IRON HACK

Data Analytics

Fantasy Premier League Points Prediction Using Machine Learning

Jonathon Clegg

March, 2023

Table of contents

Introduction

On February 20th, 1992, the 22 football clubs playing in the English First Division resigned from the Football League. Three months later they established the Premier League with the intention of gaining commercial independence from The Football Association (FA) and the Football League.

This event was a huge turning point in English football and one that has seen it become the most lucrative football league in the world with €5.5 billion in revenue in 2020/21[1]. As the potential market for football has grown so has the opportunity cost of not winning. This has seen a huge improvement in the professionalism of the game, affecting the way it is coached, played and consumed.

Statistical analysis has become a key tool for football clubs when it comes to “gaining an edge” over their rivals and is a key factor in decision making. It has spawned a whole new industry in sports analysis coupled with growth of the gambling industry and has changed the way people understand and discuss football.

Obviously, this growth in data use has made football fertile ground for analysis by fans. Data has become widely available and helped fans stay engaged and informed. One way in which fans can easily engage with football and statistics is through the Fantasy Premier League, or ‘FPL’ as it is commonly known. This is an online game in which users build a virtual team of players and compete to score the most points each week based on the players real life performance.

I am personally a big consumer of football and enjoy looking at it through the lens of statistical analysis as it combines two of my interests. I have been playing FPL for several years and consult the data on a regular basis to inform my team choices; however my performance is still very variable. Through this project, I hope to develop a machine learning algorithm that will be a much better predictor of the number of points a player will get each week than I am, currently.

[1] https://fortune.com/2023/01/19/english-premier-league-dominate-european-tournaments-deloitte-rich-money-league-soccer-clubs/

Methodology/Plan

1. Obtain data from the relevant sources
2. Load data into Python through CSVs, APIs and Web Scraping
3. Convert data to data frames and clean
4. Export clean data to MySQL
5. Join datasets for analysis
6. Run appropriate queries to process the data for analysis
7. Decide on appropriate Regression and Classifier models for predicting number of points
8. Feature Selection and Engineering
9. Predict the number of points each player will get based on relevant variables

Data sources and Data

For this project we will be harvesting data from 3 sources: Fantasy Premier League (FPL), FiveThirtyEight (538) and Understat.

FPL

FPL is an online game where players create a virtual team consisting of football players from the English Premier League (EPL). The aim of the game is to get the most points possible based on the performance of the players in real-life matches throughout the season.

This is the principal source of data and will provide the independent variable(s) for modeling. The website provides many statistics at the player and match level based on in-game events. They also provide many of their own engineered features such as the ICT index based on the Influence, Creativity and Threat each player provides in a given game.

538

FiveThirtyEight (538) is a website originally known for political predictions. They also have a lot of statistics for given sports and leagues around the world including the Premier League.

One of its most interesting stats is the Soccer Power Index (SPI) which rates teams relative performance and gives an indication of how good each team is relative to their opponent. Based on this they are able to project the win, loss and draw likelihood for each game.

Undertat

Understat is a website that provides football statistics and insight for various football teams globally. Understat provides a “precise method for shot quality evaluation” through its expected goals (xG) features, which can be used to calculate the likelihood of a shot resulting in a goal based on factors such as shot location, shot type, and the position of defenders and the goalkeeper. They provide a number of statistics around expected goals and assists which have become the new benchmark for evaluating and gaining a deeper understanding of a team's performance beyond simply goals scored.

Data Description

1. Fantasy Premier League Data

FPL data comes from the github repository https://github.com/vaastav/Fantasy-Premier-League which itself has been harvested from https://fantasy.premierleague.com/.

The data has been merged from 3 separate csv files, each containing data from individual seasons going back to 2020/2021. The available data contains the following variables:

Column Name	Description
name	Player name
position	Player position
team	The team that said player plays for
xP	Expected points based on performance in the game
assists	Number of passes provided that led to a goal
bonus	Number of bonus points awards from 0-3 based on players relative performance during the game
bps	Total number of points awarded to a player based on players relative performance during the game. The outcome of which determines the bonus.
clean_sheets	0 if the player’s team conceded at least 1 goal. 1 if they did not concede any goals.
creativity	Creativity assesses player performance in terms of producing goal scoring opportunities for others. It can be used as a guide to identify the players most likely to

	supply assists. While this analyzes frequency of passing and crossing, it also considers pitch location and the incisiveness of the final ball.
goals_conceded	Number of goals conceded by the player or players team
goals_scored	Number of goals scored by the player
ict_index	A football statistical index developed specifically to assess a player as an FPL asset. It uses match event data to generate a single score for three key areas – Influence, Creativity and Threat. These figures then combine to create an individual’s ICT Index score. It condenses more than 40 match event statistics into four distinct scores. These offer a view on player performance for factors that are known to produce FPL points.
influence	This evaluates the degree to which that player has made an impact on a match, or matches over the season. It takes into account events and actions that could directly or indirectly affect the match outcome. At the very top level these are decisive actions like goals and assists. However, the Influence score also processes significant defensive actions to analyze the effectiveness of defenders and goalkeepers.
minutes	Number of minutes played
own_goals	Number of own goals scored by a player
penalties_missed	Number of penalties scored by a player
penalties_saved	Number of penalties saved by a player (only related to goal keepers)
red_cards	1 if the player received a red card during the game . 0 if they did not.
round	Which round each game is based on from a scale of 1 to 38
saves	Number of saves made by a goalkeeper
selected	Number of teams a player has been selected by prior to a particular game

team_a_score	Number of goals scored by the away team
team_h_score	Number of goals scored by the home team
threat	Threat is a value that examines a player’s threat on goal; it therefore gauges those individuals most likely to score goals. While attempts are the key action, the Index looks at pitch location, giving greater weight to actions that are regarded as the best openings to register a goal. All three of these scores are then combined to create an overall ICT Index score. That then offers a single figure that presents a view on that player as an FPL asset.
total_points	Total points received by a player that week
transfers_in	Number of users who transferred a player into their team for that specific game week
transfers_out	Number of users who transferred a player out of their team for that specific game week
was_home	1 if the player’s team was playing at home. 0 if they were playing away.
yellow_cards	1 if the player received a yellow card during the game . 0 if they did not.

2. Understat

Understat data comes from web scraping their website (https://understat.com) for individual player data. The available data contains the following variables:

Column Name	**Description**
player_id	Unique numerical value associated with each player
player_name	Players first name and last name
goals	Number of goals scored
shots	Number of shots taken
xG	Expected Goals

time	Number of minutes played
position	Playing position
h_team	The home team for a given match
a_team	The away team for a given match
h_goals	Number of goals scored by the home team
a_goals	Number of goals scored by the away team
date	Match date
id	Match id
season	Season
xA	The sum of expected goals off shots from a players key passes
assists	Passes that lead to goals
key_passes	Passes that lead to a shot
npg	Non penalty goals
npxG	Non penalty Expected goals
xGChain	Total xG of every possession the player is involved in
xGBuildup	Total xG of every possession the player is involved in without key passes and shots

3. FiveThirtyEight (538)

The 538 API, https://projects.fivethirtyeight.com/soccer-api/club/spi_matches.csv was used to obtain statistics at the tThank eam level for individual matches and will be used to generate statistics about a players teams as well as a players opponent. The available data contains the following variables:

Column Name	Description
season	The season during which the match was played
date	The date of the match (YYYY-MM-DD)
league_id	A unique identifier for the league this match was played in
league	The name of the league this match was played in
team1	The home team's name
team2	The away team's name
spi1	The home team's overall SPI rating before the match
spi2	The away team's overall SPI rating before the match
prob1	The probability of the home team winning the match
prob2	The probability of the away team winning the match
probtie	The probability of match ending in a draw (if applicable)
proj_score1	The number of goals we expected the home team to score
proj_score2	The number of goals we expected the away team to score
importance1	The importance of the match for the home team (0-100)
importance2	The importance of the match for the away team (0-100)
score1	The number of goals scored by the home team
score2	The number of goals scored by the away team

xg1	The number of expected goals created by the home team
xg2	The number of expected goals created by the away team
nsxg1	The number of non-shot expected goals created by the home team
nsxg2	The number of non-shot expected goals created by the away team
adj_score1	The number of goals scored by the home team, adjusted for game state
adj_score2	The number of goals scored by the home team, adjusted for game state

Data collection

The data listed in the previous section was collected through a combination of API, Web Scraping and downloading existing datasets as csv files. The data was parsed using python, ready for cleaning and exploratory data analysis, which will be outlined in the next section.

Data cleaning

FPL

The data from FPL is relatively clean with very few missing values. The only issue is that features vary across seasons. This makes joining the data into a single table more difficult.

For the current season several new variables were added. Although these will be available for future predictions they are not very useful for training the model as they only go back several months. These variables will be dropped from the FPL dataset but will be replaced by similar statistics where available.

> *(expected_assists, expected_goal_involvements, expected_goals, expected_goals_conceded, starts)*

Understat

Data from Understat was obtained through web scraping in python. In order to obtain the relevant player data by match it was necessary to first scrap all match data on the EPL for the past 3 seasons. From this it was possible to identify the player ids of every player that had played in the EPL over that time frame. With these ids each player could be iterated over and the website scraped for match specific data.

Since Understat collects data from other leagues around the world and since some players have also played in those other leagues, the data needed to be filtered to include only teams that have played in the EPL since 2020.

538

The data was harvested using the 538 API and filtering by league and season. 538 provides some very interesting statistics at the match level that should help make the points predictor model more accurate.

Data Types

Data type for each column was set manually with the relevant type chosen for each variable, including Int, Float, String, Date.

Kickoff time in FPL data was converted to date to match the format in Understat data

Naming Issues

In order to join tables together, player names, team names and match dates had to match. There were several issues that had to be corrected before this could happen

1. Team Names were edited in FPL and 538 data to match those used by Understat. E.g Man United became Manchester United.
2. Player names were cleaned in both datasets.
    - Unidecode was used to remove accents from names.
    - Underscores were removed etc.
3. A bigger issue was that a large number of player names were different across the 2 datasets. ‘Cristinao Ronaldo’ was listed in the Understat data but named ‘Cristiano Ronaldo dos Santos Aveiro’ in the FPL data set. This seems to be a particular problem for Brazilian and Portuguese players who go by nicknames and often the same one.

    To try and match names the fuzzywuzzy library in python was used for fuzzy string matching. Fuzzywuzzy was chosen over other methods such as Levenshtein because it has more advanced matching capabilities, uses a combination of different algorithms to match strings and takes into account the context of the strings.

    Matches were manually checked to make sure they were correct. If there was a doubt they were checked across both websites to make sure the correct player was identified.

4. The dates for one match had been incorrectly entered for Understat data and were manually edited.

Missing data

Luckily there was no missing data for FPL and Understat data.

For 538 however there were 3 observations missing data for h_team_importance and a_team_importance. Since this is a measure we want to look at, it was manually filled. For 2 of the points, the importance scores for the same fixture the season before were used, because they occurred at roughly the same time, earlier in the season when there are still many games to play and so importance was less variable.

For the final missing value, an average of the game importance for the 2 games prior and after was used to try and mitigate effects caused by varying team quality.

Outliers

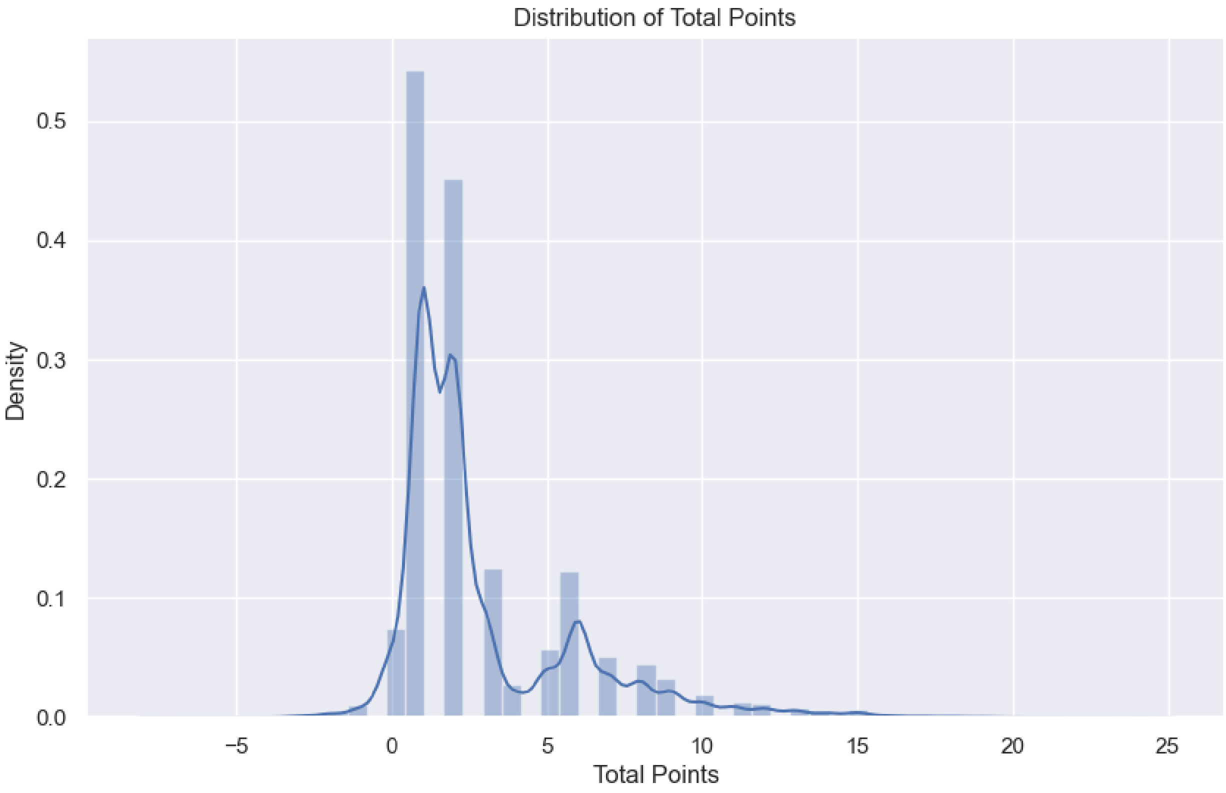
Although outliers have been identified, they will be kept in the dataset as they may provide insights later. These reflect real life players who may naturally perform at a higher level and be vital for having in your team.

Duplicate Data

There were no duplicate observations in my data. Each row is uniquely identified by a match_id and a player_id.

Exploratory Data Analysis (EDA)

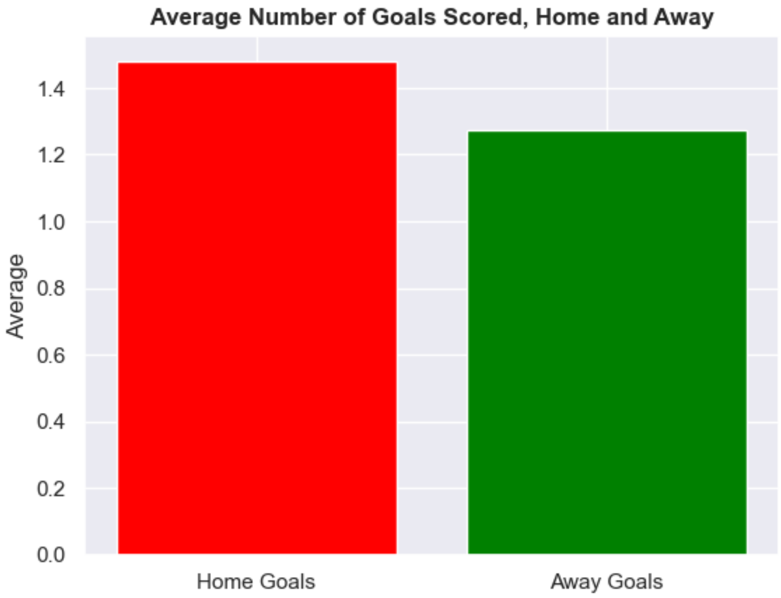
The goal of this project is to select the team that will score the highest number of points in a given game week. It therefore makes sense to analyze the distribution of the anticipated independent variable and correlation of between the variables, to see how they are related. Below are the results of this initial analysis.



As you can see, total_points has a bimodal distribution with peaks at 1, 2 and 6 points. This makes sense based on the game play and how points are distributed. Please see appendix 1 for a detailed breakdown of how points are awarded in FPL.

If a player plays at least 60 minutes of a game they get 2 points no matter what else happened in the game. They get extra points for things like scoring and for not conceding goals but the majority of players don't score each week, nor do they keep a clean sheet. This can be observed by looking at the average number of goals scored by home teams and away teams.

The average number of goals scored by the home team is: 1.48 and the average number of goals scored by the home team is: 1.27, meaning most games are low scoring both in terms of goals and points per player.



The other peak is around 6 points. This is because players are rewarded with 4 points for events linked to their position. If a defender keeps a clean sheet they get 4 points. If an attacker scores a goal they get 4 points. So if a player performs well, you'd expect them to get, on average, 6 points.

This lack of variation in the explanatory variable might cause problems when running regression models. Given its bimodal nature, a classifier model may be more appropriate with independent variables focussed on the events that players get points for. For example, if a player scores or not, or if a player keeps a clean sheet or not.

We will now look at the correlation between total_points and our dependent variables. In general, the dependent variables can be split into 2 categories:

1. Player-specific data based on performance (total points, goals, assists, etc.)
2. Match specific data (home team, away team, SPI ranking for home team and away team, etc.)

Both require treatment in different ways and will be looked at in turn, starting with player performance data.

Player Performance variables

Just by looking at the rules for assigning points in appendix 1, it is clear that players get points for different things. For example, Goalkeepers get points for making saves whereas no other players do. Defenders get points for clean sheets whereas Forwards don’t. It is therefore wise to break correlation analysis down by position.

From the heatmap on the following page it is obvious that different variables impact total points for different players. For example, for Goalkeepers and Defenders, total points is highly correlated with goals conceded and clean sheets as they are awarded points for these criteria.

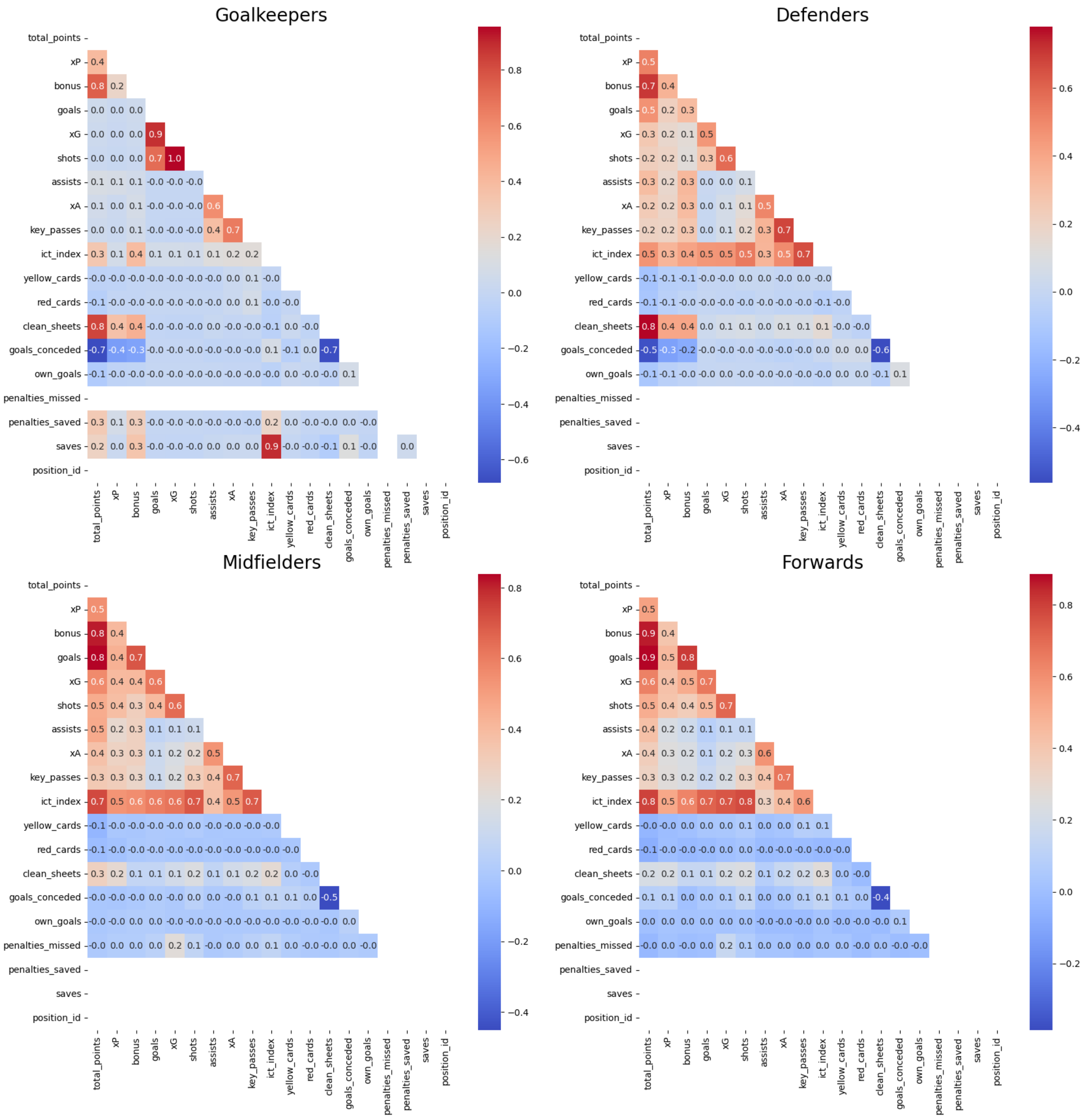
Interestingly, there is still some correlation between goals conceded and total points for forwards, despite the fact they are not awarded points for this. This may be due to the fact that conceding goals is a good indicator of a team's overall quality.

All positions are correlated with the ICT index which is a composite measure of how creative, influential and threatening a player is. This makes sense since the ICT index is composed using other criteria. This can be seen in the correlation matrix since ICT index is highly correlated with measurements such as goals, shots, assists, xP, and xG. It is more highly correlated with total points for forwards and midfielders who are rewarded for such attributes.

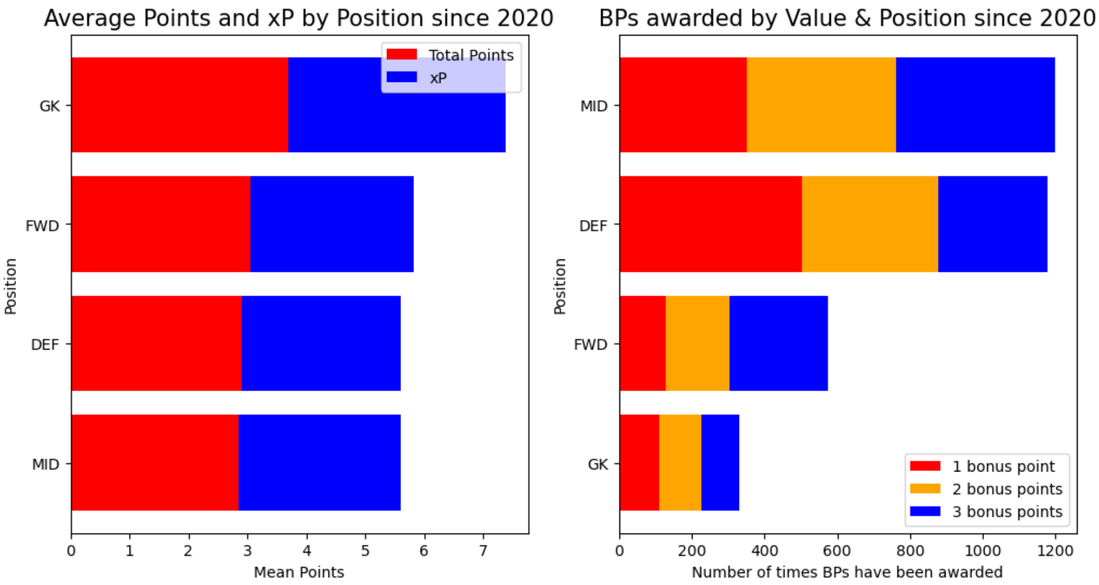
Midfielders and Forwards total points are also highly correlated with goals scored which forms a big part of their reward structure. For defenders this correlation is less as they do not score as many goals and so it plays a lesser role in total points. For goalkeepers who hardly ever score, the correlation is 0.

All positions are highly correlated with bonus points (BPs) as expected as all players can obtain bonus points through their performance. Looking more deeply however, on the following chart on page 20, we can see that the likelihood of being awarded bonus points is heavily skewed towards midfielders and defenders.

Correlation Matrix for Different Positions



Even if we account for the fact there are more midfielders and defenders in the game, this is a significant difference. It makes sense however, for 2 reasons. Both defenders and midfielders can earn rewards for defensive performance as well as attacking performance, and they are more likely to be in possession of the ball and so able to rack up points in the bonus point system for completed passes etc.



Interestingly, we can also see from the first graph that goalkeepers earn a higher number of points on average than other positions.

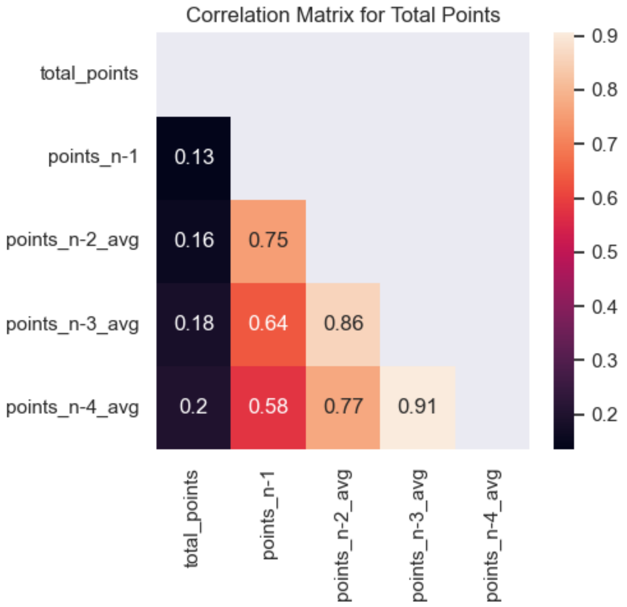
From this analysis we can see that position has a big impact on how points are generated. This means that fitting one model for all positions will likely damage the accuracy and fit. Therefore we should look at modeling for each position individually.

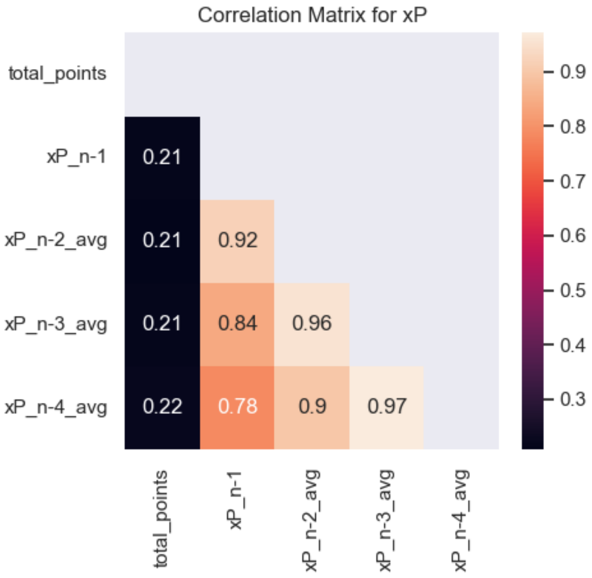
In the heatmap above we were looking at which variables affect total points in a match, but of course we are trying to make predictions about the number of points a player will get without this information. Therefore, past performance, or lagged variables must be used instead.

FPL usually provides a “form” statistic for each player defined as, “a player's average score per match, calculated from all matches played by his club in the last 30 days.” Unfortunately this data is not to hand, but it is possible to recreate it.

To do so lagged variables of ‘total_points’ for the past 4 weeks were created and the mean value taken. The same was done for expected points (xP) to try and remove the impact of random events from the points scored.

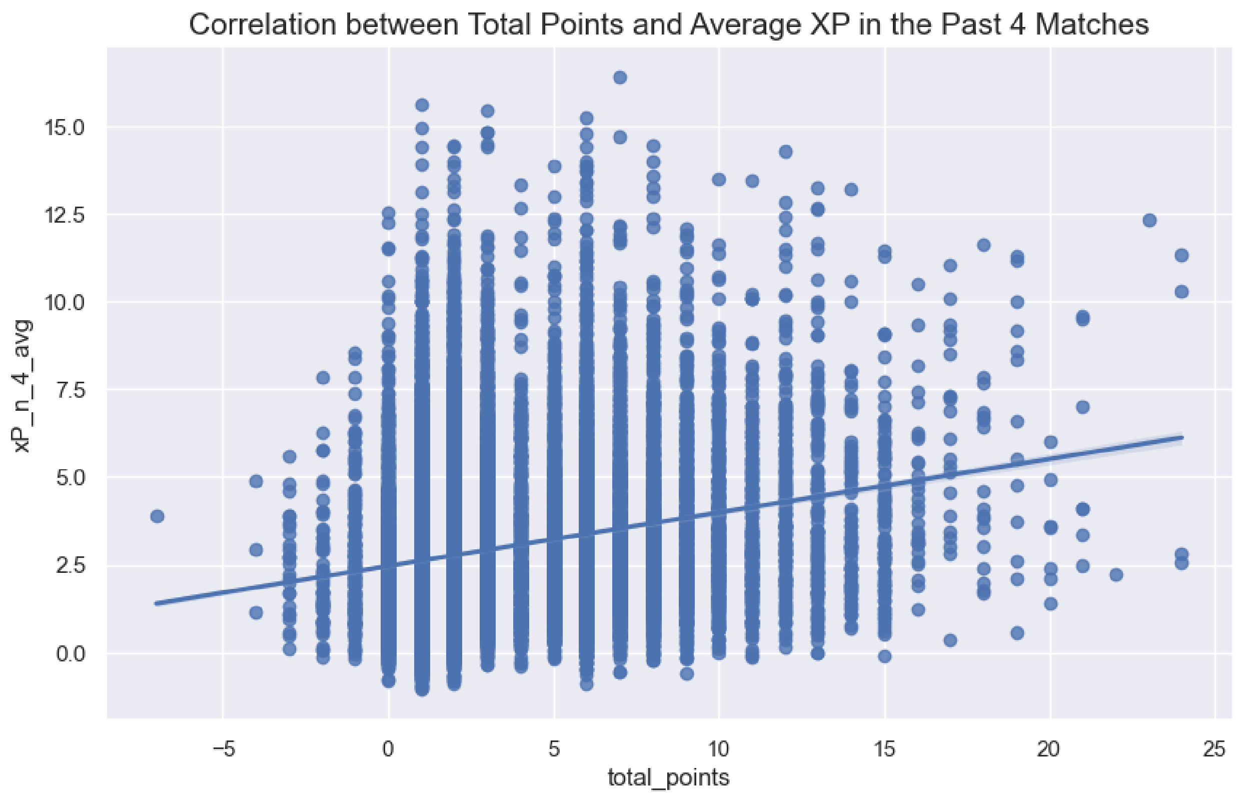
In the correlation heatmaps below you can see the correlation between total points in week N and mean average of total points earnt in weeks N-1, N-2, N-3 and N-4.





You can see that the lagged 4 week moving average, similar to the FPL “form” statistic, is correlated the most (0.2) with total_points amongst our lagged variables. The lagged 4 week moving avg of expected points (xP), however, correlates even better (0.22) as it rewards performance over output. xP is based on how many points a player should get and calculated on the underlying statistics which are a better gauge of “form” rather than actual points which can be affected by the team as a whole.

From the graph below we can visualize this correlation. Although relatively low, this is to be expected, as total points is influenced by many factors, a lot of them uncontrollable such as the performance of teammates or impossible to collect such as a players mental health, or the energy of the crowd.

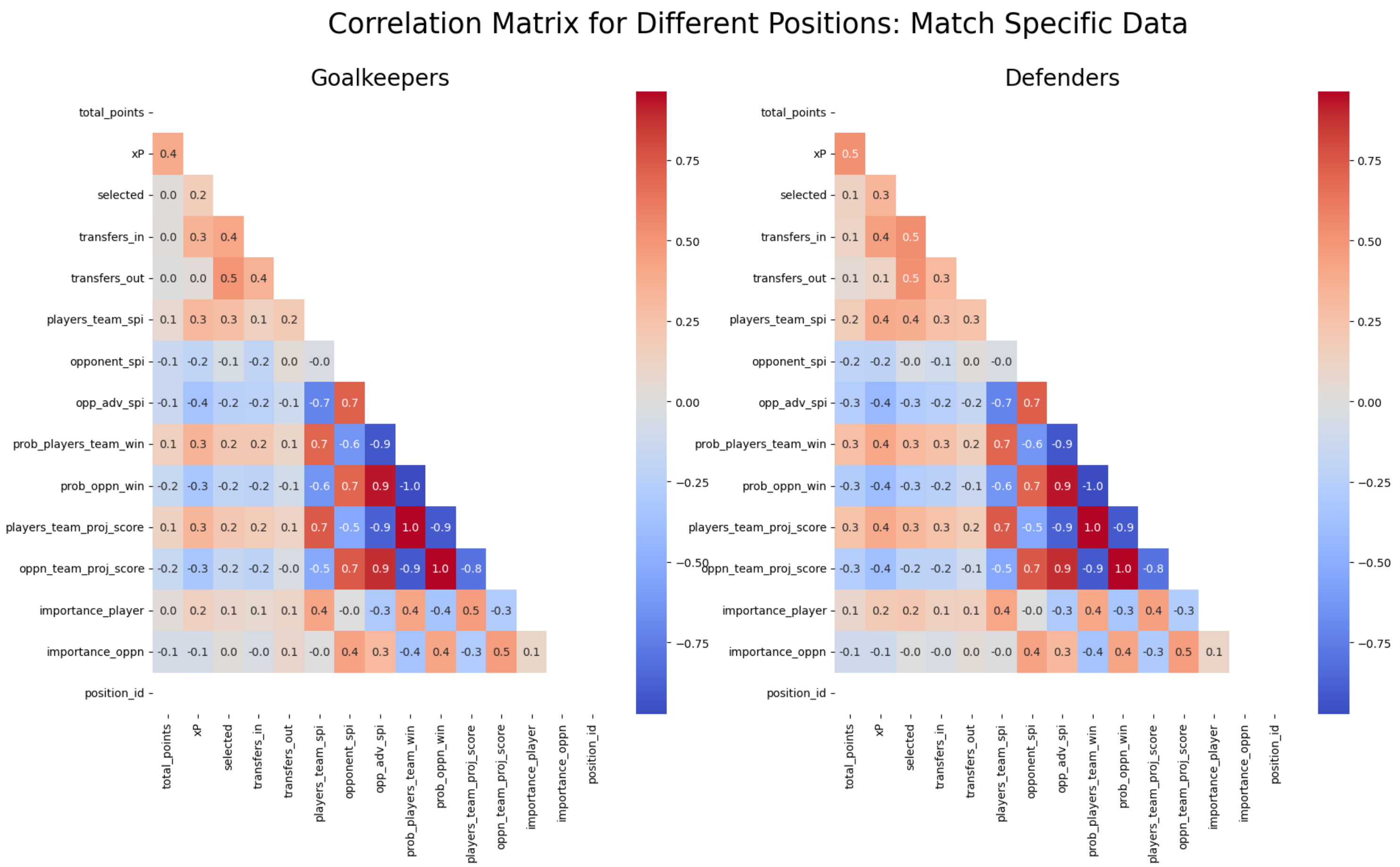


The same test will be done for other player specific variables such as goals, expected goals, ict_index, assists, expected assists etc to find the best lagged moving average.

Match Specific variables

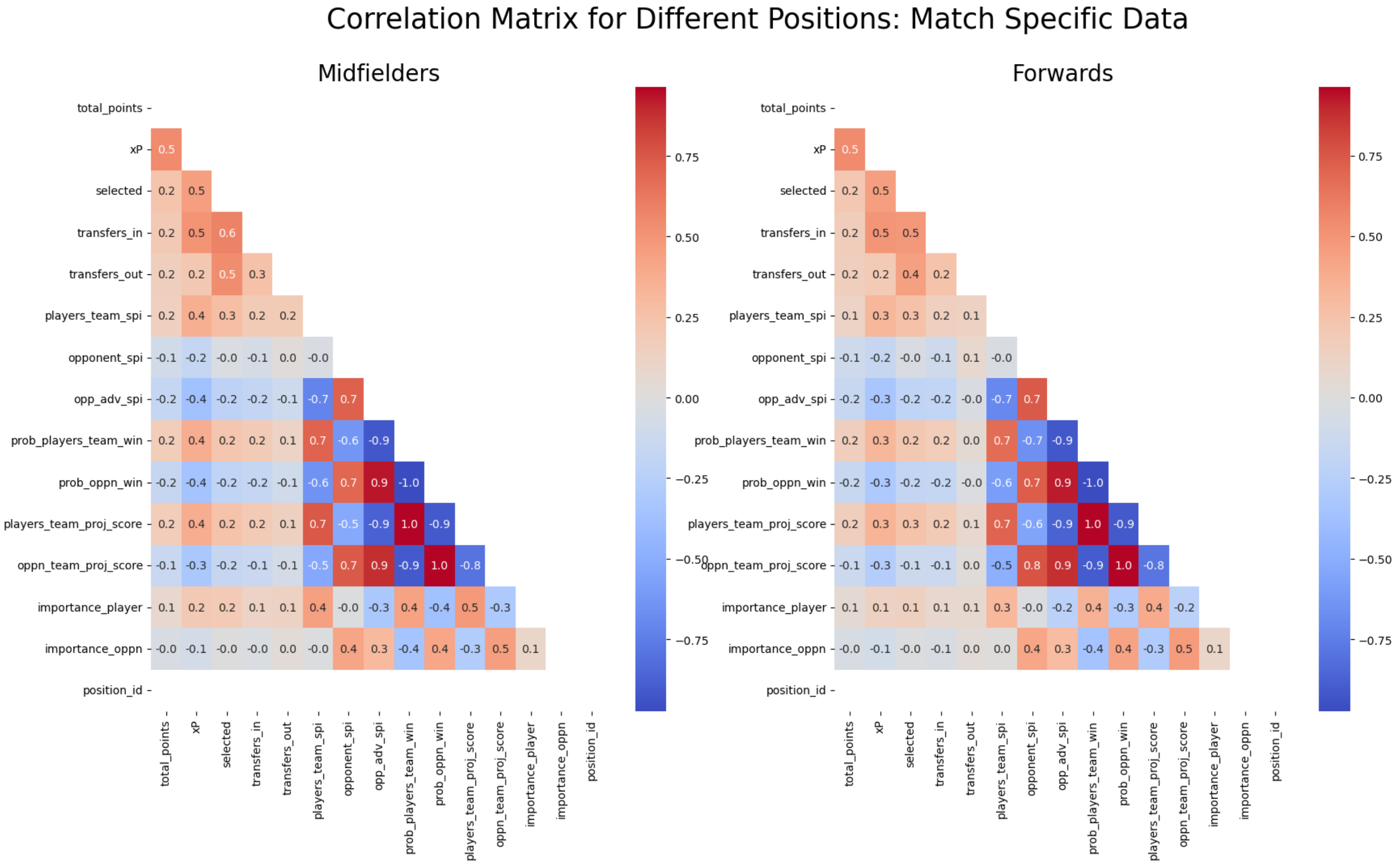
In order to look at the impact of match specific variables on players total points we first need to adjust them from ‘home’ and ‘away’ to ‘players_team’ and ‘opponents_team’. This was done by identifying if the player in question was home or away and filtering the data accordingly.

Now we will analyze the match specific variables for each position by looking at a correlation heat map for each position.


Correlation Matrix for Different Positions: Match Specific Data

Looking at the heatmaps we can see that there is weak but consistent correlation between total points and match specific variables. Correlation also varies by position, giving more weight to the idea that we should split our modeling by position.

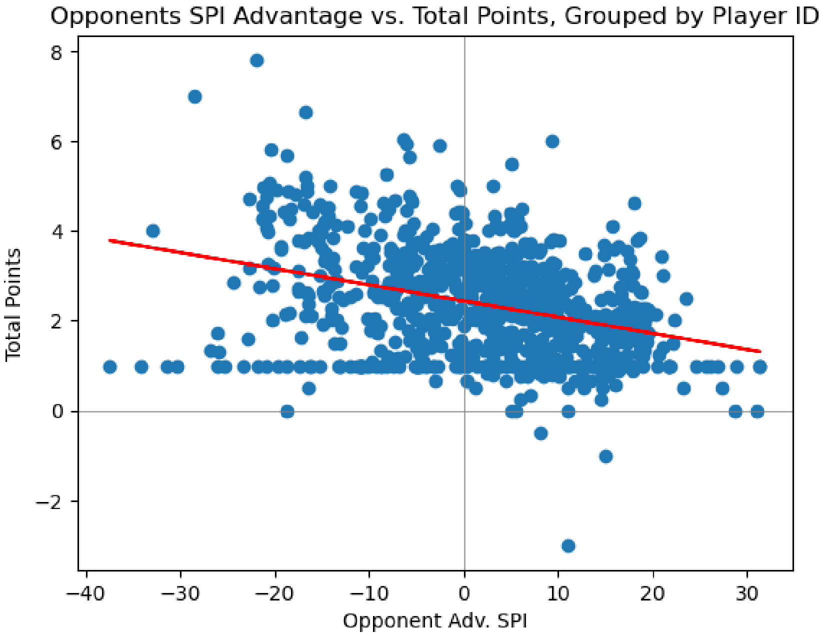
Correlation is much stronger when looking at xP (which can be considered a proxy for performance) indicating they are a good predictor of performance but not necessarily outcome, which is affected by more random events. This is important for us as we will in effect be modeling expected points.



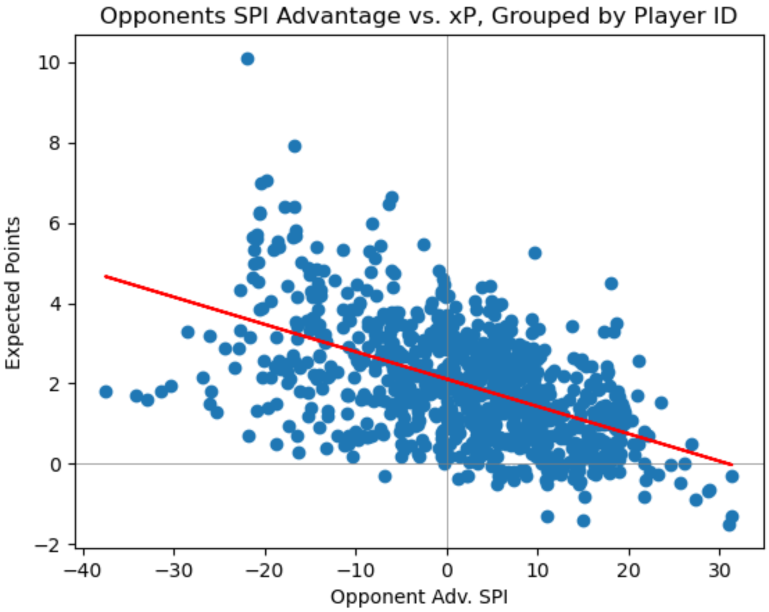
Logically the match specific variables are highly correlated with each other since they reflect 2 sides of the same coin. Probability of your team winning is obviously highly correlated with the projected score for your team in the same game and highly negatively correlated with the opponents projected score. Some of these variables will therefore be dropped to avoid overfitting the model.

Below is another visual representation of one of these match specific data points. By taking the differences between teams SPI rankings we are able to create a variable that shows how big an opponent's advantage is over a player's team, with positive numbers representing an advantage for the opponent and negative numbers representing an advantage for the players team.

When grouping by player_id and looking at average total points vs average team advantage we can see the same negative correlation as was seen in the heatmap.



Looking at xP, we can see that this correlation becomes stronger, as was also witnessed in the heatmap analysis.



Database type selection
SQL or NoSQL

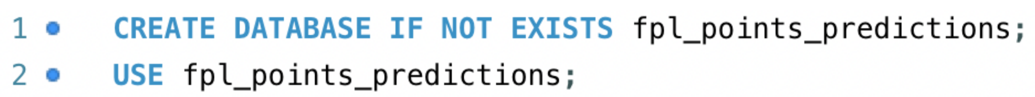
After researching the differences, advantages and disadvantages of each database type, the major differences seem to be the following:

1. SQL databases are relational, NoSQL databases are non-relational
2. SQL databases use structured query language and have a predefined schema NoSQL databases have dynamic schemas for unstructured data
3. SQL databases are vertically scalable, while NoSQL databases are horizontally scalable
4. SQL databases are table-based, while NoSQL databases are document, key-value, graph, or wide-column stores
5. SQL databases are better for multi-row transactions, while NoSQL is better for unstructured data like documents or JSON

It seems appropriate since we have structured tables with a predefined schema to use SQL. Furthermore, if we expect to add more data or other sources, SQL would be optimal due to vertical scalability and also the ability to use multi-row transactions.

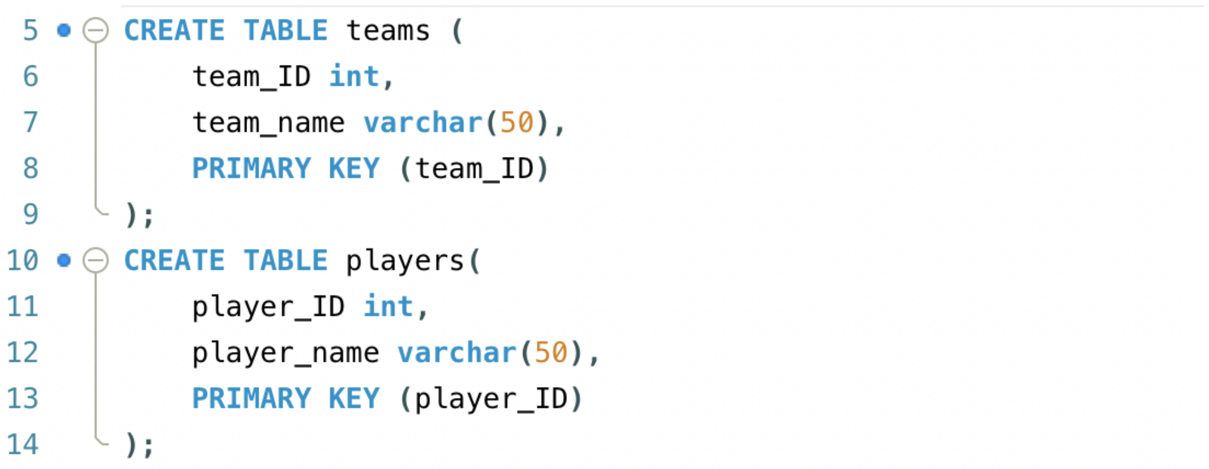
MySQL Database

A database in MySQL was first created.



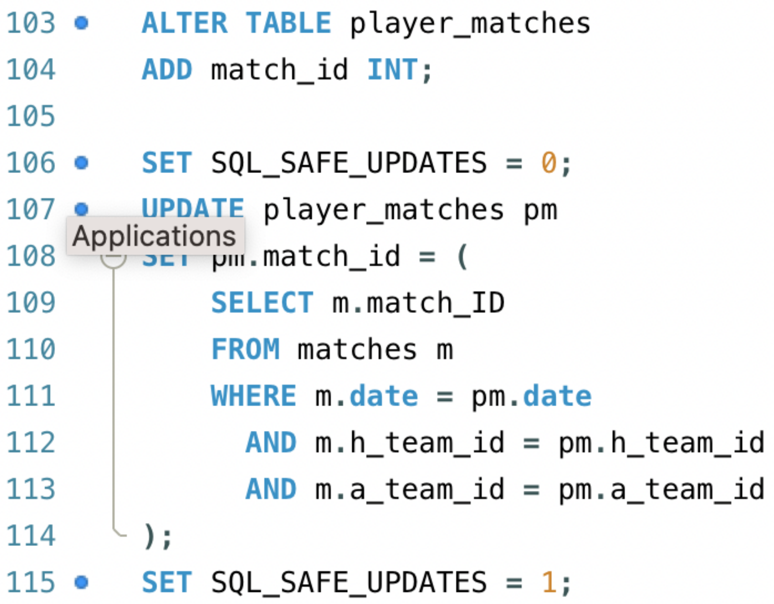
```sql
CREATE DATABASE IF NOT EXISTS fpl_points_predictions;
USE fpl_points_predictions;
```

Then the relevant tables were created before importing the data.



```sql
CREATE TABLE teams (
    team_ID int,
    team_name varchar(50),
    PRIMARY KEY (team_ID)
);
CREATE TABLE players(
    player_ID int,
    player_name varchar(50),
    PRIMARY KEY (player_ID)
);
```

Several adjustments were made in MySQL such as adding a Primary Key, match_id, to the matches table and adding matches_id as a foreign key in the player_matches table.

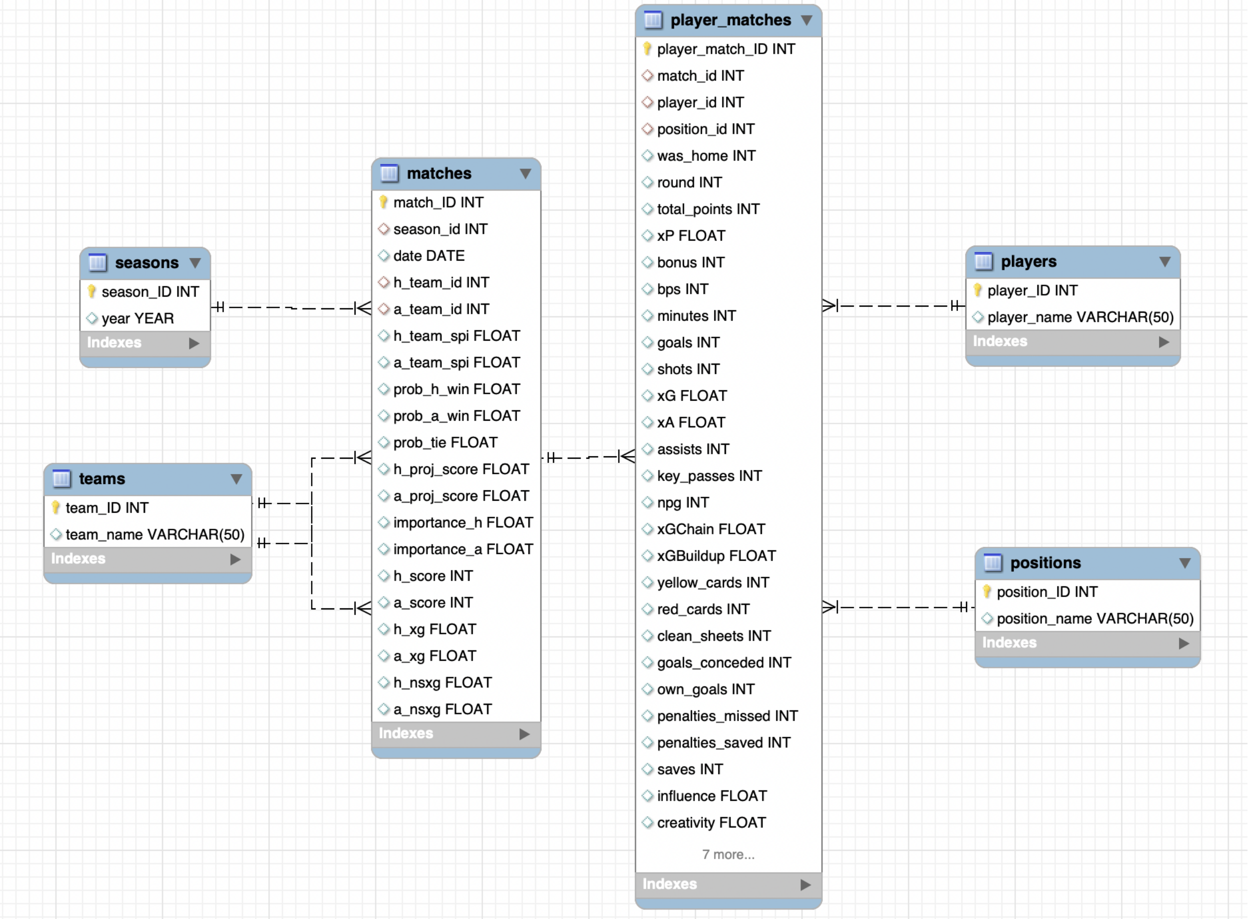


```sql
ALTER TABLE player_matches
ADD match_id INT;

SET SQL_SAFE_UPDATES = 0;
UPDATE player_matches pm
SET pm.match_id = (
    SELECT m.match_ID
    FROM matches m
    WHERE m.date = pm.date
     AND m.h_team_id = pm.h_team_id
     AND m.a_team_id = pm.a_team_id
);
SET SQL_SAFE_UPDATES = 1;
```

ERD

Here is the Entity Relation Diagram (ERD) summarizing the relationships between our tables:



MySQL Queries

Queries

1. The 5 players with the highest points total over since the 2020/2021 season

```sql
SELECT pm.player_id AS "Player ID", p.player_name AS "Player Name",
        SUM(pm.total_points) AS "Total Points"
FROM player_matches pm
LEFT JOIN players p
ON pm.player_id = p.player_ID
GROUP BY pm.player_id
ORDER BY SUM(pm.total_points) DESC
LIMIT 5;
```

Player ID	Player Name	Total Points
619	mohamed salah	652
318	harry kane	599
794	son heung-min	576
110	bruno fernandes	517
838	trent alexander-arnold	470

2. The 5 players with the highest points total, relative to expected points since the 2020/2021 season

```sql
SELECT pm.player_id AS "Player ID", p.player_name AS "Player Name",
        SUM(pm.total_points) AS "Total Points",
        ROUND(SUM(pm.xP),2) AS "Total Expected Points",
        ROUND(SUM(pm.total_points) - SUM(pm.xP),2) AS "Total Points - xP"
FROM player_matches pm
LEFT JOIN players p
ON pm.player_id = p.player_ID
GROUP BY pm.player_id
ORDER BY SUM(total_points - xP) DESC
LIMIT 5;
```

Player ID	Player Name	Total Points	Total Expected Points	Total Points - xP
120	callum wilson	282	194.7	87.3
39	allan saint-maximin	247	188.5	58.5
685	pascal gross	313	257.7	55.3
348	ivan toney	281	227.5	53.5
623	morgan gibbs-white	113	62	51

3. The 5 players who have over performed their expected points total the most times since the 2020/2021 season

```
WITH player_performance AS (
    SELECT pm.player_id, p.player_name,
    ((pm.total_points) - (pm.xP)) AS "total_points - xP",
    CASE
        WHEN (pm.total_points) - (pm.xP) > 0 THEN 1
        ELSE 0
    END AS over_performance
    FROM player_matches pm
    LEFT JOIN players p
    ON pm.player_id = p.player_ID
)
SELECT player_name AS "Player Name",
        SUM(over_performance) AS "# of Over Performances"
FROM player_performance
GROUP BY player_name
ORDER BY SUM(over_performance) DESC
LIMIT 5;
```

Player Name	# of Over Performances
joao moutinho	50
conor coady	49
ruben neves	49
douglas luiz	48
dwight mcneil	48

4. The top 5 performing teams in the 2022/2023 season ordered by Total Points.

```
SELECT t.team_name AS "Team", SUM(pm.total_points) "Total Points",
    SUM(pm.goals) AS "Goals",
    SUM(pm.assists) AS "Assists",
    SUM(pm.goals_conceded) AS " Goals Conceded",
    SUM(pm.bonus) AS "Bonus Points",
    RANK() OVER (ORDER BY SUM(pm.bonus) DESC) AS "Bonus Points Index"
FROM player_matches pm
LEFT JOIN teams t
ON pm.player_team_id = t.team_ID
WHERE pm.season_id = 3
GROUP BY pm.player_team_id
ORDER BY SUM(pm.total_points) DESC
LIMIT 5;
```

Team	Total Points	Goals	Assists	Goals Conceded	Bonus Points	Bonus Points Index
Arsenal	1471	60	42	297	133	1
Manchester City	1420	65	50	286	131	2
Newcastle United	1283	34	21	187	105	4
Liverpool	1255	43	35	319	99	5
Manchester United	1238	40	32	385	114	3

5. The top 5 players with the highest goal tally in the 2022/2023 season so far, split both home and away.

```
SELECT p.player_name AS "Player Name",
       SUM(CASE WHEN pm.was_home = 1 THEN pm.goals ELSE 0 END) AS "Home Goals",
       SUM(CASE WHEN pm.was_home = 0 THEN pm.goals ELSE 0 END) AS "Away Goals",
       SUM(pm.goals) AS "Total Goals",
       SUM(pm.total_points) AS "Total Points"
FROM player_matches pm
LEFT JOIN players p ON pm.player_id = p.player_ID
WHERE pm.season_id = 3
GROUP BY pm.player_id
ORDER BY SUM(pm.goals) DESC
LIMIT 5;
```

Player Name	Home Goals	Away Goals	Total Goals	Total Points
erling haaland	19	9	28	203
harry kane	9	9	18	165
ivan toney	8	8	16	142
marcus rashford	11	4	15	158
gabriel martinelli	7	5	12	152

Conclusion

From the EDA and the SQL queries performed above we can draw some general conclusions that will help us with building a statistical model to predict the number of points each player will get each week.

Looking at total points, we can see that the distribution is bimodal. As explained this, lack of variation in the explanatory variable might lead us to use a classifier for developing our model. At the very least we should keep in mind that regressions may not perform as well.

We have also seen that points are impacted by the position of a player, both in terms of overall points and which variables impact point generation. It is therefore necessary to look at each position individually to improve accuracy of the model.

The team you play for is also a big factor in points generation. This is because the best teams can afford the best players, but also because the best players play with other great players. Having a good team boosts your personal chances of scoring a goal, getting an assist or keeping a clean sheet and so impacts your own points. It may not positively affect bonus however, as you are now competing with other good players for the same pool of points.

Player performance variables will need to be lagged. Most likely a lagged 4 week moving average will be taken but each variable will be checked to see what fits best.

Match specific data is important for our model and is correlated with total points but even more correlated with expected points which act as a proxy for performance. Since xP and total points are highly correlated these will be important factors.

Other questions to consider during the model selection and design process include feature selection, which will be tested for, and whether to limit observations to players who have played 60 minutes or above. This will remove the variation in points due to minutes played and in theory make the model more stable.

APPENDIX

Appendix 1: How players generate points in the Fantasy Premier League.

Action	Points
For playing up to 60 minutes	1
For playing 60 minutes or more (excluding stoppage time)	2
For each goal scored by a goalkeeper or defender	6
For each goal scored by a midfielder	5
For each goal scored by a forward	4
For each assist for a goal	3
For a clean sheet by a goalkeeper or defender	4
For a clean sheet by a midfielder	1
For every 3 shots saved by a goalkeeper	1
For each penalty save	5
For each penalty miss	-2
Bonus points for the best players in a match	1-3
For every 2 goals conceded by a goalkeeper or defender	-1
For each yellow card	-1
For each red card	-3
For each own goal	-2

Appendix 2: Link to Trello Board used for project planning

https://trello.com/b/vrNaMqfo/fplpointsprediction

Appendix 3: Github link

https://github.com/JonathonClegg/fpl_points_prediction