IE 582 Statistical Learning for Data Mining Project Report

Group-06

Yağmur Özdemir	2019402057
İrem Öztürk	2024702039
Gökcan Kahraman	2024705012

1.Introduction

1.1.Problem Description

In the realm of sports betting, the dynamic nature of in-game events and fluctuating odds introduces significant challenges for creating profitable strategies. A bettor's decision-making process must account for real-time changes in the game while ensuring that the decision is both timely and based on data available at the moment.

The goal of this project is to design a live betting strategy for matches that predicts the final outcome—home win, draw, or away win. The strategy must select one optimal time during the match to make a prediction and decide whether to bet or choose "no action." Importantly, the strategy must adhere to real-world constraints: once a decision is made, it cannot be revised, and future data cannot influence earlier decisions.

Using the provided dataset, which includes pre-match and live in-game statistics, as well as odds that reflect market beliefs, we aim to develop and test a robust betting strategy. Matches starting from November 1, 2024, are reserved as test data, ensuring that the strategy is evaluated on unseen matches. The primary challenge is to balance accuracy, profitability, and timeliness while adhering to practical betting constraints.

1.2.Descriptive Analysis of the Given Data

The dataset provided for this project is comprehensive, offering in-game data that captures the dynamic nature of these matches. The richness of the dataset lies in its ability to track changes in odds, match events, and outcomes over time, allowing for the development of a detailed and dynamic betting strategy. Below, we analyze the key components of the dataset:

Dynamic Nature of the Data

The dataset reflects the evolving dynamics of soccer matches, particularly through its time-stamped odds and match events. Key highlights include:

Dynamic Odds: The odds (1, 2, and X) fluctuate based on in-game events such as goals, red cards, and substitutions. These changes offer insights into market beliefs and the perceived likelihood of each outcome as the match progresses.

Event-Driven Changes: The suspended and stopped flags ensure that odds impacted by major events or technical issues are excluded, preserving data integrity.

The plot below illustrates the average odds trend for the three possible outcomes (home win, draw, and away win) over the course of a soccer match, measured in cumulative minutes.

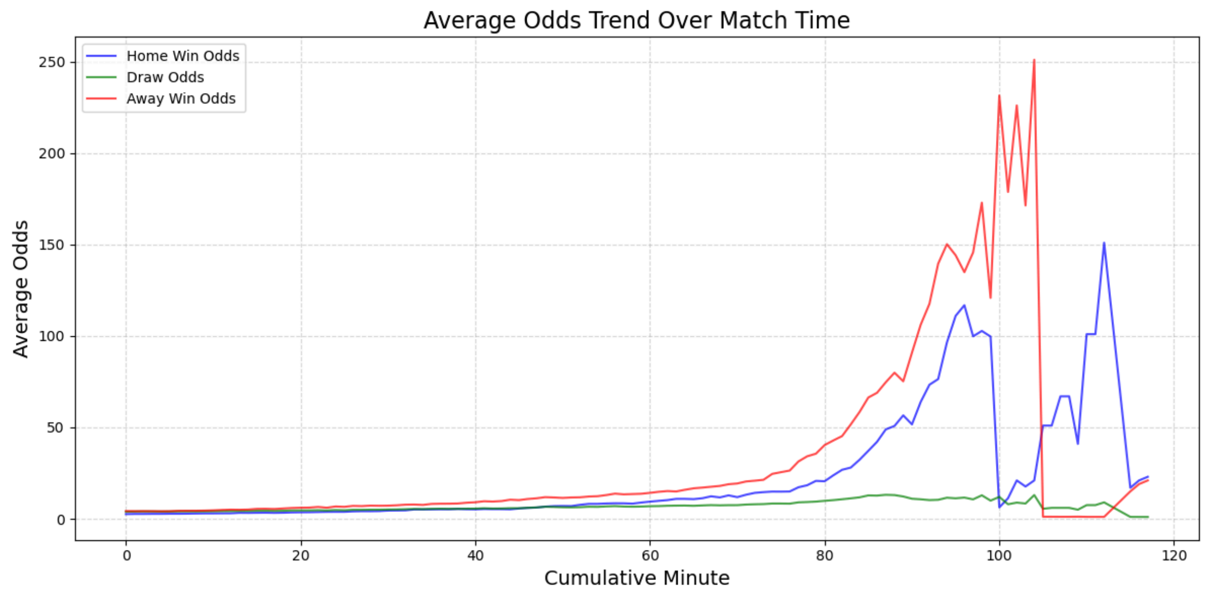


Figure 1. Average Odds Trend Over Match Time

Odds Behavior:

Odds for both home and away wins increase as the match progresses if the respective teams are not already in a strong position. This reflects the decreasing likelihood of a major shift in the match outcome as time runs out. Draw odds remain stable for most of the match, reflecting their neutral nature and likelihood across a wide range of game scenarios.

Market Dynamics:

The increasing odds toward the end of the match for home and away wins suggest a reactive adjustment by the market to the game's state. Draw odds' relative stability implies that the market perceives the possibility of a draw as less sensitive to in-game events compared to decisive wins and the volatility in odds during the final stages highlights the unpredictability of late-game dynamics and the market's rapid response to critical events.

Challenges in Using the Data

Challenges in using the data include handling cumulative minutes and dealing with market reactions and noise. Since the minute resets to 0 at the start of each half, cumulative match minutes must be calculated for consistency; for example, a row indicating "2nd-half" with a minute value of 5 represents the 50th minute of the match, requiring careful processing to account for these transitions. Additionally, the odds reflect market reactions to in-game events but may also include noise caused by speculative trading or minor events, making it essential to filter meaningful changes. Our analysis of the odds' distributions relative to match results revealed that all three plots are right-skewed, with most matches exhibiting relatively low odds for all outcomes. The clustering of odds at lower values suggests that bookmakers frequently assign higher probabilities to more likely outcomes, such as a home win, while reserving higher odds for less likely results, aligning with expected trends.

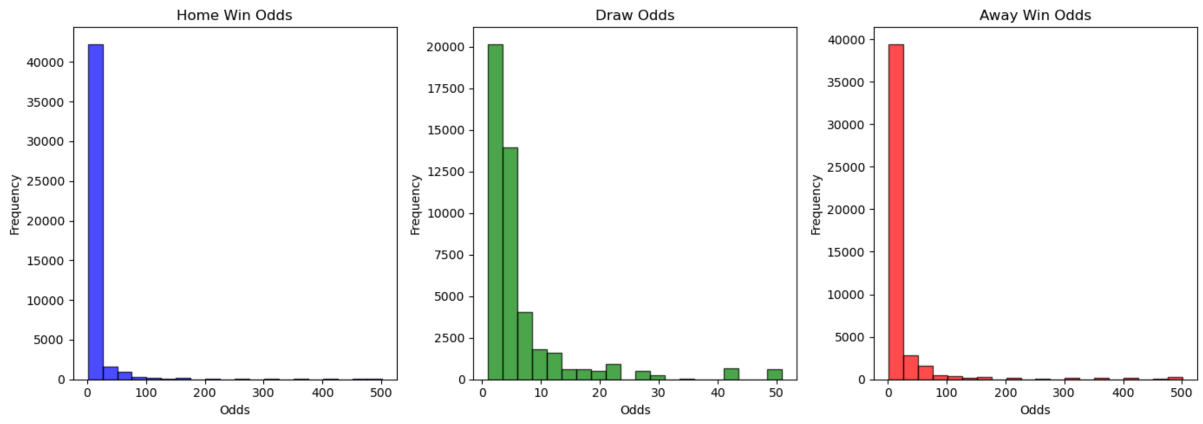


Figure 2. Histograms for odds divided into groups according to the results.

The data given has a different number of samples for each result. As the number of matches that finished with X, 1, or 2 are not equal, the imbalance of these classes may result in some inefficiencies for our model. (see details from the figure below) Maintaining balance of the class distribution is applied in the following model trials.

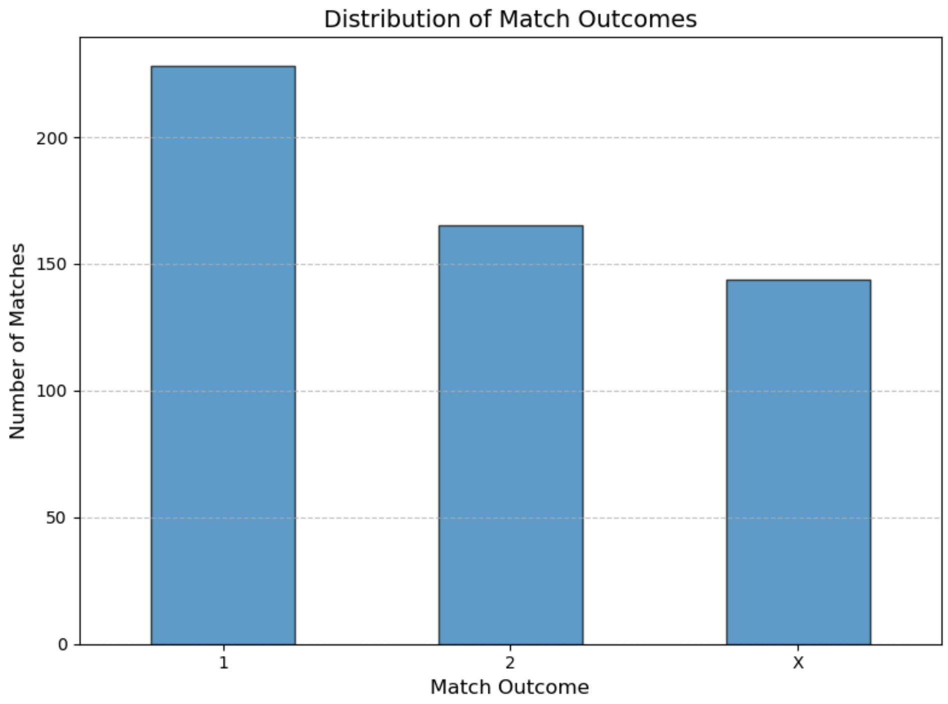
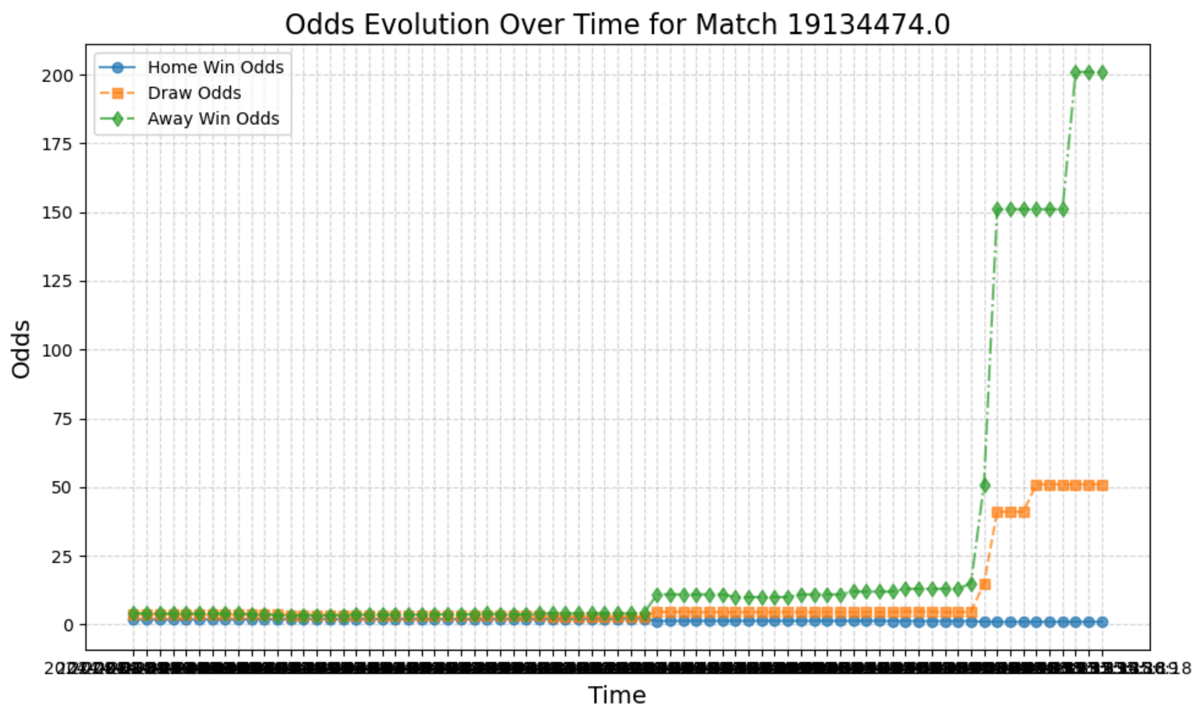


Figure 3. Distribution of the results (X, 1, or 2) with the related number of samples in each class

The dataset enforces the principle of only using data available at the time of decision-making. Features like final_score or odds updates that occur after a decision point must be excluded. Additionally, data on live-game events that are changing requires further analysis for the betting decision to be made. As can be seen from the example plots of some of the games below, it is not sufficient to make the betting decision, it is also crucial to analyze the changes in the odds of the game, and try to bet at the correct / more profitable times.



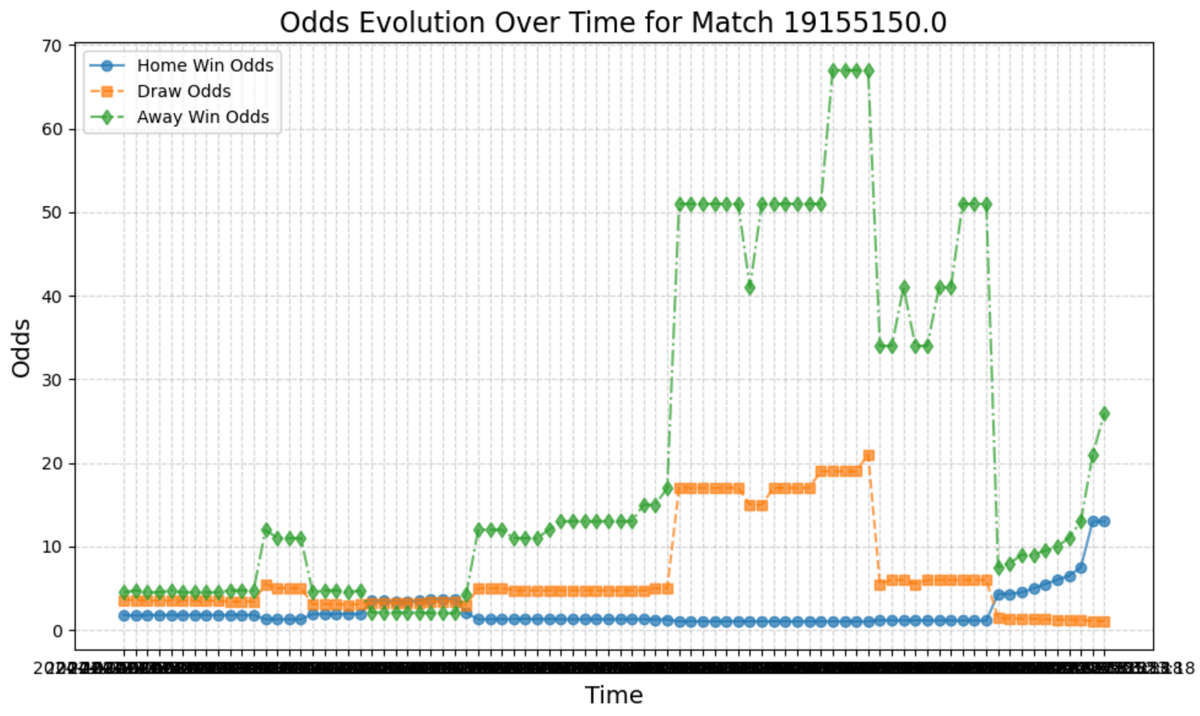


Figure 4. Odds Evaluation Over Time for spesific matches. These plots show that in-game odds are changing during the games resulting in more profitable periods to bet.

The dataset has several strengths, such as its comprehensive coverage, which includes many live in-game instances for each match. It also provides dynamic updates, capturing the real-time changes in match conditions and market beliefs. The granular time tracking offers minute-by-minute details, allowing for accurate decision-making. However, the dataset also has some limitations. One issue is inconsistent intervals, as not all matches have data for every minute or event, meaning some missing intervals need to be handled carefully. Additionally, the market-driven odds can be biased and may not always match actual probabilities, particularly in volatile matches. Features related to suspended and stopped events add another layer of complexity, requiring careful filtering and interpretation.

2. Literature Review

When we do a literature review of articles in the field of sports and betting, we come across these items that we need to pay attention to:

1) Machine Learning Models:

As a result of the articles we have researched, Random Forest or Gradient Boosting techniques are mainly used.[1][2][3] In these techniques, data from past matches are used to predict the data in the next match. We will apply a similar method in our model: After November 1, 2024 as a test set and before November 1, 2024 as a train set. Based on the results in the articles, Random Forest models are more successful than Gradient Boosting models in terms of both accuracy and sharpness.[1][2] Based on these researches, we decided to choose Random Forest as our main method. We also see that many papers use the cross-validation process for the selection of hyperparameters in forecasting models.

2) Preprocessing

In addition to filling the nan values in the data, preprocessing is also a very important element.[2]In the articles, they aim to perform a better analysis by converting discrete data into continuous datasets so that the models can analyze more successfully.[4] In addition, data is prepared with methods such as Label Encoding for better analysis within the model.[3] In addition, since there may be unbalanced match events in the dataset, class imbalances can be reduced and higher rates can be realized with techniques such as SMOTE. [1]

3) Important Features:

One of the most important features of the papers was to make the model work better by identifying the most important features.[1] The most basic feature of these models was the goal.[3] In our model, we will add various metrics such as goals-assists to our model and we decided to add new columns according to the standard deviation and averages according to the last five minutes based on the claim rates in our data and the probability of the model.

4) Model Evaluation:

The performance of the models is evaluated with various metrics such as accuracy, precision, recall and ROC curves.[2] Model performance is also analyzed using confusion matrices.[3] In order to analyze the results of our model in a good way, we decided to run both the confusion matrix and metrics such as precision and recall with our code.

5) Optimal Decision Making:

Optimal decision making in sports betting is most fundamentally related to the probability distribution of the outcome of the match and the bookmaker's offer.[1] The probability of winning is the most important metric that determines the basic decisions of the bettors, and from this the bet size can be optimized.[4] According to the situation in the articles, it is sufficient to know the median result for optimal prediction in a match. [2]

However, additional quantiles are needed to select the matches to bet on. [1] In particular, moneyline betting requires knowledge of different quantiles according to the payout odds.[3] It is not possible for us to know such candles as moneyline only due to the availability of data and betting odds. Therefore, in order to use this candle, we have converted the probability in our model to the betting odds to determine the moneyline.

6) Decision Trees

Decision trees are a popular machine learning algorithm used in both classification and regression tasks.[2] These algorithms represent data in a tree-like structure, with each internal node performing a test on a feature. Each branch represents a result of the test, while leaf nodes contain predictions or class labels. Thanks to this structure, it is possible to combine multiple decision trees together to create a more powerful prediction model with the Random Forest (RF) model.[3] RF trains each tree with a random subset of features and data points and avoids overfitting. This allows it to handle both large and incomplete datasets well.

7) General Conclusions and Future Work:

From the information in the papers and their results, it is clear that machine learning plays an important role in predicting soccer match results and making better decisions in sports betting.[1] In order to realize this importance, data quality and the features used significantly affect the success of the model.[3] In terms of data quality, null values are filled with values such as average and integer values and columns are used to reinforce the learning of the model.[2] Ensemble approaches, which are created by combining different machine learning algorithms, provide more robust and successful results. Since we are currently reading articles claiming about football, we have seen that future work in the article will aim to use machine learning in different sports (such as rugby, American football, basketball, tennis, golf, etc.) and to add more advanced features (e.g. player ratings, team defense/attack ratios, social media data, etc.) to the model. [4] Over time, it is predicted that both the changing models and the new methods to be discovered can increase the prediction rates much more.

3.Match Result Prediction Models & Betting Approaches

3.1.Summary of the Proposed Approach

We first developed a benchmark model as the foundational strategy. This model checks the odds and game state at fixed time intervals (15th, 30th, 45th, 60th, 75th, and 90th minutes). For each match, the model evaluates whether to place a bet at a given time instant, or to delay the decision until a later time interval if "no action" is chosen. Importantly, the model ensures no leakage of future information by making decisions based only on data available up to the current interval. This approach simulates real-world betting constraints.

The benchmark model relies on decision trees trained for each fixed interval to classify the match outcome. For each match, firstly, at each interval, the model checks the decision tree's output and evaluates the impurity (Gini score) of the decision. Then, a bet is placed if the tree identifies a clear advantage (Gini < 0.35). Otherwise, the model defers the decision to the next interval. Finally, if no betting decision is made by the 90th minute, the strategy defaults to "no action." While effective as a baseline, the benchmark model has notable limitations, including reliance on single decision trees and a static interval-based approach that ignores potentially important real-time dynamics.

We propose an enhanced strategy using random forest models to address these shortcomings. By leveraging the ensemble nature of random forests, the proposed strategy evaluates every game event (rather than fixed intervals) to dynamically decide whether to bet or wait for a more profitable opportunity. This approach incorporates a betting strategy that finds shortcomings in bookmakers' odds. It applies certain thresholds to decide whether to bet or wait at any given timestamp. It is a more general approach that breaks the limit of the specific betting time of the benchmark model.

3.2.Pre-Processing Steps on the Data

In this project, several pre-processing steps have been applied to the data file. Rows where the 'suspended' or 'stopped' columns are marked as 'True' have been removed. To handle missing values, the data is grouped by fixture_id, and the matches are sorted by time. Missing values are filled with information from previous events. Any remaining missing values are filled with zeros. Lastly, categorical columns like 'result' and 'current_state' are converted into integer values for the models, with 'Home win' as 1, 'Draw' as 0, and 'Away win' as 2. The data we use in this project has already been pre-processed in this way.

3.3. Benchmark Model

The benchmark model serves as a foundational strategy to evaluate the feasibility of a fixed-interval live betting system. The model combines decision trees with structured decision-making logic to make one bet per match, adhering to the constraints of real-world betting scenarios.

Objective

The benchmark model predicts match outcomes—home win, draw, or away win—at specific times during a match (15th, 30th, 45th, 60th, and 75th minutes). For each match, firstly, the model evaluates whether to place a bet or defer the decision until a subsequent interval. Intervals are defined as the nearest observation to the target minutes.Then, a bet is made only if a high-confidence prediction is identified at the current interval. Finally, if no decision is made by the 90th minute, the model defaults to "no action."

This approach ensures that the model adheres to the real-world constraint of making a single, irreversible decision per match without using future information.

Model Components

The benchmark model employs a separate decision tree for each interval (e.g., 15th, 30th minutes), where each tree is trained on pre-match and in-game data available at that interval to predict the final match outcome (home win, draw, or away win). Decision trees were chosen for their interpretability and their ability to handle both categorical and numerical features effectively. Feature engineering involves calculating cumulative minutes to address the reset at halftime, incorporating odds (1, X, 2) that reflect market beliefs and dynamically change with match events, and including in-game features such as the current match state, team names, and updated odds. Additionally, normalized probabilities derived from odds are added as features.

At each interval, the decision tree predicts the match outcome and calculates the Gini impurity for each decision node. If the Gini impurity is below a set threshold (≤ 0.2 in this case), indicating high confidence in the prediction, a bet is placed. Otherwise, the decision is deferred to the next interval, with the process continuing until a bet is made or the match concludes. If no bet is made by the 90th minute, the decision is recorded as "no action," ensuring that matches with consistently low confidence are excluded from betting.

The model training involves fitting separate decision trees for intervals like the 15th, 30th, 45th, 60th, and 75th minutes using features such as match-level statistics, odds, cumulative minutes, and normalized probabilities, with the target being the final match result (home win, draw, or away win). During the interval-based decision process, the model iterates through the intervals for each match in the test dataset, applying the corresponding decision tree. Matches without a decision at an interval are carried forward to the next one, ensuring no future data is used while refining predictions. At the end of the evaluation, each match results in either a single decision, such as placing a bet at a specific interval, or "no action" if no confident decision was reached.

Decision Trees Created for Specific Times in Matches

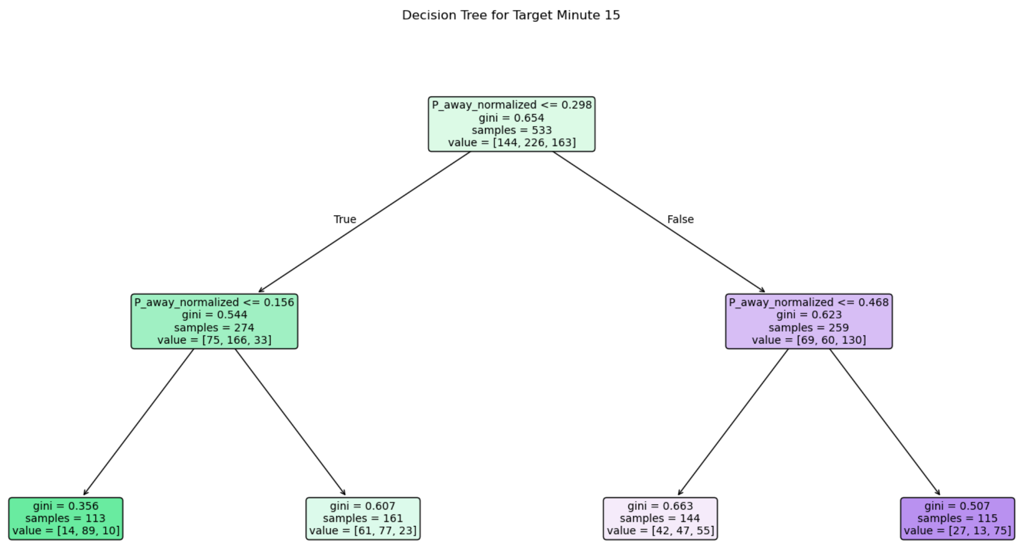


Figure 5.Decision Tree for Match Minute 15

Evaluation Results for Target Minute 15:
Accuracy: 0.53
Precision: 0.44
Recall: 0.53

F1 Score: 0.47

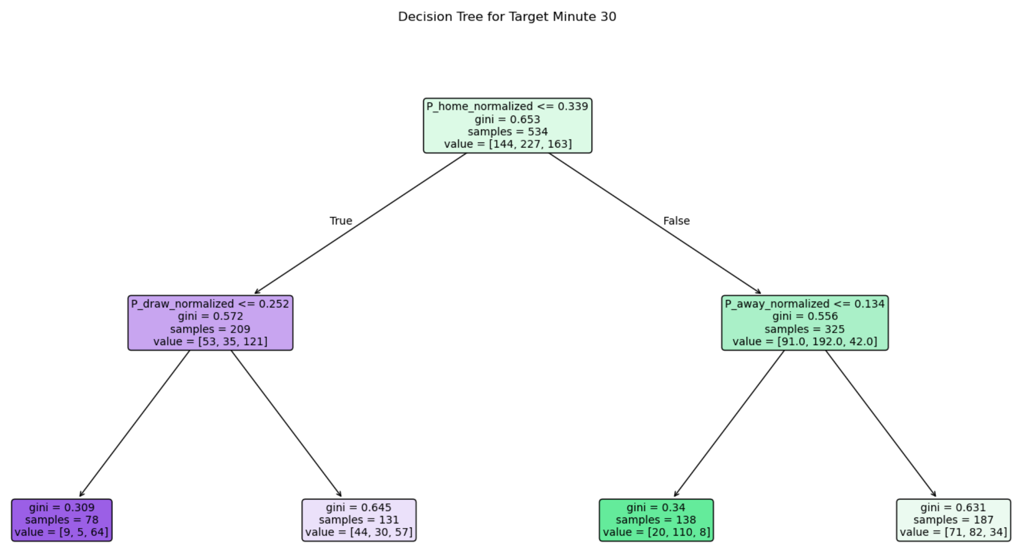


Figure 6.Decision Tree for Match Minute 30

Evaluation Results for Target Minute 30:
Accuracy: 0.58
Precision: 0.46
Recall: 0.58
F1 Score: 0.51

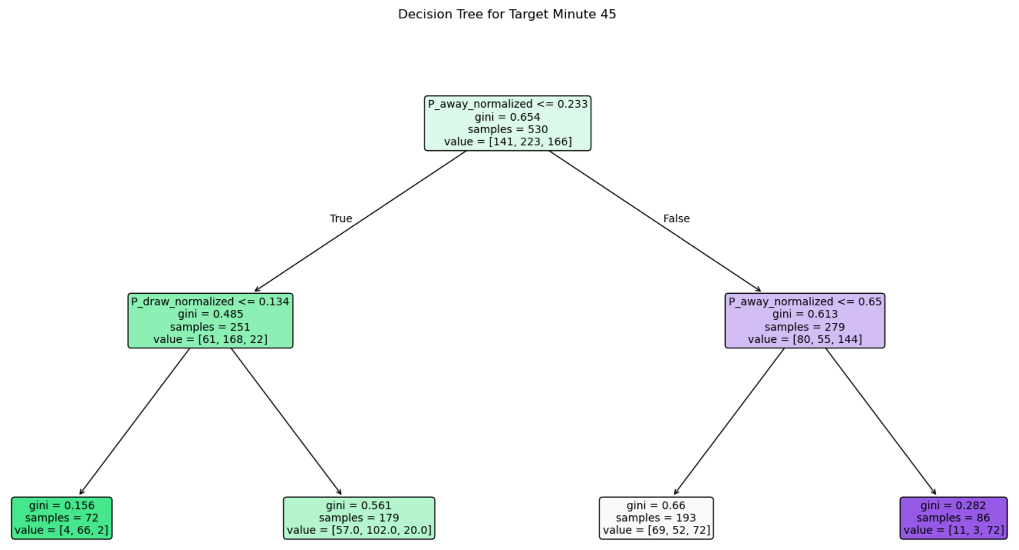


Figure 7.Decision Tree for Match Minute 45

*** Last two trees for the 60th and 75th minutes can be found in the appendix.**

Evaluation Results for Target Minute 45:
Accuracy: 0.58
Precision: 0.52
Recall: 0.58

F1 Score: 0.53

Each decision tree is trained to predict the match outcome (home win, draw, or away win) at a specific point in time (e.g., 15th, 30th, 45th minute; if an event on these exact minutes is not present in data for a match, the closest time event is chosen).

The trees generally use normalized probabilities (P_home_normalized, P_draw_normalized, P_away_normalized) as primary features for splitting, which represent the market-implied likelihoods of each outcome. The trees are chosen as relatively shallow (depth of 2), which ensures interpretability although this may limit their ability to capture complex interactions. The depth selection is based on the interpretability aspect for this basic benchmark model.
When max_depth is selected as 2, the splitting decisions at each node are based on thresholds of normalized probabilities, providing clear and straightforward decision rules. Then, the Gini impurity is used to evaluate the "purity" of the splits. Higher Gini values indicate more uncertainty in predictions, while lower values indicate more confident predictions. Nodes with lower Gini scores (e.g., leaves) are more reliable for decision-making.

Hence, our decision making process chooses to bet on the instances where the node it falls into shows low Gini scores, indicating more homogenous classes. After the 15th minute of each match is put into the related tree, the samples which end up in nodes where Gini is higher than our threshold, is chosen as ‘no action’ for the 15th minute, then put into the 30th minute tree with related instances.

The benchmark model was tested on 111 matches. Its performance metrics are summarized below:

- **Total Profit**: 37.43 units
- **Total Bets Made**: 60
- **Total No Actions**: 51
- **Accuracy of Bets**: 86.67%

The benchmark model, which evaluates specific intervals during a match and relies on a single decision tree per interval to make betting decisions, represents a straightforward and interpretable approach. By restricting decisions to predefined minutes and ensuring no future data leakage, it provides a simple yet effective baseline for developing live betting strategies. The model's profitability and adherence to real-world constraints demonstrate its utility as a foundational framework.

However, despite its strengths, this approach has significant limitations. The reliance on a single decision tree introduces high bias, as evidenced by the model's bias score of 0.7055. This overconfidence in predictions, coupled with moderate accuracy (61.04%), highlights the risk of underperformance in cases where match dynamics deviate from the training data. Furthermore, the static nature of interval-based evaluation prevents the model from reacting to critical in-game events that occur between predefined time points, potentially missing opportunities to optimize betting decisions.

To address these shortcomings, a more dynamic and robust model is necessary. Specifically, an approach that evaluates not only the decision to bet at the current moment but also the potential benefit of waiting for subsequent events is required. By accounting for the possibility of improved odds and higher profits through delayed decisions, a refined strategy can better align with the fluid nature of in-game dynamics.

To achieve this, we propose two different model strategies: **the** logistic regression model and the random forest model as an enhancement. These models both evaluate every event in the match, dynamically assess the predicted probabilities and related odds, and incorporate the potential for better assessing the more profitable decisions.

3.4.Logistic Regression Model

Before the bettings are decided, the general accuracy of the proposed models on predicting the match results(1: home win, 0: draw, 2: away win) is evaluated. In the following models, whole match data is used to predict the match result rather than the rows of a specific minute like in the benchmark model.

After the preprocessing steps mentioned above, and similar feature engineering steps (encoding of the categorical variables, normalized probability calculations, etc.), the data is divided into train, validation and test sets. Also, new features, like the mean and standard deviation of bets in the previous 5 minutes of each row, are added to the data. By adding these features, capturing the sudden shifts in a dynamic game is aimed.

Test set start date = '2024-11-01'
Validation set start date = '2024-10-20'

Number of matches in validation period: 92
Number of matches in test period: 111
Number of matches in train period: 445

The comparison of predictive performances of proposed models and the benchmark model is hard to capture because the benchmark model only uses data of predefined minutes. Hence, a simple decision tree is built that will learn the whole events rather than predefined minutes. The simple decision tree resulted in a validation accuracy of %54 and a test accuracy of %56. This simple decision tree approach shows that using each event rather than specific minutes, may result in a more comprehensive model.

Hence, a Logistic Regression model was implemented, building upon the features and preprocessing steps applied to the dataset. Logistic Regression was chosen for its simplicity, interpretability, and ability to handle multi-class classification tasks, making it a suitable choice for predicting match outcomes (1: home win, 0: draw, 2: away win). Also, the continuity of the results in the logistic regression model seemed to be a good fit for evaluating each event.

Data Preparation for Logistic Regression

Following the preprocessing and feature engineering steps:

The dataset was divided into training, validation, and test sets with the same split criteria:

Train Period: Matches before 2024-10-20 (445 matches)
Validation Period: Matches from 2024-10-20 to 2024-10-31 (92 matches)
Test Period: Matches starting from 2024-11-01 (111 matches)

Features such as normalized probabilities (P_home_normalized, P_draw_normalized, P_away_normalized) and new dynamic features (e.g., mean and standard deviation of odds over the last 5 minutes) were included to capture the temporal dynamics of odds changes.

Scaling and Parameter Tuning

The features were standardized using StandardScaler to ensure all input features had a similar scale, improving the optimization process during training. A grid search was conducted to optimize the parameters, including:

- **Solver:** Algorithm used for optimization (lbfgs, saga).
- **Regularization Strength (C):** Values of 0.1, 1.0, and 10.0 were tested to control overfitting.
- **Maximum Iterations:** Higher limits (1000, 2000, 3000) were used to ensure model convergence.

Model Training and Predictions

The final Logistic Regression model was trained using the lbfgs solver, a regularization strength of C=0.1, and a maximum of 1000 iterations, based on the results from the grid search. Predictions were made for the training, validation, and test sets, with the model outputting class probabilities for each possible match outcome (home win, draw, away win). These probabilities, namely win_probability_1_logis_reg, win_probability_X_logis_reg, and win_probability_2_logis_reg, represent the likelihood of each outcome based on the input features. The probabilities were then added to the dataset for the training, validation, and test sets for further analysis.

Prediction Rate: 61.91%
Prediction Rate in Validation: 58.26%

Model Performance Metrics:

Class	Precision	Recall	F1-score
0	0.363043	0.402642	0.381818
1	0.746410	0.713125	0.729388
2	0.616645	0.614832	0.615737

When we look at the results, Class 1 is the class in which the model performs best. Both precision and recall are high. On the other hand, Class 0 is the class in which the model performs the worst. Precision, recall and therefore F1-score are quite low. It can be said that the model has difficulty recognizing Class 0.
Compared to the other two models, Class 2 has average performance.
When we look at these results, we can say that we should pay attention to class weights. Since we do not get very good results in Class 0, we can choose models that know it well. Of course, we can further increase these results by performing hyperparameter optimization while building the other model.

3.5.Random Forest Model

Preprocessing and Balancing:

We used the same preprocessing methods as Logistic Regression to better compare our Random Forest model with the logistic regression model. In this way, we would be able to compare the results by examining metrics such as accuracy and precision-recall in a much better way. However, apart from this, we added a few small methods to enable the model to be analyzed better:

We also took actions to eliminate class imbalance to avoid imbalance in certain parameters and results within the model. To do this, we first identified what the imbalances in the classroom might be. As a result, we synthetically added examples of the minority class with the SMOTE code, thus minimizing the imbalance and preparing the preprocessing.

Hyperparameter Optimization:

Of course, the most important reason why we used hyperparameter optimization was to ensure that the model chooses the parameters that will give the best performance. To do this, we used GridSearchCV and tested many combinations of hyperparameters with it:

- n_estimators: [500, 1000, 1500]
- max_depth: [15, 20]
- min_samples_split: [5, 10, 15]
- min_samples_leaf: [4, 6]

Here, we applied cross-validation to determine the most accurate parameter and applied it in 5 layers. This way we chose what the best parameters were. To calculate predictions and probabilities, the model was run on training, validation and test sets and probabilities were calculated for each class.

After the grid search best parameter set is decided as follows:
n_estimators: 500
max_depth: 20
min_sample_split: 5
min_samples_leaf: 4

Evaluating Model Performances:
We applied widely used methods to evaluate model performance among each other and these are:
Accuracy Rate: The accuracy metric was calculated and compared in the training and validation sets.
F1 Score: Used to measure balanced performance.

Test Set Prediction Rate: 64.29%
Prediction Rate in Validation: 63.96%

Class Distribution in Train Set:

```
1    0.423456
2    0.300708
0    0.275835
```

```
Test Set Model Model Performance Metrics:
Class  Precision   Recall  F1-score
   0   0.417528 0.430337  0.423836
   1   0.799876 0.701390  0.747403
   2   0.583765 0.700098  0.636661
```

```
Validation Set Model Performance Metrics:
Class  Precision   Recall  F1-score
   0   0.426326 0.417709  0.421974
   1   0.735064 0.684964  0.709130
   2   0.666570 0.731248  0.697412
```

When we compare all the model performances mentioned above, we see that our Random Forest model is better. It would be the most natural choice to choose this model, which is ahead in every parameter and every class. So, further decision-making on betting is made based on the predictions of the random forest model.

3.6. Betting Strategy Using Predicted Probabilities from Random Forest

The predicted probabilities generated by the Random Forest model (win_probability_1, win_probability_X, win_probability_2) represent the likelihood of each potential match outcome (home win, draw, away win). These probabilities are directly used to inform and guide the decision-making process in the following way:

1. Identifying the Most Likely Outcome

For each match, the outcome with the highest predicted probability among the three categories is selected as the predicted result:

Home Win (1) if win_probability_1 is the highest.
Draw (0) if win_probability_X is the highest.
Away Win (2) if win_probability_2 is the highest.

This ensures that the decision is driven by the model's confidence in the likelihood of each outcome, as determined by the input features and tuned hyperparameters of random forest.

2. Mapping Probabilities to Betting Odds

Each match has associated odds provided for the three outcomes (1, X, and 2). These odds represent the prediction of the multiplier for a bet placed on each outcome. The predicted probabilities are compared to the associated odds to assess the risk-to-reward ratio for each possible bet:

A high probability prediction for an outcome suggests greater confidence, which, when paired with favorable odds, creates an opportunity for a profitable bet. For example, If **win_probability_1**= 0.75 (75% confidence for home win) and the odds for home win (1) are 2.0, a bet on the home team would be considered profitable.

3. Decision Rules

The decision-making process is based on a few rules. A bet is placed on the outcome with the highest predicted probability, as long as the odds for that outcome make the bet worthwhile. For example, if the model predicts a draw with a probability of 0.6 and the odds for a draw are 3.0, the decision would be to bet on a draw. If the model's highest predicted probability is below a certain threshold, such as 0.5, the strategy will be to take no action. This helps avoid placing bets when the model lacks confidence, reducing unnecessary risk. Additionally, only one decision is made per match, and once a decision is made, no further analysis or changes are allowed for that match.

4. Integration into the Betting Strategy

The predicted probabilities, combined with the odds for each outcome, form the foundation of the betting strategy. The decision-making process evaluates bets based on three key criteria:

a. **Probability Threshold**:

For each match, the predicted probabilities (win_probability_1, win_probability_X, win_probability_2) are analyzed. A bet is considered only if the highest predicted probability exceeds a predefined threshold (e.g., 0.6), ensuring that only high-confidence predictions are acted upon.

b. **Odds Threshold**:

The betting odds for each outcome (1 for home win, X for draw, 2 for away win) are evaluated. A bet is placed only if the odds for the predicted outcome exceed a predefined threshold, ensuring a favorable risk-to-reward ratio.

c. **Profitability Comparison**:

The predicted probabilities are used to calculate an implied probability-based odds value (1 / win_probability). A bet is considered only if the bookmaker’s offered odds are more favorable (higher) than the implied odds calculated from the model’s probabilities. This condition ensures that bets are only placed when the bookmaker offers a payout higher than the model’s implied fair value, indicating a potential profit opportunity.

5. Bet Placement

If all three criteria (probability threshold, odds threshold, and profitability comparison) are satisfied, a bet is placed on the outcome with the highest predicted probability. This systematic approach ensures that the betting decisions are informed, data-driven, and aligned with favorable market conditions.

Analysis of Forecast Results and Betting Decision:

Before explaining our claim algorithm, if you want to look at what the model results are and the confusion matrix section, you can examine where they are distributed in the results section. If you want to specifically observe the effects per minute and look at the probability and bet distributions in a match as an example, you can look at the appendix section.

To convert prediction results into betting decisions:
We were looking at three different parameters to determine our betting decision as explained earlier in detail:
1) In each line, we take the maximum probability of a side winning that we calculated with the random forest model. If this maximum is greater than the threshold we set, it passes the first test.
2) If the bet rates determined for the lines we choose are greater than our bet thresholds, we still can make a choice.
3) Finally, we convert the probability that comes from the random forest model into the bet rate by 1 over it. If the rate of the random forest model is less than the bet rate determined by the bookmakers, we complete our selection. This allows us to bet on a greater probability of winning than what we think is worth it.

Threshold Value and Probability Ranges:
To determine the best model confidence threshold(probability that comes from the random forest) and bet probability threshold (1/bet by bookmakers) following ranges are applied to the validation set to overcome data leakage. After determining the best pair in the validation set, this pair is applied to the test set to mimic a real-time scenario. Further comparisons of the results and betting outcomes are discussed in the results section.

Threshold values used: [0.33, 0.34, ..., 0.45]
Probability ranges: [1.8, 1.9, ..., 2.5]

Scoring Mechanism:
Points for predicting a draw: row['X'] - 1
Points for home win prediction: row['1'] - 1
Points for away win prediction: row['2'] - 1
Loss: -1 point.
Thanks to this betting strategy and win-lose state, we completed our coding selected which matches we should play, and added our results to the results accordingly.

4.Results

4.1. Match Result Prediction Results

Before writing our results for evaluation at the end of the project, let's look at the total number of matches in the train and test sets and what the match contents are:
Total matches in validation set: 92
Total matches in test set: 111

Total matches in train set: 445

To judge which model is better, we show the results of each different algorithm we wrote in the code:

Logistic Regression:
Prediction Rate: 61.91%
Prediction Rate in Validation: 58.26%

```
Model Performance Metrics:
Class  Precision   Recall  F1-score
   0   0.363043 0.402642  0.381818
   1   0.746410 0.713125  0.729388
   2   0.616645 0.614832  0.615737
```

Random Forest:

Test Set Prediction Rate: 64.29%
Prediction Rate in Validation: 63.96%

```
Class Distribution in Train Set:
1    0.423456
2    0.300708
0    0.275835
```

```
Test Set Model Model Performance Metrics:
Class  Precision   Recall  F1-score
   0   0.417528 0.430337  0.423836
   1   0.799876 0.701390  0.747403
   2   0.583765 0.700098  0.636661
```

When we look at the accuracy percentage between Logistic Regression and Random Forest, we see that Random Forest makes better predictions both in validation and other sets. Even if we consider that there are minor differences in the predictions, it is possible to say that Random Forest performs better in precision, recall, and f1 scores according to each class. However, although Random Forest is better than the other, it is difficult to say that it makes a successful prediction in general. When we look at figures 8 and 9, we see that it is especially difficult to predict the matches that the home team will win.

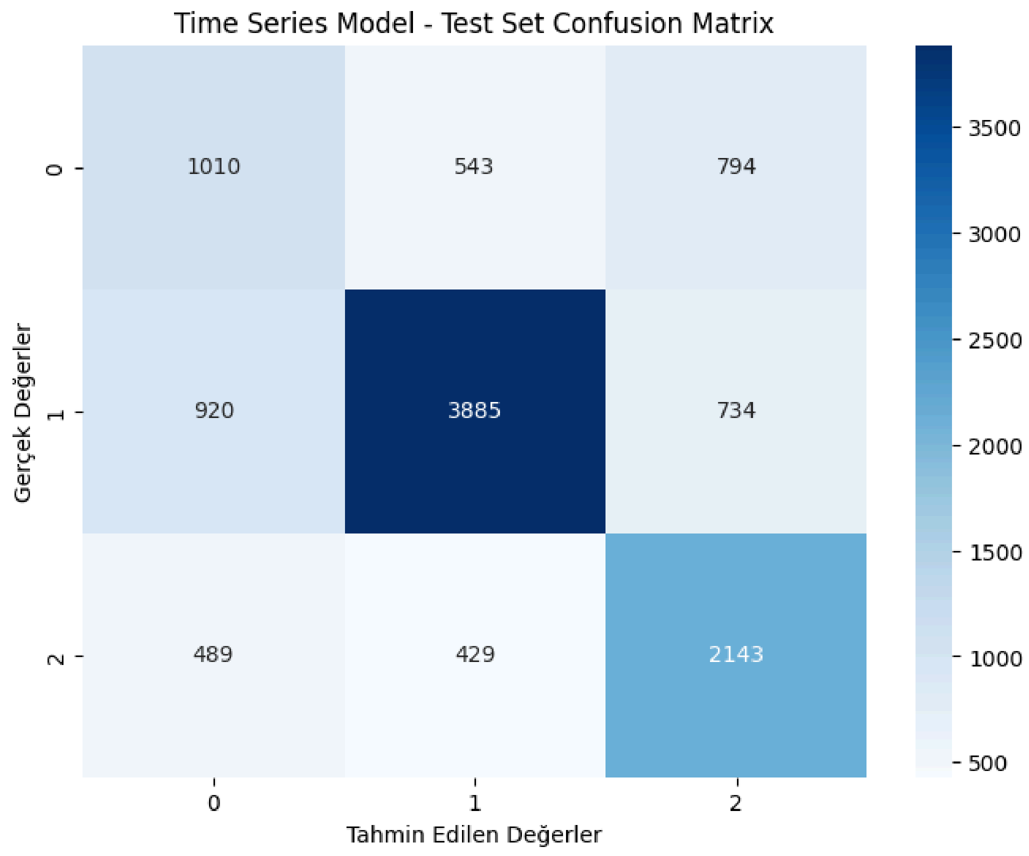


Figure 8.Predicted and Actual Classes of the Test Set

```
Validation Set Model Performance Metrics:
Class  Precision   Recall  F1-score
   0   0.426326 0.417709  0.421974
   1   0.735064 0.684964  0.709130
   2   0.666570 0.731248  0.697412
```

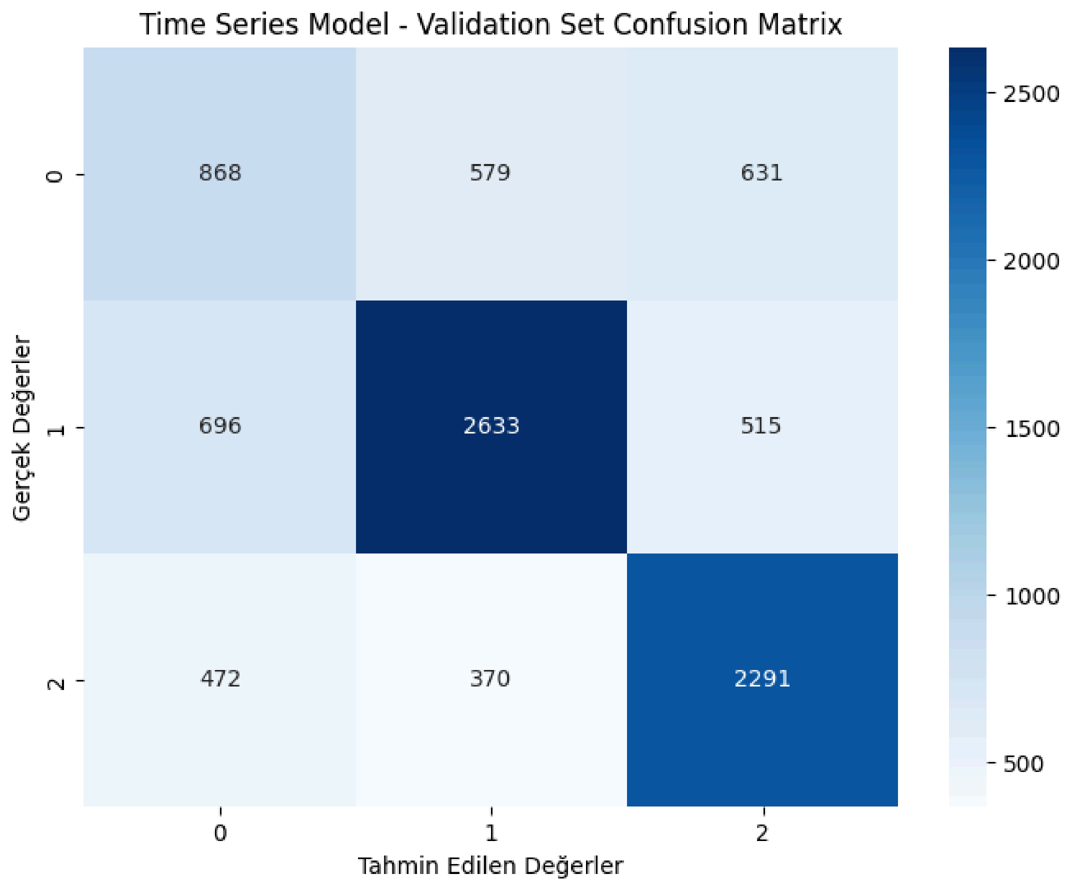


Figure 9.Predicted and Actual Classes of the Validation Set

Feature Importance:

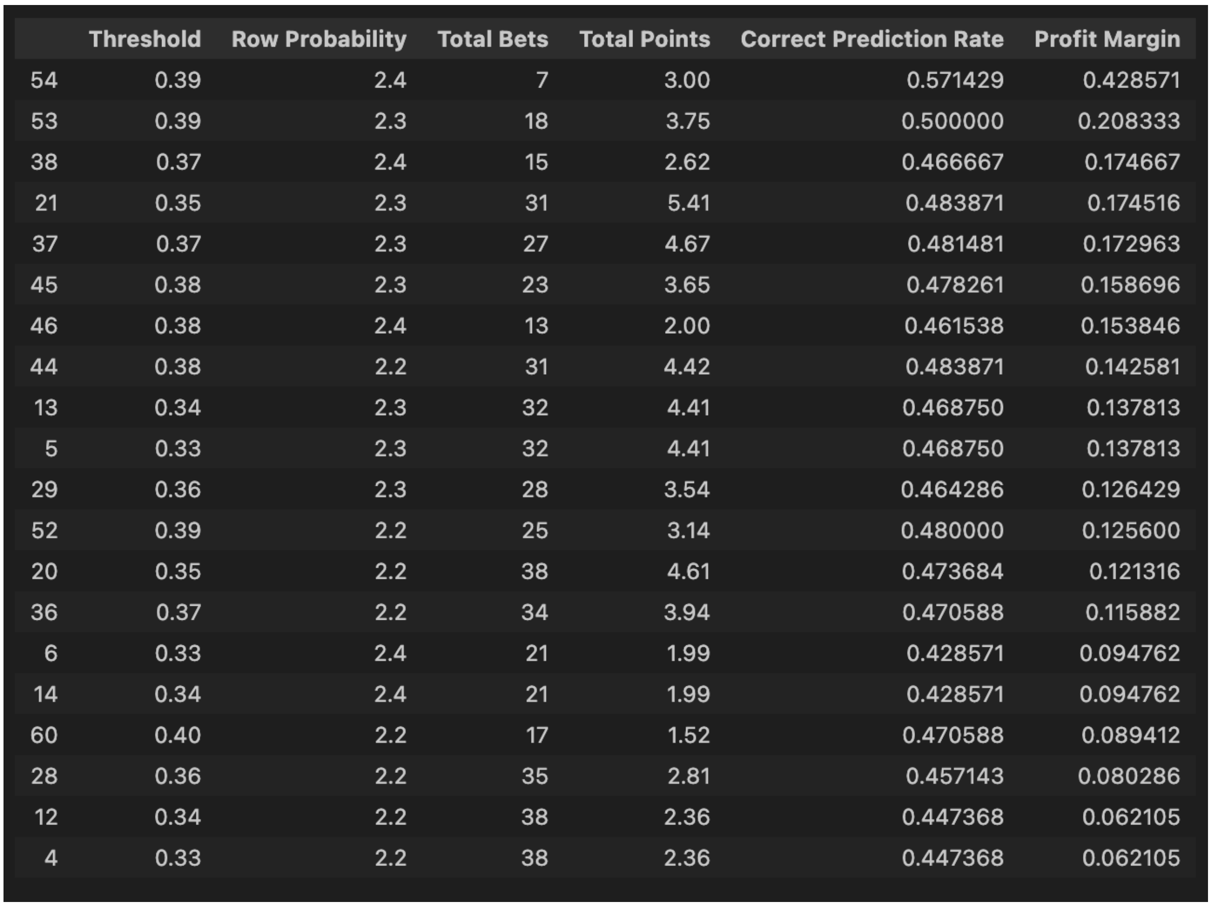
P_away_norm	0.045956
P_home_norm	0.042057
1	0.039627
P_away	0.035815
2	0.033244
P_home	0.031039
last_5_min_avg_1	0.030210
last_5_min_avg_2	0.029256
P_draw_norm	0.025206
P_draw	0.021635

Also, additional features derived from the data like last_5_minute_average bet ratios and the normalized bet probabilities are among the most important features. When we analyze the table that examines the prediction-result combinations for each minute(heat map in the appendix), we notice that the draw and home odds become better separated, particularly after the 70th-75th minutes. This might have occurred because the home team widened the gap during the match. Therefore, it can be concluded that the Random Forest model struggled to predict events in the first half, which explains why its accuracy remained at a certain level.

4.2. Betting Results

Betting is decided based on the proposed approach in the logistic regression and random forest section. This approach is built on the validation set.

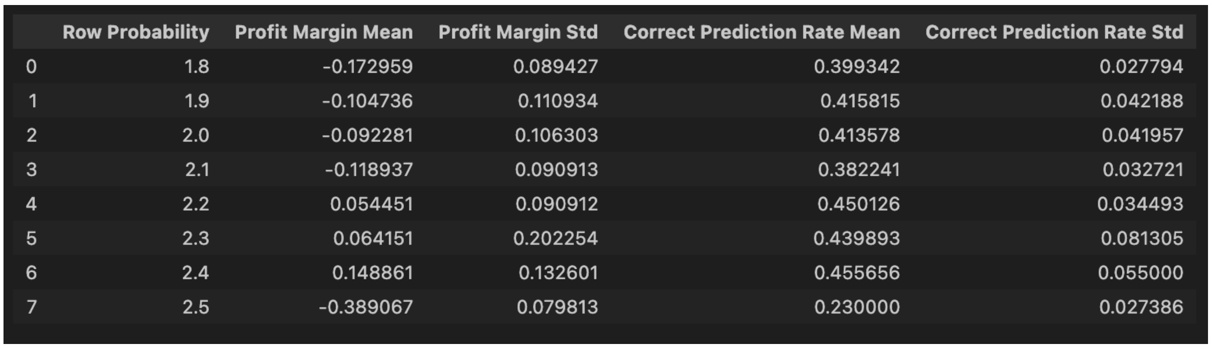
Validation Set Results based on Profit Margin:



	Threshold	Row Probability	Total Bets	Total Points	Correct Prediction Rate	Profit Margin
54	0.39	2.4	7	3.00	0.571429	0.428571
53	0.39	2.3	18	3.75	0.500000	0.208333
38	0.37	2.4	15	2.62	0.466667	0.174667
21	0.35	2.3	31	5.41	0.483871	0.174516
37	0.37	2.3	27	4.67	0.481481	0.172963
45	0.38	2.3	23	3.65	0.478261	0.158696
46	0.38	2.4	13	2.00	0.461538	0.153846
44	0.38	2.2	31	4.42	0.483871	0.142581
13	0.34	2.3	32	4.41	0.468750	0.137813
5	0.33	2.3	32	4.41	0.468750	0.137813
29	0.36	2.3	28	3.54	0.464286	0.126429
52	0.39	2.2	25	3.14	0.480000	0.125600
20	0.35	2.2	38	4.61	0.473684	0.121316
36	0.37	2.2	34	3.94	0.470588	0.115882
6	0.33	2.4	21	1.99	0.428571	0.094762
14	0.34	2.4	21	1.99	0.428571	0.094762
60	0.40	2.2	17	1.52	0.470588	0.089412
28	0.36	2.2	35	2.81	0.457143	0.080286
12	0.34	2.2	38	2.36	0.447368	0.062105
4	0.33	2.2	38	2.36	0.447368	0.062105

Figure 10. Performance of the threshold pairs of prediction probabilities and bet odds in the validation set

Row Probabilities Distribution:

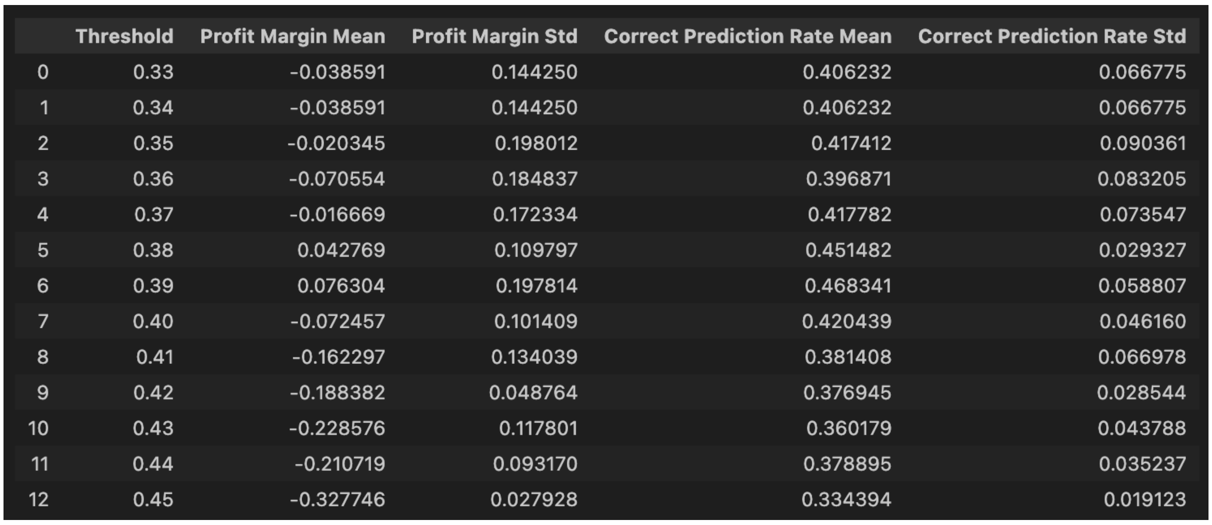


	Row Probability	Profit Margin Mean	Profit Margin Std	Correct Prediction Rate Mean	Correct Prediction Rate Std
0	1.8	-0.172959	0.089427	0.399342	0.027794
1	1.9	-0.104736	0.110934	0.415815	0.042188
2	2.0	-0.092281	0.106303	0.413578	0.041957
3	2.1	-0.118937	0.090913	0.382241	0.032721
4	2.2	0.054451	0.090912	0.450126	0.034493
5	2.3	0.064151	0.202254	0.439893	0.081305
6	2.4	0.148861	0.132601	0.455656	0.055000
7	2.5	-0.389067	0.079813	0.230000	0.027386

Figure 11. Betting Odd distributions in the validation set

Looking at the averages and standard deviations of the profit and total income applied to the validation set, Figure 11, at first glance, 2.4 appears to be the best value for row probability based on the correct prediction rate and profit margin. However, its high standard deviation indicates a significant risk. To balance the correct prediction rate and profit margin, 2.2 emerges as the optimal choice.

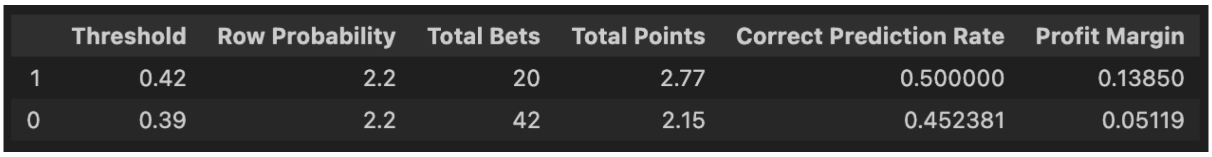
Thresholds Distribution:



	Threshold	Profit Margin Mean	Profit Margin Std	Correct Prediction Rate Mean	Correct Prediction Rate Std
0	0.33	-0.038591	0.144250	0.406232	0.066775
1	0.34	-0.038591	0.144250	0.406232	0.066775
2	0.35	-0.020345	0.198012	0.417412	0.090361
3	0.36	-0.070554	0.184837	0.396871	0.083205
4	0.37	-0.016669	0.172334	0.417782	0.073547
5	0.38	0.042769	0.109797	0.451482	0.029327
6	0.39	0.076304	0.197814	0.468341	0.058807
7	0.40	-0.072457	0.101409	0.420439	0.046160
8	0.41	-0.162297	0.134039	0.381408	0.066978
9	0.42	-0.188382	0.048764	0.376945	0.028544
10	0.43	-0.228576	0.117801	0.360179	0.043788
11	0.44	-0.210719	0.093170	0.378895	0.035237
12	0.45	-0.327746	0.027928	0.334394	0.019123

Figure 12. Threshold (random forest model predicted class probabilities) distributions in the validation set

Test Results based on Best Threshold & Bet Odds Pair:



	Threshold	Row Probability	Total Bets	Total Points	Correct Prediction Rate	Profit Margin
1	0.42	2.2	20	2.77	0.500000	0.13850
0	0.39	2.2	42	2.15	0.452381	0.05119

Figure 13. Selected pair performance in the test set

When we turn to the Figure 12., 0.39 and 0.38 stand out due to their positive profit margins and significantly higher accuracy rates. However, these thresholds are valid under the assumption that 2.4 is the best row probability. Since there is an inverse relationship between row probability and threshold due to their calculation, selecting 2.2 as the row probability would require choosing a threshold value greater than 0.39. This corresponds to approximately 0.42.

In the test set results, while the profit rate is not particularly high, the correct prediction rate is satisfactory. Comparing bets with similar odds in the training and test sets, we observe that the peaks of row probabilities differ. In other words, the profit rate can fluctuate sharply depending on the outcomes of less frequently played matches. Nevertheless, the addition of validation enables consistent predictions regarding which thresholds should be selected.

For the Random Forest model, the presented combination proved to be the best prediction strategy. Profit rates can be achieved this way because threshold values vary significantly depending on certain strategies. This variation is influenced by factors such as the limited data available to transform discrete variables like goal and assist expectations into continuous data, the inability to measure the quality of each shot, and the small number of matches.

5. Conclusions and Future Work

This project has demonstrated the development and evaluation of a betting strategy that integrates machine learning models, such as Logistic Regression and Random Forest, to predict match outcomes and inform betting decisions. The benchmark model, based on decision trees and fixed time intervals, provided a foundational framework. However, it revealed notable limitations, such as static decision points and single-model reliance.

To address these gaps, the proposed strategies incorporated dynamic, event-driven analysis and ensemble learning techniques. Both the Logistic Regression and Random Forest models showed improvements in prediction accuracy and decision-making, with the Random Forest model outperforming in most performance metrics. These results highlight the value of leveraging advanced machine learning techniques and probabilistic reasoning in developing more effective betting strategies.

Several directions can be explored to enhance the current approach. First, incorporating additional data sources, such as player statistics, weather conditions, and team dynamics, may improve prediction accuracy and decision-making. Second, exploring alternative machine learning models, such as gradient boosting machines or neural networks, could offer further performance gains. Additionally, incorporating reinforcement learning could refine the betting strategy by learning optimal actions through trial-and-error interactions. Also, evaluating the models in diverse sports and leagues would test their generalizability and robustness across different contexts, contributing to broader applicability in the field of sports analytics and decision-making. Lastly, despite its limitations and high bias, the benchmark models also showed that it is possible to capture some important details with a minute-based analysis. For future work, one can create more sophisticated models for minute-based learning.

References

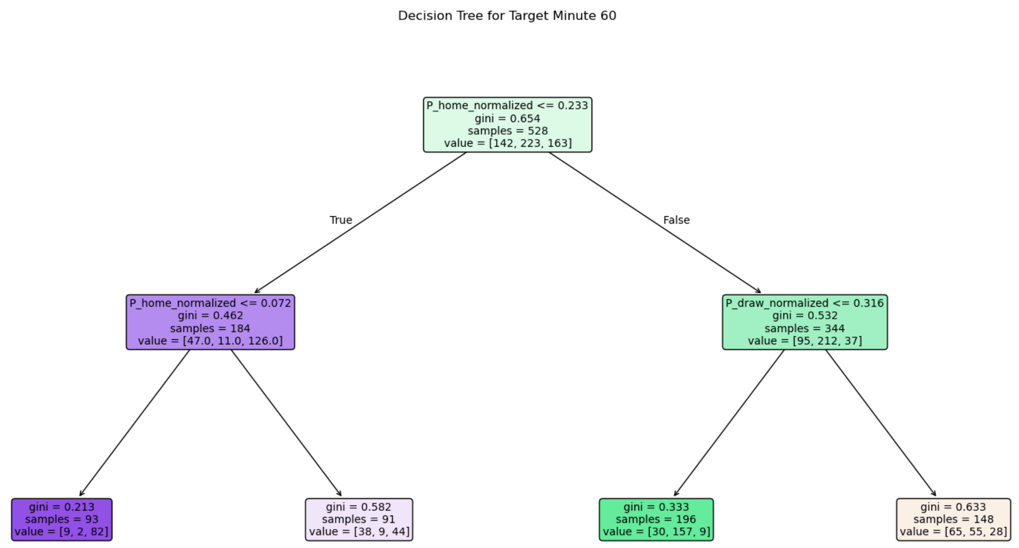
[1]J. Stübinger, B. Mangold, and J. Knoll, “Machine Learning in Football Betting: Prediction of Match Results Based on Player Characteristics,” *Applied Sciences*, vol. 10, no. 1, p. 46, Dec. 2019, doi: https://doi.org/10.3390/app10010046.

[2]J. P. Dmochowski, “A statistical theory of optimal decision-making in sports betting,” vol. 18, no. 6, pp. e0287601–e0287601, Jun. 2023, doi: https://doi.org/10.1371/journal.pone.0287601.

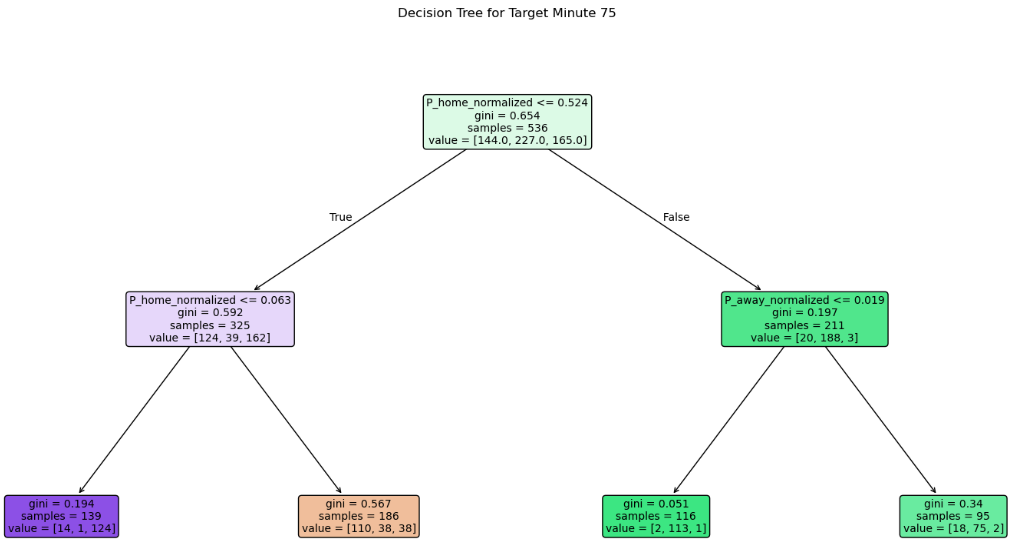
[3]X. Meng, “Soccer match outcome prediction with random forest and gradient boosting models,” *Applied and Computational Engineering*, vol. 40, pp. 99–107, Feb. 2024, doi: https://doi.org/10.54254/2755-2721/40/20230634.

[4]P. Steffen, “Statistical modeling of event probabilities subject on a sports bet : Theory and applications to soccer, tennis and basketball,” *Hal.science*, Jul. 2022, doi: https://theses.hal.science/tel-03891393.

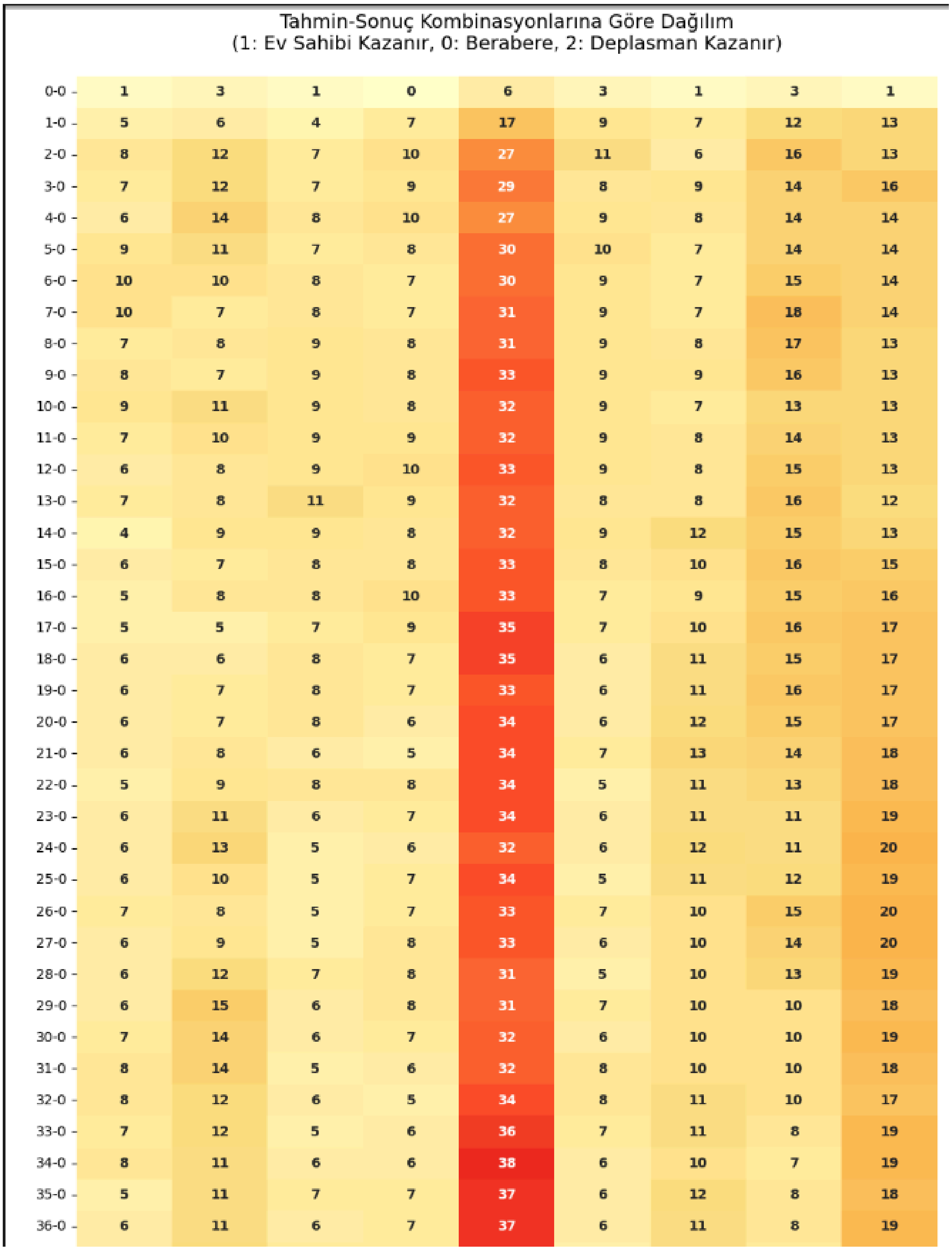
Appendix



Evaluation Results for Target Minute 60:
Accuracy: 0.68
Precision: 0.73
Recall: 0.68
F1 Score: 0.69



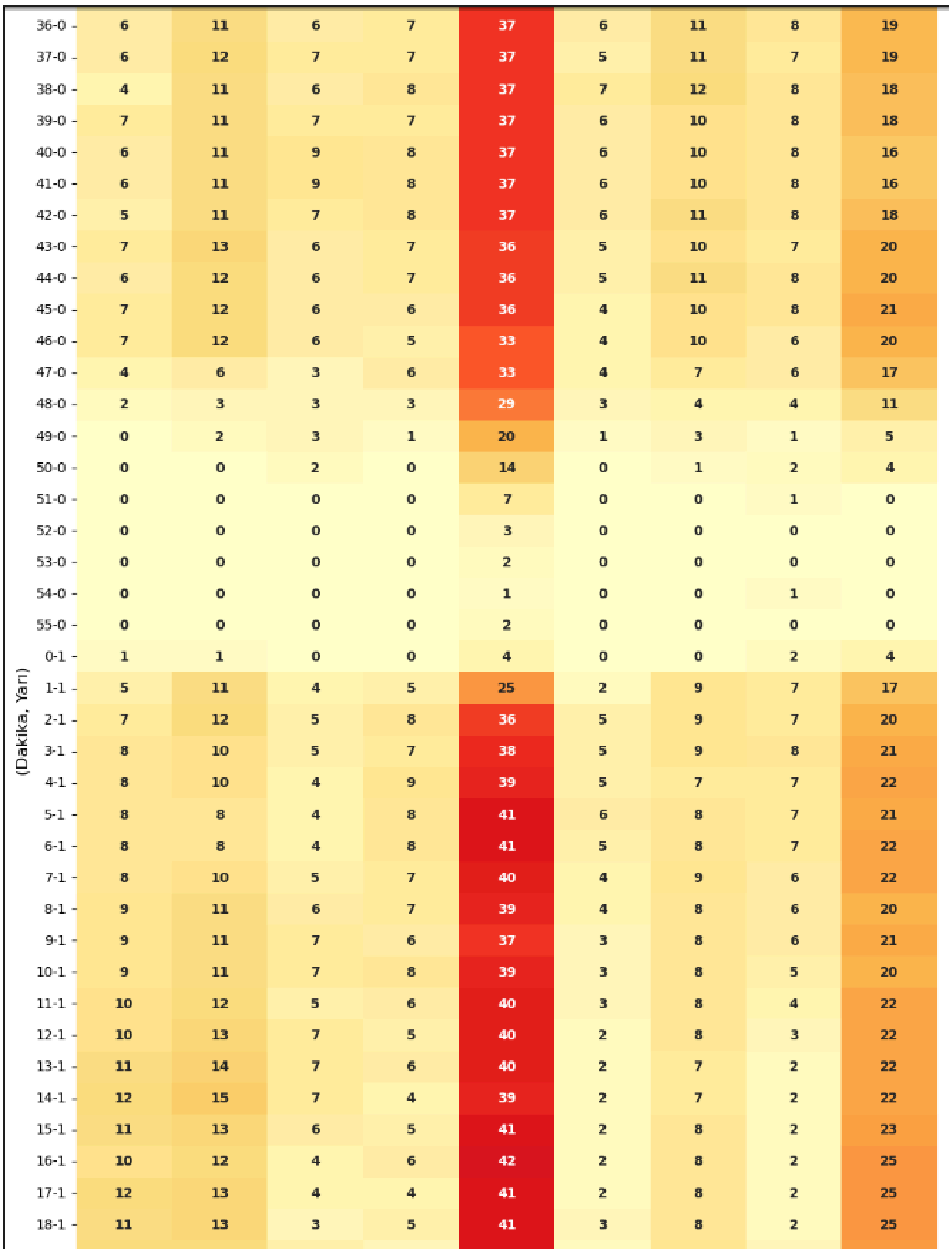
Evaluation Results for Target Minute 75:
Accuracy: 0.74
Precision: 0.78
Recall: 0.74
F1 Score: 0.75



Tahmin-Sonuç Kombinasyonlarına Göre Dağılım
(1: Ev Sahibi Kazanır, 0: Berabere, 2: Deplasman Kazanır)

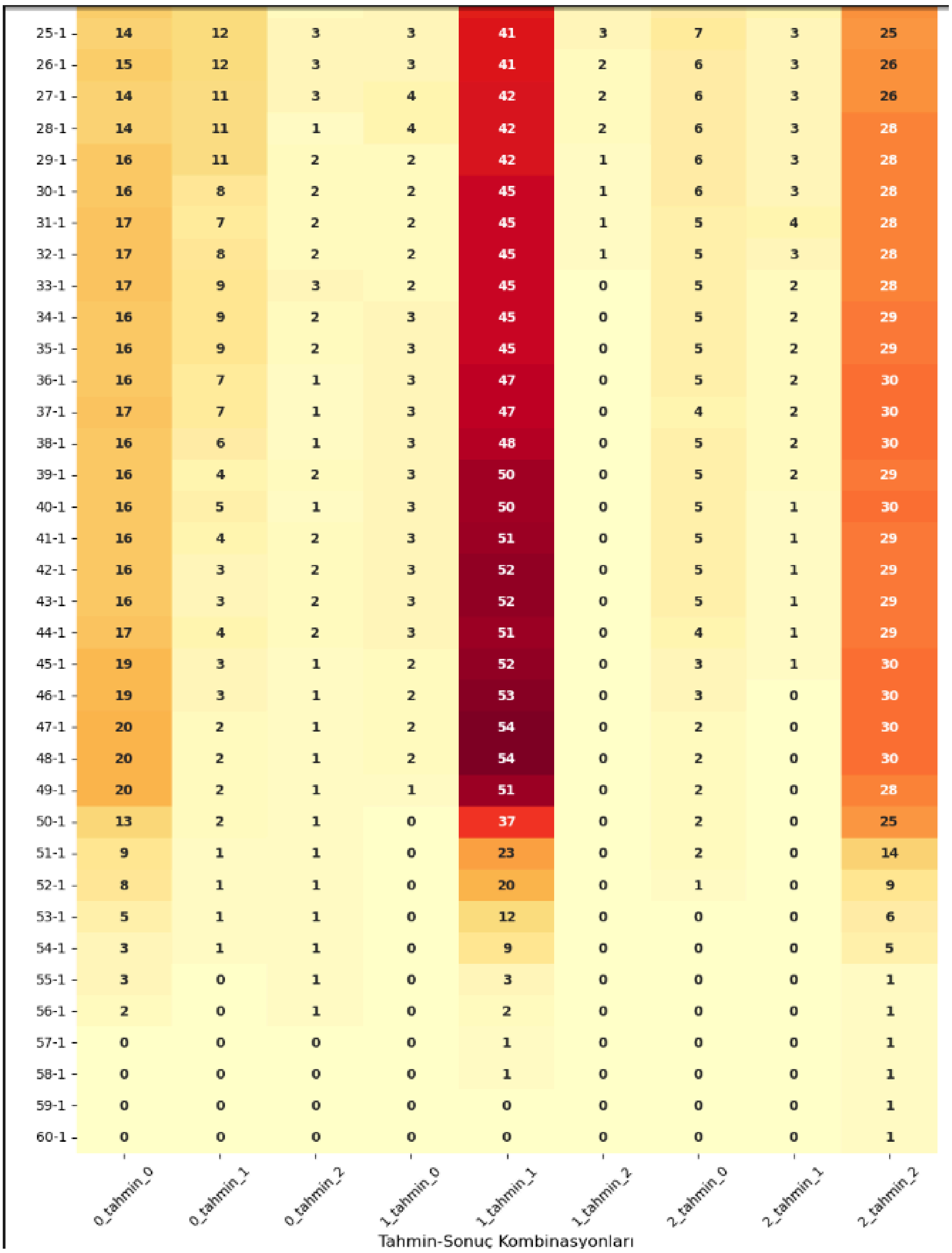
0-0	1	3	1	0	6	3	1	3	1
1-0	5	6	4	7	17	9	7	12	13
2-0	8	12	7	10	27	11	6	16	13
3-0	7	12	7	9	29	8	9	14	16
4-0	6	14	8	10	27	9	8	14	14
5-0	9	11	7	8	30	10	7	14	14
6-0	10	10	8	7	30	9	7	15	14
7-0	10	7	8	7	31	9	7	18	14
8-0	7	8	9	8	31	9	8	17	13
9-0	8	7	9	8	33	9	9	16	13
10-0	9	11	9	8	32	9	7	13	13
11-0	7	10	9	9	32	9	8	14	13
12-0	6	8	9	10	33	9	8	15	13
13-0	7	8	11	9	32	8	8	16	12
14-0	4	9	9	8	32	9	12	15	13
15-0	6	7	8	8	33	8	10	16	15
16-0	5	8	8	10	33	7	9	15	16
17-0	5	5	7	9	35	7	10	16	17
18-0	6	6	8	7	35	6	11	15	17
19-0	6	7	8	7	33	6	11	16	17
20-0	6	7	8	6	34	6	12	15	17
21-0	6	8	6	5	34	7	13	14	18
22-0	5	9	8	8	34	5	11	13	18
23-0	6	11	6	7	34	6	11	11	19
24-0	6	13	5	6	32	6	12	11	20
25-0	6	10	5	7	34	5	11	12	19
26-0	7	8	5	7	33	7	10	15	20
27-0	6	9	5	8	33	6	10	14	20
28-0	6	12	7	8	31	5	10	13	19
29-0	6	15	6	8	31	7	10	10	18
30-0	7	14	6	7	32	6	10	10	19
31-0	8	14	5	6	32	8	10	10	18
32-0	8	12	6	5	34	8	11	10	17
33-0	7	12	5	6	36	7	11	8	19
34-0	8	11	6	6	38	6	10	7	19
35-0	5	11	7	7	37	6	12	8	18
36-0	6	11	6	7	37	6	11	8	19

First part of prediction result combination distribution



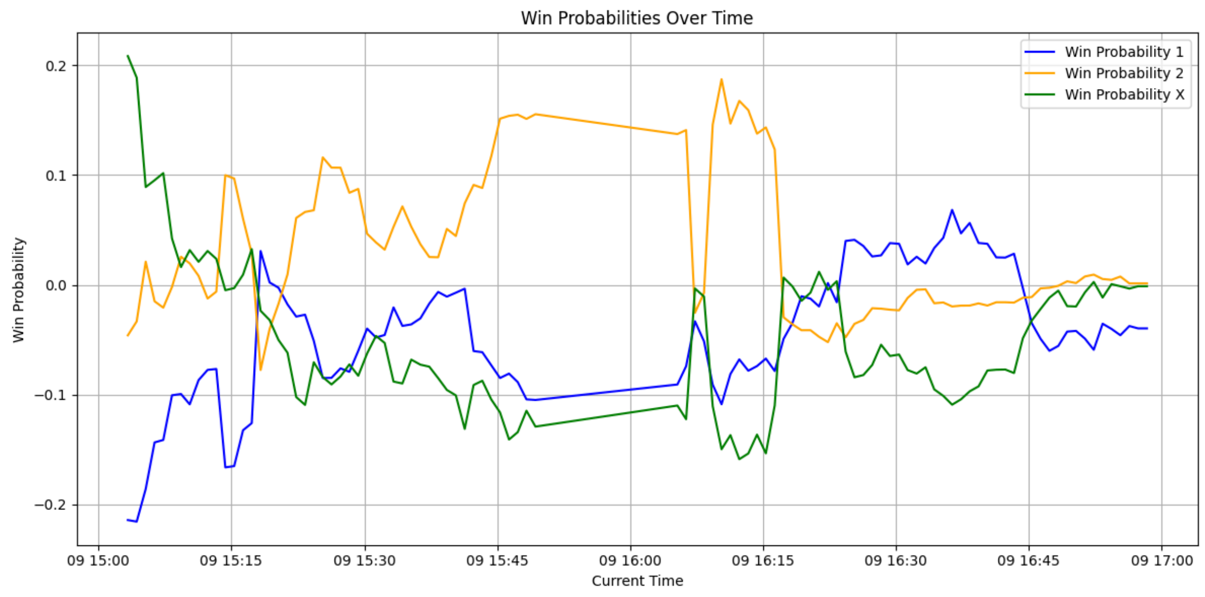
36-0	6	11	6	7	37	6	11	8	19
37-0	6	12	7	7	37	5	11	7	19
38-0	4	11	6	8	37	7	12	8	18
39-0	7	11	7	7	37	6	10	8	18
40-0	6	11	9	8	37	6	10	8	16
41-0	6	11	9	8	37	6	10	8	16
42-0	5	11	7	8	37	6	11	8	18
43-0	7	13	6	7	36	5	10	7	20
44-0	6	12	6	7	36	5	11	8	20
45-0	7	12	6	6	36	4	10	8	21
46-0	7	12	6	5	33	4	10	6	20
47-0	4	6	3	6	33	4	7	6	17
48-0	2	3	3	3	29	3	4	4	11
49-0	0	2	3	1	20	1	3	1	5
50-0	0	0	2	0	14	0	1	2	4
51-0	0	0	0	0	7	0	0	1	0
52-0	0	0	0	0	3	0	0	0	0
53-0	0	0	0	0	2	0	0	0	0
54-0	0	0	0	0	1	0	0	1	0
55-0	0	0	0	0	2	0	0	0	0
0-1	1	1	0	0	4	0	0	2	4
1-1	5	11	4	5	25	2	9	7	17
2-1	7	12	5	8	36	5	9	7	20
3-1	8	10	5	7	38	5	9	8	21
4-1	8	10	4	9	39	5	7	7	22
5-1	8	8	4	8	41	6	8	7	21
6-1	8	8	4	8	41	5	8	7	22
7-1	8	10	5	7	40	4	9	6	22
8-1	9	11	6	7	39	4	8	6	20
9-1	9	11	7	6	37	3	8	6	21
10-1	9	11	7	8	39	3	8	5	20
11-1	10	12	5	6	40	3	8	4	22
12-1	10	13	7	5	40	2	8	3	22
13-1	11	14	7	6	40	2	7	2	22
14-1	12	15	7	4	39	2	7	2	22
15-1	11	13	6	5	41	2	8	2	23
16-1	10	12	4	6	42	2	8	2	25
17-1	12	13	4	4	41	2	8	2	25
18-1	11	13	3	5	41	3	8	2	25

Second part of prediction result combination distribution

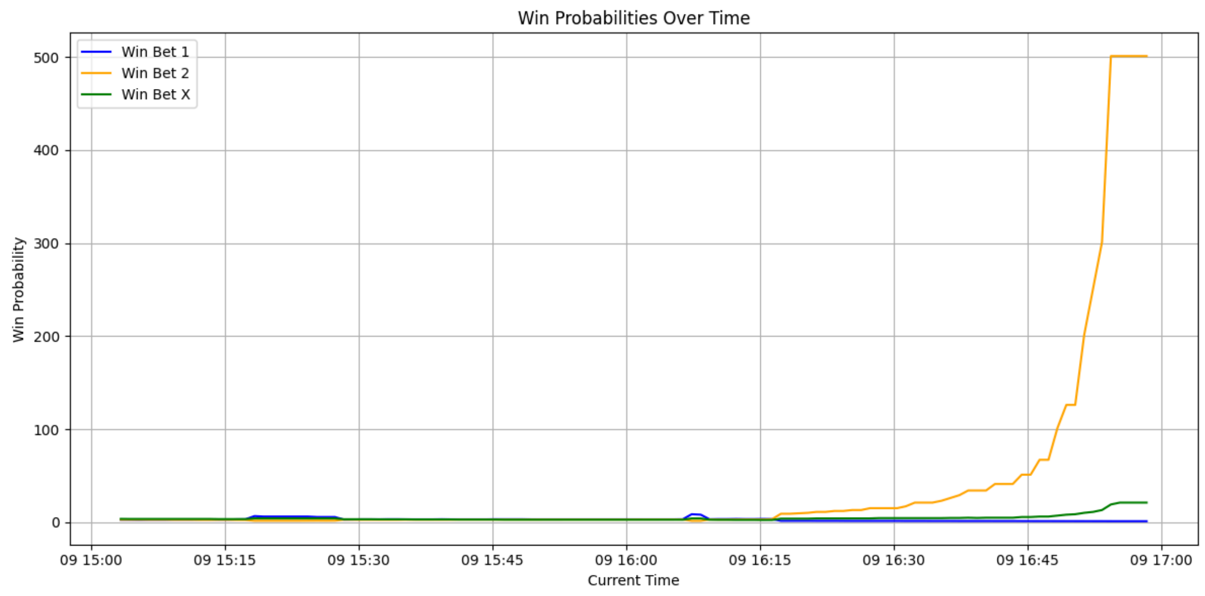


	0_tahmin_0	0_tahmin_1	0_tahmin_2	1_tahmin_0	1_tahmin_1	1_tahmin_2	2_tahmin_0	2_tahmin_1	2_tahmin_2
25-1	14	12	3	3	41	3	7	3	25
26-1	15	12	3	3	41	2	6	3	26
27-1	14	11	3	4	42	2	6	3	26
28-1	14	11	1	4	42	2	6	3	28
29-1	16	11	2	2	42	1	6	3	28
30-1	16	8	2	2	45	1	6	3	28
31-1	17	7	2	2	45	1	5	4	28
32-1	17	8	2	2	45	1	5	3	28
33-1	17	9	3	2	45	0	5	2	28
34-1	16	9	2	3	45	0	5	2	29
35-1	16	9	2	3	45	0	5	2	29
36-1	16	7	1	3	47	0	5	2	30
37-1	17	7	1	3	47	0	4	2	30
38-1	16	6	1	3	48	0	5	2	30
39-1	16	4	2	3	50	0	5	2	29
40-1	16	5	1	3	50	0	5	1	30
41-1	16	4	2	3	51	0	5	1	29
42-1	16	3	2	3	52	0	5	1	29
43-1	16	3	2	3	52	0	5	1	29
44-1	17	4	2	3	51	0	4	1	29
45-1	19	3	1	2	52	0	3	1	30
46-1	19	3	1	2	53	0	3	0	30
47-1	20	2	1	2	54	0	2	0	30
48-1	20	2	1	2	54	0	2	0	30
49-1	20	2	1	1	51	0	2	0	28
50-1	13	2	1	0	37	0	2	0	25
51-1	9	1	1	0	23	0	2	0	14
52-1	8	1	1	0	20	0	1	0	9
53-1	5	1	1	0	12	0	0	0	6
54-1	3	1	1	0	9	0	0	0	5
55-1	3	0	1	0	3	0	0	0	1
56-1	2	0	1	0	2	0	0	0	1
57-1	0	0	0	0	1	0	0	0	1
58-1	0	0	0	0	1	0	0	0	1
59-1	0	0	0	0	0	0	0	0	1
60-1	0	0	0	0	0	0	0	0	1

Last part of prediction result combination



Win probabilities over time


Win Probabilities Over Time
Win Bet 1
Win Bet 2
Win Bet X
Win Probability
Current Time
09 15:00
09 15:15
09 15:30
09 15:45
09 16:00
09 16:15
09 16:30
09 16:45
09 17:00