CSE 519: Final Project Report - Fantasy Premier League
Better Late than Never - Suboptimal Player Transfer Strategy

Fall 2021, Stony Brook University

Anonymous

Abstract

The problem of determining the optimal game play strategy to win the FPL league is challenging considering factors like the order of player combinations for optimal team building and the uncertainties like injuries, penalties etc. In this project we relax the problem statement to a "better late than never" player transfer to suggest the next better player profile. An explainable "recruiter" model will use a lazy approach of finding a player profile which will suboptimally improve the current score. The profile is then used by a "scout" model to statistically estimate an actual player from rankings table. The recruiter model's "player selection polices" are evaluated using a reinforcement learning approach for cumulative points projection. We aim to analyze the criticality of decisions by simulating the projected scores of policies from the model and understand how they differ. We compare the decisions of top 5 , average 5 and bottom 5 ranking players. We analyze the common trends in transfer strategies of the projections. We also show that the learned actions by the model from top players can also be mimicked by bottom players to further improve their points.

1 Introduction

Fantasy Premier League is a complex strategy game with exponentially complex set of possible transfers and also possible team formations. The goal is to find player swaps that could lead to the maximum points for each gameweek of the season. There are approximately 650 Premier League Players and close to 8 million FPL Managers playing the FPL. Given each team is of 15 players, there are (15*650) possible 1 player swaps in a given gameweek. This means if we were to start from a single team, there would be (15*650) possible teams in the 2nd gameweek. For a season consisting of 38 gameweek, we have $(15 * 650)^{38}$ possible teams/possible paths for just one FPL Manager starting from gameweek 1. Generating all $(15 * 650)^{38}$ possible teams and calculating the scores of these teams in each gameweek is computationally infeasible as shown in Figure 1. Secondly, consider even the problem of finding the optimal team given a random team at start and you decide on making possible transfers to get to the optimal solution. The hypothetical dream team is the set of players where the cumulative sum of points per player per game week is maximum. But we could still have greedy mechanism of transfers right off the bat and still beat this dream team. Thus, this problem is not trivial. There in comes the need of finding the optimal team transfer strategy by considering all possible transfer trajectories. We relax this problem statement by approximations using a neural network. Thus, we have chosen the pipeline analysis as an experiment in Section 4.2.

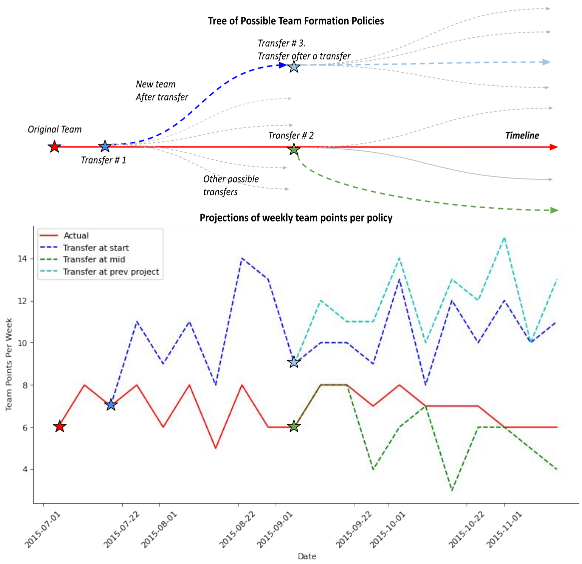

Figure 1: Point Simulations of suggested policies and Tree Space of Possible Policies

Even techniques like player point prediction can incur high degree of variance considering there are non quantifiable factors like injuries which disrupts the scoring system. Linear models and decision trees can prove to be ineffective and thus we require non linear modelling approaches like RNN,LSTM or RL based neural networks.

The objective is to identify the causal effects of decisions or strategies. In the real world, football team managers consider numerous parameters to buy in or sell players like team balance, fill in an open player type position or player attributes. But the driving force is always to maximize gain league points and rank at the top of the leaderboards. The FPL managers are similar in this approach and look at actual gameplays to come up with decisions. In this project, we aim to find such optimal decisions.

2 Related Works

In the article (2) they try to identify the relation between the statistics of the player on the field and the actual FPL Points scored. To what extent can the underlying statistics predict the player's weekly performance. To analyze the impact of fixtures on player's FPL performance. Identifying ways to create the optimal team depending on the value of the player and other alternative player options in the same position. The main approach here was applying linear regression on a group of underlying statistics and FPL Points and then finding correlation between the feature and FPL points.It was found that goals conceded, saves, touches and recoveries were the strongest predictors of FPL points. The analysis to determine the best way to choose a team was done on the basis of value of players in each category and the impact of fixtures on categories.From the analysis the author suggests to invest more in expensive midfielder and defensive positions as rather than pricier goalkeeper and forwards as for the later the value for money is lower relative to their less expensive counter parts.

In the article (4), they mention new statics like 'expected goals (xG)' to make player selection. Another is 'Average Formation', which will track the positions of all players when a team is in and out of possession. Other few are 'Live win probability' , 'momentum tracker'.

In the paper (5), they suggest a new method for predicting future player performances by automatically incorporating human feedback into our model so as to handle uncertainties like as injuries, managerial decisions. There are online tools like (6), (7) and available APIs to access datasets and get statistical inferences.

2.1 Reinforcement Learning

The FPL game week data is turned into a simulator where we use it to calculate the current state of a team and get the points of a team at a particular game week. In the course content (3), a Monte Carlo policy iterative method can be used to train a neural network which will learn the optimal policy.

3 DataSet

The FPL data of the current season(10 GameWeeks) was fetched via the FPL API-

3.0.1 WeekWise EPL Player Stats

This data was fetched using the endpoint (8). This returned the stats of all actual EPL players in the ith week. The stats include - minutes played, goals scored, assists, ICT index, total points, etc. We have a CSV for all players at each week of the current season.

3.0.2 Weekwise EPL Player Cost

For all the EPL players, the cost of that player with ID 'i' in the given week was fetched using endpoint (9). The cost for all players for all weeks is then merged into a single dataframe where the rows denote the week number and columns denote the player id. The matrix values denote the player cost.

3.0.3 Weekwise FPL Manager Stats

The leader board of all FPL Managers in a given league was fetched using endpoint (10). Using the id's of the FPL managers, their stats and team at each week was fecthed using endpoint (11). The stats include the current week points , total points, current rank, previous week rank, budget of the FPL manager for the current week.

3.0.4 EPL Player Static Stats

The static stats for an EPL player was fetched using endpoint (12). It returned - total points, yellow cards, red cards, percent selected, in dreamteam, value form, value season, etc.

4 Implementation

4.1 Player Embeddings

A player profile basically comprises of a subset of EPL player attributes like - goals scored, goals

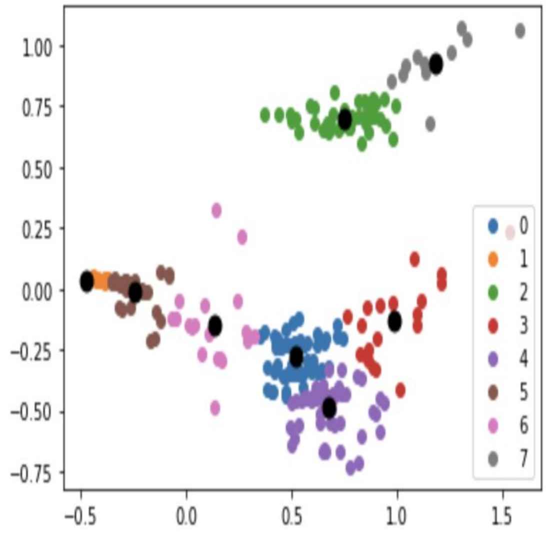

Figure 2: Generating Player Profiles Using K-Means Clustering

conceded, yellow cards, red cards, ICT index, selected percent, minutes played, etc. We use a total of 27 attributes for each EPL player and the goal is to come up with different profiles i.e. different subsets of these attributes. For instance, the profile of an attacker would have high values for ICT index and goals scored whereas the profile for an defensive player would have lower values for goals conceded and own goals. Similarly, there can be multiple profiles corresponding to an experienced player, fair player, popular player among the FPL managers ,etc.

The recruiter model spits out a player profile that it thinks will improve our score in the next week. This profile is then used to get a set of k candidate players from a pool of 700 players where 1 of them will be transferred in based on constraints like cost, min balance, etc. The player swap will be discussed in later sections of the report. This section will focus on coming up with the 'player profiles/embeddings'. The following two approaches were used.

4.1.1 Profiles by Clustering

The idea here is to cluster the players using their attributes into k clusters where each cluster corresponds to a player profile. It is implicit that each EPL player can belong to exactly 1 cluster or have only 1 profile.

K-Means clustering algorithm was used to group the players into K clusters. However, this algorithm cannot be applied directly on the raw player attributes as clustering algorithms such as KMeans have a difficult time accurately clustering data of high dimensionality (ie. too many features). It uses a distance formula (ie. Euclidean distance) to determine cluster membership. When our clustering algorithm has too many dimensions, pairs of points will begin to have very similar distances and it wont be possible to obtain meaningful clusters. As a result, we use PCA / t-SNE to reduce the number of dimensions and then use k-means clustering. However, even this linear/non-linear transformation didnt yield meaningful clusters. Despite having an average silhouette score of 0.68 for various cluster sizes, a deeper analysis of clusters and its players didnt reveal anything meaningful in terms of what kind of profile it actually was. Multiple clusters had overlaps in terms of profiles and interpretation of difference between profiles became an issue. Having profiles with overlap creates problems for our model which needs to try different actions(i.e. player profiles) for learning.

4.1.2 Synthetic Player Profiles

To ensure interpretability of player profiles and minimizing overlap in player profiles, we decided to form the player profiles manually using domain knowledge and statistics. We came up with a few draft profiles. A few of them as follows:
1. Attack Skills - goals scored
2. Experienced Player - more minutes on pitch, influence
3. Defense Skills - low goals conceded ,low own goals
4. High Creativity - assists , creativity
5. High Threat - attempts on goal
6. High Influence - on a game or season
7. Good Sportsmanship - low red cards, low yellow cards
8. Most Popular Selection - selected percent
9. GoalKeeper - saves, clean sheets, penalties saved, low goal conceded ,low penalties missed
10. Penalty Specialist - low penalties order, goals scored

We used these player profiles to sort the dataset of EPL player on the attributed and returned the top k candidate players. We compared the overlap of candidate players of one profile with all other profiles and tried to reduce the overlap by group attributes that are correlated to each other and separating the attributes that are least correlated into different profiles. Using a few iterations got

us a list of 5 profiles using 11 player attributes that had the minimum overlap. The refined profiles are:
profile 0: minutes
profile 1: own goals, yellow cards, red cards
profile 2: ict index
profile 3: selected by percent
profile 4: saves goal conceded ratio, saves, clean sheets, penalties saved, penalties missed

4.2 Pipeline

4.2.1 Simulator

The simulator is a static module which computes the new team, its score, and the latest cost based on the transfered in and out players. The simulator stores all high level variables and keeps track of the running player ids of the team after every transfer. The balance is also tracked. The Simulator does the actual transfering in and out of players, where it takes in an actual team of an FPL manager t and outputs a new team t' by swapping in the strongest player p_{in} and transferring out the weakest player p_{out}.The transferred out player p_{out} and transferred in player p_{in}n is chosen by the scout and recruiter modules as described below. This is done for every game week.

4.2.2 Scout

The scout model is responsible for determining the next best player swap. The model takes in a set of transfer in and transfer out player candidates and determines which player out of the transfer out player needs to be swaped out and which player among the transfer in players needs to be swapped in. This happens in 3 steps : 1. finding the strong candidates from the pool 2. finding weak candidates from current team 3. matching these two set of candidates based on budget and player type to get (p_{in}, p_{out}).

1. Finding Weakest players to Swap out

   The weakest players are determined by their Return on Investment(ROI). ROI is the points of the player divided by the cost of the player. For every gameweek the players are ranked according to the ROI and the weakest k players from the rankings are selected

   Here team is a list of all players in an FPL Manager's team. The top k sorted players depending on ROI are returned ,i.e the k players with the lowest ROI.

procedure TRANSFEROUT($team$)
  for $p \in team$ **do**
    $ROI \leftarrow \frac{playerpoints}{playercost}$
    $sort(ROI)$
    return $ROI[:k]$

2. Finding Strongest players to Swap in

   The recruiter model gives the player profile to the swap in function and a list of strongest players matching the corresponding profile is returned.
   A player profile consists of specific features such as own_goals ,saves ,clean_sheet and order in which the features are to be ranked either ascending or descending

procedure TRANSFERIN($pidx$,$tout$)
  $ft, order \leftarrow profiles[pidx]$
  $sortByFtAndOrder(allPlayers)$
  return $allPlayers[:k] \notin tout$

   Here pidx is the profile index(specifing the player profile), ft are the features and order is the sorting order of the particular profile and tout is the list of players to be swapped out(those players are excluded)

3. Determining the player to swap in and swap out

   In every gameweek there is a starting balance available with the FPL Manager before swapping out any player. The balance amount by swapping out a player is calculated by adding the player's cost to the current gameweek balance. To find the swap in player the cost of the strongest players is subtracted from the balance after selling the weakest player and the element type of the strongest and weakest players is compared. The swap in and swap out players need to have the same element type. The above conditions are checked for every k^{th} weakest player with every k^{th} strongest player and the 1st strongest player which satisfies the conditions is returned as the player to be swapped in.

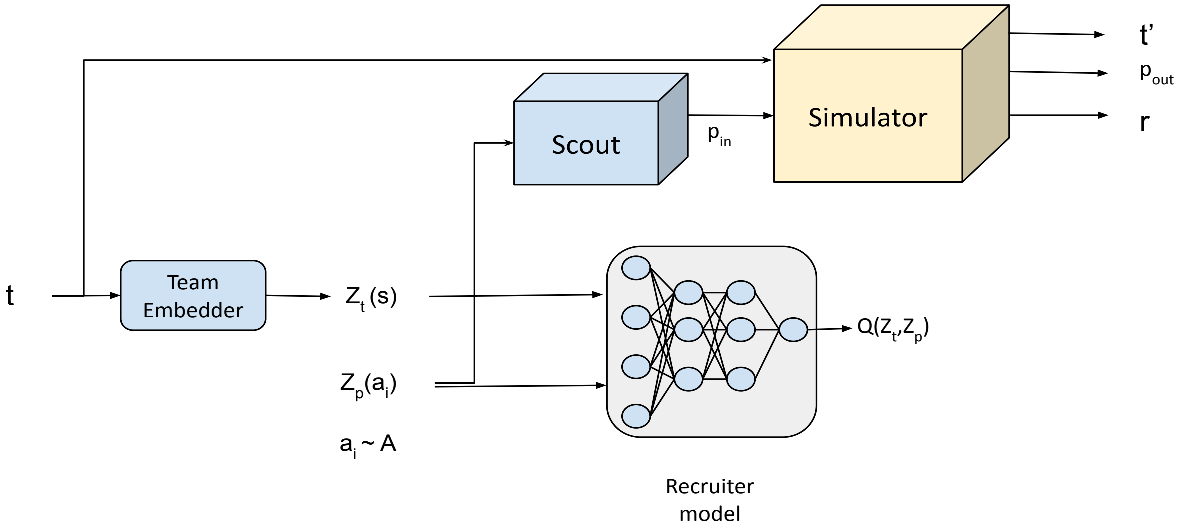

Figure 3: Flowchart of the project pipeline

procedure FINDSWAPCHOICE (tin,$tout$,bal,th,$cost$)
  for $pout \in tout$ **do**
    $bal \leftarrow bal + cost[pout]$
    for $pin \in tin$ **do**
      $rem \leftarrow rem - cost[pin]$
      if $(rem \geq th)$ and $(pin : type == pout : type)$ **then**
        $swapchoice = (pin, pout)$
        return $swapchoice$
      end if
    end for
  end for

Where tin is the list of strongest player ids to swap in, tout is the list of weakest player ids to swap out and Cost is the player's cost per gameweek. bal is the the FPL Manager's balance and th is the minimum amount threshold that needs to remain with the FPL manager. The returned swapchoice has the ids of the player to be transferred in(pin) and transferred out (pout)

4.2.3 Recruiter

We use a 3 layer feed forward fully connected neural network. The network takes in two inputs which reinforcement learning terms are the state and action. The state is Z_t which is just the team embedding, and action is the player profile. Before training , we assume the following:

1. Teams formed through player transfers follow a Markov chain. 2. We use a Monte Carlo approach to evaluate expected gains at each state. We simulate episodes where each episode is a set of transfers happening until termination (budget runs out or end of game week). Each transfer is deemed to be good or bad in terms of a reward $r = 1, 2 \ \ if \ \ p_{in}.points >= p_{out}.points \ \ else \ \ 0, -1, -2.$ The expected gain is $E[G_s|S = s]$ and $G_s = R_s + \gamma R_{s+1} + \gamma^2 R_{s+2} + \gamma^3 R_{s+3} + ... + \gamma^W R_W$. W is the number of game weeks and γ is a dampening factor which can tell neural network to focus on short term or long term rewards. The most important aspect about the recruiter is the reward function where the model is rewarded from a set $\{-2, -1, 0, 1, 2\}$ based on difference between the points of new team and the actual team. The output of the NN is the $Q(s, a)$ value which tells how good a state is. While training, we use an loss $L = MSE(Q(s, a), E[G_s|S = s]$. This will allow us to estimate goodness of states by learning from experience. We just show enough simulations using player transfers so that the NN has a good enough understanding of the team formation space.

One model is trained per manager. Each model is trained for 10000 episodes where each episode is a simulated set of transfers suggested by the model. When training the model starts off with high exploration it will hit minimas of low rewards from poor transfers. Gradually, the model is optimized with decaying epsilon(exploratory factor) and slowing boosting greedy approach. The final trained model will be able to make transfers taking into considering the future impact of current transfer over later weeks. For example, transferring in a high scoring and costly player early in the season can close off our options to buy possibly affordable and potential players in the later weeks. This a major flaw of the greedy approach where the decision making is myopic and looks for high rewards over short term. Our trained model looks for rewards in the long term.

5 Experiments

The experiments are designed to give insights much deeper than an exploratory data analysis. The set of experiments will be analyzed for 15 managers of 3 ranking ranges - top , average and bottom. We set out to see the set of model suggested transfers, how these transfer sequences are similar / different, and if we can transfer knowledge from top to bottom players and vice versa. This questions the model robustness and validity of transfers. Assertion checks for valid transfers are in place for model authenticity. The top manager IDs used here are : '2757', '4424129', '2963971','3720286','799571'. The average manager IDs used are : '492','494','495','498','499'. The bottom ranking manager IDs used here are: '992','993','994','995','1000'.

5.1 Experiment 1 : Interpretation of sample transfers

As stated in the algorithm, the recruiter model is trained per manager and we would have a set of 15 models. Here the model robustness is tested by comparing the projected points from 3 different approaches- naive , random and recruiter model. The naive is the strictly greedy approach where the best players at every game week are transferred into the team (with consideration of cost). The random approach does random transfers at every game week. The recruiter is the trained model which suggests intelligent transfer per game week. For each manager we have the points per week (PPW) and the cumulative sum of points (CP) for each manager for every game week. The PPW and CP are available for the naive, random and the recruiter model. The figure 4 shows the results for this experiment. In conclusion the final points of the team formed by the recruiter model is better (sub optimal) than all other methods even the actual score of the team at the end of the season as seen in table 1.

	Random		Naive		S-R	
Id	Actual	Predicted	Actual	Predicted	Actual	Predicted
2757	588	520	588	525	588	624
4424129	597	550	597	547	597	615
2963971	531	522	531	469	531	652
3720286	449	434	449	462	449	648
799571	457	401	457	458	457	575
492	629	520	629	493	629	596
494	597	525	597	566	597	638
495	568	421	568	502	568	626
498	595	542	595	547	595	621
499	606	563	606	488	606	662
992	636	625	636	605	636	725
993	630	601	630	545	630	625
994	642	567	642	574	642	690
995	573	486	573	509	573	632
1000	632	599	632	561	632	671

Table 1: The table shows the projected points at the end of 10 game weeks for each manager. Each manager will have actual points (from his actual team), random , naive approach and the scout-recruiter model

5.2 Experiment 2 : Manager vs Manager - Trend Analysis of Projected Transfers

This experiment is an analysis of the common trends in transfers that were suggested by the model. Here we compare the transfer set for the 10 game weeks for each of the managers and find the common matching subset of transfer. Statiscally most FPL players would opt to buy in well performing players like Ronaldo, Messi or Salah into their FPL team. These players are costly and subject to the budget. But we would like to analyze what is the popular strategy that FPL managers can take to improve their score.

The model suggests transfer-in of players with profiles 2 and 4 in most of the game weeks.
profile 2: ict index
profile 4: saves goal conceded ratio, saves, clean sheets, penalties saved, penalties missed
This suggests that a player with (1) higher attempts on goal, higher number of assists and higher influence on a game/season(this can be a forward or a midfielder) and (2) higher number of saves per goals conceded ratio, penalties saved, clean sheets (this is a goalkeeper)is more likely to fetch more points for the team as suggested by the model.

The other profiles that arent often recommended by the recruiter suggest that the players with a high

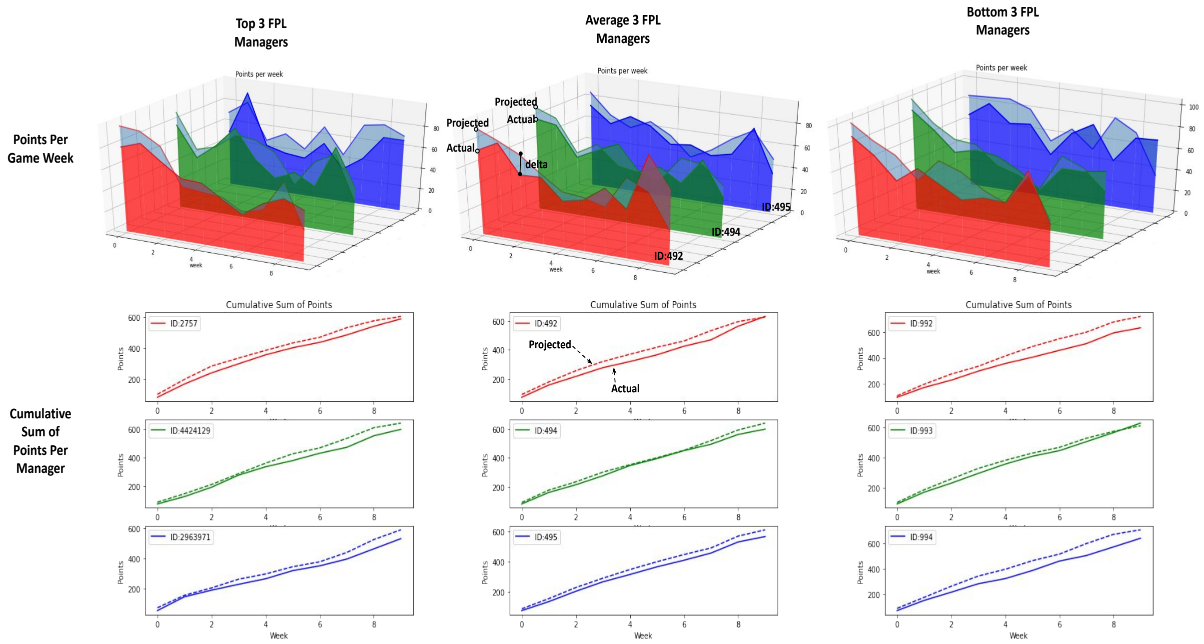

Figure 4: Experiment 1: Points per week (PDF) and cumulative points (CDF) of 3 category of managers. The 3D plot is just 3 2D plots of points per week plots stacked over one another. In one 3D plot there are 3 managers. Each manager PPW and CP are shown by color. Each manager has an actual and model suggested projection as shown in the plots. For all 9 managers the final team formed by model suggested transfers were the best as seen from the cumulative sum format.

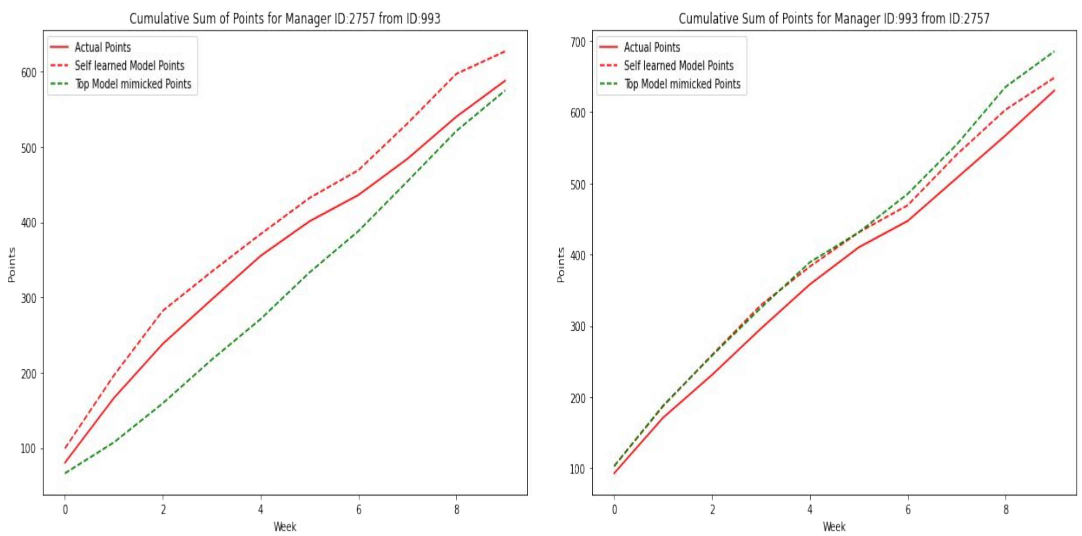

Figure 5: Experiment 3: The cumulative sum plots for top to bottom (right plot)and bottom to top (left plot) behaviour transfer. This shows that the actions took by the top player model is even better than the model trained on the bottom ranking player and has even better score than the self learned model. However, this is opposite for the bottom to top where the model is even worse than the actual points of the managers team.

’selected by percent’ number, only more minutes on the pitch and low number of fouls rarely provide any advantage in terms of score.

Another interesting observation is that the model recommendation of player profiles is consistent across the top 5, bottom 5 and random 5 FPL managers.

5.3 Experiment 3: Behaviour Transfer

This is an interesting experiment where the learned transfer strategy knowledge from a better player (top 5 FPL managers) are applied to the bottom managers. This is a usecase where we wish to see if inexperienced or poor performing players can learn from the gameplays of better players. So, for this we have two cases: top to bottom and bottom to top. The top to bottom approach has a top player (ID:2757) and a bottom ranking player (ID:993). There are two models trained on the these two player simulations and has correspondingly optimized the projected scores for both players as shown in 4. In the top to bottom approach, we use the learned model of the top ranking player to suggest transfer for the bottom ranking player. We observe, these actions are indeed much profitable and yields more points than before as shown in 5. The vice versa is not true but is as expected. A top ranking player learning from a bottom ranking player is bound to suffer and is reflected in the final team points shown by 5

References

[1] Richard S. Sutton and Andrew G. Barto. 2018. Reinforcement Learning: An Introduction. A Bradford Book, Cambridge, MA, USA.

[2] https://mathematicallysafe.wordpress.com/2019/07/01/fpl-analysis-what-five-seasons-of-data-modelling-have-revealed-about-predictive-analysis-fixture-impact-and-optimal-team-structure-in-fantasy-premier-league/

[3] CSE 525: Robotics, Stony Brook, Micheal Ryoo, http://michaelryoo.com/course/cse525/

[4] https://www.forbes.com/sites/stevemccaskill/2021/05/06/premier-leagues-new-ai-match-stats-will-predict-if-a-team-will-score-in-the-next-10-seconds/?sh=5f06318a6084

[5] Multi-stream Data Analytics for Enhanced Performance Prediction in Fantasy Football, Bonello et al,https://arxiv.org/abs/1912.07441

[6] FPL analytics , https://www.fplanalytics.com/

[7] FPL statistics , https://fantasy.premierleague.com/statistics

[8] FPL player weekwise data, https://fantasy.premierleague.com/api/event/{i}/live/

[9] FPL player weekwise cost data, https://fantasy.premierleague.com/api/element-summary/i/

[10] FPL Manager League Standings, https://fantasy.premierleague.com/api/leagues-classic/1/standings/?page_standings=1

[11] FPL Manager Weekwise Team data, https://fantasy.premierleague.com/api/entry/{manager_id}/event/{week}/picks/

[12] Bootstrap Static Data, https://fantasy.premierleague.com/api/bootstrap-static/