Supervised Machine Learning: Research on Predicting English Premier League Match Outcomes Based on an AdaBoost Classifier

Junbo Feng
Department of Engineering, Bucknell University, Lewisburg, PA, U.S.A.

Keywords: AdaBoost, Machine Learning, Ensemble Learning, English Premier League, Sports Analytics.

Abstract: This paper focuses on the application of the Adaptive Boost (AdaBoost) classifier in the task of predicting match results in the English Premier League (EPL). The research aims to predict over 80% of match results in EPL correctly, but it does not aim to accurately predict the scores. The study compares AdaBoost's performance with a baseline Random Forest classifier using data from several EPL seasons. Preprocessing steps include data cleaning, feature selection, and integration of match-related data. The AdaBoost model achieved an accuracy of 86.65%, outperforming the Random Forest's 84.15%. This indicates the practicability of using the AdaBoost model in predicting football match results. The model could be refined in future work to account for additional variables such as referee decisions, match time, and personnel choices excluded in data preprocessing. This research provides a basic approach for using advanced machine learning techniques in sports predictions and identifies areas that can be potentially improved to generate predictions with higher precision.

1 INTRODUCTION

As one of the most popular and complex sports in the world, (Premier League Competition Format & History | Premier League, 2018) football captivating millions of fans with its unpredictable (Anfilets et al., 2020) outcomes. To accurately predict (Mattera, 2021) football match results, sports enthusiasts and analysts put countless efforts to create methods that can possibly make it come true (Rahul Baboota & Kaur, 2019). However, many uncontrollable and external and factors, such as injuries, weather conditions, referee decisions, and team performances, can all determine the outcome of the match, (Raju et al., 2020) making it challenging to accurately predict the scores. Even though it is impossible to know the exact scores, machine learning techniques have become an approach that can potentially do a better job on solely predict the match result based on its ability to learn from data.

Traditional methods (Rana & Vasudeva, 2019) often rely on basic statistics, past match results between two teams, and expert opinions, which makes stable predictions impossible in the long term. Additionally, some more comprehensive machine learning models, such as Random Forest classifiers (KINALIOĞLU & KUŞ, 2020), have been applied to this problem. In those existing methods, the accuracy achieved could rarely exceed 85%. Thus, there is room for improvement. Method-wise, current models do not fully realize the potential of ensemble learning techniques that could enhance predictive performance.

This research utilizes the AdaBoost classifier as the primary method for predicting EPL match outcomes, with the Random Forest classifier serving as a baseline method (Chakraborty et al., 2024) for comparison. AdaBoost is an ensemble learning technique that enhances performance by combining multiple weak classifiers into a more robust, more accurate model. This research aims to achieve a prediction accuracy of over 85%. By comparing the performance of the AdaBoost and Random Forest models, this research aims to demonstrate the potential of ensemble methods in sports analytics and identify areas where predictive models can be further improved.

Feng and J.
Supervised Machine Learning: Research on Predicting English Premier League Match Outcomes Based on an AdaBoost Classifier.
DOI: 10.5220/0013486900004619
In *Proceedings of the 2nd International Conference on Data Analysis and Machine Learning (DAML 2024)*, pages 35-39
ISBN: 978-989-758-754-2

2 METHODS

The primary training method of this research is the AdaBoost Classifier (Chen et al., 2019) in the field of ensemble machine learning approach (Navlani, 2024). Boosting algorithms consist of multiple lower-accurate classifiers that merge into a highly accurate classifiers. A lower-accuracy classifier, or weak classifier, does better than random guessing. A highly accurate classifier, or strong classifier, has an error rate close to zero. Boosting algorithms can find models that make incorrect predictions and are less likely to overfit. The three algorithms below are prevalent in data science competitions. Figure 1 shows the basic principle of the ensemble machine learning approach.

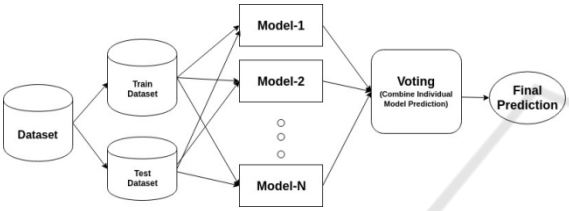

Figure 1: Basic principle of Ensemble Machine Learning Approach (Picture credit: Original)

AdaBoost Classifier was proposed as one of the ensemble boosting classifiers by Yoav Freund and Robert Schapire in 1996, combining multiple poorly performing classifiers to increase its accuracy. The core idea of AdaBoost is to train the data sample in each iteration and adjust the weights of classifiers in order to improve the accuracy of predicting difficult observations. Any machine learning algorithm can serve as the base classifier if it can handle weights on the training set. There are two conditions that need to be met in AdaBoost: Using various weighed training examples to train the classifier interactively. Also, the classifier should aim to minimize training error and provide an appropriate fit for training examples in each iteration.

In Figure 2, it shows the algorithm for adaptive boosting. The following steps are required to work properly. Initially, Adaboost randomly selects a subset of the training data. It then trains the model iteratively, with each new training set chosen based on the accuracy of the previous iteration's predictions. Misclassified observations are given higher weights for a greater chance of being correctly classified in the next iteration. The trained classifier is also assigned a weight based on accuracy, with more accurate classifiers receiving higher weights. This process continues until the entire training data is correctly classified or the maximum number of estimators is reached. For classification, the final decision is made by "voting" across all the built classifiers.

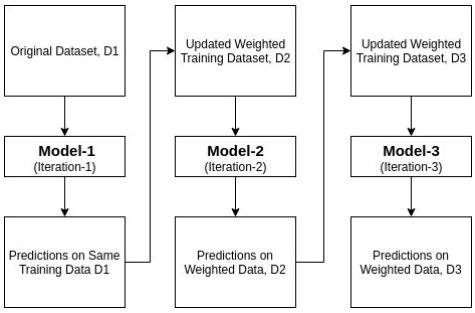

Figure 2: Algorithm of Adaptive Boosting (Picture credit: Original)

2.1 The Weak Learner

To be specific, here are some mathematical and theoretical explanations of AdaBoost Classifier. AdaBoost starts by initializing equal weights for all the training samples. This means that initially, the algorithm treats each sample as equally important.

$$w_i^{(1)} = \frac{1}{n}, \quad i = 1,2,\ldots,n \tag{1}$$

where n is the total number of training samples, while the weight $w_i^{(1)}$ represents the importance of the i-th sample in the first iteration. Since all weights are equal, each sample has an equal chance of influencing the first weak learner.

The algorithm then trains a weak learner $h_t(x)$ using the current weights $w_i^{(t)}$. A weak learner is a simple model that performs just slightly better than random guessing. Decision stumps (one-level decision trees) are commonly used as weak learners.

$$h_t(x) \in \{-1, +1\} \tag{2}$$

The weak learner $h_t(x)$ outputs a prediction for each training sample x. The output is typically binary $(-1 \text{ or } +1)$ in classification tasks. The goal at this step is to train the weak learner to minimize the weighted classification error, which is influenced by the current weights.

2.2 Weak Learner's Error

After training the weak learner, AdaBoost calculates its weighted error ϵ_t by:

$$\epsilon_t = \sum_{i=1}^{n} w_i^{(t)} \cdot 1(y_i \neq h_t(x_i)) \quad (3)$$

The weighted error ϵ_t is a measure of how well the weak learner $h_t(x)$ performs on the training data. The indicator function $1(y_i \neq h_t(x_i))$ equals 1 if the weak learner misclassifies sample i and zero otherwise. This equation sums up the weights of the misclassified samples, meaning that if a weak learner makes more mistakes on samples with high weights, the error ϵ_t will be higher.

2.3 Compute Learner's Weight

AdaBoost assigns a weight α_t to the weak learner based on its performance:

$$\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t}) \quad (4)$$

which determines the importance of the weak learner in the final classifier. If the weak learner' s error is low (approaching to 0), then α_t is high, which means that the weak learner is given more influence in the final decision-making.

2.4 Update Weights

The weights of the training samples are then updated for the next iteration:

$$w_i^{(t+1)} = w_i^{(t)} \cdot exp(a_t \cdot 1(y_i \neq h_t(x_i))) \quad (5)$$

After updating weights, they need to be normalized to sum to 1:

$$w_i^{(t+1)} = \frac{w_i^{(t+1)}}{\sum_{j=1}^{n} w_j^{(t+1)}} \quad (6)$$

which ensures that everything remains a valid probability distribution and makes convenience for the next iteration's training process. The steps above will then be repeated for a predefined number of iterations until the model achieves a desired level of accuracy.

2.5 Final Strong Classifier

The The final classifier *H(x)* is a weighted combination of all the weak learners:

$$H(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)) \quad (7)$$

where the part in sign function represents the weighted sum of the predictions made by all weak learners, while T is the total amount of weak learners. To make a prediction, the final classifier *H(x)* considers contributions of all the weak learners weighted by their actual importance.

3 EXPERIMENT

In the following experiment, there are two goals for the architectures. 1) To predict the match results correctly and successfully for more than 85% of all the test cases. 2) To evaluate the current method (AdaBoost Classifier) by comparing the accuracy with the baseline method - Random Forest Classifier. The experiment starts with data preprocessing.

3.1 Data and Preprocessing

In the preprocessing phase, raw football match datasets were first loaded from multiple CSV files using the `pandas` library. Irrelevant columns, such as 'Div', 'Date', and 'Referee', were systematically removed to eliminate noise from the dataset. The cleaned data from each file was then concatenated into a single comprehensive data frame, allowing for a unified analysis across multiple seasons. The preprocessing pipeline included essential steps such as data cleansing, feature selection, and data integration.

Following this, the dataset was prepared for machine learning by splitting it into training and testing sets using corresponding function calls. Finally, the processed dataset was saved to a specified path, ensuring that subsequent analyses could be conducted efficiently without repeating the preprocessing steps. This systematic approach to data preprocessing ensured that the dataset was in an optimal state for model training and evaluation.

3.2 Baseline Method

The baseline method for comparison is the random forest classifier, which involves training a model with 700 estimators with maximum depth of 45, on pre-processed football match datasets. The model first trained specifically for predicting the results of matches between the Big 6 teams in the EPL - Arsenal, Liverpool, Manchester United, Manchester City, Tottenham Hotspurs, and Chelsea - by utilizing

the datasets from season 2015-16 to season 2019-20. This is because the EPL has a cycle of relegation and promotion, where the teams who finish in the bottom three of the league table at the end of the campaign, are relegated to the Championship, the second tier of English football. In this sense, only the Big 6 teams can stay in Premier League for a relatively long time because the strength of these teams can prevent them from being relegated, providing sustainable and stable data for training. After obtaining enough training data, the model then studied the entire Premier League match results season by season, even though some team only played for 1 season, making a lot of difficulties the prediction. As the result, the "RandomForestClassifier" successfully predicted 84.15% of the matches correctly, which can barely meet the initial expectation of this research.

3.3 Details of Training

In the experiment, the AdaBoost classifier is designed to predict the outcomes of EPL matches in the same way as the baseline method does. The training dataset is processed and split using a custom preprocessing function, where 90% of the data is used for training and 10% for testing. The model is trained using the `.fit(X_train, y_train)` method on the training data. Once trained, the AdaBoost classifier is used to predict the outcomes of specific EPL matches between teams like Liverpool and Manchester City by converting match details into feature vectors. The model's predictions are then compared against the actual outcomes to evaluate its performance.

As the result, the "AdaBoostClassifier" successfully predicted 86.65% of the matches correctly, which satisfies the initial expectation of this research. The detailed training results are shown in Figure 3 below. The prediction accuracy of RandomForestClassifier in EPL is 84.15%, the one for Big 6 teams is 91.93%. The research method is slightly better than the baseline method: it has prediction accuracy of 86.65% in EPL, and 93.12% when Big 6 teams plays against each other.

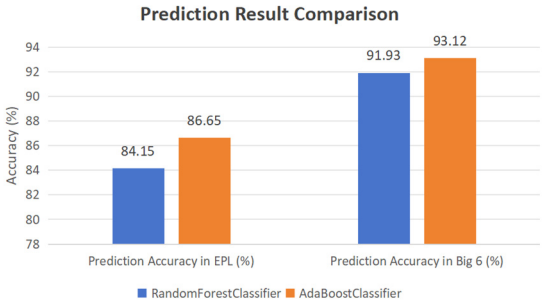

Figure 3: Prediction result between "Random Forest Classifier" and "AdaBoost Classifier"

(Picture credit: Original)

3.4 Model Evaluation

Initially, the expectation of this research is to create a machine learning model that can properly clean up the datasets and predict over 85% of the matches correctly. The chosen model, "AdaBoostClassifier", successfully satisfied the basic requirement by ending up with the overall accuracy of 86.65%. However, this accuracy is calculated by the "accuracy()" function in the model. The actual result is slightly worse than the system expected. In the prediction to all match results in EPL, season 2020-2021, the model predicted correctly for 309 out of 380 games of the entire season, which ends up with an practical accuracy of 81.32%. This difference may be led by the factor of referee, and the uncertainty of the sports of football itself. Since the referee is initially excluded from the training dataset, the model assumes that all the referees are fair and only the teams' performance and tactics can determine the outcomes. Because it cannot reach the theoretical accuracy in practice, there are still many aspects that can be potentially improved.

4 CONCLUSIONS

The research successfully demonstrated the effectiveness of the AdaBoost classifier in predicting English Premier League match outcomes, achieving an accuracy of 86.65%, which slightly outperforms the baseline Random Forest model's 84.15%. While the model met the initial goal of over 85% accuracy, practical application showed a slightly lower performance of 81.32% when tested on all matches in the 2020-2021 EPL season. This discrepancy is attributed to the exclusion of certain variables, such as referee decisions and other external factors, during the data preprocessing stage. Future work can focus on incorporating these additional variables to further improve prediction accuracy and enhance the model's robustness in handling the complexities of football matches. Additionally, experimenting with different machine learning algorithms and optimizing the AdaBoost model could provide further advancements in sports analytics.

REFERENCES

Anfilets, S., Bezobrazov, S., Golovko, V., Sachenko, A., Komar, M., Dolny, R., Kasyanik, V., Bykovyy, P., Mikhno, E., & Osolinskyi, O. (2020). DEEP

MULTILAYER NEURAL NETWORK FOR PREDICTING THE WINNER OF FOOTBALL MATCHES.

Chakraborty, S., Dey, L., Maity, S., & Kairi, A. (2024). Prediction of winning team in soccer game: A supervised machine learning-based approach. In Prediction of Winning Team in Soccer Game-A Supervised Machine Learning-Based Approach (pp. 170–186). CRC Press.

Chen, S., Shen, B., Wang, X., & Yoo, S.-J. (2019). A strong machine learning classifier and decision stumps based hybrid AdaBoost classification algorithm for cognitive radios. Sensors, 19, 23.

KINALIOĞLUİ. H., & KUŞ, C. (2020). Prediction of UEFA champions league elimination Rounds winners using machine learning algorithms. Cumhuriyet Science Journal, 41(4), 951–967.

Mattera, R. (2021). Forecasting binary outcomes in soccer. Annals of Operations Research.

Navlani, A. (2024). AdaBoost Classifier Algorithms using Python Sklearn Tutorial. www.datacamp.com.

Premier League Competition Format & History | Premier League. (2018). Premierleague.com.

Rahul Baboota, & Kaur, H. (2019). Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting, 35, 2.

Raju, M. A., Mia, M. S., Sayed, M. A., & Uddin, R. (2020). Predicting the outcome of english premier league matches using machine learning. 1–6.

Rana, D., & Vasudeva, A. (2019). Premier League Match Result Prediction using Machine Learning.