Forecasting Player Availability in Football: A Comprehensive Model

Abstract

This study presents a comprehensive model for forecasting player availability in football by predicting significant injuries. Utilizing historical performance, injury records, and FIFA attributes, the model leverages machine learning techniques to identify key predictors of player injuries. The research underscores the importance of injury history in forecasting future availability, providing actionable insights for team management and medical staff.

Introduction

Injuries significantly impact the performance and success of football teams. Understanding and predicting player availability is crucial for optimal team management. This study aims to develop a predictive model using historical data to identify players at risk of significant injuries, thereby aiding in preventive measures and strategic planning.

Data Preprocessing and Integration

Data Import and Initial Setup

- **Libraries**: Utilized `pandas`, `numpy`, `re`, and `datetime` for data manipulation and processing.
- **Data Sources**: Imported CSV files containing player data, player statistics, and match information.

Data Filtering and Preparation

- **Injury Threshold**: Defined major injuries as those leading to absences of 120 days or more.
- **Seasonal Filter**: Limited analysis to data from the 2016/17 to 2020/21 seasons.

Data Cleaning

- **Erroneous Data Removal**: Eliminated rows with invalid match dates.
- **Date Conversion**: Transformed match dates into `datetime` objects and extracted the start year for analysis.

Handling Missing and Non-Numeric Data

- **Substitution Data**: Cleaned and converted substitution times to numeric values using a custom function.

Data Merging

- **Player and Match Data**: Merged player statistics with player details using a consistent player ID (`p_id2`) and integrated match data for comprehensive analysis.

Calculating Player Minutes Played

- **Minutes Calculation**: Implemented a function to estimate minutes played per match, considering starting status and substitution times.
- **Participation Indicator**: Created a binary indicator for game participation based on minutes played.

Selection of Relevant Columns

- **Data Reduction**: Focused on essential columns: player ID, position, match ID, match date, start year, minutes played, and game participation.

Injury and FIFA Data Integration

Injury Data Preprocessing

- **Injury Data**: Standardized player names and filtered out non-injury-related entries to focus on physical injuries.
- **Duration Extraction**: Developed a function to extract and convert injury durations to numeric values.

Injury Data Aggregation

- **Seasonal Aggregation**: Grouped data to calculate total days injured per season and across the dataset's timespan.

FIFA Data Preprocessing

- **FIFA Data**: Imported FIFA player data and cleaned player names extracted from URLs.
- **Relevant Attributes**: Selected columns like height, weight, nationality, work rate, pace, physic, and overall ratings.

FIFA Data Aggregation

- **Data Grouping**: Aggregated FIFA data to compute mean values for numeric attributes and mode for categorical ones.

Combining Datasets

- **Dataset Merging**: Merged injury, player performance, and FIFA data, resulting in a comprehensive dataset with 604 unique players.

Feature Engineering and Analysis

Index Resetting

- **Index Management**: Reset the dataframe index for continuous numerical indexing.

Age Calculation

- **Age Feature**: Calculated players' age at the beginning of each season.

Cumulative Metrics Calculation

- **Performance Metrics**: Computed cumulative minutes and games played, average days injured, and games per season from historical data.

Derived Metrics

- **Minutes Per Game**: Derived from cumulative minutes divided by cumulative games.
- **BMI**: Calculated using height and weight data.
- **Work Rate Numeric Conversion**: Transformed categorical work rate values into numeric scores.

Position Numeric Conversion

- **Positional Mapping**: Converted player positions to numeric values for analysis.

Significant Injury Feature

- **Binary Indicator**: Created a feature indicating significant injuries in the previous season.

Handling Missing Values and Outliers

- **Data Integrity**: Replaced infinite values with `NaN` and ensured data completeness.

Data Export

- **File Export**: Saved the processed dataframe for further analysis and modeling.

Analysis of Features

Physical and Season Features

- **Feature Separation**: Distinctly analyzed physical attributes and seasonal performance metrics.

Exploratory Data Analysis (EDA)

- **Visual Analysis**: Generated histograms and scatter plots to explore feature distributions and relationships.
- **Correlation Examination**: Investigated correlations between features and injury occurrences.

Position-wise Analysis

- **Positional Impact**: Analyzed median days injured by position, highlighting susceptibility variations.

Categorical Target Variable

- **Threshold Analysis**: Assessed injury thresholds and defined a categorical target for major injuries.
- **Visual Relationships**: Used box plots to explore feature relationships with the target variable.

Modeling Process and Evaluation

Handling Missing Values

- **Null Dropping**: Removed rows with null values in critical columns.
- **Imputation**: Applied mean and mode imputation for less critical columns.

Dataset Preparation

- **Binary and Numerical Targets**: Created separate datasets for binary and numerical target variables.

Train-Test Split and Oversampling

- **Data Split**: Divided data into training and testing sets.
- **SMOTE Oversampling**: Addressed class imbalance in the training set.

Feature Scaling

- **Standardization**: Scaled continuous features for uniformity.

Feature Selection

- **Recursive Feature Elimination**: Used RFECV to select the most predictive features.

Model Selection and Training

- **Logistic Regression**: Initially used Logistic Regression, followed by XGBoost for improved performance.

Hyperparameter Tuning and Cross-Validation

- **Grid Search**: Optimized XGBoost hyperparameters using cross-validation.

Model Evaluation

- **Performance Metrics**: Assessed model performance using accuracy, precision, recall, F1 score, and ROC AUC score.
- **Confusion Matrix**: Visualized classification results.
- **ROC and Lift Curves**: Evaluated model discrimination capability and ranking effectiveness.

Conclusion

This study highlights the predictive strength of historical injury data in forecasting player availability. The model, while foundational, offers significant insights for enhancing team management strategies and player welfare, with potential for future enhancements through more granular data and advanced modeling techniques.

Soccer Player Injury Prediction Model: A Data-Driven Approach Using XGBoost

Abstract

This paper presents a machine learning model designed to predict the likelihood of a soccer player experiencing a major injury in an upcoming season. By integrating player attributes, injury history, and game time data, the model aims to provide insights into player availability, helping team managers and coaches make informed decisions. The model employs XGBoost, a powerful gradient boosting algorithm, to classify players based on their risk of injury, achieving an accuracy of 73.3% and an AUC-ROC score of 66.65%.

1. Introduction

Predicting injuries in sports, especially soccer, is a critical task that can help optimize team performance and player management. Injuries significantly affect player performance, and understanding the factors leading to them can assist in making data-driven decisions regarding player selection. This study focuses on predicting major injuries in professional soccer players using historical data and machine learning techniques.

2. Model Overview

2.1 Model Purpose

The Soccer Player Injury Prediction Model aims to predict the likelihood of a soccer player experiencing a significant injury (defined as injuries lasting more than 120 calendar days) in the next season. The model utilizes player attributes, past injury history, and game time data to make predictions.

2.2 Model Architecture

- **Model Type**: Binary Classification
- **Algorithm Used**: XGBoost (Extreme Gradient Boosting)
- **Model Code**: Python (Implemented using libraries such as pandas, numpy, and scikit-learn)
- **Model Version**: 1.0

2.3 Intended Use

The model is designed for use by team managers, coaches, and sports analysts to identify players at high risk of injury and manage player selection for upcoming matches.

3. Data Sources

The model uses data from the following sources:

- **Player Attributes**: FIFA 16-21 dataset.
- **Injury History**: Data from Transfermarkt, scraped using the worldfootballR R package.
- **Game Time Data**: Barclays Premier League Soccer Dataset.

The dataset covers players who participated in the British Premier League from the 2016/17 to 2020/21 seasons. The final dataset consists of 685 rows and 317 unique players.

4. Data Preprocessing and Integration

4.1 Data Collection and Merging

Data from the three sources were cleaned and merged to create a comprehensive dataset. Key features include:

- **Player attributes**: height, weight, pace, work rate, position.
- **Injury history**: number of days injured in previous seasons, significant injuries.
- **Game participation**: total minutes played, number of games played, etc.

4.2 Feature Engineering

Features were engineered to represent:

- **Cumulative game time and injury history**: The number of minutes played and the days injured in past seasons.
- **Physical attributes**: Age, height, weight, BMI, and work rate.
- **Positional features**: The player's position and its impact on injury risk.

4.3 Handling Missing Data

Missing values were handled through imputation and removal of rows with critical missing data.

5. Model Training

5.1 Dataset Split and Oversampling

- The data was split into training (70%) and test (30%) sets.
- Due to class imbalance (only ~25% of data had positive injury outcomes), the training data was oversampled using the Synthetic Minority Oversampling Technique (SMOTE).

5.2 Hyperparameter Tuning

Optimal hyperparameters for the XGBoost model were determined via a grid search, including:

- **Number of trees**: 200
- **Maximum tree depth**: 4
- **Learning rate**: 0.1
- **Subsampling**: 80% of columns and 90% of rows

6. Model Evaluation

6.1 Performance Metrics

The model was evaluated using the following metrics:

- **Accuracy**: 73.30%
- **Precision**: 26.83%
- **Recall**: 30.56%
- **F1 Score**: 28.57%
- **AUC-ROC**: 66.65%

These metrics demonstrate that the model performs reasonably well, although there is room for improvement in precision and recall.

6.2 Feature Importance

Key features contributing to injury prediction include:

- **Previous season's injuries**
- **Cumulative minutes played**
- **Player's physical attributes (e.g., pace and work rate)**

7. Ethical Considerations and Limitations

7.1 Ethical Use

The model is intended to help manage player injuries but should not be used to discriminate based on race, nationality, or other personal characteristics.

7.2 Limitations

- **Data Availability**: The model is based on a relatively small dataset of players in the British Premier League.
- **Gender Bias**: The model is trained only on male players, and the injury patterns for women may differ.
- **Generalizability**: The model is specific to the British Premier League and may not apply universally across all leagues.

8. Conclusion

The Soccer Player Injury Prediction Model provides a valuable tool for predicting player injuries, leveraging historical data and machine learning algorithms. While the model shows promising results, future work will aim to improve its performance and extend its applicability to other soccer leagues and gender groups.

9. Future Work

Future iterations of the model will include:

- Expanding the dataset to include players from other leagues and seasons.
- Exploring different machine learning algorithms to improve prediction accuracy.
- Incorporating more granular data, such as player fatigue levels, medical reports, and training intensities.