IE582 Term Project

Derin Çağan Temeltaş - 2022702138

Zelina Genel - 2023802015

1. INTRODUCTION

1.1 Problem Description

Forecasting the sports results have attracted significant popularity among sports fans in the last years. While the pre-match odds are usually dependent on the historical data and statistical models, in-game odds are dynamic which are transformed throughout the game. In-game odds are affected by many factors occurring during the game such as game statistics, market behavior and emotional reactions. The interpretation of the match dynamics is crucial for the fans that bet according. The interpretation is dependent on factors such as the player performances, goals scored, given red cards or overall possession of the game.

For in-game odds in the real world, people place bets only based on the data available up to the time they place the bet. The rest of the data after bet is placed is naturally not available for them to make use of it. Hence, without knowing the rest of the game, they should select an optimal time stamp to bet to maximize their gain.

1.2 Descriptive Analysis of Data

Before building the model, let us provide some information related to the data available to us. We are given detailed match statistics for 648 games with 106 columns each representing a feature related to a game. For convenience, we might categorize these columns as columns related to

- general game information such as 'fixture_id', 'halftime', 'current_time', 'half_start_datetime', 'match_start_datetime', 'minute', ‘halftime’
- game statistics of teams such as 'Assists - away', 'Assists - home', 'Attacks - away', 'Ball Possession % - away', 'Ball Safe - away', 'Ball Safe - home', 'Challenges - away', 'Corners - away', 'Counter Attacks - away', 'Counter Attacks - home', 'Dangerous Attacks - away',
- bookmaker related information 'latest_bookmaker_update', 'suspended', 'stopped', '1', '2', 'X',
- critical changes such as 'Penalties - away', 'Penalties - home', 'Redcards - away', 'Redcards - home', 'Yellowcards - away', 'Goals - away', 'Goals - home'.

2. LITERATURE REVIEW

For predictions related to sports, data mining, a commonly used technique for event prediction and explanation, is a suitable instrument. In recent years, a variety of data mining approaches have been applied to forecast game outcomes, including artificial neural networks

(ANN), decision trees, the Bayesian method, logistic regression, SVM, and fuzzy algorithms [9]. Once the data is collected, feature selection is a critical step to define the accuracy of the predictions. McCabe and Travathan [1] utilized 11 features that can be commonly found in all sports. Zdravevski and Kulakov [2] referred to opinions of experts in the field and selected top 10 features with most influence on the results. Trawinski [3] used Waikato Environment for Knowledge Analysis (WEKA) and eight feature selection algorithms, selecting 5 features. Buursma [4] started with a dataset containing 10 features and eliminated these features through one feature at a time method to use a classification algorithm. Using neural networks Ivankovic et al. [5] weighted 9 features.

After selecting of the features, various data mining methods are very useful assessing the predictions related to game scores. In literature, we see various data mining methods employed such as decision trees [2,3], support vector machines [6], logistic regression [2,4,6], artificial neural networks [1,3,5,6,7], Bayesian method [2,4,6,8] and fuzzy methods [3] are used to predict the results.

The accuracy of these techniques is probably the most significant portion of these types of studies. To predict the results of basketball games, Cao [6] used a simple logistic classifier, naïve Bayes, SVM, and a multilayer perceptron neural network, calculating the accuracy as 67.82%, 65.82%, 67.22%, and 66.67% respectively for mentioned techniques. Miljkovic et al. [8] employed a naïve Bayes model by adding the data of previous' games to their dataset every day to predict NBA results. They performed k-fold cross validation for test and training datasets. Their resulting accuracy was 67.0%. Zdravevski and Kulakov [2] made use of the classification methods that are available in WEKA to forecast the game winner. 37 algorithms of WEKA were used to classify the data. To estimate the soccer game results, Buursma [4] employed several classification algorithms such as Bayesian network, naïve Bayes, decision tree and simple and logistic regression with accuracies of 54.55%, 54.43%, 57.00%, 55.05%, 54.98%, respectively.

3. APPROACH

In this project, we tried to come up with this optimal betting time by analyzing the training portion of the available data in order to forecast the match result as home win, draw, or away win together with a decision "bet home win", "bet draw", "bet away win" or to choose "no action". We used matches starting from "2024-11-01" (included) as our test data (total of 111 games). Our aim is to predict the game result at single moment in a game time without utilizing the rest of the data after that specific minute as it the case in real world.

This project is based on the decision tree model constructed in Homework 2 <link>. Further expanding the initial task of Homework 2, in this project we firstly introduce a live betting strategy, and upon predicting the game result based on the highest score returned, the model is evaluated for its accuracy and recall. Other main steps, such as pre-processing the data, along with the functionality of each step are explained in detail in the next section.

The overall outline of the project structure can be seen on Figure 1 below.

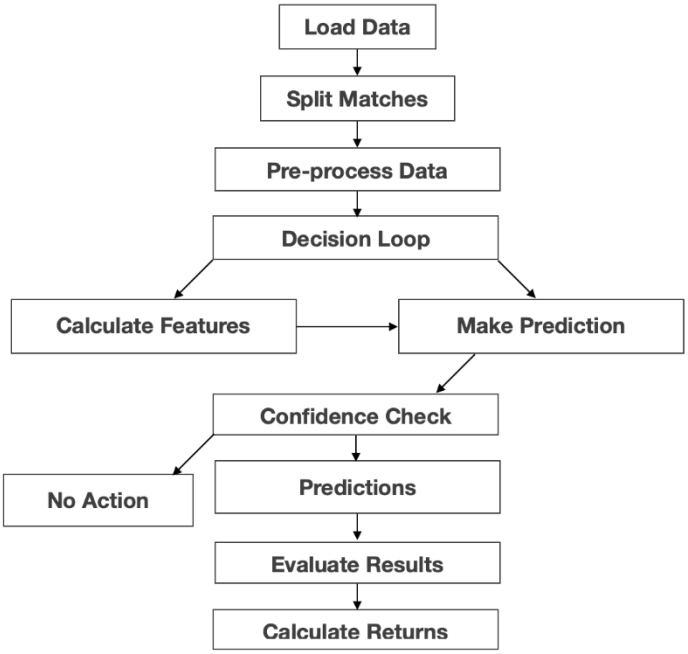

Figure 1. *The overall structure of the project.*

3.1 The Model and Functions

Step1: To start off, we have split the dataset into individual games by grouping them with their respective id's. The function and its purpose employed for Step 1 is;

split_into_matches(df: pd.DataFrame) -> List[pd.DataFrame]

- Splits the dataset into separate DataFrames for each match using the fixture_id.
- Returns a list of DataFrames, each corresponding to an individual match.

Step 2: We have defined a live betting strategy by specifying thresholds for odds, a confidence score and decision-making times. The minimum and maximum thresholds for odds are selected as 1.5 and 8.0, respectively, to filter out the unrealistic bets. We defined a minimum confidence score required to make a prediction as 0.75. We have selected decision times from 15th to 40th minute with 5-minute intervals. The function and its purpose employed for Step 2 is;

LiveBettingStrategy Class init()

- Initializes strategy parameters:
- min_odds_threshold: Minimum odds for a bet to be considered
- max_odds_threshold: Maximum odds for a bet to be considered.
- confidence_threshold: Minimum confidence score to place a bet.
- decision_times: Predefined times (15-40 minutes) for making decisions.

Step 3: In order to come up with predictions, we used match features to make a prediction about the outcome of the match. To this end, we calculated scores for the 'home', 'away', and 'draw' outcomes based on weighted features by using

- ‘home_implied_prob’, ‘home_attack_momentum’, ‘home_attack_momentum’, ‘away_attack_momentum’, ‘home_possession’ and ‘home_shot_accuracy’ for home score;
- ‘away_implied_prob’, ‘away_attack_momentum’, ‘home_attack_momentum’, ‘away_attack_momentum’, ‘away_possession’ and ‘away_shot_accuracy’ for away score;
- ‘draw_implied_prob’ for draw score.

The function employed for Step 3 is;

make_prediction(features: Dict) -> Tuple[str, float]

- Computes prediction scores for: Home win (home_score).
- Away win (away_score).
- Draw (draw_score).
- Combines factors like implied probabilities, attack momentum, possession, and shot accuracy.
- Selects the outcome with the highest score if it meets the confidence threshold; otherwise, recommends "no action."

Step 4: After creating a dictionary of scores for all outcomes, we find the outcome with the highest score and return "no action" if confidence is below the threshold. To calculate derived statistics for decision-making we manipulate the match data, that is

- We convert relevant columns to datetime format,
- Calculate the minutes played in the match,
- Compute the goal difference and total goals scored up to now,
- Calculate shot accuracy,
- Compute an attack momentum metric as a weighted combination of offensive statistics,
- Calculate possession efficiency,
- And finally Calculate the implied probabilities from betting odds.

The function employed for Step 4 is;

preprocess_match_data(df: pd.DataFrame) -> pd.DataFrame

- Adds calculated features to the dataset:
- Time Features: Match progression in minutes.
- Goal Metrics: Goal difference and total goals.
- Team Metrics: Shot accuracy, attack momentum, possession efficiency, and implied probabilities based on betting odds.

Step 5: Next, we calculate relevant features for decision making at a specific minute and get the data up to current minute to make a decision at the earliest available decision time without forward-seeking. If 'confidence score' is higher than the determined threshold the prediction is returned otherwise not.

The function employed for Step 5 is;

calculate_match_features(match_data: pd.DataFrame, current_minute: float) -> Dict

- Extracts the latest data up to the specified minute.
- Calculates features like attack momentum, shot accuracy, possession, and implied probabilities for both teams.

This approach ensures that decisions are made systematically, with all relevant match data processed in real-time. The integration of derived statistics such as goal difference, attack momentum, and implied probabilities allows for precise predictions. Additionally, the use of confidence thresholds minimizes the risk of making predictions based on uncertain data. By iterating through predefined decision times, the strategy provides the flexibility to identify optimal betting opportunities dynamically.

Step 6: With the data up to specified minute and related features, in this step, we find the optimal point to make a decision based on the confidence score and at which points it is attained.

The function employed for Step 6 is;

find_optimal_decision_point(match_data: pd.DataFrame) -> Tuple[str, float, float]

- Iterates through predefined decision times (15-40 minutes).
- Predicts the match outcome at each time and selects the first instance with sufficient confidence.
- Returns the prediction, confidence, and decision time.

3.3 Performance Evaluation Metrics and Results

In this section, we evaluate the accuracy of our model in terms of correctly predicted game results and betting returns are calculated. Our performance evaluation metrics are accuracy, precision, recall and F-measure.

To evaluate the model, below functions are employed with their respective functuonalities.

evaluate_prediction(prediction: str, final_result: str) -> bool

- Checks if the predicted outcome matches the actual result.

calculate_returns(prediction: str, final_result: str, odds: Dict[str, float]) -> float

- Calculates net returns for a correct prediction based on the betting odds.
- Returns -1 for incorrect predictions (lost stake) or 0 for no action.

run_strategy(matches_data: List[pd.DataFrame]) -> pd.DataFrame

- Applies the betting strategy to a list of matches.
- For each match: Processes the data.
- Determines the optimal decision point and prediction. Evaluates the prediction against the final result. Calculates returns and records the results.
- Returns a DataFrame summarizing the strategy's performance.
- The strategy avoids forward-looking by ensuring predictions are based only on data available up to the current decision minute.
- Implied probabilities are calculated directly from odds to standardize and normalize betting data.
- Attack momentum and shot accuracy are weighted heavily to reflect real-time match dynamics.
- The decision-making algorithm prioritizes early predictions with sufficient confidence to maximize practical applicability in live betting scenarios.
- The results include performance metrics like cumulative returns and accuracy, enabling thorough evaluation and refinement of the strategy.

We run the betting strategy on a list of 111 matches which is the test data that has been spared in the beginning. The strategy performance outcomes can be seen on the table below:

Strategy Performance:	
Total Matches Analyzed:	111
Total Predictions Made	26
Correct Predictions	18
Accuracy	69.23%
Cumulative Returns	-7.72 units
Precision	0.69230 (69.23%)
Recall	0.17480 (17.48%)
F-measure	0.27910 (27.91%)

The model demonstrates strong performance in terms of precision, effectively minimizing false positives and ensuring the quality of its predictions. It strategically selects decision times, optimizing its predictions and avoiding random outputs. The flexibility of its parameters, such as odds thresholds and confidence scores, allows customization to fit various scenarios. The inclusion of the "No Action" option highlights the model's ability to avoid unnecessary risks by only acting in high-confidence situations, reducing potential losses. Tested on a dataset of 111 matches, the model has proven its capability on a substantial sample

size, making it a reliable tool for initial investment analysis. While its recall could be improved to identify a larger portion of correct predictions, its current precision (69.23%) offers a solid foundation. Additionally, the model's systematic and structured approach ensures consistent and strategic decision-making, with significant potential for further development and optimization.

4. CONCLUSION AND FUTURE DIRECTIONS

The live betting strategy implemented in this model demonstrates a systematic and structured approach to decision-making in dynamic scenarios. With a strong emphasis on precision, the model effectively minimizes false positives by focusing on high-confidence predictions. Its design parameters, such as predefined decision times and confidence thresholds, ensure that it operates within a reliable and customizable framework, offering a practical tool for real-time betting decisions.

Despite its strengths in precision, the model's recall highlights areas for improvement, as it currently identifies a limited portion of correct predictions relative to the total possible. This limitation suggests potential enhancements in incorporating more features or refining the weighting of existing features to increase sensitivity to a broader range of scenarios. Additionally, the cumulative returns being slightly negative indicate the need for further optimization in the decision-making process, such as fine-tuning thresholds for odds and confidence scores or incorporating a dynamic adjustment mechanism based on match context.

For future directions, several improvements can be explored:

- **Feature Expansion:** Integrating additional match statistics, such as player-level data or historical team performance, could provide deeper insights and improve predictive accuracy.
- **Adaptive Strategies:** Implementing machine learning models, such as reinforcement learning, could allow the strategy to adapt dynamically based on real-time outcomes and changing match contexts.
- **Enhanced Recall:** Adjusting the balance between precision and recall to ensure a higher coverage of correct predictions while maintaining the model’s robustness.
- **Economic Analysis:** Introducing a cost-benefit analysis layer to weigh potential returns against risks, optimizing the financial outcomes of the strategy.
- **Scalability:** Testing the model on larger datasets and different leagues or sports to evaluate its generalizability and scalability.

In conclusion, the model provides a solid foundation for live betting decision-making, combining statistical rigor with practical applicability. With further enhancements, it has the potential to become a highly efficient and versatile tool in the field of predictive analytics for sports betting.

5. CODES

The Python codes for this project can be found here. You can visit our GitHub page for all the documents related to this project.

6. REFERENCES

[1] McCabe, A., Travathan, J., "Artificial Intelligence in Sports Prediction", IEEE Computer Society Washington, DC, USA, 2008, pp. 1194-1197 .

[2] Zdravevski, E., Kulakov, A., "System for Prediction of the Winner in a Sports Game", In: ICT Innovations 2009, Part 2, 2010, pp. 55–63.

[3] Trawinski, K., "A fuzzy classification system for prediction of the results of the basketball games", IEEE International Conference on Fuzzy Systems, Barcelona, Spain, 2010, pp.1- 7.

[4] Buursma, D., "Predicting sports events from past results Towards effective betting on football matches", Conference Paper, presented at 14th Twente Student Conference on IT, Twente, Holland, 21 January 2011.

[5] Ivankovic, Z., Rackovic, M., markoski, B., Radosav, D., Ivkovic, M., "Analysis of basketball games using neural networks", 11th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, November 2010, pp.251-256.

[6] Cao, C., "Sports data mining technology used in basketball outcome prediction", Master dissertation, Dublin Institute of technology, Ireland, 2012.

[7] Kahn, J., "Neural Network Prediction of NFL Football Games", 2003, available on http:// homepages.cae.wisc.edu/~ece539/project/f03/kahn.pdf

[8] Miljkovic, D., Gajic, L., Kovacevic, A., Konjovic, Z., "The use of data mining for basketball matches outcomes prediction", IEEE 8th International Symposium on intelligent and informatics, Subotica, Serbia, 2010, pp.309-312.

[9] Haghighat, Maral & Rastegari, Hamid. (2013). A Review of Data Mining Techniques for Result Prediction in Sports. Advances in Computer Science: an International Journal. 2.