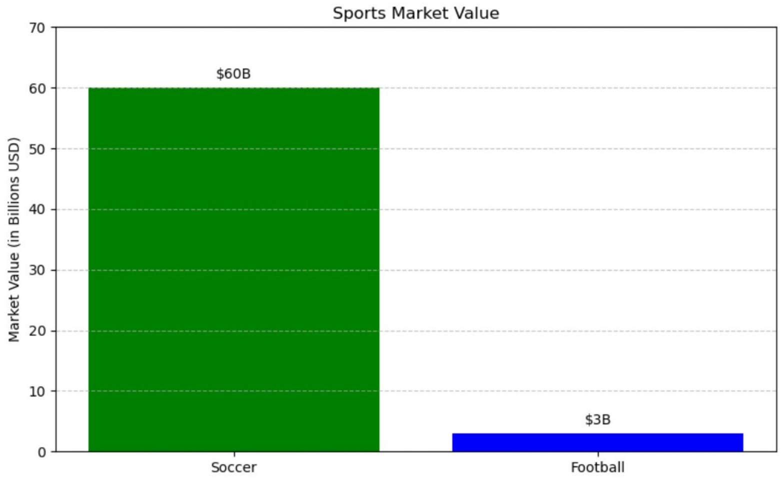
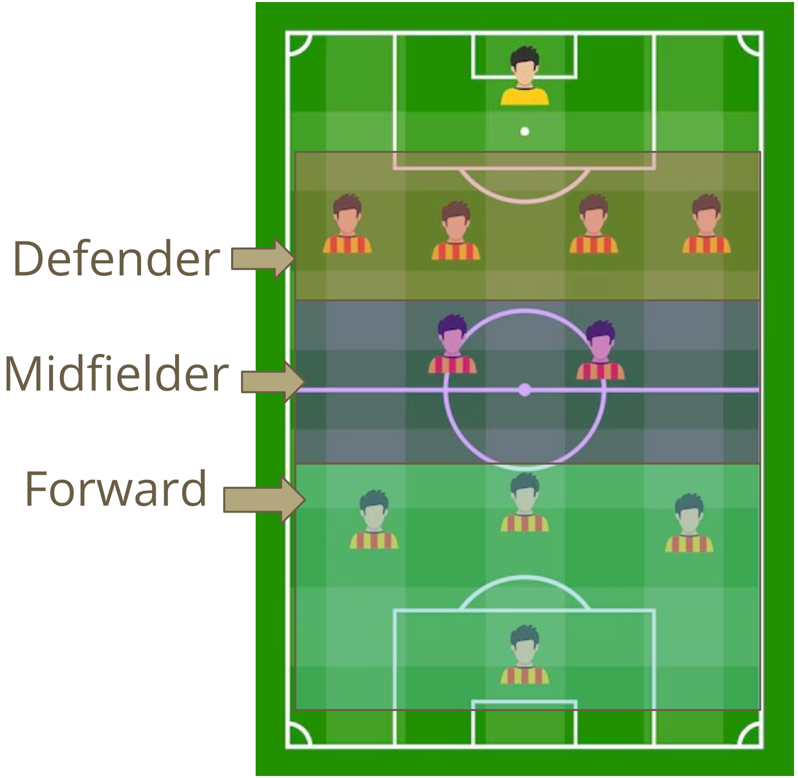
European Football

(Known as soccer)

Justin O’Donnell, Brian Pascente, Carson Holscher, Jacob Richardson

Basics of Soccer





Basics of Soccer Data

xG - Shot location (distance and angle to goal), body part used (foot, head, etc.), type of assist (cross, through ball, etc.), defensive pressure, and game context (e.g. counterattack, set-piece)

xA - the player who passed the ball before is credited with the xG of the shot that follows

xGBuildUp - If the player touches the ball before the shot is taken, then the xG is added to that player. This resets every time to other team gains possession

xGChain - Combination of xG, xA, and xGBuildup

Production per value (PPV)

- Our goal is to predict the value of soccer players based on all of their data from the season prior. This is to simulate when a team wants to buy another team's player and predict how good they will be.

$$\text{PPV (Midfielders)} = \frac{\text{Goals} \cdot \mathbf{0.4} + \text{Assists} \cdot \mathbf{0.4} + \text{Points Per Game} \cdot \mathbf{0.2}}{\text{Player Value}}$$

$$\text{PPV (Forwards)} = \frac{\text{Goals} \cdot \mathbf{0.4} + \text{Assists} \cdot \mathbf{0.3} + \text{Points Per Game} \cdot \mathbf{0.3}}{\text{Player Value}}$$

Literature Review

- Gained inspiration for what features and models to use, as many used Random Forests, Gradient Boosting, SVM, and even Neural Networks
- Many had features to be expected like age, height, market value
- Some predictors had weather conditions and injury/psychological evaluation
- One paper discussed player price optimization as a target, which was our main inspiration for our finalized target

Dataset

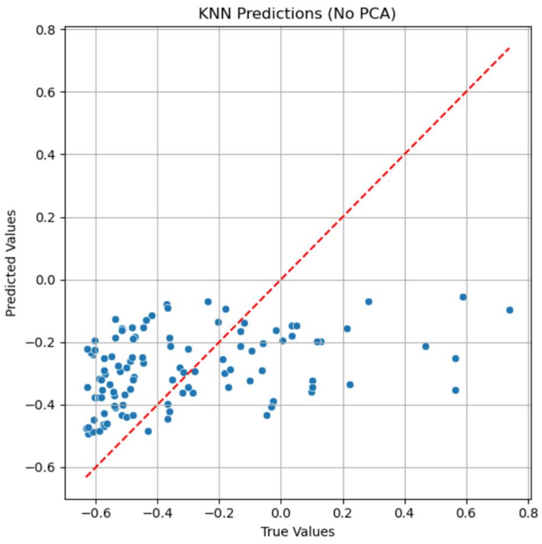
	minutes_played	goals	npg	assists	xG	xA	npxG	position_x	shots	key_passes	yellow_cards	red_cards
128	2423	0	0	2	0.886058	1.697511	0.886058	M S	21	22	8	0
106	2189	0	0	3	0.217996	1.706203	0.217996	D M S	8	17	2	0
331	996	2	2	0	1.234667	0.534926	1.234667	M S	7	8	2	0
167	2986	1	1	0	1.824859	1.692150	1.824859	D S	22	4	5	0
425	3017	0	0	2	0.134310	0.932408	0.134310	D S	8	12	4	0

xGBuildup	xGChain	market_value_in_eur	height_in_cm	age_in_months_2015	points_per_game	player_performance_valuation_standardized
8.497885	10.142531	800000.000000	177.000000	326.000000	1.314570	0.039118
5.533055	6.569234	100000.000000	180.000000	292.000000	1.213235	0.136234
1.861821	2.912522	1600000.000000	186.000000	344.000000	1.364035	-0.368572
5.403009	5.606335	400000.000000	193.000000	351.000000	1.967105	-0.075814
9.725240	10.318017	2000000.000000	182.000000	253.000000	1.157895	-0.414862

Model Performance (R^2)

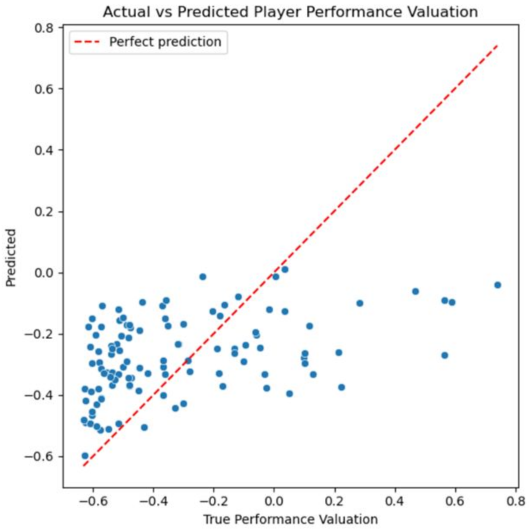
Gradient Boost	0.5018265603
KNN	0.2747
Ridge	0.10
Forest	0.11721429293514218
Lasso	0.06

KNN + PCA and Feature Engineering



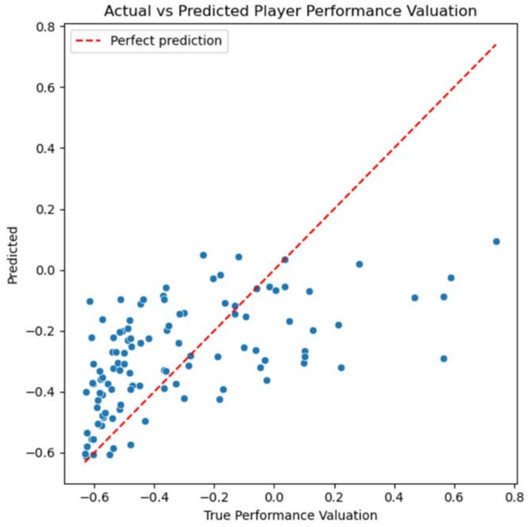
KNN (K=23)

Test R^2 = 0.114



KNN (PCA, K=23)

Test R^2 = 0.169



KNN (PCA+FE, K=15)

Test R^2 = 0.275

Gradient Boost

```
param_grid = {
    'n_estimators': [100, 200, 300, 500],
    'learning_rate': [0.01, 0.05, 0.1, 0.2],
    'max_depth': [3, 4, 5, 6, 7]
}
```

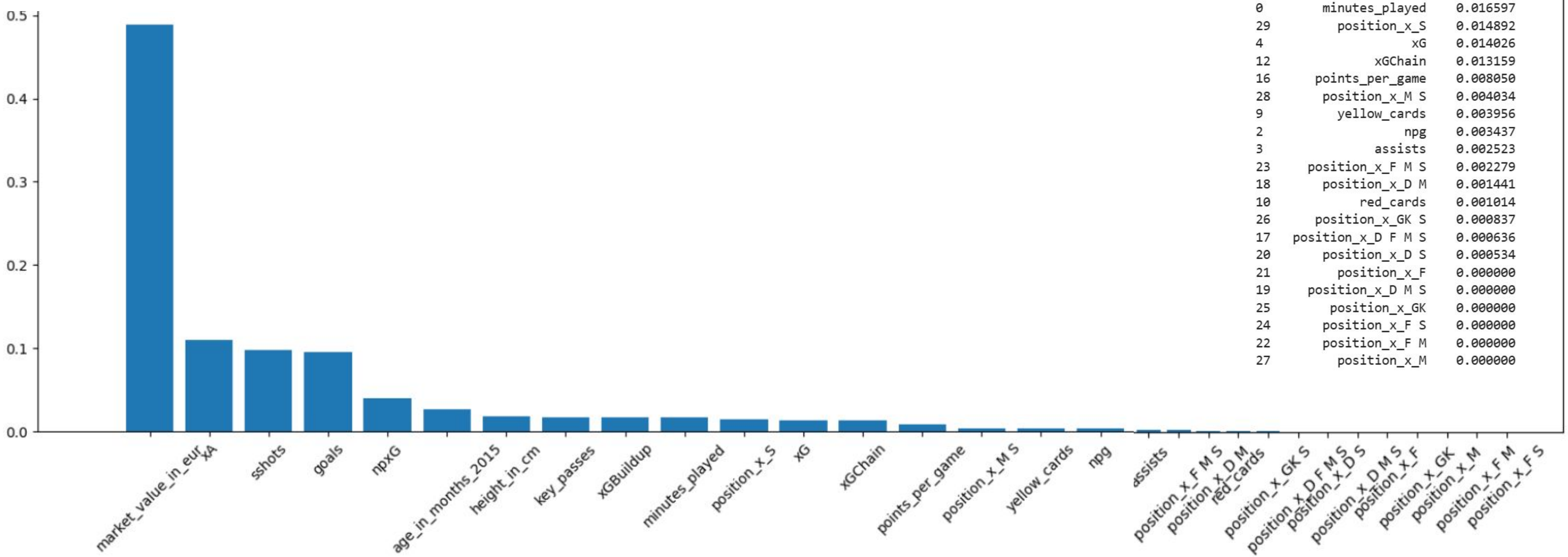
```
Best Parameters: {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 300}
```

MSE: 0.0296159665

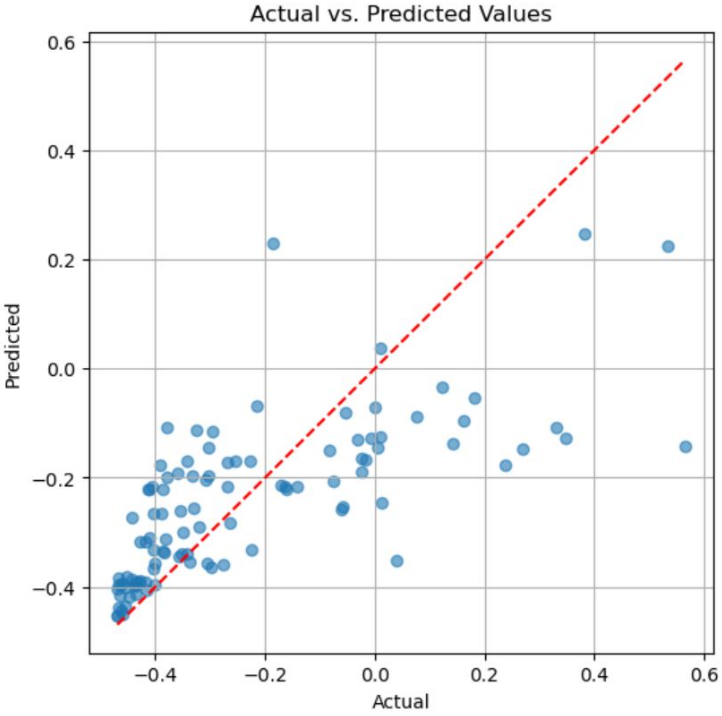
MAE: 0.1215287545

Feature Importance:

	Feature	Importance
13	market_value_in_eur	0.489266
5	xA	0.110448
7	shots	0.098139
1	goals	0.095581
6	npxG	0.039643
15	age_in_months_2015	0.027202
14	height_in_cm	0.017943
8	key_passes	0.017561
11	xGBuildup	0.016805
0	minutes_played	0.016597
29	position_x_S	0.014892
4	xG	0.014026
12	xGChain	0.013159
16	points_per_game	0.008050
28	position_x_M S	0.004034
9	yellow_cards	0.003956
2	npg	0.003437
3	assists	0.002523
23	position_x_F M S	0.002279
18	position_x_D M	0.001441
10	red_cards	0.001014
26	position_x_GK S	0.000837
17	position_x_D F M S	0.000636
20	position_x_D S	0.000534
21	position_x_F	0.000000
19	position_x_D M S	0.000000
25	position_x_GK	0.000000
24	position_x_F S	0.000000
22	position_x_F M	0.000000
27	position_x_M	0.000000



Actual vs Predicted Values



Demo Example 1:

```
data = pd.DataFrame([
    {
    'minutes_played': 2823,
    'goals': 6,
    'npg': 6,
    'assists': 7,
    'xG': 2.794280,
    'xA': 5.305932,
    'npxG': 2.794280,
    'position_x': 'F M S',
    'shots': 65,
    'key_passes': 65,
    'yellow_cards': 7,
    'red_cards': 1,
    'xGBuildup': 4.623076,
    'xGChain': 8.855647,
    'market_value_in_eur': 300000.0,
    'height_in_cm': 173.0,
    'age_in_months_2015': 345.0,
    'points_per_game': 1.184211
    }])
```

```
Predicted performance valuation: 0.2902878596
```

Jason Puncheon
Good Player
Expected: 0.258462

Demo Example 2:

```
data = [{
    'minutes_played': 2682,
    'goals': 12,
    'npg': 12,
    'assists': 7,
    'xG': 9.096988,
    'xA': 10.388413,
    'npxG': 9.096988,
    'position_x': 'M',
    'shots': 66,
    'key_passes': 92,
    'yellow_cards': 8,
    'red_cards': 0,
    'xGBuildup': 16.633573,
    'xGChain': 29.144278,
    'market_value_in_eur': 50000.0,
    'height_in_cm': 178.0,
    'age_in_months_2015': 341.0,
    'points_per_game': 1.927632
}]
```

```
Predicted performance valuation: 0.4036456920
```

David Silva
Very Good Player
Expected: 11.3

Demo Example 3:

```
data = [{
    'minutes_played': 1092,
    'goals': 0,
    'npg': 0,
    'assists': 1,
    'xG': 0.106066,
    'xA': 0.344920,
    'npxG': 0.106066,
    'position_x': 'D S',
    'shots': 7,
    'key_passes': 8,
    'yellow_cards': 0,
    'red_cards': 0,
    'xGBuildup': 2.098642,
    'xGChain': 2.395614,
    'market_value_in_eur': 1800000.0,
    'height_in_cm': 179.0,
    'age_in_months_2015': 267.0,
    'points_per_game': 1.169173
}]
```

```
Predicted performance valuation: -0.4100520727
```

Bad Player
Massadio Haidara
Expected: -0.461564

Demo 4: Random player

Future Work

- Probably look into aggregating model results. The author of one paper that we read used a combination of random forest regression, support vectors & gradient boosting.
- Remove the bench players when evaluating higher performing players.
- Fix the joins so that we don’t unnecessarily delete a good chunk of our data, due to team names.
- Procure a more complete dataset (such as the one used for FIFA)

Lessons

- Overfitting is very easy to do in the real world
- There's no perfect metric for what makes a good soccer player. Your best bet for figuring it out is common sense, but that'll only get you so far.
- A few bad assumptions can severely affect your model's performance

Questions?