Statistical Analysis of Key Variables in League of Legends

Mirko Bicchierai, Chiara Peppicelli, Jay Senoner

Università di Firenze
Statistical Learning

May 20, 2024

League of Legends

League of Legends is an online multiplayer game in which two teams of five players each compete on a virtual battlefield called Summoner's Rift, which is divided into three main lanes and a jungle area. The goal is to destroy the enemy Nexus, located within their base, while safeguarding your own.

Game

The game progresses through phases, starting with the laning phase where champions focus on gaining experience and gold by defeating minions and engaging in encounters with opponents. As the match unfolds, teams must secure objectives such as turrets, neutral jungle buffs, and epic monsters like the Dragon and Baron Nashor to gain advantages.

Game objectives

- **Towers** – Each lane is guarded by powerful defensive structures called turrets or towers. Every team has 11 towers in total.
- **Inhibitor** – Each lane contains one Inhibitor. A lane's Inhibitor can be attacked after a team has destroyed the three turrets guarding its lane.
- **Drakes** – Drakes are powerful monsters located in the map. All members of the team that kills the drake are provided with buffs that last the entire game and adds up over time, with each additional drake providing further benefits.

Game objectives

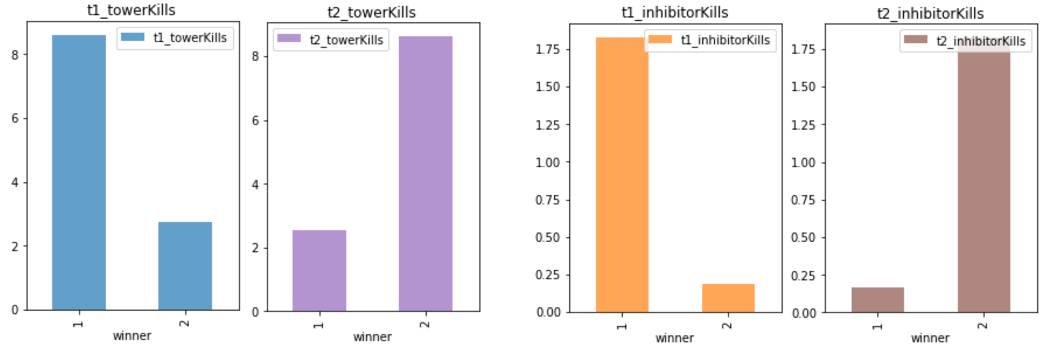
- **Rift Herald** – The Rift Herald is a powerful enemy located in the upper side of the River. Killing the Rift Herald allows it to be summoned again as a battering ram to attack enemy towers.
- **Baron Nashor** – Baron Nashor is the most powerful neutral enemy, located in the upper side of the River. It will spawn after twenty minutes, replacing the Rift Herald.
- **Nexus** – Each team has a Nexus that can only be damaged once all the turrets in a lane, that lane's inhibitor and the Nexus turrets are destroyed. Destruction of the enemy team's Nexus ends the game.

Key moments

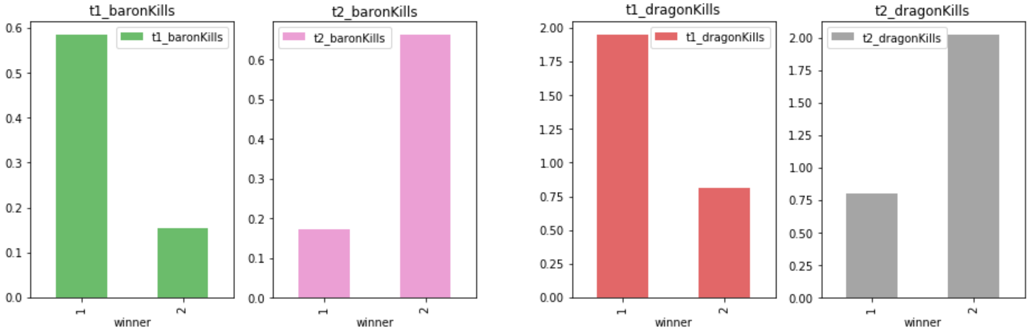
Key moments, such as securing the ”first blood” (the first kill of the game) or taking down the ”first tower” (destroying one of the outer turrets in a lane), can significantly impact the win conditions of each team.

In this project, our objective is to determine which of the initial advantages influence the flow of the game and lead to one team’s victory over another.

Data Visualization



Data Visualization



Dataset Description

- The original dataset contains 61 variables, of which we selected 17. Some variables were removed for simplicity.
- The dataset was randomly split into one-third for testing and two-thirds for training.
- The dataset was already balanced in terms of class distribution (approximately 49/51%). Therefore, we focused on methods that do not account for class imbalance.

Training CART Models

- We trained CART models using different loss functions: Gini index and Information Gain.
- Initial model with $cp = 0$ (maximum overfitting).
- Pruned the tree based on the cp value that minimizes the cross-validated relative error.

CART : Pruning the Tree

- Started with an unpruned tree (high complexity, overfitting).
- Performed pruning based on the complexity parameter *cp*.
- The optimal *cp* value is the one that minimizes the cross-validated relative error.

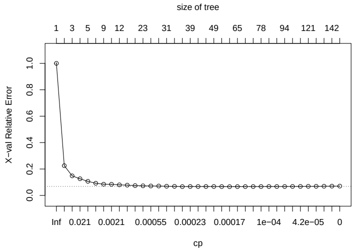


Figure: Gini ($cp = 0.0001382552$)

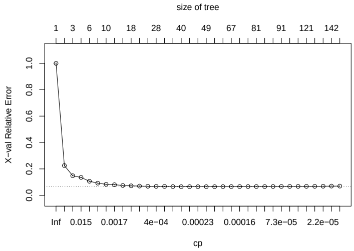


Figure: Entropy ($cp = 0.0002370089$)

CART: Confusion Matrices on Test data

Predicted	**1**	**2**
1	8378	237
2	249	8299

Table: Confusion Matrix (Gini)

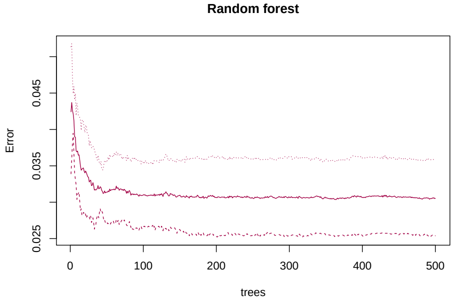
Predicted	**1**	**2**
1	8419	274
2	208	8262

Table: Confusion Matrix (Entropy)

- Evaluated models using the confusion matrix on both training and test data.
- Calculated performance metrics based on the confusion matrix.
- Accuracy: 97.17% vs. 97.19% (almost the same)

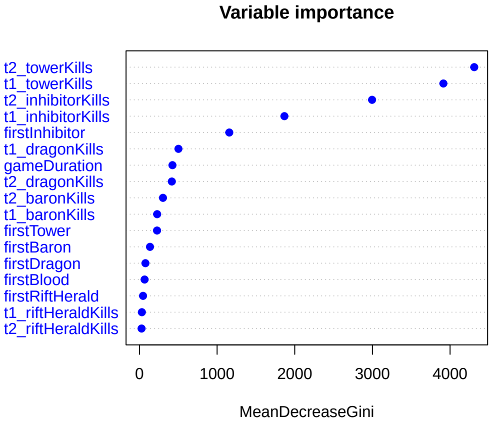
Random Forest: Model Training

- We tested the Random Forest method with ntree = 500.
- We empirically determined the minimum value of the ntree parameter for which the relative error on the data stabilizes.
- We retrained the Random Forest model with ntree = 200.



Random Forest: Variable importance

Since the Random Forest model is an ensemble model that does not provide directly interpretable results, we studied the variable importance.



BART: Theoretical Introduction

- BART is a Bayesian approach to modeling that combines multiple regression trees.
- It is used for both regression and classification tasks, providing flexible, non-linear modeling capabilities.
- BART models are composed of an ensemble of trees, where each tree contributes a small part to the final prediction.
- Each tree is assigned a prior distribution.
- The model as a whole is inferred through a posterior distribution, combining the likelihood of the data with the prior information.
- Advantages: Capable of capturing complex interactions and non-linearities. Provides uncertainty estimates for predictions. Automatically handles variable selection and regularization.

BART: Model Training and Evaluation

- We used the `bartMachineCV()` function (available in the `bartMachine` package) to perform cross-validation and inspect the best value of the ntree parameter for the BART model.
- Cross-validation indicated that the best model is obtained with ntree $= 100$.
- Using the `bart_predict_for_test_data()` function, we evaluated an **accuracy** of 97.27% .

Predicted	**1**	**2**
1	8411	216
2	252	8224

Table: Confusion Matrix

BART: Variable Importance

- We used the `investigate_var_importance()` function from the `bartMachine` package to explore the proportions of variables included in the BART model.
- If `type` is set to `"splits"`, it calculates the proportion of times each variable is used to determine a splitting rule in a tree node.
- If `type` is set to `"trees"`, it calculates the proportion of times each variable appears in any of the trees composing the model.

BART

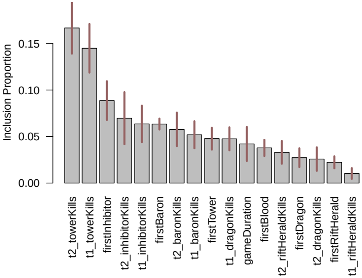


Figure: type = "splits"

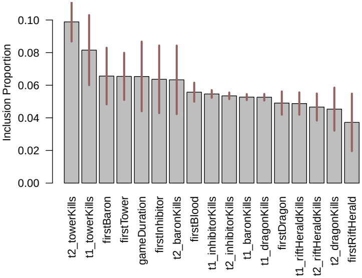


Figure: type = "trees"

Model Comparison

- Both models achieve a very high accuracy and are practically equivalent in terms of confusion matrix on test data.
- The analysis of the most relevant variables reports more or less the same results, which is what we expected.
- Accuracy 97.19% (CART) vs **97.27%** (BART).
- Precision: **97.58%** (CART) vs 97.09% (BART).
- Recall: 96.05% (CART) vs **97.49%** (BART).

Conclusion

- From the results highlighted by the variable importance analysis for the various models, it emerges that the most important variable as a condition for victory is the variable **towerKills**.
- This is consistent with the fact that to win a game, it is necessary to destroy at least 5 towers.
- In general, the variables related to the conquest of structures proved to be the most relevant for predicting the victory of a given team.
- It would be interesting to repeat the analysis without these variables.

Grazie per l'attenzione !

References

J., M. (2017).
(lol) league of legends ranked games.
Kaggle Dataset, https://www.kaggle.com/datasets/datasnaek/league-of-legends.

Kapelner, A. and Bleich, J. (2016).
bartmachine: Machine learning with bayesian additive regression trees.
Journal of Statistical Software, 70(4):1–40.