Release Clause of Soccer Players in FC 24 Prediction: Machine Learning Models and Exploration

ECE 208
Haotian Yang
Guangzhou Cai
Yicheng Shi

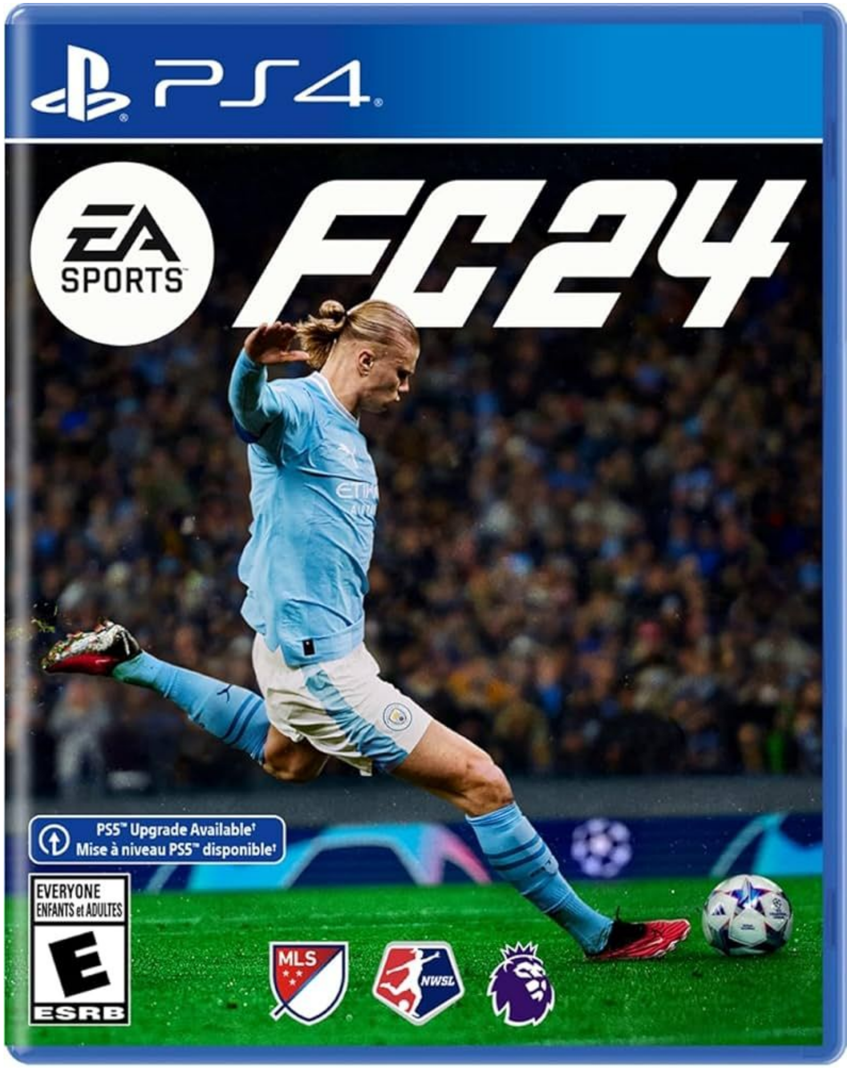
Topics that will be covered

- Motivation
- FC 24
- Data Preprocessing
- Model Selection
- Model Refinement
- Results Comparison
- Conclusion and Recommendation

Motivation

1. Help players the make quicker and better decisions in manager mode.

2. Useful tool for finding players with underrated release clause.



Our Data:

Name	Age	O...	Po...	Team & Contract	ID	Height	Weight	foot	Be...	Best...	Gr...	Value	Wage	Release...	Tota...	Cr...	Fi...	H...	Sh...	Vo...	To
Vitor Roque ST RW LW	18	76	88	FC Barcelona 2024 ~ 2031	276589	172cm / 5'8"	78kg / 172lbs	Right	78	ST	12	€17.5M	€44K	€39.4M	335	62	82	62	54	75	3
L. Vušković CB	16	62	86	Radomiak Radom 2023 ~ 2026	275192	193cm / 6'4"	86kg / 190lbs	Right	64	CB	24	€1.2M	€500	€2.8M	262	36	60	71	59	36	2
C. Palmer RM RW CAM	21	79 +2	87 +3	Chelsea 2023 ~ 2030	257534	189cm / 6'2"	74kg / 163lbs	Left	82	CAM	8	€39.5M	€76K	€83M	343	70 +2	80 +5	48 +5	82 +1	63 +8	3
K. Havertz CAM CM ST	24	82	87	Arsenal 2023 ~ 2028	235790	193cm / 6'4"	82kg / 181lbs	Left	85	CAM	5	€46M	€110K	€88.6M	404	74	83	83	84	80	3
15 Ronaldinho CAM ST	34	78	78	Querétaro 2014 ~ 2016	28130	182cm / 6'0"	76kg / 168lbs	Right	77	CAM	0	€4.1M	€45K	€0	367	84	69	50	82	82	4
W. Zaïre-Emery CM CDM	17	80 +1	90	Paris Saint Germain 2022 ~ 2025	270673	178cm / 5'10"	68kg / 150lbs	Right	82	CM	10	€43.5M	€11K	€89.2M	329	66	66 +1	66	82 +1	49	3
J. Bellingham CM CAM ST	17	66	82	Sunderland 2023 ~ 2028	270964	191cm / 6'3"	75kg / 165lbs	Right	68	CAM	16	€2.1M	€1K	€5.3M	285	51	56	64	67	47	3

Source: FC 24 Third-Party data Website

Sample Size: 15000+ players

Features: 64 (Numerical: 61, Categorical: 3)

Target: Players’ release clause value

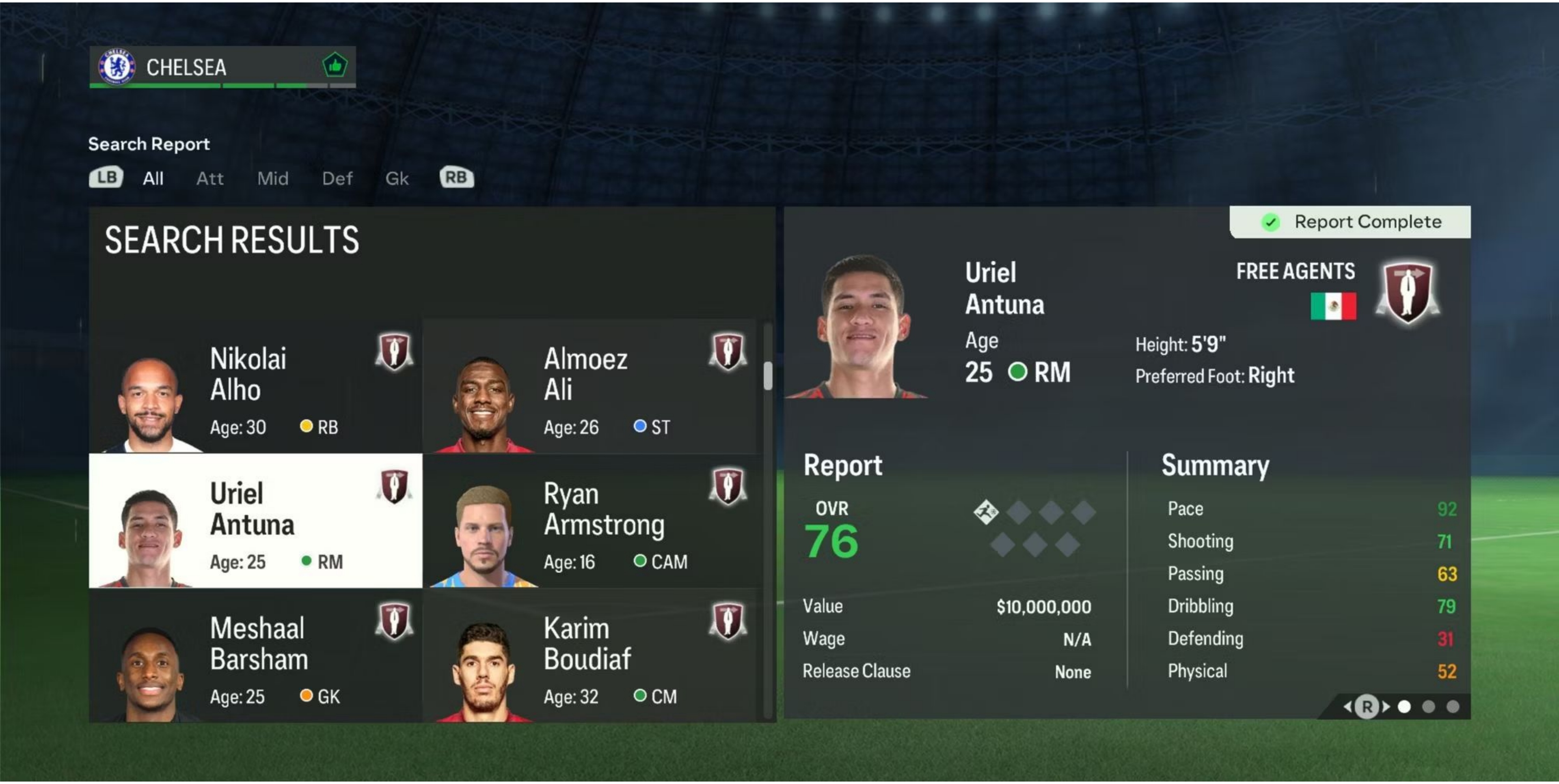
FC 24

Biggest Soccer Game

Players’ attributes are correlated with their performance in real life.

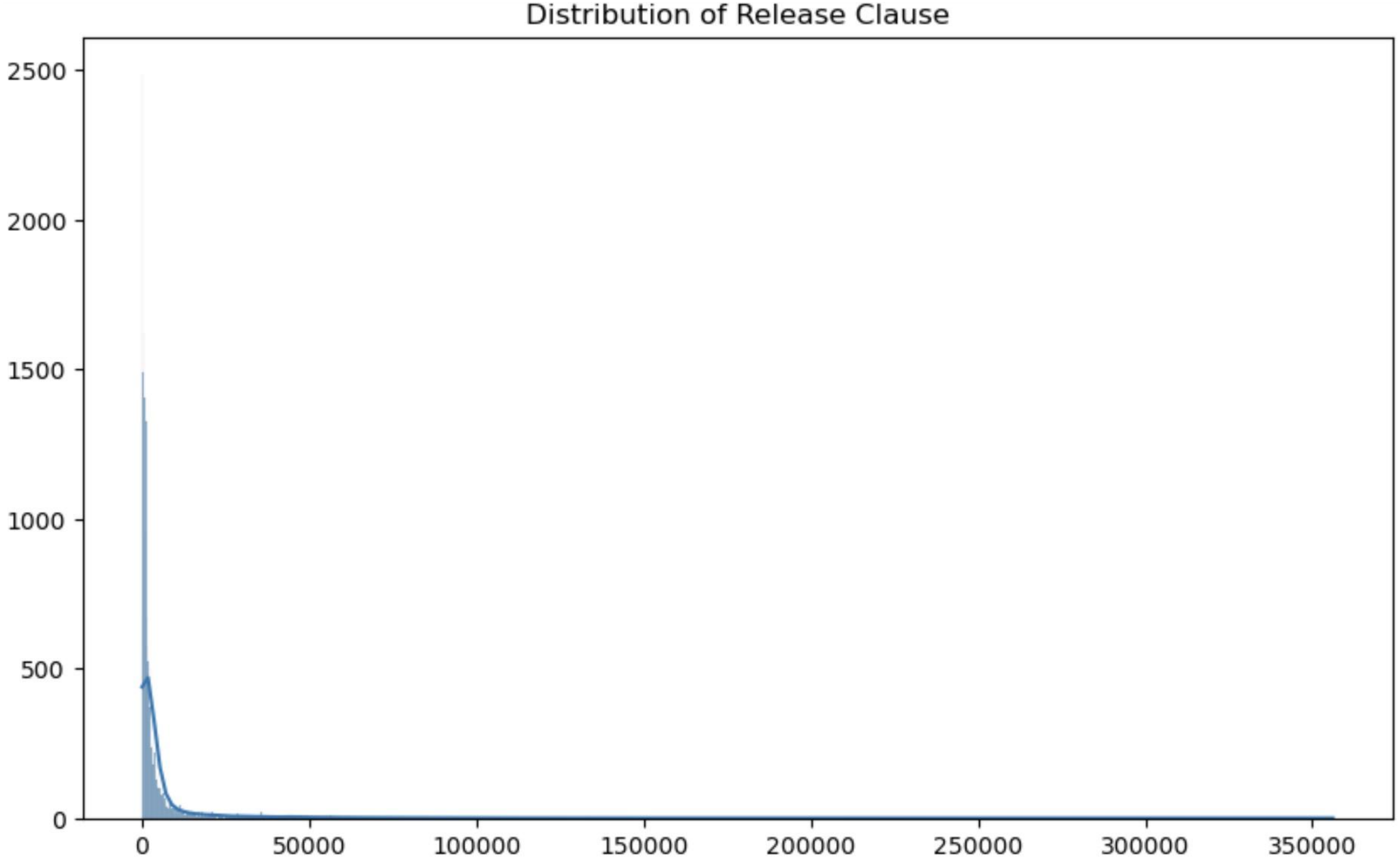
Release Clause in the game is as same as in the real life.

Manager Career



Data Preprocessing

Distribution of Release Clause



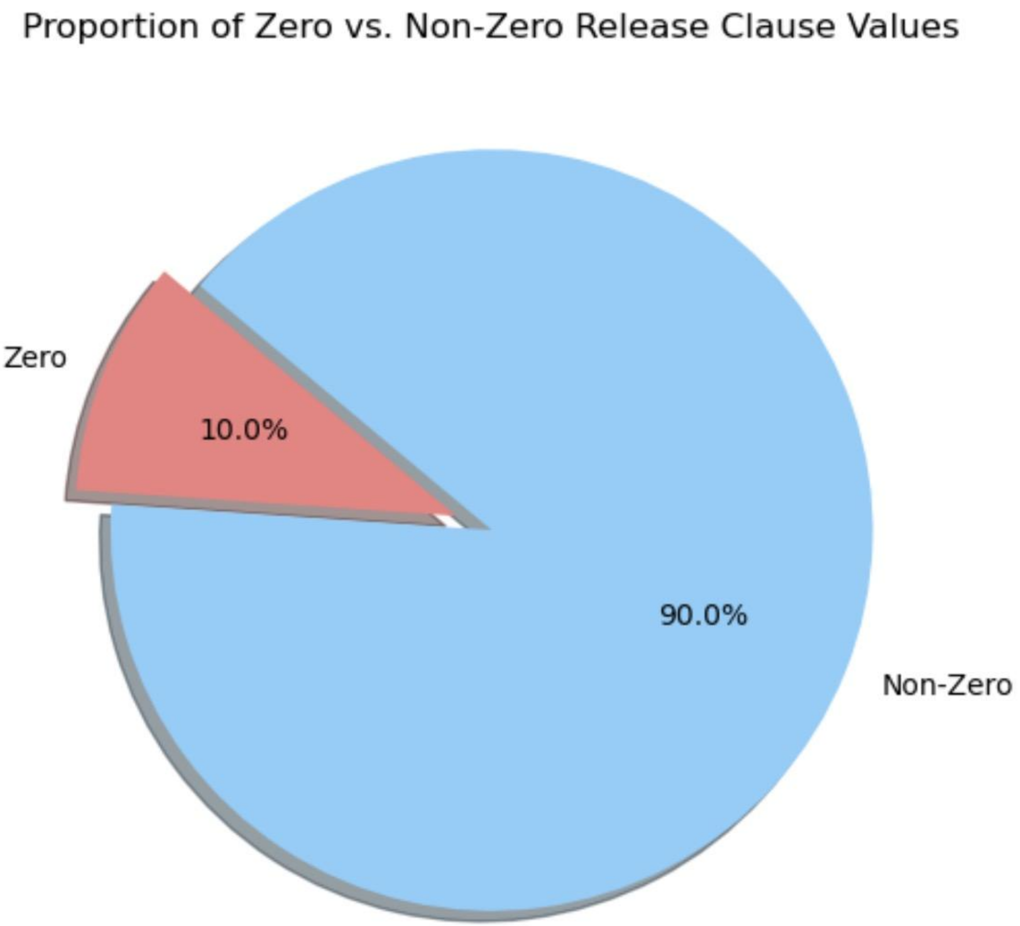
Q1 (First Quartile): 481.0
Q2 (Median/Second Quartile): 1200.0
Q3 (Third Quartile): 3300.0
Q4 (Maximum Value): 356100.0

Handling of Zero Values

Release Clause = 0 (Free agent)

Free agents in 2024 – the players who can now sign pre-contract transfer agreements

Handling of Zero Values



Logarithmic Transformation

Handling of zero values

Reducing Influence of Outliers

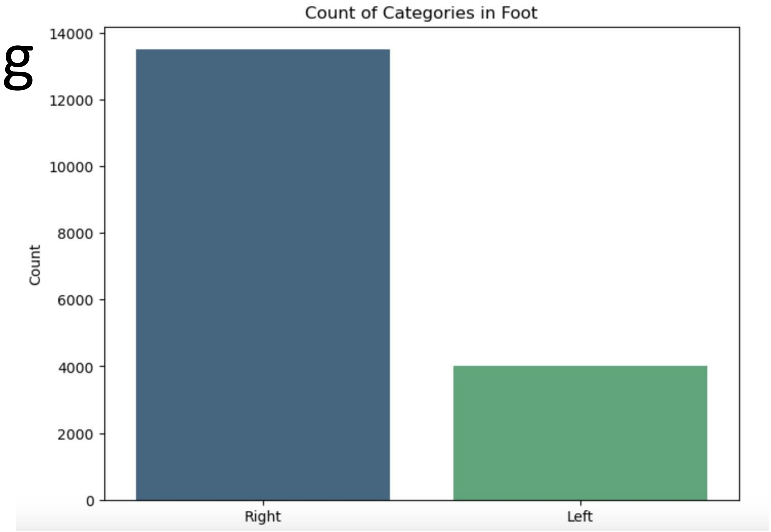
Handling Skewed Data

```
mask = new_df['Release clause(€:K)'] > 0
df1['Release clause(€:K)'] = np.log(new_df.loc[mask, 'Release clause(€:K)'])
```

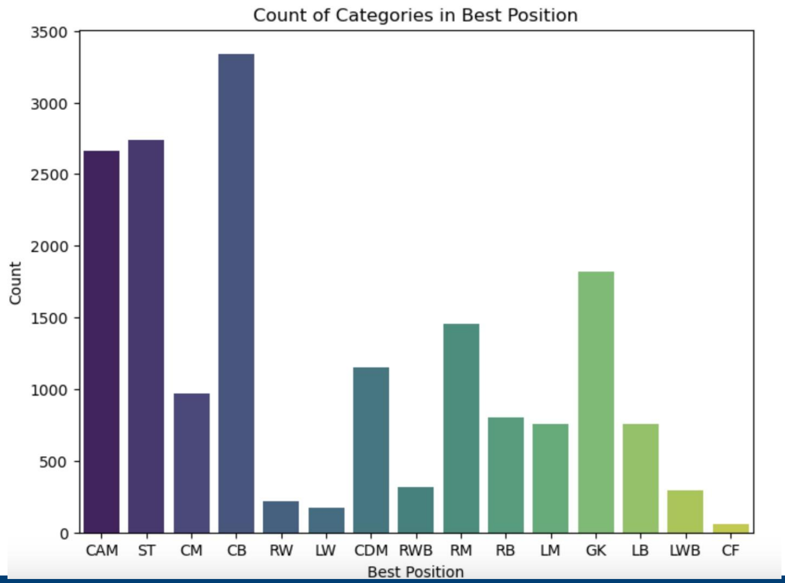
Processing Categorical Features

One-Hot Encoding

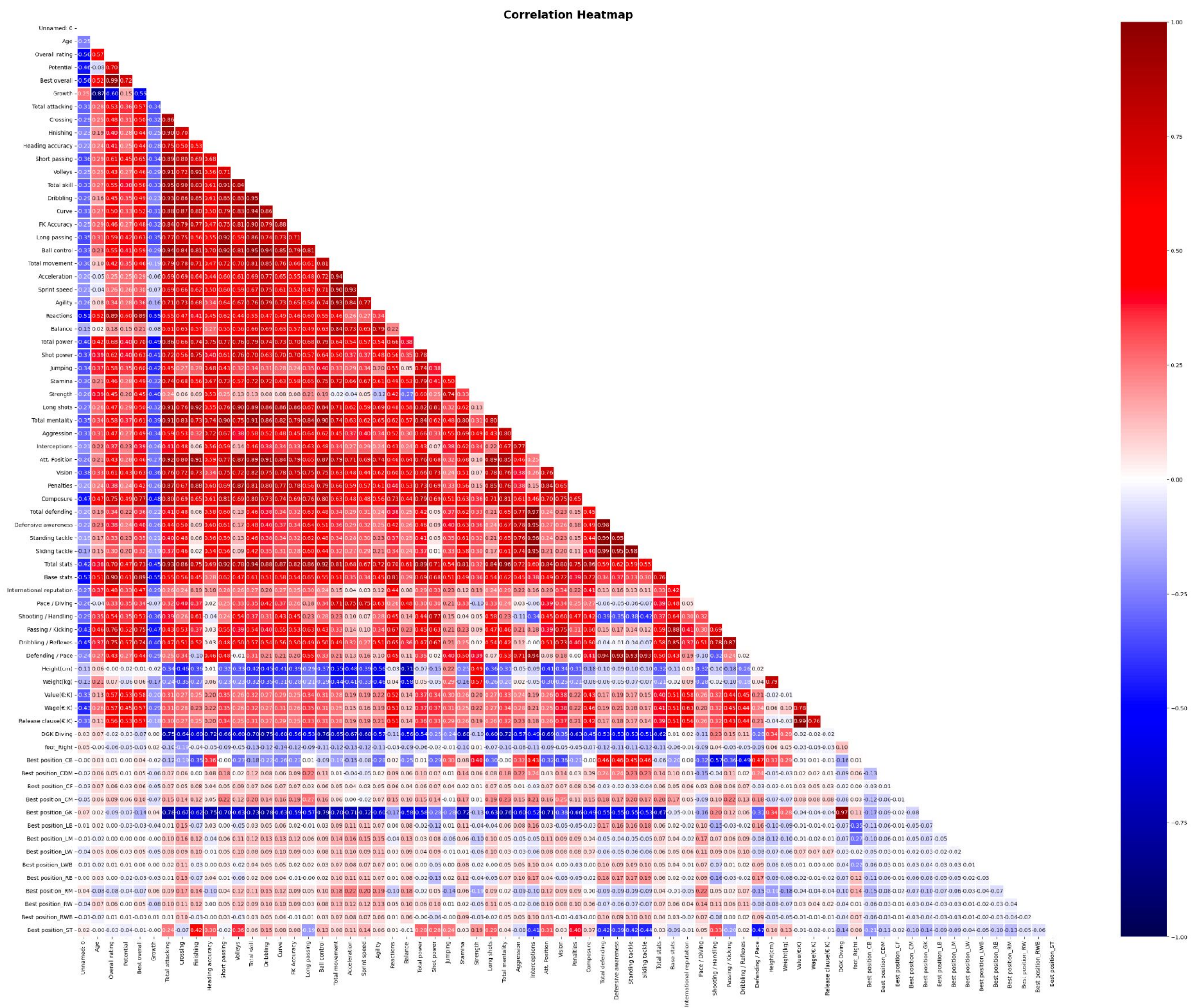
Foot: Right, Left



Best position



Correlation Heatmap

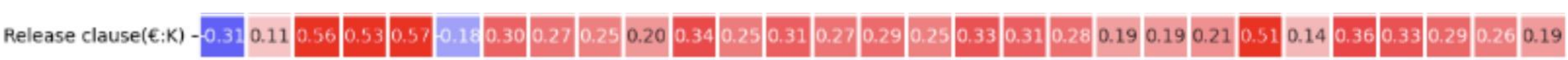


Correlation Heatmap and Feature Selection

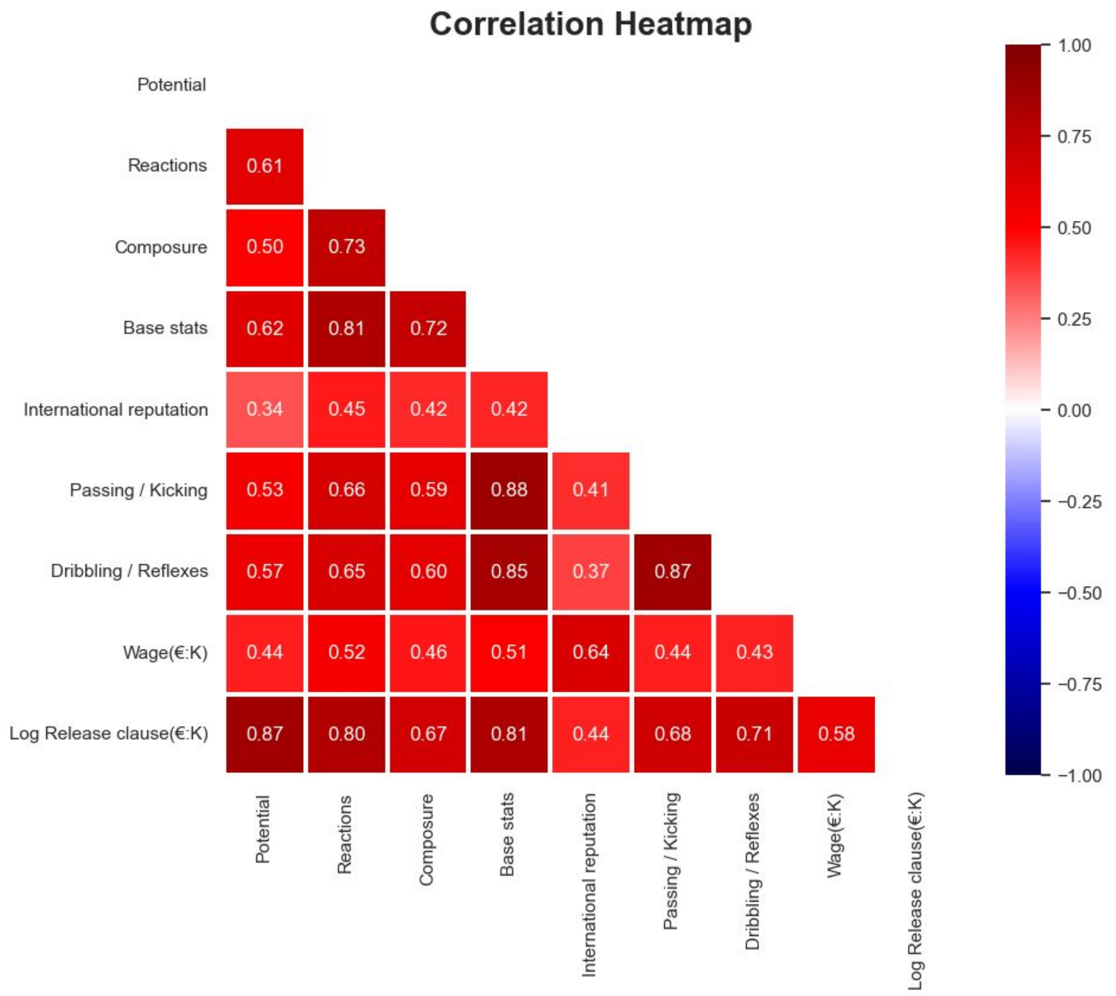
Most related: Wage(0.76), International Reputation(0.56), Potential(0.57)

Negatively related: Height(-0.04), Weight(-0.03), Growth(-0.18)

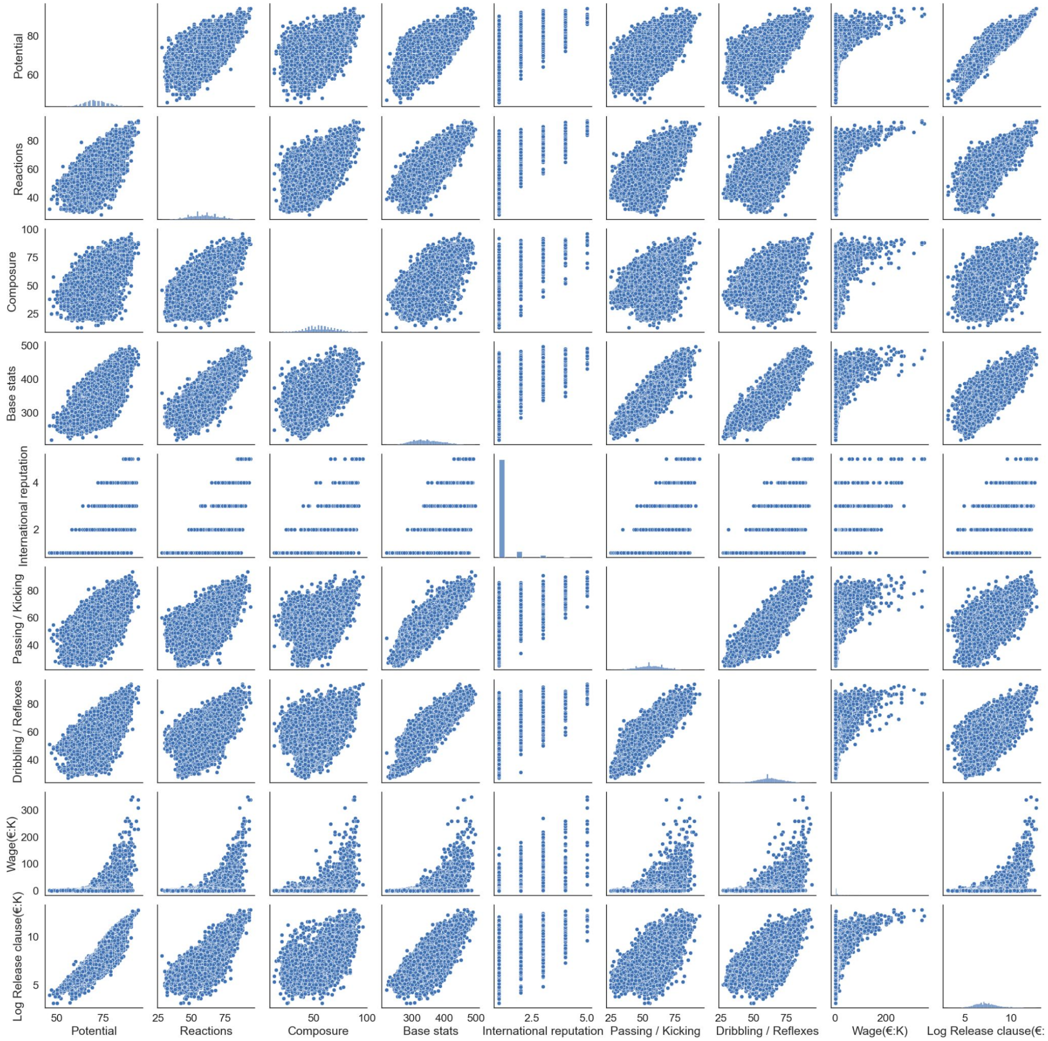
Features Selected to our Model: Potential,Reactions,Composure,Base stats,International reputation,Passing / Kicking,Dribbling / Reflexes,Wage(€:K)



Visualization



Visualization



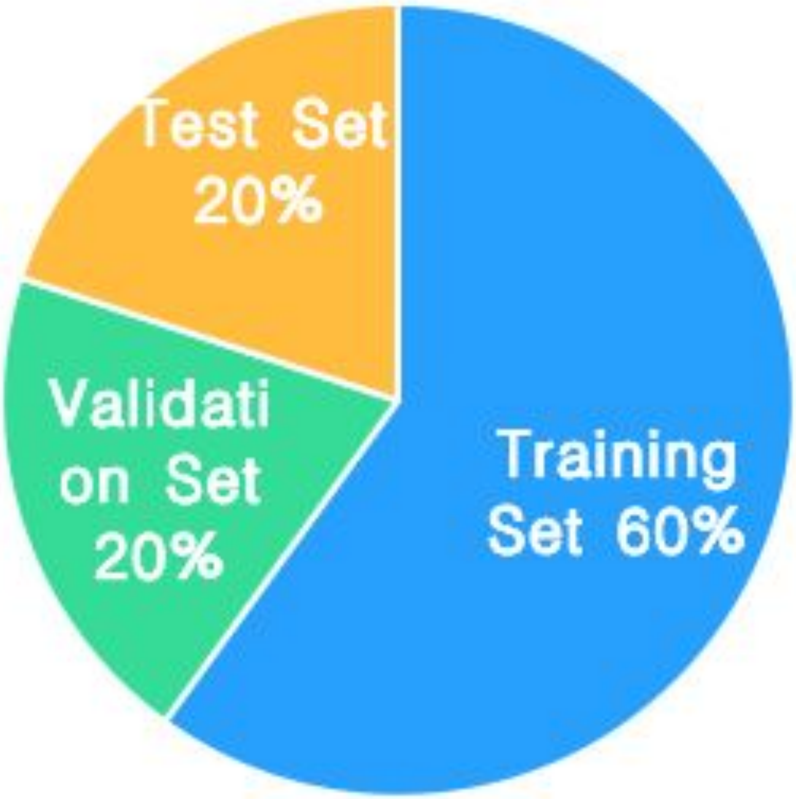
Here is a trimmed version of our data set:

	Potential	Reactions	Composure	Base stats	International reputation	Passing / Kicking	Dribbling / Reflexes	Wage(€:K)	Log Release clause(€:K)
0	88	74	92	398	1	70	81	34.0	11.015345
1	88	68	75	416	1	68	77	16.0	9.441452
2	85	59	57	400	1	58	77	0.5	9.239899
3	90	78	77	451	1	75	78	10.0	11.209114
4	82	75	81	405	1	75	80	65.0	10.718852
...	...	...	...	...	...	...	...	...	...
17511	70	60	45	346	1	55	62	0.5	7.170120
17512	69	54	56	348	1	57	62	0.6	6.748760
17513	67	56	54	334	1	54	60	0.5	6.284134
17514	78	73	72	384	1	56	63	0.9	9.568015
17517	71	70	66	334	1	50	48	0.5	7.170120

15762 rows × 9 columns

Training Set & Test Set

We split the original dataset and created a training set, a validation set, and a test set in the ratio of 6:2:2 to avoid overfitting and test the model's accuracy.



Model Training and Tuning

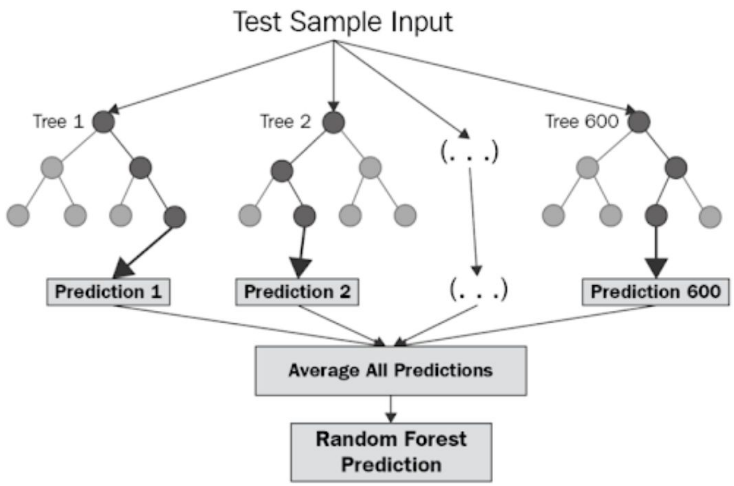
Model Selection

1. Random Forest Ression

2. XGBoost Regressor

3. Multilayer Perceptron Regressor

Random Forest Regression



Random forest is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.

It runs efficiently on large databases.

It can handle thousands of input variables without variable deletion.

It gives estimates of what variables are important in the classification.

predict outcomes based on diverse predictor variables. This ensemble method integrated predictions from multiple decision trees, each constructed using a random subset of features and data points, to enhance model accuracy and robustness.

Tune Hyperparameter

```
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 5, 10],
    'min_samples_split': [2, 5, 10]
}
```

```
Best Model Score: 0.9386381387828471
```

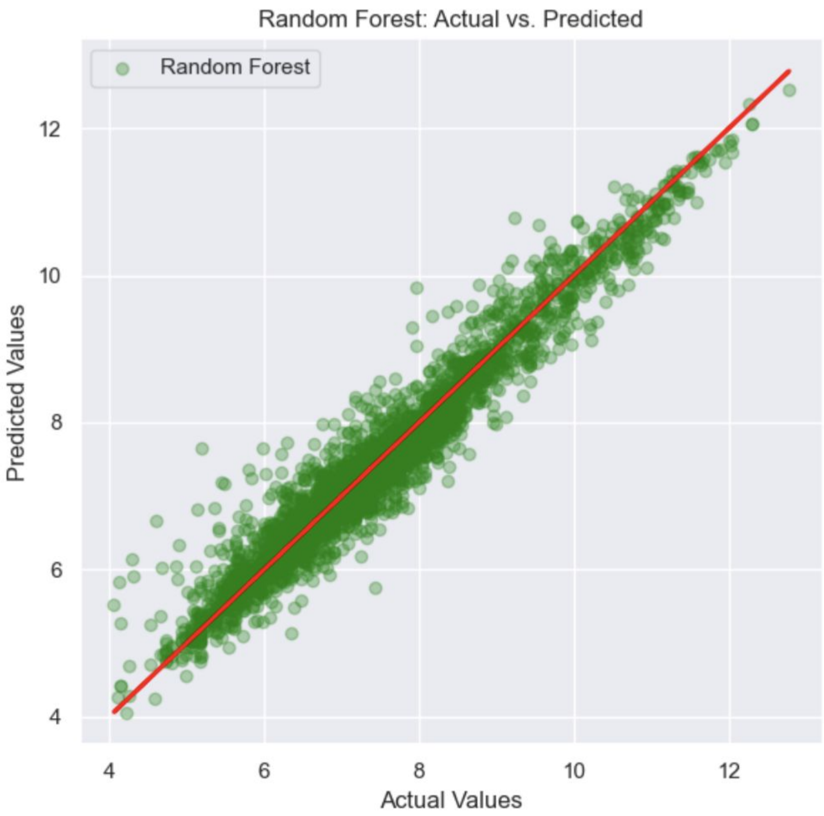
Results & Errors

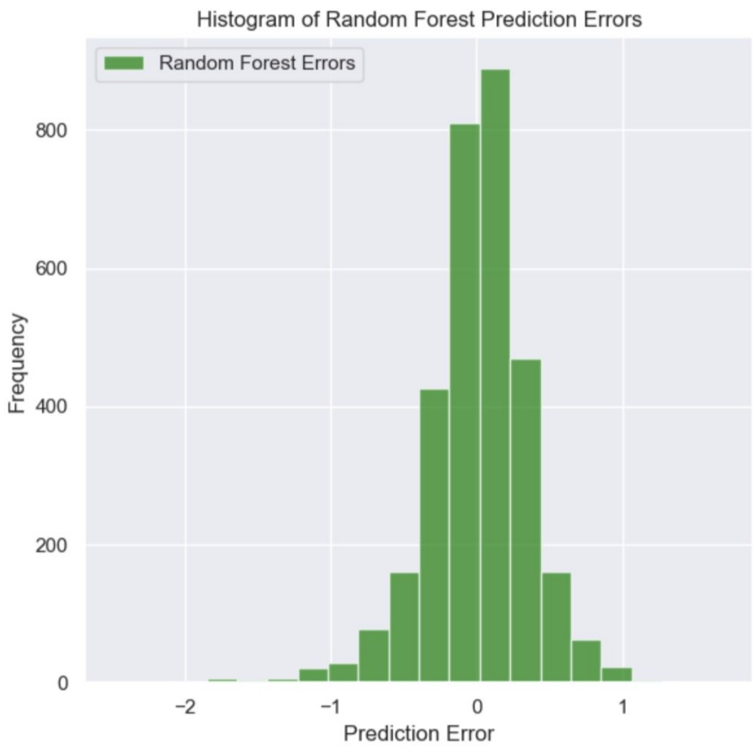
	Real Release Clause	Predicted Release Clause
10618	7.549609	7.537835
14019	6.481577	6.785472
16586	6.763885	6.888598
890	8.433812	8.413368
2407	9.553930	9.917857
...	...	...
12554	6.107023	5.893942
1274	7.244228	7.385077
1102	8.630522	8.743559
15322	9.277999	9.116128
13689	5.953243	6.072195

[3153 rows x 2 columns]

Mean Squared Error: 0.12473438540333973
R^2 Score: 0.9380135015685438

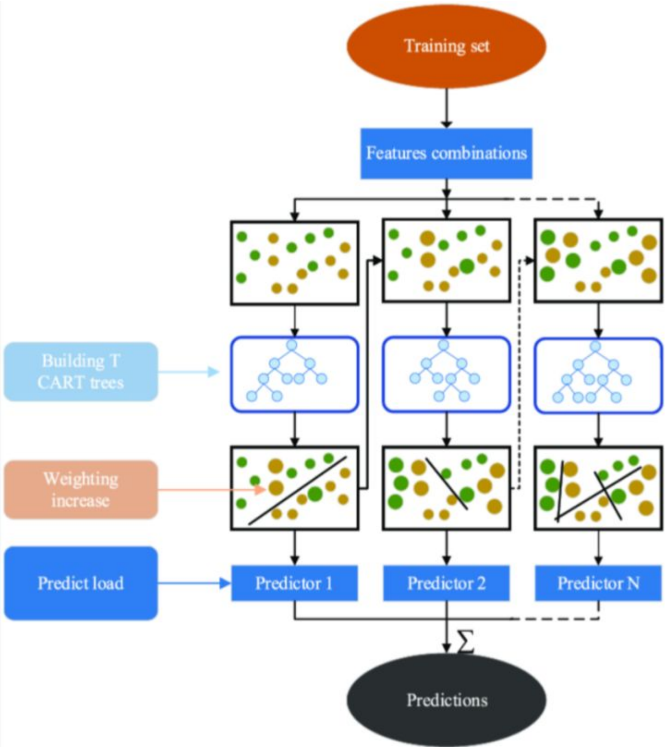
Results & Errors Visualization





XGBoost

- Extreme Gradient Boosting, iteratively builds a series of decision trees to make predictions. It optimizes the model's performance by minimizing errors at each step and combining the predictions of multiple weak learners into a strong model. XGBoost can handle both classification and regression tasks and is robust against overfitting.



Tune Hyperparameter

```
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.2],
    'subsample': [0.7, 0.8, 0.9],
    'colsample_bytree': [0.7, 0.8, 0.9]
}
```

```
Best Model Score: 0.9414046517932017
```

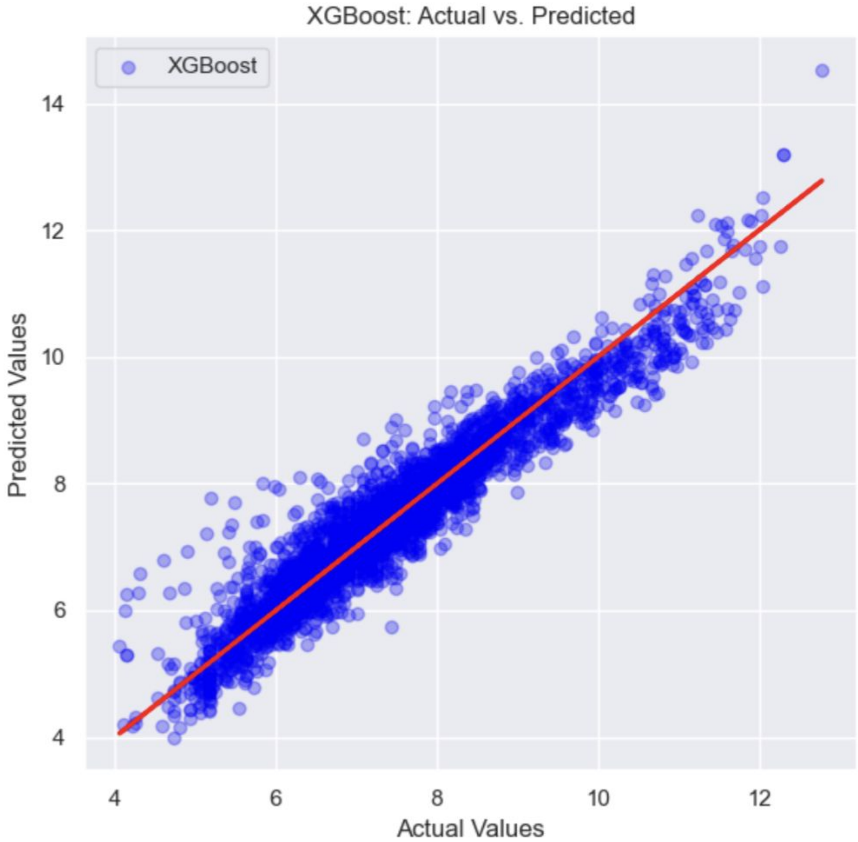
Results & Errors

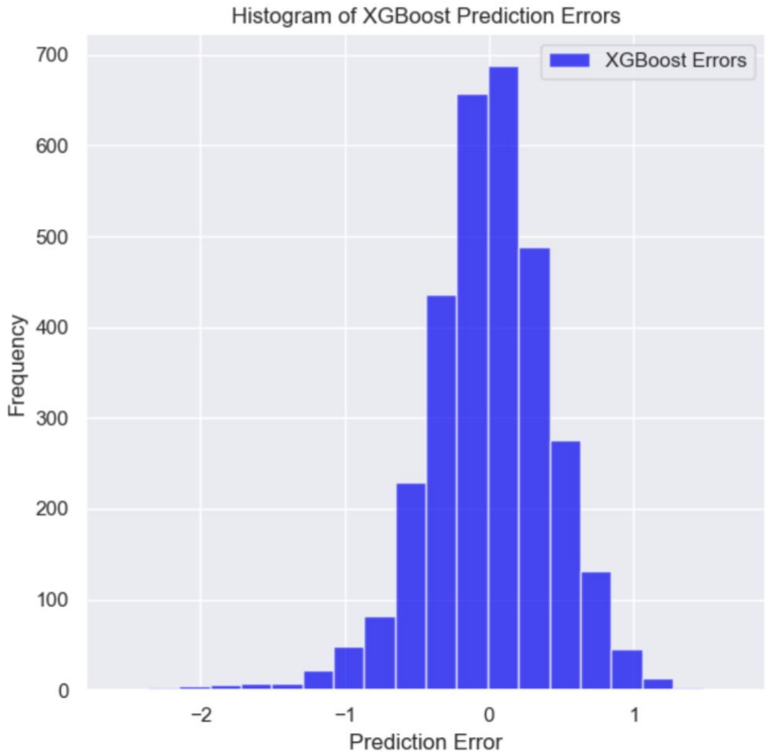
	Real Release Clause	Predicted Release Clause
10618	7.549609	7.532607
14019	6.481577	6.834077
16586	6.763885	6.953578
890	8.433812	8.383838
2407	9.553930	9.798126
...	...	...
12554	6.107023	5.985600
1274	7.244228	7.510485
1102	8.630522	8.734858
15322	9.277999	9.213490
13689	5.953243	6.080161

[3153 rows x 2 columns]

Mean Squared Error: 0.11607189882825057
R^2 Score: 0.9423183066049656

Results & Errors Visualization



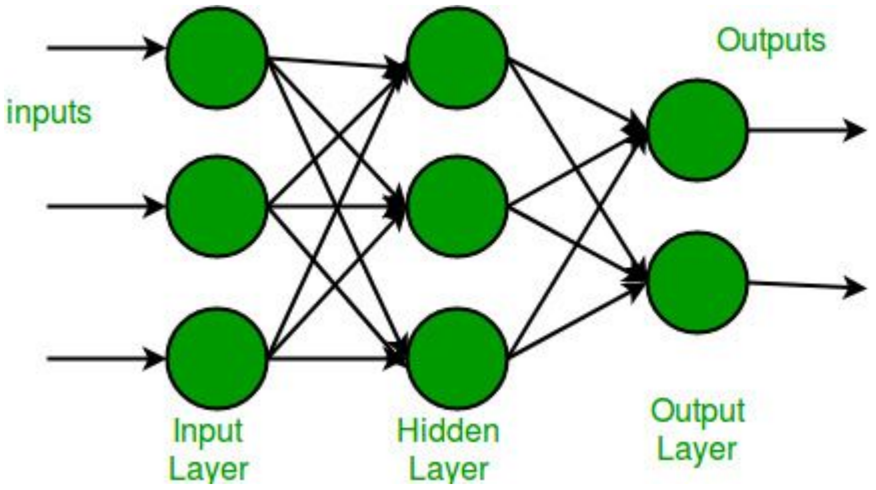


MLP

A multi-layer perceptron (MLP) is a form of artificial neural network that features several layers of neurons.

These neurons often employ nonlinear activation functions, enabling the MLP to capture intricate data patterns.

Their ability to model nonlinear relationships makes MLPs valuable tools in machine learning, suitable for various applications including classification, regression, and pattern recognition.



Tune Hyperparameter

```
{'activation': 'relu', 'alpha': 0.001, 'hidden_layer_sizes': (100, 50), 'learning_rate_init': 0.001, 'solver': 'adam'}
```

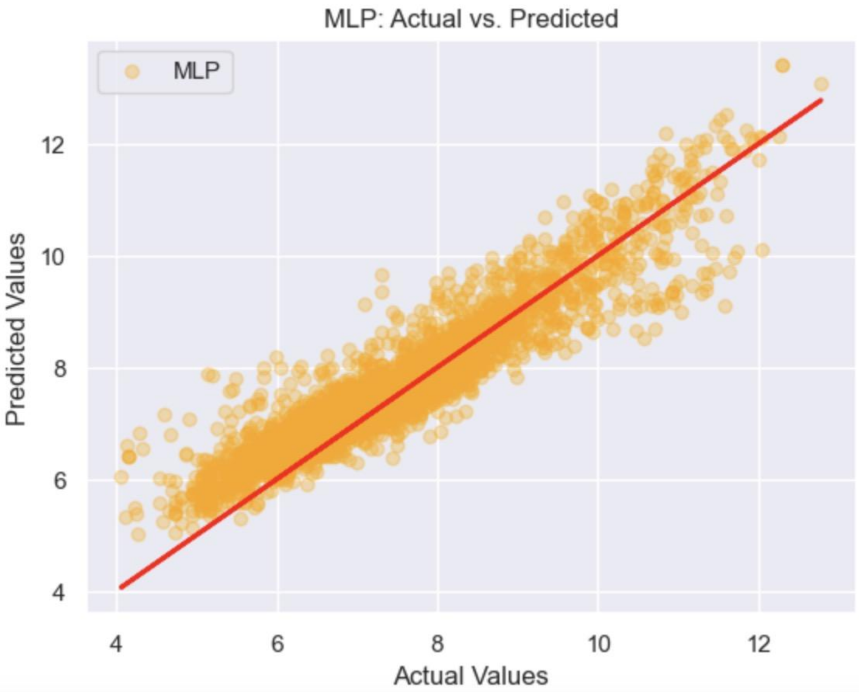
```
Best Model Score: 0.8461263141257123
```

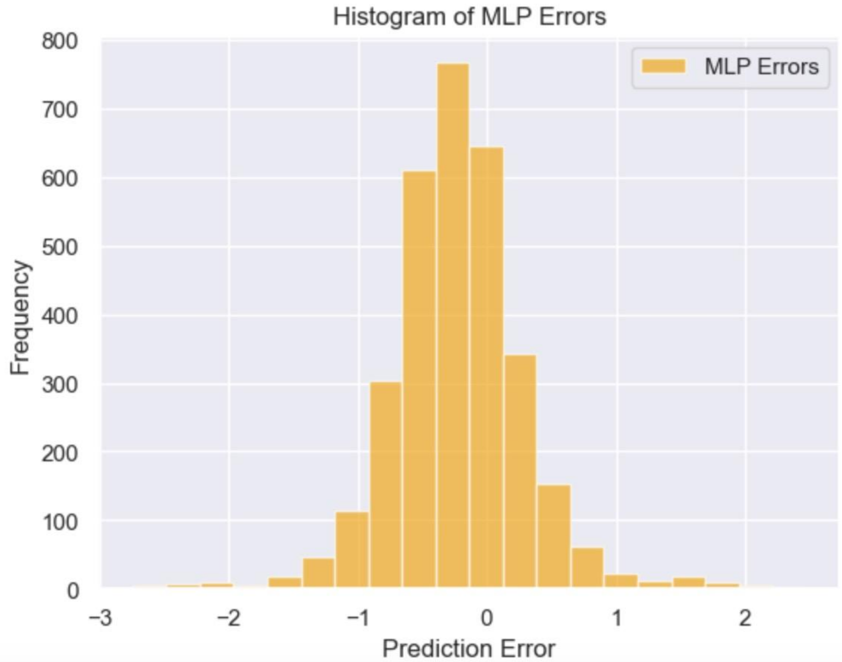
Results & Errors

```
       Real Release Clause  Predicted Release Clause
10618             7.549609                  7.509375
14019             6.481577                  6.742514
16586             6.763885                  7.185037
890               8.433812                  9.206723
2407              9.553930                 10.068987
...                    ...                       ...
12554             6.107023                  5.896831
1274              7.244228                  7.696889
1102              8.630522                  8.667505
15322             9.277999                  8.543071
13689             5.953243                  6.340005

[3153 rows x 2 columns]
Mean Squared Error: 0.31987905283936025
R^2 Score: 0.8410367570821269
```

Results & Errors Visualization





Model Accuracy Summary

In light of this, we can see:

RFR: MSE 0.12

XG: MSE 0.11

MLP: MSE 0.31

Here is XG:

```
Percentage of predictions that exceed the actual release clause: 46.31%
Percentage of predictions that are below the actual release clause: 53.69%
```

Conclusion

- 1. Understand the the usefulness of the role in releasing clause in FC24, main objective: predicting it in the game for finding players with underrated release clause (which also applies to real-world)
- 2. Challenges: large quantity of data with attributes, hard to find what's important. Hard to find suitable dataset.
- 3. Tried different models such as LASSO R, and MLP. Found RFR and XG to be the best.
- 4. Future challenges: limited applications to both gaming and real world due to limited resources but enough to provide a general sense what our predictions mean
- 5. Need to find more accurate data from the companies and teams

We’d like to thank all the TAs and our Professor Duan for all the good work that you have done for us!

who completed what

Yicheng Shi: data preprocessing, feature engineering, model tuning, report

Guangzhou Cai: data preprocessing, model training, report

Haotian Yang: data collection, visualization, model selection, report