Fantasy Premier League: Point Projections and Team Optimization

Cooper Cunliffe

According to Orbis Research, fantasy sports generated almost $14 billion in revenue in 2018 alone and is expected to grow to $33 billion by 2025. Most fantasy sports follow a similar template: select a ‘team’ of professional athletes that you believe will perform well over the course of a season. If the players you select perform well in real-life games, your ‘team’ is awarded fantasy points. Fantasy sports exist for all of the major sports leagues in the US, including the NFL, MLB, NBA, and NHL. Some fans just play for fun, while others play for cash rewards.

Everyone who plays fantasy sports is able to benefit from the many websites and magazines that offer professional opinions and historical statistics to guide your choices. Some of these sites offer free content and interactive tools, but most of them also offer ‘premium’ products for a fee. There is one thing that all of these publications have in common, though, across every fantasy sport. They are all trying to answer these two fundamental questions:

What are the future fantasy points of each player?

and

Which players should I choose for my fantasy team?

This project aims to use a statistical approach in order to answer both of these questions for the official fantasy game of the British Premier League. There are currently over 7,000,000 Fantasy Premier League (FPL) teams each trying to accomplish the same thing: score as many fantasy points as possible over the course of a season. Everyone chooses a squad of 11 players (plus four substitutes) that they believe will perform well in their next game and throughout the season. Fantasy teams receive points if the players in the team are credited with goals, assists, minutes played, and a few other measures. Each player is assigned a dollar value at the beginning of the season by the creators of the game, and you are given $100 to spend on your team. Finally, you are only allowed to select at most three players from a single Premier League team, and your starting 11 players must include:

- Exactly 1 goalkeeper
- Between 3 and 5 defenders
- Between 2 and 5 midfielders
- Between 1 and 3 forwards

After **collecting player data** from the FPL API, **machine learning** techniques will be applied to predict fantasy points for each player. These predictions will then be fed into a customized Mixed Integer Linear Programming **(MIP) solver** in order to choose the optimal team based on these predictions.

Data Collection

The offcial FPL game graciously offers an undocumented API with a handful of endpoints:

```
# https://fantasy.premierleague.com/api

# /bootstrap-static/
# /regions/
# /fixtures/
# /fixtures/?event={eventId}
# /entry/{entryId}
# /entry/{entryId}/history
# /element-summary/{playerId}/
# /me/
# /leagues-classic/{classicLeagueId}/standings
# /leagues-h2h/{h2hLeagueId}/standings
```

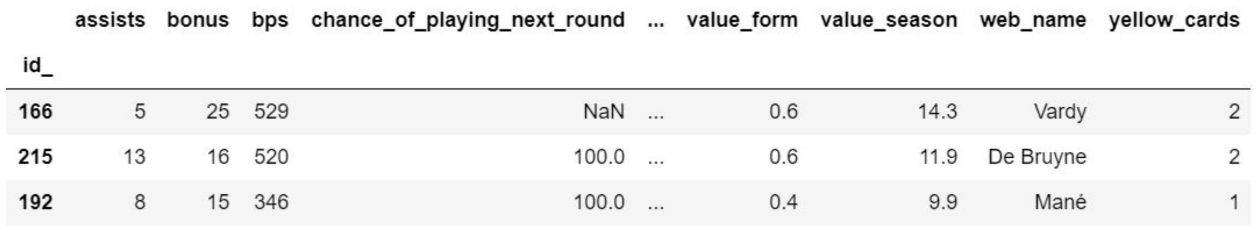
From `https://fantasy.premierleague.com/api/bootstrap-static/` we can use the requests package to access the JSON containing the list of dictionaries with accumulated data for each Premier League player. Then this list is turned into a Pandas DataFrame.

```python
import pandas as pd
import requests
url_static = 'https://fantasy.premierleague.com/api/bootstrap-static/'
r_static = requests.get(url_static)
data_static = r_static.json()
elements = data_static['elements']
df = pd.DataFrame(elements)
df = df.set_index('id')
```

Now sort the DataFrame by `'total_points'` to see the highest scoring players first:

```python
top500 = df.sort_values(by='total_points', ascending=False).iloc[:500,:]
top500 = top500.rename_axis('id_')
```

Here are the first few rows of `top500`:



	assists	bonus	bps	chance_of_playing_next_round	...	value_form	value_season	web_name	yellow_cards
id_									
166	5	25	529	NaN	...	0.6	14.3	Vardy	2
215	13	16	520	100.0	...	0.6	11.9	De Bruyne	2
192	8	15	346	100.0	...	0.4	9.9	Mané	1

Each row represents a different player and each column refers to a particular attribute.

Here are all the columns:

assists	news_added
bonus	now_cost
bps	own_goals
chance_of_playing_next_round	penalties_missed
chance_of_playing_this_round	penalties_saved
clean_sheets	photo
code	points_per_game
cost_change_event	red_cards
cost_change_event_fall	saves
cost_change_start	second_name
cost_change_start_fall	selected_by_percent
creativity	special
dreamteam_count	squad_number
element_type	status
ep_next	team
ep_this	team_code
event_points	threat
first_name	total_points
form	transfers_in
goals_conceded	transfers_in_event
goals_scored	transfers_out
ict_index	transfers_out_event
in_dreamteam	value_form
influence	value_season
minutes	web_name
news	yellow_cards

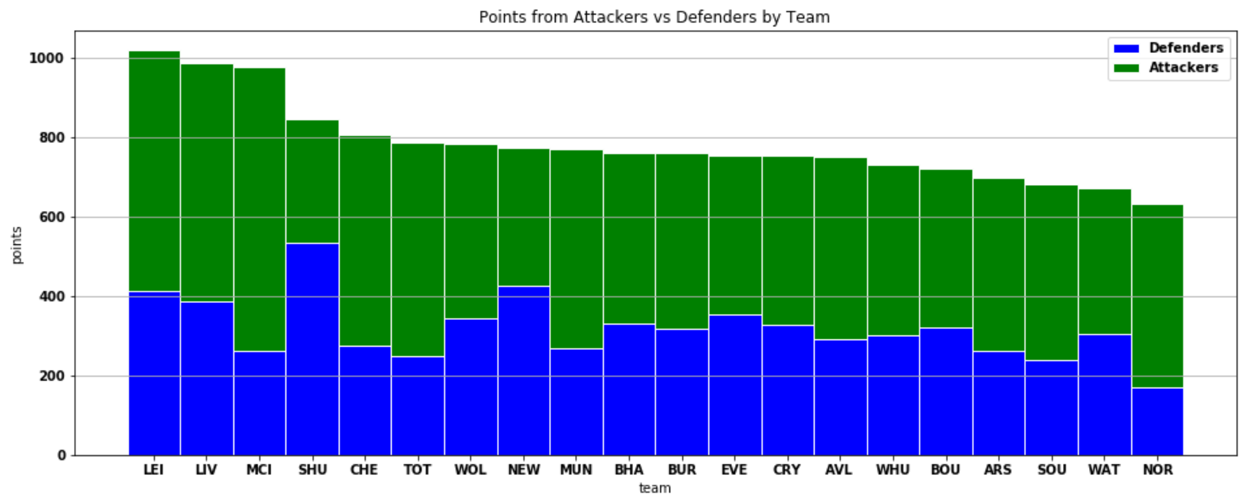
Total Points Preliminary Visualizations

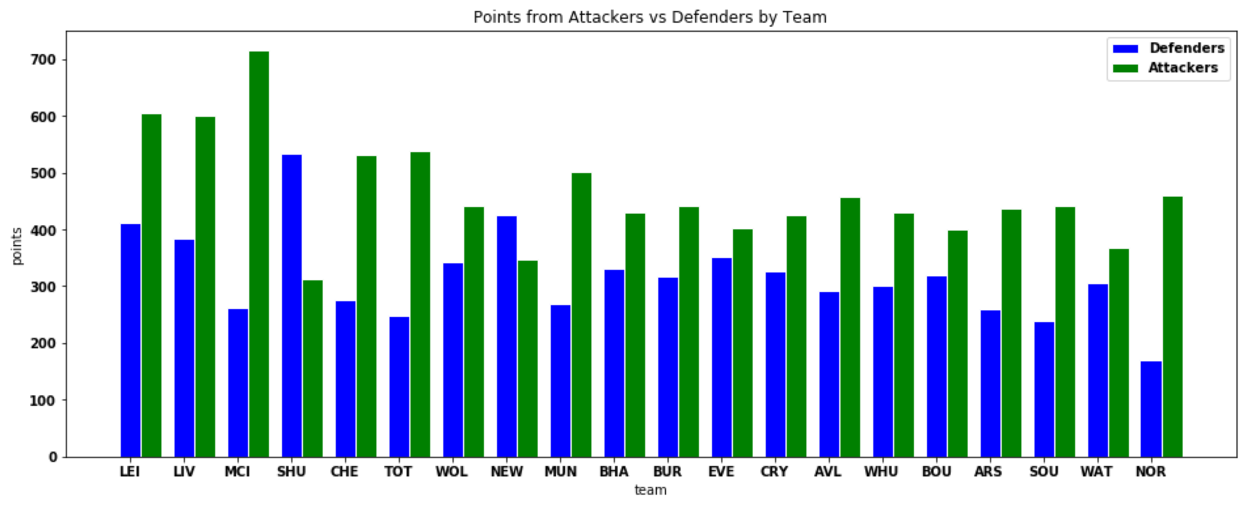
`'total_points'` in this DataFrame represents the total accumulated points so far this season for each player. Since each fantasy team is trying to maximize the fantasy points of the players on their fantasy team, it would be valuable if we could predict future points based on previous performances.

The best players in the Premier League tend to gather the most fantasy points, which are roughly calculated based on the following formulas:

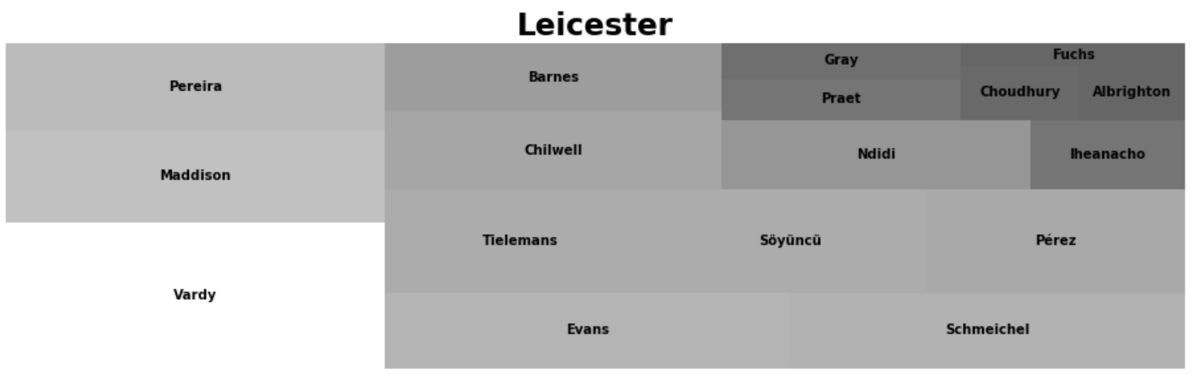
forward points = assists * 3 + goals * 4 + bonus
midfield points = assists * 3 + goals * 5 + clean_sheets + bonus
defender/keeper points = assists * 3 + goals * 6 + clean_sheets * 4 + bonus

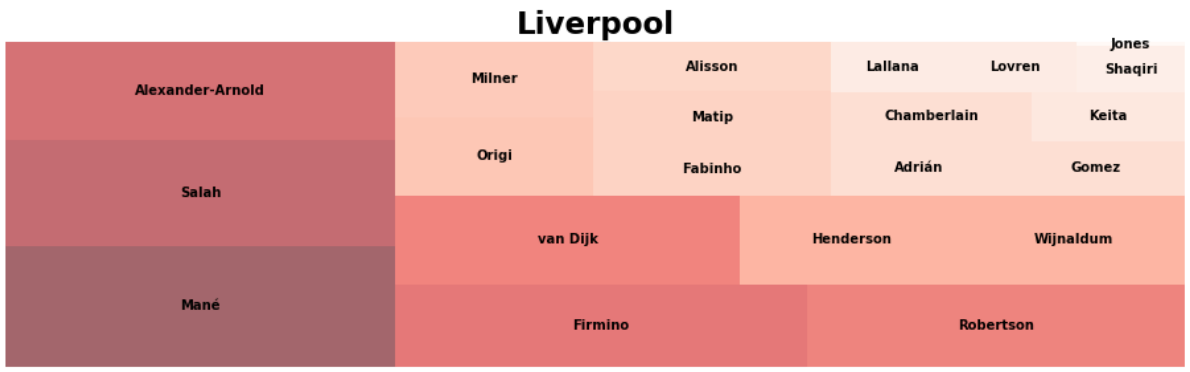
Let's see how attackers and defenders contribute to the total fantasy points of each Premier League team:

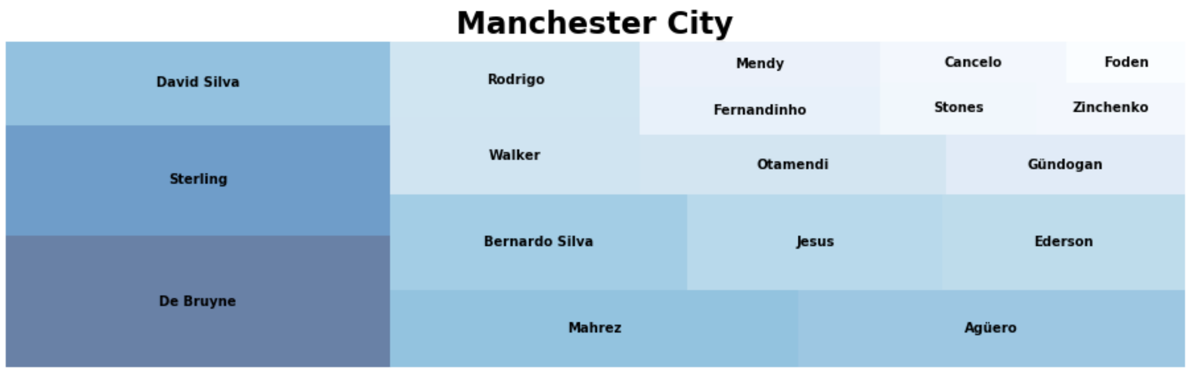




It looks like Leicester, Liverpool, and Manchester City are in a class of their own when it comes to producing fantasy points. Let's take a look at the individual contributors on each of these teams:







The biggest contributors on the highest-scoring Premier League teams may be good targets for our fantasy team selection. However, top players on the best teams tend to be the most expensive in the fantasy game and these prices must be taken into account if we plan to build any kind of meaningful 'Dream Team' (unlike the official Fantasy Premier League Dream Team, which simply lists the game's highest scoring players and results in an unattainable team, well over the maximum spending limit.)

https://fantasy.premierleague.com/dream-team/

Prediction Data

If we hope to make any meaningful fantasy point predictions, we need to collect the game-by-game player data describing how the players performed in each game so far this season. This information is stored in the `# /element-summary/{playerId}/` API endpoint. The `{playerId}` corresponds to the index of the `top500` DataFrame and represents a unique identifier for each player. After pulling the JSON for each player and converting to DataFrames, several columns are added to each one. Finally, all 500 DataFrames are concatenated into a single file, `top500histories`, and narrow our attention down to the following columns:

assists	saves	transfers_out
bonus	selected	id_
bps	team_a_score	sum_points
clean_sheets	team_h_score	sum_minutes
goals_scored	total_points	team_a_conceded_rank
minutes	transfers_in	points_per_90
round		next_week_points

Each row of this DataFrame represents an individual Premier League player's performance in a single Premier League match. `'total_points'` in this case refers to the total points achieved for that player in that specific game. `'next_week_points'` represents the total fantasy points that the same player received in the following week, and will be the feature we try to predict using both a **linear regression model** and a **random forest ensemble**.

Selection Optimization: MIP

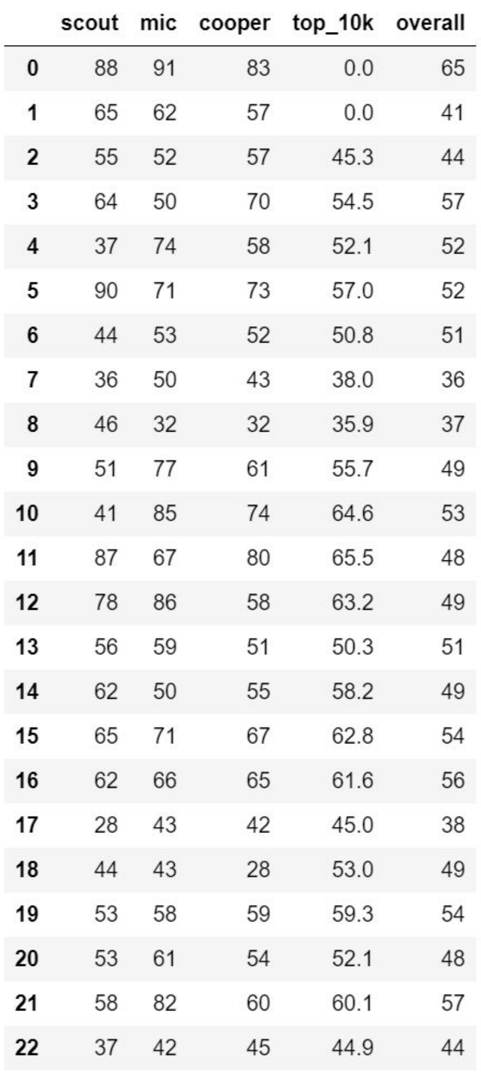
Finding useful predictions for each of the Premier League players is only half the story. How can we use these predictions to select an optimized team of 11 starting players? It would be nice if we could simply select the 11 players with the highest expected future points, but this would almost surely put our team over the $100 maximum budget.

In order to maximize the total expected points of our starting 11 players without going over budget, we'll use a **Mixed Integer Programming (MIP)** solver. The Python-MIP package provides the tools needed to define the fantasy team selection process as an optimization problem while also coding constraints to follow the rules of the game. First, a model is created that aims to maximize the sum of the predicted points for a list of players. Capacity constraints are then added to ensure that the model chooses a valid team:

- Total price of all selected players does not exceed the max budget
- Maximum of 11 players are chosen
- The proper number of each position is selected
    - 1 keeper, 3-5 defenders, 2-5 midfielders, 1-3 forwards
- Maximum of 3 players chosen from any individual Premier League team

The model accepts two lists in order to choose an optimal team. The first is a **profit** list, and the second is a **weight** list. The weights will always refer to the player prices, and the model will try to optimize the 'profit' without going over the maximum 'weight.' The 'profit' will represent our list of predictions for each player, and the model will try to maximize the sum of the predictions while still staying under budget. The output of the model is a list of players that represents the maximum possible profit sum while staying within the indicated constraints.

We can gauge the success of our prediction models by examining the teams chosen by the MIP solver when different predictions are used as the profit list. Let's start by building a DataFrame with some real-life fantasy point totals:



	scout	mic	cooper	top_10k	overall
0	88	91	83	0.0	65
1	65	62	57	0.0	41
2	55	52	57	45.3	44
3	64	50	70	54.5	57
4	37	74	58	52.1	52
5	90	71	73	57.0	52
6	44	53	52	50.8	51
7	36	50	43	38.0	36
8	46	32	32	35.9	37
9	51	77	61	55.7	49
10	41	85	74	64.6	53
11	87	67	80	65.5	48
12	78	86	58	63.2	49
13	56	59	51	50.3	51
14	62	50	55	58.2	49
15	65	71	67	62.8	54
16	62	66	65	61.6	56
17	28	43	42	45.0	38
18	44	43	28	53.0	49
19	53	58	59	59.3	54
20	53	61	54	52.1	48
21	58	82	60	60.1	57
22	37	42	45	44.9	44

scout: The Scout is a professional blogger that selects a dream team each week. The scores seen in this column represent what the team would have achieved that week.

mic: My brother, FPL extraordinaire. Having a fantastic season so far and well within the top 1% of all teams.

cooper: My personal FPL team. Started the season well but struggling as of late. Still well within the top 5% of all teams.

top_10k: The average score of all the teams ranked inside the top ten thousand each week.

overall: The average score of every fantasy team each week.

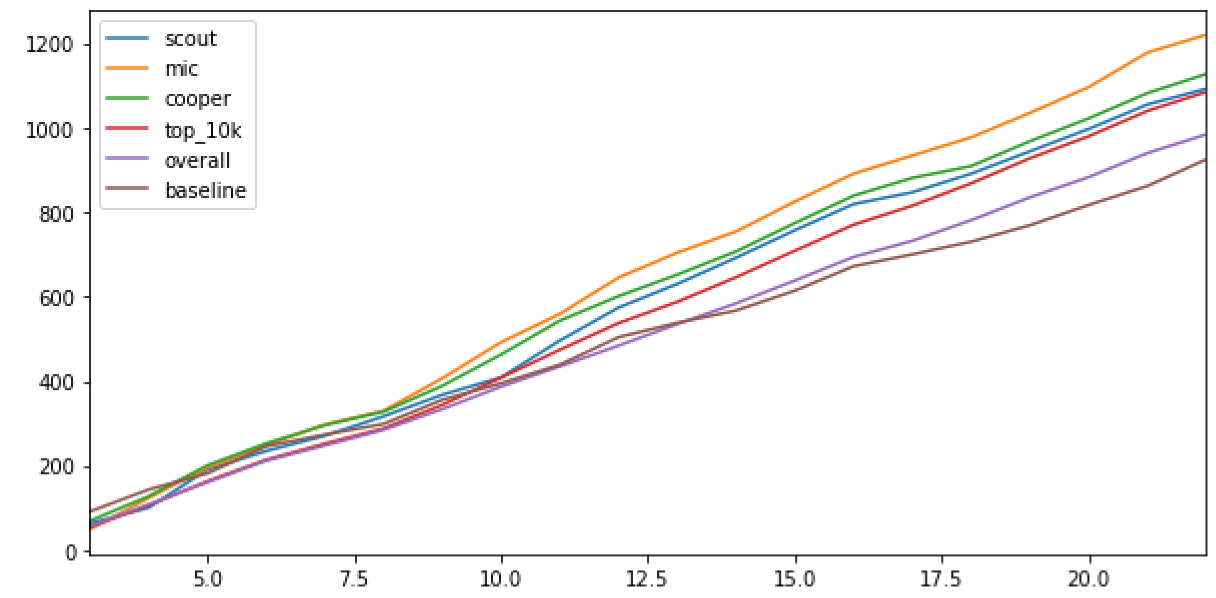
Baseline Predictions

In order to better understand the success of our machine learning models, let's first build a simple, naive set of predictions that we can later compare our ML models with. We can call this the baseline model. The predicted score for each player for the next week will simply be the score they received in the current week; that is, we will predict repeat performances for all players.

For each gameweek, we'll rank the players according to their current week points, and then, using the MIP solver and staying under a $100 budget, we'll choose a starting 11 and a 4 player bench while taking into account the current week's player values. Finally, we add up the points that this team would have received the following week (captain goes to the player that had the highest predicted points.)

Unsurprisingly, The baseline teams perform significantly worse than experienced humans. However, it isn't too far off the overall average in the same time period.

Fantasy Points through week 23



mic	**1220.0**
cooper	**1127.0**
scout	**1092.0**
top_10k	**1084.6**
overall	**984.0**
baseline	**925.0**

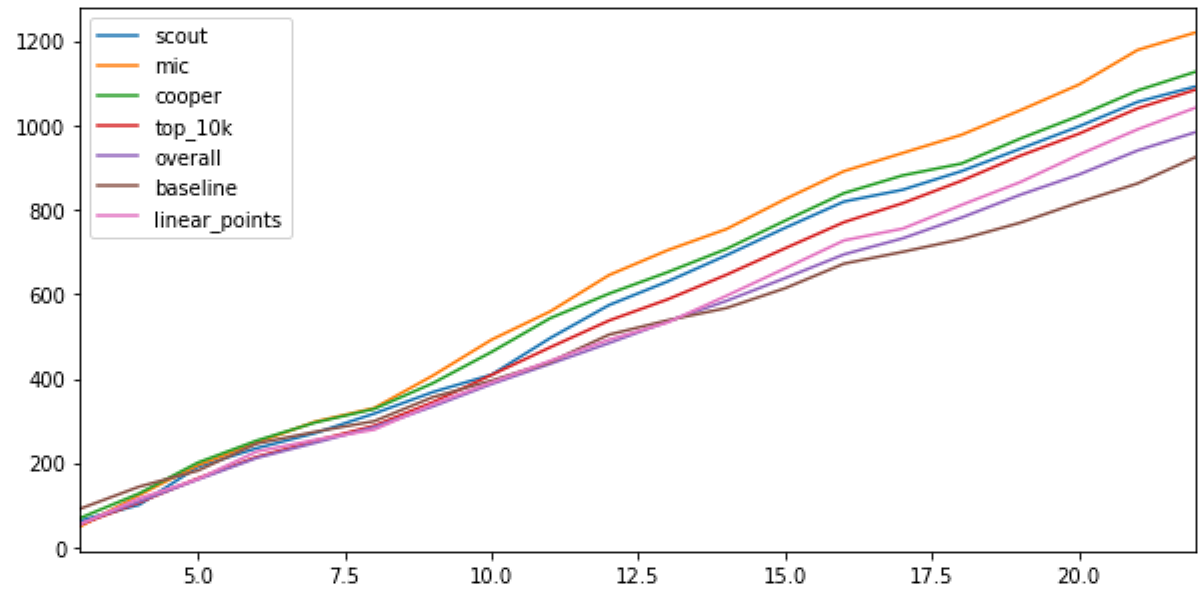
Linear Regression

Now that we've established a simple baseline, let's see how a linear regression model will perform. For each gameweek, we can look at all of the data for each player up until that point in time, and apply the statsmodels OLS to a subset of features. The target variable, once again, is `'next_week_points'`.

For each gameweek, we run through a similar loop as before, but this time, instead of ranking the players by their current points, we rank them by the linear model's predictions for the following week. Then, we use our MIP solver to select a starting lineup and a bench based on these linear model predictions.

The results of the linear model are very promising.

Fantasy Points through week 23



scout	**1092.0**
mic	**1220.0**
cooper	**1127.0**
top_10k	**1084.6**
linear_points	**1042.0**
overall	**984.0**
baseline	**925.0**

Random Forest

We'll implement a random forest model for the final set of predictions. Using sklearn's RandomForestRegressor. Here is the code that, given a gameweek, returns a DataFrame with the predictions for the following week attached:

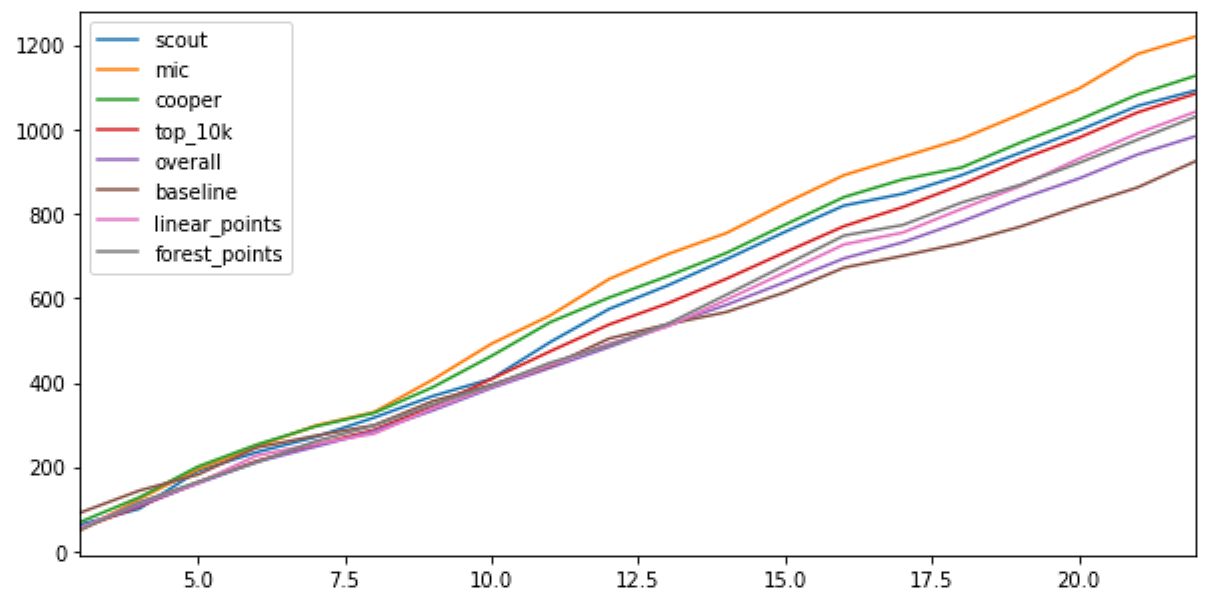
```
def forest_predict(gameweek):
    df_clean = df_forest.drop('id_', axis=1)
    df_clean = df_clean[df_clean['round'] <= gameweek - 1]
    X_train = df_clean.drop('next_week_points', axis=1)
    y_train = df_clean.next_week_points
    rfr = RandomForestRegressor(n_estimators=250, max_depth=4, random_state=42)
    rfr.fit(X_train, y_train)
    next_week = df_forest[df_forest['round'] == gameweek]
    X_test = next_week.drop(['id_', 'next_week_points'], axis=1)
    y_pred = rfr.predict(X_test)
    next_week['forest_predictions'] = y_pred
    return next_week
```

For each gameweek, we train a new random forest that includes the recently acquired data of the current week. These predictions are merged with the original DataFrame, `top500`, and from there the MIP solver is implemented, using the random forest predictions as profit and the player values as they stand each gameweek.

Here is an example of the team selections for gameweek 23 and their corresponding fantasy point totals:

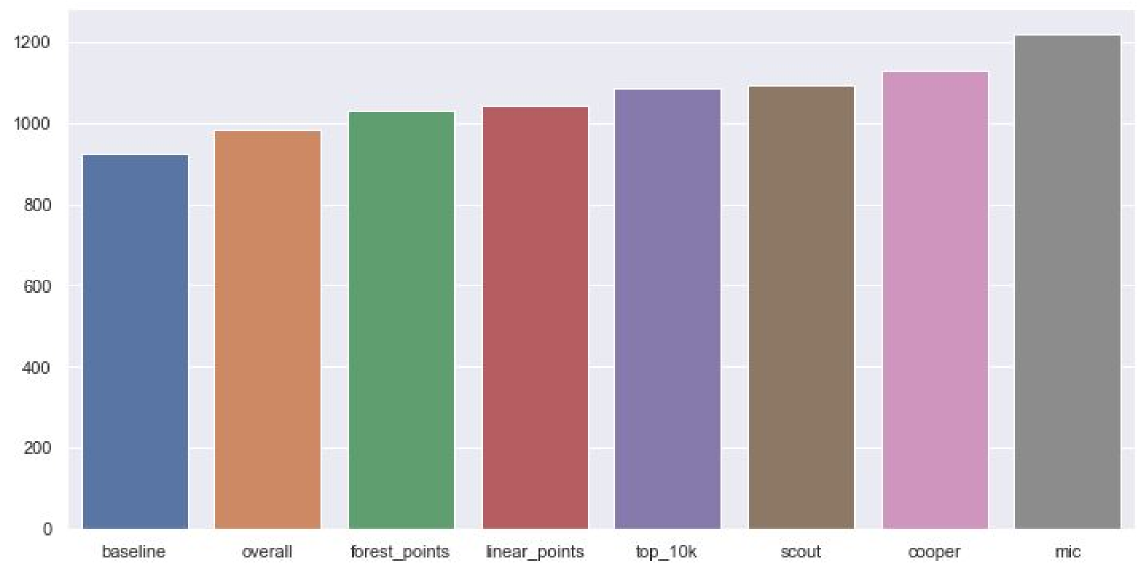
Cooper 23	Baseline 23	Linear 23	Forest 23
Alisson	Alisson	Schmeichel	Ryan
Soyuncu	Digne	Alexander-Arnold	Alexander-Arnold
Dunk	Azpilicueta	van Dijk	van Dijk
Lundstram	James	Lundstram	Lundstram
Traore	Williams	Mane	De Bruyne
Mane	Mahrez	Maddison	Sterling
De Bruyne	Doucouré	Richarlison	Maddison
Maddison	Praet	Traore	Cantwell
Salah	Mata	Vardy	Vardy
Vardy	Rashford	Abraham	Jimenez
Abraham	Agüero	Ings	Ings
Total Points: 45	Total Points: 63	Total Points: 52	Total Points: 55

Fantasy Points through week 23



mic	**1220.0**
cooper	**1127.0**
scout	**1092.0**
top_10k	**1084.6**
linear_points	**1042.0**
forest_points	**1030.0**
overall	**984.0**
baseline	**925.0**

Final point totals



Conclusions

Originally we set out to try to beat the Scout with our ML inspired teams. In this endeavor, we came up short. The Scout still outperforms the best models by a pretty significant margin. Our ML models, however, still performed well above the baseline model, and also outperformed the average fantasy team by a good chunk. It is tough to accurately determine what these teams would have ranked back in gameweek 23 when the data was compiled, but with the help of FPL Analytics' ranking tool, we can safely assume that both the linear regression model and the random forest ensemble both led to teams performing well within the top 20% of all players. It will be interesting to see if the random forest improves over time as the model is able to collect more data.