CRISP-DM Based Analysis of Second-Half Goal Scoring in Football

Gabriele Santi
University of Bologna
Email: gabriele.santi6@studio.unibo.it

***Abstract*—Football is a highly complex and unpredictable domain, where a limited number of events can strongly influence the final outcome of a match. In this work, the focus is the problem of predicting the number of goals scored by the home team in the second half of a football match, using only information available at half-time. The analysis follows the CRISP-DM methodology, starting from business understanding and data understanding, through feature engineering and modeling, up to evaluation and interpretation of results. Several machine learning models are tested and compared, highlighting both their predictive capabilities and their limitations in such an uncertain domain.**

1. Introduction

The work presented in this paper follows the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, which provides a structured framework widely used in data mining problems. Rather than focusing exclusively on model training, the analysis emphasizes problem understanding, data exploration, feature construction and result interpretation. Each phase of the CRISP-DM process is reflected in the structure of this paper, from the definition of the business problem to the evaluation of the obtained results. The complete implementation, including data preprocessing and model training, is provided in a dedicated Google Colab notebook available here : https://colab.research.google.com/github/gabbosanti/machine-learning-assignment/blob/main/assignement.ipynb
This paper focuses on explaining the methodological choices and the insights derived from the analysis.

1.1. Motivation of the domain and the problem

Football is not only a very popular sport, but also a domain characterized by a high level of uncertainty. Unlike many other contexts, match outcomes are often influenced by a small number of key events such as goals, penalties, or red cards, which can strongly affect the final result. For this reason, good overall performance does not always lead to a positive outcome.

Moreover, the information available in football datasets usually provides only a partial description of what happens during a match. Event-based data can capture observable actions on the field, but they do not include important factors such as players' psychological condition, refereeing decisions, or unexpected situations. In addition, football matches are highly dynamic, since teams often change their tactical approach during the game in response to goals or other events.

Because of these characteristics, football represents a challenging domain for machine learning. Instead of trying to fully eliminate uncertainty, this work uses football as an example to study how much predictive information can be extracted from noisy and incomplete data, and to understand the limitations imposed by the domain itself.

1.2. Goals of the work

The goal of this work is to evaluate whether match event data can be used to predict the offensive outcome of the home team in the second half of a football match.

More specifically, the task is formulated as a binary classification problem that aims to predict whether the home team scores at least one goal during the second half, based only on information available earlier in the match.

This target was chosen for both domain-related and modeling reasons. The second half is often influenced by what happened in the first half, including team performance, tactical adjustments, and game dynamics, making it an interesting phase to analyze.

In addition, the problem is defined in a realistic predictive setting. Only events from the first half of the match are used as input features, ensuring that no future information is available to the model. This choice avoids information leakage and makes the prediction task consistent with a real-world scenario.

2. Proposed Method

2.1. Data Understanding

The first step of the CRISP-DM process concerns the identification and understanding of data that are useful for the domain and for the defined objective. Several publicly available football datasets can be found online; however, most of them are limited to match-level statistics and rely on a small number of aggregated features, such as final score or total shots.

For the purpose of this work, it is important to collect more detailed information, including both offensive and disciplinary events, in order to better describe the dynamics of a football match. For this reason, an event-based dataset was preferred.

One of the most complete and general-purpose datasets was found on Kaggle at the following URL: *https://www.kaggle.com/datasets/secareanualin/football-events*.

The dataset is organized into three files:

- `events.csv` which contains detailed event-level data for each match;
- `ginf.csv` which contains match metadata and market odds;
- `dictionary.txt` which provides a textual description for categorical variables encoded as integers.

A preliminary inspection highlights the large amount of available data: almost one million event records are stored in `events.csv`, referring to approximately nine thousand matches. However, `ginf.csv` contains around ten thousand matches. This difference is due to the presence of matches without any associated event-level information.

Since the feature construction in this work relies entirely on match events, matches without recorded events cannot contribute to the analysis and were therefore excluded. The remaining matches were uniquely identified using the `id_odsp` field, which is common to both datasets.

The main idea is to aggregate the occurrences of relevant events for each match and team (home and away) and then merge the resulting features at match level.

2.2. Data Preparation

The data preparation phase focuses on transforming raw event data into a structured dataset suitable for machine learning.

As a first step, only events occurring in the first half of the match were considered. Although injury time may extend beyond the 45th minute, the dataset aggregates these events under minute 45; therefore, minute 46 is consistently treated as the beginning of the second half.

The original dataset contains a large number of attributes, many of which are either redundant or not directly relevant for the predictive task. Including too many features would increase model complexity and the risk of overfitting. For this reason, features related to textual descriptions, individual players, detailed spatial information and event timing were discarded.

Another important issue concerns missing events: matches without event records cannot be used for feature engineering and were excluded from the analysis, as discussed in the Data Understanding phase.

The core of the data preparation process is feature engineering. Using the numerical encoding provided in `dictionary.txt`, new features were created to represent the total number of relevant events occurring during the first half of each match. In particular, the following events were considered: shots on target, shots off target, corners, free kicks, offsides, fouls, yellow cards and goals.

Since each event is associated with either the home or the away team, all features were computed separately for the two sides. The resulting home and away datasets were then merged at match level.

Finally, the target variable was defined by comparing the total number of home goals scored in the full match with the number of home goals scored during the first half. If the difference was greater than zero, the target was set to one, indicating that the home team scored at least one goal in the second half; otherwise, it was set to zero.

2.3. Modelling

The dataset was split into training and test sets using an 80/20 ratio. The training set was used for model fitting and hyperparameter tuning, while the test set was kept aside and used only for the final evaluation in order to avoid data leakage.

2.3.1. Decision Tree. A Decision Tree classifier was selected as a baseline interpretable model, due to its ability to capture non-linear relationships and interactions between features without requiring feature scaling. The performance of a Decision Tree classifier strongly depends on its hyperparameters, which control the complexity of the model and the trade-off between bias and variance. For this reason, hyperparameter tuning was performed using *RandomizedSearchCV* with 5-fold cross-validation on the training set.

The following hyperparameters were tuned:

- **criterion**
  Controls the purity index used to measure the quality of a split. Gini impurity and entropy are the two most common splitting criteria and may lead to slightly different tree structures.
- **splitter**
  Determines whether the best possible split or a random split among the best candidates is selected. The random option can reduce variance and help avoid overly deterministic trees.
- **max_depth**
  Limits the maximum depth of the tree, directly controlling model complexity. Small depth values were chosen to reduce overfitting and encourage better generalization.
- **min_samples_split**
  Defines the minimum number of samples required to split an internal node. This parameter further regularizes the tree and prevents splits based on very few samples.
- **min_samples_leaf**
  Specifies the minimum number of samples required to be present in a leaf node. Higher values lead to smoother decision boundaries and reduce sensitivity to noise.
- **max_features**
  Limits the number of features considered at each split, introducing randomness and reducing correlation between splits.

RandomizedSearchCV was preferred over GridSearchCV in order to explore a broader hyperparameter space with lower computational cost. A total of 50 random configurations were evaluated, optimizing accuracy as the target metric.

2.4. Random Forest

The Random Forest model was trained using a Randomized Search with 5-fold cross validation in order to find a good set of hyperparameters. The search explored different configurations to balance model complexity and generalization.

The main hyperparameters considered during the search were:

- **number of trees**
  Values between 100 and 300, to evaluate the trade-off between stability and computational cost.
- **naximum depth**
  limited to relatively small values to reduce overfitting.
- **minimum samples to split a node**
  increased to avoid overly specific splits.
- **minimum samples in a leaf**
  used to smooth predictions and improve generalization.
- **number of features considered at each split**
  set to `sqrt` or `log2` to introduce randomness among trees.

The best model was selected based on cross-validation accuracy. On the test set, Random Forest achieved slightly better results than the single Decision Tree, but the overall performance remains limited.

This result is not surprising: even if Random Forest reduces overfitting by averaging multiple trees, it still relies on the same input features, which are only weakly informative for predicting football match outcomes. Precision and recall values remain relatively low, showing that the model struggles to correctly identify the true class.

Overall, Random Forest improves stability and robustness, but it cannot overcome the intrinsic unpredictability of football matches.

2.5. AutoML

AutoML was used to automatically select both the model and its hyperparameters within a fixed time budget of 60 seconds. The goal was to evaluate whether an automated approach could discover a better configuration compared to manual tuning.

During the AutoML process, the following aspects were automatically optimized:

- **Model selection**: different classification algorithms were evaluated without manual intervention.
- **Hyperparameter tuning**: each candidate model was internally tuned to maximize validation accuracy.
- **Model complexity**: simpler models were preferred when more complex ones did not provide clear improvements.

The model selected by AutoML achieved performance comparable to the manually tuned Decision Tree and Random Forest models. However, no significant improvement was observed on the test set.

These results suggest that the main limitation is not the lack of tuning or model selection, but the intrinsic difficulty of the task. The available features do not capture many external and unpredictable factors that influence football matches.

In conclusion, AutoML confirms that low predictive performance is a realistic outcome for this problem, rather than a failure of the modeling approach.

3. Results

In this section, the results obtained from the different modeling approaches are presented and compared. Three models were evaluated: a Decision Tree, a Random Forest, and an AutoML solution. All models were trained on the same training set and evaluated on the same test set to ensure a fair comparison.

The evaluation was based on Accuracy, Precision, and Recall, since the task is a binary classification problem. In particular, Recall is important because it measures how well the model identifies matches where the home team scores in the second half.

Table 3 summarizes the results obtained on the test set.

Model	**Accuracy**	**Precision**	**Recall**
Decision Tree	0.567	0.571	0.901
Random Forest	0.583	0.590	0.831
AutoML	0.577	0.580	0.881

From the results, it can be observed that the Random Forest achieved the highest accuracy and precision among the tested models. However, the Decision Tree obtained the highest recall, meaning it was better at detecting matches where the home team scored in the second half.

The AutoML approach produced competitive results, slightly improving over the Decision Tree in terms of accuracy, but without a clear advantage over the Random Forest. Overall, the performance differences between the models are relatively small.

4. Conclusion

The goal of this project was to predict whether the home team would score at least one goal in the second half of a football match, using only information from first-half events. The analysis followed the CRISP-DM methodology, covering data understanding, preparation, modeling, and evaluation.

Several machine learning approaches were tested, including a Decision Tree, a Random Forest, and an AutoML solution. Although hyperparameter tuning and cross-validation were applied, none of the models achieved very high predictive performance.

This result is not necessarily a failure. Football matches are highly unpredictable, and many important factors influencing second-half goals (such as tactical changes, player fatigue, substitutions, or unexpected events) are not captured by the available data. Therefore, a moderate accuracy close to 60% is reasonable for this type of problem.

The comparison between models shows that ensemble methods like Random Forest slightly improve stability and accuracy, while simpler models such as Decision Trees can still perform well in terms of recall. AutoML proved to be a useful tool to quickly obtain a competitive baseline without manual model selection.

Possible future improvements could include the use of additional contextual features, such as team strength indicators, historical performance, or betting odds, as well as experimenting with more advanced models or longer time budgets for AutoML.