Abstract

This project aims to predict the winning team in League of Legends (LoL) matches using a Random Forest classifier. By analyzing various in-game statistics, we identify key features that influence match outcomes and evaluate the model's performance. The classifier achieves high accuracy and provides insight into which aspects of the game are most strongly associated with victory.

Introduction

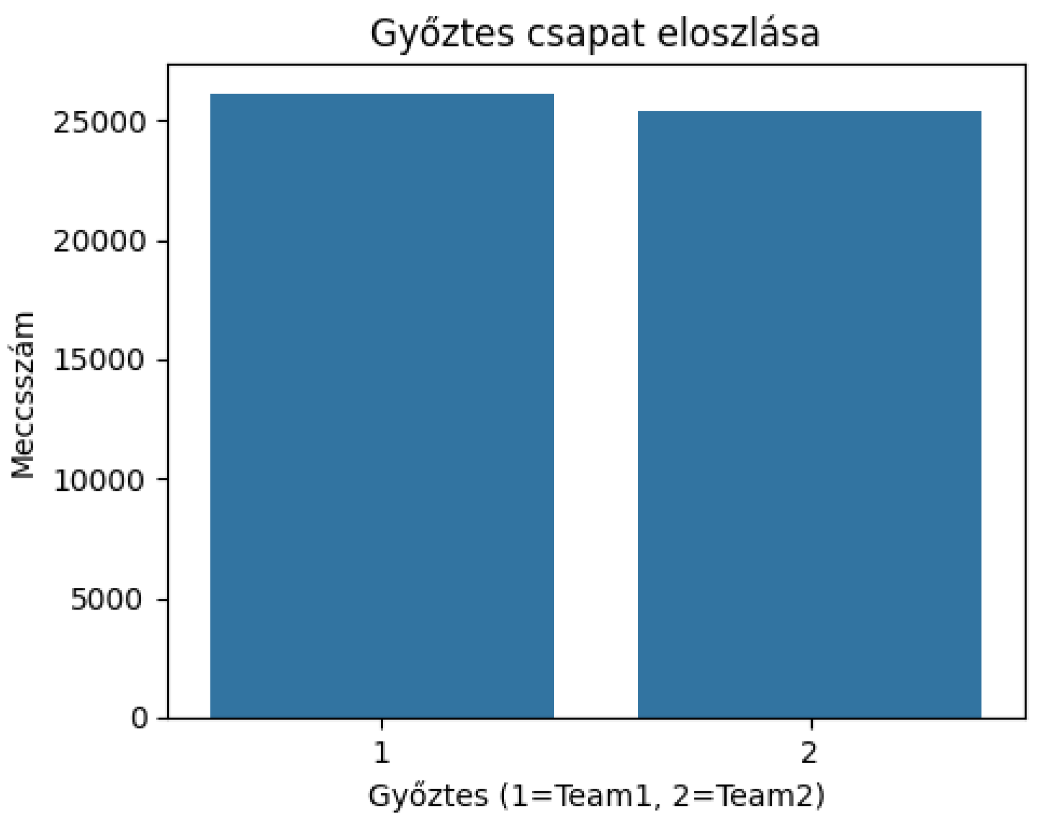
League of Legends is a popular multiplayer online battle arena (MOBA) game where two teams of five players compete to destroy the enemy Nexus. Throughout a match, many statistics are recorded, such as kills, objectives, towers, and other actions. These statistics reflect the state of the game and can be used to infer which team is likely to win.

The goal of this work is to investigate how well we can predict the final winner of a match from in-game features, using a Random Forest classifier trained on historical ranked game data.

Dataset

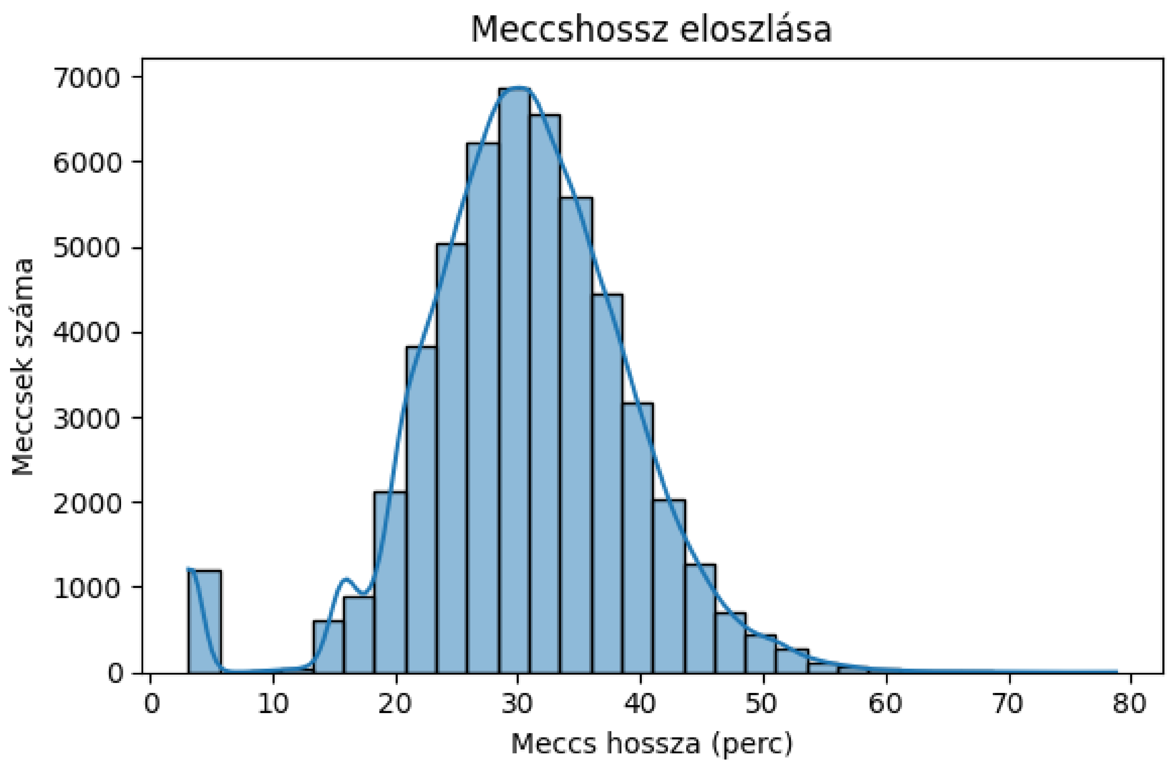
- Source: Kaggle League of Legends ranked games dataset.
- Contains a wide range of in-game statistics: kills, characters, objectives taken, towers, inhibitors, and more.
- Target variable: winning team, encoded as 1 for Team 1 and 2 for Team 2.
- Identifier columns were removed, as they do not provide useful information for the model.

The distribution of winning teams is relatively balanced: Team 1 wins 51.2% of matches, Team 2 wins 48.8%. This balance is important for training a fair and unbiased model.



Game Length Analysis

We also examined the length of the matches. The distribution of game length shows that most games last between 30 and 40 minutes. This suggests that the majority of ranked games are decided in a typical mid- to late-game time window.



Model Training

- The dataset was split into a training set (80%) and a test set (20%).
- A Random Forest classifier was trained on the training set.
- The model uses a collection of decision trees whose predictions are combined to obtain robust and accurate results.

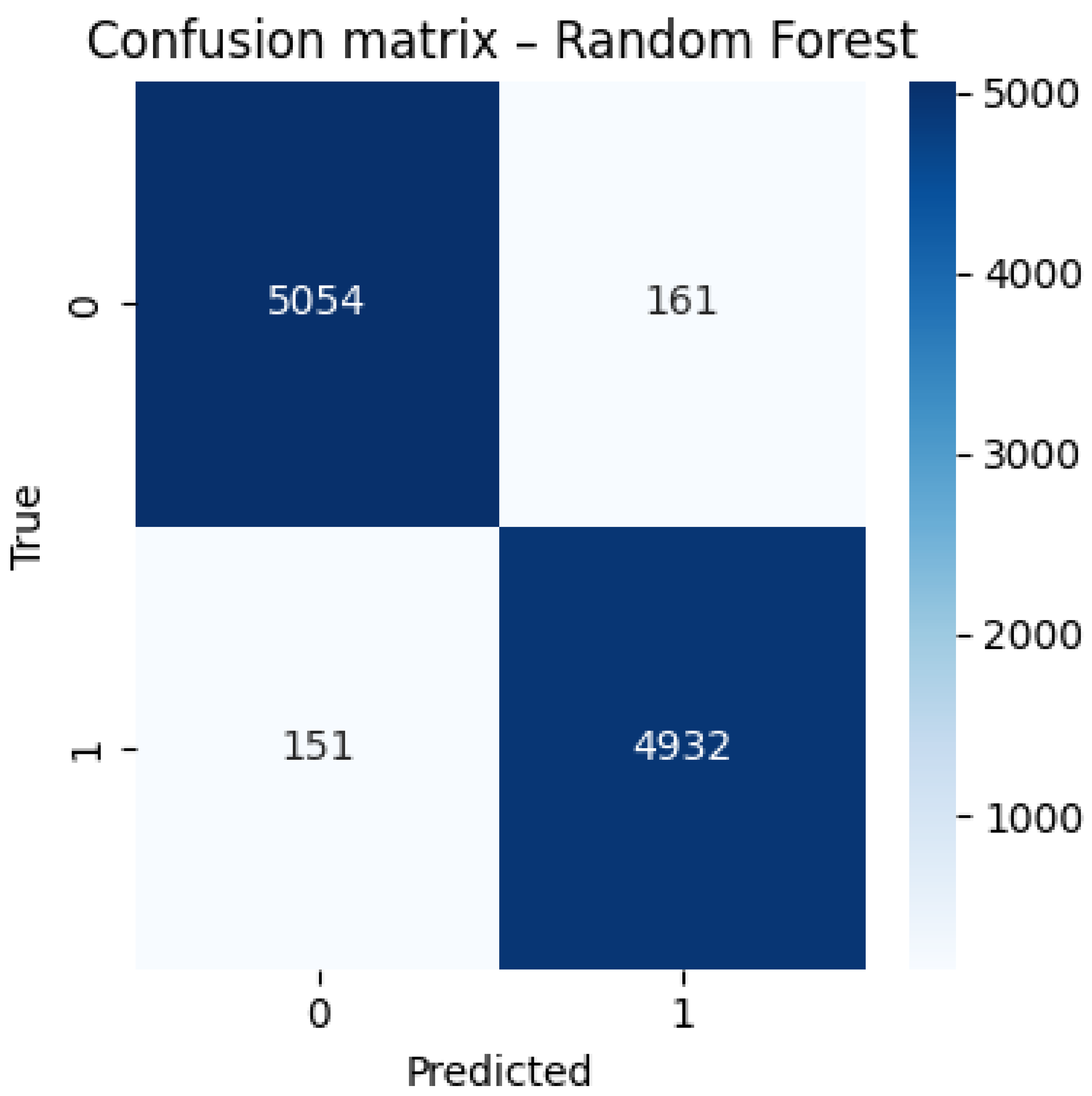
Random Forests are well-suited for this task because they can handle many features, capture non-linear relationships, and provide feature importance scores that help interpret which variables matter most for the predictions.

Results

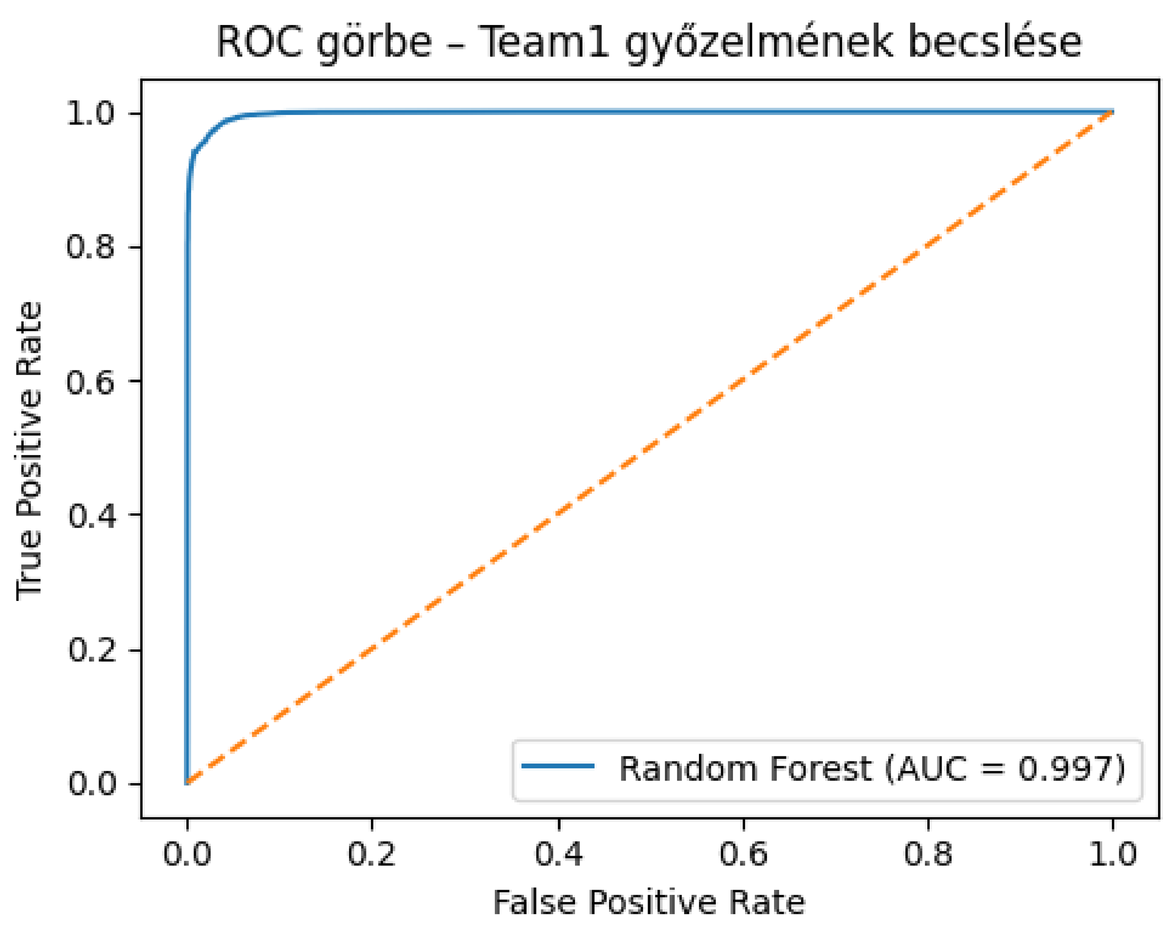
The performance of the model was evaluated on the test set using accuracy as the primary metric.

- Test accuracy: 97%.
- Out of 2000 test matches, the model correctly predicted 1944 outcomes.

The confusion matrix illustrates the number of correctly and incorrectly classified matches.



To further evaluate the classifier, the ROC curve and the Area Under the Curve (AUC) were computed. The model achieves an AUC of 0.99, indicating excellent capability to distinguish between the two classes.

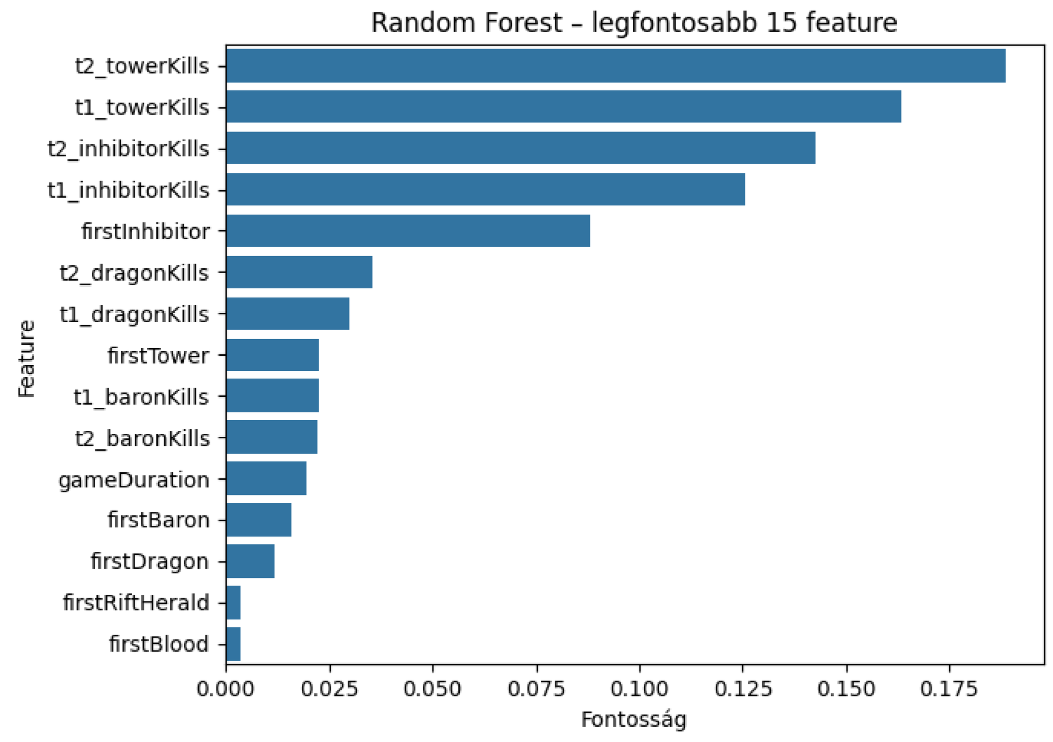


Feature Importance

To understand which features contributed most to the predictions, we analyzed the feature importance scores provided by the Random Forest classifier.

The top features include:

- Total number of tower kills.
- Total number of inhibitor kills.
- Other objective-related statistics (such as dragons and barons taken).



These results are intuitive: destroying towers and inhibitors is crucial for winning, because towers provide defense and inhibitors, when destroyed, allow a team to spawn stronger minions and exert more pressure on lanes. A team that destroys more towers and inhibitors gains a significant strategic advantage that often leads to victory.

Interestingly, champion picks do not appear among the most important features. This suggests that the outcome of a game is driven more by in-game actions and strategies than by the specific champions chosen, indicating that champions are relatively well-balanced with respect to match outcome.

Conclusion

The Random Forest classifier effectively predicts the winning team in League of Legends ranked matches based on in-game statistics. Key features such as tower and inhibitor kills play a major role in determining match outcomes. The model's high accuracy and strong evaluation metrics show its potential for practical applications in game analysis and strategic decision support.

Future work could explore additional features, alternative modeling techniques, or time-dependent representations of the match (for example, predicting win probability at different points in time), in order to further improve prediction performance and provide more detailed insights into game dynamics.