

Actionable League of Legends Game Outcome Prediction Using Machine Learning Methods

Ryan Stevens, Elle Isabel Angell, Nick Lee, Coleman Garnier

Abstract

This project assesses multiple different ML approaches (GLM, SVM, RF) to create stable models across multiple cross validation folds that inform over 75% predictive success on our sample data.[Elaborate on results]

1. Introduction

The game being focused on for this project is League of Legends, a complex, competitive 5v5 multiplayer online battle arena (MOBA) game. The company that created League of Legends, Riot Games, provides high-quality data through its API, gaining access to game data such as player performance and objectives. While there are existing websites to track player statistics such as match history and specific counter pick options, they mainly focus on surface-level statistics and outcomes, not how one's decisions can influence the result of the game.

The main focus of this project is to analyze and interpret player behavior and decision-making to determine how these patterns may impact the outcome of the game. By examining factors such as participation levels, whether their team had more objectives, and experience/gold levels over the course of the game, key behavioral patterns and the player's impact on the game can be interpreted and identified.

2. Methods

2.1 Data

All data was gathered through the Riot games developer API on patch 25.22, November 1st through 3rd, 2025. This was accomplished using a data collector written in python that collected corresponding game data from two different api endpoints. First, the Riot match data, which is composed of end-of-game summary statistics. This includes which team won the match, the characters (or Champions) played by each team, the ending gold quantity for each team, and other descriptive statistics. Second, the Riot timeline data, which includes snapshots of the game state in one-minute intervals throughout the match. This data includes (for every minute in the game) information such as the position of each player or inventory of each player. Finally, all data was collected from games played where the average MMR (match making rating) of the participating players was 'Diamond II', which consists of players in approximately the top 1.5% of skill. The primary motivation for this choice was to focus on high-quality yet still plentiful match data. Had a lower skill bracket been selected, the results would be less interpret-able within the context of an organized environment. Had a higher skill bracket been selected, data sparsity would have likely become a problem.

Using the data described above, three primary data sets were collected for modelling and exploring data. First, a small set of n=250 matches (2,500 total players) was used for general data analysis and data

Table 1: Datasets

| Data | Size | Type |
|-----------------|----------------------------|--------------------|
| Small Dataset | n=250 (2,500 players) | Matches, Timelines |
| Large Dataset | n=5000 (50,000 players) | Matches, Timelines |
| Matches Dataset | n=50,000 (500,000 players) | Matches |

Table 2: Model Evaluation

| Model.Type | Num..Predictors | Test.Accuracy |
|---------------|-----------------|---------------|
| Random Forest | 23 | 74.7% |
| SVM | 6 | 73.5% |
| GLM | 23 | 75.1% |

exploration this was the ‘small dataset’. Second, a larger set of n=5,000 (50,000 total players) made up the ‘large dataset’. Both the small and large datasets consist of combined match and timeline data. Finally, a matches-only dataset n=50,000 (500,000 total players) collected on the “Diamond IV” skill bracket was used to gain insights on the impact of team composition on the overall win but did not directly contribute to our results.

2.2 Preprocessing

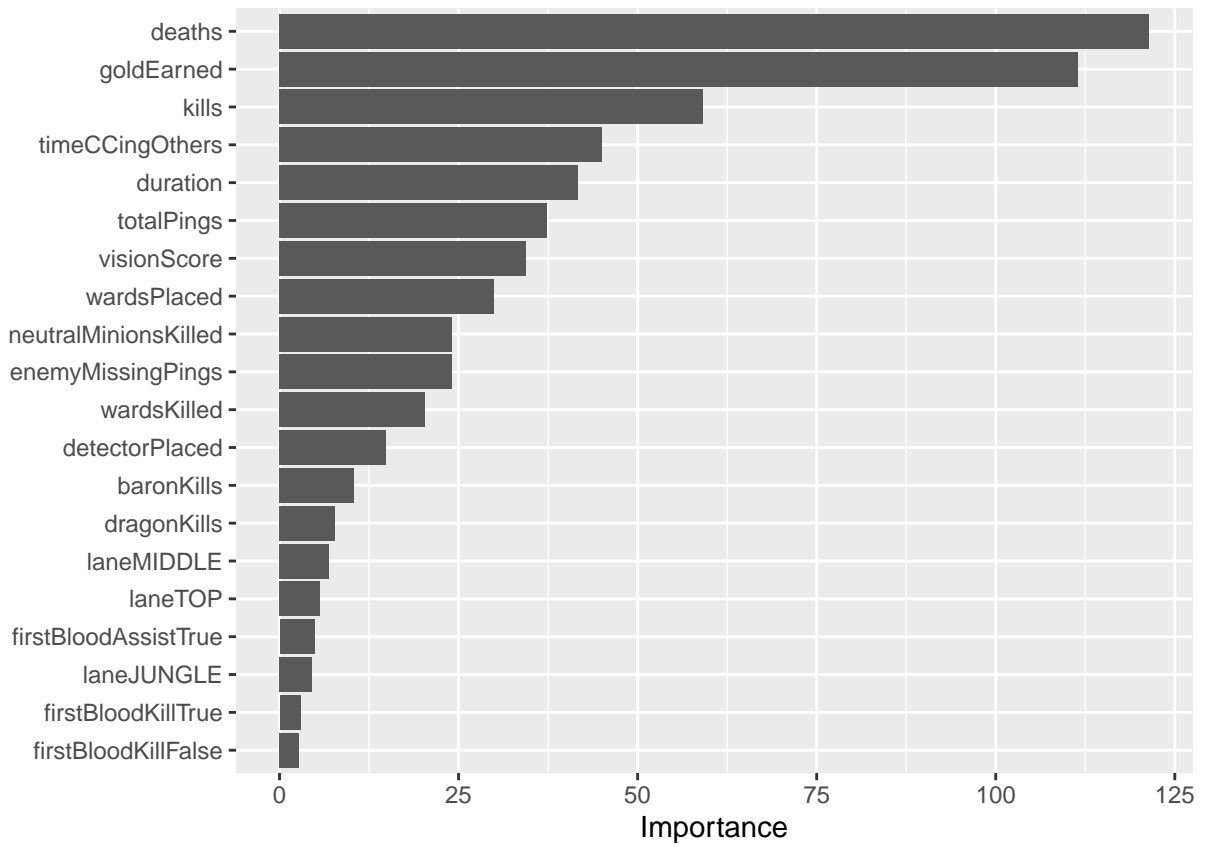
The Riot Games developer API returns a highly nested sequence of DTOs (Data Transfer Objects). These structures contain match level, team level, as well as player level level data embedded within multiple, tightly coupled transfer layers, requiring complicated extraction before modeling. All desired JSON responses were flattened into a simpler form by isolating the relevant participant objects and merging them with corresponding match metadata. To accomplish this, data was cleaned using a python script.

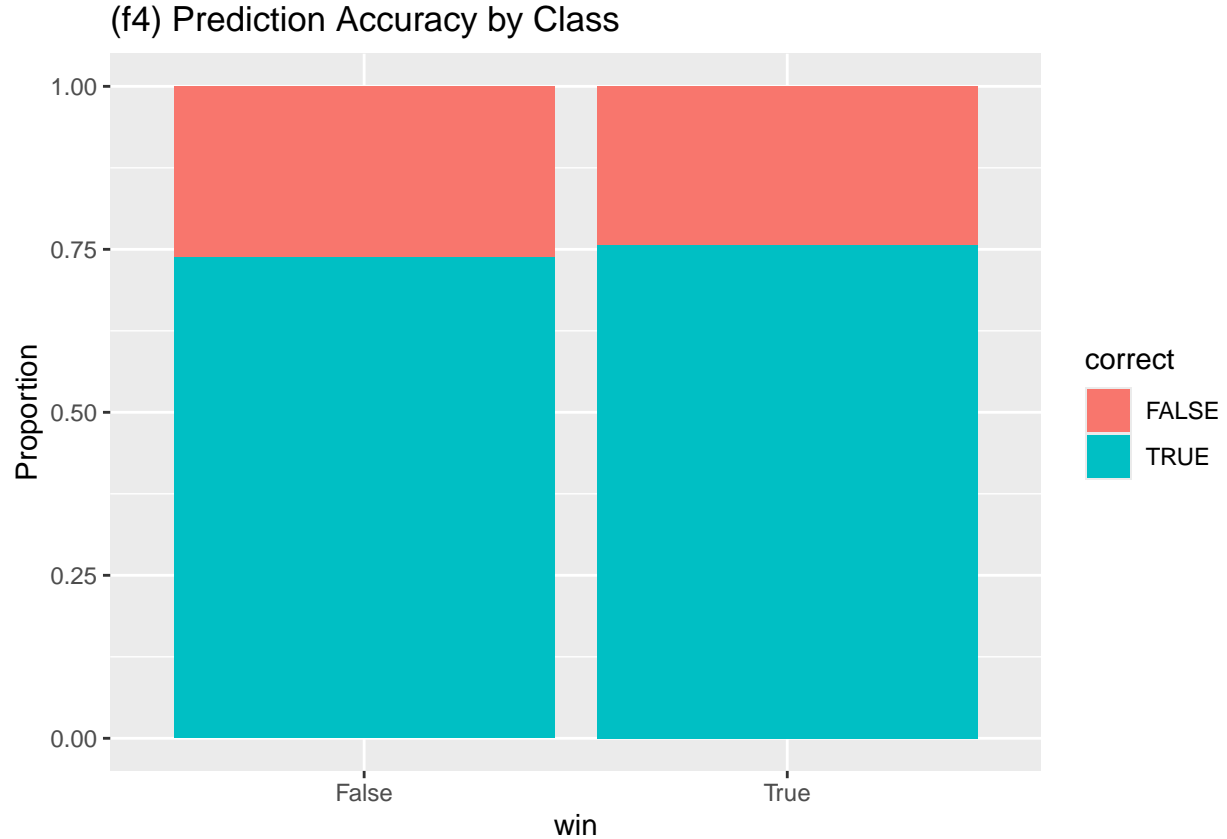
2.3 Model Evaluation

Three primary models were fit to the small dataset in order to determine an optimal strategy. Models were evaluated with a 70/30 train/test split and 5-fold cv was used to find optimal hyperparameter tunings.

As can be seen above in table 2, the GLM model maintained highest test accuracy, while the SVM model was the lowest. However, the SVM model was trained on 6 predictors (‘deaths’, ‘dragonKills’, ‘baronKills’, ‘firstBloodKill’, ‘firstBloodAssist’, ‘kills’), whereas GLM was trained on 23.

The following is the results from the Random Forest model, chosen for its interpretability.





As seen above in f4, the random forest model achieves a high prediction accuracy with a similar number of false positives and false negatives. Furthermore, as seen in f3, we can see that deaths and gold earned play a significant role in influencing the outcome of games. First blood kill and some other statistics provide less insight.

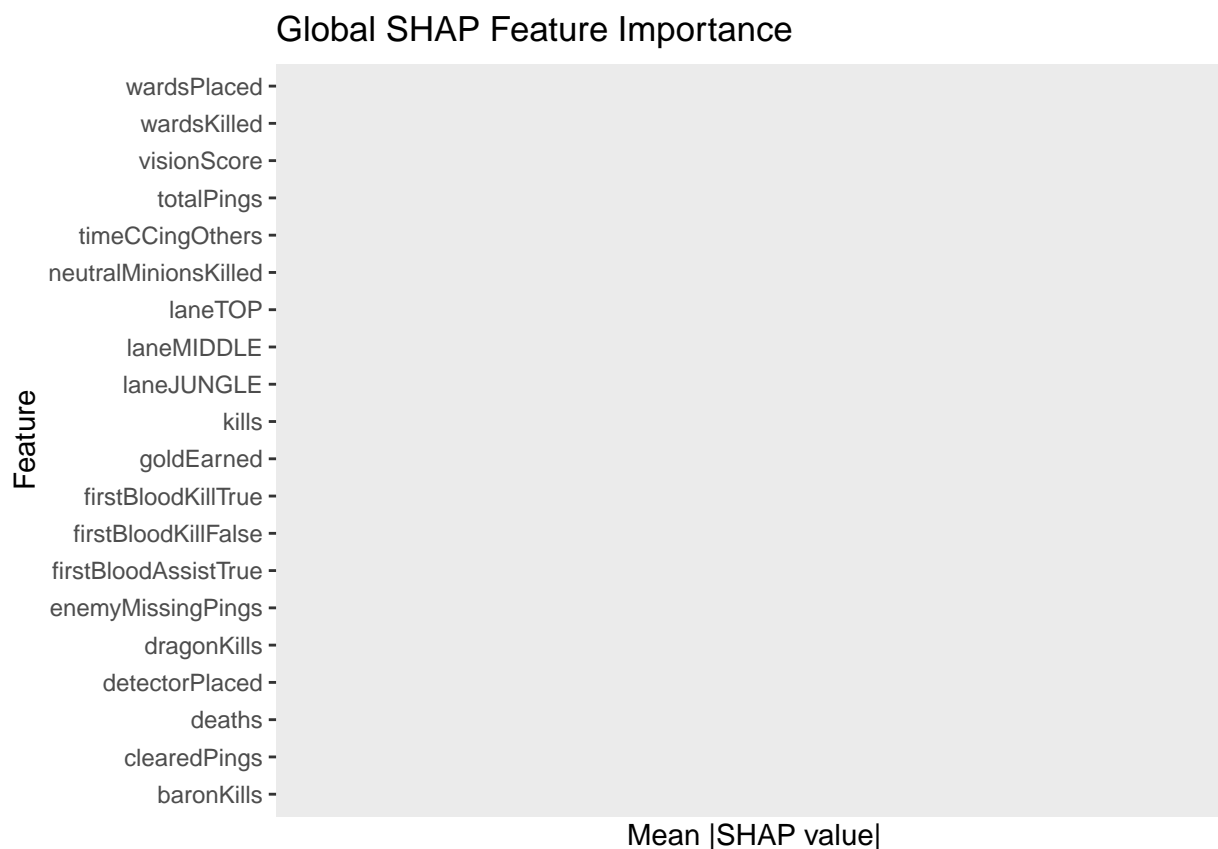
2.4 Interpretability

The approach used builds heavily on accurately classifying game outcomes to categorize samples as either win or defeat. However, in order to determine which factors led to the outcome of the match, SHAP (SHapley Additive exPlanations) was used to quantify each feature's contribution to the model's predictions. SHAP values provide a game-theoretic interpretation of model behavior by breaking down each individual prediction into additive contributions from each feature. This allows interpretability in two ways:

- Global interpretability, where SHAP summarizes which features most strongly influence predictions across all matches
- Local interpretability, where SHAP explains why the model predicted a win or loss for a specific match.

A primary advantage of using SHAP over model-specific analysis is that it is agnostic to the type of model being used and also provides more in-depth insights.

Futhermore, by looking at the local prediction for a specific match, we can begin to provide feedback to the player on things to improve. This aligns with the stated goals of this project and will be explored further later in the paper.



3. Results and Discussion

Model performance across the three supervised approaches shows that logistic regression (GLM) achieved the highest test accuracy at 75.1%, followed closely by the random forest at 74.7%, while the SVM trailed slightly at 73.5%.

Vitaly, both SHAP and the random forest internal importance ranking highlight deaths and goldEarned as the two most influential predictors. This is an expected result as the amount of gold earned throughout the course of a match directly leads to the player with high gold earned purchasing more items, and therefore having more frequent access to victory conditions throughout the match. What is surprising, however, is that the internal estimate from the random forest model holds the total number of deaths as the strongest predictor. Deaths have two primary side-effects throughout the course of the match. First, the enemy player responsible for the death is rewarded with gold, increasing their total gold earned and providing them with more opportunity to engage with victory conditions. However, this relationship is more directly captured with the goldEarned statistic. The second side-effect of a death is that you lose the ability to interact with any aspect of the game for 6-56 seconds (based on level) [riot wiki citation]. This directly impacts the player's presence within the game. In domain-specific terms: deaths decrease your tempo.

On the other hand, first blood-related statistics were commonly ranked lowest for importance within the models. This suggests that early, isolated advantages do not often lead to determining the match outcome. This is counter to some community sentiment and may be the grounds for future analysis.

4. Conclusion

This project demonstrates a pipeline and plausible solution for integrating SHAP and machine learning models to create interpretable, actionable insights about League of Legends match wins. Working with a

high-level dataset, all three models got high predictive accuracy (even when compared to similar projects in the same area). Vitally, SHAP provided a highly detailed and robust vision of the key parts of the dataset, identifying gold, kills, deaths, vision control and influential factors and first blood statistics as non-influential.

The hope is that the work done in this project can help form a foundation for future market-targeted platforms such as analytical systems to support competitive teams in esports, automated coaching for players, or personalized platforms such as the already existent mobalytics.gg.

References

Zamri, M. Asyraf Zamir, et al. “Online Game Outcome Prediction Model Using Weighted-Based Feature Approach.” *Fusion: Practice and Applications (FPA)*, vol. 15, no. 2, 2024, pp. 132–144. DOI: <https://doi.org/10.54216/FPA.150212>

J. A. Hitar-García, L. Morán-Fernández and V. Bolón-Canedo, “Machine Learning Methods for Predicting League of Legends Game Outcome,” in *IEEE Transactions on Games*, vol. 15, no. 2, pp. 171-181, June 2023, doi: 10.1109/TG.2022.3153086. keywords: {Games;Predictive models;Feature extraction;Prediction algorithms;Neural networks;Data mining;Proposals;Esports;feature creation; *League of Legends* (*LoL*);model ensemble;prediction;video games},