Football Player Scouting and Recruitment: Performance Prediction Using Player Skills and Injuries with Machine Learning

MSc Research Project
Data Analytics

Abhilash Janardhanan
Student ID: X23121424

School of Computing
National College of Ireland

Supervisor: Shubham Subhnil

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Abhilash Janardhanan
Student ID:	X23121424
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Shubham Subhnil
Submission Due Date:	12/12/2024
Project Title:	Football Player Scouting and Recruitment: Performance Prediction Using Player Skills and Injuries with Machine Learning
Word Count:	XXX
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Abhilash Janardhanan
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Football Player Scouting and Recruitment: Performance Prediction Using Player Skills and Injuries with Machine Learning

Abhilash Janardhanan
X23121424

Abstract

Recruiting and scouting football players requires a more complex understanding of both the skills and injury history of a player for optimal team performance. Most of the existing methods might look at these factors separately which might in turn overlook how a player's health and their skills affect each other. Current techniques might not fully capture the holistic view necessary for making crucial decisions while scouting and recruiting players. To address these shortcomings and gaps this study introduces a new approach which integrates both player features and injury data to predict future performance more comprehensively and effectively. A new performance metric is developed that combines key skill attributes with injury data. The study tries to provide a more complete and balanced evaluation of football recruits and scouts. The initial results from the implementation of this combined model are encouraging. They suggest that it could improve how players are assessed by considering both player's ability and injury risks. Further research aims to improve these predictions by including additional factors such as psychological attributes and how well can a player fit into team strategies.

1 Introduction

Scouting and recruiting players for football have always been the basics of building teams in professional football. This process has been successful in the game and outside the field. Traditionally, scouting has been based on the subjective appraisals of technical abilities and physical attributes of players. However, as football clubs operate more and more in highly competitive environments with significant financial stakes, there is a need for a more objective approach that will allow the analysis of players based on large data sets. Machine learning would allow a club to identify meaningful patterns in large data sets, which could transform the way clubs assess players and their risk of injury. This study focuses on using machine learning to predict football player performance by integrating technical attributes and injury histories, thereby providing a comprehensive tool for recruitment decisions.

1.1 Motivation

What motivates this research is that nowadays, with the advent of modern recruitment, two pressures surround football clubs: improving performances of their players, along with minimizing injuries and, correspondingly, injuries cost-related risks. Injuries indeed trouble football professionally as well as tactically. Injuries can result in huge financial losses for football teams because even if a player is injured clubs have to pay them their wages without the player contributing to the club and this often accounts for a large portion of annual player expenses. More so, injury often impacts team dynamics and performance. Therefore, player selection needs to balance ability and resilience. In addition to this, the growing interest in advanced analytics in sports shows how information-driven decisions help different teams (23). Clubs are increasingly utilizing data analytics to refine performance analysis. Still, fewer clubs include injury data into their analyses. Existing studies focused on either technical skill assessment or injury prediction ; fewer attempted to combine these two dimensions to formulate a comprehensive evaluation metric. This work fills this gap, introducing a new performance metric that merges both the data of skill and injury (18).

In addition to theoretical benefits, the practical benefits to be gained from this system are enormous. Football clubs spend a lot of money scouting for players, as is often done by using intuition or subjective opinions (4). Machine learning can help in the process, reducing biases, improving accuracy, and saving time. A model that predicts a player's overall performance, including skills and injury risks, will help guide clubs in better recruitment decisions. Besides, hosting such a model as a web application makes the club have access to real-time predictions, thus simplifying the recruitment process.

Research Questions

- **Integration of Data:** How can historical injury data and player skill attributes be combined to effectively predict a football player's overall performance?
- **Model Accuracy:** Which machine learning model provides the most accurate predictions of player performance based on the chosen features?
- **Real-time Implementation:** How can a machine learning-based system be implemented to assist in real-time decision-making for football scouting and recruitment?

Study Objectives

- **Create a Unified Dataset:** Combine player injury records with skill-based attributes.
- **Engineer a Performance Metric:** Develop a composite metric called performance score that penalizes players for their injury history and rewards technical skills.
- **Model Training and Evaluation:** Train and evaluate multiple machine learning models (Linear Regression, Decision Tree, Random Forest and XGBoost Reegressor) to identify the best model for predicting player performance.
- **Deployment:** Deploy the chosen model through an easy-to-use Flask-based web API for implementation in football recruitment.

2 Related Work

Machine learning in football analytics has transformed the very face of player performance analysis, injury prediction, and team decision-making. Football has now evolved into a data-intensive sport where clubs are making the best use of machine learning for the first time ever to identify insights from large datasets of player attributes, injury histories, and match performances. It outlines the key contributions in this area, which may be broadly categorized into three categories: performance modeling, injury prediction, and applied football machine learning.

2.1 Performance Modeling in Football

The key area that football analytics covers through performance modeling includes rating the players according to technical, physical, and psychological dimensions. Al-Asadi and Tasdemır (1) have shown that machine learning can be used to predict player market value using data from the FIFA video game. Their results highlight the promise of structured datasets for actionable insights in real-world player assessments. Similar Almulla and Alam (2) reported some key performance metrics affecting the match outcomes significantly. They employed machine learning models to identify a few of the features, including passing accuracy, speed, and endurance as the predictors of team success.

Ati et al. (3) in their systematic review generalized this approach by integrating machine learning with multi-criteria decision-making for player selection and performance prediction. Their work provides an integration of technical skill with contextual factors to give it a more robust framework assessing players in a comprehensive sense. This holistic view aligns well with the work that Bongiovanni et al. (4) have highlighted the aspect of anthropometric features of predicting the physical performance of elite youth soccer players. Their machine learning model was, therefore, in a position to effectively correlate the metrics of body composition with the outcomes that would show results in a talent development program.

The underpinning from Kelly et al. (13), who explored talent processes through machine learning in an English football academy, further gives this youth focus. They conclude that early technical and physical profiling of attributes impacts the player's development in significant ways and points towards the use of analytics in building future players. Besides, Markopoulou et al. (16) introduced models predicting possibilities for goal scoring in elite leagues to aid tactical decisions for recruiting and strategy for play.

2.2 Injury Prediction and Prevention

Injury prediction has emerged as a new frontier in football analytics for the reasons of financial and competition costs involved in player unavailability. Injuries disturb teams' flow and most importantly lead to significant monetary losses so predictive tools are invaluable for implementing preventive strategies. Nassis et al. (19) discussed machine learning applications in soccer; they emphasized the use of predictive analytics to mitigate potential risks of injury. Their result showed a significant improvement in injury prevention with the help of historical as well as physiological data.

This is well shown by the work of Rommers et al. (22) who used machine learning to identify injury risk in elite youth football players by relating them to historical injury, training loads, and physiological markers. Using these factors, their model was able to identify risk patterns. Another study by Van Eetvelde et al. (24) reviewed systematically all machine learning approaches for predicting and preventing injuries, further emphasizing algorithms that detect risk early. These studies highlight the need to deploy predictive models for better health and availability of the players.

Prys et al. (20) suggest an intelligent system based on the machine learning model and contextual data to predict and manage injury. The model will utilize both real-time and historical data for actionable insights for the medical teams to take the correct interventions. This is aligned with the work of Mandorino et al. (15), who designed a machine learning tool to predict the fitness status, thus having practical applications in the optimization of training regimens and management of workloads.

Chang et al. (6) examined the association of workload with injuries and performance for the players of the English Premier League. The paper used machine learning for the determination of the training intensities that can optimally be achieved, provided that they minimize injury risks while maximizing performance. An approach like this can potentially integrate workload management into holistic prevention strategies.

2.3 Practical Applications in Football

Beyond performance modeling and injury prediction, applications of machine learning include scouting, transfer market analysis, and tactical planning. Ghar et al. (11) proposed a scouting assistant which simulates the performance of the player based on machine learning predictions. Their model has enabled scouts to evaluate various conditions of players using this data-driven add-on to traditional methods. Likewise, Sulimov (23) provided AI-based tools for transfer fee estimation as an additional example of monetary effects that machine learning can bring to the game of football.

Dijkhuis et al. (9) showed an application of machine learning in tactical planning through the development of models for predicting performance in physical activities while playing elite soccer matches. Therefore, their approach allows for real-time decisions like making substitutions to maintain team performance. Similar to this, Wisdom and Javed citewisdom2023 mentioned that machine learning has a strategic value in quantifying performance and enhancing game strategies.

McHale and Holmes (17) applied machine learning to transfer market analysis and estimated the value of players using sophisticated performance metrics. The study clearly shows the potential of applying machine learning in financial decision-making so that clubs invest cheaply. Similarly, Chandra (5) proposed integrated models that use technical, psychological, and tactical attributes for player evaluation and argued for a holistic approach to scouting and recruitment.

Although there was huge development, integration of machine learning into football analytics is still far from optimal as data quality becomes a bigger challenge with varying collection procedures and fewer accessibility to a better quality dataset which reduces its predictive accuracy. For example, in their recent work Wisdom and Javed (25) also said that proper standardized framework on data will boost the preciseness of predictive models. Algorithm interpretability remains a challenge, at least for complex models such as neural networks. Explainability and transparency will therefore be necessary to build stakeholders' trust and enable actionable insights.

Scalability is another issue; in case of deployment to different teams and leagues, machine learning models need to be robust to handle different sources and conditions. Rico-González et al. (21) pointed to more robust systems for handling diverse sources and conditions. They advocated collaboration between researchers and practitioners to bridge the gap between theoretical advancements and practical implementation.

Future studies should be on the expansion of data sources into psychological and tactical metrics, which are sometimes overlooked in models. Chandra (5) suggested including such dimensions in the assessment of players, hence giving a direction for the development of more holistic models. Furthermore, Ati et al. (3) and Nassis et al. (19) presented multi-criteria frameworks that involve technical, physical, and contextual considerations to make better decisions.

Others involve applying real-time analytics in decision making for players within games. Dijkhuis et al. (9) demonstrated that substitution may be optimized, along with players' workload control through applying the real-time model. For example, Mandorino et al. (15) showed that monitoring fitness will ensure that enough players are available and playing at the right time for their games.

It represents a complete paradigm shift with football analytics, where data-driven methods can be used in the evaluation, management of injuries, and tactics. This offers a chance for researches and practitioners to find hidden information using advanced algorithms, allowing better decision-making and operational efficiency in the game. Obstacles to be tackled if this potential is to be unleashed: data quality, interpretability of algorithms, and scaling.

More in the future research agenda should be holistic models, which would consider data from technical skills to psychological attributes. The interface between academia and industry would also play a role in bridging the gap between research and practice, thus making innovation in football analytics more feasible.

3 Methodology

This project was carried out with a systematic approach towards data collection, preprocessing, integration, feature engineering, model development and evaluation in order to predict the performance scores of players from their injury histories and performance metrics. This section elaborates each step in detail, from preparation of data to deployment of models.

3.1 Data Collection

In this project we will consider two datasets one is the injury dataset and the other one is player statistics dataset. The injury dataset contained detailed information about how many days each player is injured, how many season games they play, cumulative amount of minutes played as well as overall FIFA ratings. These were applied in subsequent decision-making processes to estimate players' potential during their time at a given season. The player statistics data provided key performance attributes in the form of ball control, dribbling, pace, physical strength, attacking position, crossing, stamina, short pass, curve, finishing and long shots. The dataset also contained other features like the market-related data, such as player value, making the analysis have yet another dimension.

Both datasets were read into Python using the pandas library which is famous for its capabilities in handling tabular data. The ISO-8859-1 encoding was used to read the statistics on players dataset in order to handle any non-ASCII characters that might appear in player names or other text fields. This transition nicely flows into the data cleaning phase.

3.2 Data Cleaning

Data cleaning is always a necessary step in order to preprocess the datasets for further analysis. We start by looking into the data types of each column and thereby performing initial data exploration to identify any issues. We looked into summary statistics and distributions so that we can understand the data better and identify potential outliers. Normalization of player identifiers from both datasets was done. This means removing unwanted whitespaces and converting all text to lowercase. This ensures that there is proper matching during the merge step. A compact version of player name from player statistics dataset was created inorder to match with players from the injury dataset. This preprocessing step would remove potential discrepancies because of capitalization and formatting differences of player names.

Missing values were another problem. The marking column in player stats data has all values as NaN so we dropped it. Null values were searched for in both datasets, and any blanks found in the entries were filled with zeroes using the fillna() function. This was done not to interrupt the analysis phase, especially the feature engineering and modeling stages. Further cleaning was required for the value column in the player statistics dataset. This column was a measure of player market value. It contained currency symbols and commas, making it not numerical and incompatible with arithmetic operations. That was the removal of the symbols by using regular expressions in transforming the column to numerical format for easy analysis.

Two datasets were then merged into a single dataset with injury histories and player performance metrics for each. We will merge the Injury Data and Player Stats Data to create a unified dataset that combines both player injury information and performance statistics. This was done by using the identifier columns for players, where p_id2 from the injury dataset was used in conjunction with player from the player stats dataset. We will perform an inner join merging the Injury Data and Player Stats Data based on player identifiers as it will include only rows where both datasets have matching players thereby ensuring the merged dataset contains complete and relevant information. The key columns for the merge are p_id2 from the Injury Data player_compact from the player stats Data. By merging these two datasets we gain access to both performance metrics and injury history for each player. Some columns, such as age, were found to appear in both datasets, and during the merge, they were renamed, such as age_x and age_y.

3.3 Exploratory Data Analysis

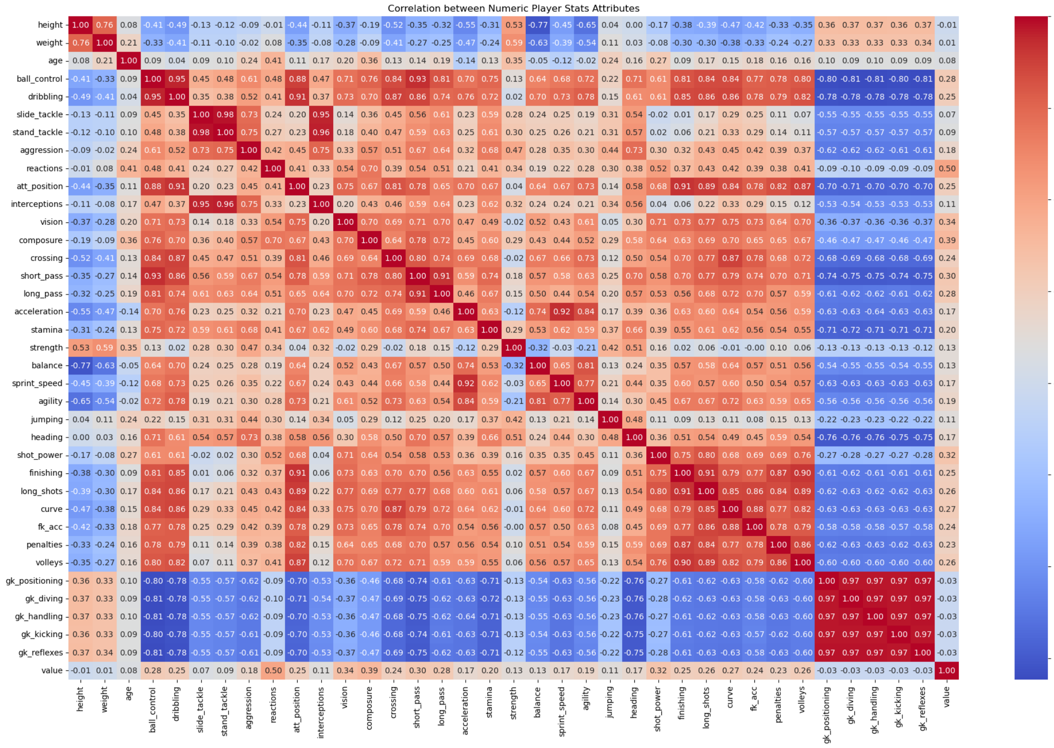

Figure 1: Player Attribute Correlation Heatmap

We generated a correlation heatmap 1 visualize the relationships between the numeric attributes our dataset. This heatmap shows us insights into how different player skills and characteristics can relate to each other for example the strong positive correlation between ball_control and dribbling and the negative correlation between age and sprint_speed. These insights can be very useful for feature selection and engineering for subsequent modeling.

The histogram 2 describes the distribution of total days players have been injured. The x-axis is used to represent the total days injured, and the y-axis indicates the count of players. The higher bars near the start of the x-axis describe the majority of players having shorter injury durations. The number of players drops sharply for longer injury durations. This right-skilled distribution suggests that although short-term injuries are quite common, among players in this dataset, prolonged injuries are quite rare.

Scatter plot "Games Played vs Total Days Injured," 3 shows the relation of football games played in a season, with total days injured for a player. Here in the x-axis indicates from season games played between 0 to 35, while on the y-axis it's from 0 to 2500, about the total days injured. The points for data are spread rather widely and indicate no discernible relationship between the number of games played and the injury time span. Most injuries lie under 500 days for all game counts, meaning most of the injuries incurred are of shorter spans, regardless of the number of games played.

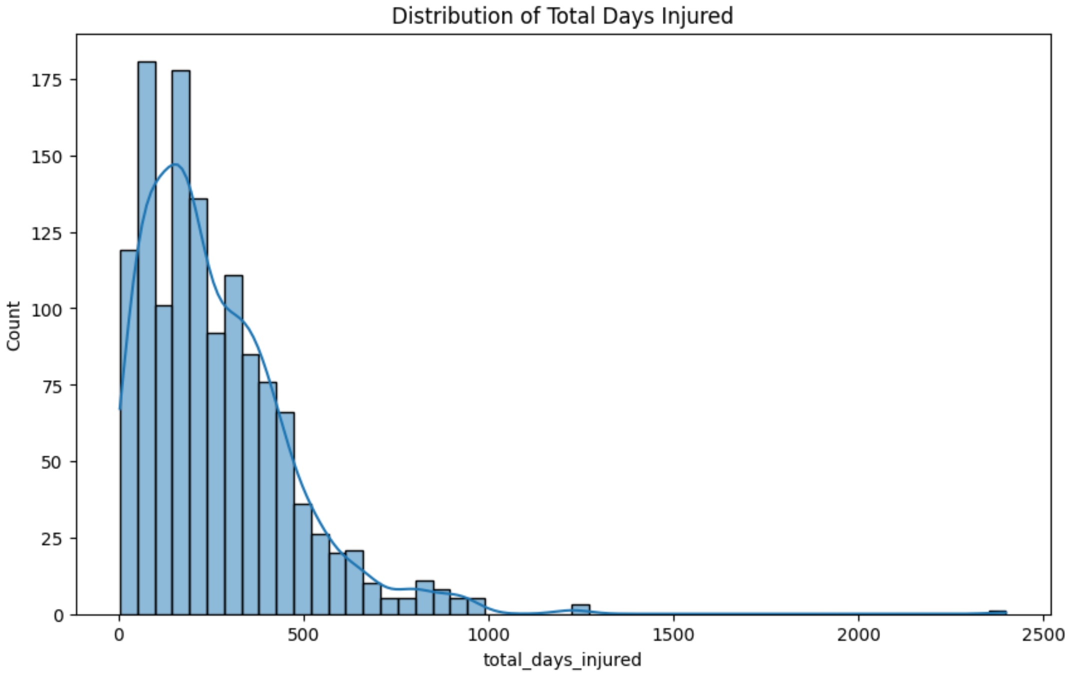

Figure 2: Distribution of Total Days Injured

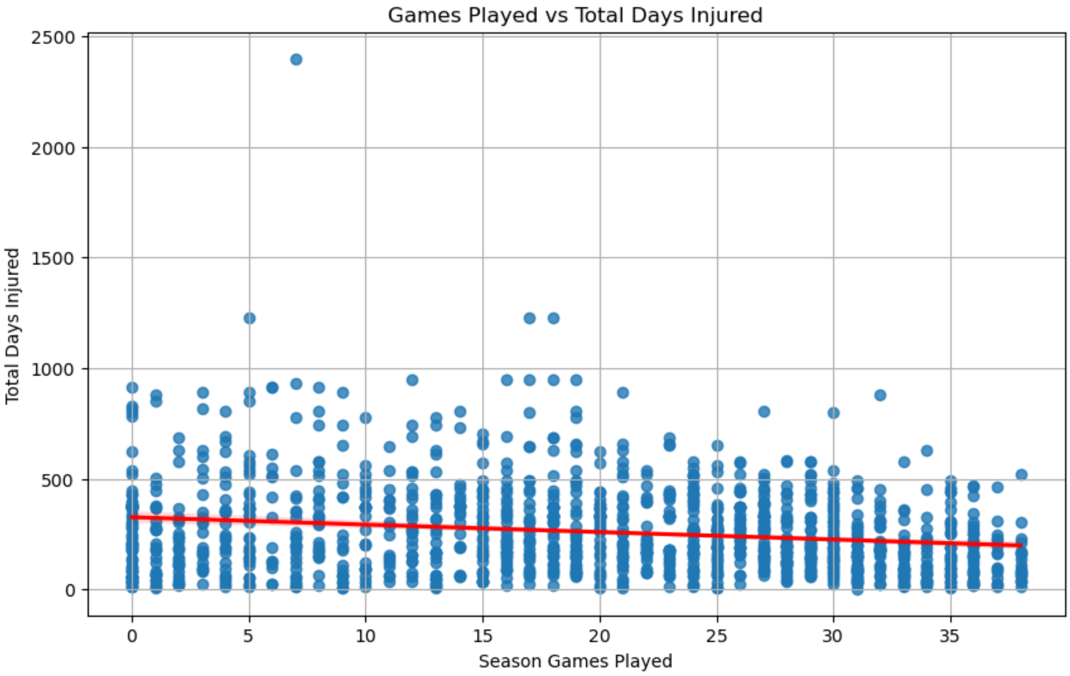

Figure 3: Games Played vs Total Days Injured

3.4 Feature Engineering

Feature engineering was an important part of our methodology, since this would give us the ability to create new variables and improve the predictive power of our model. A specific metric called performance_score is used to calculate the overall performance of a player. This will be our target variable. This performance score is the summation of the ball control and the dribbling rating minus total days injured as follows:

$$\text{performance\_score} = \text{ball\_control} + \text{dribbling} - \text{total\_days\_injured}$$

The logic behind this formula was simple i.e players with higher ball control and dribbling skills should have higher scores but frequent injuries would negatively impact their performance. This metric was designed to balance a player's technical abilities with their availability and durability.

To further enhance the metric and make it more comprehensive and robust we added additional steps to scale the performance score and integrated the FIFA rating as a weighting factor. The FIFA rating was selected mainly as it provides a very comprehensive evaluation of a player's abilities thereby serving as a very reliable benchmark for comparing different players across different skills and attributes. The revised performance score calculation is done by 1st normalizing the total_days_injured. The injury days were scaled to a 0-100 range by using min max normalization to standardize the impact. After this the performance score will now integrate the fifa_rating thereby balancing it with player's technical skills and injury histories.

$$\text{performance\_score\_final} = \left(\frac{\text{ball\_control} + \text{dribbling} - \text{total\_days\_injured\_normalized}}{\text{fifa\_rating}} \right) \times 10$$

In the end a final scaling was done to ensure consistency and interpretability. The performance score was further scaled to a range 0-100. The min_score represents the lowest performance score in the dataset meanwhile the max_score represents the highest. These values are basically used to scale all scores in a range of 0 to 100.

$$\text{performance\_score\_final\_scaled} = \left(\frac{\text{performance\_score\_final} - \text{min}}{\text{max} - \text{min}} \right) \times 100$$

3.5 Feature Selection

Features relevant to player performance 4 were carefully selected after visualizing the correlation with target variable to train machine learning models. The features used included the following:

- **ball_control**: A measure of a player's technical skill with the ball.
- **dribbling**: A rating of how well a player can maneuver with the ball.
- **att_position**: Player's attacking position awareness.
- **crossing**: Crossing ability rating.
- **stamina**: Stamina level of the player.
- **short_pass**: Short passing ability rating.
- **curve**: Curving ability in shots and passes.
- **finishing**: Finishing skill in scoring goals.

- **long_shots**: Ability to take accurate long-range shots.
- **total_days_injured**: The total days the player was injured within the season.
- **season_games_played**: Total games the player played in a season.

The target variable that would be predicted is performance_score_final_scaled.

Negative features related to goalkeeper were avoided because they are only relevant for goalkeepers and do not contribute to other players but other features can contribute to goalkeeper selection.

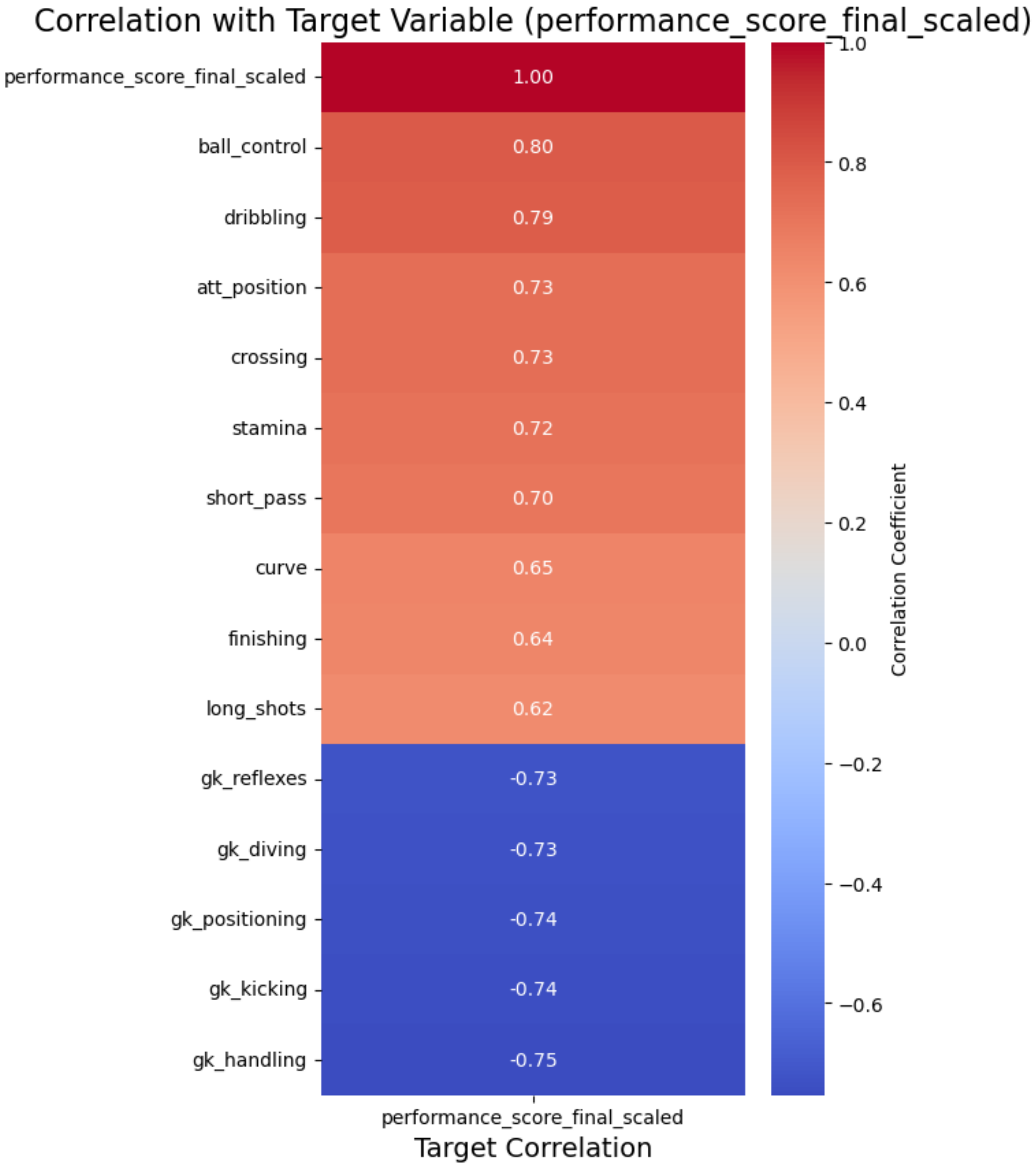

Figure 4: Correlation with Target Variable

3.6 Data Splitting

The data was split into training and testing sets using an 80:20 ratio to prepare the data for model training and evaluation. The training set was used to train the models, while the testing set was reserved for evaluating model performance on unseen data. This was done using the train_test_split function from the scikit-learn library with a fixed random seed (random_state=42) to ensure reproducibility.

3.7 Model Development

The four machine learning models that we developed for the prediction of performance score were Random Forest, Linear Regression, Decision Tree and XGBoost Regressor. We used these four models as they balance robustness and accuracy and also as discussed in related work these models have shown better performance in similar prediction projects. Random Forest uses multiple decision trees to make the predictions more accurate and robust against overfitting. Linear Regression is most popular for simplicity and interpretability in modeling a linear relationship. Decision Tree Regressor usually involves data segmentation based on values of the features to form decisions. XGBoost Regressor is powerful ensemble method that uses gradient boosting to build multiple decision trees sequentially thereby optimizing performance and handling complex patterns very efficiently. These models were then trained and tested against a particular data set and Random Forest had 100 estimators set. Effectiveness of a model is measured by metrics such as Mean Squared Error (MSE) and R-squared (R^2). It is better in performance for lower MSE and high values of R^2. Additionally a further cross validation test by 5-fold was also used to obtain mean and standard deviation of MSE for every model. After this we did Hyper parameter tuning to enhance the models performance in which we used grid search. We used grid search as it was executable in our dataset. Then MSE and R-squared values are represented graphical bar chart generated by Plotly to make comparisons and choose the best model. The best model was then saved using the joblib library so that it could be reused in other real applications also making sure the model is scalable and useful in actual football scouting and recruitment.

3.8 Model Evaluation

The criteria chosen were MSE and R-squared as the former is the measure that highlights larger errors in the prediction and the latter measures the proportion of variance in the dependent variable that can be explained by the model and shows the efficiency of the model. The other measures such as MAE or RMSE are less sensitive to larger errors or perhaps less required for comparison between models with constant predictors.

- **Mean Squared Error (MSE):** This measures the average of squared differences between the actual and the predicted values. Lower values are a measure of better performance.

- **R-squared ($\mathbf{R^2}$):** This is the proportion of the variance in the target variable that it explains; high values indicate good predictive accuracy.

For further evaluation of the models, 5-fold cross-validation was done. This was by dividing the data into five folds where the model was trained on four folds and tested on one fold. The procedure was repeated five times. Then, the mean and standard deviation of the MSE for each model were calculated. For hyperparameter tuning a grid search was performed to find the best combination of parameters for each model and it involved testing different values for key parameters then training the model on the data and finally evaluating performance using 5 fold cross-validation. In the end best parameters were selected based on the lowest Mean Squared Error (MSE).

3.9 Model Selection and Deployment

The best model was then selected for deployment based on the highest R-squared value. The joblib library is used to save this model thereby allowing it to serialize and save the model for future use without retraining. This means that we can use the model to achieve its intended deployment in real-world application. Pre-saving the model means that it's ready to use making it very useful in football recruitment and assessment of players.

4 Design Specification

Architecture diagram 5 of the player performance score prediction illustrates the flow of data from raw input datasets to the deployment of the best-performing machine learning model via a Flask REST API. Key steps include data cleaning, feature engineering, model training, evaluation and deployment. The proposed system relies on systematic architecture, integrating robust data handling, machine learning techniques, and deployment frameworks that are used to predict the scores of player performance based on injury histories and statistical data. This section details the major techniques, architecture, frameworks and associated requirements as well as the functionality of the proposed algorithm.

The project applies different techniques and frameworks to maintain smooth workflow. It makes use of pandas and NumPy libraries for all kinds of data handling operations from loading and cleaning to manipulating. Those tools are very good at the manipulation of tabular data as well as support merging data sets, handling missing values and feature transformation. The two data sets, namely injury data set and player statistics data set merged together form the core structure of the implementation. Missing values in the merged data would have to be handled with care without any loss of data integrity.

Feature engineering was very important part of our project. A custom metric called performance_score_final_scaled was designed to measure a player's performance. The score measures technical skills such as ball control and dribbling etc but penalizes players for too many injuries.

Algorithms from the scikit-learn library provide the basis for a predictive modeling system. Four models were tested, which are: Random Forest Regressor, Linear Regression, Decision Tree and XGBoost Regressor. The Random Forest models is an ensemble technique, which aggregates many decision trees and it's really powerful and accurate. In a nutshell, it's simple, very interpretable as linear regression, and also has very clear, explainable decision rules for the Decision Tree and XGBoost Regressor. These models train using target variable and other features engineered with performance score.

A 5-fold cross-validation is further used to validate the models and assess the variability in their performance for different subsets of data. Then hyper parameter tuning is performed to enhanced the models performance and fine tune them. Thus making sure that the models are good generalizers for unseen data.

Results of model evaluations are represented with the help of Plotly. Plotly is an interactive tool for displaying plots in python. A grouped bar chart was used to compare the MSE and R-squared values of the models. This visualisation helps in understanding and displaying the effectiveness of the models. Once the best model is found it will be

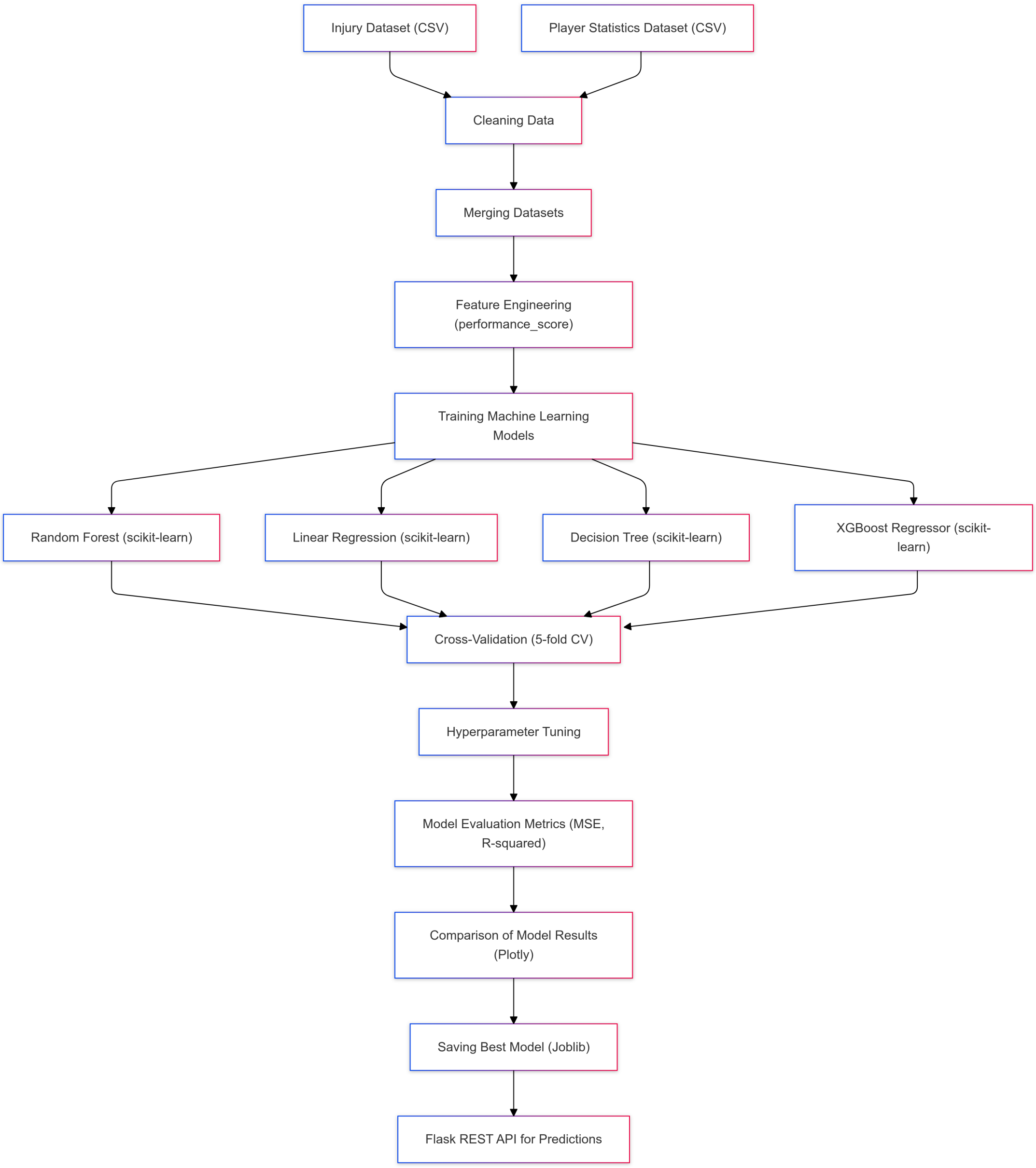
Injury Dataset (CSV)
Player Statistics Dataset (CSV)
Cleaning Data
Merging Datasets
Feature Engineering (performance_score)
Training Machine Learning Models
Random Forest (scikit-learn)
Linear Regression (scikit-learn)
Decision Tree (scikit-learn)
XGBoost Regressor (scikit-learn)
Cross-Validation (5-fold CV)
Hyperparameter Tuning
Model Evaluation Metrics (MSE, R-squared)
Comparison of Model Results (Plotly)
Saving Best Model (Joblib)
Flask REST API for Predictions

Figure 5: Architecture Diagram

saved using the joblib library for later use with efficient reuse. The model is deployed through a Flask REST API, which gives predictions in real-time. The lightweight and modular architecture of Flask makes it a good choice for serving machine learning models, and CORS enables secure communication between the API and any client application.

The hardware and software requirements is basically the prerequisite to running this system. The processor needs to be at least multi-core processor for smooth training and at least 8GB RAM testing the model. For the software part the primary use would have been Python 3.x plus libraries such as pandas, NumPy, scikit-learn, Matplotlib, Seaborn, Plotly, and Flask. The datasets injury and player statistics data should be in clean, consistent and CSV format to smooth out processing. This application can then be served locally or even on remote cloud services like AWS or Heroku, depending on the amount of use anticipated. This is built with Flask so that it can be deployed as a REST API.

5 Implementation

The solution of the proposed method to predict the players performance score through the injury history and player statistics consist of various interconnected stages of several steps thus resulting in a workable and a real-time predictive system. This part addresses the last stage of the implementation wherein the final output would emerge as follows: transformed data; models and tools applied to service the solution.

5.1 Processed Data

The implementation was started with preprocessing and merging two datasets from the injury dataset and the player statistics dataset. The two datasets were converted into a common structure through cleaning, merging and feature engineering process. The outputs of this transformation were as follows:

A single dataset was created by joining the two datasets based on the identifiers between the two datasets i.e the injury dataset would serve as the main dataset with all relevant features obtained from the player statistics dataset.

Engineered Attributes: Introduced new attributes for features that capture all performances of players. Introduced an overall measure performance_score-a composite that can compute the actual performance attributes including technical properties like possession control and ball dribble ability, where it demotes players having long-injury durations in the performance scale. Hence, there is an entirely transformed version dataset with incorporation of original and engineerd features.

The transformed data formed the basis of all the modeling tasks, wherein all the variables were clean and consistent and ready for analysis.

5.2 Models Developed

In the building stage we developed four different types of machine learning models that predict performance score for the players. The chosen models in this project are interpretable and robust, with good performance on numerical features. Outputs from the model building stage include trained models with their performance metrics, cross-validation results and fine tuned results using hyper parameter tuning.

- **Random Forest Regressor:**
  - **Objective:** This model was selected in order to capture the non-linearity between the feature variables and the target. Being an ensemble algorithm, Random Forest is a combination of various decision trees in order to have stable predictions and minimize variability.
  - **Output:** It offers a trained Random Forest that generated very accurate predictions. Feature importance scores, based on the relative contribution of every feature in input for making that prediction on the performance score by the model, was another output of this.
- **Linear Regression:**
  - **Objective:** A baseline model, Linear Regression was included for comparison purposes. Since it is simple and has an easy interpretation of which features

contribute to the target variable, it became apparent how features contributed to it.

  - **Output:** The trained model of Linear Regression explained how linearly the features had relationships with the score obtained from it. The model also gives coefficients for each feature input, which is very handy in interpreting the importance of input variables.

- **Decision Tree Regressor:**
  - **Objective:** It possesses a high degree of simplicity and can easily give out understandable rules for the predictions done. This is helpful in analysing decision-making in data.
  - **Output:** The model trained was a Decision Tree, which explained the clear path of the decision and the rules involved in predicting the score that might occur. The output also featured a tree structure which can be used to view for explaining purposes.

- **XGBoost Regressor:**
  - **Objective:** XGBoost was selected because it handles complex data well. It uses a technique called gradient boosting to build multiple trees one after the other, each one improving on the last. This method is recognized for being very fast with high accuracy.
  - **Output:** The XGBoost model showed very precise predictions and kept errors low. It also provided scores that showed which features were most important in predicting the performance score.

Each model was tested on the basis of its predictive accuracy with the help of key metrics such as Mean Squared Error (MSE) and R-squared (R^2). Cross-validation and hyper parameter tuning were performed in order to test the robustness of the models, and it resulted in mean and standard deviation values for MSE across different folds.

5.3 Performance Metrics

The models gave the following outputs upon performance evaluation

- **Mean Squared Error:** This evaluated the average difference between predicted and actual values. In this case, this measure accounted for how close a model was to the real values of actual performance scores.

- **R-squared (R^2):** This measure captured the percentage of performance score variance explained by the model. The higher R^2 measures, the better was its predictive accuracy.

5.4 Cross-Validation Summary

The cross-validation summaries comprise of Mean Cross-Validation MSE: This is the average MSE across all validation folds, thus providing an estimate of how well the model generalizes.

Standard Deviation of Cross-Validation MSE: This provides a measure of the variability in model performance across different folds, hence the stability of the model.

Cross-validation results showed the robustness of the Random Forest model, yielding the best balance between the accuracy and stability for all folds.

5.5 Hyperparameter Tuning Summary

To further enhance models performance we performed hyperparameter tuning using grid search. For each model a range of parameter combinations was evaluated using 5-fold cross-validation thereby optimizing for Mean Squared Error (MSE). This process identified the best parameter settings for each model thereby balancing accuracy and generalization. Hyperparameter tuning ensured that each model was very well fine-tuned to achieve optimal performance score thereby improving their robustness and accuracy in comparison to the default settings.

5.6 Visualizations Produced

In order to display the results effectively, a number of visualizations were produced during the implementation phase:

Model Performance Comparison: A bar plot 6 of the grouped comparison of the values for MSE and R-squared across all four models is created. The strength of one model against the rest has been well defined based on its strengths and weakness in comparison to other three.

All the above plots are designed using Plotly. Since the outputs have been presented on these plots, their presence increases interpretability and enables much better decision-making in presenting results.

5.7 Deployments Outputs

The deployment stage focused on making the best model available for real-time predictions. Outputs of this stage included the following:

- **Serialized Model:** The best model was saved as a binary file using joblib. This serialized file enabled loading and prediction without the need for retuning the model.
- **REST API:** A REST API was created using Flask to expose the predictions. The API was able to take in input features as JSON and return the predicted performance score as a response. This ensured that it would seamlessly integrate with external systems, like web or mobile applications.

5.8 Code Composed

The code composing was done by writing modular and reusable code for data preprocessing, modeling, evaluation, and deployment. Key outputs from the codebase were:

- **Data Preprocessing Scripts:** Cleans, merges, and transforms the data thereby ensuring that data preprocessing steps are reproducible.

- **Model Training Scripts:** Hyperparameter tuning and cross-validation performed for the four machine learning models, which will be described later.
- **Evaluation Scripts:** Calculates and plots performance metrics.
- **Deployment Code:** A Flask application is designed to take in the API requests and return a prediction. The deployment script included logic that used serialized models and had knowledge of how to handle the input data.

The entire codebase was developed using the Python language with libraries that would include pandas, NumPy, scikit-learn, Plotly, and Flask. These were based on diversity and compatibility, to meet the needs of this particular project.

5.9 Outputs

The last phase of the deployment gives the following outputs.

- **Transformed and Merged Dataset:** The cleaned and combined dataset, now ready for modeling.
- **Trained ML models:** Developed ML models - Random Forest, Linear Regression, Decision Tree, and XGBoost Regressor in that order. This comparison is done through the evaluation.
- **Serialized model:** The saved model is now ready to use.
- **REST API:** An operational API that is invoked in real time for the generation of a prediction. The outputs above result from the process of implementation so that it may provide a strong, scalable solution for predicting a score with respect to player performance.

5.10 Tools and Languages Applied

The project used a mix of tools and languages to generate the outputs:

- **Python:** Main programming language used for data preprocessing, modeling, and deployment.
- **pandas and NumPy:** Libraries used to manipulate and transform tabular data.
- **Scikit-learn:** A comprehensive library for machine learning model development and evaluation. The challenge made use of a mix of gear and languages to generate the outputs.
- **Plotly:** A library for making interactive, dynamically updatable visualizations.
- **Flask:** It is a light-weight web framework that one can run to deploy the REST API, hosting the machine learning model and its deployment.
- **joblib:** This is a library that saves and loads machine learning models.

Utilizing the tools above, this implementation would be able to transform from raw data into a fully working system that is able to deliver actionable predictions in a scalable manner. The outputs of this implementation are a rich resource to understand player performance and to make informed decisions in sports analytics.

6 Evaluation

The initial implementation of all four machine learning models—Random Forest Regressor, Linear Regression, Decision Tree Regressor and XGBoost Regressor—showed the following results for R-squared (R^2) and Mean Squared Error (MSE) as shown in Table 1. These results were calculated on the test set to check the accuracy and reliability of the models.

Table 1: Model Performance Metrics

#	Model	MSE	R-squared
0	Random Forest	23.719805	0.915188
1	Linear Regression	35.053496	0.874663
2	Decision Tree	40.523237	0.855105
3	XGBRegressor	33.027145	0.881908

To further improve the performance and robustness of our models we did a 5-fold cross-validation making sure we achieve stable results by training on four folds and then testing on one fold in iteration. The results of this is displayed in Table 2.

Table 2: Model Comparison after Cross Validation

Model	CV R-squared Mean	CV R-squared Std Dev	CV MSE Mean	CV MSE Std De
Random Forest	0.748466	0.192682	57.106401	23.866678
Linear Regression	0.874490	0.116636	25.220786	11.657493
Decision Tree	0.599541	0.339020	88.149086	37.499271
XGBoost	0.738404	0.146938	67.393423	31.288111

After that we fine tuned by applying hyperparameter tuning to each model to further optimize parameters such as the number of trees, learning rate and depth. This process helped us to enhance the models performance thereby resulting in the updated metrics displayed in Table 3

Table 3: Model Comparison after Hyper Parameter Tuning

#	Model	MSE	R-squared
0	Random Forest	23.662442	0.915393
1	Linear Regression	25.220786	0.874490
2	Decision Tree	29.349569	0.895058
3	XGBRegressor	22.414871	0.919854

Finally in the end the refined and fine tuned models were compared visually using a bar plot 6 which highlights the R-squared and MSE metrics for all models. This visualization makes it easier to identify the best-performing model and in our case it was XGBoost Regressor while Random Forest provides robust generalization.

—

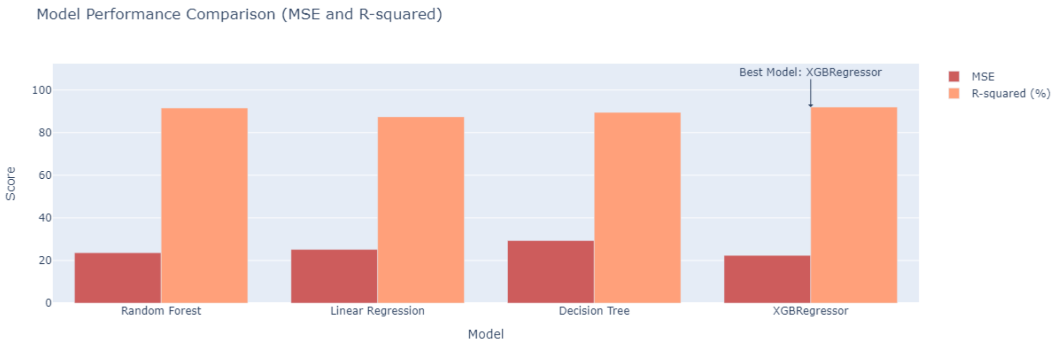

Figure 6: Model Performance Comparison Bar Plot

Detailed Model Analysis

- **XGBoost Regressor**: XGBoost was best performing model in our analysis due to its ability to handle complex patterns in the data. It achieved highest R-squared and lowest MSE values when compared to other three models even after cross-validation and tuning. This model is very robust for handling large datasets and non linear relationships.

- **Random Forest Regressor**: Random Forest performed equally well utilizing multiple decision trees in order to minimize overfitting and also provide reliable predictions thereby making it practical for real-world use.

- **Linear Regression**: This simple model provided insights into the linear relationships between features and the target variable. It provided decent accuracy but it struggled to capture non-linear interactions thereby limiting its overall performance.

- **Decision Tree Regressor**: The Decision Tree model was easy to interpret and it provided clear decision paths and it was prone to overfitting which impacted its stability and generalization as compared to ensemble models like Random Forest and XGBoost Regressor.

Evaluation Summary

XGBoost Regressor displayed the best performance after fine tuning, it showed high accuracy and robustness. Random Forest Regressor was very close to XGBoost by providing balanced results and interpretability through feature importance. Linear Regression lacked the flexibility to handle non linear relationships and Decision Tree Regressor provided simplicity and transparency but was less robust than other two ensemble models.

—

Conclusion

The XGBoost Regressor is the recommended model for predicting player performance scores in my project due to its superior accuracy and ability to handle complex data patterns. Random Forest also performed well but XGBoost's efficiency and precision make it much more suitable choice for our task. These results shows us that the importance of combining multiple evaluation techniques to select the best-performing model for practical applications.

7 Conclusion and Future Work

The primary objective of this research was to develop a strong machine learning model that predicts player performance scores by incorporating other statistical features of players as well as injury history. This research aimed at answering whether it is possible to derive actionable insights from combining these two data sources. This includes preprocessed and merged datasets, feature engineering, machine learning model development and best model deployment through Flask REST API. The developed models were thoroughly tested against two main metrics such as Mean Squared Error (MSE) and R-squared (R^2). It has been of great help in understanding the insights coming out from the Random Forest, Linear Regression, Decision Tree and XGBoost models.

- **Integration of Data: How can historical injury data and player skill attributes be combined to effectively predict football player's overall performance?**

  Historical injury data and skill attributes were successfully merged and integrated into a single dataset and evaluated with a custom metric called performance score. This metric combines technical skills such as dribbling, ball control and with penalties for injuries thereby providing a comprehensive measure of player performance. This approach makes sure both skill and durability factors are considered in the prediction process.

- **Model Accuracy: Which machine learning model provides the most accurate predictions of player performance based on the chosen features?**

  Among the tested models, the **XGBoost Regressor** emerged as the most accurate and robust model with R-squared value of 0.919854 and a Mean Squared Error (MSE) of 22.41487. While the **Random Forest** model achieved equally good R-squared value of 0.915393 and an MSE of 23.662442. The **Decision Tree Regressor** and **Linear Regression** model produced an R-squared value of 0.895058 and 0.874490 respectively.

- **Real-time Implementation: How can a machine learning-based system be implemented to assist in real-time decision making for football scouting and recruitment?**

  The **XGBoost Regressor** model was selected for deployment based on its performance compared to other models. It was saved using the `joblib` library thereby making it reusable and scalable for real-time applications. Additionally the model provides feature importance scores, enabling scouts to identify key performance indicators for decision making. This system was designed for practical use in real-world recruitment scenarios.

This model helped us to answer the research question well, meeting objectives through successful evaluation of the performance of players. In the final model produced from testing for both generalizability and robustness, XGBoost Regressor preformed remarkably well compared to other models. It showed a very high R-squared value and established the strength of modeling. The Random Forest also did a good job with accuracy near to XGBoost and ot tried to minimize overfitting. These results confirm that machine learning models when trained on well-engineered features can effectively predict player performance scores.

The findings have important implications for sports analytics. Identification of some of the main predictors for performance of players by the study contributes a lot toward designing an evaluation framework for talents and their management. For the practical purposes of processing the real-time data and feeding back actionable predictions through API, the system is significantly beneficial for scouting teams, coaches and analysts. There are limitations of this research that still need discussion. First, the utilization of historical data limits how well the model can learn to adapt to unexpected occurrences, such as injuries which may happen after the period of training. Second, the datasets which are used might not even capture all external factors which could affect a player's performance, for example, psychological conditions and team dynamics.

Future work will include the overcoming of these limitations and widening the scope of this research. The meaningful avenue is incorporating live data streams, including live match statistics and wearable sensor data, to make the system more adaptive and accurate. Follow-up research will also be conducted in order to integrate contextual variables such as team formation, opponent strength and weather conditions into the system to evaluate the player's performance in a more holistic way. Advances in explainable AI techniques could also be applied to improve the interpretability of models thus making it clearer for stakeholders.

From the perspective of commercial applicability, the system developed in this research has great applicability in the sports industry. It could be used as a software-as-a-service (SaaS) platform for clubs thereby offering real time analytics and predictions for player performance and recruitment strategies. The can also be used in other sports based on the adaptation of the feature engineering and modeling process to specific performance metrics for each discipline.

In brief the research successfully demonstrated the possibility of using history of injury and player statistics for predicting player performance scores through machine learning. The results and findings give a foundation for further advancement as well as commercialization opportunity in the field of sports analytics based on data-driven decision-making. Addressing such limitations by further research would add more refinements and scope to the system and help significantly in the sports industry.

References

[1] Al-Asadi, M.A. and Tasdemır, S., 2022. Predict the value of football players using FIFA video game data and machine learning techniques. *IEEE Access*, **10**, pp.22631-22645.

[2] Almulla, J. and Alam, T., 2020. Machine learning models reveal key performance metrics of football players to win matches in Qatar Stars League. *IEEE Access*, **8**, pp.213695-213705.

[3] Ati, A., Bouchet, P. and Jeddou, R.B., 2024. Using multi-criteria decision-making

and machine learning for football player selection and performance prediction: A systematic review. *Data Science and Management*, **7**(2), pp.79-88.

[4] Bongiovanni, T., Trecroci, A., Cavaggioni, L., Rossi, A., Perri, E., Pasta, G., Iaia, F.M. and Alberti, G., 2021. Importance of anthropometric features to predict physical performance in elite youth soccer: A machine learning approach. *Research in Sports Medicine*, **29**(3), pp.213-224.

[5] Chandra, B., 2024. Prediction of Football Player Performance Using Machine Learning Algorithm.

[6] Chang, V., Sajeev, S., Xu, Q.A., Tan, M. and Wang, H., 2024. Football Analytics: Assessing the Correlation between Workload, Injury and Performance of Football Players in the English Premier League. *Applied Sciences*, **14**(16), p.7217.

[7] Ćwiklinski, B., Giełczyk, A. and Choraś, M., 2021. Who will score? A machine learning approach to supporting football team building and transfers. *Entropy*, **23**(1), p.90.

[8] Daniel, C., 2021. Developing Predictive Model for Football Match Result in Ethiopian Premier League Using Machine Learning Algorithm. *Doctoral dissertation.*

[9] Dijkhuis, T.B., Kempe, M. and Lemmink, K.A., 2021. Early prediction of physical performance in elite soccer matches—a machine learning approach to support substitutions. *Entropy*, **23**(8), p.952.

[10] Duarte, R., 2022. Utilizing machine learning techniques in football prediction.

[11] Ghar, S., Patil, S. and Arunachalam, V., 2021. Data Driven football scouting assistance with simulated player performance extrapolation. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 1160-1167). IEEE.

[12] Jamil, M., Phatak, A., Mehta, S., Beato, M., Memmert, D. and Connor, M., 2021. Using multiple machine learning algorithms to classify elite and sub-elite goalkeepers in professional men's football. *Scientific Reports*, **11**(1), p.22703.

[13] Kelly, A.L., Williams, C.A., Cook, R., Sáiz, S.L.J. and Wilson, M.R., 2022. A multidisciplinary investigation into the talent development processes at an English football academy: A machine learning approach. *Sports*, **10**(10), p.159.

[14] Khan, M.A., Habib, M., Saqib, S., Alyas, T., Khan, K.M., Al Ghamdi, M.A. and Almotiri, S.H., 2021. Analysis of the Smart Player's Impact on the Success of a Team Empowered with Machine Learning. *Computers, Materials & Continua*, **66**(1).

[15] Mandorino, M., Clubb, J. and Lacome, M., 2024. Predicting Soccer Players' Fitness Status Through a Machine-Learning Approach. *International Journal of Sports Physiology and Performance*, **1**(aop), pp.1-11.

[16] Markopoulou, C., Papageorgiou, G. and Tjortjis, C., 2024. Diverse Machine Learning for Forecasting Goal-Scoring Likelihood in Elite Football Leagues. *Machine Learning and Knowledge Extraction*, **6**(3), pp.1762-1781.

[17] McHale, I.G. and Holmes, B., 2023. Estimating transfer fees of professional footballers using advanced performance metrics and machine learning. *European Journal of Operational Research*, **306**(1), pp.389-399.

[18] Musa, R.M., Majeed, A.P.A., Kosni, N.A. and Abdullah, M.R., 2020. Machine learning in team sports: Performance analysis and talent identification in Beach Soccer & Sepak-takraw. *Springer Nature.*

[19] Nassis, G., Verhagen, E., Brito, J., Figueiredo, P. and Krustrup, P., 2023. A review of machine learning applications in soccer with an emphasis on injury risk. *Biology of Sport*, **40**(1), pp.233-239.

[20] Prys, M., Rosiński, Ł., Buryta, R., Radzimiński, Ł., Różewski, P. and Rejer, I., 2023. Integrating Machine Learning for Football Injury Prediction: A Concept for an Intelligent System. *Procedia Computer Science*, **225**, pp.4139-4147.

[21] Rico-González, M., Pino-Ortega, J., Méndez, A., Clemente, F. and Baca, A., 2023. Machine learning application in soccer: A systematic review. *Biology of Sport*, **40**(1), pp.249-263.

[22] Rommers, N., Rössler, R., Verhagen, E., Vandecasteele, F., Verstockt, S., Vaeyens, R., Lenoir, M., D'Hondt, E. and Witvrouw, E., 2020. A machine learning approach to assess injury risk in elite youth football players. *Medicine and Science in Sports and Exercise*, **52**(8), pp.1745-1751.

[23] Sulimov, D., 2024. Performance Insights-based AI-driven Football Transfer Fee Prediction. *arXiv preprint arXiv:2401.16795.*

[24] Van Eetvelde, H., Mendonça, L.D., Ley, C., Seil, R. and Tischer, T., 2021. Machine learning methods in sport injury prediction and prevention: A systematic review. *Journal of Experimental Orthopaedics*, **8**, pp.1-15.

[25] Wisdom, C. and Javed, A., 2023. Machine Learning for Data Analytics in Football: Quantifying Performance and Enhancing Strategic Decision-Making. *Available at SSRN 4558733.*