Key Determinants of Football Player Valuation: A Machine Learning Study

github.com/Samuelvermeulen/capstone-project-2025

Samuel Vermeulen
University of Lausanne

January 2026

Abstract

This study examines the determinants of professional football player market value using machine learning methods applied to Premier League data from 2018–2019 to 2022–2023. Performance statistics from FBref are combined with financial and physical attributes from FIFA-based datasets to construct a player-season panel. A linear baseline model is compared with Ridge Regression, Random Forest, and XGBoost using temporal validation. XGBoost achieves the best predictive performance, highlighting the importance of individual performance and availability, while club-level indicators add little explanatory power. The results illustrate both the strengths and limits of machine learning in football valuation.

1 Introduction

The valuation of professional football players is a central yet complex issue in sports economics, as market values depend on a combination of performance, physical characteristics, and contextual factors. This complexity makes player valuation a natural application for machine learning methods, which can capture non-linear relationships beyond traditional econometric models.

This project analyzes the determinants of player market value in the English Premier League between the 2018–2019 and 2022–2023 seasons. By combining performance data from FBref with financial and physical attributes from FIFA-based datasets, several models are compared, ranging from a simple linear baseline to advanced algorithms such as Random Forest and XGBoost. The study focuses on the role of playing time, goal-scoring, and player position, while emphasizing reproducibility, temporal validation, and model interpretability.

2 Research Context and Literature Review

We begin our analysis by formulating a clear research question: **Which characteristics improve the market valuation of a professional football player?**
Based on this question, we develop two hypotheses to guide our empirical investigation.

The first hypothesis is that **for players competing in top 10 clubs, the number of minutes played has a stronger positive impact on their market value**. The relevance of this hypothesis lies in the idea that playing for a highly visible club increases a player's exposure and perceived value. However, limited playing time in a top club may be more detrimental to valuation than regular playing time in a smaller club, as it can signal a lower competitive standing within elite teams.

The second hypothesis examines whether **goals are more relevant for forwards than for players in other positions**. This assumption is motivated by the specific role of forwards as primary finishers within a team. A forward who scores few goals may be perceived as underperforming relative to positional expectations, which could significantly reduce his market value, whereas goal-scoring may be less critical for midfielders or defenders.

Turning to the existing literature, research on football player valuation remains relatively limited. One

explanation is the absence of an objective or "true" market value for players: most available valuations are estimates or observed transfer fees, which may not accurately reflect a player's intrinsic value. This lack of ground-truth data can reduce the precision and reliability of machine learning models. Additionally, institutions that publish player valuations, such as *Transfermarkt*, do not disclose their valuation methodologies, as these models constitute a core part of their business. Furthermore, the use of their data for machine learning purposes is restricted due to concerns over competition.

The existing empirical literature primarily relies on **Random Forest** and **XGBoost** models, often incorporating many detailed performance statistics. Player positions are generally defined in broad categories (e.g. forward, midfielder, defender) rather than more granular roles such as striker or winger. The analysis conducted by *Idris and Ng (2025)* serves as a key inspiration for this project; however, this study aims to adopt a **less complex modeling framework** while focusing on a more interpretable set of explanatory variables.

3 Data treatment

3.1 Data source

The data used in this study originate from two distinct sources. First, performance statistics—such as goals, assists, and minutes played—are obtained from the FBref website[1], which provides detailed match-level and season-level data for professional football players. Data were collected for Premier League players across all positions (forwards, midfielders, defenders, and goalkeepers) and for each season from 2018 to 2023. This source offers rich performance information and constitutes a valuable basis for more advanced analyses.

Second, player market values and physical attributes—including preferred foot, height, and weight—are sourced from publicly available Kaggle datasets derived from the FIFA video game database (FIFA 19 to FIFA 23)[2]. This source is used due to the absence of a fully comprehensive and publicly accessible database of historical player market valuations.

Additional variables such as player name, club, and age are common to both datasets and are used as key identifiers to merge the two sources at the player-season level.

3.2 Treatment

Each FIFA season is carefully cleaned to retain only Premier League players. Non-informative elements (photos, logos, irrelevant statistics) are removed, and all variables are standardized across seasons, including currency, height, and weight. The five seasons are then merged into a single player-season dataset.

In parallel, FBref data are processed by position and season, stacked, and cleaned through minor spelling corrections and column filtering. This results in four consolidated datasets (defenders, midfielders, forwards, goalkeepers) containing key performance metrics such as matches played, minutes played, goals, assists, and starts.

The two sources are subsequently merged at the player-season level. FIFA data provide market value and physical characteristics, while FBref data capture on-field performance. When a FIFA match is unavailable, the corresponding financial and physical variables are left missing to preserve data integrity.

Finally, all observations from the 2023–2024 season are retained, while earlier seasons are restricted to player-season records with valid FIFA matches, removing duplicates and unmatched entries. The resulting dataset is a clean and consistent panel with one row per player per season and all variables required for valuation analysis. The full data construction pipeline is fully reproducible and documented in the Dataset_creation directory.

[1] See Bibliography.
[2] See Bibliography.

4 Methodological Design and Implementation Strategy

4.1 Baseline model implementation

In the methodology section, we first aim to construct a **baseline model**, which will serve as a reference point for comparison throughout the remainder of the study.

This baseline model is intentionally simple and relies on only two explanatory variables: **age**, which is a numerical variable, and **playing position**, which is a categorical variable. To address the issue of non-numerical data associated with player position, we apply **one-hot encoding**, transforming the categorical variable into four binary indicators corresponding to the main playing positions: **Forward, Midfielder, Defender, and Goalkeeper**. This encoding allows the position information to be incorporated into the regression framework and will also be useful in subsequent models.

Before estimating the baseline model, we apply a logarithmic transformation to the dependent variable, namely the player's market value. Specifically, we use the `np.log1p()` function, which computes the natural logarithm of (1 + x) and avoids potential issues related to zero-valued observations. This transformation is necessary given the strong right-skewness of the market value distribution, which ranges from approximately 60,000 to 150 million. Applying the logarithmic transformation substantially reduces skewness and results in a distribution closer to normality, improving the suitability of linear regression techniques.

The baseline model is then estimated using **ordinary least squares (OLS)** via the *scikit-learn* linear regression implementation.

Finally, we evaluate the performance of the baseline model. As expected, the results are weak due to the deliberately simple structure of the model. The baseline yields a **root mean squared error (RMSE) of approximately 16.4 million**, which is relatively large, and an **R-squared value close to zero**, indicating very limited explanatory power. These poor performance metrics motivate the introduction of additional explanatory variables and the implementation of more complex modeling approaches in the subsequent stages of the analysis. The table 1 is in the appendix.

The ordinary least squares (OLS) model exhibits an algorithmic complexity of $O(n \cdot p^2)$, where n denotes the number of observations and p the number of features—limited here to only five after encoding (age and four positional indicators). This computational simplicity is deliberate and defines the baseline nature of the model. However, it also largely explains its weak predictive performance: constrained to linear relationships and unable to model nonlinear effects or interactions between variables, the model suffers from high bias despite its low variance. These structural limitations prevent it from adequately capturing the complexity of football player valuation and therefore fully justify the subsequent introduction of richer explanatory variables and more advanced machine learning algorithms.

4.2 Advanced feature engineering

This section introduces major improvements to our dataset through an extensive data cleaning and feature engineering process.

We begin with an in-depth exploratory analysis of the dataset. The 2023–2024 season is excluded from the analysis, as it contains substantial missing information—most notably the absence of market value data—which renders it unsuitable for modeling. This exploratory phase also confirms the validity of the preprocessing steps implemented in the baseline model, particularly the encoding of player positions, and allows us to verify the quality and consistency of this encoding.

Next, we handle missing values. Given the very limited number of missing observations, imputation is carried out using the median and applied exclusively to the training set in order to prevent any form of data leakage. We then construct new performance ratios—goals per minute played and assists per minute played—which provide a normalized measure of offensive contribution that is comparable across players with different playing times. To reduce skewness and ensure consistency with the transformation applied to the target variable, these ratios are log-transformed. The transformation is defined as follows:

$$\text{Goals\_per\_minute\_log} = \log(1 + \text{Goals}/\text{Minutes\_played})$$

$$\text{Assists\_per\_minute\_log} = \log(1 + \text{Assists}/\text{Minutes\_played})$$

This formulation avoids division by zero while preserving interpretability and improving the statistical properties of the variable, thereby facilitating more stable learning by the model.

We subsequently incorporate club-level information through a structured encoding strategy designed to capture differences in prestige and competitive exposure. The 27 Premier League clubs included in the dataset are classified into four hierarchical categories based on their average league position over the 2018–2023 period. The first category comprises top clubs (10 teams), identified using European performance rankings from *European Club Rankings* over the 2019–2023 period. The second category includes middle-table clubs (8 teams) with average league positions between 8th and 14th. The third category consists of relegation-battle clubs (7 teams) with average league positions of 15th or lower. The final category groups the remaining clubs into a residual "other" category.

Each category is encoded using dummy variables, enabling the model to account for systematic differences in competitive level and institutional prestige across clubs. This hierarchical encoding provides a nuanced representation of club effects and directly supports the empirical evaluation of the first hypothesis regarding the impact of playing for top clubs on player market value.

We further confirm the appropriateness of the logarithmic transformation applied to market value, which reduces skewness from approximately **3 to nearly 0**, resulting in a distribution much closer to normality.

All preprocessing steps and feature engineering procedures are then integrated into a **reproducible data pipeline**, ensuring consistent treatment of both training and test datasets. After validating this pipeline, we obtain a final set of **25 explanatory variables** ready for model estimation.

Finally, to guarantee full reproducibility, all processed datasets (`X_train`, `X_test`, `y_train_log`, `y_test_log`), as well as pipeline metadata (including club classifications and imputation values), are saved in the `data/processed/` directory and fully documented in a dedicated JSON file.

The pipeline is implemented manually rather than relying on scikit-learn's Pipeline or ColumnTransformer abstractions. This design choice ensures full transparency at each stage of data processing and enables the explicit storage of key metadata—such as the list of elite clubs (`top_clubs`) and imputation parameters (`imputation_dict`)—in the `feature_metadata.json` file, thereby guaranteeing reproducibility. While more verbose, this approach facilitates detailed inspection and controlled deployment.

4.3 Comparison of the 4 models

This section is dedicated to the implementation of the four models used in the analysis. We begin by loading the datasets generated through the preprocessing pipeline described in the previous section.

To recall, the objective is to estimate a player's **market value** (log-transformed) using **25 explanatory variables**. The training set covers the seasons from **2018–2019 to 2021–2022**, comprising **1,690 players**, while the test set corresponds to the **2022–2023 season**, with **382 players**.

In addition to the baseline linear regression model presented in Section 4.1, we implement three additional models.

Ridge Regression.
Ridge regression extends the baseline linear model by incorporating L2 regularization, which penalizes large coefficient magnitudes and thereby reduces the risk of overfitting. An intercept term is estimated during training and can be interpreted as the expected market value of a hypothetical player with zero minutes played, no goals, no assists, and no club-specific effects. This specification enables an evaluation of whether a regularized linear framework is sufficient to capture the primary determinants of player valuation. In accordance with the project roadmap, the Ridge model is implemented with a regularization parameter set to $\alpha = 1.0$, explicitly defined in both `models.py` and `model_training.py`.

Random Forest.
The Random Forest model represents the first non-linear approach applied in this study. It is composed of an ensemble of decision trees whose predictions are aggregated through averaging. The model is configured with 100 trees and a maximum depth of 10, a standard setup that balances complexity and generalization. This relatively large depth allows the model to capture interactions between features, such as age and playing position. Random Forest is particularly well suited for modeling non-linear relationships and managing correlated explanatory variables, which are likely present in our dataset. The model is executed with `n_jobs = -1`, as explicitly defined in `models.py` and `model_training.py`, enabling parallel computation across all available CPU cores. This configuration substantially reduces

training time while preserving reproducibility through a fixed random seed.

XGBoost.
XGBoost is a gradient boosting model that constructs decision trees sequentially, with each successive tree correcting the errors of its predecessors. Model complexity is controlled using a learning rate of 0.1 and a maximum tree depth of 5, promoting a structure of multiple shallow trees rather than a few highly complex ones. Within our framework, XGBoost represents the most flexible and potentially highest-performing model, making it the natural benchmark for maximum predictive accuracy. Although the parameters `objective='reg:squarederror'` and `eval_metric='rmse'` are not explicitly set in the `XGBRegressor()` call, they correspond to the default values for XGBoost version ≥ 1.7 (as specified in `environment.yml`) and are documented in comments within `models.py` and `model_training.py`.

Before training the models, we address issues related to **temporal feature availability**. A representative example is Leicester City, which does not appear in the 2022–2023 dataset due to relegation. This creates a mismatch in feature space between the training and test sets. To resolve this issue, we implement an `align_features` function that ensures consistent feature matrices across datasets.

Finally, all models are trained and evaluated using a common set of performance metrics: **Root Mean Squared Error (RMSE)**, **Mean Absolute Error (MAE)**, **R-squared**, and **Mean Absolute Percentage Error (MAPE)**. In addition, we employ **cross-temporal validation** to re-estimate model performance and assess their stability over time. Model stability is evaluated using the **mean R-squared** across temporal folds, as well as its **standard deviation**, which serves as an indicator of performance consistency. The comparative results of these models are discussed in detail in Section 7 of this report.

4.4 Optimization of XGBoost hyperparameters via RandomizedSearchCV

The objective of this section is to improve the performance of the XGBoost model by optimizing its hyperparameters using **RandomizedSearchCV**.

To achieve optimal performance, the XGBoost model is fine-tuned through **50 iterations of randomized search**, targeting eight key hyperparameters that govern model complexity, learning dynamics, and regularization strength. Model validation is performed using a **3-fold TimeSeriesSplit**, which strictly preserves the temporal order of the data to prevent information leakage and ensure robust out-of-sample evaluation. Details of all eight hyperparameters are provided in table 2 of the appendix.

The optimized model is assessed using the same performance metrics as in the previous section—**RMSE, MAE, and R-squared**—both in the log-transformed space and after back-transformation to the original euro scale. Its performance is directly compared with that of the default XGBoost model presented in Section 4.3, enabling a clear quantification of the improvements achieved through hyperparameter tuning. The optimal hyperparameter configuration is automatically saved to `results/hyperparameter_tuning/best_params_TIMESTAMP.txt` via `random_search.best_params_`.

Finally, the optimized model is saved in **Joblib** format to ensure reproducibility and facilitate future deployment or further analysis.

5 Implementation Analysis and Algorithmic Evaluation

5.1 Data pipeline implementation

This section is dedicated to the implementation of the data processing pipeline.

We first developed the `data_loader.py` module, which is designed to automate the entire data preparation workflow. This module includes five core functions: `load_raw_data()` for loading the source CSV file, `clean_data()` for handling outliers and data inconsistencies, `explore_data()` for exploratory data analysis, `split_data_temporally()` for performing the temporal train–test split, and `save_processed_data()` for persisting the processed datasets. An orchestration function, `get_data_pipeline()`, coordinates the sequential execution of these operations to ensure a consistent and reproducible workflow.

The temporal split is then performed using the `split_data_temporally()` function. This procedure divides the data into a training set covering the **2018–2019 to 2021–2022 seasons**, and a test set corresponding to the **2022–2023 season**, strictly respecting the chronological order of the data.

In addition, we integrate a dedicated logging and reporting module that documents each stage of the pipeline in detail, including **initial data loading**, **cleaning statistics**, **seasonal distributions**, and **split metrics**. The presence of built-in validation mechanisms ensures the correct functioning and reliability of the pipeline at every step.

5.2 Model implementation

This section is dedicated to the implementation, testing, and evaluation of the models.

We begin by validating each module independently to ensure correct functionality. Unit tests confirm the proper construction of the feature matrices—comprising five columns (age and four positional indicators)—as well as the consistency of the feature space between the training and test sets. Pytest unit tests are implemented to validate both the data pipeline and the baseline model. All five tests execute successfully in 5.5 seconds, confirming the correctness, reproducibility, and stability of the codebase.

Subsequently, we develop the `run_baseline.py` script, which orchestrates the entire baseline workflow. This script handles data loading, baseline feature construction, model training on the **2018–2019 to 2021–2022 seasons** (1,690 observations), and model evaluation on the **2022–2023 season** (382 observations). Model predictions and performance metrics are systematically saved to enable further analysis and comparison.

5.3 Feature importance analysis

The objective of this section is to interpret the model's predictions and to empirically validate the two research hypotheses:
(1) **Minutes played for a top club increase a player's market value**, and
(2) **Goals are more important for forwards than for players in other positions**.

To achieve this, we assess feature importance using the **XGBoost gain metric**, which measures the average reduction in prediction error attributable to each feature across all splits in which it appears. For interpretability, importance scores are normalized and expressed as percentages.

We then categorize and present the results. The **15 most important features** are displayed and automatically classified into thematic groups: **affiliation**, **position**, **goals**, **passes**, **minutes played**, **age**, **physical attributes**, and **nationality**. In addition, summary statistics report the cumulative importance share of the **top five** and **top ten** features, providing insight into the concentration of explanatory power within the model. These top 15 features are in Figure 1 of the appendix.

The next step focuses on the validation of the two hypotheses.

For **Hypothesis 1**, we examine whether features associated with the **top ten Premier League clubs** (such as Manchester City and Liverpool) and variables capturing **minutes played** rank among the most important predictors. The analysis computes their cumulative importance and identifies which top clubs appear within the top ten features, thereby assessing the role of club prestige and playing time in player valuation.

For **Hypothesis 2**, we compare the total importance attributed to **goal-related features** with that of **position-related features**, with particular attention to forwards. This comparison evaluates whether goal-scoring ability is more discriminative than positional classification alone and whether goal variables are represented among the overall top ten predictors.

Finally, the results are visualized using two complementary figures: a **horizontal bar chart** displaying the most important features, color-coded by category, and a **pie chart** illustrating the distribution of importance across feature categories. A concise textual report synthesizes the findings, confirms or rejects the hypotheses, and highlights key business insights—for example, whether **club affiliation outweighs individual goal-scoring performance** in determining market value.

5.4 Error and residual analysis

This section plays a critical role in the report, as it aims to identify where and why the XGBoost model generates prediction errors. The objective is to uncover systematic patterns in these errors and to highlight potential avenues for model improvement.

The analysis begins with the computation of prediction errors for each player in the test set corresponding

to the **2022–2023 season**. Three types of errors are calculated: **raw residuals** (actual value minus predicted value), **logarithmic residuals** (actual log value minus predicted log value), and **percentage errors**. These results are stored in the `error_data.csv` file. Overall, the model yields a **mean absolute error (MAE) of 10.8 million euros** and an **average percentage error of 58.9%**.

The scatter plot of **true versus predicted values** reveals a systematic pattern of underestimation for high-value players. While predictions remain relatively accurate for low- and medium-valued players, the model fails to fully capture extreme market values, leading to a compression of predictions in the upper tail of the distribution. This behavior suggests that the model adopts a conservative valuation strategy when faced with rare and highly priced player profiles.

We then assess the statistical properties of the error distribution. A **Shapiro–Wilk test** is conducted to test for normality, complemented by the computation of **skewness** and **kurtosis**. The results indicate that the residuals do not follow a normal distribution and reveal a systematic bias, with the model **underestimating player values by approximately 2.5 million euros on average**. Moreover, the **residuals versus predicted values** plot highlights clear **heteroscedasticity**, as the variance of errors increases with predicted market value. This finding indicates that valuation uncertainty grows for higher-valued players, reflecting the greater volatility and subjectivity of market prices in this segment.

The distribution of **percentage errors** is strongly right-skewed, with a long tail driven by extreme relative errors for low-valued players. This pattern reflects a well-known limitation of percentage-based metrics, which tend to magnify errors for players with small absolute market values.

Subsequently, we perform a **segment-level error analysis**, the results of which are summarized in the appendix. Additional tables listing the largest errors in both absolute and relative terms are provided in `worst_absolute_errors.csv` and `worst_percentage_errors.csv`. This segmented analysis highlights that **goalkeepers** are associated with the largest prediction errors and that the most error-prone age group centers around **29.8 years**. Detailed segment-level results are reported in Appendix Figures 2 and 3.

Finally, the cumulative distribution of absolute errors provides additional insight into the concentration of prediction mistakes. Approximately **50% of the total absolute error is generated by only 12.8% of players**, indicating that overall model performance is driven by a relatively small subset of difficult-to-predict profiles. In contrast, the majority of players are valued with comparatively moderate errors.

The full set of visual diagnostics, saved as `error_distribution.png`, `errors_by_segments.png`, and `final_visualizations.png`, includes six detailed plots—most notably **actual versus predicted values**, **residuals versus predictions**, and a **heatmap of prediction errors by position and age**. Together, these visualizations provide an intuitive understanding of the model's limitations and offer clear guidance for future improvements, such as segment-specific modeling or the inclusion of additional explanatory variables.

6 Code maintenance and updates

6.1 Project structure

We established the project structure on **GitHub** to enable version control and collaboration, and subsequently cloned the repository locally on **macOS**. The raw data files (CSV format) were stored within the predefined directory structure to ensure seamless integration with the codebase and to guarantee the reproducibility of the analyses.

An `environment.yml` file was created to specify **Python 3.11** along with the required core libraries (including *pandas*, *scikit-learn*, *XGBoost*, among others). The Conda environment was then initialized using the `conda env create` command, resulting in an isolated and fully reproducible working environment. All necessary Python source files were organized within the `src/` directory, and `main.py` was configured as the primary entry point of the project.

6.2 Resolving environmental and integration issues

In this section, we identify and resolve a module import issue (**ModuleNotFoundError** for *pandas*) by reactivating the dedicated Conda environment. In addition, the *scikit-learn-intelex* package—known

to be incompatible with **Apple Silicon** architecture—was removed from the `environment.yml` file to ensure the stability and proper functioning of the runtime environment.

6.3 Packaging the model for production

The objective of this final phase is to transform the optimized XGBoost model into a fully autonomous prediction system, ready for deployment in a production environment. The resulting package encapsulates the entire processing pipeline, from data ingestion to prediction generation, thereby ensuring reproducibility, robustness, and ease of use.

The deployment package is structured to clearly separate prediction logic, dependencies, example inputs, and trained artifacts. The main prediction script (deployment/predict.py) integrates a command-line interface (CLI) and contains the `PlayerValuationPredictor` class, which faithfully reproduces all feature transformations developed in previous phases, including ratio computations, logarithmic transformations, and one-hot encoding of positions and clubs. The `deployment/requirements.txt` file specifies the minimal set of Python dependencies required to run the model in a production environment, ensuring portability and environmental compatibility. An example dataset (deployment/example_players.csv) provides representative player profiles, allowing immediate testing of the system without prior data preparation. Finally, the optimized XGBoost model is serialized and stored in `deployment/trained_model.joblib`, together with its associated preprocessing metadata.

All feature engineering steps—such as goals-per-minute and assists-per-minute ratios, `log1p` transformations, and elite club encoding—are fully replicated within the prediction pipeline, eliminating any risk of data leakage between training and inference. The system supports both batch predictions from CSV files and individual predictions through Python dictionary inputs. Prediction outputs are expressed in euros and enriched with estimated confidence intervals, while detailed logging ensures full traceability of the prediction process. In addition, built-in input validation mechanisms verify the consistency and completeness of incoming data prior to prediction, enhancing the reliability of the system.

Overall, the final outcome is a fully encapsulated and standalone prediction environment that enables both technical and non-technical users to obtain market value estimates without requiring model retraining or detailed knowledge of the underlying implementation. The clear separation between development code and the production-ready package ensures long-term maintainability and facilitates incremental updates to the model.

Finally, the entire deployment workflow was version-controlled using Git, with structured and descriptive commit messages documenting each stage of the packaging process. This includes the integration of the prediction pipeline, dependency management, input validation mechanisms, and model serialization. Version control ensures full traceability of changes, facilitates maintenance, and guarantees the reproducibility of the production-ready system.

7 Results

7.1 Comparison of model performance

The results of the model comparison are presented in Table 1. They clearly indicate that **XGBoost outperforms all other machine learning models** considered in this study. The model achieves substantially lower **MAE** and **RMSE**, and the **R-squared** further confirms its superior explanatory power. In relative terms, XGBoost performs particularly well, with percentage errors that are nearly **half those obtained with linear regression models**. These results justify the selection of XGBoost as the primary model for further improvements and analysis. These results are in table 3 in the appendix.

We then compare the optimized XGBoost model with the baseline linear regression. This comparison reveals an error reduction of approximately 11.5% ($p < 0.001$), underscoring the substantial performance gains achieved by modeling non-linear relationships and richer interactions among features. In addition, XGBoost is benchmarked against two simple reference models—Naïve Mean and Naïve Median—which predict player values using the sample mean and median, respectively. The optimized XGBoost model significantly outperforms both naïve baselines, achieving a 36.1% lower error relative to the Naïve Mean model ($p < 0.001$) and a 34.9% lower error relative to the Naïve Median model ($p < 0.001$).

Although the Naïve Median model exhibits relatively competitive performance given its extreme simplicity, with an RMSE approximately 55% higher than that of XGBoost, it remains clearly inferior in

predictive accuracy. The consistent statistical significance of XGBoost's superiority over all baseline models (all p-values < 0.001) confirms the added value of the more sophisticated modeling approach for estimating football player market values. All detailed results from these comparisons are reported in Appendix table 4.

7.2 Analysis of the importance of variables

All results are reported in the analysis summary document. Based on the feature importance analysis derived from the XGBoost model, several important insights emerge regarding the determinants of football player valuation. A striking result concerns the absence of club-related effects: the variable *club_ Other* and all other club indicators (*club_ Top_ Club*, *club_ Middle_ Table_ Club*, *club_ Relegation_ Battle_ Club*) exhibit 0.0% importance in the model. This unexpected outcome suggests that the four-group club categorization fails to capture any meaningful influence of club affiliation on player market value. Contrary to initial expectations, distinctions between elite clubs, mid-table teams, and relegation-battle clubs do not contribute to the predictive power of the model.

As a consequence, the first hypothesis—stating that playing for prestigious clubs increases player market value—is entirely refuted. Club-related variables collectively explain none of the model's explanatory power, and no club category appears among the important features. This result challenges the common assumption that institutional prestige plays a central role in player valuation, instead suggesting that market prices are driven primarily by individual characteristics rather than by the club environment.

The second hypothesis—according to which goal-scoring is more important for forwards than their positional classification—is partially validated. Goal-related variables (*Goals* and *Goals_ per_ minute_ log*) account for a combined importance of 14.4%, whereas the forward position indicator (*is_ Forward*) represents only 6.9%. This indicates that measurable offensive output is a stronger determinant of value than positional labels alone. However, an unexpected pattern emerges for goalkeepers: the *is_ Goalkeeper* feature dominates all positional indicators with an importance of 14.0%, highlighting that goalkeepers are evaluated according to a distinct and highly specific valuation logic.

More broadly, individual characteristics clearly emerge as the primary drivers of player valuation. The most important feature is *Matchs_ played* (19.0%), followed by goalkeeper position (14.0%) and goals scored (10.0%). This highlights the central role of availability, durability, and observable performance metrics in market valuation, while institutional affiliation and even positional roles appear secondary. All the features analysis are in Appendix Figure 1.

From a modeling perspective, the failure of the four-group club categorization—despite its conceptual sophistication—suggests either that club effects are genuinely negligible in determining market values or that the chosen categorization does not reflect the economically relevant distinctions perceived by the market. It is also likely that club influence is indirectly captured through performance variables, as individual statistics naturally depend on the competitive environment provided by the club.

From an operational standpoint, these findings imply that recruitment and valuation strategies should prioritize player availability, measurable productivity (such as goals and assists), and the recognition of distinct evaluation mechanisms for goalkeepers. Club affiliation, whether elite or not, appears to have little direct impact on market value, which may reflect an increasingly rationalized and fluid football labor market in which individual performance dominates institutional context.

7.3 Error analysis

Although the detailed error analysis is presented in Section 6.5, this section provides a concise quantitative summary of the main findings. The model exhibits an average absolute error of €12.6 million (MAE) and a Root Mean Squared Error of €21.0 million, with a mean relative error of 58.9%. The residuals indicate a systematic underestimation bias of approximately €10.8 million, combined with a right-skewed error distribution characterized by positive skewness (1.969) and high kurtosis (4.405). This pattern reflects heavy tails driven by large underestimations for a subset of players.

Model performance varies substantially across player segments. Midfielders (€14.1 million MAE) and players aged 21–25 (€15.0 million MAE) are the most difficult to predict accurately, while younger players under 21 (€4.9 million MAE) are estimated with considerably higher precision. Goalkeepers exhibit the highest relative error (73.8%) despite a moderate absolute MAE (€8.2 million), highlighting particularly poor proportional accuracy for this position.

Error analysis by club affiliation reveals a pronounced discrepancy. Players from top clubs display more than double the prediction error (€18.1 million MAE) compared to players from other clubs (€6.9 million MAE). This substantial gap suggests that the model struggles to capture valuation mechanisms for high-profile players, whose market prices may incorporate premiums linked to prestige, media exposure, or strategic importance—factors not explicitly represented in the current feature set. All error analysis figures are in the appendix Figures 2 and 3.

The largest absolute prediction errors predominantly involve high-value defenders and midfielders playing for elite clubs, particularly Manchester City and Liverpool, with several individual errors exceeding €70 million. In contrast, the most extreme relative errors occur among low-value players, where percentage errors exceed 400% for players valued below €5 million. This asymmetry highlights the inherent difficulty of modeling market values at both extremes of the valuation distribution.

7.4 Temporal cross-validation

This section evaluates model performance using cross-validation while explicitly accounting for the temporal structure of the dataset. In this setting, standard random cross-validation is inappropriate, as it may introduce **data leakage** by allowing information from future seasons to influence the estimation of past observations. Such an approach would also be inconsistent with real-world forecasting, where future information is unavailable at prediction time.

To address these issues, we adopt a **time series split strategy** with three folds, strictly preserving the chronological order of the data:

- Fold 1: Train on **2018–2019** → Test on **2019–2020**
- Fold 2: Train on **2018–2020** → Test on **2020–2021**
- Fold 3: Train on **2018–2021** → Test on **2021–2022**

The results, summarized in **Figure 4** in the appendix, indicate strong overall stability across models. The figure reports the average **RMSE** and **R-squared** across folds, together with their associated variability. XGBoost consistently achieves the **lowest RMSE** and the **highest R-squared**, confirming its superior predictive accuracy relative to all other models. In contrast, linear and Ridge regression models display comparatively high RMSE values and limited explanatory power, despite their high stability.

Model stability across folds further highlights important differences between approaches. While linear models exhibit the lowest variability, their predictive performance remains weak. Random Forest, although achieving strong average performance, is the **least stable model**, displaying the highest variation across temporal folds. XGBoost offers a favorable compromise, combining strong predictive accuracy with moderate variability.

Finally, the fold-level performance of XGBoost shows that, despite some variation across seasons, the model remains consistently within a narrow performance range, with a **mean R-squared of 0.476**. This confirms the **temporal robustness** of the model and suggests that its superior performance is not driven by a single favorable season.

Overall, these findings indicate that none of the models suffer from overfitting and confirm that XGBoost provides the best trade-off between **predictive accuracy and temporal stability** in a realistic forecasting setting.

7.5 Model performance in production

This section aims to assess the predictive capacity of the selected and optimized model prior to deployment. As discussed earlier, we chose to optimize the XGBoost model; this section provides a concise summary of its performance.

The optimized XGBoost model achieves the best overall results, with an **RMSE of €17.6 million** and an **R-squared of 0.459**, explaining nearly **46% of the variance** in player market values. It outperforms the other models by approximately **6 to 8% in predictive accuracy** and reduces the baseline model error by **36%**. In addition, the model exhibits a **MAPE of 58.9%** and demonstrates strong temporal robustness, with a **cross-validated R-squared of 0.452 ± 0.032**. These results indicate that XGBoost offers the most balanced trade-off between accuracy, stability, and model complexity, effectively capturing the non-linear dynamics of the football transfer market.

Based on these findings, we proceed to export the optimized XGBoost model, as described in Section

7.3.

7.6 Project limitations

The main limitation of the model emerges from the results and residual analyses. The residual distribution is clearly non-normal, as indicated by the Shapiro–Wilk test ($p < 0.001$) and a negative skewness (-0.56), revealing systematic bias and fat-tailed errors. Such behavior is expected for a tree-based model like XGBoost and does not undermine its overall validity. However, it confirms the segmentation results: the model tends to underestimate the market value of players from top clubs and goalkeepers. This suggests that segment-specific determinants—such as institutional prestige, media exposure, or specialized performance metrics—are not fully captured by the current feature set. Future improvements could rely on segment-specific learning strategies or the inclusion of targeted variables designed for these profiles.

A second major limitation lies in the restricted number of explanatory variables, which limits the model's ability to fully capture player market value. This partly explains the relatively moderate R-squared and the non-negligible prediction errors. Football valuation depends on numerous factors that are inherently difficult to quantify, such as the contextual importance of goals, leadership, consistency, or subjective assessments of performance quality. Moreover, the dataset originates from external sources and reflects constraints in variable availability, combined with deliberate but limited feature selection choices, which further restrict explanatory power.

Finally, the model performs particularly poorly for goalkeepers, whose valuation follows distinct dynamics compared to outfield players. This highlights a structural limitation of using a unified model across all positions. One potential improvement would be to focus on a single player category—such as strikers—or to develop position-specific models, potentially leveraging data from multiple leagues to increase sample size and improve generalization.

8 Conclusion

This study shows that non-linear machine learning models, particularly XGBoost, significantly outperform linear approaches in predicting football player market values. Individual performance and availability emerge as the main drivers of valuation, while club-level indicators provide little additional explanatory power. Goal-scoring is more informative than positional labels alone, although goalkeepers follow a distinct valuation pattern.

Despite strong predictive performance, the model systematically underestimates high-value players and performs less accurately for certain segments, highlighting inherent limits in football valuation. Overall, the results confirm the relevance of machine learning for player valuation while pointing to future improvements through richer data and position-specific modeling.

9 Bibliographie

Idris, M. A. J., & Ng, S. L. (2025). Developing a predictive model for football players' market value using machine learning. *Informatics and Web Engineering*, 4(3). https://mmupress.com/index.php/jiwe/article/view/1760/1081

Geng, B. (2024). Predicting football player transfer values using bagging and hybrid machine learning approaches. *Informatica*, 49(22). https://www.informatica.si/index.php/informatica/article/view/7715

Tamim, A., Jahan, M. W., Chowdhury, M. R. S., Hossain, A., Rahman, M. M., & Imon, A. H. M. R. (sans date). Machine learning-driven market value prediction for European football players. *Journal of Computational Mathematics and Data Science*. https://www.sciencedirect.com/science/article/pii/S2772415825000100

European Club Ranking. (2025, March 23). *5-Year Ranking (2019-23)*. Retrieved January 2026, from https://www.clubranking.eu/5-year-ranking-2023/

Bayar, E. (2019). FIFA 19 Dashboard with R Shiny [code and data]. Kaggle. Available at: https://www.kaggle.com/code/ekrembayar/fifa-19-dashboard-with-r-shiny/input. Accessed January 2026.

Leone, S. (2020). FIFA 20 Complete Player Dataset [dataset]. Kaggle. Available at:

https://www.kaggle.com/datasets/stefanoleone992/fifa-20-complete-player-dataset. Accessed January 2026.

Bayar, E. (2021). FIFA 21 Complete Player Dataset [dataset]. Kaggle. Available at: https://www.kaggle.com/datasets/ekrembayar/fifa-21-complete-player-dataset. Accessed January 2026.

Leone, S. (2022). FIFA 22 Complete Player Dataset [dataset]. Kaggle. Available at: https://www.kaggle.com/datasets/stefanoleone992/fifa-22-complete-player-dataset. Accessed January 2026.

Cashncarry. (2023). FIFA 23 Complete Player Dataset [dataset]. Kaggle. Available at: https://www.kaggle.com/datasets/cashncarry/fifa-23-complete-player-dataset. Accessed January 2026.

FBref. (2024). Premier League Statistics [dataset]. Sports Reference LLC. Available at: https://fbref.com/en/comps/9/Premier-League-Stats. Accessed January 2026.

10 Note on the Use of Assistance Tools

This project utilized artificial intelligence tools as technical assistance for two aspects:

1. **Coding Assistance**: The pre-code instructions as well as implementation choices were defined by the author, with artificial intelligence being used as a targeted assistance tool for code setup and adjustment, particularly for data pipeline structuring and certain technical aspects of optimization.

2. **Writing Revision**: Certain technical formulations in the report were optimized with the help of AI-assisted writing tools to improve clarity and consistency.

The overall methodological design, results analysis, data interpretation, and conclusions remain the product of original and supervised intellectual work. The AI tools served solely as aids for technical and writing implementation.

11 Appendix

11.1 Tables and Figures

Table 1: Baseline Model Performance (Age + Position only)

Metric	Value
RMSE	27 202 612 €
MAE	16 352 092 €
R^2	-0.324

Table 2: Optimized hyperparameters for the XGBoost model

Hyperparameter	`Values tested`
n_estimators	`[50, 100, 150, 200]`
max_depth	`[3, 4, 5, 6, 7, 9]`
learning_rate	`[0.001, 0.005, 0.01, 0.05, 0.1, 0.2]`
subsample	`[0.6, 0.7, 0.8, 0.9, 1.0]`
colsample_bytree	`[0.6, 0.7, 0.8, 0.9, 1.0]`
gamma	`[0, 0.1, 0.2, 0.3, 0.4]`
reg_alpha	`[0, 0.001, 0.01, 0.1, 1]`
reg_lambda	`[0.1, 0.5, 1, 5, 10]`

Table 3: Model performance comparison on 2022-2023 test set (sorted by RMSE)

Rank	Model	RMSE (M€)	MAE (M€)	R^2	MAPE%
1	**XGBoost**	17.4	10.8	0.459	58.9 %
2	Linear Regression	18.7	12.1	0.371	107.2 %
3	Ridge Regression	18.8	12.1	0.371	107.2 %
4	Random Forest	19.0	11.5	0.354	61.6 %

Table 4: Comprehensive model performance comparison including naive baselines

Model	RMSE (M€)	MAE (M€)	R^2	MAPE%
XGBoost Optimized	17.6	10.8	0.459	58.9 %
Random Forest	19.0	11.5	0.354	61.6 %
Linear Regression	18.7	12.1	0.371	107.2 %
Naïve Median	19.2	11.5	0.343	92.1 %
Naïve Mean	27.7	18.1	−0.374	104.3 %

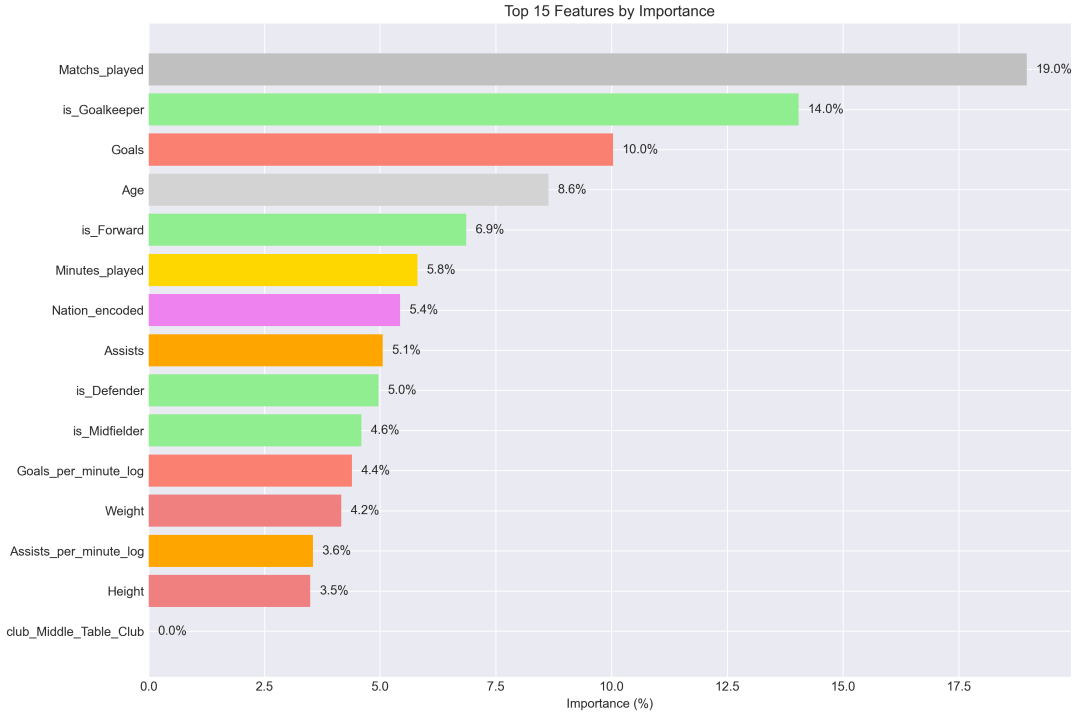

Figure 1: Feature Importance by Category (Top 15 Features)

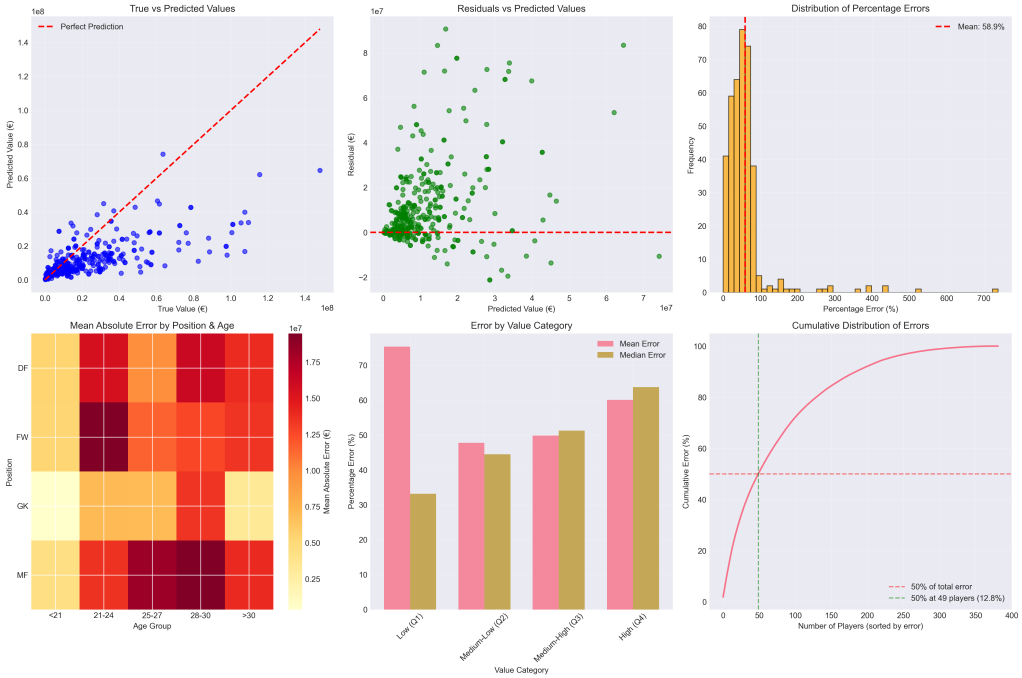

Figure 2: Final visualizations of the model results

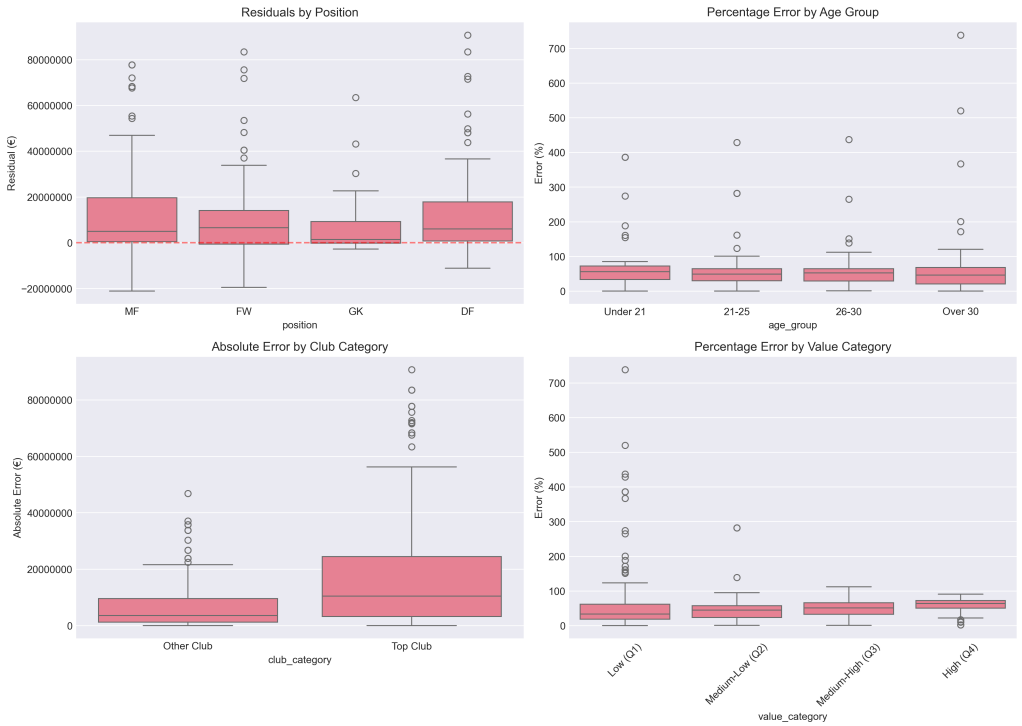

Figure 3: Analysis of errors by different segments

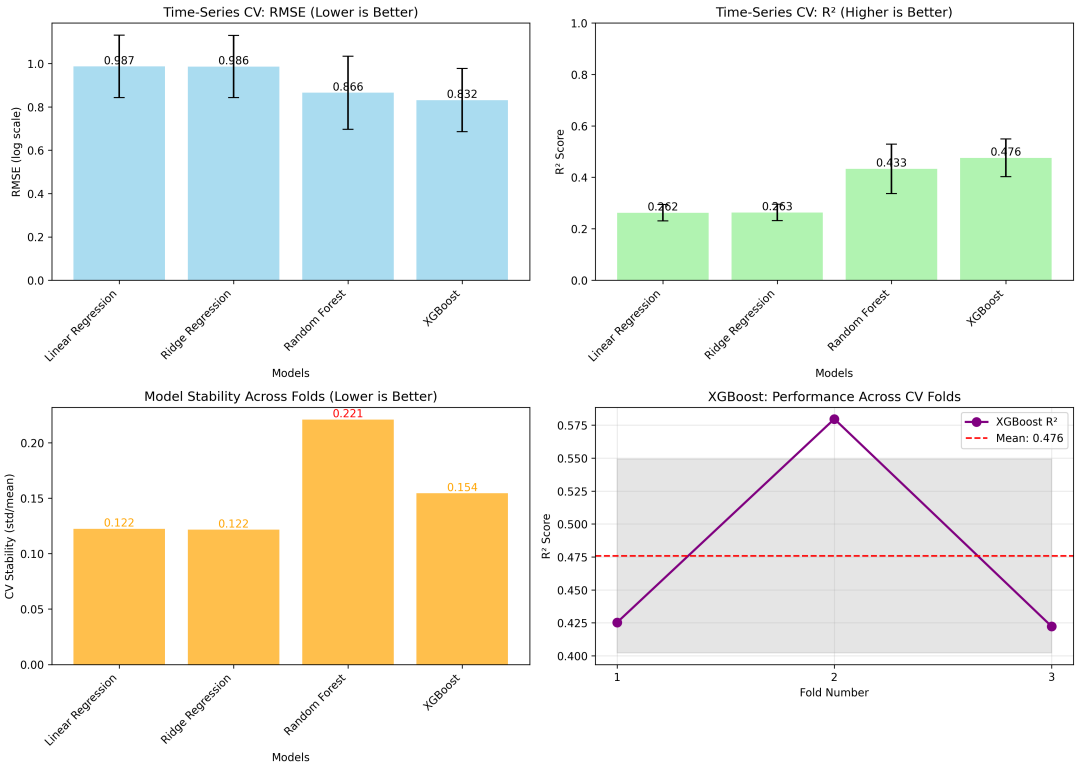

Figure 4: Time series analysis results

11.2 Club categorisation based on average league position (2018–2023)

Using the average final league position over the 2018–2019 to 2022–2023 Premier League seasons, clubs are classified into three distinct categories reflecting their competitive status.

Top clubs (excluded from the analysis)

- Manchester City
- Liverpool
- Chelsea
- Manchester United
- Arsenal
- Tottenham
- Leicester
- West Ham
- Wolves
- Newcastle

Middle-table clubs These clubs generally finish in mid-table positions and are not consistently involved in title races or relegation battles.

- Everton
- Aston Villa
- Brighton
- Crystal Palace
- Southampton

- Bournemouth
- Leeds
- Brentford

Relegation-battle clubs Clubs with an average league position greater than or equal to 15, frequently involved in relegation battles.

- Norwich
- Watford
- Burnley
- Sheffield United
- Fulham
- Cardiff
- Huddersfield