Volume 40(1), pp. 69–107
http://orion.journals.ac.za

ORiON
ISSN 0259–191X (print)
ISSN 2224–0004 (online)

An optimisation approach towards soccer Fantasy Premiere League team selection

V Venter* and JH van Vuuren†

Received: 24 January 2023; Accepted: 3 March 2024

Abstract

Fantasy Premiere League (FPL) is a popular online sports prediction game based on the well-known (soccer) *English Premiere League* (EPL). In FPL, each participant forms a series of imaginary teams over the course of a season of the EPL, each composed of real-world soccer players. Points are then awarded to participants based on the real-world performances of the players in the EPL. The goal is to accumulate as many points as possible during the season. FPL participants are allocated a budget for team selection which gives rise to the constrained team selection optimisation problem of filling playing positions in the team for each game week of the FPL season. This team selection problem is further complicated by the facts that player performance is not known in advance with certainty and that participants may only make limited changes to their team compositions in the form of transfers during any game week. In this paper we adopt a combinatorial optimisation approach towards participating in FPL, in which future player performances are forecast by statistical and machine learning techniques. We demonstrate retrospectively that our approach would have placed within the top 4% of players worldwide during the 2020/2021 FPL season.

Key words: Sports team selection, Optimisation, Forecasting, Decision support, Fantasy Premiere League.

*Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: vanzylventer@gmail.com

†**(Fellow of the Operations Research Society of South Africa)** Corresponding author. Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa, email: vuuren@sun.ac.za

http://dx.doi.org/10.5784/ - -004

1 Introduction

Fantasy sports is a game genre, usually played online, in which participants assemble imaginary or virtual teams composed of real players of a particular professional sport. One of the largest fantasy sport leagues in the world is the *Fantasy Premiere League* (FPL), which boasted more than 8 million participants during the iteration of the game based on the 2020/2021 *English Premier League* (EPL) soccer[1] season.

Participants of the FPL, usually referred to as *managers*, receive points based on the real-world performances of the players in their teams. These performances are converted into points for each match played and the manager with the most points at the end of the season is crowned the winner.

The popularity of the FPL has grown to such an extent that major prizes are awarded to the winners of the league. For example, the prizes for first place in the FPL, based on the 2021/2022 EPL soccer season, included (but were not limited to) VIP hospitality at two EPL matches and visits to a selection of popular tourist attractions in the United Kingdom, a 7-night stay there for two, and a Hublot watch.

The FPL is played around the world and managers are required to make team selection decisions on a weekly basis in order to remain competitive globally. These decisions are complicated by a variety of constraints and must take into account both long-term and short-term rewards in terms of potential cumulative points gains. During the 2021/2022 FPL season, for instance, each manager was required to select a squad of fifteen players from a pool of 533 available players, while potentially substituting players into and out of their squads at multiple decision points which separate the *game weeks* (GWs) of the season. Effective team selection in this context is not a trivial task.

Each player in the pool of available players is assigned one of four playing positions, based on his real-world position, together with a monetary value specified in *Pound sterling* (£). The playing positions are *goalkeeper* (GK), *defender* (DEF), *midfielder* (MID), and *forward* (FWD). Each player featuring in the FPL is also associated with one of twenty (real-world) EPL teams (the player's real team when competing in the EPL). At the start of the season, each FPL manager is required to select a team of fifteen players, called a *squad* and consisting of specified numbers of GKs, DEFs, MIDs, and FWDs, with the total monetary value of the selected squad not exceeding £100 million. No more than three players from any single real-world EPL team may be selected for inclusion in the (FPL) squad during any GW. An example of such a squad selection is provided in Table 1.

A physical EPL match is played during each GW of the FPL season. FPL managers are free to buy and sell players between GWs, where players sold exit the squad (freeing up their cost as available capital) and players bought enter the squad (incurring a cost from the available capital). Managers are, however, granted only a specific number of free transfers per GW (where a transfer refers to one player exiting and another player entering the squad). Penalty points are deducted from the manager's total score, per transfer, for any additional transfers performed (over and above the available free transfers). Free

[1]While soccer is known as *football* in the United Kingdom, we refer to the sport here as *soccer* so as distinguish between it and the American ball game also called *football*.

Name	Position	Team	TP	GP	F	Fix	C
de Gea	GK	MUN	26	2	5.2	EVE(H)	5.0
Sa	GK	WOL	28	14	5.5	NEW(H)	5.0
Thiago Silva	DEF	CHE	24	2	6.0	SOU(H)	5.4
Rudiger	DEF	CHE	34	1	5.5	SOU(H)	5.6
Keane	DEF	EVE	27	8	6.0	MUN(A)	5.0
Cancelo	DEF	MCI	44	12	9.0	LIV(A)	6.1
Dia	DEF	MCI	39	5	6.5	LIV(A)	6.1
Saka	MID	ARS	26	13	5.8	BHA(A)	6.2
Gallagher	MID	CRY	28	0	6.2	LEI(H)	5.7
Douccoure	MID	EVE	38	11	6.5	MUN(A)	5.6
Sarr	MID	WAT	39	9	7.0	LEE(A)	6.3
Townsend	MID	EVE	33	10	6.8	MUN(A)	5.5
Vardy	FWD	LEI	40	11	8.2	CRY(A)	10.4
Saint-Maximin	FWD	NEW	35	8	6.8	WOL(A)	6.8
Ronaldo	FWD	MUN	21	2	7.0	EVE(H)	12.7

Table 1: An example of an FPL squad, showing player surname, player position, club (Team), total points (TPs), GW points (GPs), form (F) on a scale of 1–10, upcoming fixture (Fix), and cost (C) in millions of Pounds sterling.

transfers can be accumulated up to a maximum of two if not utilised during previous GWs. The anticipated performance of a player depends on a combination between his *total points* (TPs) earned thus far during the season according to an intricate FPL scoring system (described in Appendix A), his *GW points* (GPs) earned according to the same scoring system, his current *form* (F) and the fixture in which he will be competing (the opposing team, as well as whether the match will be on *home* (H) ground or *away* (A)). The *cost* (C) of a player is typically correlated with his real or anticipated performance.

The chief challenge associated with participating in the FPL is twofold: First, it is difficult to forecast or estimate the performances of all the FPL players in advance — not only because there are many players, but also because this performance can vary substantially over the course of a single FPL season. Secondly, the number of possible FPL squad combinations at any point during the season is so large that it is almost impossible to resolve squad inclusion decisions based solely on intuition. In this paper we show how a closing-window combinatorial optimisation modelling approach may be adopted towards effectively participating in the FPL, which takes as input future player performances forecast over the remainder of the FPL season — either statistically or by invoking appropriate machine learning techniques. We demonstrate retrospectively that our approach would have placed within the top 4% of players worldwide during the 2020/2021 FPL season.

The remainder of the paper is organised as follows. Upon having conducted a brief review in §2 of the literature related to FPL modelling and player performance forecasting, we review the various forecasting techniques employed in our modelling approach in §3. An integer programming model is formulated in §4 for recommending high-quality FPL squad selections for each GW of an FPL season, after which we present a case study in §5 in which the relative performances of our forecasting methods and the working of the optimisation model are evaluated in the context of the 2020/2021 FPL season. The paper finally closes, in §6, with some concluding remarks.

2 Related literature on fantasy sports

The literature on decision support modelling for fantasy sports is large. Two main areas of focus in this literature are *Fantasy American Football* (FAF) and FPL soccer. In this section, we briefly review these two main areas of the literature after having noted the origin of the notion of fantasy sports.

2.1 The origin and popularity of fantasy sports

The humble origins of fantasy sports can be traced back to a restaurant in Manhattan, New York called *La Rotisserie Française* in 1980 [27]. A publishing consultant for the magazine *Texas Monthly* conceived of the idea for a game now called *Fantasy baseball* while on a flight. Afterwards, he described the rules of the game to colleagues and friends over lunch at La Rotisserie Française. While not everyone was gripped by the idea, ten people initially decided to play the game, and so the first fantasy sport was born, called *Rotisserie baseball league* — named after the restaurant. The number of players grew from ten in 1980 to more than a million ten years later.

This rise in the number of players has led to the establishment of many other fantasy sports. By 1990 there were participants in fantasy football, fantasy baseball, fantasy basketball, fantasy hockey and fantasy soccer. The number of fantasy sports genres has grown considerably in recent years, and in 2022 there were 62.5 million fantasy sports players worldwide in a global fantasy sports market worth US$ 22.78 billion [22]. A break-down of the percentages of players per fantasy sport genre in this market during 2022 may be found in Table 2.

Fantasy sports genre	Governing body	%
Football	National Football League	54
Baseball	World Baseball Softball Confederation	25
Basketball	National Basketball Association	22
Football	National Collegiate Athletic Association	18
Basketball	National Collegiate Athletic Association	14
Football	Canadian Football League	14
Stock car racing	National Association for Stock Car Auto Racing	14
Football	United States Football League	14
Hockey	National Hockey League	13
Soccer	Fantasy Premier League	12
Various	Fantasy eSports	12
Mixed martial arts	International Mixed Martial Arts Federation	11
Golf	United States Golf Association	11
Football	Bachelor Fantasy League	6

Table 2: A break-down of global fantasy sports players per type of fantasy sport and governing body in 2022 [22]. The percentages do not add up to 100, because players typically participate in multiple fantasy sports genres.

2.2 Forecasting and model prediction in FAF

A large body of research has been devoted to the analysis of FAF data — the fantasy sports genre with by far the largest data analytic literature. The main types of analysis

available in this body of research pertain to

1. the psychology of, biases inherent in, and skills underlying fantasy football participation [17],
2. big data and how it influences participants of fantasy football [45],
3. how fantasy football affects public consumption of real *National Football League* (NFL) events [15, 43],
4. analyses of fantasy sport consumer segmentation [16],
5. the establishment of accurate empirical probability distributions of match events of different types, and
6. the generation of fantasy point projections or other forms of player performance projection forward in time, based on past data.

Instead of attempting to cover this entire literature or even a representative part of it, we devote this section to brief descriptions of three exemplars of relatively recent publications pertaining to the last point above. The examples cited were chosen to highlight work closest in spirit to our pursuit in the current paper.

In 2015, Lutz [38] applied support vector regression (equipped with a linear kernel) and artificial neural networks to analyse FAF data with a view to perform points prediction. Mean square errors were compared when no specific features were selected, when such features were selected manually, and when iterative, systematic feature elimination was applied in conjunction with cross validation. The support vector regression performed best when iterative, systematic feature elimination was applied, yielding a mean square error of approximately six points. This result was not very satisfactory, however, since many NFL games end in a points difference close to six points. The author claimed, in conclusion, that there was considerable room for improvement in his work as a result of the very limited number of features utilised, recommending that smarter feature selection methods perhaps be used in future.

Two years later, Landers and Duperrouzel [35] employed least squares and averaged neural networks, as well as boosted decision trees, to predict the number of points scored by specific players in a single FAF round. The authors found that boosted decision tree regression (equipped with player filtering[2]), along with a brute-force team selection algorithmic approach, was able to outperform the average performances of randomly selected FAF players' teams.

In 2018, King [31] set out to predict the number of fantasy points scored by quarterback players in the American Football League. The author employed a variety of algorithms for this purpose, including support vector regression, regression trees and artificial neural networks. It was found that the method of support vector regression was able to outperform the other algorithms, achieving a root mean squared error of 4.36% compared to errors of 8.53% and 8.77% achieved by regression trees and artificial neural networks, respectively. Moreover, it was found that the support vector regression implementation outperformed a similar prediction model used by CBS Sport, which achieved a root mean squared error

[2]The process of removing players with high performance variances and large maximum point scores.

of 7.32%. The final takeaway was that the fine-tuning of the hyper-parameters of all four prediction models would result in improved prediction accuracy with respect to their 2018 baselines.

2.3 Previous attempts at FPL forecasting and model prediction

The data analytic literature on the FPL is considerably smaller than that associated with FAF, and is mainly concerned with

1. the psychology of, biases inherent in, and skills underlying FPL participation [20, 44],
2. the mental health and behaviour of FPL participants [59, 60],
3. social media and wisdom-of-the-crowd effects on the FPL [3],
4. the establishment of accurate empirical probability distributions of match events of different types [57], and
5. the generation of fantasy point projections or other forms of player performance projection forward in time, based on past data.

Again we describe very briefly a few examples of relatively recent work conforming to the last point above in a vein similar to our current paper, so as to highlight how our work in this paper differs from previous work, and how it is, in fact, a natural extension to some of this work.

In 2018, Thapaliya [57] applied a Gaussian Naive Bayes algorithm in an attempt to predict whether or not a player will score at least eight points during a given GW. The algorithm was able to achieve an accuracy of 86% during the first GW going forward.

During the same year, Dykman [18] used various machine learning and statistical forecasting algorithms to predict FPL player performance one GW ahead into the future. These point forecasts were then taken as input to a mixed integer programming model aimed at recommending an FPL squad selection for the next GW. The objective in the latter model was to maximise the combined performance of the squad subject to budgetary, player transfer and other FPL rule constraints. The modelling approach was applied retrospectively to the 2017/2018 FPL season during a validation bid. It was found that the approach would have placed Dykman in the top 7.53% of FPL participants globally during the 2017/2018 season. A significant disadvantage of Dykman's modelling approach, however, was that he only adopted a one-week look-ahead period, which may yield significantly suboptimal results during subsequent future GWs in terms of when player performances may peak and as a result of transfer penalties incurred.

Also in 2018, Kristiansen *et al.* [33] proposed a mathematical model for FPL player selection which takes as input forecasts of player points. Three methods were employed to generate forecasts. The first centred on the most recent average points accumulated by each FPL player, the second was based on regression involving multiple explanatory variables, and the third utilised bookmakers' odds to predict points. The model was solved by applying a rolling-horizon heuristic and its efficacy was demonstrated by solving the model retrospectively in the context of the first 35 GWs of the 2017/2018 FPL season. The results were compared with the performance of actual FPL participants during that

season and it was found that the model was able to achieve a position among the top 30% of all FPL participants consistently.

The following year, Khamsan and Maskat [30] considered the problem of how to mitigate highly imbalanced data in terms of output class labels when machine learning techniques are used to predict virtual FPL player price changes over the course of a given time period.

In 2021, Pukdee [50] examined the use of network analysis when analysing soccer data, focusing on the EPL. Network analytic techniques were used to rank players and to derive the nature of the interaction between players based on publicly available data only. The results of the study were comparable with those utilised by the media, but required much less data.

In 2022, Bangdiwala *et al.* [2] compared the efficacy of three statistical and machine learning models (linear regression, decision trees and random forests) in terms of predicting the number of points that each player would earn over the course of an FPL season. The learning was based on features such as fixture difficulty, the forms of the two opposing teams relative to one another, the creativity of the player and the threat posed by the player. It was found that the modelling approach aided players of this game to make more informed FPL squad selections.

Rajesh *et al.* [51] proposed a player recommendation system in 2022 aimed at enabling an "average interested person" to make informed decisions related to player selection based on the application of data science techniques and analytics, graphical visualisations, and a variety of statistical measures. The objective was to help FPL participants resolve the so-called favoritism bias — where participants tend to select players from their favorite EPL teams — by generating actionable insights from data. The working of the system was demonstrated in the context of the 2021/2022 FPL season.

Also in 2022, Maniezzo and Aspee Encina [40] performed a business analytics experiment in which predictive and prescriptive analytics were used to provide real-time bidding support in the context of fantasy soccer draft auctions. Forecasting methods were invoked to quantify the expected return of each investment alternative, upon which adaptive online recommendations on playing squad selections were generated by means of sub-gradient optimisation. A distributed front-end implementation was also established in order to demonstrate the viability of the modelling approach.

3 Forecasting player performance

We applied a wide variety of time series forecasting methods to the past performance data of players participating in the 2020/2021 EPL season in order to predict their anticipated future FPL performances. These methods belong to different classes of forecasting methods and are described very briefly in this section in pursuit of self-containment of the paper. The forecasting methods described in this section were also ensembled in the hope of achieving superior forecasting performance, and so various ensembling methodological approaches are also described very briefly towards the end of the section.

3.1 Baseline forecasting methods

In order to provide a basis for comparing the forecasting performances of more sophisticated models, we applied certain simple forecasting methods as benchmark (or baseline) models. These methods are straightforward, but might perform relatively well in certain contexts.

Given a time series $\boldsymbol{x} = (x_1, \ldots, x_T)$ of past player performance scores over a period of T GWs, the *average* (or historical mean) method dictates that the prediction of h values of the time series into the future is taken as the mean of all previous observations in the series, and is given by

$$\hat{x}_{T+i} = \frac{x_1 + \cdots + x_T}{T}, \qquad i = 1, \ldots, h.$$

A *naive forecast*, on the other hand, involves setting all the forecast values equal to the last historical observation of the time series, such that

$$\hat{x}_{T+i} = x_T, \qquad i = 1, \ldots, h.$$

A naive forecast is typically only adopted as a baseline model for data emanating from a stationary process in which future values are highly unpredictable [25]. The *seasonal naive* method is a variation on this basic naive forecasting method in which the predicted values are taken as

$$\hat{x}_{T+i} = x_{T+i-m(k+1)}, \qquad i = 1, \ldots, h.$$

Here, m denotes the seasonal period and k denotes the integer part of the fraction $(h-1)/m$ (*i.e.* the number of complete seasons prior to time $T + h$). According to the seasonal naive method, the predicted value for a given period is therefore simply the corresponding observed value of the previous season [64].

An alternative baseline model is based on the *drift* method which entails extrapolating a straight line in order to forecast future values. This line is drawn through the first and last observations on a time plot of the series. Forecast values are therefore given by

$$\hat{x}_{T+h} = x_T + \frac{h}{T-1}\sum_{t=2}^{T}(x_t - x_{t-1}) = x_T + h\frac{x_T - x_1}{T-1},$$

with the values of $\hat{x}_{T+1}, \hat{x}_{T+2}, \ldots, \hat{x}_{T+h-1}$ being read off from this straight line.

3.2 Moving averages

The basic assumption underlying the use of *moving averages* (MAs) is that time series observations which are in close temporal proximity are likely to exhibit similar values [39]. MAs are typically either *simple*, *weighted*, or *exponential* in nature [64]. A simple MA of odd order d (d-MA) involves estimating the trend-cycle component of a performance score time series at time $T + 1$ as

$$\hat{x}_{T+i} = \frac{1}{d}\sum_{j=-k}^{k} x_{T+i+j}, \qquad i = 1, \ldots, h,$$

where $d = 2k + 1$ and $\hat{x}_t$ denotes the mean of the observations of the time series over d time periods. This is known as a symmetric MA (of odd order, *e.g.* 3, 5 or 7) of the historical values and removes some of the noise in the time series data, thereby producing a smooth trend-cycle component [64]. The order of the MA determines the smoothness of the estimate. Although a smoother trend-cycle estimate is produced by a larger-order MA, such an MA may result in underfitting the data [25, 39].

It is also possible to calculate a simple MA of even order, but such an MA would not be symmetric. Another MA is therefore typically applied to the result of the even-order MA so as to achieve symmetry (also known as a *centred* MA or *double* MA) [64]. A 2-MA might, for example, be applied to a 4-MA and the result denoted a 2×4-MA. An odd-order MA may, similarly, be followed by applying another odd-order MA [64]. Such combinations produce *weighted* MAs.

A simple MA is a special case of a weighted MA in which all weights equal $1/d$. A weighted d-MA may be expressed as

$$\hat{x}_{T+i} = \sum_{j=-k}^{k} w_j x_{T+i+j}, \qquad i = 1, \ldots, h,$$

where k is defined as above and w_j denotes the weight of the j^{th} observation. This set of weights is referred to as the *weight function*, and the weights should sum to one and be symmetric (*i.e.* $w_j = w_{-j}$) [25]. Weighted MAs produce smoother trend-cycle estimates, which sometimes holds a significant advantage.

3.3 Exponential smoothing

Exponential smoothing methods were first proposed by Brown [6], Holt [24] and Winters [62], and involve the use of weighted averages of historical observations, with larger weights being assigned to more recent observations as the weights of older observations decay exponentially [64].

Simple exponential smoothing entails assigning weights that decrease exponentially with an increase in the age of observations, so that

$$\hat{x}_{T+i} = \alpha \left(x_T + (1-\alpha) x_{T-i} + \cdots + (1-\alpha)^{T-i} x_1 \right) + (1-\alpha)^T x_0, \qquad i = 1, \ldots, h, \quad (1)$$

where $\alpha \in [0, 1]$ is a smoothing parameter [25, 42]. The parameter α determines the rate at which the weights of the older observations decrease [64]. The forecast for time $T + 1$ is a weighted average of all the training observations $x_t, \ldots, x_T$. As T increases, $(1 - \alpha)^T$ becomes increasingly smaller and the effect of x_0 on $\hat{x}_T$ becomes negligible [42].

Simple exponential smoothing is well-suited to time series data exhibiting no clear trend or seasonal component. The expression in (1) may be derived by invoking different forms of exponential smoothing, namely the component form or the weighted average form [64]. The component form of exponential smoothing potentially comprises three components, namely a *level* component, denoted by ℓ_t, a *trend* component, denoted by b_t, and a *seasonal* component, denoted by s_t. An additional component expression is added to the forecast expression for each of the components which form part of the forecasting method.

Simple exponential smoothing, which only has a level component, may be expressed mathematically as

$$\begin{aligned} \text{Forecast expression:} \quad & \hat{x}_{t+i} = \ell_t, \text{ and} \\ \text{Level expression:} \quad & \ell_t = \alpha x_t + (1-\alpha)\ell_{t-1}, \end{aligned}$$

for all $t = 1, \ldots, T$ and $i = 1, \ldots, h$, where ℓ_t denotes the smoothed value (level) of the observation at time t and $\alpha \in [0, 1]$ is the smoothing parameter for the level.

The *linear trend method* (also known as *double exponential smoothing*) was proposed by Holt [24] to account for the difficulty encountered when using simple exponential smoothing to forecast time series data which exhibit a trend component. The method is expressed mathematically as

$$\begin{aligned} \text{Forecast expression:} \quad & \hat{x}_{t+i} = \ell_t + ib_t, \\ \text{Level expression:} \quad & \ell_t = \alpha x_t + (1-\alpha)\left(\ell_{t-1} + b_{t-1}\right), \text{ and} \\ \text{Trend expression:} \quad & b_t = \beta\left(\ell_t - \ell_{t-1}\right) + (1-\beta)b_{t-1}, \end{aligned}$$

for all $t = 1, \ldots, T$ and $i = 1, \ldots, h$, where α and ℓ_t are defined as before, b_t denotes the estimated trend of the time series at time t and $\beta \in [0, 1]$ is a trend smoothing parameter.

Holt's linear trend method was extended to the well-known Holt-Winters method [24, 62] to arrive at a seasonal method (also known as *triple exponential smoothing*) by adding a component which accounts for seasonality in the time series. The Holt-Winters method may be expressed mathematically as

$$\begin{aligned} \text{Forecast expression:} \quad & \hat{x}_{t+i} = \ell_t + ib_t, \\ \text{Level expression:} \quad & \ell_t = \alpha x_t + (1-\alpha)\left(\ell_{t-1} + b_{t-1}\right), \\ \text{Trend expression:} \quad & b_t = \beta\left(\ell_t - \ell_{t-1}\right) + (1-\beta)b_{t-1}, \text{ and} \\ \text{Seasonal expression:} \quad & s_t = \gamma\left(x_t - \ell_{t-1} - b_{t-1}\right) + (1-\gamma)s_{t-m}, \end{aligned}$$

for all $t = 1, \ldots, T$ and $i = 1, \ldots, h$, where the frequency of seasonality is denoted by m and the integer part of $(h-1)/m$ is denoted by k. The initial component values and smoothing parameters first have to be estimated before an exponential smoothing method may be applied to time series data.

Exponential smoothing methods can also be described in terms of their so-called state space formulations [64]. *State space models* were introduced by Kalman [28] and comprise a so-called measurement equation, describing the time series observations, together with one or more state equations describing changes in the components of a time series [64]. In the case of exponential smoothing, the components of a time series are *error*, *trend*, and *seasonal*. A model instance takes the form ETS$(\cdot, \cdot, \cdot)$, where the arguments represent (error, trend, seasonal). Modelling options for the components are error $\in$ {additive, multiplicative}, trend $\in$ {none, additive, additive damped}, and seasonal $\in$ {none, additive, multiplicative}. The full exponential smoothing list in the ETS framework therefore comprises twenty four possible models [25]. The term ETS is adopted when referring to exponential smoothing models in the remainder of this paper.

3.4 Regression

A specified relationship is assumed between a forecast variable x and one or more predictor variables $y_1, \ldots, y_k$ in regression models tailored to time series forecasting. This relationship may either be linear or non-linear[3]. *Simple regression* involves one predictor variable (*i.e.*, $k = 1$), while *multiple regression* involves more than one variable (*i.e.*, $k > 1$). Simple linear regression may be performed by fitting the equation

$$x_t = \beta_0 + \beta_1 y_t + \varepsilon_t, \qquad t = 1, \ldots, T \tag{2}$$

to the original time series data, where β_0 and β_1 denote the intercept and slope of the resulting straight-line relationship, respectively, and ε_t denotes the deviation of observations from the straight line model [42]. Estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ of the coefficients β_0 and β_1 may be computed by invoking the least squares principle, according to which the sum of the squared errors $\sum_{t=1}^{T} \varepsilon_t^2$ is minimised [64].

In the case of *multiple linear regression*, the relationship in (2) becomes

$$x_t = \beta_0 + \beta_1 y_{1,t} + \beta_2 y_{2,t} + \cdots + \beta_k y_{k,t} + \varepsilon_t, \qquad t = 1, \ldots, T, \tag{3}$$

where the coefficients $\beta_0, \ldots, \beta_k$ provide a measure of the marginal contributions of the predictor variables $y_{1,t} \ldots, y_{k,t}$, respectively, when predicting the value of the forecast variable x_t [54]. Linear regression is, however, only appropriate in time series forecasting when linearly estimating the trend component of the series for the purpose of trend adjustment.

A standard approach towards non-linear regression involves either transforming the forecast variable, the predictor variables or both, before estimating the coefficients of a regression model [25]. A popular non-linear regression model involves performing a logarithmic transformation such that

$$\log x_t = \beta_0 + \beta_1 \log y_t + \varepsilon_t, \qquad t = 1, \ldots, T.$$

Theoretically, an infinite number of non-linear relationships may be modelled. Two examples are the popular quadratic relationship $\left(x_t = \beta_0 + \beta_1 y_t + \beta_2 y_t^2 + \varepsilon_t\right)$ and the cubic polynomial relationship $\left(x_t = \beta_0 + \beta_1 y_t + \beta_2 y_t^2 + \beta_3 y_t^3 + \varepsilon_t\right)$ [25, 39]. Naturally these relationships would apply to a non-linear trend component. Overfitting due to polynomial regression of higher degrees should be avoided [39].

Cubic splines are a natural extension of polynomial regression in which the time series is partitioned into a number of intervals and a polynomial regression function of degree three is fitted to each interval [54]. This technique provides a *piecewise* mapping of the predictor variables to the forecast variable, also known as *local regression* [25, 39].

The most popular implementation of local regression is locally weighted regression (often referred to as *Loess*), proposed by Cleveland and Devlin [11], which is more robust to outliers than local linear regression [64]. The well-known *seasonal and trend decomposition using Loess* procedure was proposed by Cleveland *et al.* [10] and is based on sequential applications of Loess.

[3]In this context, linearity refers to the relationship between the variables x and $y_1, \ldots, y_k$ as opposed to linearity in the parameters.

3.5 ARIMA models

ARIMA models produce predictions by describing autocorrelations in the data, as opposed to ETS models which model trend and seasonality in time series data. The thinking behind the ARIMA approach is that observations which follow one another commonly exhibit some form of dependence [42]. ARIMA models comprise an *autoregressive* (AR) component, an integration component, and an MA component.

The rationale behind AR models is that an observation x_t may be modelled as a linear function of p past observations $x_{t-1}, x_{t-2}, \ldots, x_{t-p}$, where p is called the *order* of the AR model [54]. This is comparable with the expression in (3) for multiple linear regression, although the forecast variable x is regressed against its history in the AR case [64]. An AR model of order p, denoted by AR(p), may be expressed as

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \varepsilon_t, \qquad t = p+1, \ldots, T, \tag{4}$$

where c denotes a constant and $\phi_1, \ldots, \phi_p$ are model parameters [7]. Univariate time series exhibiting no trend or seasonality are amenable to forecasting by AR models.

An MA model employs past forecast errors as predictor variables in a regression-like manner, rather than regressing the forecast variable against its history [25]. The standard notation MA(q) denotes an MA model of order q and is expressed as

$$x_t = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t, \qquad t = q+1, \ldots, T, \tag{5}$$

where the mean of the observations is denoted by μ and $\theta_1, \ldots, \theta_p$ are model parameters [7].

To form an ARMA(p, q) model, the models in (4) and (5) are combined, yielding

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t, \qquad t = 1 + \max\{p, q\}, \ldots, T.$$

ARMA(p, q) models are, however, only defined for stationary processes while, as mentioned before, real-world data (such as FPL player performance data) are typically not compatible with this assumption. In order to address this problem, an additional integration parameter of order d may be included [64]. The resulting model is the general ARIMA(p, d, q) model [25]. During the execution of ARIMA models, d differencing transformations are first applied to the data in order to generate a stationary time series, after which an ARIMA(p, q) model is applied. Integration, in this context, can be seen as reversing the differencing procedure [7].

The inclusion of differencing complicates the mathematical representation of ARIMA models and so back-shift operator notation is typically utilised to represent these models. The backshift operator of order d is defined as $B^d x_t = x_{t-d}$. By implementing this notation, an ARIMA(p, d, q) model may be written as

$$\underset{\text{AR}(p)}{\uparrow}\left(1 - \phi_1 B - \cdots - \phi_p B^p\right) \underset{d \text{ differences}}{\uparrow}(1 - B)^d x_t = c + \underset{\text{MA}(q)}{\uparrow}\left(1 + \theta_1 B + \cdots + \theta_q B^q\right) \varepsilon_t$$

for all $t = 1 + \max\{p, d, q\}, \ldots, T$, where the integration component is included to allow for the application of ARIMA models to non-stationary processes [25].

The notation ARIMA$(p, d, q)(P, D, Q)_m$ model denotes a so-called *seasonal ARIMA* (SARIMA) model, where m represents the number of observations per seasonal period [39]. In a SARIMA model, a linear combination of past seasonal observations and/or forecast errors is added to the forecast [64]. By utilising the standard backshift operator notation introduced above, an ARIMA$(1, 1, 1)(1, 1, 1)_{1,2}$ model may, for instance, be written as

$$\left(1 - \phi_1 B^1\right)\left(1 - \phi_1 B^4\right)\left(1 - B^1\right)\left(1 - B^4\right)(x_t - \mu) = \left(1 - \theta_1 B^1\right)\left(1 - \Theta_1 B^4\right)\varepsilon_t$$

for all $t = 1 + 1, \ldots, T$, where the new seasonal terms are simply multiplied by the non-seasonal terms and μ denotes the mean of all past observations [25].

3.6 Decision trees

Decision tree methods may be used to forecast the value of a target variable by inferring certain decision rules from a set of training data in a supervised machine learning paradigm, with each decision represented by a node. These nodes collectively form tree-like output data structures.

In order to apply decision trees to time series forecasting, the data have to be transformed into subsets of observations $(\boldsymbol{y}_1, \boldsymbol{z}_1), (\boldsymbol{y}_2, \boldsymbol{z}_2), (\boldsymbol{y}_3, \boldsymbol{z}_3), \ldots$ where, for any integer value of i, $\boldsymbol{y}_i$ is a vector, called a *feature vector*, and contains predictor values influencing the value(s) to be forecast and where $\boldsymbol{z}_i$ is also a vector, called a *target vector*, and contains the corresponding actual forecast values(s). Each such pair of vectors represents a training sample. In the case where a time series $x_1, x_2, \ldots, x_T$ has been observed and has to be forecast h points into the future, each successive feature vector $\boldsymbol{y}_i$ contains $p < T - h$ time series observations, taken over a single-step advancing rolling horizon of length T [7]. That is, $\boldsymbol{y}_i = [x_i, \ldots, x_{i+p-1}]$. The corresponding target vector $\boldsymbol{z}_i$ contains the following h data points of the series. That is, $\boldsymbol{z}_i = [x_{i+p}, \ldots, x_{i+p+h-1}]$. There are $T - p - h + 1$ observations in the training set in total.

The data transformation described above essentially yields a standard multiple regression problem and is illustrated graphically in Figure 1. In this example, the partial time series $x_1, x_2, \ldots, x_{11}$ (with vertices denoting data points) is partitioned into four training samples, each with $p = 5$ observations as an input, upon which the following $h = 3$ observations are taken as the target vector. The training set $([x_1, \ldots, x_5], [x_6, x_7, x_8]), ([x_2, \ldots, x_6], [x_7, x_8, x_9]), ([x_3, \ldots, x_7], [x_8, x_9, x_{10}]), ([x_4, \ldots, x_8], [x_9, x_{10}, x_{11}])$ is therefore constructed iteratively. Including multiple observations of a time series in the target feature $\boldsymbol{z}_i$ facilitates the prediction of multiple steps into the future in a natural way. This type of transformation is called *reduction*, since the task of forecasting is reduced to the simpler task of tabular regression [37].

The decision tree constructed when solving the above regression problem contains three types of nodes. The *root node* (or initial node) is representative of the entire training set. The features exploited when traversing the tree and performing predictions are represented by the *interior nodes* (with each branching of a parent node into children inducing a decision rule). Finally, the *leaf nodes* represent the final predicted regression values.

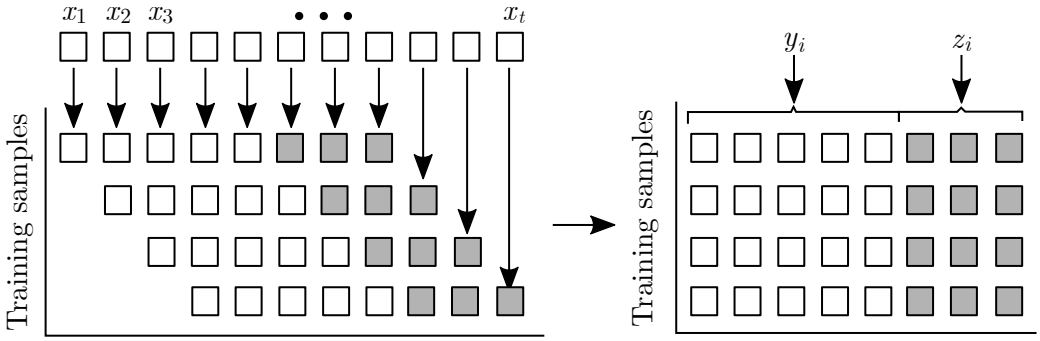

Figure 1: A time series data transformation facilitating machine learning.

The decision tree is traversed from the root node until a leaf node is reached in order to obtain a single prediction for an unseen data instance. The final prediction is the mean dependent variable value in the leaf node. The decision rules of a tree are deduced from the principle of *standard deviation reduction* in the case of regression problems.

The depth of the resulting tree affects its *goodness of fit* (GOF) and the number of iterations performed in order to obtain the final tree. A decision tree may be prone to over-fitting, despite having tuned these hyper-parameters appropriately. Overfitting may be mitigated by employing *random forest* models [5], which are combinations of decision trees.

3.7 The method of k-nearest neighbours

When training a *k-Nearest Neigbours* (kNN) algorithm, the data are sorted according to some distance measure, based on their features. In order to classify a new datum x_0, a set of k observations which are closest to this new data point according to the distance measure adopted are identified within the training data set and are collectively denoted by $\mathcal{N}_0(k)$.

A weighted average of the observations in $\mathcal{N}_0(k)$, weighted proportionally to the inverse of their distance from x_0, is assigned to x_0 [36]. Although many other alternatives are available, a popular choice for the distance metric is the Euclidean distance. Distances are first calculated between the point to be regressed and the labelled points, after which the labelled points are ordered according to increasing distance [36]. Next, the inverse distance-weighted average over the k nearest multivariate neighbours is calculated [36].

The hyper-parameters of the kNN algorithm are the distance measure adopted and the number k of neighbours considered during each iteration. The value of k controls the *bias-variance trade-off* in the resulting model and may be determined empirically based on the *root-mean-square deviation* (RMSD) and using cross-validation [36].

3.8 Croston's method

Croston's method is based on the notion of exponential smoothing [25] and was proposed for use in the context of intermittent time series forecasting. Two new series are constructed on which forecasts are based according to Croston's method — a positive time

series, called the *z-series*, and another series capturing the intermittent time between these observations, called the *a-series*. Denote the original time series by $x_1, \ldots, x_T$, let z_i denote the i^{th} non-zero observation and let a_i denote the time that has elapsed between observations z_{i-1} and z_i. When $x_i = 0$, exponential smoothing is applied to each series, yielding

$$\hat{z}_i = \hat{z}_{i-1} + \alpha_z \left(z_{i-1} - \hat{z}_{i-1}\right)$$

and

$$\hat{a}_i = \hat{a}_{i-1} + \alpha_a \left(a_{i-1} - \hat{a}_{i-1}\right)$$

for all $i = 1, \ldots, j$, where j denotes the last time period during which a positive value was observed, while $\alpha_z, \alpha_a \in [0, 1]$ are smoothing parameters. During periods when $x_i > 0$, however, it is assumed that $\hat{z}_i = \hat{z}_{i-1}$ and $\hat{a}_i = \hat{a}_{i-1}$. Future forecasts based on Croston's method are subsequently calculated as

$$\hat{x}_{T+q} = \frac{\hat{z}_{j+1}}{\hat{a}_{j+1}}$$

for all $q = 1, \ldots, h$, where j again denotes the last time period during which a positive value was observed in the original series. The parameters α_z and α_a are commonly assigned the same value, such as $\alpha_z = \alpha_a = 0.1$ (as Croston [12] originally proposed). Different values for these parameters, with ranges[4] $\alpha_z, \alpha_a \in [0.1, 0.3]$, have been been experimented with by Teunter *et al.* [56], among others.

3.9 Model ensembling

Ensemble methods are based on the notion of crowd wisdom, and may be applied to regression problems by combining the predictions of several models [41, 55]. By providing more accurate and robust results than those of the underlying individual models, ensembling has been shown to be capable of increasing the quality of predictions [14, 46]. Two phases make up the ensembling procedure, namely the building of models (also known as *ensemble generation*) and the subsequent combination of these models (known as *ensemble integration*) [58]. *Ensemble pruning* is an additional phase which may form part of an intermediate step aimed at reducing the ensemble size (*i.e.* the number of models included in the ensemble), thereby potentially improving the quality of the forecast obtained [52]. The general procedure of ensembling is illustrated graphically in Figure 2, where the training data $\boldsymbol{x}$, are used to train n different models. In order to obtain the prediction $\hat{y}$, the model outputs are combined in some pre-specified manner.

The *diversity* of the models contained in the ensemble determines the performance of the ensemble [49]. An ensemble may consist either of homogeneous or heterogeneous models. *Homogeneous models* are different instantiations of the same learning model which have been created by adjusting hyper-parameter values, training records and/or input values. Popular methods for creating homogeneous models are *bagging* and *boosting*. Different models which are fitted on the same training data, on the other hand, are called *heterogeneous models*. Heterogeneous models differ in their underlying assumptions and

[4]The utilisation of values for α_z and α_a within specified ranges is sometimes referred to as the *optimised Croston method* [64].

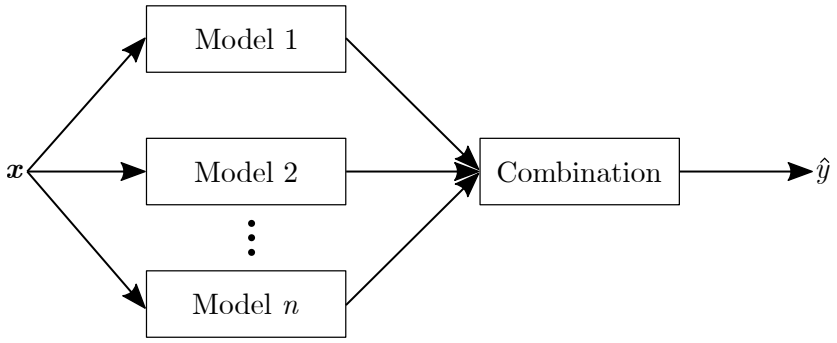

Figure 2: The common procedure of model ensembling (adapted from [47]).

methods of time series prediction [58]. A *model-centred* or *data-centred* approach may be followed to achieve diversity among ensemble members in the case of regression problems. A model-centred approach involves fitting different algorithms to the same data so as to produce different (heterogeneous) models [1]. In a data-centred approach, on the other hand, the time series provided as input to a forecasting model is altered in different ways in order to produce different (homogeneous) models [64].

Breiman [4] introduced the ensembling method of bagging, also known as *bootstrap aggregation*, in which various instantiations of the same forecasting model are generated and their results combined to produce an aggregated forecast. By forming bootstrapped learning sets from the original training data in which a random selection of observations from the learning sets allow for repetition, multiple model instantiations of the same forecasting model can be created [49].

In the case of time series regression problems, independent records may be drawn randomly from the training data. In order for the characteristics of the bootstrapped data to resemble those of the original data, however, the non-stationarity and autocorrelation properties of time series data have to be accounted for [48]. A *block bootstrap* may be employed for this purpose, where a decomposition of the original time series is required to obtain a so-called pattern and remainder component [25]. Neighbouring sections of the remainder component are then randomly selected and joined together [64].

Circular block bootstrapping, *linear process bootstrapping* and *moving block bootstrapping* are all different approaches towards block bootstrapping [48]. In moving block bootstrapping, a series of T observations is provided as input and blocks of length ℓ are specified, in which case there are a total of $T - \ell + 1$ different blocks. It is illustrated graphically in Figure 3 how an original series of 18 observations may be used to generate 15 blocks of length 4 (as illustrated by the solid arrows). The origin of the bootstrapped series is randomly selected within the first block, and $(n/\ell) + 2$ blocks are randomly chosen (as illustrated by the dotted arrows) and combined.

A bootstrapped version of the original time series is formed by adding the bootstrapped remainder series to the pattern component. The different learning sets considered during the application of bagging are then formed by generating multiple bootstrapped series. A model is fitted to each of the learning sets and a simple mean of the prediction values is taken as a combination of the individual model outputs [4].

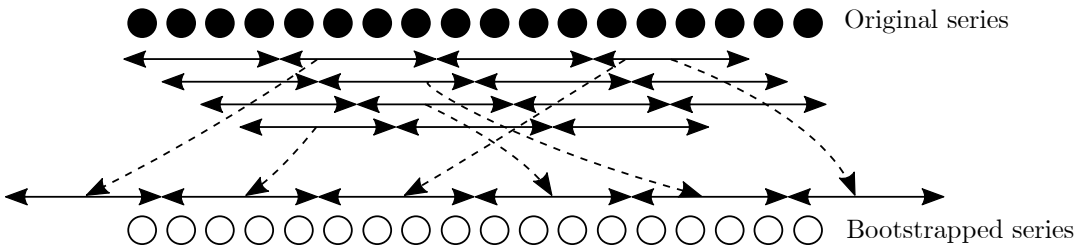

Figure 3: Moving block bootstrapping of a time series (adapted from [48]).

Schapire [53] proposed an alternative ensembling method in 1990, known as boosting, in which the performance of weak models are 'boosted' in order to achieve better performance. As in the case of bagging, inputs to the same model are slightly adjusted in an attempt to improve forecasting accuracy. Boosting requires time series data to be transformed into a standard supervised learning format where each training instance is represented by an input vector $\boldsymbol{x}$ and a corresponding target variable y. A sliding window approach is employed to achieve this [37].

In contrast to bagging, boosting is a sequential and dependent process. Each training instance has a weight associated with it at the start of the boosting procedure. During each subsequent boosting iteration, these weights are adjusted based on how accurately the forecasting model has predicted future values. The motivation for this approach is that the ensemble should be focused on instances for which the forecasting model performs poorly. Increasing the weights of these instances should theoretically result in an ensemble model which better accounts for any bias in these instances during the ensuing boosting iteration [64]. The final estimate of the boosting ensemble is calculated by taking a weighted combination of the results obtained from base learners. Initial base learners are usually outperformed by these types of ensembles, since the new model is aimed at improving the weaknesses of the previous model during each iteration [1].

The popular *XGBoost* (extreme gradient boosting) [8], and *LightGBM* (light gradient boosting machine) [29] algorithms, which are tree-based models, are based on Schapire's [53] original boosting procedure. LightGBM and XGBoost both employ gradient-boosted decision trees. Boosting procedures typically outperform bagging procedures when there is less noise in the training data, although boosted ensembles are prone to overfit data which exhibit significant noise [1].

In order to reduce the size of an ensemble (*i.e.* the number of models included in the ensemble), ensemble pruning may be applied before combining the models [58]. The ultimate goal of this process is to select the best subset of models from the set of base learners initially considered [47].

The efficiency and forecast accuracy of an ensemble may be improved as a result of ensemble pruning. More computational power is naturally required when an ensemble includes a large number of models. The predictive performance of an ensemble depends on the models included in the ensemble. Poorly performing models included in an ensemble may have a degrading effect on the performance of the ensemble. Model diversity is also reduced when ensembles include similar models. Pruning poorly performing models by excluding them from the ensemble, while model diversity is also ensured, is therefore an effective technique for model ensembling [58].

The following categorisation of pruning methods has been proposed by Tsounakas *et al.* [58]:

1. *Ranking-based* methods, where some evaluation function is used to rank models, and selection occurs based on the resulting ranking,
2. *clustering-based* methods, where models are first grouped together by some clustering algorithm, which provide similar prediction results, after which ensemble diversity is increased by pruning each identified cluster individually,
3. *optimisation-based* methods, and
4. other methods which do not fall in any of the above categories.

4 FPL squad recommendation model

This section is devoted to the derivation of a mixed-binary programming model for FPL squad selection. The model is designed to generate a recommended squad composition for any GW of the FPL season. It takes as unalterable historical input data the FPL squad compositions of those GWs of the FPL season that have already elapsed (if any), as well as performance score forecasts of FPL squad candidates for the remaining GWs of the season and the attributes of the EPL players available for selection (such as their costs, their team roles, and their EPL teams). The model then generates a recommendation as to the player composition of the FPL squad during the GW in question, together with tentative recommendations as to FPL squad compositions for all the GWs of the season remaining after that. Application of the model therefore corresponds with a closing-window strategy in the sense it is solved for each successive GW of the FPL season, upon which the historical FPL squad compositions are recorded and an updated version of the tentative future squad compositions for the remainder of the season is computed.

4.1 Sets employed in the model

Let $\mathcal{T} = \{1, \ldots, \Lambda\}$ index the GWs of the FPL season over which squad selection is planned and let $\Omega \in \mathcal{T}$ index the current GW for which the squad composition has to be resolved, taking into account tentatively favourable future squad compositions over the remaining GWs in the set $\mathcal{T}_\Omega = \{\Omega, \ldots, \Lambda\}$. Define the set

$$\mathcal{T}'_\Omega = \begin{cases} \emptyset & \text{if } \Omega = 1, \\ \{1, \ldots, \Omega - 1\} & \text{if } \Omega > 1 \end{cases}$$

of past GWs. Note, therefore, that $\mathcal{T}'_\Omega \cup \mathcal{T}_\Omega = \mathcal{T}$. Also, let $\mathcal{C}_\Omega = \{1, \ldots, C_\Omega\}$ index the set of EPL players available for inclusion in the FPL squad during GW Ω. Each of these players fulfils exactly one role in the squad, indexed by the set $\mathcal{D} = \{1, 2, 3, 4\}$, with 1 denoting *goalkeeper*, 2 denoting *defender*, 3 denoting *midfielder*, and 4 denoting *forward*. These roles are not considered to be time-dependent, because they represent particular skill sets of players which are not easily changed without extensive training. Moreover, let $\mathcal{S}_\Omega = \{1, \ldots, S_n^{(\Omega)}\}$ index the collection of EPL teams, each containing a subset of players

in $\mathcal{C}_\Omega$. Recall that no more than three players in EPL team $s \in \mathcal{S}_\Omega$ may be present in the FPL squad during GW $t \in \mathcal{T}_\Omega$.

4.2 Model parameters

Denote the projected performance score of player $c \in \mathcal{C}_\Omega$ during GW $t \in \mathcal{T}_\Omega$ by $p_{c,t}$ and let $k_{c,t}$ be the cost of including player $c \in \mathcal{C}_\Omega$ in the FPL squad during GW $t \in \mathcal{T}_\Omega$. Also, let r_d be the required number of players fulfilling role $d \in \mathcal{D}$ in the FPL squad.

Furthermore, let B be the budget available for player inclusion in the FPL squad at any decision point during the season. The FPL requires that the total cost of players included in the team at any one time should never exceed B. Whenever a player is included in the team, the current cost associated with that player is incurred, while if a player leaves the team, the current cost of that player becomes available again for use to procure replacements. Also, denote the amount of money that remains unspent at decision point $t \in \mathcal{T}$ by b_t.

Suppose that all past (unalterable) squad inclusion decisions are captured by the parameters

$$x_{c,t} = \begin{cases} 1 & \text{if player } c \in \mathcal{C}_\Omega \text{ was included in the FPL squad during GW } t \in \mathcal{T}'_\Omega, \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{c,t} = \begin{cases} 1 & \text{if candidate } c \in \mathcal{C}_\Omega \text{ was brought into the FPL squad for GW } t \in \mathcal{T}'_\Omega, \\ 0 & \text{otherwise} \end{cases}$$

and

$$z_{c,t} = \begin{cases} 1 & \text{if player } c \in \mathcal{C}_\Omega \text{ was removed from the FPL squad for GW } t \in \mathcal{T}'_\Omega, \\ 0 & \text{otherwise.} \end{cases}$$

Also, define the binary parameters

$$\alpha_{c,d} = \begin{cases} 1 & \text{if player } c \in \mathcal{C}_\Omega \text{ fulfils role } d \in \mathcal{D} \text{ in the FPL squad,} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\beta_{c,s} = \begin{cases} 1 & \text{if player } c \in \mathcal{C}_\Omega \text{ is a member of EPL team } s \in \mathcal{S}_\Omega, \\ 0 & \text{otherwise.} \end{cases}$$

It is a working assumption that player $c \in \mathcal{C}_\Omega$ remains in EPL team $s \in \mathcal{S}_\Omega$ for all GWs $t \in \mathcal{T}_\Omega$. Moreover, denote the (actual past or projected) performance score of player $c \in \mathcal{C}_\Omega$ during GW $t \in \mathcal{T}$ by $p_{c,t}$ and let $k_{c,t}$ be the (actual past or projected) cost of including player $c \in \mathcal{C}_\Omega$ in the FPL team during GW $t \in \mathcal{T}$.

The nested relationship $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \mathcal{C}_3 \cdots \subseteq \mathcal{C}_\Lambda$ between the EPL teams is assumed. That is, members can only be added to an EPL team from any GW to the next (no players are removed during any GW). If a player c only becomes available during some GW $t' \in \mathcal{T}_\Omega$, then the performance score and cost of that player is merely set to $p_{c,t} = 0$ and $k_{c,t} = M$, respectively, where M is a large positive value, for all $t = 1, \ldots, t' - 1$. Similarly, if a player c were no longer to be available for FPL squad inclusion from some GW $t^* \in \mathcal{T}$ onwards, then the performance score and cost of that player is merely set to $p_{c,t} = 0$ and $k_{c,t} = M$, respectively, for all $t \in \{t^*, \ldots, \Lambda\}$.

4.3 Model variables

Define the binary decision variables

$$x_{c,t} = \begin{cases} 1 & \text{if player } c \in \mathcal{C}_\Omega \text{ is included in the FPL squad during GW } t \in \mathcal{T}_\Omega, \\ 0 & \text{otherwise,} \end{cases}$$

as well as the binary auxiliary variables

$$y_{c,t} = \begin{cases} 1 & \text{if player } c \in \mathcal{C}_\Omega \text{ is brought into the FPL squad for GW } t \in \mathcal{T}_{\Omega+1}, \\ 0 & \text{otherwise} \end{cases}$$

and

$$z_{c,t} = \begin{cases} 1 & \text{if player } c \in \mathcal{C}_\Omega \text{ is removed from the FPL squad for GW } t \in \mathcal{T}_{\Omega+1}, \\ 0 & \text{otherwise.} \end{cases}$$

4.4 Model constraints

The linking constraint set

$$x_{c,t} - x_{c,t-1} \leq y_{c,t}, \quad \begin{cases} c \in \mathcal{C}_1,\ t \in \mathcal{T}_2 & \text{if } \Omega = 1 \\ c \in \mathcal{C}_\Omega,\ t \in \mathcal{T}_\Omega & \text{if } \Omega > 1 \end{cases} \tag{6}$$

is required so as to ensure that $y_{c,t} = 1$ if $x_{c,t} = 1$ and $x_{c,t-1} = 0$, while the linking constraint set

$$x_{c,t-1} - x_{c,t} \leq z_{c,t}, \quad \begin{cases} c \in \mathcal{C}_1,\ t \in \mathcal{T}_2 & \text{if } \Omega = 1 \\ c \in \mathcal{C}_\Omega,\ t \in \mathcal{T}_\Omega & \text{if } \Omega > 1 \end{cases} \tag{7}$$

ensures that $z_{c,t} = 1$ if $x_{c,t-1} = 0$ and $x_{c,t} = 1$. Moreover, the constraint

$$\sum_{c \in \mathcal{C}_\Omega} x_{c,\Omega} = 15 \tag{8}$$

ensures that the FPL squad size is fifteen during GW Ω, while the constraint set

$$\sum_{c \in \mathcal{C}_\Omega} y_{c,t} - \sum_{c \in \mathcal{C}_\Omega} z_{c,t} = 0, \quad \begin{cases} t \in \mathcal{T}_2 & \text{if } \Omega = 1 \\ t \in \mathcal{T}_\Omega & \text{if } \Omega > 1 \end{cases} \tag{9}$$

is required to ensure that the FPL squad size remains unaltered over the remainder of the season. It is assumed that at most δ_t free player substitutions may be affected during GW $t \in \mathcal{T}_\Omega$, after which a penalty of four performance score points is incurred for each additional player substitution. This penalty scheme is enforced by imposing the constraint set

$$\sum_{c \in \mathcal{C}_\Omega} y_{c,t} + \sum_{c \in \mathcal{C}_\Omega} z_{c,t} \leq 2(\delta_t + \sigma_t), \quad \begin{cases} t \in \mathcal{T}_2 & \text{if } \Omega = 1 \\ t \in \mathcal{T}_\Omega & \text{if } \Omega > 1, \end{cases} \tag{10}$$

where σ_t represents the number of player substitutions over and above the free number δ_t of player substitutions available during GW $t \in \mathcal{T}_\Omega$, which therefore has to be penalised in the objective function.

The constraint set

$$\sum_{c\in\mathcal{C}_\Omega} k_{c,t}x_{c,t} + b_t = \begin{cases} B & \text{for } t = 1 \text{ if } \Omega = 1 \\ \sum_{c\in\mathcal{C}_\Omega} k_{c,t-1}x_{c,t-1} + b_{t-1} & \text{for all } t \in \mathcal{T}_2 \text{ if } \Omega = 1 \\ \sum_{c\in\mathcal{C}_\Omega} k_{c,t-1}x_{c,t-1} + b_{t-1} & \text{for all } t \in \mathcal{T}_\Omega \text{ if } \Omega > 1 \end{cases} \tag{11}$$

further fulfils the role of budgetary conservation of flow constraints, while the constraint set

$$\sum_{c\in\mathcal{C}_\Omega} \alpha_{c,d}x_{c,t} = r_d, \quad d \in \mathcal{D},\ t \in \mathcal{T}_\Omega \tag{12}$$

ensures that exactly r_d players fulfil role $d \in \mathcal{D}$ in the FPL squad during GW $t \in \mathcal{T}_\Omega$. The constraint set

$$\sum_{c\in\mathcal{C}_\Omega} \beta_{c,s}x_{c,t} \leq 3, \quad s \in \mathcal{S}_\Omega,\ t \in \mathcal{T}_\Omega \tag{13}$$

ensures that no more than three members of EPL team $s \in \mathcal{S}_\Omega$ are included in the FLP squad during GW $t \in \mathcal{T}_\Omega$.

Finally, the following domain constraint sets are imposed:

$$x_{c,t} \in \{0,1\}, \quad c \in \mathcal{C}_\Omega,\ t \in \mathcal{T}_\Omega, \tag{14}$$

$$y_{c,t}, z_{c,t} \in \{0,1\}, \quad \begin{cases} c \in \mathcal{C}_1,\ t \in \mathcal{T}_2 & \text{if } \Omega = 1, \\ c \in \mathcal{C}_\Omega,\ t \in \mathcal{T}_\Omega & \text{if } \Omega > 1, \end{cases} \tag{15}$$

$$\sigma_t \geq 0, \qquad \begin{cases} t \in \mathcal{T}_2 & \text{if } \Omega = 1, \\ t \in \mathcal{T}_\Omega & \text{if } \Omega > 1, \end{cases} \tag{16}$$

$$b_t \geq 0, \qquad t \in \mathcal{T}_\Omega. \tag{17}$$

4.5 Model objective

The model objective is to

$$\text{maximise} \quad Z = \sum_{c\in\mathcal{C}_\Omega}\sum_{t\in\mathcal{T}} p_{c,t}x_{c,t} - 4\sum_{t\in\mathcal{T}_2}\sigma_t. \tag{18}$$

The total projected performance score of all players included in the FPL team over all GWs is therefore maximised, while a penalty of four points is incurred during each GW for which the number δ_t of free substitutions is exceeded, and so the total penalty incurred has to be minimised (unless such a penalty is sufficiently cross-subsidised by the anticipated performance scores of the players involved in the substitutions).

5 Case study

The efficacy of the forecasting methods described in §3, when applied in the context of the mathematical model of §4, is evaluated in this section in the form of a case study involving real data obtained from the 2020/2021 FPL season. The case study allows for a retrospective comparative evaluation of the relative performance of FPL squad selections recommended by the model, based on forecast player performances, with the performances

of actual participants during the 2020/2021 FPL season. The section opens in §5.1 with a brief review of the FPL rules and a background discussion in §5.2 on the data pertaining to the case study. The results returned by the player performance forecasting methods described in §3 are then presented for the 2020/2021 FPL season, compared and discussed in §5.3. A brief discussion follows in §5.4 of the implementation of the model presented in §4, after which the model results are reported and placed in the context of the actual FPL competition pertaining to the 2021/2021 season in §5.5.

5.1 The rules of the FPL

This section contains a brief summary of the FPL rules, which are necessary to understand in the context of the case study in order to grasp the method of player performance quantification as a function of time in the FPL. After describing the composition of the initial squad in §5.1.1 and that of a required so-called opening-eleven subsquad in §5.1.2, the rules governing player transfers into and out of the squad are summarised in §5.1.3. Thereafter, the notion of playing special FPL chips is discussed in §5.1.4 and the schedule according to which decision deadlines are enforced in the FPL is described in §5.1.5. Finally, the method of scoring FPL player performance is recounted briefly in §5.1.6, and this is elaborated upon in the appendix at the end of the paper.

5.1.1 Selecting the initial squad

Managers are required to select an initial FPL squad of fifteen players (two goalkeepers, five defenders, five midfielders, and three forwards) from a large pool of EPL players. As mentioned in the introduction, each player has a monetary value associated with him (the cost of being included in the manager's squad) which typically ranges between £4 million and £14 million. The total value of all fifteen players in the manager's initial squad may not exceed £100 million. Moreover, managers may only select up to three players from any one of the twenty EPL teams.

5.1.2 Squad composition

Among the fifteen players selected for the FPL squad, eleven players have to be selected before each GW deadline — typically 90 minutes before the kick-off time of the first (EPL) match of the GW — in order to form a manager's FPL starting-eleven squad. All manager points for the GW are scored by these eleven players. In the case where some of the starting-eleven squad players do not play (in the EPL) during that GW, they may automatically be substituted by the remaining four players not included in the starting-eleven squad. Based on priorities specified by the manager, automatic substitutions are processed at the end of the GW, as follows:

- If the starting-eleven goalkeeper does not play during the GW, he is substituted by the replacement goalkeeper (if the replacement goalkeeper played during the GW).
- If any of the outfield players (who are not goalkeepers) do not play during the GW, they are substituted by the highest-priority outfield substitute who played during

the GW, provided that this does not violate the squad formation rules. These rules specify that the starting-eleven squad can play in any formation, provided that one goalkeeper, at least three defenders and at least one forward are selected at all times.

The manager is also required to select a captain and a vice-captain among the starting-eleven squad. The captain's score is doubled for the particular GW. In the case where the captain plays zero minutes during the GW, the vice-captain assumes the role of captain. If both the captain and the vice-captain play zero minutes during a GW, then no player's score is doubled.

5.1.3 Squad transfers

After the manager has selected an initial squad, (s)he is allowed to affect transfers to the squad by buying and selling players *via* a so-called transfer market. Unlimited transfers may be performed at no cost, until the first GW deadline. After the first GW deadline, however, managers receive one free transfer per GW. (Recall that four points are deducted from the manager's total score for each additional transfer made over and above this allowed number of free transfers during the same GW.) In the case where a manager decides not to use the free transfer, an additional free transfer is allowed during the following GW. If this saved free transfer is again not utilised during the following GW, it is carried over to the next GW, and so on, until used. Managers may, however, never have more than one saved free transfer during any GW.

Player costs change during the FPL season based on the popularity of the players in the transfer market. If many managers include a particular player in their squads, the player's popularity and hence his cost increases. Furthermore, player costs are fixed between the end of an FPL season and the start of the next FPL season.

5.1.4 FPL chips

The notion of using FPL chips may be understood as an opportunity to multiply a manager's points accumulated during a particular GW with a view to enhance a manager's squad performance over the season. At most one chip can, however, be played during any single GW. Four FPL chips are available to managers to choose from, as summarised in Table 3.

Chip	Effect
Bench Boost	The points scored by the substitute players during the next GW are included in the manager's total.
Free Hit	Unlimited free transfers can be made during a single GW. At the next deadline, the squad is, however, returned to how it was at the start of the GW.
Triple Captain	The captain's points are tripled instead of doubled during the next GW.
Wildcard	All transfers (including those already made during the GW) are free of any penalty score incurred.

Table 3: The available FPL chips and descriptions of their effects.

The Bench Boost and Triple Captain chips may be used only once during the course of a season and may be cancelled at any time before the relevant GW deadline. The Free Hit chip may also be used only once per season, is played when confirming player transfers, and cannot be cancelled after confirmation. The Wildcard chip may be used twice per season. The first Wildcard is available from the start of the FPL season, usually until 28 December. The second Wildcard becomes available after this date, in anticipation of the opening of the January transfer window, and remains available until the end of the FPL season. The Wildcard chip is played when confirming transfers that cost points and cannot be cancelled once played. When playing either a Wildcard or Free Hit chip, any saved free transfers from previous GWs are lost immediately.

5.1.5 FPL deadlines

All changes to a manager's team (including the starting-eleven squad, transfers, captain selection and the specification of substitution priorities) have to be made by the GW deadline in order to take effect for the set of matches played during the GW. There are 38 GWs throughout an FPL season. This number is based on the fact that twenty teams participate in the EPL each season and each team plays two matches against every other team (one game at their home stadium, and one game at an away stadium, against the same opposing team) — $19 \times 2 = 38$ matches.

5.1.6 FPL scoring

During an FPL season, players are allocated points based on their real-world performances in the EPL. As mentioned in the introduction, their performances are based on actions performed during EPL matches. Each action has a specific number of points associated with it, as described in some detail in the appendix at the end of the paper. These points measure the performance of an FPL player and the historical points allocations of each player are available as a performance time series for the player.

5.2 Input data pertaining to the 2020/2021 FPL season

FPL data acquired by the authors spanned multiple seasons over the period 2016–2021 and are publicly available[5]. These data sets contain information on FPL player performance, player attributes (*i.e.* their names, positions, EPL clubs, costs during each GW and a variety of other FPL information), the fixtures of the FPL season, and general information on teams participating in the EPL. The overall performance of each player also formed part of the data, as computed by the FPL according to the scoring method described in the appendix. The data sets of the 2016/2017 to 2019/2020 FPL seasons were reduced to include only those players participating in the most recent FPL season (2020/2021) for which data were available. This was done in order not to have to forecast player performance for players who could not be included in the FPL squad during the 2020/2021 season, because they no longer competed in the EPL.

[5]Available at: `https://github.com/vaastav/Fantasy-Premier-League`

The resulting five seasons of data were partitioned in the manner illustrated in Figure 4. The first four seasons, spanning GW 1 to GW 152, formed the so-called *training and validation set* on which the relative cross-validated performances of the player performance forecasting models were evaluated. The training and validation set sizes were chosen so as to be large enough that the forecasting models would be afforded a realistic chance to predict the performance of each player over an entire FPL season relatively accurately.

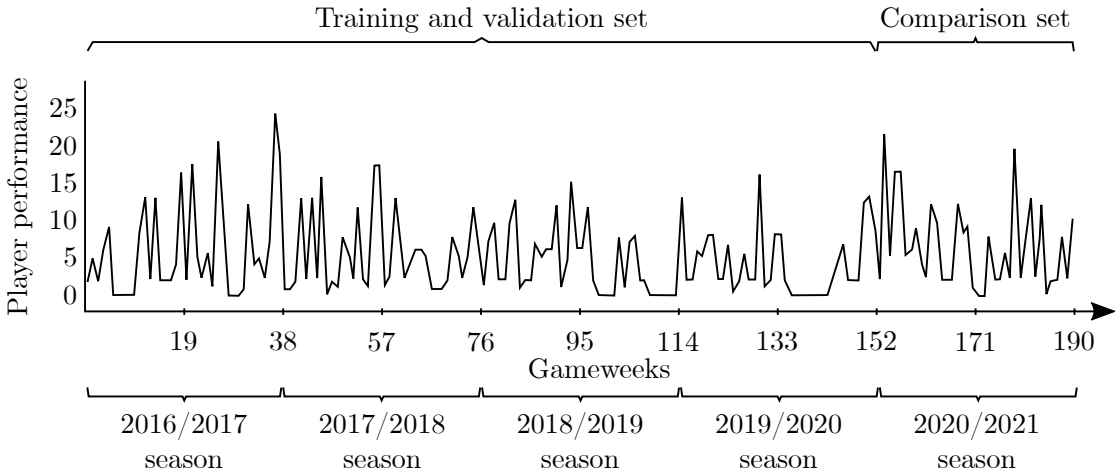

Figure 4: Partitioning of the historical performance data for the validation case study.

The player performances from GW 153 to GW 190 formed the so-called *comparison set* in terms of which the forecast performances returned by the forecasting methods of §3 and the optimisation model of §4 were compared retrospectively with the actual performances of managers participating in the FPL.

5.3 Forecasting player performance for the 2020/2021 FPL season

This section contains a brief description in §5.3.1 of the methodology adopted to train and compare the various forecasting methods of §3 in respect of the training and validation set, and a discussion in §5.3.2 of the process followed to ensemble top-performing combinations of these methods for the purpose of predicting FPL player performance based on past data.

5.3.1 Determining model (hyper)parameters

A rolling origin cross validation approach towards determining algorithmic parameters was adopted for the forecasting methods reviewed in §3. This approach was aimed at increasing the robustness of the forecasting methods employed and improving the accuracy of the final performance forecast for each FPL player. The rolling origin cross validation starting point, forecasting horizon and stride were taken as illustrated in Figure 5.

The statistical methods employed included a number of baseline methods (*i.e. naive mean* (Naive mean), *seasonal naive* (Naive last) and *naive drift* (Naive drift)) as well as ETS methods (EXP, EXP seas, EXP trend, EXP seas trend, and Theta), and an ARIMA method. In the case of the ARIMA method, the Auto ARIMA algorithm was employed

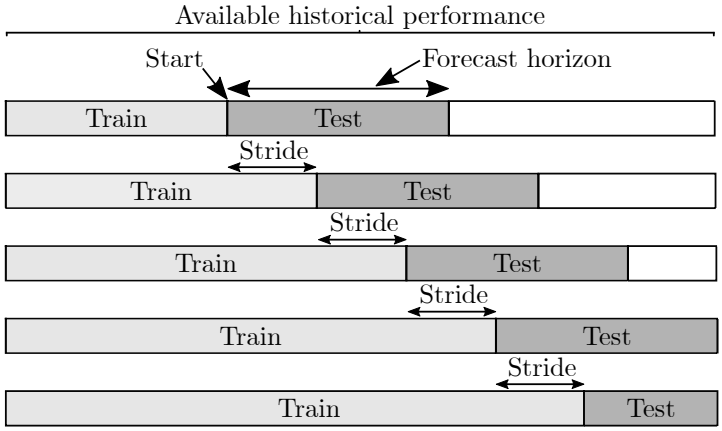

Figure 5: Input parameters required for cross validation during the training of the forecasting methods of §3.

as the sole ARIMA method. For the baseline methods, seasonal adjustments were implemented in the case of seasonal data.

A number of supervised machine learning methods were also implemented during the case study, namely tree-based methods (which included a *random forest* (RF), the *k-nearest neighbours* (KN) algorithm, the *XGBoost* (XGB) algorithm, the *LightGBM* (LGBM) algorithm) and *linear regression* (LR). The implementation of all of these methods was based on the time series reduction approach towards applying machine learning methods to time series data.

The performance of a particular forecasting method was evaluated in terms of its mean cross-validated error score. In cases where the forecast horizon extended past the available historical performance data, a reduced number of observations were merely included in the error score calculation, as illustrated in Figure 5.

The *mean absolute standard deviation* (MASE) performance measure was adopted when evaluating the relative performances of different forecasting methods in order to determine which methods may be labelled as appropriate for each time series. The MASE was chosen due to the intermittent nature of some of the time series. During this process, the parameters of the forecasting algorithms were also tuned by means of a grid search. The distributions of cross validated performance scores achieved by the various methods during training are illustrated by means of box plots in Figure 6.

5.3.2 Model building

An automated approach towards model building was adopted due to the large number of time series which had to be forecast. This automated approach entailed evaluating the relative performances of all the possible ensemble combinations of the three best-performing forecasting methods (as well as the individual performance of each method) for each player's performance time series. The best-performing ensemble or individual method was then taken as the final model ensemble or forecast model for each FPL player. The performances of each individual forecasting method, as well as ensembles represented by the sets Ensemble 1 = {Method 1, Method 2}, Ensemble 2 = {Method 1, Method 3}, Ensemble 3={Method 2, Method 3}, and Ensemble 4={Method 1, Method 2, Method 3}

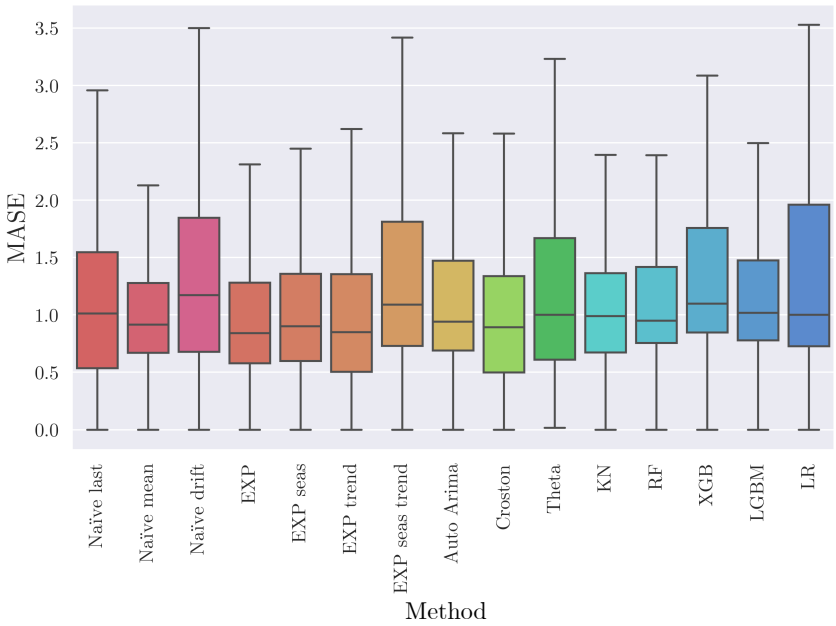

Figure 6: An evaluation of the relative cross-validated performances of the different time series forecasting methods.

were therefore evaluated in respect of each FPL player. The MASE performance measure was again adopted when evaluating the performances of the ensembled methods, based on a rolling origin cross validation. The results are presented in Figure 7. It is clear that the ensembled methods performed better than the individual forecasting methods.

The best-performing forecasting method (either an individual method or an ensemble), in terms of MASE score, was chosen for each FPL player as the final forecasting model for that player. Not all FPL players had enough data available for the (hyper)parameter tuning and model building phases of the aforementioned forecasting process, since these players had yet to compete in a season of the EPL on which the FPL is based. For these players, the best performing ensemble method for a player in a similar playing position and in the same EPL team was employed as the final forecasting model.

The final performances of the forecasting models are illustrated graphically in Figure 8 by means of box plots of the MASE scores over all GWs of the 2020/2021 FPL season. It is clear from the figure that the MASE score decreases over the course of the 2020/2021 FPL season, indicating that the accuracy of the forecasts increased from GW 153 to GW 190, as the forecasting horizons decreased.

5.4 FPL squads recommended for the 2020/2021 FPL season

Upon taking $\Lambda = 190$ and having forecast the performance scores $p_{c,\Omega}, \ldots, p_{c,190}$ for each player c in the set of $C_\Omega = 533$ EPL players for each of the GWs remaining in the 2020/2021 FPL season at our disposal, the model of §4 was invoked iteratively for $\Omega = 153, \ldots, 190$ to determine FPL squads (as well as a provisional recommendation of FPL squads for the following weeks of the 2020/2021 season).

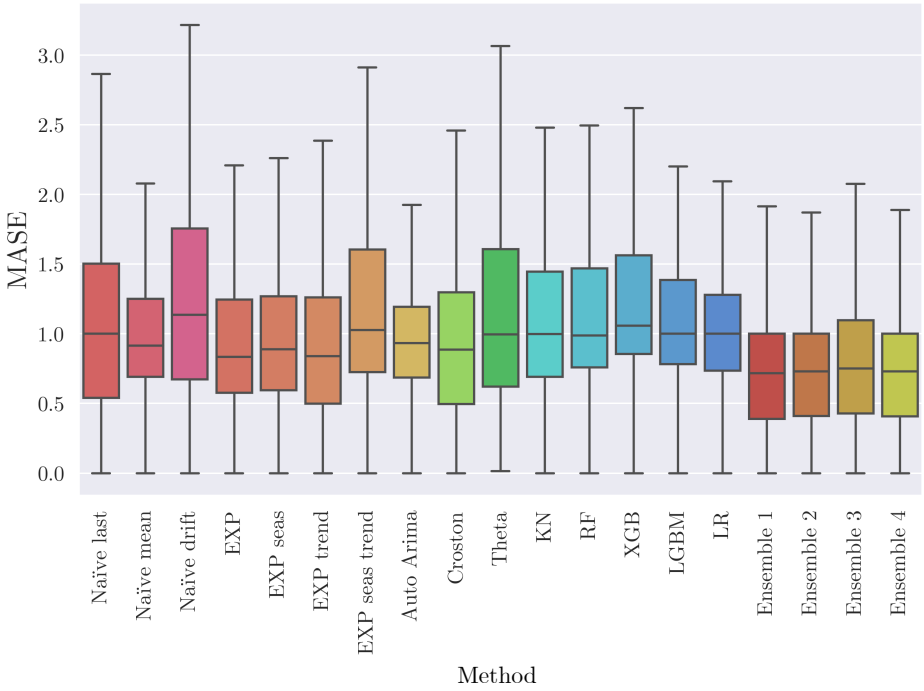

Figure 7: The relative cross-validated performances of the different time series forecasting methods, as well as method ensembles.

A simple binary programming model (called the *starting eleven model*) was employed to suggest a starting eleven from the FPL squad associated with each GW. The objective of this model was to

$$\text{maximise } Z = \sum_{c \in \mathcal{C}_\Omega} p_c x_c s_c, \tag{19}$$

where p_c denotes the performance score forecast for player c, and

$$s_c = \begin{cases} 1 & \text{if player } c \text{ is included in the starting eleven,} \\ 0 & \text{otherwise} \end{cases}$$

is a decision variable, while

$$x_c = \begin{cases} 1 & \text{if player } c \text{ is recommended for inclusion in the FPL squad,} \\ 0 & \text{otherwise} \end{cases}$$

is a parameter.

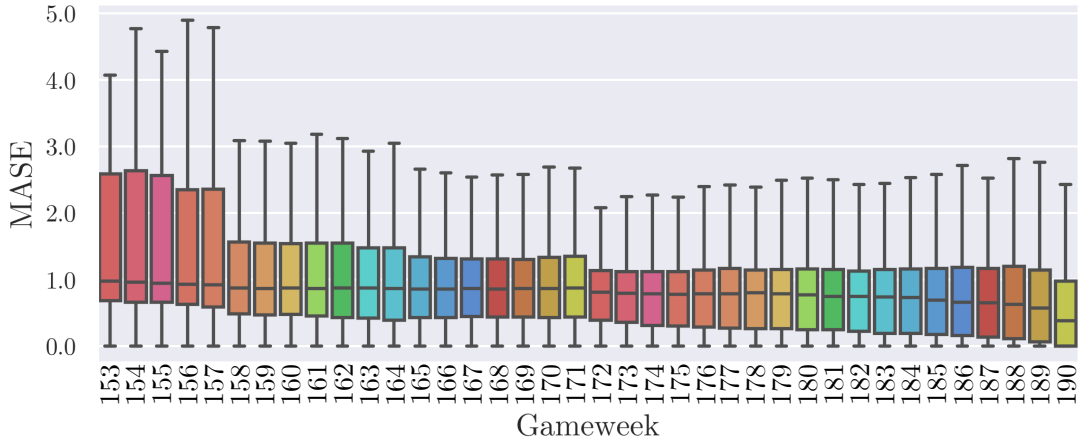

Figure 8: MASE scores for performance forecasts over the entire 2020/2021 FPL season.

The model constraints are

$$\sum_{c \in \mathcal{C}_\Omega} s_c = 11, \tag{20}$$

$$\sum_{c \in \mathcal{C}_\Omega} G_c s_c = 1, \tag{21}$$

$$\sum_{c \in \mathcal{C}_\Omega} D_c s_c \geq 3, \text{ and} \tag{22}$$

$$\sum_{c \in \mathcal{C}_\Omega} F_c s_c \geq 1, \tag{23}$$

where

$$G_c = \begin{cases} 1 & \text{if player } c \text{ is a goalkeeper,} \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, the parameters D_c and F_c are assigned the values 1 if player c is a defender or forward, respectively. Constraints (20)–(23) were included so as to ensure that all the required positions are filled in accordance with the FPL rules.

The captain and vice-captain were chosen as the players in the squad with the largest and second largest forecast performance scores. When a player in the starting eleven did not play during the GW for which he was chosen, he was replaced by a player from the bench, while ensuring that the constraints pertaining to the starting eleven are still satisfied. The substitute order was based on a ranking of those players not in the starting eleven according to decreasing forecast performance score.

Recall that the *free hit* and *wildcard* chips allow the FPL manager to make unlimited transfers, without incurring any penalty for exceeding the team transfer allowance specification. Therefore, two game weeks were identified during which the team transfer specification was set to 15, indicating that all the FPL players included in the FPL squad could potentially be transferred without incurring a penalty point reduction. Since the FPL manager may decide whether or not to use FPL chips, we opted to use the *wildcard, bench boost* and *triple captain* chips.

GW	Remaining budget	FPL score (no penalty)	Transfers	Penalties	FPL score (penalties included)
153	52	53	–	0	53
154	32	43	3	8	35
155	2	86	2	4	82
156	37	73	3	8	65
157	17	99	5	16	83
158	30	82	3	8	74
159	26	63	2	4	59
160	1	71	7	24	47
161	5	74	3	8	66
162	13	83	3	8	75
163	77	84	6	20	70
164	18	89	5	16	73
165	28	59	11	0	59
166	7	87	7	24	63
167	7	72	0	0	72
168	6	66	5	16	50
169	35	107	6	20	87
170	14	57	3	8	49
171	20	60	3	8	52
172	41	60	4	12	48
173	26	79	8	28	51
174	15	83	5	16	67
175	9	69	3	8	61
176	1	65	5	16	51
177	28	51	2	4	47
178	17	54	13	0	54
179	22	100	4	12	88
180	66	74	3	8	66
181	6	59	7	24	35
182	1	70	2	4	66
183	7	49	5	16	33
184	31	45	3	8	37
185	2	60	5	16	44
186	8	49	1	0	49
187	13	103	9	32	71
188	34	85	7	24	61
189	67	86	4	12	74
190	1	98	3	8	90
Total		2 755	170	448	2 307

Table 4: Case study results for the 2020/2021 FPL season.

A summary of statistics associated with the FPL squads recommended by the model of §4 for the GWs of the 2020/2021 FPL season may be found in Table 4. The remaining budget, number of transfers and the FPL score achieved for all GWs of the 2020/2021 FPL season are also shown in the table.

Note that the number of transfers during GWs 165 and 178 (the thirteenth and twenty sixth GWs of the 2020/2021 FPL season, respectively) are much higher than for the other GWs. This is a result of the implementation of the FPL chips. The *wild card* chip was implemented during GWs 165 and 178, since these points partitioned the planning horizon into three sections of approximately equal length. The *triple captain* and *bench boost* chips were randomly used during any of the last GWs of the season since the forecasting models were expected to be more accurate during the final stages of the season. GW 187 (the thirty fifth GW of the 2020/2021 FPL season) was chosen for this purpose — it was also decided to use these two FPL chips together to maximise their effect.

5.5 Appraisal of results achieved for the 2020/2021 FPL season

In order to evaluate the quality of our results above for the 2020/2021 FPL season, an optimal FPL squad was selected for each GW retrospectively, based on actual player performance, in the sense that a perfect "forecast" was employed after the fact. The optimal score achieved, along with the forecast score and the score actually achieved by our model for the 2020/2021 FPL season are shown in Figure 9.

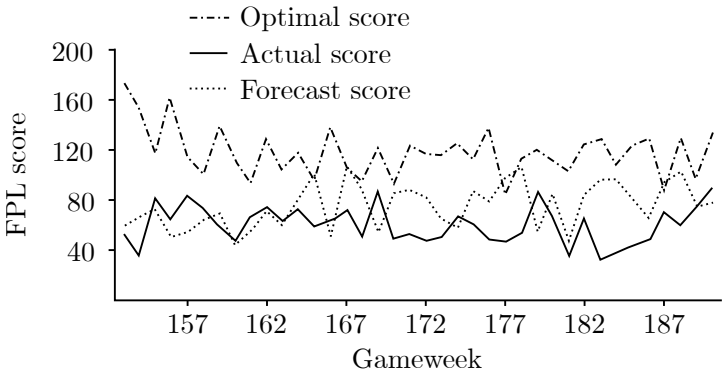

Figure 9: Scores over the entire 2020/2021 FPL season.

The relative quality of our results are summarised in Table 5. A total score of 2 307 was obtained by the model of §4, which would have resulted in an FPL rank of 336 193. A total of 8 240 321 FPL managers participated during the 2020/2021 FPL season, and so the forecasting methods of §3, in conjunction with the model of §4, would have allowed us to finish in the top 4.08% globally.

	Score	Rank	Percentile
Perfect forecast	4 501	1	$< 0.01\%$
Best FPL manager	2 680	1	$< 0.01\%$
The model of §4	2 307	336 193	4.08%

Table 5: Scores and ranks pertaining to the 2020/2021 FPL season.

We would likely have outperformed Dykman [18], who also employed time series forecasting methods to predict player performances but only optimised FPL squad selection one week in advance (*i.e.* without adopting an optimisation horizon stretching to the end of the season). The ranking achieved by Dykman [18] was 393 762 (based on a total score of 2 119), for a percentile of 6.7% during the 2017/2018 FPL season in which 5.2 million FPL managers participated.

5.6 Discussion

The results obtained during the forecasting process revealed which forecasting methods are better suited for forecasting FPL player performances. It was found that the EXP, Croston, EXP trend, EXP seas, Auto Arima, Naive mean and LGBM algorithms performed relatively well in respect of the FPL data. Furthermore, the forecasting results showed that the ensembled forecasters outperformed all other single forecasting methods.

The optimisation model of §4 returned promising results. Although the model is exact, it only placed within the top 4.08% of all FPL managers for the 2020/2021 FPL season because the quality of the model results depends on the quality of the forecast input data.

The computation time (measured in hours) required by the model of §4 is illustrated graphically in Figure 10. It is clear that the computation time significantly decreased as the GWs progressed towards the end of the FPL season, as expected. This decrease in computation time was a result of shorter and shorter look-ahead periods remaining in the season as the GWs elapsed towards the end of the FPL season.

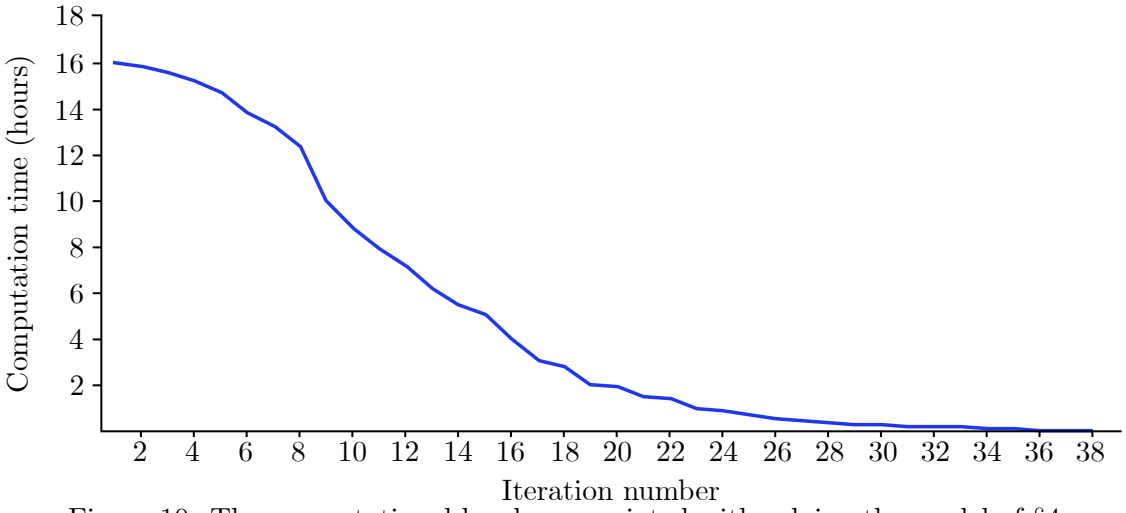

Figure 10: The computational burden associated with solving the model of §4.

6 Conclusion

In this paper, various forecasting methods were employed in conjunction with a combinatorial optimisation model to generate decision support for managers participating in the FPL. The forecasting methods were implemented in Python as part of the `sktime` package

while the optimisation model was implemented in CPLEX. As expected, the branch-and-cut method employed by CPLEX required considerable solution times at the start of the FPL season, but this decreased as the season progressed.

A practical case study was performed by applying the forecasting methods and optimisation model to historical FPL data dating from 2016 to 2021. After having forecast FPL player performance scores for the 2020/2021 FPL season, the optimisation model was invoked to recommend high-quality FPL squad selections for each GW of the 2020/2021 FPL season. The overall optimisation approach proved beneficial in recommending FPL squad selections that would have placed an FPL manager in the top 4.08% of all participants during the 2020/2021 FPL season.

References

[1] Aggarwal CC, 2015. *Data Mining: The Textbook*, Springer, New York (NY).

[2] Bangdiwala M, Choudhari R, Hegde A & Salunke A, 2022. *Using ML models to predict points in Fantasy Premier League*, Proceedings of the 2^{nd} Asian Conference on Innovation in Technology, Ravet, pp. 1–6.

[3] Bhatt S, Chen K, Shalin VL, Sheth AP & Minnery B, 2019. *Who should be the captain this week? Leveraging inferred diversity-enhanced crowd wisdom for a Fantasy Premier League captain prediction*, Proceedings of the 13^{th} International AAAI conference on Web and Social Media, Münich, pp. 103–113.

[4] Breiman L, 1996. *Bagging predictors*, Machine Learning, **24(2)**, pp. 123–140.

[5] Breiman L, 2001. *Random forests*, Machine Learning, **45(1)**, pp. 5–32.

[6] Brown RG, 1959. *Statistical Forecasting for Inventory Control*, McGraw Hill, New York (NY).

[7] Cerqueira V, Torgo L & Soares C, 1990. *Machine learning vs statistical methods for time series forecasting: Size matters*, [Online], [Cited February 2022], Available from `https://arxiv.org/pdf/1909.13316.pdf`.

[8] Chen T & Guestrin C, 2016. *Xgboost: A scalable tree boosting system*, Proceedings of the 22^{nd} Association for Computing Machinery (ACM) Special Interest Group International Conference on Knowledge Discovery and Data Mining, New York (NY), pp. 785–794.

[9] Clausen J, 1999. *Branch and bound algorithms — Principles and examples*, Lecture Notes, Department of Computer Science, University of Copenhagen, Copenhagen, pp. 1–30.

[10] Cleveland RB, Cleveland WS, McRae JE & Terpenning I, 1990. *STL: A seasonal-trend decomposition*, Journal of Official Statistics, **6(1)**, pp. 3–73.

[11] Cleveland WS & Devlin SJ, 1988. *Locally weighted regression: An approach to regression analysis by local fitting*, Journal of the American Statistical Association, **83(403)**, pp. 596–610.

[12] Croston JD, 1972. *Forecasting and stock control for intermittent demands*, Journal of the Operational Research Society, **23(3)**, pp. 289–303.

[13] Dantzig GB, 1963. *Linear Programming and Extensions*, Princeton University Press, Princeton (NJ).

[14] Dietterich TG, 2000. *Ensemble methods in machine learning*, Proceedings of the International Workshop on Multiple Classifier Systems, Berlin, pp. 568–581.

[15] Drayer J, Shapiro SL, Dwyer B, Morse AL & White J, 2010. *The effects of fantasy football participation on NFL consumption: A qualitative analysis*, Sport Management Review, **13(2)**, pp. 129–241.

[16] Dwyer B & Drayer J, 2010. *Fantasy sport consumer segmentation: An investigation into the differing consumption modes of fantasy football participants*, Sport Marketing Quarterly, **19**, pp. 207–216.

[17] Dwyer B & LeCrom CW, 2013. *Is fantasy trumping reality? The redefined National Football League experience of novice fantasy football participants*, Journal of Contemporary Athletics, **7(3)**, pp. 119–139.

[18] Dykman R, 2018. *Decision support for effective team selection in Fantasy Premier League*, Final-year Project, Department of Industrial Engineering, Stellenbosch University, Stellenbosch.

[19] Eiselt HA, 2000. *Branch and bound methods*, pp. 205–228 in Eisels HA & Sandblom CL (Eds), *Integer Programming and Network Models*, Springer, New York (NY).

[20] Gangal A, Talnikar A, Dalvi A, Zope V & Kulkarni A, 2015. *Analysis and prediction of football statistics using data mining techniques*, International Journal of Computer Applications, **132(5)**, pp. 8–11.

[21] Fantasy Premiere League, 2021. *Help*, [Online], [Cited September 2021], Available from `https://fantasy.premierleague.com/help/rules`.

[22] *Fantasy Sports and Gaming Association*, 2022. *Industry Demographics*, [Online], [Cited: Jan 2023], Available from: `https://thefsga.org/industry-demographics/`

[23] Gomory RE, 2010. *Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem*, pp. 77–103 in Junger M, Liebling TM, Naddef D, Nemhauser GL, Pulleybank WR, Reinelt G, Rinaldi G & Wolsley LA (Eds), *50 Years of Integer Programming 1958–2008*, Springer, Berlin.

[24] Holt CC, 2004. *Forecasting seasonals and trends by exponentially weighted moving averages*, International Journal of Forecasting, **20(1)**, pp. 5–10.

[25] Hyndman RJ & Athanasopoulos G, 2020. *Forecasting: Principles and Practice*, OTexts, Melbourne.

[26] James G, Witten D, Hastie T & Tibshirani R, 2013. *An Introduction to Statistical Learning*, Springer, New York (NY).

[27] Johnston M, Lathrop A & Mondor N, 2012. *Pigskin party: A statistical analysis on fantasy football and "the machine"*, Unpublished Research Report, Worcester Polytechnic Institute, Worcester (MA).

[28] Kalman RE, 1960. *A new approach to linear filtering and prediction problems*, American Society of Mechanical Engineers Journal of Basic Engineering, **82(1)**, pp. 35–45.

[29] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q & Liu TY, 2017. *LightGBM: A highly efficient gradient boosting decision tree*, Proceedings of the 13^{th} Annual Conference on Neural Information Processing Systems, Barcelona, pp. 3146–3154.

[30] Khamsan MM & Maskat R, 2019. *Handling highly imbalanced output class label: A case study on Fantasy Premier League (FPL) virtual player price changes prediction using machine learning*, Malaysian Journal of Computing, **4(2)**, pp. 304–316.

[31] King NA, 2017. *Predicting a quarterback's fantasy football point output for daily fantasy sports using statistical models*, PhD Thesis, University of Texas at Arlington, Arlington (TX).

[32] Kolesar PJ, 1967. *A branch and bound algorithm for the knapsack problem*, Management Science, **13(9)**, pp. 723–735.

[33] Kristiansen BK, Gupta A & Eilertsen W, 2018. *Developing a forecast-based optimisation model for Fantasy Premier League*, MSc Thesis, Norwegian University of Science and Technology, Trondheim.

[34] Land AH & Doig AG, 2010. *An automatic method for solving discrete programming problems*, pp. 105–132 in Junger M, Liebling TM, Naddef D, Nemhauser GL, Pulleybank WR, Reinelt G, Rinaldi G & Wolsley LA (Eds), *50 Years of integer programming 1958–2008*, Springer, Berlin.

[35] Landers JR & Duperrouzel B, 2018. *Machine learning approaches to competing in fantasy leagues for the NFL*, IEEE Transactions on Games, **11(2)**, pp. 159–172.

[36] Larose DT, & Larose CD, 2014. *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons, Hoboken (NJ).

[37] Löning M, Bagnall A, Ganesh S, Kazakov V, Lines J & Király FJ, 2019. *Sktime: A unified interface for machine learning with time series*, [Online], [Cited February 2022], Available from `https://arxiv.org/pdf/1909.07872.pdf`.

[38] Lutz R, 2015. *Fantasy football prediction*, arXiv preprint, arXiv:1505.06918.

[39] Makridakis S, Wheelwright SC & Hyndman RJ, 2008. *Forecasting methods and applications*, John Wiley & Sons, New York (NY).

[40] Maniezzo V & Aspee Encina FA, 2022. *Predictive analytics for real-time auction bidding support: A case on fantasy football*, Operations Research Forum, **3(3)**, pp. 1–23).

[41] Mendes-Moreira J, Soares C, Jorge AM & Sousa JFD, 2012. *Ensemble approaches for regression: A survey*, ACM Computing Surveys, **45(1)**, pp. 1–40.

[42] Montgomery DC, Jennings CL & Kulahci M, 2015. *Introduction to Time Series Analysis and Forecasting*, John Wiley & Sons, Hoboken (NJ).

[43] Nesbit TM & King KA, 2010. *The impact of fantasy football participation on NFL attendance*, Atlantic Economic Journal, **38(1)**, pp. 95–108.

[44] O'Brien JD, Gleeson JP & O'Sullivan DJ, 2021. *Identification of skill in an online game: The case of Fantasy Premier League*, PloS One, **16(3)**, Manuscript e0246698.

[45] Patel D, Shah D & Shah M, 2020. *The intertwine of brain and body: A quantitative analysis on how big data influences the system of sports*, Annals of Data Science, **7(1)**, pp. 1–6.

[46] Pawlikowski M & Chorowska A, 2020. *Weighted ensemble of statistical models*, International Journal of Forecasting, **36(1)**, pp. 93–97.

[47] Petrakova A, Affenzeller M & Merkurjeva G, 2015. *Heterogeneous versus homogeneous machine learning ensembles*, Information Technology and Management Science, **18(1)**, pp. 135–140.

[48] Petropoulos F, Hyndman RJ & Bergmeir C, 2018. *Exploring the sources of uncertainty: Why does bagging for time series forecasting work?*, European Journal of Operational Research, **268(2)**, pp. 545–554.

[49] Polikar R, 2012. *Ensemble Learning*, pp. 1–34 in Zhang C & Ma Y(Eds), *Ensemble Machine Learning*, Springer, Boston (MA).

[50] Pukdee R, 2021. *Network analysis in team sports and applications to English Premier League*, MA thesis, University of Oxford, Oxford.

[51] Rajesh V, Arjun P, Jagtap KR, Suneera CM & Prakash J, 2022. *Player recommendation system for fantasy premier league using machine learning*, Proceedings of the 19^{th} International Joint Conference on Computer Science and Software Engineering, Bangkok, pp. 1–6

[52] Roli F, Giacinto G & Vernazza G, 2001. *Methods for designing multiple classifier systems*, Proceedings of the International Workshop on Multiple Classifier Systems, Cambridge, pp. 78–87.

[53] Schapire RE, 1990. *The strength of weak learnability*, Machine Learning, **5(2)**, pp. 197–227.

[54] Shumway RH & Stoffer DS, 2000. *Time Series Analysis and its Applications: With R Examples*, Springer, New York (NY).

[55] Surowiecki J, 2005. *The Wisdom of Crowds*, Anchor Books, New York (NY).

[56] Teunter RH, Syntetos AA & Babai MZ, 2011. *Intermittent demand: Linking forecasting to inventory obsolescence*, European Journal of Operational Research, **214(3)**, pp. 606–615.

[57] Thapaliya R, 2018. *Using machine learning to predict high-performing players in Fantasy Premier League*, [Online], [Accessed: January 2023], Available from: `https://medium.com/@277roshan/machine-learning-to-predict-high-performing-players-in- fantasy-premier-league-3c0de546b251`.

[58] Tsoumakas G, Partalas I & Vlahavas I, 2009. *An ensemble pruning primer*, pp. 1–13 in Okun O & Valentini G (Eds), *Applications of Supervised and Unsupervised Ensemble Methods*, Springer, Berlin.

[59] Whittaker D, 2022. *A study of information behaviour in the Fantasy Premier League community*, MSc Thesis, City University of London, London.

[60] Wilkins L, Dowsett R, Zaborski Z, Scoles L & Allen PM, 2021. *Exploring the mental health of individuals who play fantasy football*, Human Behavior and Emerging Technologies, **3(5)**, pp. 1004–1022.

[61] Winston WL & Goldberg JB, 2004. *Operations Research: Applications and Algorithms*, Thomson Brooks/Cole, Belmont (CA).

[62] Winters PR, 1960. *Forecasting sales by exponentially weighted moving averages*, Management Science, **6(3)**, pp. 324–342.

[63] Zanjirani R & Ghasemitart F, 2001. *A branch-and-bound method for finding flow-path designing of AGV systems*, International Journal of Engineering, **15(1)**, pp. 81–90.

[64] Zietsman HJ, 2021. *Quantitative time series forecasting*, Technical Report 2021-05, Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, Stellenbosch.

Appendix: FPL points scoring

Each action of a player in an EPL match has a specific number of FPL points associated with it, as summarised in Table 6.

An assist is awarded to the player from the goal scoring team who makes the final pass before a goal is scored. Such an assist is awarded irrespectively of whether the pass was

Points	Action
1	For playing up to 60 minutes
2	For playing 60 minutes or more (excluding stoppage time)
6	For each goal scored by a goalkeeper or defender
5	For each goal scored by a midfielder
4	For each goal scored by a forward
3	For each goal assist
4	For a clean sheet by a goalkeeper or defender (zero goals against their team)
1	For a clean sheet by a midfielder (zero goals against their team)
1	For every three shot saves by a goalkeeper
5	For each penalty save by a goalkeeper
–2	For each penalty miss by an outfield player
1–3	Bonus points for the best players in the match
–1	For every two goals conceded by a goalkeeper or defender
–1	For each yellow card
–3	For each red card
–2	For each own goal

Table 6: Actions corresponding to FPL points earned during EPL matches.

intentional (*i.e.* it actually created a goal-scoring opportunity) or unintentional (*i.e.* the goal-scoring player first had to dribble the ball before scoring, or an inadvertent touch or shot created the chance). If an opposing player touches the ball after the final pass just before a goal is scored, significantly altering the intended destination of the ball, then no assist is awarded. Should a touch by an opposing player be followed by a defensive error by another opposing outfield player, then no assist is awarded. Also, in the case where the goal scorer loses and then regains possession, no assist is awarded. Other intricacies of awarding assists to players are governed by the following three rules:

Rebounds. If a shot on goal is blocked by an opposing player, saved by the goalkeeper, or hits the woodwork of the goal box, and a goal is then scored from the rebound, an assist is awarded.

Own goals. If a player shoots or passes the ball and thereby forces an opposing player to put the ball in his own net, then an assist is awarded.

Penalties and free-kicks. The player earning the penalty or free-kick is awarded an assist if a goal is scored directly from the penalty or free-kick, but not if he takes it himself, in which case no assist is awarded.

A clean sheet is awarded to a player for not conceding a goal whilst on the field and playing at least 60 minutes of the match. In the case where a player has been substituted in the EPL match, and afterwards a goal is conceded, this does not affect any clean sheet bonus.

If a player receives a red card (resulting in the player no longer participating in the remainder of the match), he will continue to be penalised for goals conceded by his team. Red card deductions also include any points deducted for yellow cards.

A *Bonus Points System* (BPS) utilises a range of statistics in order to generate a BPS score for every player. The three best performing players in each match are awarded bonus points, with three points being awarded to the highest scoring player, two points to the second highest scoring player, and one point to the third highest scoring player.

Bonus point ties are resolved as follows:

1. If there is a tie for first place, Players 1 and 2 receive 3 points each, and Player 3 receives 1 point.
2. If there is a tie for second place, Player 1 receives 3 points, and Players 2 and 3 receive 2 points each.
3. If there is a tie for third place, Player 1 receives 3 points, Player 2 receives 2 points, and Players 3 and 4 receive 1 point each.

Players score BPS points based on a different set of actions, as listed in Table 7. The allocation of points for assists, clean sheets and receiving red cards are handled in the same manner as described in Table 6.

Points	Action
3	Playing less than or equal to 60 minutes
6	Playing more than 60 minutes
12	Goal scored by a goalkeeper or defender
18	Goal scored by a midfielder
24	Goal scored by a forward
9	Assists
12	Goalkeepers and defenders keeping a clean sheet
15	Saving a penalty
2	Saving a shot on goal
1	Successful open play cross
3	Creating a significant chance (a chance where the receiving player should score)
1	For every two clearances, blocks and interceptions (total)
1	For every three recoveries
1	Key pass
2	Net successful tackles (total of successful tackles minus unsuccessful tackles)
1	Successful dribble
3	Scoring a match winning goal
2	70 to 79% pass completion (at least 30 passes attempted)
4	80 to 89% pass completion (at least 30 passes attempted)
6	90%+ pass completion (at least 30 passes attempted)
–3	Conceding a penalty
–6	Missing a penalty
–3	Yellow card
–9	Red card
–6	Own goal
–3	Missing a significant chance
–3	Making an error which results in conceding a goal
–1	Making an error which results in an attempt at goal
–1	Being tackled
–1	Conceding a foul
–1	Being caught offside
–1	Shot off target

Table 7: Actions corresponding to FPL points earned during EPL matches.