Predicting Match Outcomes in League of Legends: A Machine Learning Approach

# 1. Problem Formulation

**a. Problem Background**

In recent years, the esports industry has grown exponentially, and *League of Legends (LoL)* has emerged as one of the most influential multiplayer online battle arena (MOBA) games globally. Predicting match outcomes (win or loss) in competitive games is a crucial task for players, analysts, and developers alike. For players, such predictions can aid in strategy optimization, while for developers and analysts, they provide insights into game balance and performance evaluation.

High-level ranked matches, such as those in the "Diamond" rank, represent the pinnacle of skill and strategy. Studying these matches offers a valuable window into the tactics and patterns that drive competitive gameplay. The complexity of such matches stems from the interplay between player performance, team coordination, and resource management, making outcome prediction a challenging and meaningful task.

Furthermore, the availability of detailed match data from competitive games like *League of Legends* provides an excellent opportunity to apply and test machine learning techniques. This combination of real-world complexity and data availability makes the problem both academically interesting and practically impactful.

**b. Problem Description**

The objective of this project is to develop a machine learning model to predict the outcome of a *League of Legends* match (win or loss for the blue team) based on in-game statistics recorded at the 10-minute mark. The research focuses on analyzing high-ranking matches using the "High Diamond Ranked 10-Minute Match Data."

The target variable, blueWins, is a binary outcome indicating whether the blue team won (1) or lost (0). The dataset consists of 9879 entries and includes 40 features that summarize key aspects of the match, such as team kills, deaths, assists, gold differences, ward placements, and objective control (e.g., dragons and towers). This classification problem aims to predict the binary target variable using these multidimensional features.

**c. Problem Core**

The core of this research lies in addressing the following challenges and questions:

1. **Feature Selection and Importance**:
   * Among the 40 features available, which ones are the most influential in determining match outcomes?
   * For example, is the gold difference (blueGoldDiff) a more decisive factor than the number of kills (blueKills)? How do objectives like dragons and towers contribute to the final result?
2. **Model Selection and Optimization**:
   * What machine learning models (e.g., logistic regression, XGBoost, or Naive Bayes) are best suited for this classification problem?
   * How can hyperparameters be tuned to maximize predictive performance?
3. **Data Balance and Bias**:
   * Are there any issues with data balance? For instance, is the distribution of blue team wins and losses skewed? If so, how should it be handled?
4. **Model Explainability**:
   * Beyond predicting match outcomes, can the model provide actionable insights into the factors that determine the result? For instance, can it identify which team metrics (e.g., kills, gold, wards) are most critical?

This project not only focuses on building accurate predictive models but also emphasizes understanding the underlying factors that contribute to match outcomes, providing interpretability and actionable insights.

**d. Research Significance**

This research holds both academic and practical significance:

1. **Academic Significance**:
   * The project provides a test case for applying machine learning techniques to complex classification problems in real-world settings.
   * By analyzing feature importance, it contributes to understanding how different factors interact in dynamic and competitive environments like online gaming.
2. **Practical Value**:
   * **For Players and Teams**: Predictive models can help players and teams adjust their strategies during gameplay. For example, if a prediction indicates a low likelihood of success, a team may adopt a riskier, high-reward strategy.
   * **For Developers**: Insights into feature importance (e.g., economic metrics, vision control) can guide game balancing efforts and improve the fairness of matches.
   * **For Esports Industry**: Predictive tools can support live match analysis, enhancing audience engagement through advanced insights into team performance and strategies.

# 2. Dataset Description

The dataset used in this study is titled **"League of Legends Diamond Ranked Games - 10 Minute Stats"**, which is publicly available on Kaggle. The dataset can be accessed at the following link: [Kaggle Dataset: League of Legends Diamond Ranked Games](https://www.kaggle.com/datasets/bobbyscience/league-of-legends-diamond-ranked-games-10-min). It provides detailed statistics for high-level competitive matches played in the Diamond rank of *League of Legends*.

**Dataset Overview**

The dataset contains **9879 entries**, each representing a single match recorded at the **10-minute mark**. It consists of **40 features**, capturing the in-game statistics for both the blue team and the red team. All features are numeric, with 34 being integers and 6 being floats. The memory usage of the dataset is approximately **3.0 MB**, and it does not contain any missing values, making it ready for analysis.

Dataset Characteristics：

Data Balance: The target variable blueWins is evenly distributed, indicating no significant class imbalance. This ensures the suitability of machine learning models for classification without additional balancing techniques.

Feature Diversity: The dataset includes a wide range of features that capture different aspects of gameplay, such as kills, vision control, and economic metrics. This diversity enables a comprehensive analysis of the factors influencing match outcomes.

High Data Quality: All columns are complete (no missing values), and the features are well-labeled and structured, making the dataset straightforward to preprocess and analyze.

**Dataset Significance**

This dataset is ideal for predicting match outcomes in competitive League of Legends gameplay due to its:

**Rich Feature Set:** It captures early-game dynamics, including economic and combat metrics, which are crucial for determining match trajectory.

**High-Level Matches**: The data focuses on Diamond-ranked games, representing skilled and strategic gameplay.

**Timeliness**: The 10-minute snapshot allows for early-game predictions, which is valuable for decision-making during live matches.

# 3. Methodology

In this section, we describe the methods, algorithms, and analytical tools used to address the problem of predicting match outcomes (blueWins) in the dataset. We implemented three classification models: Logistic Regression, XGBoost: Random Forest, and Naive Bayes. Each model was optimized using GridSearchCV, evaluated with Feature Importance, validated through Cross-Validation, and assessed using ROC Curve and AUC metrics. Below, we provide a detailed description of each model, the procedures involved, and the underlying mathematics where applicable.

3.1 Logistic Regression

Logistic regression is a simple yet powerful model for binary classification. It predicts the probability of the target class using a sigmoid function:



Here:

* is the probability of the blue team winning.
* are the features of the game.
* are the model coefficients.

**Steps for Logistic Regression**:

1. **GridSearchCV**: We tuned the following hyperparameters:
   * Penalty: L1(Lasso regularization) and L2 (Ridge regularization).
   * Regularization Strength (C): A range of values (e.g., 0.01, 0.1, 1, 10, 100).
   * Solver: Tested liblinear for smaller datasets and lbfgs for faster optimization.
2. **Feature Importance**: After training, feature importance was determined by examining the absolute values of model coefficients:



1. Cross-Validation: We performed 5-fold cross-validation to validate model performance and ensure robustness.
2. ROC Curve and AUC:
   * Predicted probabilities were used to calculate the Receiver Operating Characteristic (ROC) curve.
   * AUC (Area Under the Curve) was computed to measure classification performance:



where and .

3.2 XGBoost: Random Forest

XGBoost (Extreme Gradient Boosting) is an ensemble learning algorithm that builds multiple decision trees sequentially to minimize classification error. Its objective function combines a loss function and regularization:



Here:

* : The classification loss (e.g., log-loss for binary classification).
* : Regularization term to control overfitting.
* : Individual decision tree in the ensemble.

Steps for XGBoost:

1. GridSearchCV:
   * Tuned hyperparameters such as:
     + n\_estimators: Number of trees (e.g., 50, 100, 150).
     + max\_depth: Depth of trees (e.g., 3, 5, 7).
     + learning\_rate: Shrinkage parameter to control tree updates (e.g., 0.01, 0.1, 0.2).
     + subsample: Fraction of samples used to grow each tree (e.g., 0.8, 1.0).
2. Feature Importance:
   * Importance was calculated using the frequency and impact of features used in decision splits:



* + Visualized using bar plots.

1. Cross-Validation:
   * Conducted 5-fold cross-validation with ROC AUC as the evaluation metric.
2. ROC Curve and AUC:
   * Predicted probabilities were used to generate the ROC curve.
   * AUC values were compared across models to assess overall performance.

3.3 Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming feature independence. For binary classification, the model calculates the posterior probability:



Here:

* : The likelihood of features XX given the class yy.
* : The prior probability of the class.

We used the Gaussian Naive Bayes variant, where  is modeled as a Gaussian distribution:



Steps for Naive Bayes:

1. GridSearchCV:
   * Tuned the var\_smoothing parameter to adjust the variance of the Gaussian distribution (e.g.,).
2. Feature Importance:
   * Feature importance was approximated by comparing the class-conditional means ():



1. Cross-Validation:
   * Performed 5-fold cross-validation using accuracy as the primary metric.
2. ROC Curve and AUC:
   * Predicted probabilities for the positive class (P(y=1∣X)) were used to generate the ROC curve and calculate AUC.

3.4 Comparative Steps

To ensure the models' robustness and validity, the following steps were taken for all three methods:

1. Preprocessing:
   * Features were scaled using StandardScaler to normalize input data.
   * The dataset was split into training (67%) and testing (33%) subsets to evaluate model generalization.
2. Performance Metrics:
   * Models were compared using metrics such as accuracy, precision, recall, F1-score, and AUC.
   * Cross-validation scores were averaged to ensure consistency.
3. Visualizations:
   * Feature importance was visualized using bar charts to identify the most critical factors in predicting match outcomes.
   * ROC curves were plotted for each model to compare their classification performance.

3.5 Summary of Methodology

Each model was optimized and evaluated systematically:

* Logistic Regression: Provided a strong baseline with interpretable coefficients.
* XGBoost: Achieved the highest accuracy and AUC due to its ability to model complex feature interactions.
* Naive Bayes: Delivered competitive results with simplicity and speed, though it was outperformed by XGBoost.

By implementing GridSearchCV, feature importance analysis, cross-validation, and ROC curve evaluation, we ensured that all models were rigorously tested and fine-tuned for the best performance.

**4. Discussion**

From the experiments and analyses of the three machine learning models (Logistic Regression, XGBoost, Naive Bayes), the following key findings emerged:

1. **Model Performance Comparison**:
   * **XGBoost** outperformed all other models, achieving the highest accuracy and AUC (approximately 0.92). Its strength lies in capturing complex feature interactions and non-linear relationships.
   * **Logistic Regression** provided a strong baseline with good interpretability. The coefficients allowed us to understand the contribution of each feature to match outcomes.
   * **Naive Bayes** excelled in speed and computational efficiency but was slightly less accurate due to its assumption of feature independence.
2. **Key Features**:
   * **Economic metrics** (e.g., blueGoldDiff, blueTotalGold) and **kills** (e.g., blueKills, redKills) were identified as the most critical factors for determining match outcomes.
   * **Objective control** (e.g., blueDragons, blueHeralds) was also highly important, highlighting the impact of strategic map control on game results.
3. **Cross-Validation and Robustness**:
   * During 5-fold cross-validation, XGBoost demonstrated the highest stability, with consistent accuracy and AUC across all folds, indicating superior generalization performance.
4. **ROC Curve Analysis**:
   * XGBoost had a significantly better ROC curve compared to other models, and its AUC value reflected strong classification capabilities.

**Research Significance**

1. **Academic Significance**: This study illustrates how machine learning algorithms can effectively tackle complex classification tasks. The analysis of feature importance offers insights into understanding high-dimensional data.
2. **Practical Value**:
   * **For Players and Teams**: The models can assist in devising more effective strategies during the early game phase.
   * **For Game Developers**: By analyzing feature importance, developers can refine game mechanics and enhance underutilized aspects of gameplay.
   * **For the Esports Industry**: Predictive models can be used in live commentary to enhance audience engagement with advanced insights into team performance and strategies.

**Future Research Directions**

1. **Time-Series Analysis**: The current data only considers static features from the first 10 minutes. Future research could incorporate time-series models (e.g., LSTM or GRU) to capture the dynamic progression of matches.
2. **Data Expansion**: Adding more granular features (e.g., player-level actions, team communication) could improve predictive accuracy.
3. **Deep Learning**: Exploring neural network models (e.g., Deep Forests, CNNs) could reveal more complex non-linear relationships.