As I identified before the first part of this they key parts of the model are predicting how many goals might be in the match and the type of player and therefore do they have a scoring impact on a match or not. So first things first I need to create a score forecasting model.

**Data**

In order to do this I need to have a methodology for identifying how strong each team is in the match and then correlate that difference in team strength to an estimation of the scores for both teams in the match. Fundamentally I’m going to work out how the average team performs in the selected metric and then compare each teams performances to average and the difference between to two will be the variable to compare to the score.

**Metric options**

* Goals scored – Teams goals scored is the truest sense of how good a team is. Can be effected by short term variability in performances. This is freely available data
* xG – xG or expected goals is the expected scoring rate when you take into account various details about the shot in question
* shots taken- better teams will probably take more shots

I have decided to use expected goals data for each team. As a prediction metric it is highly correlated to actual performance and there is a source for the data thanks to fivethirtyeight.com. They publish team level expected goals data for many leagues freely available on their GitHub. I can read the data into R using the below code:

matchdat <- read\_csv("https://projects.fivethirtyeight.com/soccer-api/club/spi\_matches.csv")

Therefore I can use that to rate a team over the season and compare that to the average team. The data is easily read into R using the read\_csv function.

## calaculating what the average performances

homeav <- matchdat %>% filter(league\_id == 2411) %>% #working out the average home xg

summarise(avfor = mean(xg1, na.rm = T), avag = mean(xg2, na.rm = T))

homfor <- homeav[[1]] # extracting the number for home and away

homeag <- homeav[[2]]

# comparing each teams performance to average for xg for and against

homeratdat2 <- matchdat %>% filter(league\_id == 2411) %>%

group\_by(team1) %>%

summarise(avfor = mean(xg1, na.rm = T), avag = mean(xg2, na.rm = T)) %>%

mutate(xgfh = avfor - homfor, xgah = avag - homeag) %>%

select(team1, xgfh, xgah)

The code above calculates what the average team performs like for expected goals for and against. This is then compared to the teams actual performance to give each team in each season a delta. The code above is the calculation for home teams but the same code is used for way teams.

## creating a data frame with matches the for and against scores and the xg deltas

matches <- matchdat %>% filter(league\_id == 2411) %>%

left\_join(homeratdat2, by = "team1") %>%

left\_join(awayratdat2, by = "team2") %>%

mutate(deltafh = xgfh + xgaa, deltagh = xgah+xgfa) %>%

select(season, score1, score2, deltafh, deltagh)

## splitting it up for home and away

mat1 <- matches %>% select(season, score1, deltafh) %>%

mutate(loc = "home")

colnames(mat1)[2] <- "score"

colnames(mat1)[3] <- "delta"

mat2 <- matches %>% select(season, score2, deltagh) %>%

mutate(loc = "away")

colnames(mat2)[2] <- "score"

colnames(mat2)[3] <- "delta"

## putting it together and creating the score as a factor

matall <- mat1 %>% bind\_rows(mat2) %>%

mutate(scorecat = as.factor(if\_else(score > 5,"5", as.character(score)))) %>%

select(-score, -season) ### data for calculating the chance of goals scored

Next, I joined each teams xG deltas to the actual scores for historical seasons. Therefore I have data which links each teams xG delta to the actual goals they scored and conceded in an historical match. I also split the data out so I could add the category if the team was home or away. Finally, as matches with more then 5 goals are rare I changed each time a team scored of conceded more then 5 to 5. Matches with more goals then that are so rare that the model will not be able to make good predictions on that.

```{r}

##splitting the data into training and testing

score\_split <- initial\_split(matall, prop = 0.9, strata = scorecat)

score\_train <- training(score\_split)

score\_test <- testing(score\_split)

##creating the classification random forest

rand1 <- rand\_forest() %>% #type of model

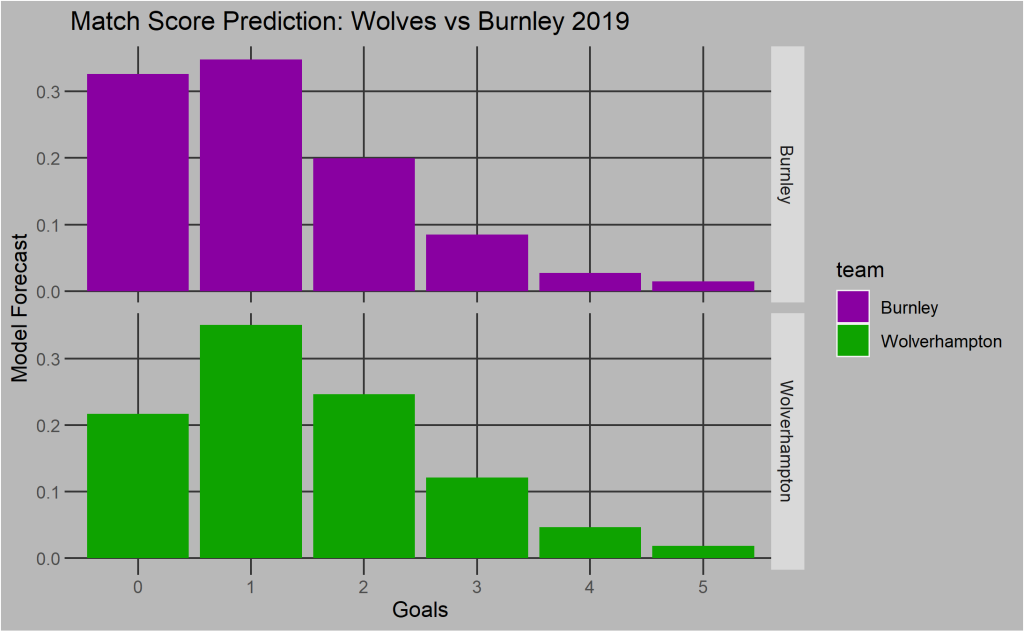
set\_engine("ranger") %>% # the engine used to fit the model. For randomm forest rf is a another one

set\_mode("classification") %>% # the mode for the model as random forest can do both classification and regression

fit(scorecat ~., data = score\_train) # the function to fit the model

### rand1 is the random forest for the score of the match

above I then used the functions from the tidy models packages to simply fit a random forest for the score prediction for a match. Based on if the team is home or away and the different in the expected goals ratings.



Above you can see the output from the initial model for the the Wolves vs Burnley game in 2019. For both teams there’s an estimate of how likely they are to score that amount of goals. Wolves are likely to score more then Burnley but there isn’t too much difference in the teams. Now for a player playing in this match I can have 10000 runs of this match and just over 30% of them Wolves will score 1 goal. This then becomes one of the arguments in the models to predict goals scored by a player and assists by the player. Defensive points can be directly calculated from the Oppositions estimated goals scored. Now I the basis an a rough prediction of match outcomes I can move onto the precise player predictions to make this relevant to FPL