# Predicting The Best Team: Barclays Premier League Fantasy Football

## Introduction

I've been really into BPL Fantasy Football this season and have been building a fantasy team since the start of the season last year.

I'm not very good at picking the right squad based on my instincts, so my basic goal is to use the data from the present and past weeks to determine the best picks for the upcoming week.

## How Fantasy Football Works

I play Fantasy Football off the official BPL Fantasy site. For the uninitiated, Fantasy Football is an online game where users pick a squad of 15 real-life football players from a league. In this case, I'm playing the BPL, which is the English top division. Points are scored and collated depending on a player's actions in the actual game. The aim of the game is to amass the highest number of points each game week. The rules are here.

In general, players will register points for scoring goals, notching assists, clean sheets, penalties. They will lose points for getting yellow cards, red cards, missing penalties and conceding goals.

Friends tend to form mini-competitions to see who's better at picking the best squad week after week. At the end of the season, the winner of their league with the higest points gets the honour of having the best fantasy football instincts (although there's usually money involved in mini-leagues amongst friends). There's a budget allocated to each user so you can't always choose the best player because he might be too expensive. Players get transferred in and out of teams weekly, so there's an entire transfer market as well.

## My Objective

The aim of my analysis is to predict a team of 15 players who will score the highest fantasy points within a budget of £100m in the upcoming gameweek.

## Fantasy Football Data

Data is available from an API from the official site on players' basic information, team information, fixture information and performance stats. Each player's performance is aggregated every gameweek to give an overall view of the players' performance. There are nested data sets on each player's weekly performance history, as well as performance in past seasons.

The dataset contains a database of 550 players that are distributed amongst 20 teams in the BPL. There are a total of 63 data attributes for any single player. Each player has a unique player id, and data is updated once every *gameweek*. That is to say, a full round of fixtures played amongst 20 teams. Fixtures sometimes get postponed from one gameweek to another, so I'm only focusing on the first 26 gameweeks, with the full 10 games being played.

The dataset also contains the "fixture history" of a player, ie. the performance of each and every player in each and every game he has played so far. This is a *t* by 20 dataframe, where *t* is the gameweek number. I will be tapping on the fixture history of each individual player to construct features about their form. The description of the full attributes and feature set can be found in the Annex.

## Preliminary Model Specification & Feature Selection

I conducted some preliminary modelling with a sample data set from Gameweek 26. The main purpose of my analysis was to uncover a base model specification and select a set of features for my model.

I came to the following main conclusions in my preliminary analysis.

1. **Form matters more than cumulative performance**

    - Based on regression that I ran on Gameweek 26 data, I inferred that features on cumulative performance are not as significant as "form" variables. These measure the “streakiness” of a players' performance.

    - However, defining a form variable is tricky and highly arbitrary. I'm going to have to restrict myself to certain specification of "form" for a start.

- To define form, I will take the sum total of the various performance attributes up to a certain lag length, *r*. I will then have to run the model through different lag lengths to determine which would be the best lag specification for the model.

2. **Individual performance matters more than team performance**

- I found that team performance variables, like team form, or whether a team is playing home or away is not as significant a feature as individual performance variable.

3. **The distribution of fantasy points is highly skewed**

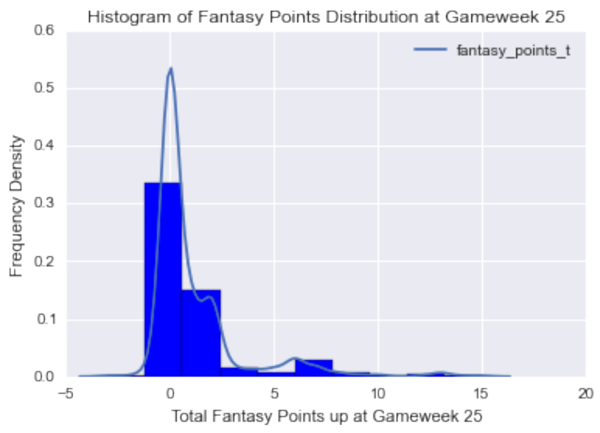
- Figure 1 shows a distribution of the number of fantasy points per scored by each player in a typical gameweek. Approximately half of the players score below 2 points and about 30% of them score zero points.

- There are 550 players in the database but only a maximum of 11 per team, so 220 players can be starting in a round of games. Each team gets a maximum of 3 substitutes per game so there can only be a maximum of 260 point scorers out of 550 players in the database.

- Players who start also typically score a minimum of 2 points, which is scored having played about 60 minutes of the game. Substitutions before 60 minutes are quite rare and usually due to injury.

- I would therefore have to try to stay away from models that rely on normality assumptions.

**Figure 1**



I will specify my model as a regression problem. I will try to predict the number of fantasy points each player will score in the *t*-th gameweek, based on the features collected in the *t*-1-th gameweek. I will only consider features of lag 1 for a start. This means that I am assuming that a player's performance in gameweek *t-1* is relevant to his performance in gameweek *t*.

A simple linear model is thus specified as such:

$$Y_{t,i} = \beta^T . \boldsymbol{X}_{t-1,i} + \varepsilon_i$$

where $Y_{t,i}$ is the number of fantasy points scored by the *i*-th player in the *t*-th gameweek and $\boldsymbol{X}_{t-1,i}$ is the feature set associated with the *i*-th player recorded in the *t-1*-th gameweek. I will try to specify a model that will return the best predictions.

Bearing in mind that the overall objective is to choose 15 players that will make up my fantasy team, I will approach the problem as follows:

1. Subset the dataset into different types of players, ie. Goalkeepers, Defenders, Midfielders and Forwards.
2. Select a set of features based on each player type. This is because GKs score points for saves and clean sheets, while FWs do not, etc.
3. Model the data and narrow down to the best candidate model based on several model selection criteria.
4. Adjust model parameters like lag length until best candidate model is reached.
5. Predict the 5 best scoring Goalkeepers, as well as 10 Defenders, Midfielders and Forwards based on the model specified.
6. Pick 15 players (2 GKs, 5 DFs, 5 MFs and 3 FWs) from 35 subject to the budget cap of £100m.

## Model Comparison and Selection

Given the data, I considered three different kinds of models that would be suited to a regression type problem.

- Simple Linear Regression
- Multinomial Logistic Regression
- Random Forest Regression

I decided against a simple linear regression because the normality and linearity assumptions would not hold given the skewed distribution of the response variable.

This would result in a high out-of-sample generalisation error. Further, I would also have to scale my feature set because of the points and form variables are all on different scales. This would result in loss of interpretability.

In the case of multinomial logistic regression, I would have to define my response variable as an ordinal variable in terms of high, medium or low fantasy points scores. This has some intuitive appeal. I would be able to predict a set of players that could be classified as “high scoring”. This also deals with the skewed nature and non-normality of my response variable. However, defining ordinal variables might not help me make decisions in the case of tied scores. How I decide to rank and order variables eg. “High scoring” is defined as more than 5 points, I also have to scale my variables, as in linear regression, which results in loss of interpretability.

The data will instead be modelled using a random forest regression. Random forest regression stand as a non-parametric counterpart to linear regression where multiple decision trees are constructed based on re-sampled or bootstrapped data from the original training set. There are three main parameters in random forests that could affect model performance, namely (a) the depth of each tree, (b) the number of features in each tree, and (c) the number of trees in the forest. To narrow down my parameters on model selection, I have set the number of trees in the forest as *ntrees=* 20, and left number of features as unconstrained in each forest. I have chosen to leave the number of features as unconstrained as I have been fairly parsimonious with my feature selection to begin with.

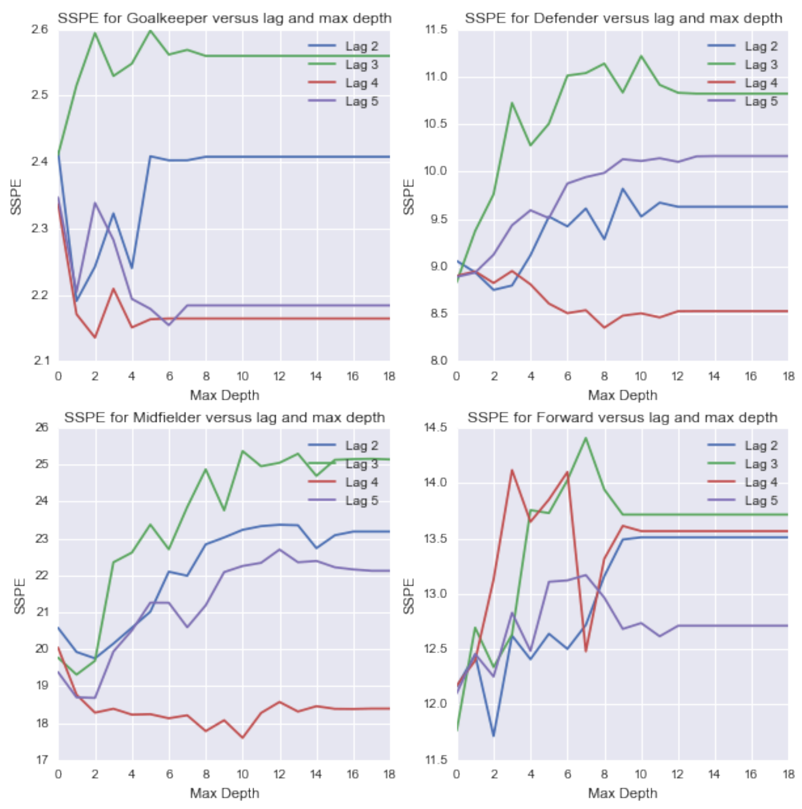
I will focus on tuning the maximum depth of each tree as well as the *lag* length of my form features as earlier described. (Recall that I had earlier arbitrarily defined lag length, and that my form features are a sum of the number of points up to the given lag length, say *r*. The variable “fantasy_form” for instance, is the sum of all fantasy points scored for the past *r* gameweeks.

Since I had multiple cross-sections of data over regular time intervals, I decided to train the random forest model on the full data set on the *t-1-th* gameweek. I would then run the model through the test set, which would be the full data set of the *t-th* gameweek to predict the scores on the *t+1-th* gameweek. To evaluate model performance across different choices of *r* and maximum depth, I would then calculate the sum of squared prediction errors(SSPE), defined by:

$$SSPE = \frac{\sum_n^1 \left(\hat{Y}_{t+1,i} | \boldsymbol{X}_{t,i} - Y_{t+1,i}\right)^2}{n-1},$$

Where *n* is the number of players in the dataset that are subset according to player type and $\boldsymbol{X}_t$ represents the feature set corresponding to the players in the subset. I plotted SSPE for each player type across different levels of maximum depth of each tree as well as lag ranges from 2 to 5. I would be looking out for an elbow-like plot to determine if the minimum required depth that would achieve the lowest SSPE without over-fitting. See Figure 2 for the plot for SSPE for a model that is trained on Gameweek 6 data, tested on Gameweek 7 data and used to predict Gameweek 8 outcomes.

**Figure 2: Plots of Sum of Squared Prediction Errors versus Max Depth and Lag**



It's pretty apparent from the plots (except for the FW position players) that 4-lagged variables yield the lowest SSPE as the curve lies almost entirely below the other curves according to lag. I'm going to go with the 4-lagged variables for my model. I also selected max depth for the Goalkeepers, Defenders, Midfielders and Forwards as 4, 8, 10 and 7 respectively. Once I had the optimal lag length, I reconstructed my features accordingly, fitted the model with max features as unconstrained, number of trees =20, and the maximum depth of each tree in the forest as outlined above. This gives me the predicted scores of each player in any given gameweek.

## Optimisation Step

Next, we have to formulate the linear optimisation problem to obtain a set of 15 players that can be considered for next week's fantasy team. Our objective function is therefore the predicted scores for 15 players. I also have a set of constraints set by the game. I can only pick 2 GKs, 5 DFs, 5 MFs, and 3 FWs. I can also only pick 3 players from any single team. Lastly, I cannot exceed a budget of £100m to buy these players. The problem is thus formulated as such:

$$\max_{x} \sum_{i,j,k} \widehat{p_{ijk}} . x_{ijk}$$

$$\sum_{j} x_{1jk} = 2, \sum_{j} x_{2jk} = 5, \sum_{j} x_{3jk} = 5, \sum_{j} x_{4jk} = 3\,;$$

$$\sum_{j} x_{ijk} \leq 3 \text{ for } j = 1, \dots, \text{N\_teams}$$

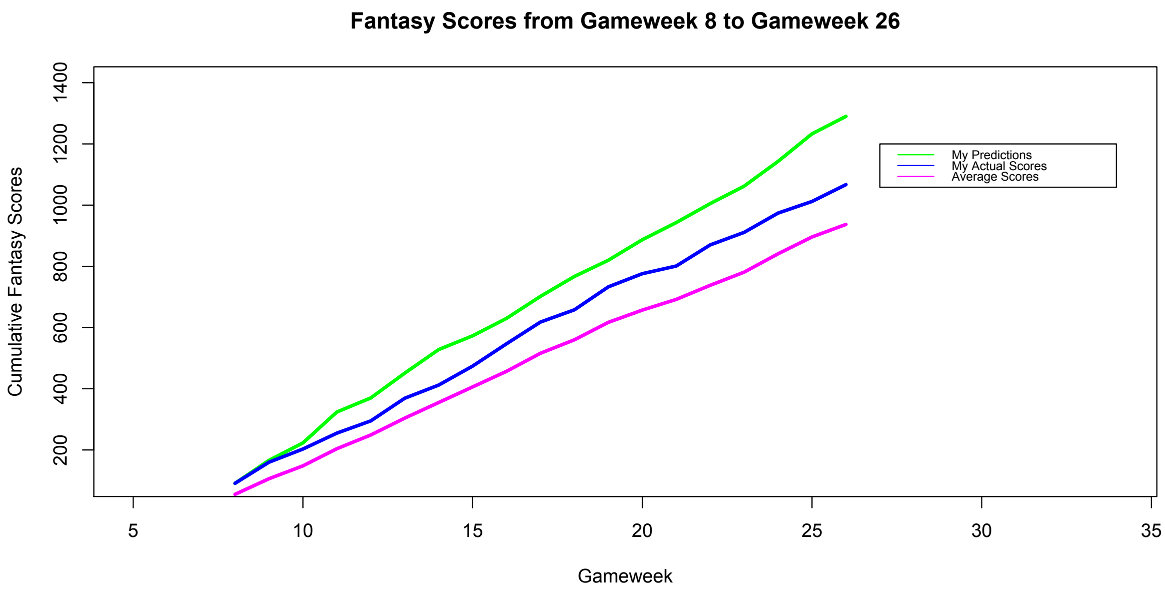
$$\sum_{i,j,k} C_{ijk} . x_{ijk} \leq 100;\; x_{ijk} = 0 \textit{ or } 1 \text{ for all } i, j, k.$$

where $x_{ijk}$ denotes a binary variable of whether *k*-th player from playing in the *i*-th type of position ( 1 = GK, 2 = DF, 3 = MF, 4 = FW) and the *j*-th team from the teams that are feature in the 35 player set (N_teams). $C_{ijk}$ denotes the cost of the *ijk*-th player in £m, and $\widehat{p_{ijk}}$ is the predicted fantasy score of the *ijk*-th player. I performed the optimisation in the *R* environment, with code adapted from Martin Eastwood's blog.

## Discussion of Results

To figure out if I've actually done alright for my model, I decided to compare the predicted scores for my model to how I actually performed. Figure 3 shows a line plot of cumulative predicted scores versus the actual scores that I attained, as well as the average scores achieved by fantasy teams in over gameweeks 8 to 26.

**Figure 3:**



Judging by the plot, I am satisfied to some extent with the performance of my model because I'd predict better scores in every gameweek from 8 to 26 than I actually did based on my instincts. The improvement in performance is a whole 223 points. That would actually put me at the **top** of my mini-league with my friends, and ranked around 86th in Singapore (~8000 jump in league positions), and around **6,000**th globally ( ~860,000 jump in league positions).

There are however, there is an important caveat (*disclaimer) to add before my model and procedures are actually usuable. Fantasy football rules actually dictate that only one transfer can be made a week. Transfers exceeding one transfer a week will incur a penalty of 4 points. So far, my predictions incur an average of about 12 transfers from week to week. This means that although I have a pretty good idea of who's going to perform well in a given week, it doesn't mean that I'd be able to bring him in without incurring a huge points penalty. I'll have to explore constructing an additional constraint in my optimisation step to make it such that I limit my number of transfers either just the one, or I can re-specify my objective function to take into account the penalty incurred by making excessive transfers.

Other areas for future work include explicitly modelling the lagged parameters by looking at auto-correlation of the performance data set. I would also consider adding more features that are not captured in the BPL Fantasy Football set, ie. real performance statistics like shots per game, dribbles per game etc.

• • • • •

## Annex

This Annex describes the features and attributes that I had used from the BPL Fantasy Football data set.

| Attributes/Features | Type | Description | Remarks |
|---|---|---|---|
| **Personal Attributes** | | | |
| id | int | Unique Player ID | |
| first_name | str | Player's First Name | |
| second_name | str | Player's Second Name | |
| team_name | str | Team which player belongs to | |
| team_id | int | Unique ID number of the team that the player belongs to | |
| type_name | str | Categorical. Whether the player is a Goalkeeper, Defender, Midfielder or Forward | |
| cost_t1 | int | Cost of player in £100,000 in the *t-1th* gameweek | Used as constraint in the optimisation step |
| minutes | int | Total number of minutes played up to the *t-th* gameweek | Used to filter out inactive players |
| **Team Attributes** | | | |
| own_team_form | int | Total number of points that the player's team has garnered from the *t-rth* to the *t-1th* gameweek | |
| own_team_pos | int | League position of the player's own team at gameweek *t* | |
| home | bool | Whether the *t-th* game will be played at the player's team home stadium | |
| next_team_t | str | The name of the team that the player is up against in the *t-th* week | |
| next_team_pos | int | The league position of the team that the player is up against in the *t-th* week | |

| **Attributes/Features** | **Type** | **Description** | **Remarks** |
|---|---|---|---|
| **Performance Attributes** | | | |
| bonus_form_t1 | int | The sum of a player's bonus points for the *t-rth* to the *t-1th* gameweek | |
| fantasy_form_t1 | int | The sum of a player's fantasy points for the *t-rth* to the *t-1th* gameweek | |
| minutes_t1 | int | The number of minutes that a player has played for from the *t-rth* to the *t-1th* gameweek | A player scores 1 point for playing up to 60 minutes, 2 points for >60 minutes |
| goals_t1 | int | The number of goals a player has scored for the *t-rth* to the *t-1th* gameweek | For each goal, GK or DEF gets 6 points, MF gets 5, and FW gets 4 points. |
| assists_t1 | int | The number of assists a player has notched for the *t-rth* to the *t-1th* gameweek | Each assist earns a player 3 points |
| clean_sheets_t1 | int | The number of clean sheets (ie. 0 goals conceded) that a player has notched for the *t-rth* to the *t-1th* gameweek | For each clean sheet, GK or DEF gets 4 points, MF gets 1 point. FW does not get any points for a cean sheet |
| goals_conceded_t1 | int | The number of goals the team has conceded whilst the player was playing for the *t-rth* to the *t-1th* gameweek | For every 2 goals conceded, GK or DEF gets deducted 1 point. FW and MF have no penalties. |
| saves_t1 | int | The number of saves a GK has made for the *t-rth* to the *t-1th* gameweek | For every 2 saves made, GK gets 1 point |
| Per Minute Features | float | The ratio of all performance attributes to the number of minutes was also included in the feature set. | |