_nationality : The player's nationality

current_club : The club the player currently plays on

player_height_mtrs : The player's height

player_foot : The player's dominant foot

joined_from : The former club the player joined from

contract_expiry : The time when the player's contract expires

player_market_value_euro : The valuation of the player in Euros

date_joined : The date the player joined