Optimized selection-based technique of Score Prediction for Fantasy Premier League

Final Project Report - CSE-519: Data Science Fundamentals - Fall 2021 - Stony Brook University
Link to Google Drive

Abstract

In Fantasy Premier League (FPL), managers create teams with the objective of scoring the highest number of points. In this report, we propose a model of team creation with the aim to maximize the points following FPL rules and constraints.The initial step involves feature selection, based on correlation and model performance for predicted scores. Prediction on the selected features is used to predict the final scores for the players. Linear Optimization**[8]** is adopted to select a team using the expected scores and transfer strategies are applied to optimize the team's total score over the game weeks. The results are evaluated against actual points for the predicted team and points scored by renowned manager Magnus Carlsen in the GWs 1-12 for season 2021-22.

1 Project Overview

In this project we aim to predict scores of players and using the predicted scores create an optimal team while satisfying the rules and constraints of FPL. We also develop the strategies to transfer players between seasons so the team created gives the best score.

- **Feature Selections**: Features were selected based on the correlation matrix. These features were then preprocessed further to suit our needs.

- **Feature Predictions**: Some of the features such as goals scores, goals conceded are not available before the match starts. We predict values for these features using data from past seasons and past GWs.

- **Point Predictions**: Using the features, we tried different Regressors to predict the points scored by a player in the first game week of a season. We choose Random Forest Regressor since we obtained the lowest RSME value for it while predicting total points.

- **Team Selection for GW**: Using the points predicted , we use Linear Optimization to get an optimal team within the FPL rules and constraints.

- **Transfer Strategy**: Based on certain parameters, we incorporate a transfer strategy using Linear Optimization. We use linear programming so that the swaps performed are based on FPL rules and constraints.

- **Evaluation**: To evaluate our team, we use the current season i.e 2021-22. The predicted scores of the team is compared with the actual scores of the team for the current season. We also compare the team with the team of one of the renowned managers Magnus Carlsen. The evaluation metric used is **Root Mean Squared Error (RMSE)**.

- **Project Progress**: In the mid progress report, we had developed a baseline model by using performance cost index(PCI). This PCI was developed by using the average performance of a player of the previous GWs divided by the cost of the player. The baseline model did not take into consideration features which could have impacted the performance from the previous GWs. We improve on our baseline model by predicting points of players for a particular GW and then using linear programming to develop a team out of it. To support the prediction of teams we use feature selection, feature predictions which was not used while developing the baseline model.
The transfer strategy for the baseline model used fixtures and team ranking into considerations. We now take into fixtures, team rankings, performance in previous matches and home away and use linear programming to get the best transfers under the constraints.
In the previous report, we did not evaluate our

team. We now evaluate our team based on the actual performance of the predicted team and also compare its performance with the team of Magnus Carlsen who is one of the renowned FPL managers.

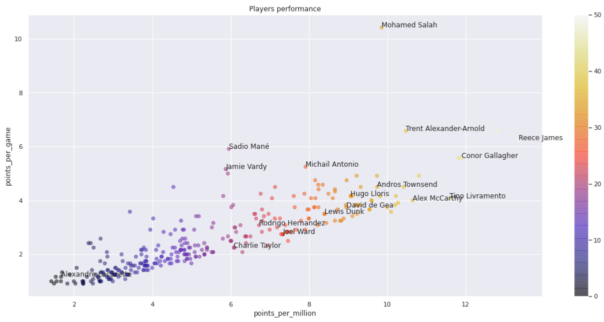

Figure 1: Distribution of Value for Money of player w.r.t Total Points in current season

2 Previous Challenges, Review Comments, and Resolutions

- The majority of the comments were regarding the data pre-processing which we did not mention in the progress report. For using RNN or regression, firstly, we do need to have feature values for the next GW, and then predict the total points, which we didn't predict earlier.Pre-processing of every feature has been explained in **Section 3**.

- We didn't include much visualization and graphs earlier. We plotted the Initial Playing XI of GW1 and team predicted after swaps with and without penalty. We didn't show what's the difference between the actual best team and our predicted team. We have incorporate in **Section 5**.

- We considered the transfer strategy considering only the team fixtures and did not explain the features used. Now, we have considered player form, team fixture, team rankings, home vs away match, and player availability. Discussed in. **Section 4.1.2**.

- Model Evaluation and validation were not explained in depth. We have explained them in **Section 5**.

3 Data Pre-Processing

The problem turns out to be predicting features values for the next GW first, then using these predicted features, we will predict the total points scored. After the analysis of features on the correlation matrix, we've selected the features: **minutes played, goals, assists, ICT index, influence, threat, creativity, clean sheet, yellow cards, red cards, saves, goals conceded, bonus, bps(bonus point system)**. We'll consider the player's performance in last years but we'll use the fudge factor over the season so that the past performances are still considered but not have equal impact- suffices our feature predictions for new season's GW1(**History**). Similarly, over the current season, player's current form(performance in last 5 matches) and previous History are considered. Some other features like yellow card, we are extrapolating using probability of occurrence of the event, so that while extrapolating for every GW, values won't be similar. A detailed analysis of feature pre-processing as follows:

- It's better to break the data based on players position(GK, FWD, DEF, MID), as goalkeepers are more correlated with clean sheets, goals conceded, saves, etc. They generally play complete 90 minutes and they cost nearly same i.e. their budget bracket don't vary much(£4.5 - £6 M). On the other hand, goals scored, assist, ICT, bps, etc, much correlates with Forwards and Midfielders. We did the prediction on complete dataset at once and also segregating the data based on positions, the **RMSE** obtained was much lower for the later than the original data. **validation RMSE on complete - 0.65, on segregated - 0.38**

- Features like Yellow Card, Clean Sheets, bonus are better to predict using probability. In every match, there are 4 possible value of **bonus points**: 0,1,2,3. For each new GW, we can find whether the player will concede a yellow card or not and whether a GK will have a clean sheet or a bonus based on the probability of these events occurring. (**Refer 4.2**) Probability of getting a yellow card and clean sheet by each player defined as:

$$P(YC) = \frac{Total\ Yellow\ cards}{Total\ number\ of\ matches\ played}$$

- We have to consider fudging in some features like minutes played, goals scored, assists, ICT, goals conceded, etc and divided it by total matches played. We want to consider the players who generally play long runs(by minutes), and whenever they are on field, there are many chances of scoring of goals(like Mason Mount, who doesn't play every match). We have used the fudge factor to give some significance to the past performance. Taking the avg of this season, adding it to the avg of last season * fudge factor.

$$currMean = \frac{Total\ assists\ in\ a\ season}{Total\ matches\ played}$$

*the assist formula will be applied for goals scored, bps, ICT index, influence, creativity, threat, value, goals conceded,and saves.

$$ftrVal = \frac{oldVal{*}fudgeFactor\ +\ currMean}{1+\ fudgeFactor}$$

- **Data Imputation**: We have removed the players who played till last season but not featuring in current season. We have done the mean imputation for players who didn't play till last season but featuring now for GW1 predictions.

4 Implementation

4.1 Baseline Model Implementation

The objective of baseline model implementation was to construct a bench mark for comparison of the final model performance and understand how to improve on the features selection. This also allowed us to understand how naive approach for team selection and formation, performs over the entire league. The data used for the model creation was for the league seasons from 2015-21. The model evaluation was performed against actual data from the season 2021-2022.

The factors used in the indexing of the players were as follows. The attributes used are *Performance (pf)* i.e. the scores obtained by the player and the *Cost (ct)* of the Player. Using these factors, we calculated an index, called *Performance-Cost-Index (pci)*.

$$pci_i = \frac{pf_i}{ct_i} \qquad \forall i \in players$$

For the analysis of performance of the players, we used the the scores per games played rather than using the raw score data. This was done to standardize the judgment scale of how good or bad a player performs. In other words, if a player plays large number of games but was able to perform extremely well in only a handful of them, could be perceived as high-performing player. In contrast, a consistent player who performed well in average over the games played, could be a better choice in team selection.

4.1.1 Team Selection

- Based on the index of the players, as explained above, grouping was performed on the positions of players in the team. After grouping, the players were ranked in top down order to select the best players in each group.

- Once these players rankings are calculated, the next step was to form a team of 15 players that can be formed for the budget limit of £100M and defined constraints on positions of players & team restrictions (add the constraints). This is a variation of classic Knapsack Optimization problem. The aim was to form the team by using the limit of budget allowed and best possible ranked player selection.

- Since there is a linear relationship between the constraints for team formation, this becomes a linear optimization problem and hence linear programming was applied to optimize the team selection.

- Linear Programming consists of an objective function (add the objective) that needs to be maximized [1]. Here, the constraints were on the cost of the player than can be selected in the team against the available budget & the position of the players i.e. the limits on the number of players from each position that can be included to form the team.

- The top rewarding player out of the selected 15 players is made the captain, since the scores of the captain are doubled. The least rewarding players are considered as substitutes depending upon the position constraints.

4.1.2 Transfer Strategy

The fixtures of next GW and the rankings of the teams is taken into consideration for the transfer decision on players. The team with higher ranking would tend to dominate team with lower rank, and so the players from a higher ranked team would have advantage if he plays against such team in next GW. A scaling factor was assigned to each player's score based on above understanding and then decided which player should be transferred out.

Once decided, using the linear programming model again, we find the player that could be transferred in based on the available budget and free position after transfer out. The budget available is calculated considering the current value of the player.

4.2 Final Model Implementation

For the baseline model, '**features selection**' was not an implemented and it consisted of using only two parameters: performance and cost. In the final model implementation, the feature selection was performed by evaluating two analysis:

1. Analysis of correlation between features and especially the total points scored by each player. Since these features are highly dependent on the position of the player, the correlation was analyzed segregated by the positions.

2. Using the features set to analyse the performance of the prediction models against the data available. (if time remains add the accuracy of the best chosen model).

4.2.1 Features Prediction

Based on the analysis on features selection, a set of features that impacted the scores of the players the most were narrowed down. These features were { }. Since the distribution and pattern of these features values are not uniform, the prediction of features was divided into two approaches.

- **Weighted Moving Average**: This approach involved using the average values for the features for each of the players for the seasons 2015-21. Due to possibility of change in forms of the players in prior seasons and to improve the judgment of a player's current form, a weight factor was used.

  This weight factor allowed the older seasons data for the players to be interpreted of slightly less importance than the most recent performances of the players. The weight factor (also termed as '**fudge factor**' was decided after multiple iterations and evaluating the proximity of predicted feature values with the actual values.

  The weighted averaging of the values was performed for each player against an entire season with the values obtained from prior seasons. The sliding window of the timeline to be used for this averaging was one season till the beginning of current season 2021-22 (for which the final scores predictions are being made).

  For the current season, the optimum moving average window decided is of **5 GWs**. The fudge factor used for weighted average in this season is different from the value used for past seasons weighted average.

- **Random Probabilistic Sampling**: For the features "**bonus, yellow cards, red cards, clean sheets**", the weighted average model did not make much relevance. To predict values for these, the probability of values for each of the players were used to generate random values as per the allowed feature values. This allowed better distribution of the feature values based on the likelihood of the occurrence for the player in actual game.

Algorithm 1 Random probability sampling for feature values prediction

function PREDICTPROBVAL($feature, player$)
  $init$: $featureValues$
    ▷ count distinct feature values for a player
  $init$: $allowedVal$
    ▷ allowed values for that feature
  $init$: $probValues[]$
    ▷ probability of each distinct feature value
  for $val \leftarrow 1$ to $featureValues$ **do**
    $probVal = calculateProb(val)$
      ▷ calculate probability for 'val'
    $randVal = rand(probVal, allowedVal)$
    $probValues \leftarrow$ append $randVal$
  end for
  return $probValues$
end function

4.2.2 Score Prediction Model

Once the feature values prediction are completed, the next significant task was to predict the scores for the players for the GWs in season 2021-22. For the baseline model, we did not incorporate any learning model for score prediction. For the final

Algorithm 2 Fudge-Weighted Averaging for feature value predictions

function PREDICTFUDGEVALUE(feature, player)
$init : fVal$
▷ feature value to be calculated using past seasons historical data
$init : fudgeFactor$
▷ fudging factor to generalize past performance with current performance
$pastVal \leftarrow$ calMeanHistoricalVal(player, feature)
$currentVal \leftarrow$ current feature value
$pastVal = pastVal$ x $fudgeFactor$
▷ Applying fudge factor to generalize past player performance with current
$fVal = weightedMean(pastVal, currentVal)$
▷ Calculating weighted average over fudged past value and current value
return $fVal$
end function

Algorithm 3 Predicting feature values for upcoming GW

function PREDICTVALUEGW($player$)
$init : fVals$
▷ feature value to be calculated
$featureSet = \{$assists, bps, ict index, influence, creativity, threat, goals scored, goals conceded$\}$
for $feature$ in {bonus, yellow cards, red cards} **do**
$fVals \leftarrow predictProbVal(feature, player)$
end for
for $feature$ in $featureSet$ **do**
$fVals \leftarrow predictFudgeValue(feature, player)$
end for
end function

model, we implemented and compared Regression, Decision Trees and RNN-LSTM learning models for score predictions.

- **Final Model Implemented**: Using the weighted moving average method ensemble with the **Random Forest Regression** algorithm produced good results for the score prediction. The performance and evaluation of the final model have been explained in later section. The performance of the Random Forest was found to be slightly better than **XG-Boost Regression** algorithm.
- **Regression Models**: Other regression models that were implemented and compared were **Linear**, **Ridge** and **Lasso** Regression models. The ranking of the models based on the accuracy and overall RMSE for all the players was found to be in the aforementioned order.
- **RNN-LSTM Model**: As the data for the FPL is present with each of the GWs in the seasons from 2015-21, the initial plan was to use Time Series Analysis for score predictions using RNN-LSTM model. This model was not adopted as the final model due to the following conclusions:
  - The data for few players like Salah was abundant, due to the reason that, a particular player plays more GWs in almost all of the seasons than the others. This resulted in high difference in availability of data points for the time series for the players over the GWs.
  - On implementing the RNN model over these time series, the score prediction results over predicted feature values did not fare well and large difference between the predicted and actual scores was observed.

4.2.3 Final Team Selection Model

- In contrast with the baseline model, wherein, the indexing and ranking of the players was performed to make the team selection decisions, for the final model the predicted scores for upcoming GWs are used. The linear optimization problem for satisfying the team, cost and position constraints remain the same. The maximizing linear programming algorithm was implemented for the final team selection as well.

Position	minStart	maxStart	total
GK	1	1	2
DEF	3	5	5
MID	3	3	5
FWD	1	3	3

Table 1: Constraints on number of players for each position in team selection.

Based on the constraints and data at hand the selected players are segregated as playing XI and substitute players. The pseudo code presented explains how these players selections are made. It also states how the captain selection is done in the team.

Team Selection Constraints
Let F be the objective function to maximize the linear optimization problem.

- **Constraints on number of players from each position:**
  - Midfielders can be between 3 and 5: $F_m = \sum_{k=3}^{5} \binom{N_m}{k} \cdot L_p(x)$
  - Defenders can be between 3 and 5: $F_d = \sum_{k=3}^{5} \binom{N_d}{k} \cdot L_p(x)$
  - Forwards can be between 1 and 3: $F_f = \sum_{k=1}^{3} \binom{N_f}{k} \cdot L_p(x)$
  - Goalkeepers can be between 1 or 2: $F_g = \sum_{k=1}^{2} \binom{N_g}{k} \cdot L_p(x)$
- **Constraints on number of players from each club:** There can be at most 3 players from each club in the selected 15. Let C be the function that limits the selection of players in team based on this constraint.

$$1 \leq \sum_{n=1}^{15} C(P_n) \leq 3$$

4.2.4 Transfer Strategy

Transfer strategy is as important as building up the team. As we have predicted the feature values and expected points for each player, and we know the fixtures of future GWs too, we can certainly penalize/reward the predicted points of the teams based on home and away match and current standings of teams. We have used data from **[10]** to find the probability of winning and losing of each team based on next GWs fixtures.**[16]**

Algorithm 4 Transfers Constraints

```
function TRANSFERS(inDecisions, freeInDecisions, outDecisions, budgetNow, buyPrice, sellPrice)
    LpModel += sum(freeInDecisions) <= 1
    LpModel += sum(inDecisions)==sum(outDecisions)

    transferInCost = sum(inDecisions * buyPrice)
    transferOutCost = sum(outDecisions * sellPrice)

    budgetNextWeek = budgetNow + transferOutCost - transferInCost
    LpModel += budgetNextWeek >= 0
end function
```

Algorithm 5 Formation Constraints

```
function TEAMFORMATION(squad, starters, subs, captains)
    for position, data in {PositionData} do
        LpModel += sum(starters[position]) >= position[minStarters]
        LpModel += sum(starters[position]) <= position[maxStarters]
        LpModel += sum(squad[position]) <= position[numTotal]
    end for
    for club in {clubs} do
        LpModel += sum(squad[club]) <= 3
                            ▷ max 3 players from each club
    end for
    LpModel += sum(starters) == 11
    LpModel += sum(squad) == 15
    LpModel += sum(captain) == 1

    for i in {players} do
        LpModel += (starters[i] - captains[i]) >= 0
                            ▷ Captain should be in playing XI
        LpModel += (transferIn[i] != transferOut[i])
                            ▷ Transfer in and out should not be same
        LpModel += (starters[i] + subs[i]) <=1
                            ▷ Subs should be different from playing XI
    end for
end function
```

Figure 2: Fixtures of GW2 & GW3 with win and draw%

Following are the approaches in our Optimal transfer strategy:

- Refer the algo[6] for Optimized Transfer Strategy with Formation algo[5] and Transfer Constraints[4] provided to the LP Model.**[4]**

- There are 2 transfer strategies: with and without penalty. We'll consider both the strategies with the objective to Maximize our expected score. There are some

- There are 2 transfer strategies: with and without penalty. We'll consider both the strategies with the objective to Maximize our expected score. The transfer strategy with penalty will be beneficial at times when a lot of players in playing X1 are injured or when you have to move-in a player in red hot form but don't have enough budget left, so we have to adjust some players.[11]

- We fetch the data of availability of a player playing next GW from FPL official API**[official API]** from the field **unavailable**. If a player is unavailable next GW, we mark his predicted scores as 0.

- figure 3 we see that Manchester united is losing matches back to back and is currently #10 in the points table. Managers have started transferring out these players. Similarly, Chelsea stands #1 in the points table and it's evident that managers have started swapping them in. In our predicted team, we can verify manager's sentiment with the transfer policy of our model. **Refer section [5]**

Evaluation parameter	**RMSE**
Prediction for all players	2.51
Per Predicted Teams	1.67
Per Selected Player	3.21

Table 2: RMSE values for predicted scores against actual scores

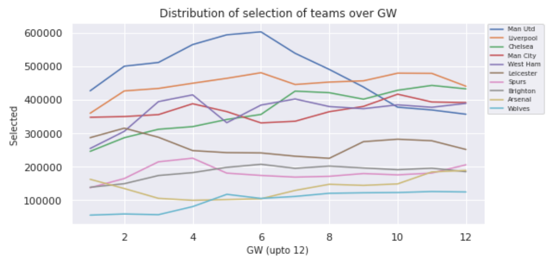

Figure 3: Distribution of players of each team selected by the managers every GW

5 Evaluation and Results Analysis

Prediction of the players' scores were obtained, from implementation of the final model, for the **GWs 1-12 for season 2021-22**. Since this is the ongoing season, it was selected for evaluation and validation since the statistics for the current trend in the scores and team points was available through official FPL sources. This also allowed us to expand the period of training data, spanning from season 2015 to 2020, providing better insights into feature selection and prediction.

The model predicted scores are evaluated against the actual scores using the metric **RMSE**. As tabulated in Table 2, the overall RMSE for the entire predicted scores of all players and the average RMSE for the team's predicted scores against actual is observed to be good and less than the RMSE for the prediction for all the players over entire GWs 1-12.

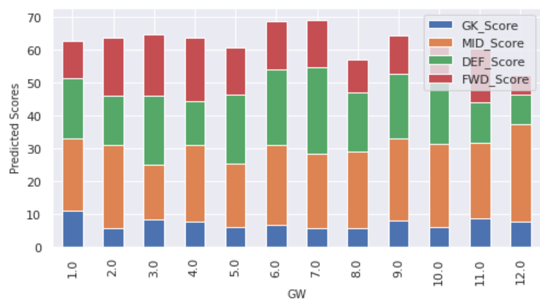

Figure 4: Contribution of Players per GW by Position

Algorithm 6 Optimized Transfer Strategy(Considering Penalty)

function TRANSFER($player, buyPrice, sellPrice$)
    $init : playerNames$
▷ *Name of players present in the 15 of prev GW*

    freeInDecision, paidInDecision, outDecisions, InDecisions, subDecisions, captainDecisions = IntilializeLPModel()
    $featureSet$ = {GW2 data with predicted points based on fixtures and wasHome}

    for player in {playerNames} **do**
        if player injured next GW:
            player[expectedPts] = 0
            ▷ *Player in XI but injured for next GW*
    end for

    InDecisions = freeInDecision + paidInDecision
    nextWeekSquad = squadDecision + InDecision - outDecisions
    starters = nextWeekSquad - subsDecision
    ▷ *calculate team for next GW with transfers*

    transferPenalty = sum(paidInDecision) * 4
    ▷ *Penalty for paid transfers*
    transferConstraints(inDecisions, freeInDecisions, outDecisions, budgetNow, buyPrice, sellPrice)
    ▷ *refer Algo 4*
    formationConstraints(nextWeekSquad, starters, subDecisions, captainDecision)
    ▷ *refer Algo 5*
    LpModel += starterPoints + captainPoints - transferPenalty
    LpModel.solve()

    return inDecisions, outDecisions, starters, subDecisions, captainDecision, LpModel.teamScore
end function

- The data for the predicted scores of players for GW 1-12 grouped by the position has been visualized in *Figure* 4. It is evident that the major contribution of the total points is from Mid Fielders and Defenders. This is also due to the constraint of having more players from these positions in the squad.

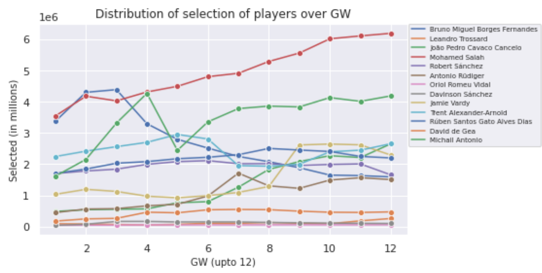

Figure 5: Distribution of Selection of Players over Game Weeks

- We also analysed the selection of certain top performing players[12] which were picked up in the predicted teams for multiple GWs and related it with the actual selection statistic from actual data. This is illustrated in *Figure* 5. This also shows that the model was able to pick the players which were amongst the popular picks in the GWs 1-12 for the current season 2021-22.

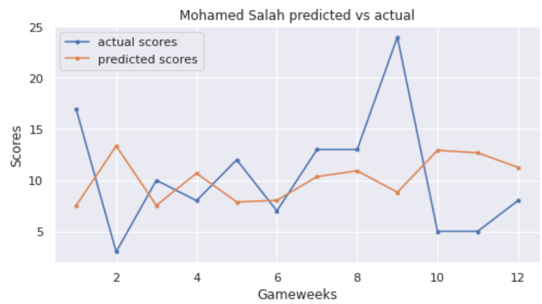

Figure 6: Comparison of Mohamed Salah's Predicted Scores against Actual Score

- Another analysis over the model predictions was proximity of the predicted scores and actual scores for the players. In *Figure* 6, we illustrate the predicted scores against actual for the GWs 1-12. The selection of this player is also quite significant over the game weeks. The model selected Salah as the captain for almost all of the 12 weeks. This also falls in line with the current season's performance of

Salah, suggesting that model performed well in selecting the optimum player squad.[17]

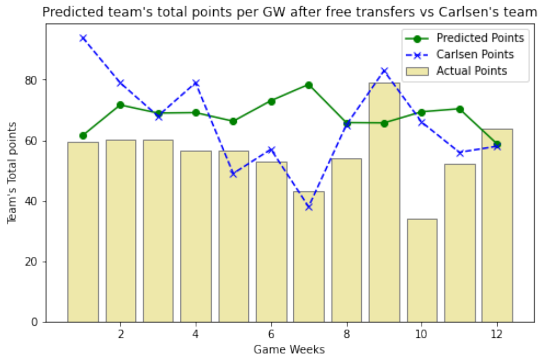

Figure 7: Comparison of Predicted Team's total points per GW after Free Transfers against Magnus Carlsen's Team

- One of the significant evaluation and analysis of the model performance was to compare the team's total points after transfers against the actual points. Also, as planned, we compared the predicted team's performance against renowned manager in FPL, Magnus Carlsen in *Figure* 7. As can be seen, the model predicted team's score was found to be quite close to Carlsen's team between GWs 5-9 for the current season 2021-22. **Refer to Appendix for Transfers.**

6 Conclusion

- Observed that prediction of FPL scores for the GWs is a difficult task, owing to the large number of features that impact the scores and the close correlation between them.

- Despite such complexities in the data and the problem statement, we were able to process the data and form a model, that showed good results on evaluation. Regardless, the scope of improvement would exists in terms of better understanding of impact of each feature and their prediction.

- The dynamic nature of the Football League and the external elements affecting the performance and outcomes, makes this problem highly complex. We found that although selecting a small set of features could bring the model closer to better predictions, however, it would be very difficult to weigh in each an every factor contributing to the feature values or overall scores.

- This problem relates to the hindsight optimization, wherein we optimize the model parameters based on events yet to occur. We believe that better model performance could be obtained possibly by using Deep Neural Networks.

7 References

[1] Datasets:fantasy.premierleague.com/api; https://github.com/vaastav/Fantasy-Premier-League

[2] William Eilertsen, Akash Gupta, Bjørn Kare Kristiansen, "Developing a Forecast-Based Optimization Model for Fantasy Premier League", Industrial Economics and Technology Management, June 2018

[3] Matthews, T., Ramchurn, S.D., Chalkiadakis, G.: Competing with Humans at Fantasy Football: Team Formation in Large Partially— Cbservable Domains. AAAI (2012)

[4] Kunj Mehta, Fantasy Premier League x Data Analysis: Being Among the Top 2%, https://towardsdatascience.com/fantasy-premier-league-x-dataanalysis-being-among-the-top-2-98a714a1d170, Feb 2021

[5] Dr Nick Barlow, Dr Angus Williams, A machine learning manager for Fantasy Premier League, https://www.turing.ac.uk/news/airsenal

[6] Nicholas Bonello, Joeran Beel, Seamus Lawless, Jeremy Debattista. "Multi-stream Data Analytics for Enhanced Performance Prediction in Fantasy Football". In 27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science. 2019

[7] Gupta, Akhil. "Time Series Modeling for Dream Team in Fantasy Premier League". International Conference on Sports Engineering ICSE 2017, 23-25 October 2017, Jaipur, India

[8] Schulze, Mark. (2000). Linear Programming for Optimization

[9] https://github.com/solpaul/fpl-prediction

[10] https://understat.com/team/Newcastle$_United/$2021

[11] https://fantasy.premierleague.com/api/bootstrap-static

[12] https://towardsdatascience.com/python-for-fpl-data-analytics-dadb414ccefd

[13] https://www.fplanalytics.com/PointsPGPMC.html

[14] https://fploptimized.com/ownership$_rates.html$

[15] https://fplreview.com/elite-data

[16] https://github.com/HaraldNordgren/betting-crawler

[17] https://www.fpl-data.co.uk/fixtures

Appendix

GW	Transfer In	Transfer Out	In(Cst)	Out(Cst)	In(xP)	Out(xP)	In(Pt)	Out(Pt)
2	M. Antonio	K. Iheanacho	7.6	7.5	9.17	3.2	16	1
3	Rúben Dias	Ben White	6	4.4	6	1	6	0
4	Paul Pogba	H. Barnes	7.6	6.9	5.8	2.34	9	2
5	Wan-Bissaka	C. Kouyaté	5.4	4.5	5.85	2.8	2	1
6	João Cancelo	Wan-Bissaka	6	5.7	9	2	12	2
7	L. Trossard	Paul Pogba	6.4	7.8	5.87	2.53	3	1
8	J. Veltman	D. Sánchez	4.4	4.5	6	1.93	0	0
9	João Filipe	A. Doucouré	5	5.7	5.33	2.2	2	0
9	*João Filipe(To XI)*	*J.Veltman*	*5*	*4.4*	*5.33*	*2.3*	*3*	*2*
10	N. Redmond	A. Westwood	5.9	5.3	5.49	2.38	0	2
10	*N.Redmond(ToXI)*	*R. Vidal*	*5.9*	*4.5*	*5.49*	*2.2*	*0*	*0*
11	Raúl Jiménez	M. Antonio	7.7	8.2	9	3.2	2	2
12	Tomas Soucek	João Filipe	5.5	5	7.66	2.2	2	3

Team transfers for subsequent GWs with free transfers over the predicted Team at GW1. Some players(at GW9 and GW10(marked in blue)) have been transferred from substitutes and again swapped from not well performing players in XI