Soccer: Best Goal Scorer Prediction

# 1. Introduction

This report is a detailed assessment on the variables that are needed to predict the best goal scorer/player in football/soccer. In order to predict the best goal scorer in football, we will look at each players’ goals per match and compare it with all the other players across all the teams in the league. The data set i used is from Kaggle and is a league data set from the English Premier League season of 2015-2016. I tried to get the latest data, but failed to do so. Nonetheless, it is still a very good data set with numerical variables and also includes all the players from that season with numerous records about their individual qualities and attributes.

#reading the dataset  
library(regclass)

## Loading required package: bestglm

## Loading required package: leaps

## Loading required package: VGAM

## Loading required package: stats4

## Loading required package: splines

## Loading required package: rpart

## Loading required package: randomForest

## randomForest 4.7-1

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

## Important regclass change from 1.3:  
## All functions that had a . in the name now have an \_  
## all.correlations -> all\_correlations, cor.demo -> cor\_demo, etc.

library(ggplot2)

##   
## Attaching package: 'ggplot2'

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

library(corrplot)

## corrplot 0.92 loaded

data <- read.csv("pl\_15-16.csv")

# 2.Data Cleaning

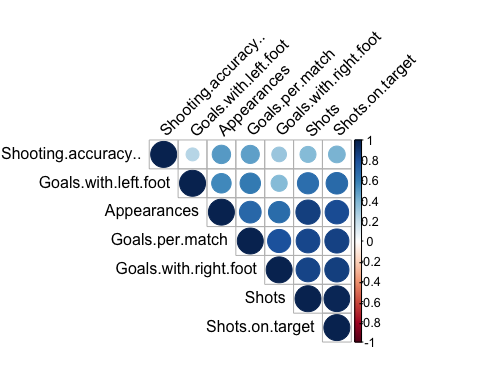
Now that we’ve read the data, we will clean the data set by filtering the variables we don’t need to predict the goals, per match. In football, the best goal scorers are usually the players that play in a forward position. Hence, we eliminate or drop the other positions(including: Goalkeeper, Defender, Midfielder). In this case we create a subset with only the players in the forward position. Next, we drop the variables or features that do not have an affect on the goals per match variable. I have been playing football ever since i was a kid and thankfully i can add my knowledge to this report by removing the variables that don’t have an affect on goals per match.

#Cleaning the dataset for Forwards  
df <- subset(data, Position!="Defender" & Position!="Goalkeeper" & Position!="Midfielder")  
  
drop <- c("Clean.sheets", "Goals.conceded", "Tackles", "Tackle.success", "Last.man.tackles", "Blocked.shots", "Interceptions", "Clearances", "Headed.Clearance", "Clearances.off line", "Successful.50.50s", "Errors.leading.to.goal", "Tackle.success.%", "Clearances.off.line", "Recoveries", "Own.goals", "Duels.won", "Duels.lost", "Aerial.battles.lost", "Passes", "Crosses", "Accurate.long.balls", "Through.balls", "Yellow.cards", "Red.cards", "Fouls", "Freekicks.scored", "Position", "Passes.per.match", "Headed.goals", "Penalties.scored", "Big.chances.created", "Assists", "Aerial.battles.won", "Big.chances.missed", "Offsides", "Goals", "Name")  
df = df[,!(names(df) %in% drop)]  
df[1:5,]

## Appearances Goals.with.right.foot Goals.with.left.foot Goals.per.match Shots  
## 5 2 0 0 0.00 2  
## 7 12 0 0 0.08 12  
## 10 15 1 0 0.27 14  
## 11 15 0 1 0.07 10  
## 12 30 16 3 0.80 119  
## Shots.on.target Shooting.accuracy..  
## 5 0 0.00  
## 7 4 0.33  
## 10 5 0.36  
## 11 6 0.60  
## 12 52 0.44

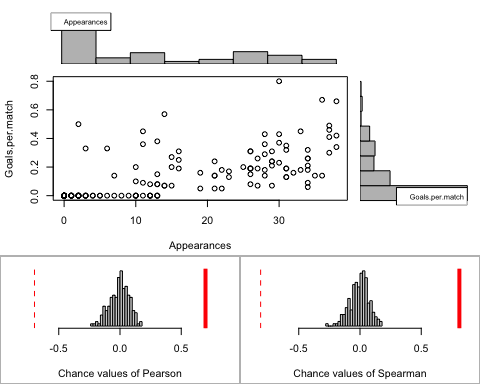
Now, to confirm that i only left the variables that helps in predicting the goals per match, I deduced a correlation matrix, which showls they are correlated with each other.

#correlation matrix  
cm <- cor(df)  
corrplot(cm, type = "upper", order = "hclust",   
 tl.col = "black", tl.srt = 45)

 # 3. Association Analysis For the association analysis, since we’re predicting the goals per match variable, it will be our y variable. therefore, the other remaining variables will be our x variables.

#Association analysis where Goals per match is my y variable and Appearances is my x variable  
associate(Goals.per.match~Appearances,data= df, seed=2022, permutations = 200)

## Association between Appearances (numerical) and Goals.per.match (numerical)  
## using 168 complete cases

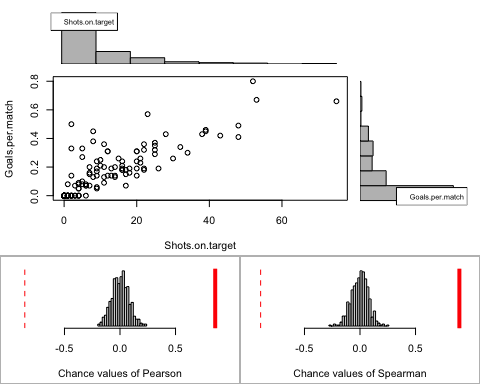


## Permutation procedure:  
## Value Estimated p-value  
## Pearson's r 0.6981147 0  
## Spearman's rank correlation 0.8113188 0  
## With 200 permutations, we are 95% confident that:  
## the p-value of Pearson's correlation (r) is between 0 and 0.018   
## the p-value of Spearman's rank correlation is between 0 and 0.018   
## Note: If 0.05 is in this range, increase the permutations= argument.  
##   
##   
##   
## Advice: If stream of points is well described by an ellipse, use Pearson's r.  
## Otherwise, as long as stream is monotonic, use Spearman's rank correlation  
## or try logs, e.g. associate( log10(y)~log10(x) )

We can see the results and say that the association between Goals per match and Appearances is statistically significant because the p-value is less than 0.05. Therefore appearance (or more games played) is a major factor in predicting the goals per match. the graph also shows that when a player has a higher number of appearances it tends to be that the player has the chance of getting a higher goals per match.

#Association analysis where Goals per match is my y variable and Shots.on.target is my x variable  
associate(Goals.per.match~Shots.on.target,data= df, seed=2022)

## Association between Shots.on.target (numerical) and Goals.per.match (numerical)  
## using 168 complete cases



## Permutation procedure:  
## Value Estimated p-value  
## Pearson's r 0.8573528 0  
## Spearman's rank correlation 0.8948672 0  
## With 500 permutations, we are 95% confident that:  
## the p-value of Pearson's correlation (r) is between 0 and 0.007   
## the p-value of Spearman's rank correlation is between 0 and 0.007   
## Note: If 0.05 is in this range, increase the permutations= argument.  
##   
##   
##   
## Advice: If stream of points is well described by an ellipse, use Pearson's r.  
## Otherwise, as long as stream is monotonic, use Spearman's rank correlation  
## or try logs, e.g. associate( log10(y)~log10(x) )

From the results we can say this association between Goals per match and shots on target is statistically significant because the p-value is less than 0.05. Also from the scatter plot we can see a rise in the goals per match when there is a rise in the shots on target. Therefore when there is an increase in shots on target, the goals per match is likely to increase.

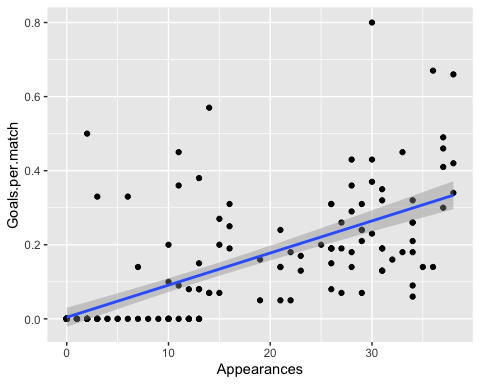
# Simple Linear Regression

#Simple Linear Regression  
model <- lm(Goals.per.match ~ Appearances, data = df)  
model

##   
## Call:  
## lm(formula = Goals.per.match ~ Appearances, data = df)  
##   
## Coefficients:  
## (Intercept) Appearances   
## 0.004264 0.008671

ggplot(df, aes(Appearances, Goals.per.match)) +  
 geom\_point() +  
 stat\_smooth(method = lm)

## `geom\_smooth()` using formula 'y ~ x'



summary(model)

##   
## Call:  
## lm(formula = Goals.per.match ~ Appearances, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.23908 -0.04781 -0.00664 -0.00426 0.53560   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0042643 0.0130405 0.327 0.744   
## Appearances 0.0086712 0.0006902 12.563 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1173 on 166 degrees of freedom  
## Multiple R-squared: 0.4874, Adjusted R-squared: 0.4843   
## F-statistic: 157.8 on 1 and 166 DF, p-value: < 2.2e-16

# We can say that the equation is as follows:

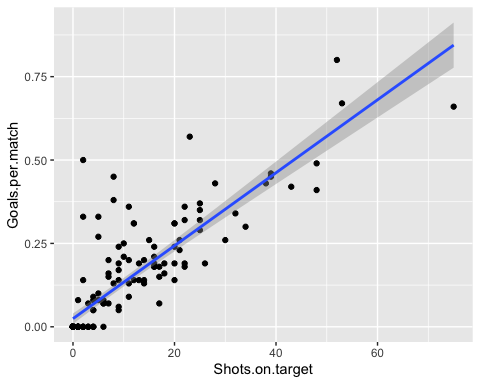
# Goals per match = 0.004264 + 0.008671\*Appearances

model <- lm(Goals.per.match ~ Shots.on.target, data = df)  
model

##   
## Call:  
## lm(formula = Goals.per.match ~ Shots.on.target, data = df)  
##   
## Coefficients:  
## (Intercept) Shots.on.target   
## 0.02436 0.01094

ggplot(df, aes(Shots.on.target, Goals.per.match)) +  
 geom\_point() +  
 stat\_smooth(method = lm)

## `geom\_smooth()` using formula 'y ~ x'



summary(model)

##   
## Call:  
## lm(formula = Goals.per.match ~ Shots.on.target, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.18459 -0.03099 -0.02436 0.00951 0.45377   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.0243613 0.0079451 3.066 0.00253 \*\*   
## Shots.on.target 0.0109363 0.0005096 21.460 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.08434 on 166 degrees of freedom  
## Multiple R-squared: 0.7351, Adjusted R-squared: 0.7335   
## F-statistic: 460.5 on 1 and 166 DF, p-value: < 2.2e-16

# We can say that the equation is as follows:

# Goals per match = 0.004264 + 0.008671\*Shots on target