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

Yicheng Shi
University of Rochester
Rochester, Ny, USA
yshi40@u.rochester.edu

Haotian Yang
University of Rochester
Rochester, NY, USA
hyang57@u.rochester.edu

Guangzhou Cai
University of Rochester
Rochester, NY, USA
gcai2@u.rochester.edu

ABSTRACT

The purpose of this project is to predict the release clause of soccer players in FC 24. Datasets were extracted by web scraping from a third-party FC 24 statistics website. The datasets were used to provide a comprehensive and in-depth analysis of release clauses and build effective models for further utilization. Through data preprocessing, including logarithm and encoding strategies to deal with target, categorical features, and missing values, three machine learning models (Random Forest Regressor, XGBoost Regressor, and Multilayer Perceptron Regressor) were compared, and their performance was evaluated using R-squared value and mean squared error. Based on the comparison, the XGBoost Regressor is superior to other models in predicting release clauses. Higher accuracy was achieved by hyperparameter tuning.

KEYWORDS

Random Forest, XGBoost, MLP, Machine Learning, Soccer, Release Clause, FC 24

1 INTRODUCTION

In FC 24, data-driven strategies can offer accurate guidance for game players in making decisions about budget spending in manager career mode. Machine learning models serve as valuable tools for accurately predicting players' release clauses, thereby assisting game players in determining whether the deal to pay the release clause to hire a player is economical. The datasets for this project primarily include the player's basic information and attributes within the game, encompassing over 15,000 entries, which include 61 numerical features and 3 categorical features. The primary motivation is to facilitate quicker and more informed trade decisions in manager mode and to provide a comprehensive view of identifying players with underrated release clauses. The insights from the underlying data patterns and regression models derived from this project are intended to contribute not only to decisions made by game players but also to the real-world applications concerning the connection between game attributes and players' actual release clauses.

, ,

2 DATA PREPROCESSING

2.1 Dataset Preparation

The dataset was sourced from the FC 24 Third-Party Data website, utilizing the most recent update available. This comprehensive dataset encompasses data pertaining to over 15,000 football players, characterized by 64 distinct attributes. These attributes include 61 numerical features and 3 categorical features, offering a broad spectrum of information relevant to player analytics.

2.2 Goalkeepers Data Exclusion

In preparing the data for analysis, the step involved the exclusion of all goalkeeper-related attributes. Given that goalkeepers represent a relatively small fraction of the dataset, their exclusion allows for a more focused analysis on outfield players and more accurate model results.

2.3 Release Clause

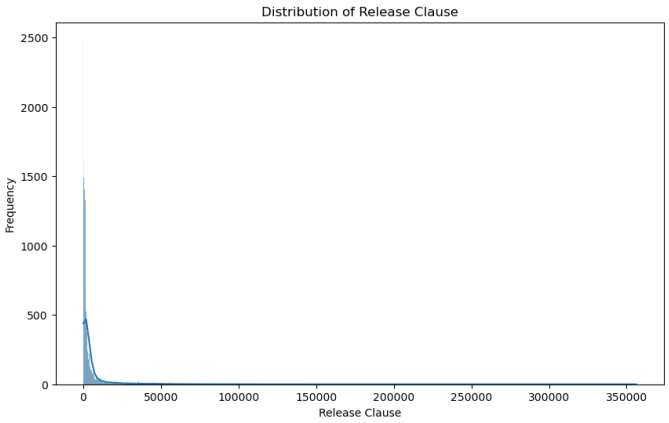

Figure 1: Distribution of Release Clause Value

2.3.1 Distribution of Release Clause. An initial examination of the Release Clause values across the dataset revealed a distribution skewed by outliers. The analysis of quartiles indicated that the first quartile (Q1) is positioned at €481, while the third quartile (Q3) reaches €3,300. Notably, the dataset included instances with a Release Clause value of zero, which were considered anomalies for the purposes of predictive modeling.

2.3.2 Free Agent. Free agent players don't have contracts with any clubs which means that their release clauses are zero. To maintain the integrity of the model and avoid undefined calculations, all

data entries of free agent players were removed. This strategy was guided by the aim to focus on meaningful, actionable data that reflects financial transactions within the player market.

2.3.3 Logarithmic Transformation. Given the wide range of Release Clause values and the presence of significant outliers, a logarithmic transformation was applied to this variable. This transformation is a standard technique for reducing skewness in a dataset, making the data more amenable to statistical modeling by decreasing the influence of outliers. Consequently, this approach enhances the predictive performance of the model by stabilizing variance and normalizing the distribution of the Release Clause values.

2.4 Process Categorical Features

In the dataset, three categorical features were initially identified: Foot, Best Position, and Team and Contract. Given the broad and diverse range of categories encompassed by the "Team and Contract" feature, which could introduce a high degree of dimensionality and potential model complexity, it was decided to exclude it from the analysis. The "Foot" and "Best Position" features were retained, applying one-hot encoding to these variables. This method transforms categorical variables into a binary vector format, where each category is represented by a vector with one '1' and the rest '0's. One-hot encoding is advantageous as it prevents the model from implying any natural ordering among categories, which is crucial for nominal data where such hierarchies do not exist.

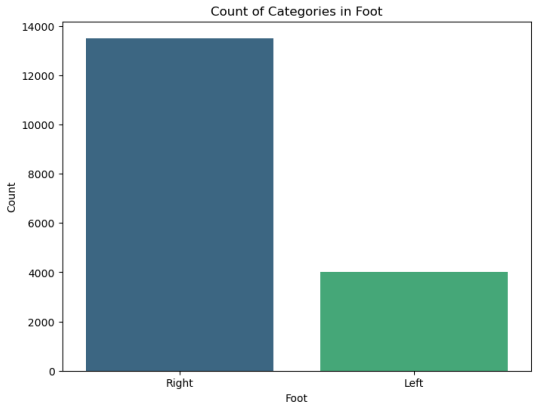

Figure 2: Count of Categories in 'Foot' Feature

2.5 Correlation Matrix and Feature Selection

A correlation matrix heatmap was used to visually illustrate the interdependencies between various features within the dataset. This matrix was rendered using advanced visualization tools in Python, with colors ranging from red to blue to represent negative to positive correlations, respectively. By presenting the correlation coefficients in a clear, concise manner, the heatmap enabled the quick identification of which variables were most strongly associated with each other. This analysis was crucial for determining potential predictors and refining the predictive models, thus ensuring that the approach was both data-driven and efficient.

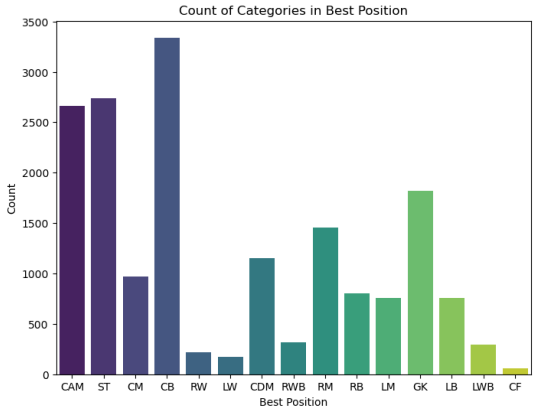

Figure 3: Count of Categories in 'Best Position' Feature

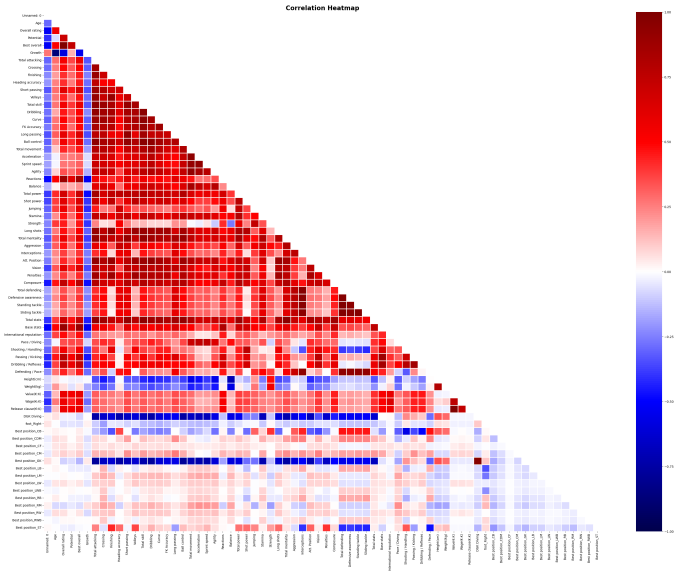

Figure 4: Correlation Heatmap

Based on the correlation values from the row of the Release Clause in the dataset, the variables most closely related to the Release Clause were identified. These include Wage (0.76), International Reputation (0.56), and Potential (0.57). Eight variables with the highest correlation were selected for further analysis, namely Potential, Reactions, Composure, Base Stats, International Reputation, Passing/Kicking, Dribbling/Reflexes, and Wage (expressed in €K). This selection prioritizes features that are strongly linked to the Release Clause, aiming to enhance the accuracy and reliability of the predictive model.

2.6 Feature Visualization

Figure 5 in the analysis featured box plots for all eight variables selected, providing a detailed visualization of their range and distribution. These box plots clearly displayed key statistics such as the

median, quartiles, and potential outliers, offering a comprehensive overview of each variable's behavior within the dataset.

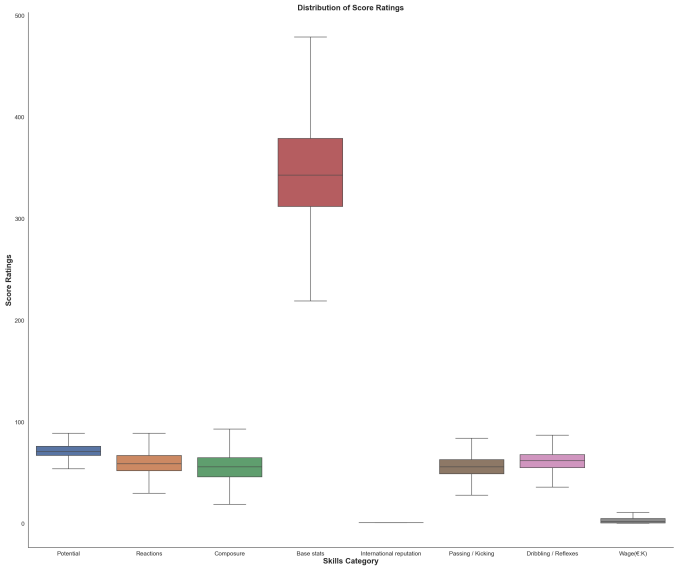

Figure 5: Distribution of Score Ratings

Figure 6 presented a heatmap of these eight variables, illustrating the correlation between each pair. This heatmap effectively highlighted the strength and direction of relationships, with color intensities representing the degree of correlation, thus allowing for easy identification of highly correlated variables.

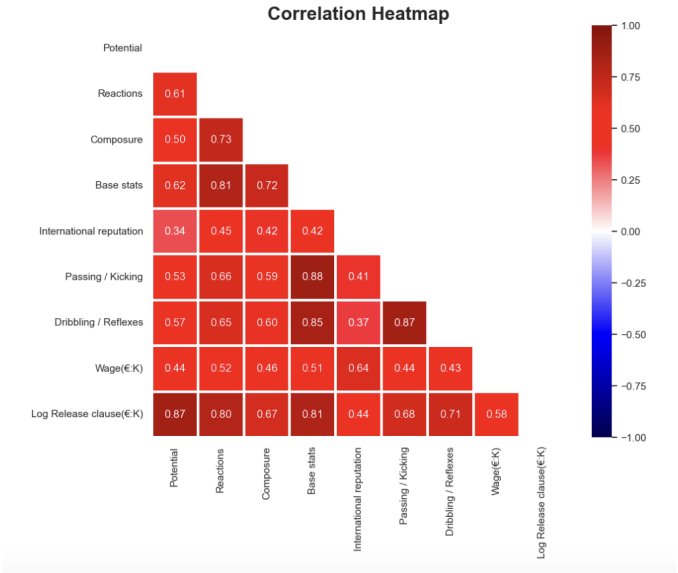

Figure 6: Seletced Correlation Heatmap

Figure 7 depicted the exact mathematical distribution of two selected variables from the set of eight. This figure employed probability density functions to show how the data points are distributed across the range of values, providing insights into the underlying patterns and tendencies in the data, such as skewness or kurtosis. These distributions are crucial for understanding the characteristics of individual variables and for modeling their interactions in predictive analytics.

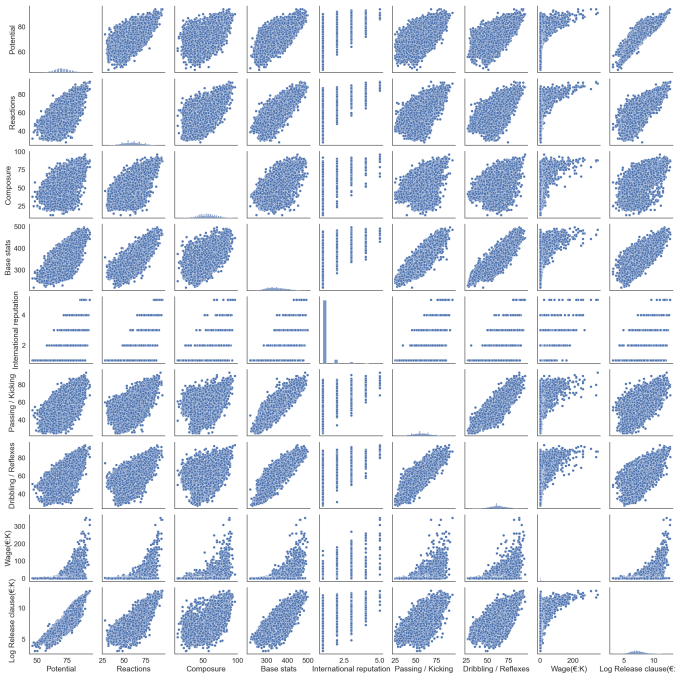

Figure 7: Density

Finally, Figure 8 was the trimmed version of the dataset, which was used to train the models.

	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

Figure 8: Trimmed Dataset

3 MODEL TRAINING AND TUNING

3.1 Model Selection

The initial model selection phase was a complex one in that not only was the model adaptability taken into consideration but also was the model interpretability. By model adaptability, it meant whether the selected model was suitable for transforming the incoming datasets to be fitted correctly and processed seamlessly with our goal. There were two big areas for data selection, namely

the categoricals model and linear regression models. For instance, categorical models included logistic regression and random forest classifier while linear regression consisted of support vector regression and linear regression. The initial proposition was to only choose those from the regression models since those were designed for prediction statistical numbers. However, after careful reconsideration, the initial thought was ditched and replaced with an intention to meticulously choose one categorical model as well as a regression model. The reason behind this action was rather an adoption of a more precise interpretation and perception of the datasets selected from the game, FC24, due to their various complexity levels such as resource locations, designs of dataset layouts, and availability of online publications. Upon inspecting several datasets acquired from the official FC24 website, the most presentable datasets were selected with most clarity in areas such as column definitions for player attributes and convincing large quantities of player database for monotonicity. Datasets contained more than 23 attributes imported from real-world play attributes such as 'Foot', 'Overall Rating', 'Potential', 'Dribbling / Kicking', 'Wage', and etc. In the data processing phase, all of these attributes were carefully tested with a correlation matrix as well as a heat map generated by 'sns.heatmap.'

With this model selection in mind, the initial machine learning model selection commenced with incipient thought of choosing: Lasso Regression, Random Forest Regression, MLP, and XG Boost. Three models were distributed among the members of the team simultaneously for the initial phase of model development which was composed of fitting the trimmed data (multiple datasets created in different years) in disparate models, training the models, fine-tuning and visualizing them.

3.2 Lasso Regression

Least absolute shrinkage and selection operator (Lasso) is a type of linear regression technique used for feature selection and regularization. It operates by adding a penalty term to the standard linear regression objective function, which forces the sum of the absolute values of the regression coefficients to be less than a constant. Lasso regression is particularly useful when dealing with high-dimensional datasets where the number of features exceeds the number of observations. It requires a dataset with both input features and corresponding output values for training. Lasso regression predicts continuous numerical outcomes, making it suitable for regression tasks. However, it has limitations, such as difficulty in handling multicollinearity among predictor variables and potential instability when the number of predictors is large compared to the number of observations. Additionally, lasso regression tends to shrink coefficients to zero, leading to potential loss of predictive accuracy.

With the dataset fit into this model, the initial optimized Mean Squared Error (MSE) on the validation set was calculated to be 2.76 million, which signified a significant fitting and model selection error although it appeared to be a perfect regression model for predicting numerical values. Contrary to its purported sufficiency, the Lasso test set proved to fluctuate around zero, further signifying an implausible prediction on real-world player release clause values. As a result, this method was abandoned despite careful data fitting and meticulous calculations.

3.3 Random Forest Regression

3.3.1 Explanation. A Random Forest Regression model was deployed to predict outcomes based on diverse predictor variables. This ensemble method integrates predictions from multiple decision trees, each constructed using a random subset of features and data points, to enhance model accuracy and robustness. The Random Forest algorithm mitigates overfitting through its averaging technique, where the individual predictions from each of the trees (up to 600 in this case) are averaged to produce a final output[1].

3.3.2 Model Training and Tuning. The number of estimators was tuned using values from a list that included 100, 200, and 300. The maximum depth was also adjusted from a selection of 0, 5, and 10, and the minimum sample split was tuned with options of 2, 5, and 10. The score for the best model tuned is 0.9386.

3.3.3 Prediction Result. Figrue 9 displayed a comparison between the predicted release clauses and the actual release clauses for the test set. The Mean Squared Error (MSE) was calculated to be approximately 0.1247, while the R-squared score was around 0.9380. Figure 10 illustrated the relationship between the actual and predicted values, including all data points from the test set. A red line represents the ideal scenario where predictions perfectly match the actual values. Data points closer to this red line indicate more accurate predictions, while points further from the line represent less accurate predictions. The third graph is a histogram depicting the distribution of prediction errors from the Random Forest model. It categorizes errors from -2 to 1, where negative values represent

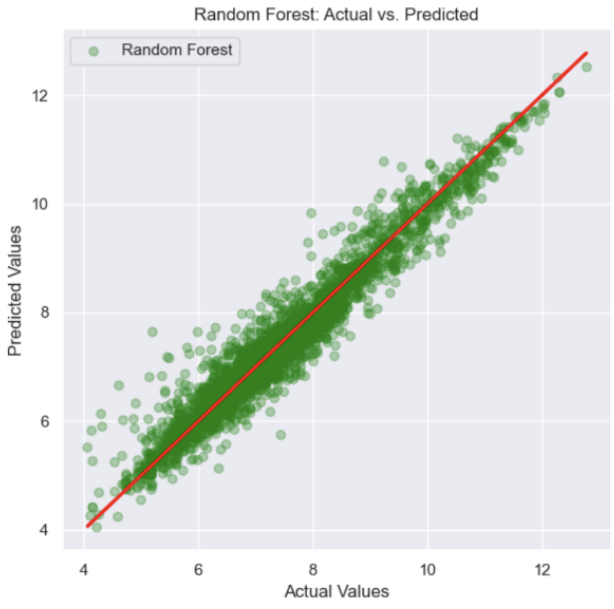

Figure 9: Random Forest: Actual vs. Predicted

underestimations and positive values overestimations of the actual release clauses. Most predictions cluster around zero, indicating high accuracy, with the distribution showing a balanced spread of errors around the perfect prediction line.

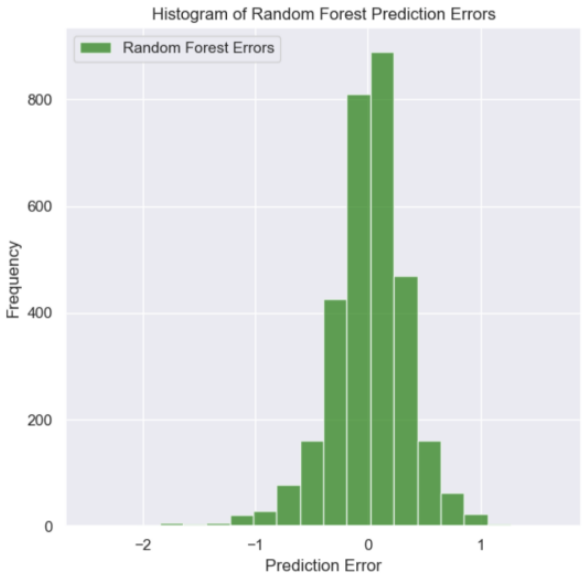

Figure 10: Random Forest Prediction Errors

3.4 XGBoost Regression

3.4.1 Explanation. XGBoost, or Extreme Gradient Boosting, is a powerful machine learning algorithm that 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. However, it may require careful tuning of hyperparameters, and its interpretability can be challenging due to its complex nature[2].

3.4.2 Model Training and Tuning. Using the innate functions in Python, the 'xgboost' algorithm was imported and trained in a timely manner. The parameters necessary for this model to function are listed in Figure 11. 'N_estimators' defines how many trees to include in the model, while 'max_depth' determines the depth of each tree in the boosting process, controlling the complexity of the trees. 'Learning_rate' is used to prevent overfitting and scales the contribution of each tree. 'Subsample' is used for fitting the individual trees and controls the sampling of the data points. Finally, 'colsample_bytree' is used to randomly sample features for each tree, controlling the sampling of features.

3.4.3 Prediction Result. Figure 11 illustrates the best parameters, as the model's hyper-parameter function selected after evaluating options of 100, 200, and 300. The Mean Squared Error (MSE) was found to be approximately 0.11, which was profoundly promising. The 'Best_model_score' was calculated through 'grid_search.fit(X_train, y_train)' and resulted in a score of 0.94, marking another advancement from the previous calculations.

```
Best parameters found:  {'colsample_bytree': 0.9, 'learning_rate': 0.1, 'max_depth': 5, 'n_estima
mple': 0.7}
```

Figure 11: XGBoost Regressor Best Parameters

Figure 12 illustrates the real release clause and predicted release clause using the test sets. The results were surprisingly similar, with an MSE of 0.11 and an $R\hat{2}$ of 0.94.

	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

XGBOOST

Figure 12: Sample Comparison: Actual Value vs. Predicted Value

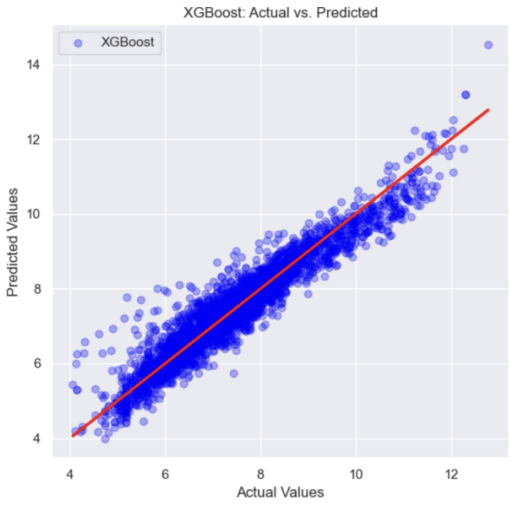

Figure 13: XGBoost: Actual vs. Predicted

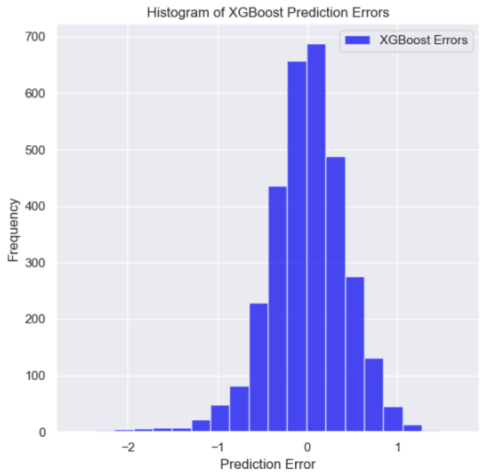

Figure 14: XGBoost Prediction Errors

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

Figure 15: MLP Best Parameters

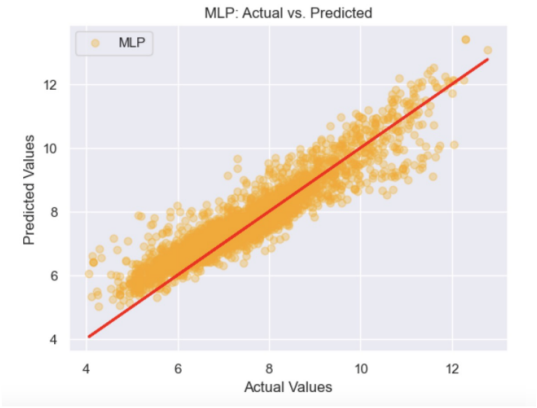

Figure 16: MLP: Actual vs. Predicted

3.5 Multilayer Perceptron Regression

3.5.1 Explanation. A multi-layer perceptron (MLP) is a type 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[3].

3.5.2 Model Training and Tuning. In the optimization of the machine learning model using Python's versatile library 'scikit-learn', the most effective parameters were identified to enhance the model's performance. The 'activation' function selected was 'relu' (Rectified Linear Unit), which introduces non-linearity to the model, allowing it to learn more complex patterns in the data. The parameter 'alpha', set at 0.001, regulates the amount of regularization applied, mitigating the risk of overfitting by penalizing larger weights. The 'hidden_layer_sizes' parameter was configured to (100, 50), specifying the architecture of the neural network with two layers containing 100 and 50 nodes, respectively, to process information hierarchically. The 'learning_rate_init' was set at 0.001 to determine the step size at each iteration while moving toward a minimum of the loss function, optimizing the convergence speed. Lastly, the 'solver' for weight optimization was chosen as 'adam', known for its efficiency in handling large datasets and its adaptive learning rate capabilities. These parameters collectively form a robust framework for the model, ensuring optimal learning and predictive performance.

3.5.3 Prediction Results. Figure 15 illustrates the best parameters. The MSE was found to be approximately 0.32, which was larger than that of other models. The 'Best_modeL_score' was calculated through 'grid_search.fit(X_train, y_train)' and resulted in a score of 0.85.

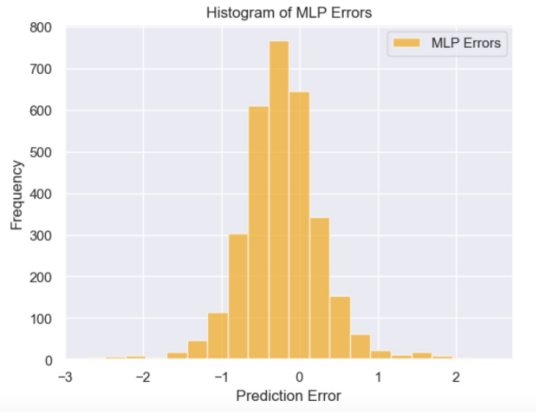

Figure 17: MLP Prediction Errors

4 CONCLUSION

In conclusion, this project has demonstrated significant advancements in predicting the release clauses of soccer players in FC 24 by utilizing machine learning techniques. Through meticulous data preprocessing and exploratory data analysis, the study successfully managed a dataset that encompassed over 15,000 players with varied attributes. Notably, the exclusion of goalkeeper-specific attributes and the application of logarithmic transformations streamlined the dataset, ensuring more accurate model performance by mitigating issues associated with skewed data distributions.

Among the three machine learning models evaluated—Random Forest Regressor, XGBoost Regressor, and Multilayer Perceptron Regressor—the XGBoost Regressor emerged as the most effective, delivering superior results in terms of both Mean Squared Error and

R-squared values. This outcome underscores XGBoost's robustness and its capability to handle complex, high-dimensional data while minimizing overfitting, attributes that are critically important in predictive modeling.

The project's findings are crucial not only for game players in FC 24, seeking to make informed decisions in manager career mode, but also offer insights that could translate into real-world applications. The correlation analysis and feature selection processes highlighted key predictors of release clauses, such as Wage and International Reputation, providing a deeper understanding of factors that influence player valuations.

However, the study also encountered limitations, particularly in the handling of multicollinearity and the model's ability to manage the sheer volume of predictors relative to observations. Future studies could explore alternative regularization techniques and consider ensemble methods that combine the strengths of different models to further enhance prediction accuracy.

In advancing this research, it would be beneficial to incorporate additional data sources and possibly real-time data to refine the models further. Additionally, exploring other predictive modeling techniques and expanding the feature set to include newly emerging metrics in player performance could offer new dimensions to this research area.

Overall, this project sets a foundational framework for leveraging machine learning in sports analytics, with promising directions for future enhancements that could significantly impact both virtual and real soccer player market dynamics.

5 DIVISION OF WORK

Haotian Yang worked on the initial idea scripting, data collecting and cleaning, data visualization, model selection, and report formatting. Guangzhou Cai worked on data preprocessing, data visualization, model training, and report writing. Yicheng Shi worked on data preprocessing, data visualization, feature engineering, model tuning, and report writing.

6 ACKNOWLEDGMENTS

We would like to express our heartfelt gratitude to Professor Duan for his invaluable guidance and support throughout our classes and this research project. His expertise and insightful feedback were instrumental in shaping both the direction and execution of this study. Professor Duan's lenient and supportive class policy, along with his kindness and approachability, greatly enhanced my academic experience. I am deeply appreciative of his commitment to fostering a nurturing and rigorous academic environment, and his dedication to advancing knowledge within our field.

7 BIBLIOGRAPHY

REFERENCES

[1] G. Biau, "Analysis of a Random Forests Model," *Journal of Machine Learning Research*, vol. 11, pp. 1234-1246, 2010.

[2] T. Chen and C. Guestrin, "XGBoost Documentation," 2021. [Online]. Available: https://xgboost.readthedocs.io/en/stable/python/python_api.html. [Accessed: Date].

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "scikit-learn: Machine Learning in Python," in *Neural Network Models (Supervised)*, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html. [Accessed: Date].