IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/46AI/2025/4637725/
Vol. 2025 (34), Article ID 4637725

Predicting Soccer Match Outcomes with Machine Learning Methods*

Krzysztof TREPKA and Joanna WIŚNIEWSKA

Institute of Information Systems, Faculty of Cybernetics,
Military University of Technology, Gen. S. Kaliskiego 2, 00-908 Warsaw, Poland
, Joanna.Wisniewska@wat.edu.pl

Correspondence should be addressed to: Krzysztof TREPKA, krzysztof.trepka@gmx.com

* Presented at the 46th IBIMA International Conference, 26-27 November 2025, Ronda, Spain

Abstract

This study explores the application of machine learning methods to predict football (soccer) match outcomes. Football, as a highly dynamic and data-rich sport, provides a valuable source of information for predictive modeling. The research focuses on evaluating and comparing the performance of several machine learning algorithms: a naïve baseline model, logistic regression, random forest, XGBoost, and an artificial neural network. The dataset used for training and testing consists of historical match statistics, team performance indicators, and situational variables such as home advantage. Feature engineering and data preprocessing steps, including normalization and handling of missing data, were applied to improve model performance and generalizability. Each model was assessed using standard evaluation metrics such as accuracy, precision, recall, and F1-score. The results indicate that while simple models like logistic regression provide solid baseline performance, ensemble methods such as random forest and XGBoost achieve higher predictive accuracy. The artificial neural network, although more computationally demanding, shows promising results in capturing complex, nonlinear relationships between match variables. The study highlights the challenges of modeling football outcomes, such as the inherent randomness of the sport and the influence of non-quantifiable factors, but demonstrates the potential of machine learning in enhancing predictive analytics in sports.

Keywords: sports analytics, XGBoost, random forests, logistic regression, artificial neural networks

Introduction

Predicting the outcome of sporting events, including football matches, presents a compelling challenge that finds application in many practical areas, primarily in the work of bookmakers, who earn money by determining the probability of a given outcome. Due to the complexity and nonlinear nature of data describing the course of sporting events, traditional statistical methods often prove insufficient. Consequently, machine learning-based approaches are gaining importance, enabling the construction of predictive models based on large and diverse datasets.

The aim of this work is to develop and compare selected machine learning models for predicting the outcome of a football match, understood as a multi-class classification: home win, draw, or away win. The work covers the entire process of designing a predictive system—from data analysis and preparation, through feature selection and algorithm selection, to implementation, validation, and evaluation of the selected models' effectiveness.

The application of machine learning-based approaches to sports analytics is justified from both a research and practical perspective. The developed models can be used, for example, to support decision-making in the

Cite this Article as: Krzysztof TREPKA and Joanna WIŚNIEWSKA, Vol. 2025 (34) "Predicting Soccer Match Outcomes with Machine Learning Methods " Communications of International Proceedings, Vol. 2025 (34), Article ID 4637725, https://doi.org/10.5171/2025.4637725

bookmaking industry or team analytics. The work also aims to identify the limitations and challenges associated with modeling complex phenomena using artificial intelligence techniques.

The paper first characterizes football as a source of data that can be used to train machine learning models, and then describes the results of the training experiments conducted for selected models.

Football as a Source of Data

As a global sporting phenomenon, football attracts the attention not only of fans and journalists but also of data analysts and forecasting researchers. Every football match generates a wealth of data regarding both the final score and the course of the match, such as ball possession statistics, shots, interceptions, fouls, and tactical changes. The complexity of the sport, resulting from player interactions, the dynamics of on-field events, and the random nature of certain events (e.g., injuries, refereeing decisions, weather conditions), makes football a challenging yet fascinating field for predictive analysis.

Unlike sports, which are characterized by greater predictability and a more established structure (e.g., tennis or chess), football is characterized by high variability and a low frequency of key events, such as goals. This specificity means that classical statistical methods are sometimes insufficient to accurately predict match outcomes. In this context, artificial intelligence and machine learning-based methods are gaining in importance, allowing for modeling the complex, often nonlinear relationships present in match data. The use of this data in predictive models is also becoming increasingly common. These models are designed to predict the outcome of a match based on a set of input variables – such as past performance, team form, number of injuries, match location, or ranking differences. Predicting outcomes can take the form of classification (e.g., predicting one of three classes: home win, draw, away win) or regression (e.g., the exact number of goals scored by each team).

Soccer data is characterized by a wide variety. Its structure depends on the purpose of the analysis and its source. When designing predictive models for predicting match outcomes, it is particularly important to properly understand the types of data available, what features can be extracted from them, and how they can be processed within a training set. Soccer data can be divided into several basic categories.

Team statistics: data describing the behaviour of teams during the match – e.g. ball possession, number of shots, passing accuracy, number of corner kicks, number of fouls, number of cards.

Player Statistics: detailed statistics for individual players, including mileage, passing accuracy, offensive participation, expected goals (xG), expected assists (xA) by Powers (2011), and physical metrics like sprints. This data is more difficult to obtain but can significantly improve the quality of predictions.

Team historical data and form: this includes recent match results, goal difference, points earned in recent matches, win/loss streaks, average shots and goals. These characteristics often act as indicators of "form" and are used as input variables in predictive models.

Contextual and external data: factors that influence the outcome but are not directly related to the game itself – e.g., weather conditions, surface type, team travel distance, number of rest days, player absences (injuries, suspensions), and predicted lineups. Although difficult to standardize, these factors can have a significant impact on predicting the outcome.

In the case of match outcome prediction, this data is most often transformed into a tabular form, where each row represents one match and the columns correspond to individual features.

It's also worth noting that football is a relational game, at least when analyzing league matches—each team plays against another, which causes the data to be bilateral. Therefore, relative characteristics (e.g., points difference, form difference) are often used, which better reflect the relationships between teams than raw values.

Machine Learning in Sports Analysis

Machine learning in football match analysis can be divided into several main application areas. A significant area is the analysis of player performance, based on event data and tracking data. To this end, models are built that allow for the assessment of, among other things:

- player's contribution to offensive and defensive actions,
- probability of scoring a goal after a given shot (xG),
- added value of passes (e.g. expected assists – xA),
- the player's influence on space control.

Machine learning also enables the grouping of players with similar playing profiles (e.g., offensive wingers, defensive midfielders) and the identification of unusual on-field behavior. Algorithms used in this area include multiclass classifiers, long-line structured time-of-flight (LSTM) recurrent networks, and unsupervised methods such as k-means and DBSCAN..

Predictive and classification models also support processes related to squad selection, rotation planning, and injury risk assessment. For example, classification algorithms can warn of player overload based on match and training loads, allowing for better squad management. Machine learning is also used in scouting, for example, to identify players with specific physical and tactical characteristics, using large sets of statistical data from multiple leagues and seasons.

Another area of application for machine learning methods is predicting the outcome of football matches. This can be used by a team's coaching staff, for example, to determine which matches don't require the use of their best players or to predict the outcome of other matches that may be important in determining league standings. Such machine learning models are also used by bookmakers, where the accuracy of their predictions impacts their financial profits. Examples of input data used to create such models include:

- Match statistics (average number of goals, number of shots, ball possession),
- Team form in recent rounds,
- Match location,
- Differences in team strength (rankings, league table),
- Information on injuries and lineups,
- Statistics of players selected for the match.

In summary, machine learning (ML) is a crucial component of modern sports analysis, including football match analysis. Its effective application, however, requires not only access to high-quality data but also appropriate processing, feature selection, and result validation. Despite numerous limitations, such as the randomness of on-field events, incomplete data, and limited observations, appropriately selected ML models can significantly aid in predicting football match outcomes and in the decisions made by football club employees.

Selection of Data

Before building a predictive model, a crucial step is analyzing available datasets to decide on the most appropriate one and which features to consider. The quality and characteristics of input data directly impact the effectiveness of machine learning models, so selecting the appropriate data source and features is the foundation of the entire machine learning process. All datasets analyzed in this work were sourced from the website kaggle.com.

The first dataset analyzed was the "Football Data from Transfermarkt" dataset (see: www.kaggle.com/datasets/davidcariboo/player-scores). The data in the dataset comes from the Transfermarkt website (see: www.transfermarkt.pl). This dataset contains CSV files, in which rows from one file can be linked to other files. The structure resembles a relational database. This dataset is updated weekly and contains all available leagues and matches from Transfermarkt. In summary, this dataset contains some useful information, such as the number of goals scored in individual matches by teams, as well as their positions in the table at kickoff. This dataset lacks statistics currently very popular for assessing team strength, such as xG and xA indicators for both teams and individual players. Another missing statistic that could be useful is shots, especially shots on target. Ball possession for both teams could also be a desirable feature, but such data is lacking.

The next dataset considered is the "Football Database" (see: www.kaggle.com/datasets/technika148/football-database). It contains matches from the top five European leagues: the English, German, Italian, Spanish, and French leagues. The matches are from the 2015/2016 to 2020/2021 seasons. This is significantly fewer matches than in the previously analyzed dataset. However, it does contain many additional player and team statistics, including xG (for teams and players) and xA (for players), shots, including shots on target, and several other statistics that will be described in more detail in the detailed file description. According to the dataset description, the data comes from the Understat (see: understat.com) and Football-data (see: www.football-data.co.uk) websites. The files included in the dataset are as follows:

- appearances.csv – player performances in matches. It includes columns such as goals, own goals, shots, xG, xGoalsChain, xGoalsBuildup by Lawrence (2018), assists, key passes, xA, yellow and red cards. All of these columns can be useful in building predictive models.
- games.csv – matches. The file contains data such as goals scored by the home team and the away team. It also includes information about which team is the home team.
- leagues.csv – football leagues. Does not contain columns useful for creating predictive models.
- players.csv – player data. This file does not contain any data relevant to modeling.
- shots.csv – shot data from matches. The file contains information about who shot, who passed the ball to the shooter, the match it took place in, the result of the shot, and the type of shot. A wealth of data that, if used appropriately, can be used for modeling.
- teams.csv – team. The file does not contain data useful for creating machine learning models.
- teamstats.csv – team statistics for the match. The file contains columns that can be used for modeling, such as team xG, team shots, team shots on target, and number of passes into the opponent's penalty area.

In summary, this dataset offers significantly more qualitative data that can be used to create predictive models. Even using team statistics alone, excluding player statistics, it seems possible to generate satisfactory ML models. Furthermore, with appropriate data transformation and the inclusion of player data, even better results can likely be achieved.

After comparing the "Football Data from Transfermarkt" and "Football Database" collections, the latter was chosen due to the greater amount of data that influences the outcome of a football match.

Selecting the appropriate columns, or explanatory variables (features), is one of the key stages of data preparation for predictive modeling in machine learning. It is based on these features that the model learns to recognize patterns and make decisions, so their proper selection significantly impacts the effectiveness and interpretability of the final results. The goal of this stage is not merely to mechanically transfer data to the model, but to consciously select information that has explanatory potential in the context of predicting the outcome of a football match. Based on this, the input features were selected. All features, except for the days since the last match, are presented in two versions. One is the average for the entire season, and the other is the average for the last five matches. According to this description, the following features were selected:

- Average points per match scored by the home team,
- Average points per match scored by the away team,
- Average goal difference per match for the home team,
- Average goal difference per match for the away team,
- Average points per match scored by the home team during home matches,
- Average points per match scored by the away team during away matches,
- Average goal difference per match for the home team during home matches,
- Average goal difference per match for the away team during away matches,
- Number of days since the last home team match,
- Number of days since the last away team match,
- Average shots per match for the home team,
- Average shots per match for the away team,
- Average shots on target per match for the home team,
- Average shots on target per match for the away team,
- Average passes into the opponent's penalty area per match for the home team,
- Average passes into the opponent's penalty area per match for the away team,
- Average xG per match for the home team,
- Average xG for the away team's match.

The column selection reflects both the specific nature of football data and the goals of the machine learning models in this work. The features considered include both quantitative variables describing the course of previous matches (e.g., points scored per match, shots per match) and categorical variables with contextual significance (rest period since the last match). The selection process was guided by their availability, completeness, stability over time, and, above all, potential predictive value.

Columns that were redundant, highly correlated, or contained information difficult to obtain in real-world application were eliminated. This approach aimed not only to improve the algorithms' efficiency but also to increase their usability and robustness to input data errors.

The set of selected variables creates a coherent and possibly compact set of features, which forms the basis for training the classification models described later in the paper.

Implementation and Analysis of Selected Model Results

This section presents the practical implementation of four selected prediction algorithms: multinomial logistic regression (see: Agresti (2013)), random decision forest (see: Breiman (2001), Witten et al. (2011)), XGBoost (see: Chen and Guestrin (2016)), and artificial neural network (see: Goodfellow et al. (2016)). Implementation details will be discussed, including data division into training and test sets, selection of evaluation metrics, and hyperparameter optimization strategies for each model. After training, the results will be systematically analyzed, including comparison of metrics such as accuracy, precision, recall, and F1 score, as well as evaluation of forecast quality against a benchmark model. Additionally, conclusions from the analysis of feature significance will be presented and the strengths and weaknesses of individual algorithms in the context of predicting the results of football matches will be discussed.

Naive Baseline

To evaluate the effectiveness of selected machine learning models, it is essential to benchmark their results against an appropriate reference point. A commonly used approach is the so-called naive baseline, which is a simplified classifier based on the most frequently occurring class or a simple, intuitive assumption. Although it does not consider any input data or feature dependencies, it serves an important comparative function—allowing us to assess whether more advanced models actually add value over trivial prediction strategies.

In the context of predicting soccer match outcomes, the most commonly used naive model is the home-win classifier. This strategy is based on the historical observation that home-field advantage often increases the home team's chances of victory. This model uses no statistics or additional information, but assigns the same class label to each match. Using Python and the Pandas library, we calculated the proportion of matches won by the home team, which is the model's accuracy.

As a result of this testing, the model achieved an accuracy of 44.73%. For comparison, drawn matches account for 25.13% of the total, while away wins account for 30.14%. This means that, considering the test set, home teams are most likely to win. This result will serve as a reference for all models presented below.

Logistic Regression

Despite its simplicity, logistic regression remains one of the most commonly used classification algorithms in machine learning. Its efficiency, interpretability, and low computational requirements make it a starting point for many classification tasks, including football match outcome prediction. In this work, we use its polynomial version, which allows for classification into more than two classes – in this case: home win, draw, away win.

This model is based on the logistic (sigmoid) function, which estimates the probability of an observation belonging to a given class based on a linear combination of input variables. Multiclass classification employs a generalization of this function, the softmax function, which returns the probability distribution between classes.

First, input data from training and test files are taken, as well as output data from both training and test. Next, the data is standardized, which is necessary for logistic regression; otherwise, data with larger absolute values on average would have a greater impact on the results. For this purpose, the "StandardScaler" class object from the

"scikit-learn" library is used, which standardizes the data so that it follows a normal distribution with a mean of 0 and a standard deviation of 1. After standardizing the input data from the test and training sets, a logistic regression model is created. The model was created using the LogisticRegression class built into the "scikit-learn" library with parameters:

- multiclass = ”multi_nominal” – which indicates the use of more than 2 classes,
- max_iter = 1000 – the maximum number of iterations when training the model will be 1000.

The object thus created uses the softmax function to create a probability distribution.

The next step was to test the model parameters. Analysis of the intercepts assigned to each class suggests that the model shows a slight bias toward favoring the "home win" class as the most probable outcome, in the absence of other attributes. A positive offset for this class (0.3237) indicates that – with all input variables set to zero – the model tends to predict a home win. Conversely, negative values for a draw (–0.1743) and an away win (–0.1495) indicate a lower "baseline" probability of these outcomes. In practice, this reflects the well-known phenomenon of home-field advantage, often observed in historical data, as well as the fact that draws are less frequent than away wins, albeit by a small margin.

The model performed relatively well in classifying home wins. The high sensitivity value (0.82) indicates that the vast majority of actual cases of this class were correctly identified. Precision was also satisfactory (0.53), resulting in the highest F1 score of the three classes (0.64). This model behavior may be due to the home team's advantage in historical match data, which is reflected in both the class distribution and the model's dominant predictive patterns.

For the "away team wins" class, the model achieved moderate performance – both precision and sensitivity were 0.49, indicating a symmetrical but average level of detection and prediction accuracy for this class. The F1-score also remained at 0.49, confirming there was no clear advantage or underestimation in this case.

By far the worst performance was achieved for the "tie" class. The model was virtually unable to correctly classify any instances belonging to this category, as reflected by a sensitivity value of 0.00 and an F1-score close to zero (0.01). Although precision reached 0.33, the low sensitivity completely undermines the model's usefulness in recognizing ties. This result may be due to an unbalanced dataset (fewer ties compared to other classes) as well as the model's limited ability to capture patterns characterizing tie situations.

Random Forest

The random forest model is one of the most commonly used approaches in classification tasks, characterized by high resistance to overfitting and good generalization ability. It is an extension of the decision tree method, in which multiple independent trees are trained in parallel on randomly selected subsets of data and features. The model's final prediction is based on majority voting (for classification), which significantly reduces variance and increases forecast stability.

In the context of predicting football match outcomes, the random decision forest model is useful for its ability to capture nonlinear relationships between numerous input features, such as team statistics or recent match form. This model does not require prior data scaling and handles features of varying nature and distribution well. An additional advantage is the ability to estimate feature importance, which provides valuable interpretive information when analyzing the impact of individual variables on match outcomes.

First, input and output data are collected from the training and test sets. Data standardization is not necessary at this time, as this type of model does not require it.

The "scikit-learn" library was used to train the model. The class used to create the random decision forest model is "RandomForestClassifier," which can generate a random decision forest based on training data and hyperparameter values.

To find the best hyperparameter values, a grid search with cross-validation method was used. This method creates models with different hyperparameter values and compares their accuracy scores. The model hyperparameters

with the highest accuracy are considered the best for training the model, and the resulting model will be used for further analysis. The selection of value ranges for individual hyperparameters was based on literature recommendations and experience described in the "scikit-learn" library documentation. The implementation of the grid search method itself was based on the "GridSearchCV" class from the "scikit-learn" library.

The "n_estimators" parameter, which specifies the number of trees in the random forest, was set to a range of 100-300. This number is sufficient to obtain stable and repeatable classification results while maintaining a reasonable training time. These values are consistent with typical recommendations for medium-sized datasets, such as football match datasets. The "max_depth" hyperparameter, which limits the depth of each tree, was set to 10, 20, and no limit (None). Deeper trees allow the model to capture more complex relationships but can lead to overfitting. Limiting the depth, on the other hand, controls model complexity and improves generalization. The "min_samples_split" and "min_samples_leaf" parameters, which control the minimum number of samples required to split a node and to create a leaf, were set to [2, 5] and [1, 2], respectively. These values were chosen to test the impact of more conservative splits on reducing overfitting, which can be important when analyzing data with a large number of variables. Additionally, we tested different settings for the "max_features" parameter, which defines the number of features considered for each split. The values 'sqrt' and 'log2' were included, in line with literature recommendations for classification, which suggest using feature subsets to introduce more randomness and improve model generalization. The "class_weight" parameter was also included, with values 'None' and 'balanced', because match data can be unbalanced due to the higher prevalence of certain classes, such as home team wins (see: Géron (2019), Probst and Boulesteix (2018)).

A 5-fold cross-validation method was used to evaluate the hyperparameter sets. This means that the training set is divided into five parts, and then in each iteration, one of these parts is selected as the test set, while the remaining four parts are treated as the test set. In each iteration, a different part of the original set is considered as the test set. The average accuracy is then calculated, based on which the best hyperparameter set is selected.

Using the hyperparameter grid search method, the following best hyperparameter values were found:

- Class weights: none. All classes are treated equally;
- Maximum tree depth: 10;
- Maximum number of features considered: 'sqrt' – the square of the number of input features. Since there are 34 input features, the maximum number of features considered is 5;
- Minimum number of samples required to create a leaf: 1;
- Minimum number of samples required to split a node: 5;
- Number of decision trees in the model: 300.

The best accuracy result obtained when training the model using grid search and cross-validation was 51.85%. The model with the best result was selected for prediction.

The model performs relatively well in predicting home team wins. The high sensitivity value (0.81) in this class indicates that the model correctly identified most actual home team wins. At the same time, the F1 score of 0.64 indicates a good compromise between precision and sensitivity in this class, making the model particularly effective in this match scenario.

In the case of away team wins, the model demonstrates average prediction quality. The achieved precision and sensitivity values (0.50 and 0.49, respectively) and an F1 score of 0.49 suggest that the model is slightly less effective at identifying this class than for home teams, but still retains its basic utility. Therefore, it can be concluded that the model is capable of detecting patterns associated with away team wins, although not as effectively as for home teams.

The model exhibits the greatest difficulty predicting draws. A particularly low sensitivity value (0.01) and a negligible F1-score (0.02) indicate its near-complete inability to recognize this class. This means that even though the data includes matches that end in draws, the model almost always assigns them a different class – most likely a victory for one of the teams. The low precision value (0.23) further confirms that even the few cases classified as draws are inaccurate.

XGBoost

This section presents the implementation of the XGBoost (Extreme Gradient Boosting) model, which is one of the most advanced ensemble learning methods based on gradient boosting of decision trees. Due to its high efficiency, resistance to overfitting, and ability to handle data with complex structures, XGBoost is widely used in classification tasks, including the analysis and prediction of football match results.

First, input and output data for the test and training sets are read from files. The XGBClassifier class from the xgboost library for Python (see: Chen and Guestrin (2016)) is used to train the model.

As with the previous model, a grid search with cross-validation of the model's hyperparameters was used to find the optimal set. The validation type chosen was 5-fold cross-validation. Using the grid search method, the following best hyperparameter values were found for the XGBoost model:

- Observation sampling rate: 0.5;
- Gamma parameter: 5;
- Learning rate: 0.3;
- Maximum tree depth: 10;
- Number of trees: 100;
- Proportion of features selected for building each tree: 0.5.

The selected model achieved an accuracy of 50.75%, which is significantly lower than the best result obtained by searching the hyperparameter grid during cross-validation. All features were used to train the model.

The model achieved the highest performance in identifying home team wins, as reflected in both a high sensitivity value (0.80) and a relatively high precision value (0.52). This means that the model is able to accurately detect the majority of actual home team wins, and at the same time, most of its positive predictions for this class are correct. This performance characteristic may indicate that the model naturally favors home team wins – which may have statistical justification, as home team wins constitute a significant percentage of all outcomes in many football leagues.

In the case of ties, the model's performance is clearly unsatisfactory. The particularly low sensitivity value (0.01) indicates that the model practically fails to identify this class correctly—almost all actual ties are incorrectly assigned to other categories. The low precision value (0.30) further suggests that even the few cases in which the model indicates a tie are unreliable. This result may be due to both the relatively smaller number of ties in the training data and the difficulty in distinguishing this class based on the available features.

For the "away team wins" class, the model achieved balanced precision (0.48) and sensitivity (0.49), which translates into a moderate F1-score (0.48). While not exceptional, this result indicates the model's relatively stable, albeit limited, ability to recognize this class.

Artificial Neural Network

To test how well the neural network handles the classification task (see: Masters and Luschi (2018)), data was first loaded from previously prepared files. Then, the input data was standardized in the same way that the data was standardized when creating the logistic regression model.

To implement the model, we used the MLPClassifier class from the "scikit-learn" library, which allows for easy model creation. The most important thing is to select the appropriate hyperparameter values, while the training process itself is automatic by running the "fit" function.

To fine-tune the artificial neural network model, a grid search strategy was employed, allowing for systematic testing of various combinations of hyperparameter values. The hyperparameter grid search used was a 5-fold cross-validation version. The choice of tested values was based on a literature review and recommendations for the use of multilayer perceptrons (MLPs) in classification tasks.

Particular attention was paid to the network architecture, which consists of the number of network layers and the number of neurons in each layer. Networks with one, two, and three hidden layers, with different numbers of neurons, were considered. Configurations included [64, 128] neurons in a single layer, sets [(64, 32), (128, 64),

(256, 128)] as two-layer networks, and the set (256, 128, 64) as a three-layer variant. This choice was made to investigate the impact of architecture depth and width on the performance of soccer match outcome classification.

The activation function was limited to "rel," considered standard in modern neural networks due to its nonlinear properties and computational efficiency. The adam algorithm was adopted as the optimization algorithm, combining the advantages of adaptive methods and accelerating the learning process in complex parameter spaces.

The search found the following hyperparameters that achieved the highest score of 51.94% in 5-fold cross-validation:

- Alpha parameter: 0.0001. This means weaker regularization, meaning a smaller penalty on large weights than when choosing a value of 0.001;
- Batch size: 32;
- Number of layers: 2;
- Number of neurons in the first layer: 256;
- Number of neurons in the second layer: 128;
- Learning rate: 0.0001. This means that the model will update its weights in small steps, or at least smaller ones than if it were set to 0.001 or 0.01;
- Maximum number of iterations: 100.

Based on the obtained metrics assessing the quality of the artificial neural network model, it can be seen that the model performs best at predicting home team wins. A high sensitivity score of 0.82 indicates that most cases in this class were correctly identified. The precision value for this category is 0.52, meaning that more than half of the home team's victory predictions were accurate, and an F1-score of 0.63 indicates a good balance between sensitivity and precision.

The model performs significantly worse for the "tie" class. Both precision (0.26) and sensitivity (0.02) are very low, resulting in an F1-score of just 0.05. This means that the model hardly identifies ties correctly at all, which may be due to insufficient examples of this class in the dataset or insufficient discrimination of tie-specific features.

For the "away team win" category, the model achieved moderate results: a precision of 0.49 and a sensitivity of 0.44, resulting in an F1-score of 0.47. This indicates some model performance in this category, but there is still room for improvement, especially in correctly identifying this class in the test data.

Summary

The aim of this work was to build and compare the effectiveness of selected machine learning models in predicting football match outcomes. Based on collected match data and team statistics, a set of features was created, including recent form, match location, and contextual information. Additionally, exponentially smoothed indicators were introduced to account for the teams' current form dynamics.

In the experimental section, four classification models were tested: multiclass logistic regression, random forest, XGBoost, and artificial neural network. In each case, in addition to the logistic regression model, hyperparameter selection was performed using 5-fold cross-validation (grid search with 5-fold cross-validation), and prediction quality was assessed using standard metrics: accuracy, precision, sensitivity, F1-score, and logarithmic loss.

Analysis of the results showed that the logistic regression model, theoretically the simplest of the models used, achieved the best accuracy. All models achieved very similar results in metrics and the error matrix. The largest difference between the artificial neural network model and the logistic regression model was observed in the number of predicted draws. Using a naive reference model (naive bias), assuming a home team victory in every situation, provided a benchmark against which to evaluate the effectiveness of more complex models.

The similarity in the model results suggests that the data may be the problem. It's possible that more class balance is required. Difficulties predicting ties might indicate that it would be worthwhile to generate additional columns, one that would predict a tie. Another option could be to split the prediction into two parts. First, predict whether there will be a tie or not, and then, if the model determines there won't be a tie, predict the remaining two classes.

Possible directions for further research include incorporating temporal characteristics, dynamic weightings for various variables, and experimenting with more advanced neural network architectures. Furthermore, utilizing a larger dataset or incorporating individual player statistics in some form could also be interesting avenues for research development.

The research has the potential to further refine and develop machine learning models that predict the outcomes of football matches.

Acknowledgment

The work was financed by the Military University of Technology in Warsaw, Poland as part of the project No. UGB 531-000023-W500-22.

References

- Powers, DMW. (2011), 'Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation', International Journal of Machine Learning Technology 2:1, 37-63.
- Lawrence, T. (2018), 'Introducing xGChain and xGBuildup'. [Online], [Retrieved October 22, 2025], https://www.hudl.com/blog/introducing-xgchain-and-xgbuildup
- Agresti, A. (2013), Categorical Data Analysis, Wiley.
- Breiman, L. (2001), 'Random Forests'. Machine Learning 45, 5–32, https://doi.org/10.1023/A:1010933404324
- Witten, I.H., Frank, E., and Hall, M.A. (2011) Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann.
- Chen, T. and Guestrin, C. (2016), 'XGBoost: A Scalable Tree Boosting System'. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.
- Goodfellow, I., Bengio, Y., Courville, A. (2016), Deep Learning. MIT Press.
- Géron, A. (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
- Probst, P. and Boulesteix, AL. (2018), 'To Tune or Not to Tune the Number of Trees in Random Forest'. Journal of Machine Learning Research, 18(181), 1–18.
- Chen, T. and Guestrin, C. (2016), 'XGBoost: A Scalable Tree Boosting System'. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.
- Masters, D. and Luschi, C. (2018), 'Revisiting Small Batch Training for Deep Neural Networks'. arXiv:1804.07612