League of Legends – Is Leona OP?

DSC80 Project – Early Game Analysis

View the Project on GitHub DEli1610/LeagueOfStatistics

League of Legends – Is Leona OP?

Name(s): Dimitrij Eli

Introduction

The dataset used in this project contains detailed match-level and player-level statistics from professional League of Legends games. It includes information on champions, player roles, in-game events, and early-game performance metrics, providing a strong foundation for both exploratory data analysis and predictive modeling. Gaining an initial understanding of the structure and scope of the data is a crucial first step in the data science lifecycle.

Several questions emerge naturally from this dataset. For instance, how much do early-game advantages influence the final outcome of a match? Are certain champions associated with higher win rates? And to what extent can match outcomes be predicted using only information from the early stages of the game?

Central Research Question

This project focuses on the following research question:

- **Is Leona a better pick as a support champion when it comes to winning a game?**

This question motivates the first part of the analysis, which examines champion-specific performance with a particular emphasis on Leona's impact as a support pick.

Prediction Objective

Building on the hypothesis-driven analysis, the project then shifts toward a predictive task. The goal is to determine whether early-game information, specifically statistics from the first 10 minutes of a match, is sufficient to predict the final outcome of a game. This objective provides a coherent theme that connects data exploration, hypothesis testing, and prediction.

Data Cleaning and Exploratory Data Analysis

Data Cleaning

To ensure a focused and consistent analysis, the dataset was reduced to a subset of relevant columns. The selected variables are grouped based on their purpose within the project.

General Columns

These columns are required for identifying games and assessing data completeness:

- `gameid`
- `datacompleteness`
- `league`
- `playerid`

Columns for Hypothesis 1

These features are used to analyze champion-related performance and in-game outcomes:

- `position`
- `champion`
- `gamelength`
- `result`
- `kills`
- `deaths`
- `assists`
- `teamkills`

- `teamdeaths`

Columns for Hypothesis 2 and Prediction

These early-game features capture events and advantages within the first 10 minutes and are used for the prediction task:

- `firstblood`
- `firstdragon`
- `goldat10`
- `xpat10`
- `csat10`
- `golddiffat10`
- `xpdiffat10`
- `csdiffat10`
- `killsat10`
- `assistsat10`
- `deathsat10`

Dataset Overview (Placeholder)

The table below provides an overview of the cleaned dataset after selecting the relevant columns.

gameid	datacompleteness	league	position	playerid	champion	gamel
LOLTMNT03_179647	complete	LFL2	top	oe:player:c659697694306de62d978569b84c344	Gnar	
LOLTMNT03_179647	complete	LFL2	jng	oe:player:dbdc61a1c41acedcbc7d399727155ac	Maokai	
LOLTMNT03_179647	complete	LFL2	mid	oe:player:694d028e62f4ea668b206ab752b6f94	Hwei	
LOLTMNT03_179647	complete	LFL2	bot	oe:player:90704735ca9fc01f2244f23f6e5d635	Jinx	
LOLTMNT03_179647	complete	LFL2	sup	oe:player:74f3f60a44ee916ecc257a5381be756	Leona	
LOLTMNT03_179647	complete	LFL2	top	oe:player:2df5432fbfcc85dc85c33e269ebd063	Renekton	
LOLTMNT03_179647	complete	LFL2	jng	oe:player:11389d6d8a29729807c3cf528a98050	Ivern	
LOLTMNT03_179647	complete	LFL2	mid	oe:player:7b0aeb1bb297b0d44629e94186bcb6a	Orianna	
LOLTMNT03_179647	complete	LFL2	bot	oe:player:de75f2eb439368d9b39281bd0c4bdab	Varus	
LOLTMNT03_179647	complete	LFL2	sup	oe:player:bea6c089fd517fff3bc020290ae48f7	Braum	

Removal of Team-Level Rows

Upon inspection of the `position` column, rows labeled as `team` were identified as team-level statistics rather than individual champion or player statistics. Since the subsequent analyses focus on champion performance and player-level outcomes, all team-level rows were removed from the dataset before proceeding.

Data Cleaning

After checking the `datacompleteness` column it shows that partialy completed data is missing the most important data for hypothesis 2. Lets get a overview how much of the data is missing.

Datacompleteness Distribution

The figure below shows the distribution of complete and partial observations in the dataset.

Datacompleteness Distribution

The analysis shows that a small portion of the dataset contains partial observations, accounting for approximately 8% of all rows. At first glance, it may seem reasonable to remove these incomplete entries. However, before doing so, we further investigate the underlying missingness mechanisms in Step 3. In particular, we examine whether the missing values are missing completely at random (MCAR), missing at random (MAR), missing not at random (MNAR), or structurally missing (MD).

Support Picks

The table above shows the top 10 champions ranked by win rate, considering only champions with at least 20 games played. While these champions achieve relatively high win rates, the results should be interpreted with caution, as some champions appear with a comparatively small number of games. This overview provides initial insight into strong-performing champions but does not account for role-specific effects or early-game dynamics explored later in the analysis.

Top 10 Champions

Assessment of Missingness

Classification of Missingness

A brief investigation shows that all leagues containing partial data are based in China, whereas leagues with complete data did not participate in matches hosted on Chinese servers. Based on this observation, the missingness can be classified as **missing by design**. One likely explanation is that Chinese servers are operated by Tencent Games, which provides only limited API access compared to other regions. As a result, certain match information is not available through the data collection process.
Source

Hypothesis Testing

In this hypothesis test, we examine whether selecting **Leona as the support champion** is associated with a higher probability of winning a game compared to selecting other support champions. This analysis aims to evaluate the common assumption that Leona's strong engage and crowd control provide a competitive advantage that translates into higher win rates.

Null Hypothesis (H_0)

Teams that pick **Leona as their support** do **not** have a higher probability of winning a game compared to teams that select other support champions.

Alternative Hypothesis (H_1)

Teams that pick **Leona as their support** have a **higher** probability of winning a game compared to teams that select other support champions.

Test Statistic

The **difference in mean win rates** between:

- teams with **Leona as support**, and
- teams with **other support champions**,

where the win rate is defined as the mean of the binary game outcome variable (`result`, with 1 indicating a win and 0 indicating a loss).

Method

A **one-sided permutation test** was conducted under the null hypothesis that picking Leona as support has no effect on the probability of winning. The champion labels were randomly permuted **10,000 times** while keeping the game outcomes fixed, generating a null distribution of the difference in mean win rates.

Significance Level

5%

Results

The observed difference in win rate between teams picking Leona as support and teams picking other support champions was **−2.82 percentage points**, indicating that teams with Leona as support won slightly less often.

The permutation test produced a **p-value of 0.9923**.

Conclusion

Since the p-value is far greater than the chosen significance level of 5%, we **fail to reject the null hypothesis**. There is no statistical evidence that picking Leona as support increases a team's probability of winning. The observed negative difference is small and fully consistent with random variation rather than a true performance advantage.

Framing a Prediction Problem

Prediction Problem

The goal of this prediction task is to **predict whether a team will win a game** (`result`) using only information available from the **first 10 minutes** of gameplay.

The prediction is formulated as a **binary classification problem**, where:

- `result = 1` indicates a win,
- `result = 0` indicates a loss.

Motivation

Early-game performance in League of Legends often sets the trajectory of the entire match. By restricting the model to variables observed within the first 10 minutes, this prediction problem reflects a realistic in-game scenario in which teams or analysts aim to estimate the likelihood of victory before the game is decided.

This approach also aligns with the broader project theme of early-game impact, particularly relevant for champions like **Leona**, who are generally considered strong in the early to mid game.

Features (First 10 Minutes Only)

The model uses the following predictors, all of which are observable by minute 10:

- `firstblood` – indicator for whether the team secured first blood
- `firstdragon` – indicator for whether the team secured the first dragon
- `goldat10` – total team gold at 10 minutes
- `xpat10` – total team experience at 10 minutes
- `csat10` – total team CS at 10 minutes
- `golddiffat10` – gold difference relative to the opposing team at 10 minutes
- `xpdiffat10` – experience difference relative to the opposing team at 10 minutes
- `csdiffat10` – CS difference relative to the opposing team at 10 minutes
- `killsat10` – total team kills at 10 minutes
- `assistsat10` – total team assists at 10 minutes
- `deathsat10` – total team deaths at 10 minutes

All features are restricted to early-game information and do not leak post–10-minute outcomes.

Target Variable

- **Target:** `result` (binary game outcome: win or loss)

Framing as a Machine Learning Task

This problem is framed as a **supervised binary classification task**:

- **Input:** early-game indicators from the first 10 minutes
- **Output:** predicted probability of winning the game

The resulting model aims to quantify how strongly early-game advantages translate into eventual victory.

Baseline Model

Logistic Regression Model

We train a logistic regression classifier using a preprocessing pipeline that first imputes missing values with the mean and then standardizes all features. Standardization ensures that each feature contributes comparably to the model. After fitting the model on the training data, we evaluate its performance on the test set using accuracy, precision, and recall.

Logistic Regression Accuracy: 0.6061689025731252 Precision: 0.6075766016713092 Recall: 0.5996261271167803

Decision Tree Model

We train a decision tree classifier with mean imputation to handle missing values. The tree depth is restricted to reduce overfitting and improve generalization. The model is trained on the training data and evaluated on the test set using accuracy, precision, and recall.

Decision Tree Accuracy: 0.6016604354519464 Precision: 0.5932613739533945 Recall: 0.6466901253573785

Both prediction models achieved similar overall performance, with small but meaningful differences in their evaluation metrics. Although the decision tree achieved a higher recall, logistic regression was selected as the final model due to its higher precision, slightly better accuracy, and superior interpretability. In this context, false positive predictions, where a win is predicted but the team ultimately loses, are particularly undesirable. Therefore, precision was prioritized over recall.

As a result, logistic regression was chosen as the final model and further fine-tuned.

Final Model

Model Comparison and Feature Selection

Reducing the number of features led to a slight decrease in model performance, with lower accuracy, precision, and recall. This indicated that the reduced feature set was unable to capture all relevant early-game information.

Expanding the feature set to include opponent-level and difference-based early-game features resulted in improved performance across all evaluation metrics. To preserve model interpretability, the final feature set was restricted to variables with the strongest influence on the prediction. Specifically, only features with standardized logistic regression coefficients exceeding an absolute value of **0.10** were retained, as smaller coefficients contribute minimally to the model. This approach strikes a balance between predictive performance and interpretability, making the final model both effective and explainable.

Final Model Results

The final prediction model is a logistic regression classifier trained on a carefully selected set of early-game features: gold difference, experience difference, creep score difference, assists at 10 minutes, and opponent assists at 10 minutes. These features capture both economic advantages and early team-fight dynamics while maintaining strong interpretability.

On the held-out test set, the model achieves an accuracy of **0.620**, a precision of **0.620**, and a recall of **0.621**. The close alignment of these metrics indicates balanced performance, with no single metric being disproportionately optimized at the expense of others. In particular, the relatively high precision suggests that the model is reliable when predicting wins, which is important given that false positive win predictions are especially undesirable in this context.

Overall, this result demonstrates that early-game information alone contains meaningful predictive power for match outcomes. The final model provides a strong balance between predictive performance and interpretability, making it well suited for analyzing how early-game advantages influence the probability of winning a match.

The final model achieves an **accuracy of approximately 62%**, indicating that it correctly predicts the outcome of about **two thirds of the matches based solely on early-game information.** Precision and recall are both close to 0.62, suggesting a well-balanced model that does not strongly favor one type of prediction error over the other. While these values are far from perfect, they demonstrate that early-game features already contain meaningful predictive signal. The consistency across all evaluation metrics indicates that the model captures relevant early-game dynamics in a stable and interpretable manner.

Fairness Analysis

In this section, we evaluate whether the final prediction model performs differently across meaningful subgroups of teams. Specifically, we investigate whether the model exhibits disparities in predictive performance between teams that are **ahead early in the game** and teams that are **behind early in the game**.

Group Definition

Teams are divided into two groups based on their gold difference at 10 minutes:

- **Early-Game Advantaged Teams:** `golddiffat10 ≥ 0`
- **Early-Game Disadvantaged Teams:** `golddiffat10 < 0`

This grouping is well aligned with the model’s feature set, which relies exclusively on early-game information.

Evaluation Metric

To assess fairness, we compare **precision** across the two groups. Precision is defined as the proportion of games predicted as wins that are actually wins. This metric is particularly relevant in this context, as false positive predictions (predicting a win when the team eventually loses) are especially undesirable.

Hypotheses

- **Null Hypothesis (H_0):**
  The model is fair. Precision is approximately the same for early-game advantaged and early-game disadvantaged teams, and any observed difference is due to random chance.

- **Alternative Hypothesis (H_1):**
  The model is unfair. Precision is **lower** for early-game disadvantaged teams than for early-game advantaged teams.

Method

A **one-sided permutation test** was conducted to compare the precision of the two groups. The final trained model was kept fixed, and group labels were randomly permuted 10,000 times to generate a null distribution of precision differences under the assumption of fairness.

Results

The observed precision for early-game advantaged teams was substantially higher than for early-game disadvantaged teams. The permutation test produced a **p-value effectively equal to zero** ($p < 0.0001$), indicating that none of the random permutations resulted in a precision difference as large as the observed one.

Conclusion

Since the p-value is far below the 5% significance level, we **reject the null hypothesis**. There is strong statistical evidence that the model's precision is lower for teams that are behind at 10 minutes. This indicates a systematic performance disparity: the model is significantly more reliable when predicting wins for early-game advantaged teams than for early-game disadvantaged teams.

This result is consistent with the model's reliance on early-game features, which naturally provide clearer signals for teams that are already ahead and more ambiguous signals for teams attempting to recover from an early deficit.

This project is maintained by DEli1610

Hosted on GitHub Pages — Theme by orderedlist