Virtual Internship 6.0

TransferIQ: Dynamic Player Transfer Value Prediction using AI & Multi-source Data

Submitted by

Dhanshri R Supratkar

Submitted To

Mr Raj Yadav

Abstract

The football transfer market involves complex and high-value decisions that are often influenced by subjective judgment. This project, TransferIQ, presents an AI-based approach to predict football player transfer market values using multi-source data. The system integrates player profiles, match performance statistics, injury history, transfer records, team information, and social media sentiment. Advanced data preprocessing and feature engineering techniques are applied to derive meaningful indicators affecting player valuation. Multiple machine learning models are evaluated, with LightGBM selected as the final model based on its superior predictive performance. The trained model is deployed through a Streamlit web application, enabling real-time market value prediction. TransferIQ demonstrates the effectiveness of data-driven decision-making in sports analytics and aligns with the learning objectives of the Infosys Springboard AI program.

CHAPTER 1: INTRODUCTION

1.1 Background

The football transfer market is a multi-billion-dollar global industry where clubs invest heavily in acquiring and selling players to strengthen team performance and achieve long-term strategic goals. Player transfer values are influenced by a wide range of factors, including on-field performance, age, injury history, contract duration, team success, and public perception. With the increasing commercialization of football, accurate player valuation has become critical for clubs, agents, and analysts.

However, traditional player valuation approaches often rely on subjective assessments, limited statistical indicators, and market speculation. These methods struggle to capture the complex interactions between performance metrics, player availability, and market sentiment. As a result, there is a growing need for data-driven and intelligent systems that can objectively evaluate player value by integrating multiple data sources and analytical techniques. Advances in artificial intelligence and machine learning provide an opportunity to transform player valuation into a more transparent, scalable, and reliable process.

1.2 Problem Statement

Player transfer market valuations are highly subjective and volatile, often influenced by short-term performance trends, media narratives, and public sentiment rather than comprehensive data analysis. This subjectivity can lead to inconsistent valuations and irrational market behavior.

Such volatility poses significant financial risks for football clubs, including overpaying for players with inflated market value or undervaluing emerging talent. The absence of an integrated, data-driven valuation framework limits informed decision-making and increases

uncertainty in transfer negotiations. Therefore, there is a need for an objective and predictive system that can assess player transfer value using historical data and multiple influencing factors.

1.3 Objectives

The primary objectives of this project are as follows:

- To design and develop an AI-based system for predicting football player transfer market values
- To integrate and analyze multi-source data, including player performance, injury history, transfer records, team information, and social media sentiment
- To apply machine learning techniques for accurate and interpretable market value prediction
- To deploy a real-time prediction system through an interactive web application for practical usability

1.4 Scope of the Project

This project focuses on building a data-driven framework for player transfer value prediction using historical and contextual data. The scope includes:

- Analysis of professional football player datasets
- Integration of structured and unstructured data sources
- Feature engineering and machine learning-based valuation
- Deployment of a real-time prediction interface

CHAPTER 2: SYSTEM OVERVIEW

2.1 Proposed System

The proposed system, TransferIQ, is an AI-driven player valuation framework designed to predict football player transfer market values using multi-source data and machine learning techniques. The system integrates diverse datasets related to player demographics, match performance, injury history, transfer records, team information, and social media sentiment to generate an objective and data-backed estimation of player market value.

The framework follows an end-to-end data science workflow, beginning with data acquisition and preprocessing, followed by feature engineering, model training, evaluation, and deployment. Advanced ensemble learning models are employed to capture complex non-linear

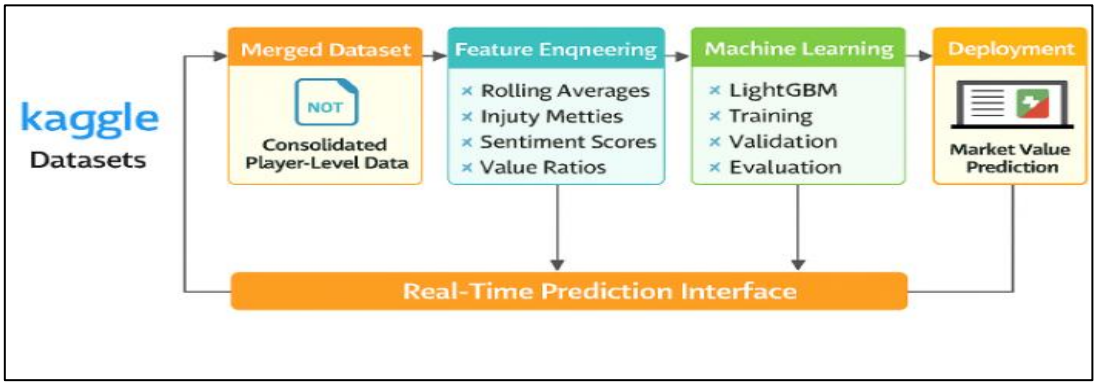
relationships between player attributes and market value. The system emphasizes explainability through feature importance analysis, enabling stakeholders to understand the key factors influencing player valuation. By deploying the model through an interactive web application, the system supports real-time prediction and practical decision-making for football clubs, scouts, and analysts.

2.2 System Architecture

The system architecture of TransferIQ is designed to ensure scalability, modularity, and clarity in data flow. It consists of multiple interconnected layers that collectively transform raw data into actionable insights.

At the data layer, raw datasets are collected from multiple sources, including player profiles, match performance statistics, injury records, transfer history, market value data, team information, and social media platforms. The preprocessing layer handles data cleaning, missing value imputation, date conversions, and outlier treatment to ensure data consistency.

The feature engineering layer derives meaningful indicators such as rolling performance averages, injury risk metrics, sentiment scores, and value efficiency ratios. These engineered features are consolidated into a final analytical dataset, which serves as input to the machine learning layer. The modeling layer trains and evaluates multiple regression models, with LightGBM selected as the final model based on performance metrics. Finally, the deployment layer integrates the trained model into a Streamlit-based web application for real-time prediction.



2.3 Technology Stack

The TransferIQ system is built using a robust Python-based tech stack, supporting all stages from data processing to deployment:

Programming Language

- **Python**: Primary language for data analysis, model training, and deployment.

Data Handling & Manipulation

- **Pandas**: For data loading, cleaning, transformation, aggregation, and feature engineering.

- **NumPy**: For numerical operations, array manipulation, and integration with Pandas.

Machine Learning Libraries

- **Scikit-learn**: Used for preprocessing (StandardScaler, OneHotEncoder), model selection (train_test_split, cross_val_score), and algorithms like Linear Regression, Lasso, Decision Tree, and Random Forest.
- **LightGBM (LGBMRegressor)**: Primary regression model; used for training and feature importance analysis.
- **XGBoost (XGBRegressor)**: Comparative gradient boosting model.

Data Visualization

- **Matplotlib & Seaborn**: For visualizing data distributions, feature importances, and model performance (e.g., actual vs predicted plots, boxplots).

Model Persistence

- **Pickle**: For serializing and deserializing the trained LightGBM model (lightgbm_model.pkl) for deployment.

Deployment Framework

- **Streamlit**: Framework for building interactive web applications to predict player market value.
- **pyngrok**: Exposes the local Streamlit application to the internet for remote access.

Development Environment

- **Google Colaboratory (Colab)**: Cloud-based Jupyter notebook environment used for coding, data analysis, and model development.

Version Control / File Management

- **Google Drive**: Stores datasets, project files, and saved models. Acts as a central repository for project assets.

The above technology stack provides a comprehensive ecosystem for building, training, evaluating, and deploying a production-ready AI-based football player valuation system.

CHAPTER 3: DATA COLLECTION & EXPLORATION

3.1 Data Sources

The project utilizes multiple datasets covering player, team, and transfer information. Each dataset is described below:

Dataset Name	Description / Potential Use
player_injuries	Player injury history including type, days injured, and recovery period. Useful for modeling player availability and risk.
player_latest_market_value	Most recent market value of players. Used as the target variable in predicting transfer value.
player_market_value	Historical market values of players. Useful for trend analysis and feature engineering.
player_national_performances	Player performances in national team competitions. Adds performance context beyond club-level stats.
player_performances	Detailed club-level statistics: goals, assists, minutes played, passes, etc.
player_profiles	Player demographics: age, nationality, position, and contract details. Essential for basic features.
player_teammates_played_with	Records of teammates players have played with. Can help model synergy or team influence on individual performance.
team_children	Youth or secondary teams associated with main clubs. Useful for understanding player career trajectories.
team_competitions_seasons	Team-level statistics across competitions and seasons. Adds team context for player performance.
team_details	Metadata about teams: name, league, country, and ranking. Useful for league-level modeling.
transfer_history	Player transfer fees and dates. Key for analyzing market trends and transfer patterns.

3.2 Exploratory Data Analysis (EDA)

EDA was conducted to understand patterns, trends, and anomalies before model building.

3.2.1 Market Value Distribution Analysis

Reason:
Player market value is the target variable of the project. Understanding its distribution is essential to identify skewness, outliers, and scaling requirements.

Method Applied:

- Histogram and kernel density estimation (KDE)

- Outlier detection using distribution tails

Datasets Used:

- player_market_value
- player_latest_market_value

Insights:

- Market value shows a right-skewed distribution, with most players valued lower and a few elite players valued extremely high.
- Presence of outliers indicates the need for log transformation or normalization before model training.

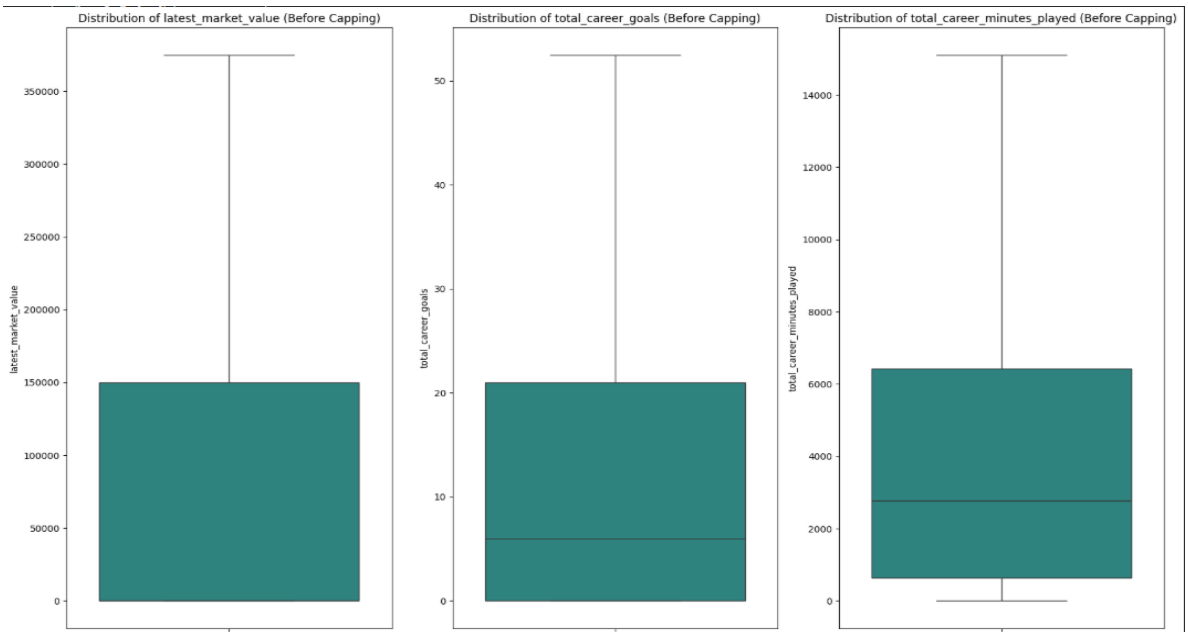


fig before Capping

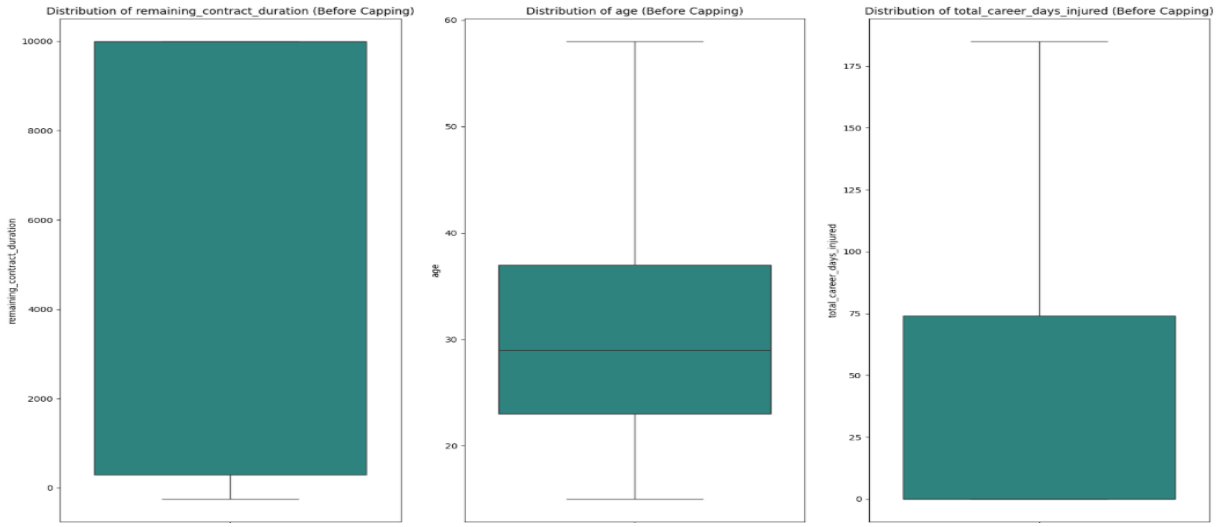


fig Before Capping

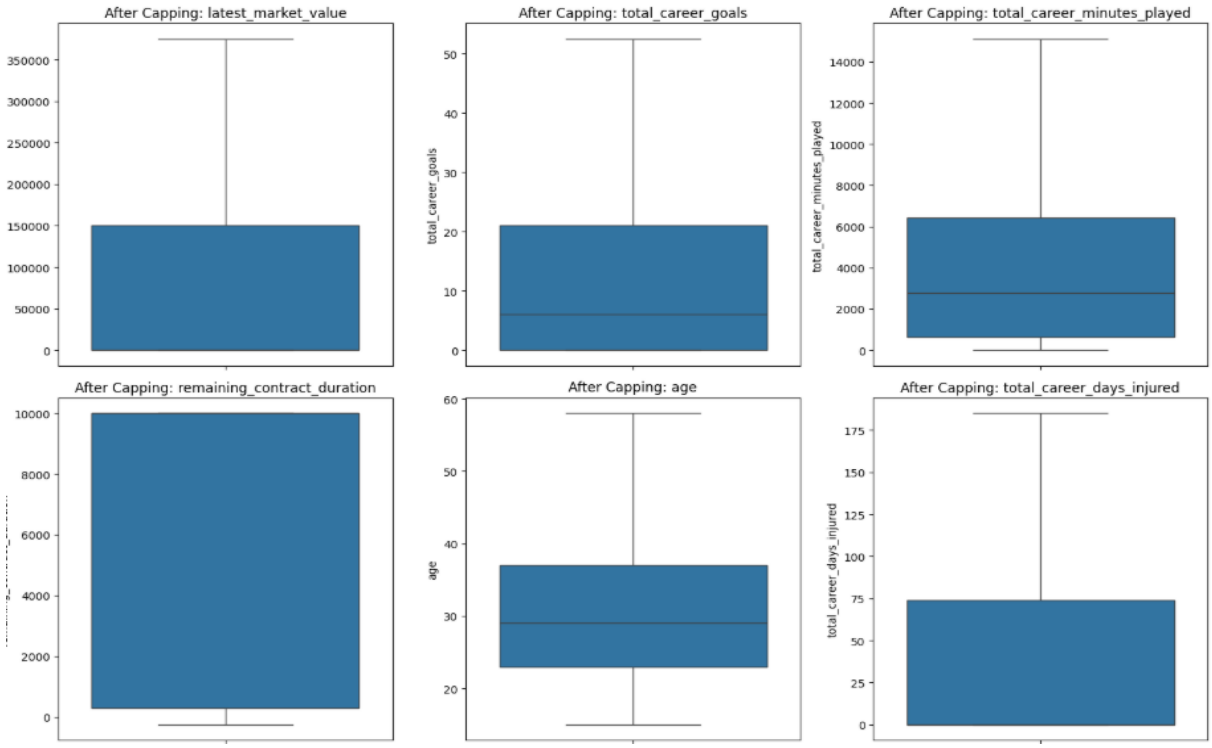


fig After Capping

3.2.2 Age vs Market Value Relationship Analysis

Reason:
Age is a fundamental factor influencing player performance, career stage, and transfer value.

Method Applied:

- Scatter plot visualization
- Correlation trend analysis

Datasets Used:

- player_profiles
- player_latest_market_value

Insights:

- Market value increases with age until approximately 24–27 years, after which it declines.
- The relationship is non-linear, indicating that age should not be treated as a linear predictor.

3.2.3 Performance Metrics Analysis

Reason:
On-field performance directly affects a player's valuation and transfer demand.

Method Applied:

- Descriptive statistics
- Normalization using per-90-minute metrics

Datasets Used:

- player_performances

Insights:

- Players with higher goals per 90 minutes and assists per 90 minutes tend to have higher market values.
- Raw totals are misleading; rate-based metrics provide better performance representation.

3.2.4 Injury Impact Analysis

Reason:
Injury history affects player availability, longevity, and risk assessment by clubs.

Method Applied:

- Frequency analysis per player

Datasets Used:

- player_injuries

Insights:

- Majority of injuries are short-term, but long-term injuries act as strong negative indicators.
- Players with frequent injuries experience lower and unstable market values.

3.2.5 Transfer History Analysis

Reason:
Previous transfer fees reflect historical market confidence in a player.

Method Applied:

- Trend analysis of transfer fees
- Comparative analysis with current market value

Datasets Used:

- transfer_history

Insights:

- Players with high past transfer fees generally maintain higher market value.
- Frequent transfers introduce volatility in player valuation.

3.2.6 Team Influence Analysis

Reason:
A player's value is influenced by the competitive level and success of their team.

Method Applied:

- Group-wise aggregation by team and league
- Comparative league-level analysis

Datasets Used:

- team_details
- team_competitions_seasons

Insights:

- Players from top-tier leagues and successful teams have consistently higher market values.
- Team context adds indirect value amplification.

3.2.7 National Team Performance Analysis

Reason:
International exposure increases player visibility and global market demand.

Method Applied:

- Participation frequency analysis
- Performance-based comparison

Datasets Used:

- player_national_performances

Insights:

- Regular national team players show higher market valuation.
- International matches act as career value multipliers.

3.2.8 Social Media Sentiment Analysis

Reason:
Public perception and media sentiment influence short-term market behavior and transfer interest.

Method Applied:

- Sentiment polarity analysis
- Temporal sentiment trend observation

Datasets Used:

- Twitter sentiment data

Insights:

- Positive sentiment aligns with performance peaks and transfer rumors.
- Negative sentiment correlates with injuries or poor form.

CHAPTER 4: DATA PREPROCESSING & FEATURE ENGINEERING

4.1 Introduction

This chapter details the systematic preprocessing and feature engineering steps applied to transform raw, heterogeneous football datasets into a clean, consistent, and model-ready analytical dataset. The objective is to enhance data quality, reduce noise, and construct meaningful predictors that improve the accuracy and interpretability of player market value prediction models.

4.2 Data Preprocessing Pipeline

4.2.1 Data Integration Strategy

The core analytical dataset, merged_final_df, was constructed using a series of left-join merge operations to consolidate diverse football-related data sources into a unified, player-centric view. The player_id attribute served as the primary integration key, ensuring that all information—ranging from demographics and on-field performance to injuries, transfers, and public sentiment—was accurately linked to the correct individual player.

Left joins were intentionally selected to preserve the completeness of the base player population, ensuring that all players present in the primary *player_profiles* dataset were retained even when auxiliary information was partially unavailable.

The following datasets were integrated:

- **Player Profiles (player_profiles)**: Served as the foundational table, providing core demographic and positional information.

- **Market Value Data (player_latest_market_value, player_market_value)**: Merged to capture both historical trends and the most recent market valuation.
- **Player Performances (player_performances, player_national_performances)**: Aggregated club-level and national-level statistics including goals, assists, and minutes played.
- **Injury History (player_injuries)**: Summarized injury frequency, total days missed, and recovery durations.
- **Transfer History (transfer_history)**: Contributed data on transfer events, associated fees, and valuation at the time of transfer.
- **Team Context (team_competitions_seasons, team_children, team_details)**: Aggregated club competitiveness indicators such as competitions played, league divisions, and youth-team affiliations.
- **Teammate Interactions (player_teammates_played_with)**: Provided synergy-related metrics including joint goal participation and average points per game with teammates.
- **Sentiment Analysis (tweets_premier_league_footballers)**: Aggregated polarity scores and emotion distributions derived from social media mentions.

Post-merge validation checks ensured referential integrity, removed duplicate records, and verified aggregation correctness. The final integrated dataset consists of **92,671 player-level observations and 56 engineered .**

4.2.2 Feature Conversion to Numerical Data

Machine learning models require numerical representations of input data. Accordingly, all relevant features were converted into model-compatible numerical formats using the following strategies:

- **Date/Time Conversion**: Date attributes such as date_of_birth, joined, and transfer_date were converted to datetime objects. From these, continuous numerical features including age, days_since_joined, and days_since_last_transfer were derived to represent temporal and career-stage information.
- **Boolean to Integer Conversion**: The is_eu feature was transformed from boolean format into binary integers (1 for True, 0 for False) to ensure compatibility with numerical models.
- **Ordinal Categorical Mapping**: The main_position variable was mapped to an ordinal scale (main_position_encoded) reflecting functional player roles (Attack, Defender, Midfield, Goalkeeper).
- **One-Hot Encoding**: Nominal categorical variables such as place_of_birth, country_of_birth, and citizenship were initially one-hot encoded to prevent artificial ordinal relationships. High-cardinality or low-importance encoded features were later excluded to control dimensionality and model complexity.

4.2.3 Handling Missing Values

Missing data arose primarily due to incomplete injury logs, transfer records, or sentiment availability. The following strategies were applied:

- **Numerical features**: Median imputation to preserve distribution robustness.
- **Categorical features**: Imputation with Unknown where semantically appropriate.
- **Derived contract-related fields**: Placeholder values used where contracts were unavailable.
  This approach minimized data loss while avoiding biased estimations.

4.2.4 Outlier Detection and Treatment

Extreme values in performance, injury duration, and market value-related variables were addressed using the **Interquartile Range (IQR) capping method**. This controlled the influence of anomalous observations while retaining legitimate high-performing elite players.

4.2.5 Feature Reduction and Cleaning

Redundant and non-informative columns were removed, including:

- Raw identifiers already used for merging
- Original categorical columns replaced after encoding
- Intermediate aggregation fields no longer required post-feature construction

4.3 Feature Engineering

Extensive feature engineering was performed to enrich the dataset with meaningful, predictive variables. Each engineered feature or feature group was selected with a clear analytical and business-driven rationale, ensuring relevance, interpretability, and contribution to market value prediction.

4.3.1 Date-Based and Career Progression Features

Reason: Player valuation is strongly influenced by career stage, contract security, and recency of transfers. Temporal features help model depreciation, growth potential, and contractual leverage.

- remaining_contract_duration – Longer contracts increase bargaining power and market value.
- age – Captures career phase; younger high-performing players command premium valuations.

- days_since_joined – Indicates team stability and integration level.
- days_since_last_transfer – Reflects market recency and transfer momentum.

4.3.2 Performance Metrics

Reason: On-field productivity and consistency are primary drivers of market value.

Career Aggregates: Measure long-term ability and experience.

- total_career_matches, total_career_goals, total_career_assists, total_career_minutes_played

Seasonal Averages: Capture consistency within recent competitive cycles.

- avg_season_goals, avg_season_assists, avg_season_minutes_played

Rolling Averages (Last 3 Seasons): Emphasize recent form, which clubs value more than historical peaks.

- avg_rolling_avg_3_seasons_goals, avg_rolling_avg_3_seasons_assists, avg_rolling_avg_3_seasons_minutes_played

4.3.3 National Team Exposure Features

Reason: International appearances increase visibility, brand value, and perceived quality.

- national_total_matches, national_total_goals
- national_avg_rolling_avg_3_entries_matches, national_avg_rolling_avg_3_entries_goals

4.3.4 Injury and Fitness Indicators

Reason: Injury history directly impacts player availability and financial risk.

- total_career_days_injured – Measures long-term availability loss.
- total_career_injury_frequency – Indicates recurring fitness issues.
- overall_avg_recovery_time – Reflects injury severity.
- total_distinct_injury_types – Captures injury diversity and fragility.
- injury_rate_per_career_year – Normalizes injury risk across career length.
- injury_impact_score – Composite metric summarizing injury risk.

4.3.5 Transfer Activity Features

Reason: Transfer history provides direct market signals of demand and perceived worth.

- total_transfers – Indicates market mobility.
- total_transfer_fees, most_recent_transfer_fee, total_value_at_transfer – Represent historical financial valuation.
- transfer_activity_score – Aggregates transfer intensity and value.

4.3.6 Team Context Features

Reason: Playing environment influences exposure, competition level, and valuation.

- num_unique_competitions – Measures competitive diversity.
- num_unique_seasons – Indicates experience longevity.
- num_unique_club_divisions – Reflects league hierarchy exposure.
- num_child_teams – Captures club development ecosystem.

4.3.7 Teammate Dynamics Features

Reason: Football performance is collaborative; synergy enhances perceived value.

- avg_ppg_with_teammates – Measures effectiveness within team setups.
- total_joint_goal_participation – Captures collaborative attacking contribution.
- total_minutes_played_with_teammates – Reflects tactical trust and cohesion.
- num_unique_teammates – Indicates adaptability across team structures.

4.3.8 Market Value Normalization Ratios

Reason: Raw market value is scale-sensitive; ratios improve comparability.

- value_per_goal – Measures cost-efficiency of scoring output.
- value_per_minute_played – Normalizes value by playing time.
- market_value_to_age_ratio – Balances valuation against career stage.

4.3.9 Sentiment-Based Features

Reason: Media perception and fan sentiment influence demand and valuation.

- Polarity metrics: vader_polarity, tb_polarity
- Emotion distributions: positive, neutral, and negative sentiment scores

4.3.10 Categorical Encoding

Reason: Machine learning models require numerical input without artificial ordering.

- is_eu encoded as binary to reflect regulatory advantages.
- main_position mapped to ordinal values reflecting functional roles.
- Other categorical variables were one-hot encoded where analytically relevant, with dimensionality control applied to prevent sparsity.

4.4 Important Feature Selection and Rationale

Feature relevance was assessed using model-driven techniques, particularly **LightGBM intrinsic feature importance**, to identify the most influential predictors of player market value. The selected features capture performance quality, future potential, health risk, and market dynamics.

Key influential features include:

- **Target Variable**: latest_market_value, representing the player's current valuation.
- **Performance Indicators**: Career and recent metrics such as total_career_goals, total_career_minutes_played, avg_season_goals, and rolling averages, which directly reflect productivity and consistency.
- **Age (age)**: A critical determinant of market potential, capturing career longevity and resale value.
- **Contractual Status (remaining_contract_duration)**: Longer contracts typically increase bargaining power and valuation.
- **Transfer History**: Features such as total_transfer_fees, most_recent_transfer_fee, and transfer_activity_score provide direct signals of historical market demand.
- **Injury Profile**: Metrics like overall_avg_recovery_time and injury_impact_score quantify availability risk, negatively influencing valuation.
- **Value Normalization Ratios**: Features such as value_per_goal, value_per_minute_played, and market_value_to_age_ratio offer efficiency-based perspectives, improving predictive strength.
- **Team Contribution Metrics**: total_joint_goal_participation and avg_ppg_with_teammates capture collaborative effectiveness and team impact.

4.5 Final Dataset Readiness

Following integration, preprocessing, and feature engineering, the dataset demonstrates reduced noise, enhanced interpretability, and strong alignment with machine learning requirements. The **92,671 × 56** consolidated dataset is fully prepared for robust model training

and evaluation, forming the foundation for the predictive analyses presented in subsequent chapters.

CHAPTER 5: MODEL DEVELOPMENT DETAILS

This chapter explains the machine learning models developed to predict football player market value. Multiple regression techniques were applied to compare performance and understand how different algorithms capture relationships within the data.

1. Linear Regression (Base Model)

Objective:
To establish a simple baseline model that explains player market value using a linear relationship between input features and the target variable.

Configuration:

- LinearRegression from sklearn.linear_model
- No advanced hyperparameter tuning

Preprocessing:

- Numerical features were standardized using StandardScaler to ensure equal contribution of all features.

Performance (R-squared): 0.78

Details:
Linear Regression served as the benchmark model. The moderate R-squared score indicates that while the dataset contains linear relationships, player market value is influenced by more complex, non-linear patterns that cannot be fully captured by a simple linear model.

2. Polynomial Regression

Objective:
To model non-linear relationships between player attributes and market value by introducing polynomial feature transformations.

Configuration:

- PolynomialFeatures from sklearn.preprocessing
- Polynomial degrees tested: 1, 2, and 3
- Final model trained using LinearRegression on transformed features

Preprocessing:

- Polynomial feature expansion
- Feature scaling using StandardScaler

Performance (R-squared for Degree 3): 0.8201

Details:
Polynomial Regression improves flexibility by fitting curved relationships. However, higher-degree models showed signs of overfitting, highlighting the need for careful degree selection and validation.

3. Lasso Regression

Objective:
To reduce model complexity and automatically select the most influential features by shrinking less important coefficients to zero.

Configuration:

- LassoCV with cross-validation for optimal alpha selection
- max_iter = 10000, n_jobs = -1

Preprocessing:

- Feature scaling using StandardScaler (essential for Lasso)

Performance (R-squared): 0.9674

Details:
Lasso Regression achieved high predictive accuracy while significantly reducing the number of active features. This confirms that only a subset of engineered features strongly drives market value prediction.

4. Forward Feature Selection (with Linear Regression)

Objective:
To build a compact and interpretable model by gradually adding only those features that significantly improve prediction performance.

Configuration:

- Custom forward selection process using LinearRegression
- Selection stopped when R-squared improvement < 0.001

Preprocessing:

- Feature scaling using StandardScaler

Performance (R-squared): 0.9767

Details:
This approach resulted in a minimal yet effective model using only a few highly impactful features such as market_value_to_age_ratio and transfer_activity_score, proving that simpler models can still perform competitively.

5. Random Forest Regressor (Optimized)

Objective:
To predict continuous player market value using multiple decision trees aggregated to improve robustness and accuracy.

Configuration:

- RandomForestRegressor
- n_estimators = 200
- max_depth = 15
- min_samples_leaf = 5
- n_jobs = -1
- Trained on top 30 LightGBM-selected features with polynomial degree 2

Preprocessing:

- OneHotEncoding for categorical variables using a Pipeline

Performance (R-squared): 0.9818

Details:
Random Forest demonstrated excellent performance by capturing complex interactions between features. The combination of feature selection and polynomial expansion significantly enhanced prediction accuracy.

6. XGBoost Regressor

Objective:
To model player market value using boosted decision trees that sequentially reduce prediction errors.

Configuration:

- xgb.XGBRegressor
- n_estimators = 500
- learning_rate = 0.05

- max_depth = 7
- subsample = 0.8
- colsample_bytree = 0.8
- random_state = 42, n_jobs = -1

Preprocessing:

- OneHotEncoding for categorical features within a Pipeline

Performance:

- Test R-squared: **0.9156**
- Mean Cross-Validation R-squared: **0.9069**

Details:
XGBoost delivered strong and stable performance. Although slightly lower than Random Forest, it remained effective and demonstrated good generalization across folds.

7. LightGBM Regressor

Objective:
To efficiently predict player market value while handling large feature spaces and providing feature importance insights.

Configuration:

- lgb.LGBMRegressor
- n_estimators = 500
- learning_rate = 0.05
- num_leaves = 31
- max_depth = -1
- min_child_samples = 20
- random_state = 42, n_jobs = -1

Preprocessing:

- OneHotEncoding for categorical features using a Pipeline

Performance:

- Test R-squared (all features): **0.9629**
- Mean CV R-squared: **0.9554**
- Test R-squared (top 20 features): **0.9891**

- Validation R-squared: **0.9736**

Details:
LightGBM balanced speed, accuracy, and interpretability. Feature importance extracted from this model played a crucial role in feature selection for other models.

Sr. No.	Model Name	Model Type	Performance Metric	Score (R^2)
1	Linear Regression	Regression (Baseline)	R-squared	**0.78**
2	Polynomial Regression (Degree 3)	Regression (Non-linear)	R-squared	**0.82**
3	Lasso Regression (L1 Regularization)	Regularized Regression	R-squared	**0.9674**
4	Forward Feature Selection (Linear Regression)	Feature-selected Regression	R-squared	**0.9767**
5	Random Forest Regressor (Optimized)	Ensemble Regression	R-squared (Test Set)	**0.9818**
6	XGBoost Regressor	Gradient Boosting Regression	R-squared (Test Set)	**0.9156**
7	LightGBM Regressor (All Features)	Gradient Boosting Regression	R-squared (Test Set)	**0.9629**
8	LightGBM Regressor (Top 20 Features)	Gradient Boosting Regression	R-squared (Test Set)	**0.9891**

CHAPTER 6: MODEL SELECTION

6.1 Objective of Model Selection

The primary objective of model selection in this study is to identify a predictive model that delivers **high accuracy**, **robust generalization**, and **interpretability** for estimating football player market value. Given the complex and non-linear relationships between performance, injury history, transfer activity, team context, and public sentiment, the selected model must effectively handle high-dimensional data while avoiding overfitting.

6.2 Comparative Analysis of Models

Multiple regression models were evaluated using the **R-squared (R^2)** metric, which measures the proportion of variance in player market value explained by the model.

- **Linear Regression ($R^2 = 0.78$)** served as a baseline, capturing only linear relationships.
- **Polynomial Regression ($R^2 = 0.82$)** improved performance by modeling non-linear interactions but showed limited scalability for higher dimensions.
- **Lasso Regression ($R^2 = 0.9674$)** demonstrated strong predictive power while performing automatic feature selection, reducing model complexity.
- **Forward Feature Selection with Linear Regression ($R^2 = 0.9767$)** confirmed that a small subset of engineered features could explain a large portion of variance.
- **Random Forest Regressor ($R^2 = 0.9818$)** effectively captured complex interactions and reduced variance through ensemble learning.
- **XGBoost Regressor ($R^2 = 0.9156$)** delivered competitive performance but was comparatively less effective in this dataset.
- **LightGBM Regressor (All Features) ($R^2 = 0.9629$)** provided efficient learning and strong performance while offering feature importance insights.

6.3 Final Model Selection: LightGBM Regressor (Top 20 Features)

Based on empirical evaluation, the **LightGBM Regressor trained on the top 20 selected features** achieved the **highest predictive performance with an R^2 score of 0.9891**, making it the most suitable model for final market value prediction.

Reasons for Selecting LightGBM (Top 20 Features)

1. **Highest Predictive Accuracy**
   The model achieved the best R^2 score among all evaluated models, indicating superior explanatory power for player market value.
2. **Effective Handling of Non-linearity**
   LightGBM efficiently models complex, non-linear relationships between player performance, injuries, transfers, and market dynamics.
3. **Feature Efficiency and Reduced Overfitting**
   Training on the top 20 most important features minimized noise and redundancy, leading to better generalization on unseen data.
4. **Computational Efficiency**
   LightGBM's histogram-based learning enables faster training and lower memory usage compared to traditional boosting algorithms.
5. **Interpretability through Feature Importance**
   The model provides clear feature importance scores, allowing domain-level interpretation of factors influencing market value.
6. **Scalability and Robustness**
   LightGBM scales well with large datasets and high-dimensional features, making it suitable for real-world football analytics systems.

6.4 Final Prediction Framework

The selected LightGBM model was used as the **final prediction engine** for estimating player market value. The model consumes preprocessed and engineered features derived from player profiles, performance metrics, injury indicators, transfer history, team context, and sentiment analysis.

This model forms the foundation for the predictive system presented in the subsequent chapter, where results, evaluation, and practical implications are discussed.

Here's a polished **next chapter** describing the deployment, updated to include **FastAPI** along with Streamlit and ngrok, suitable for your thesis or report:

CHAPTER 7: MODEL DEPLOYMENT

7.1 Objective of Deployment

The deployment phase focuses on making the trained machine learning model accessible to end-users through an **interactive, web-based interface**. The goal is to allow real-time predictions of football player market values while maintaining usability, efficiency, and scalability.

Two main deployment frameworks were utilized:

1. **Streamlit** – for a user-friendly GUI for non-technical users.
2. **FastAPI** – for a programmatic interface and API-based access, enabling integration with other applications or services.

Ngrok is used to **expose the local environment** to the internet for demonstration and testing purposes without a dedicated server.

7.2 Deployment Architecture Overview

Components:

1. **Trained Machine Learning Model**
    - Model: LightGBM Regressor trained on the **top 20 selected features**.
    - Serialized using Python's pickle as lightgbm_model.pkl.
    - Loaded at runtime by both Streamlit and FastAPI applications.
2. **User Interface (Streamlit)**
    - Sidebar inputs for 19 important features (e.g., total_career_matches, age, total_transfer_fees, value_per_goal, market_value_to_age_ratio).

- Default values are provided for ease of testing.
- Users click a **Predict Market Value** button to receive predictions in real-time.
- Predicted market value is displayed in a clear format

3. **API Interface (FastAPI)**
   - Provides an endpoint /predict to send **JSON payloads** of player features.
   - Receives feature inputs programmatically, constructs a DataFrame in the correct feature order, and returns the predicted market value.
   - Enables integration with other applications, web services, or automated pipelines.
4. **Ngrok Tunneling**
   - Creates a **secure, publicly accessible URL** for the local Streamlit or FastAPI server.
   - Allows external users to access the application from anywhere without a dedicated hosting server.
   - NGROK_AUTH_TOKEN is set as an environment variable for secure authentication.
   - The tunnel runs on the specified local port (default 8501 for Streamlit, or another for FastAPI).

7.3 Deployment Workflow

Step 1: Model Saving and Loading

- Save the trained LightGBM model:

```
import pickle
with open('lightgbm_model.pkl', 'wb') as f:
    pickle.dump(lightgbm_model, f)
```

- Load the model in the deployment environment:

```
with open('lightgbm_model.pkl', 'rb') as f:
    loaded_model = pickle.load(f)
```

Step 2: Streamlit Application (app.py)

- Sidebar input fields using st.sidebar.number_input for all relevant features.
- Prediction triggered by a button, which passes input values to loaded_model.predict().

- Display results on the main page using st.write().

Step 3: FastAPI Application (main.py)

- Define API endpoint /predict:

```
from fastapi import FastAPI
import pandas as pd
app = FastAPI()
@app.post("/predict")
def predict_player_value(data: dict):
    df = pd.DataFrame([data])
    prediction = loaded_model.predict(df)[0]
    return {"predicted_market_value": prediction}
```

Step 4: Ngrok Integration

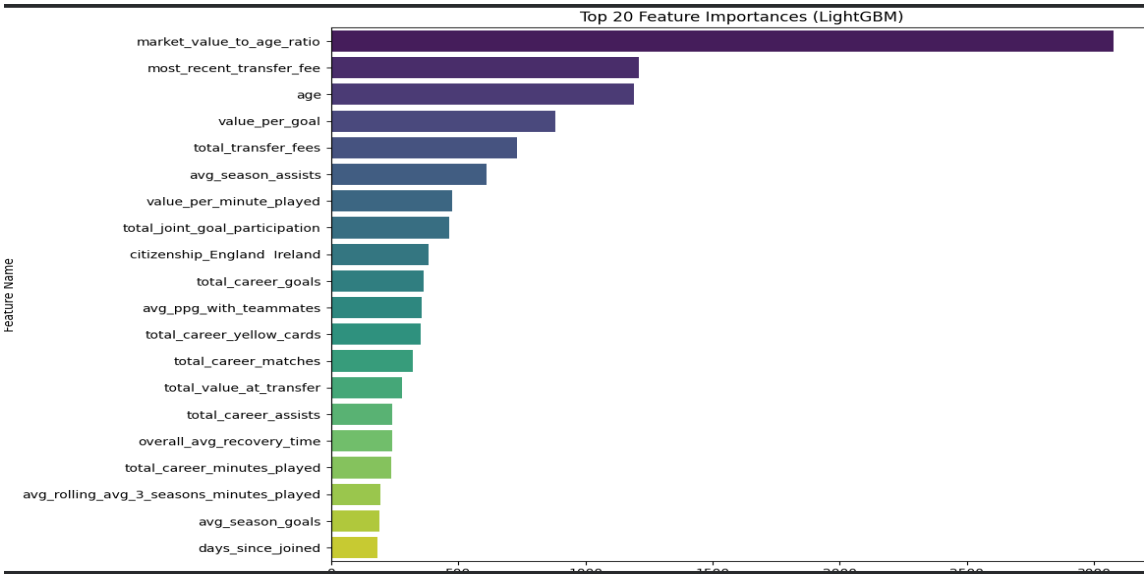
- Connect local server to public URL:

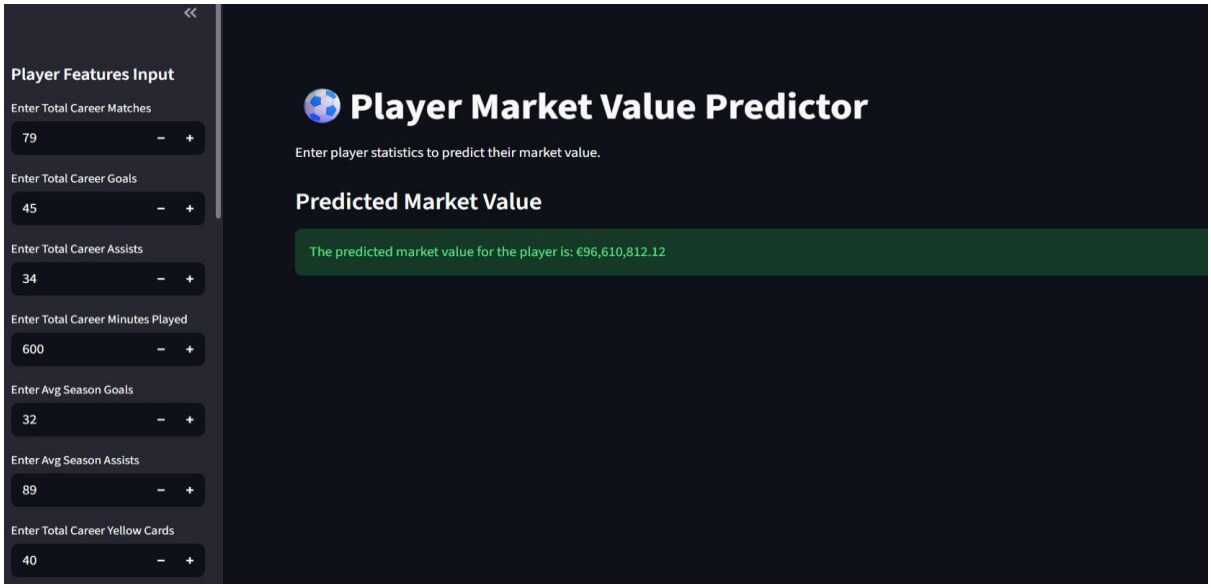
```
from pyngrok import ngrok
public_url = ngrok.connect(port=8501)  # Streamlit port
print("Public URL:", public_url)
```

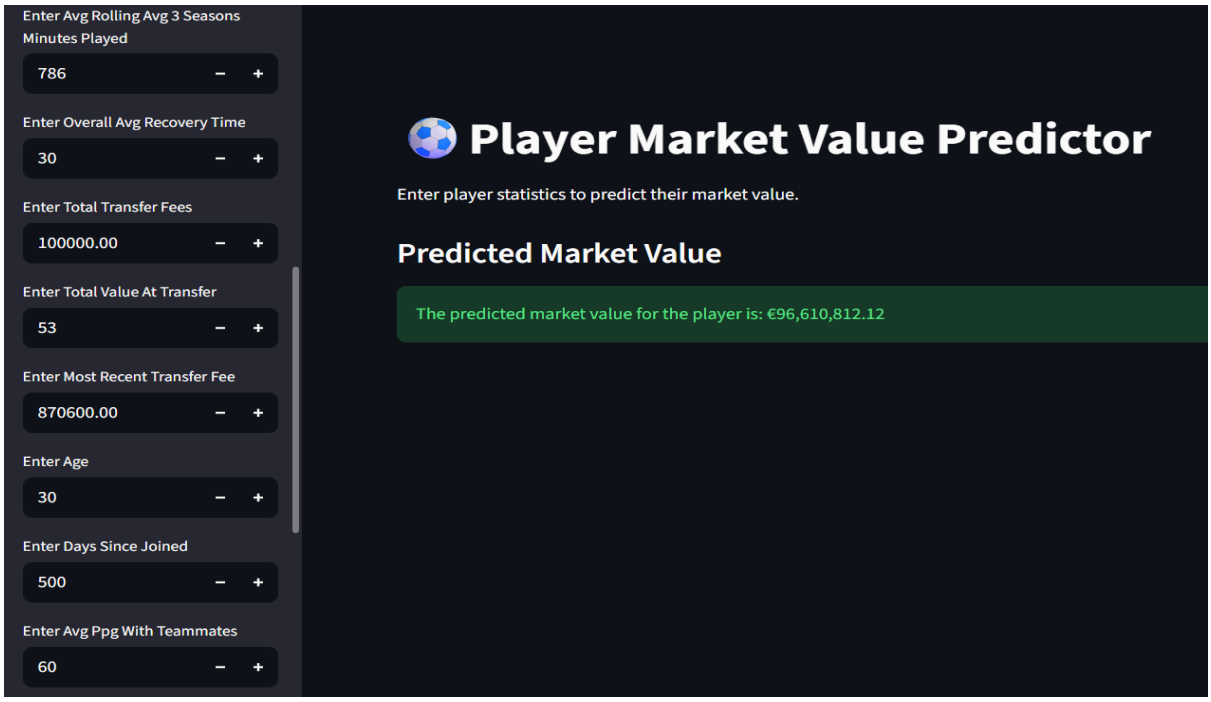
- Streamlit or FastAPI server can now be accessed externally via the generated URL.

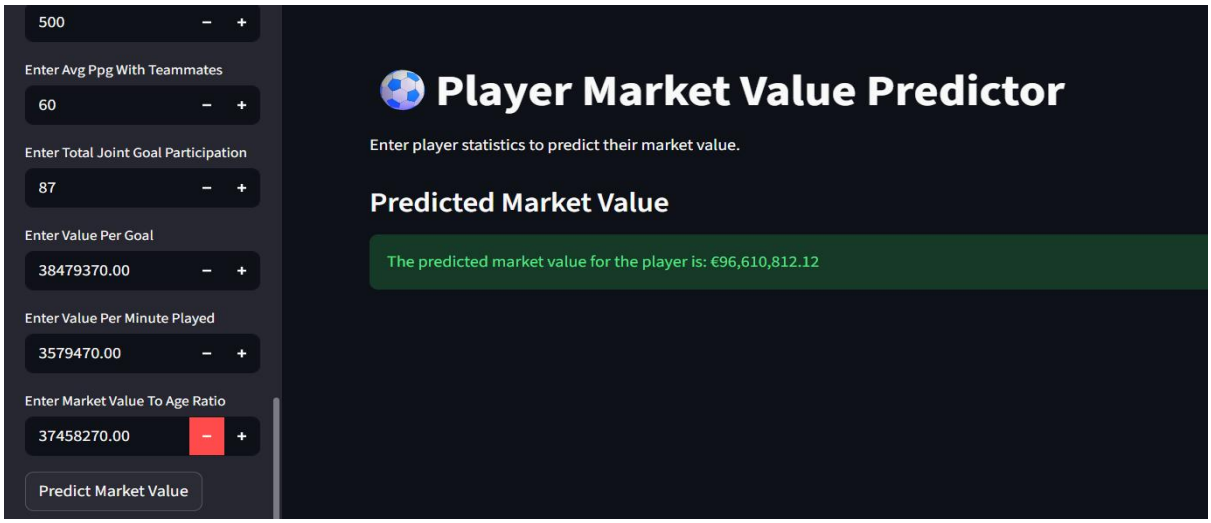
1. FastAPI share the same preprocessed feature set and model, ensuring consistency.

CHAPTER 8: RESULT




Player Features Input
Enter Total Career Matches
79
Enter Total Career Goals
45
Enter Total Career Assists
34
Enter Total Career Minutes Played
600
Enter Avg Season Goals
32
Enter Avg Season Assists
89
Enter Total Career Yellow Cards
40
Player Market Value Predictor
Enter player statistics to predict their market value.
Predicted Market Value
The predicted market value for the player is: €96,610,812.12


Enter Avg Rolling Avg 3 Seasons Minutes Played
786
Enter Overall Avg Recovery Time
30
Enter Total Transfer Fees
100000.00
Enter Total Value At Transfer
53
Enter Most Recent Transfer Fee
870600.00
Enter Age
30
Enter Days Since Joined
500
Enter Avg Ppg With Teammates
60
Player Market Value Predictor
Enter player statistics to predict their market value.
Predicted Market Value
The predicted market value for the player is: €96,610,812.12


500
Enter Avg Ppg With Teammates
60
Enter Total Joint Goal Participation
87
Enter Value Per Goal
38479370.00
Enter Value Per Minute Played
3579470.00
Enter Market Value To Age Ratio
37458270.00
Predict Market Value
Player Market Value Predictor
Enter player statistics to predict their market value.
Predicted Market Value
The predicted market value for the player is: €96,610,812.12

Chapter 9: CONCLUSION & FUTURE WORK

9.1 Future Work

To further enhance and expand the project's capabilities, the following areas are recommended:

1. Advanced Model Optimization: Use techniques like Bayesian Optimization for exhaustive hyperparameter tuning of top models.
2. Exploring Additional Model Architectures: Consider neural networks (MLPs, RNNs) or complex ensembles (CatBoost, Voting/Stacking).
3. Dynamic Feature Engineering: Capture player form and team trends using time-series features over granular periods.
4. Incorporating External Factors:
    - Team performance metrics (league standing, recent matches)
    - Media attention and sentiment analysis beyond Twitter
    - Economic indicators affecting transfer market
    - Contract clauses like release clauses and bonuses
5. Interpretability Enhancements: Apply SHAP or LIME for regression models to explain individual predictions.
6. Interactive Dashboard Development: Expand Streamlit app into a full dashboard for comparisons, trend visualizations, and dynamic feature insights.

These enhancements will improve predictive accuracy, expand the analytical scope, and ensure long-term usability and maintainability of the market value prediction system**.**

9.2 Conclusion

The project successfully built a powerful analytical framework capable of providing valuable insights into football player market dynamics. The developed models can assist clubs, agents, and analysts in strategic decision-making, offering both accurate predictions and interpretable outputs. The framework demonstrates the practical potential of integrating advanced machine learning with rich football data for real-world applications.