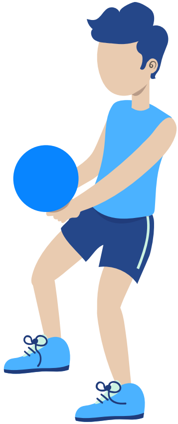
Fantasy Premier League Soccer Prediction Project

Ethan Chang and Flora Yuan

Research Motivation



- Sports-related
- Inspired by the 2022 World Cup
- Passion for soccer and fantasy leagues (I’ve lost a lot)

So we wanted to find a way to use ML models to find the best players to choose on Fantasy Teams

Introduction

1. Fantasy Soccer (football in Premier League England)
2. Game:
    - Draft a virtual team (11) before the season
    - Earn points based on real-life performance

The actual formula is a bit more complicated: some linear combination of time played, goals, assists, clean sheets, and more. 6 pointer per goal, 3 points per assist, 1 point per clean sheet, captain x2, etc.

3. Our Goal:
    - Use player statistics and past data from 2021
    - Predict who will perform the best in 2022-2023 Season (ongoing) on aggregate and each week
    - We backtest results to check in 2022-2023
    - Data is updated each week (explain more)

Roadmap

01 Data Collection

02 Data Extrapolation

03 Machine Learning

04 Results

Data Collection and Cleaning

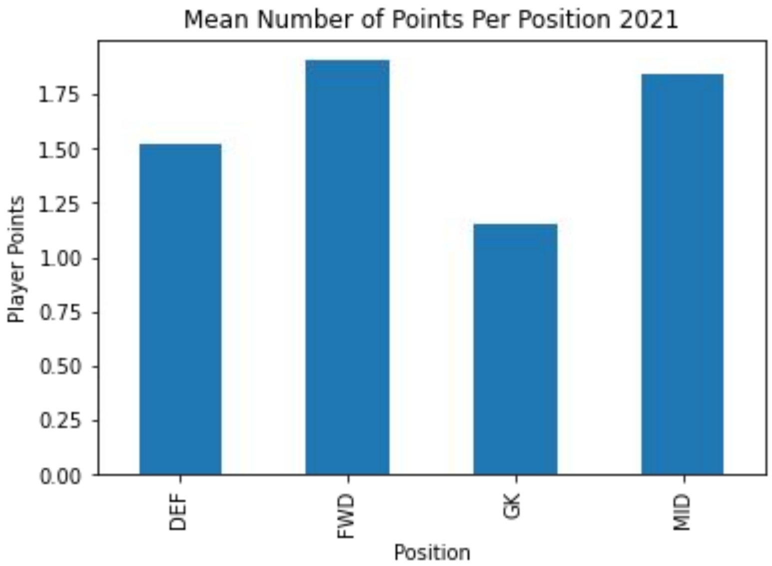
We want a dataset with each individual player in each gameweek

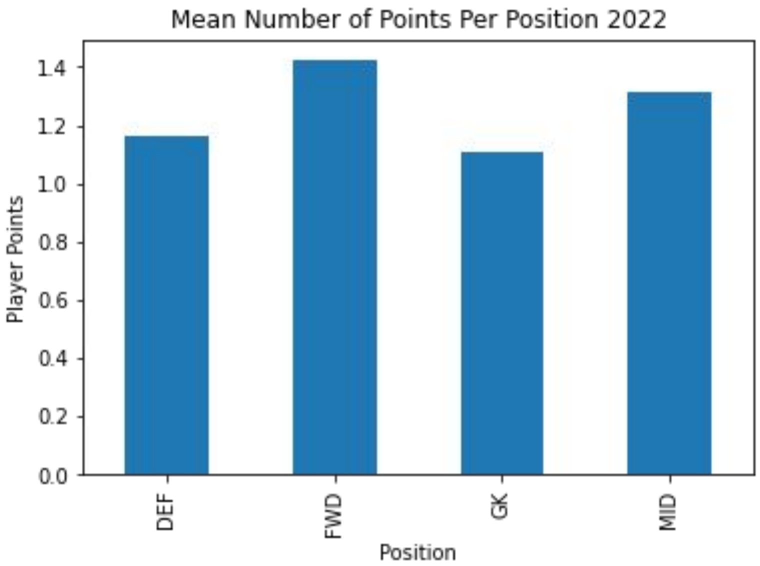
- Created datasets using joins and web scraping from https://understat.com/league/EPL, https://www.spotrac.com/epl/, and Github datasets
- For test data, we want to be able to use a player row of certain stats that we know and then predict fantasy scores.
- Cleaning:
    - Extract salary characters to turn into Euros
    - Remove duplicates during joins and extra spaces
    - Removing NaN’s
    - Filtered players with terminated and null contracts (salaries > 0)
    - 25,000 rows → 10,000 rows for 2021

Ex. 2021 Data

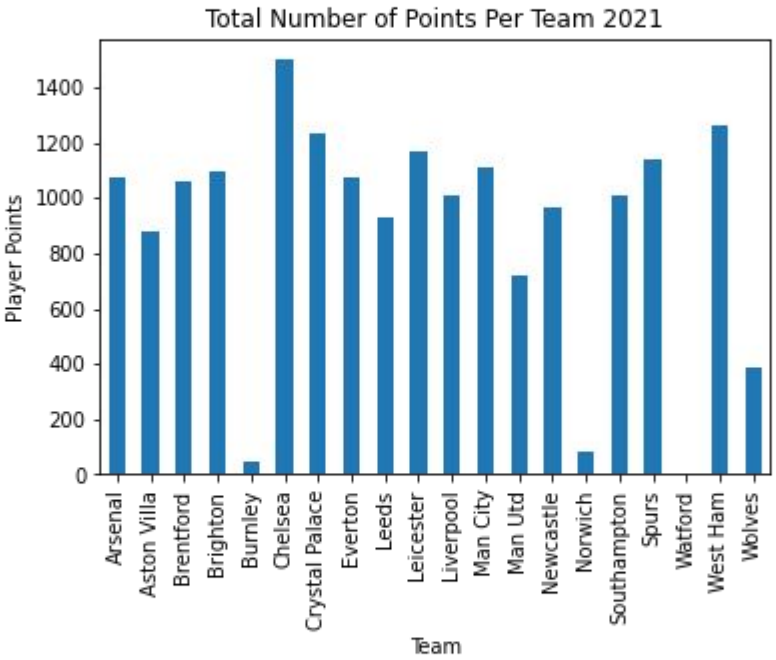
	name	position	team	xP	assists	bonus	bps	clean_sheets	creativity	element	...	threat
1	Junior Stanislas	MID	Bournemouth	1.1	0	0	3	0	0.0	58	...	0.0
2	Armando Broja	FWD	Chelsea	2.0	0	0	3	0	0.3	150	...	19.0
5	Brennan Johnson	FWD	Nott'm Forest	1.3	0	0	3	0	0.9	394	...	6.0
9	Fin Stevens	DEF	Brentford	1.0	0	0	0	0	0.0	540	...	0.0
10	Brandon Austin	GK	Spurs	1.5	0	0	0	0	0.0	451	...	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
18248	James Justin	DEF	Leicester	0.0	0	0	0	0	0.0	268	...	0.0
18251	Hugo Lloris	GK	Spurs	0.0	0	0	0	0	0.0	425	...	0.0
18252	Nick Pope	GK	Newcastle	0.8	0	0	15	0	0.0	376	...	0.0
18254	Ryan Sessegnon	DEF	Spurs	0.0	0	0	0	0	0.0	436	...	0.0
18258	Philip Billing	MID	Bournemouth	4.8	0	3	30	1	2.4	70	...	19.0

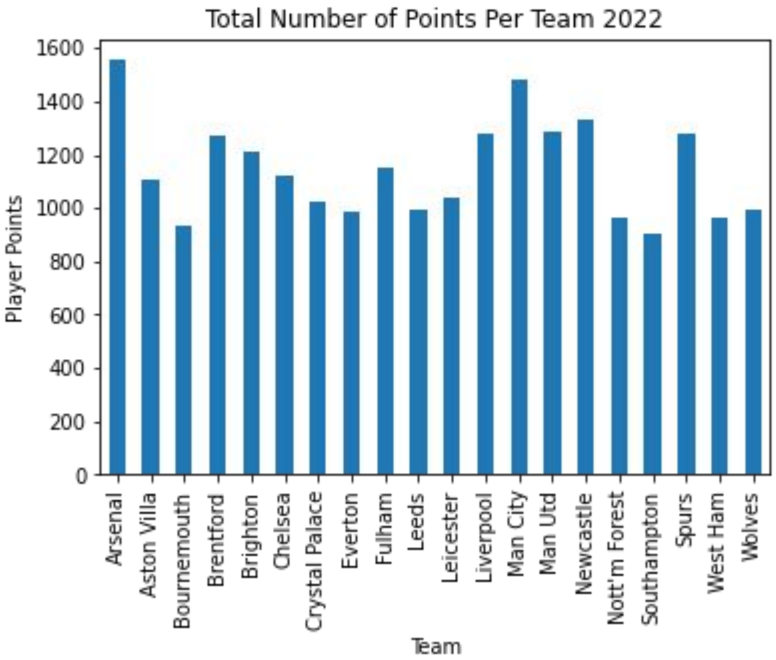
Data Extrapolation



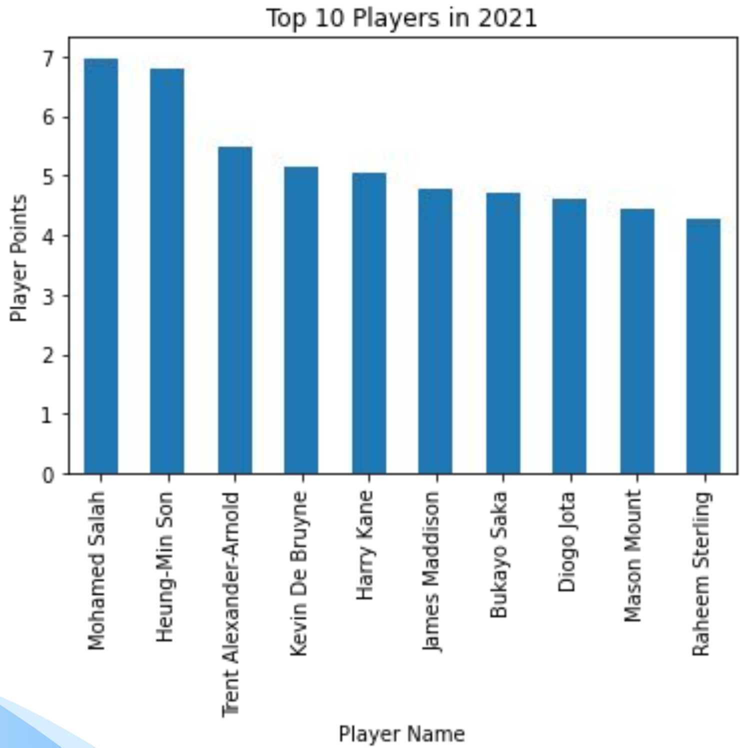


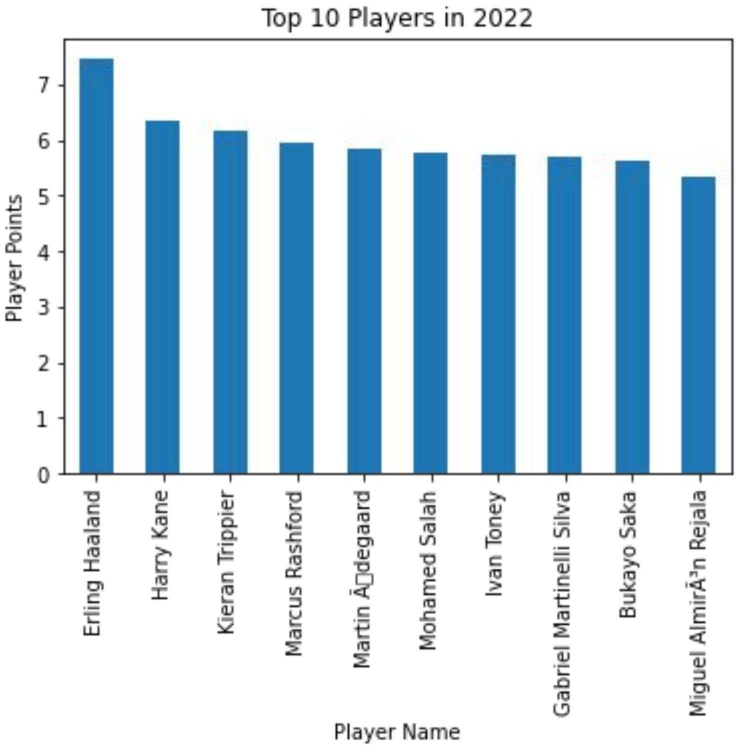
Data Extrapolation



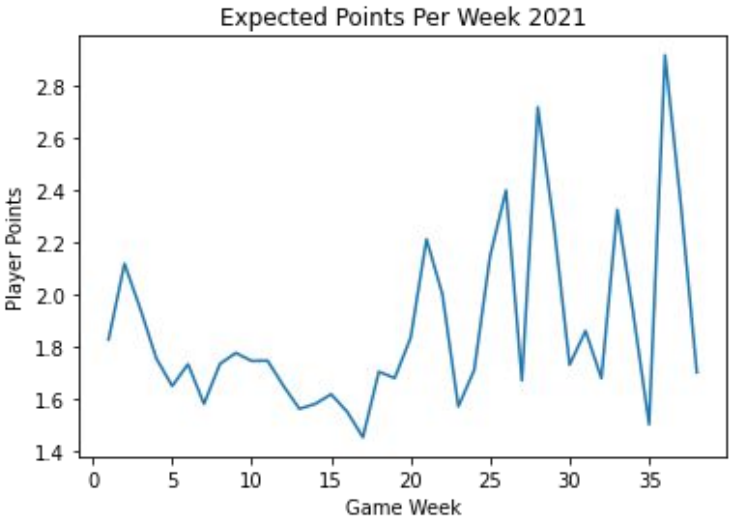


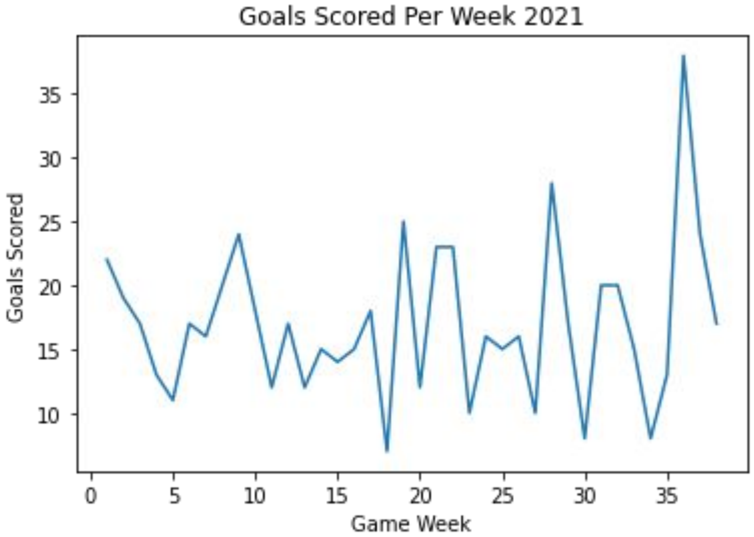
Data Extrapolation



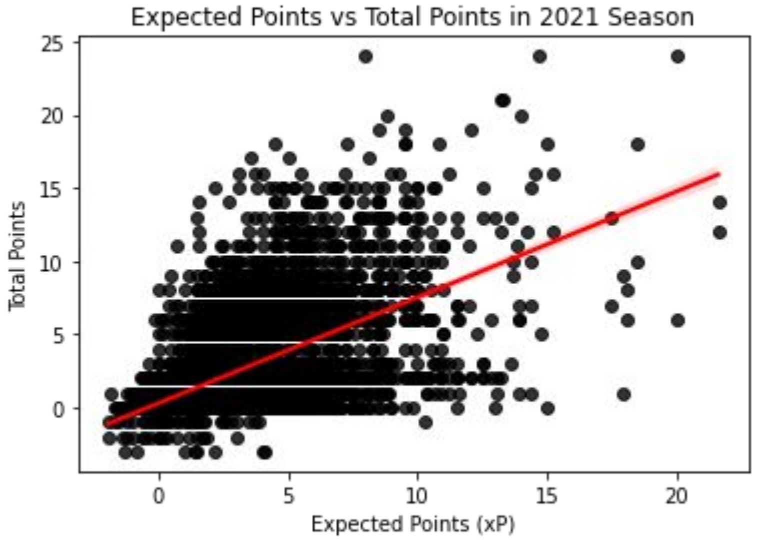


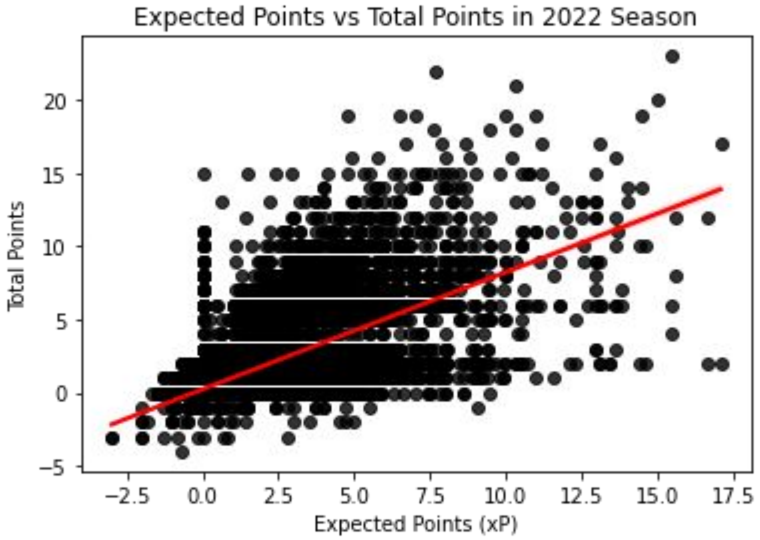
Data Extrapolation



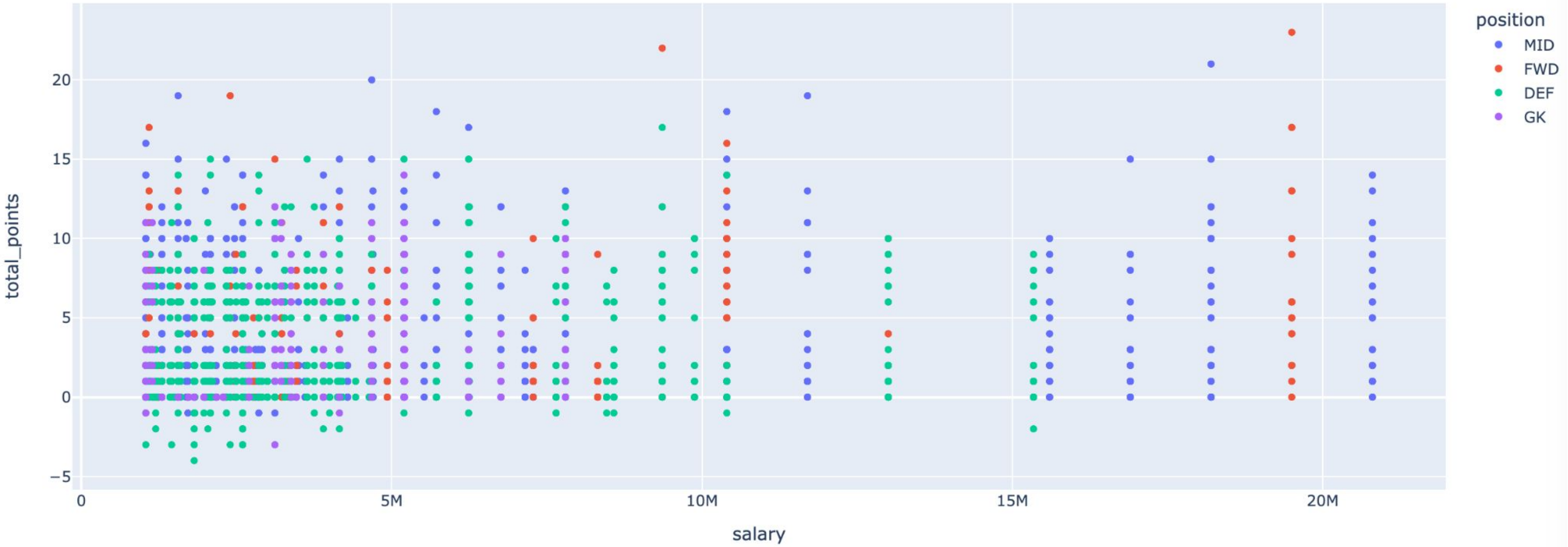


Data Extrapolation





Data Extrapolation



Machine Learning (Feature Selection)

- Original Dataset: 36 columns = 36 variables
- Eliminate overfitting – dropped variables that can influence outcome
  - Data is historic
  - Postdiction analysis – We can't use future information to predict in backtesting
- Used Variance Threshold from sklearn.feature_selection
  - Dropped variables (assists, goals, cleansheets)
  - Removes all the low variance features (bad Indication)
  - Ordinal Encoder: unique category value – integer value
  - Trained on remaining variables
  - 0.2 threshold = dropping 80%+ similar
  - Ended with 11 variables

```
X_train1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10682 entries, 0 to 25484
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   position           10682 non-null  object
 1   xP                 10682 non-null  float64
 2   own_goals          10682 non-null  int64
 3   team_a_score       10682 non-null  int64
 4   team_h_score       10682 non-null  int64
 5   threat             10682 non-null  float64
 6   transfers_balance  10682 non-null  int64
 7   transfers_in       10682 non-null  int64
 8   transfers_out      10682 non-null  int64
 9   value              10682 non-null  int64
 10  was_home           10682 non-null  bool
 11  salary             10682 non-null  float64
```

```
var_thr = VarianceThreshold(threshold = 0.2)
var_thr.fit(X_train1)

var_thr.get_support()
```

```
array([ True,  True, False,  True,  True,  True,  True,  True,  True,
        True,  True,  True])
```

X_train 2021 data

position	xP	transfers_balance	transfers_in	transfers_out	value	was_home	threat	salary	team_a_score	team_h_score
MID	1.1	0	0	0	50	True	0.0	1820000.0	0	2
FWD	2.0	0	0	0	55	False	19.0	2080000.0	1	0
FWD	1.3	0	0	0	60	False	6.0	1560000.0	0	2
DEF	1.0	0	0	0	40	False	0.0	280000.0	2	2
GK	1.5	0	0	0	40	True	0.0	80000.0	1	4
...	...	...	...	...	...	...	...	...	...	...
DEF	0.0	-839	48	887	42	True	0.0	429000.0	3	1
GK	0.0	-7647	97	7744	54	True	0.0	5200000.0	1	3
GK	0.8	-35410	32306	67716	54	True	0.0	3120000.0	1	2
DEF	0.0	-1668	14	1682	44	True	0.0	3000000.0	1	3
MID	4.8	3887	5788	1901	51	True	19.0	2080000.0	0	1

Machine Learning (Models)

Some features were categorical – used OneHotEncoder
KNN Neighbors performed the best of the models

- Used Grid-Search for KNN → neighbors = 16, metric = “manhattan”
- Error of 3.37

Linear Regressor

- Error of 3.89

Ensemble Method – Stacker and Voting

- Stacker was better than voting
- Stacker RMSE was 3.19
- Voter RMSE was 3.35
- Hence, we chose to use stacker model for predictions

y_train.std()**2 - if we were to average the predicted points on the players, on average we get squared mean error of ~7.73

Backtesting and Results Analysis

Using our Stacker Model, we can backtest data throughout the current 2022-2023 season.

- 26 weeks/38 weeks played
- Map individual player point predictions and compare to Actual Total

We find that deviation in points is greater for week by week predictions (more volatile) than over entirety of 2022-2023 season so far.

```
DF_final["difference"].sum()/8546 = ~0.9611 -> absolute difference
```

However it is easier to more accurately get top performing players compared to getting the exact points of each week. Points are very volatile with outlier cases.

- Mo Salah scoring a hatrick against a weak opponent – 3*6 = 18
- Our model predicts he will be a top performer but not necessarily with such high points

Df_final – 2022- 2023

	name	total_points	predicted_points	difference
1	Junior Stanislas	1	0.343730	0.656270
2	Armando Broja	1	1.913760	0.913760
5	Brennan Johnson	2	0.829961	1.170039
9	Fin Stevens	0	0.467828	0.467828
10	Brandon Austin	0	0.229094	0.229094
...	...	...	...	...
18248	James Justin	0	0.039556	0.039556
18251	Hugo Lloris	0	0.312156	0.312156
18252	Nick Pope	3	1.004712	1.995288
18254	Ryan Sessegnon	0	0.094469	0.094469
18258	Philip Billing	10	4.229407	5.770593

8546 rows × 4 columns

Top Performing all Weeks (left - our prediction and right is actual)

index	name
4347	Leandro Trossard
4102	Phil Foden
4099	Erling Haaland
12178	Solly March
9242	Erling Haaland
9104	Marcus Tavernier
15349	Marcus Rashford
1984	Roberto Firmino
16860	Mohamed Salah

index	name
4099	Erling Haaland
1984	Roberto Firmino
16860	Mohamed Salah
4347	Leandro Trossard
4102	Phil Foden
12178	Solly March
7626	Callum Wilson
7818	Reiss Nelson
3145	Marcus Rashford
4575	James Maddison

Results

We used the model to predict three things:

- Aggregate (predicted) top performers for 2022
  - No room for transfers, best overall players
  - Use groupby("name")
- Best performers by position
  - More specified – easier to form team
  - Mask "position" == "DEF
- Predict for next gameweek (this case 27)
  - Data is scraped/published weekly

Aggregate Predicted Top

```
name
Erling Haaland          5.971429
Harry Kane              5.512500
Mohamed Salah           5.066667
Bukayo Saka             4.623214
Kevin De Bruyne         4.183929
                          ...
Jack Butland            0.048148
Marcus Bettinelli       0.047619
Kristoffer Klaesson     0.042593
Willy Caballero         0.039286
Tom Heaton              0.012963
Name: predicted_points, Length: 325, dtype: float64
```

Predicted Week 27

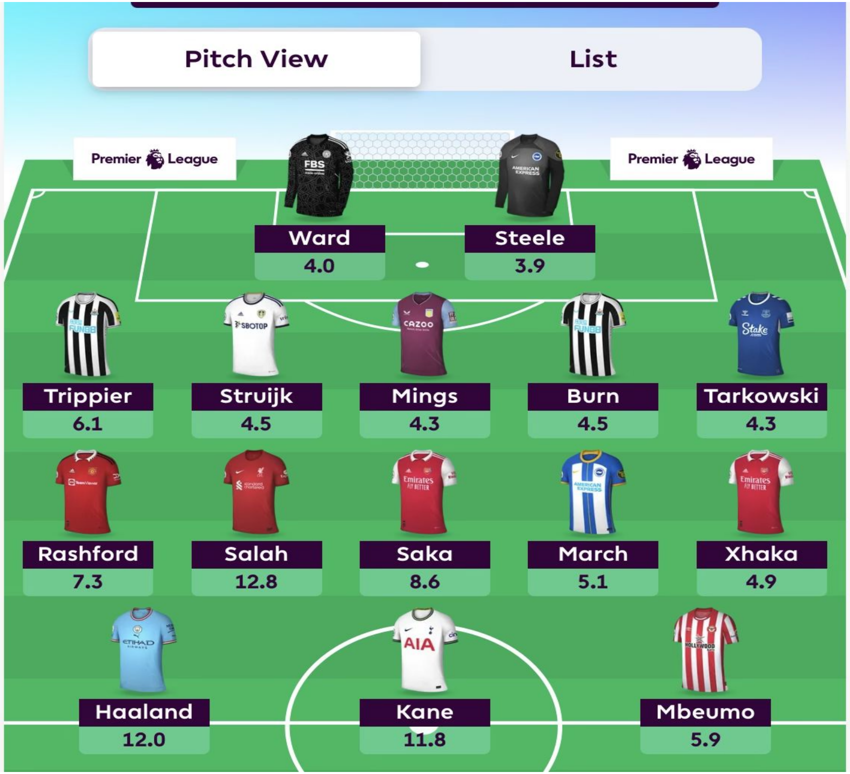
	name	predicted_points ▼
18056	Harry Kane	10.107825695463932
17392	Jason Steele	8.100023428228535
17437	Jack Harrison	8.008093579678642
17743	Kyle Walker-Peters	7.8816000843527725
18225	Alexis Mac Allister	7.7733747996567235
17606	Aaron Hickey	7.6033731147790755
17993	Ben Davies	7.41374205991279
17805	Kai Havertz	7.139329253621185
17508	Armel Bella-Kotchap	7.102682529893377
17688	Lewis Dunk	7.073155806563346
18218	Romain Perraud	7.024628246349321
17672	Gavin Bazunu	6.923768498439523
17636	Jan Bednarek	6.892657835317649
17610	Solly March	6.831285554807097
18242	Ethan Pinnock	6.763384546147829
17924	Adam Webster	6.5687006792012275
17833	Ollie Watkins	6.561547329906484

Aggregate Predicted Top

```
name
Erling Haaland        7.464286
Harry Kane            6.357143
Kieran Trippier       6.153846
Marcus Rashford       5.962963
Mohamed Salah         5.777778
                        ...
Lyle Taylor           0.000000
Scott Carson          0.000000
Brandon Williams      0.000000
Brandon Austin        0.000000
Tyler Roberts         0.000000
```

Applied Results : *http://fantasy.premierleague.com/*

Aggregate Team



Thank You!