Hacettepe University

Artificial Intelligence Engineering Department

AIN 429 Data Mining Laboratory - 2024 Fall

Mastering the Early Game: Predicting Gold Gain, Game Outcome, and Key Features in League of Legends

---

December 25, 2024

Student Names: Süleyman Yolcu, Emirhan Utku

Student Numbers: 2210765016, 2210765029

1. Introduction

In this project, a dataset of Challenger-Ranked League of Legends games (15-minute snapshot) is explored and modeled. The first goal is the classification task, where predictions are made regarding which team (Blue side or Red side) will win the match based on various in-game statistics (e.g., gold, kills, objectives). The second goal is the regression task, where predictions are made for the total gold (for the Blue side) at the 15-minute mark. Extensive preprocessing, exploratory data analysis (EDA), feature engineering, and both classification and regression modeling were conducted to achieve these goals. The dataset consists of 51 columns, including information about kills, deaths, assists, objective control, wards, and more for both the Blue and Red teams.

2. Method

2.1 Data Preprocessing

The dataset was split into two separate subsets: blue_data (containing features for the Blue side) and red_data (containing features for the Red side). String/object columns not needed for numeric modeling (e.g., blueDragnoType, blueFirstTowerLane, blueFirstBlood, similarly for Red) were dropped. Rows with negative gold values were replaced with the mean of non-negative gold values. Features were rescaled to a 0–1 range using a MinMaxScaler, which is often beneficial for certain models (e.g., distance-based or gradient boosting models).

2.2 Exploratory Data Analysis

Distribution plots were generated, showing that most numerical features (like gold, kills, and assists) are right-skewed, indicating that while moderate values are observed in many games, a smaller subset of matches can "snowball," resulting in high kills and gold totals. Binary or low-frequency features (such as first tower or first dragon) were found to display a large spike at zero, reflecting that these objectives are not always secured early in every game. Overall, the data are found to capture both shorter, subdued matches and longer, higher-action games.

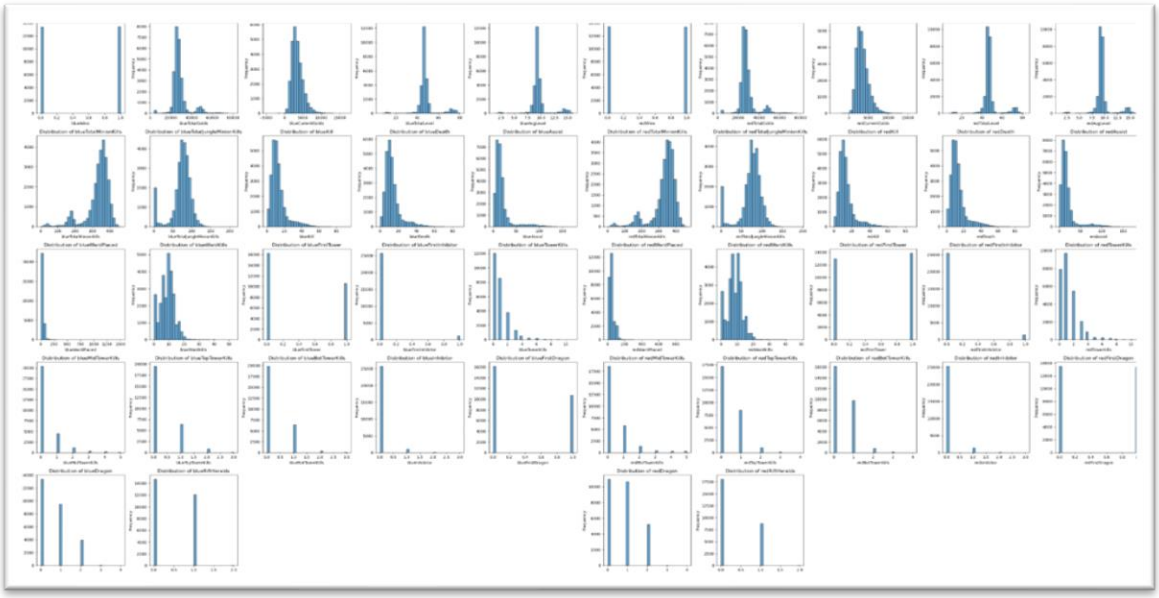

Figure 1. Distributions for Blue and Red team

Correlation matrices for Blue and Red features were examined separately to determine which factors are most strongly correlated with blueWins or redWins.

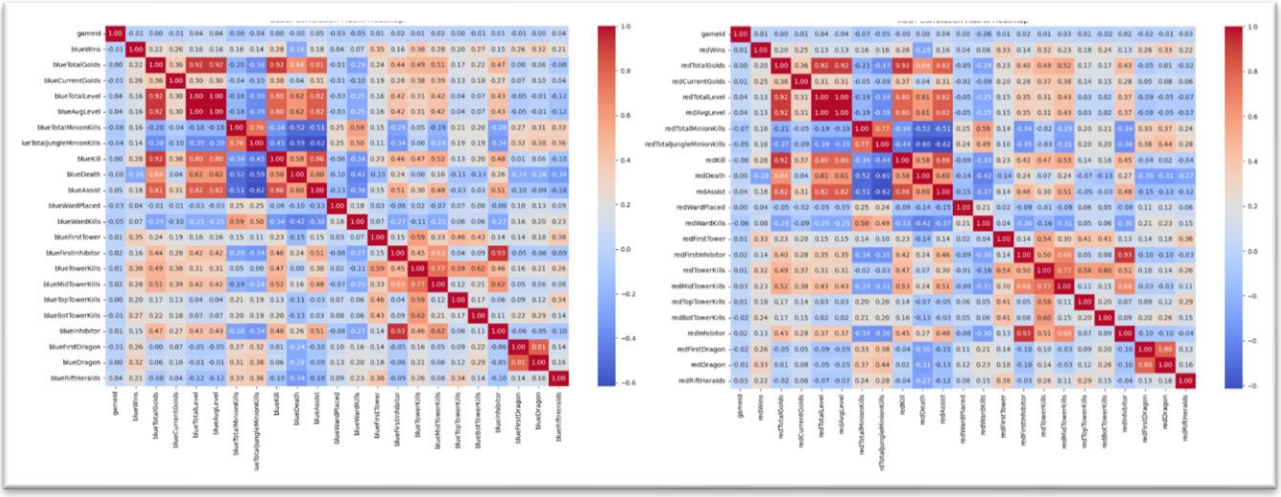

Figure 2. Correlation Matrices for Blue and Red Team

For both teams, total/minion kills, tower kills, first tower, and gold-related features showed the highest correlation with winning.
New features engineered such as:

- **Objective Control Score**: The sum of towers, first tower, dragons, inhibitors, etc.
- **Gold Efficiency**: Total gold earned per (kills + 1).
- **Gold per Minute**: totalGold / 15 (since data is from a 15-minute snapshot).
- **K/D Ratio**: kills / deaths (with a fallback if deaths = 0).
- **Ward Efficiency**: wardKills / wardPlaced (with a fallback if wards placed = 0).

2.3 Model Building (Classification)

The classification for Blue and Red is treated as two separate tasks, with predictions made regarding whether that side ultimately wins (a binary outcome). An 80/20 train_test_split is used. Feature importances were derived using a RandomForest classifier, and the top 10 features were selected for further modeling.
Several modeling approaches were tested:

- RandomForest: A standard random forest classifier to see how it performs on the top 10 features.
- XGBoost, with GridSearchCV and polynomial features for hyperparameter optimization
- Additional classifiers, such as K-Nearest Neighbors, Logistic Regression, Decision Tree, and Support Vector Classifier
- Stacking, combining multiple base classifiers with a meta-learner.

The models were evaluated based on accuracy, precision, recall, F1-score, and a classification report.

2.4 Model Building (Regression)

For the regression task, predictions were made for Blue's total gold at 15 minutes. The label blueTotalGolds was isolated, and columns irrelevant or non-numeric (e.g., blueWins, blueDragnoType) were dropped. An 80/20 train/test split was performed.

The following regression models were applieda:

- **Linear Regression**: Baseline linear model.
- **Random Forest Regressor**: A popular ensemble method for regression tasks.
- **XGBoost Regressor**: Gradient boosting specialized for tabular data.

Performance was evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 score.

3. Results

3.1 Exploratory Findings

The correlation matrices and heatmaps revealed that tower kills, dragons, and kills have a relatively high correlation with winning. Gold (both current and total) exhibited a moderate correlation, suggesting that while gold is crucial, early tower and objective control are equally or more influential. A negative correlation between deaths and winning was also observed, aligning with the understanding that more deaths diminish victory odds. These findings guided the selection of impactful metrics like tower kills, dragon control, and kill counts.

Numerical variables were predominantly right-skewed, indicating that a few matches feature very high kill/gold totals. Binary variables (e.g., firstTower) were mostly 0, reflecting that not all teams secure early objectives. The dataset's balance in win/loss splits for Blue and Red ensures unbiased binary classification.

3.2 Classification Performance

After various classification models were trained and evaluated, an analysis of feature importances was conducted using a Random Forest classifier. Notably, the newly engineered features—KDRatio, GoldEfficiency, GoldPerMin, ObjectiveControlScore, and WardEfficiency—were found to be among the top 10 most important predictors for both the Blue and Red subsets.

The top 10 Random Forest feature importances for the Blue and Red sides are provided below:

Blue Team Feature Importances

Feature	RandomForestImportance
blueKDRatio	0.161556
blue_GoldEfficiency	0.096186
blueGoldPerMin	0.094232
blueTotalGolds	0.089067
blueDeath	0.061633
blueCurrentGolds	0.050172
blueTotalMinionKills	0.048364
blue_ObjectiveControlScore	0.042592
blueWardEfficiency	0.040090
blueTotalJungleMinionKills	0.039408

Red Team Feature Importances

Feature	RandomForestImportance
redKDRatio	0.155347
red_GoldEfficiency	0.093328
redTotalGolds	0.088096
redGoldPerMin	0.087420
redDeath	0.067365
redTotalMinionKills	0.044170
red_CurrentGolds	0.042025
red_ObjectiveControlScore	0.041562
redCurrentGolds	0.040901
redWardEfficiency	0.036946

By selecting the top 10 features using Random Forest's feature importances, the model achieved ~79% accuracy for both Blue and Red. A balance between precision and recall (~0.79–0.80) was observed. GridSearch hyperparameter tuning and polynomial interaction terms improved the XGBoost model's accuracy to ~78–79%. The ensemble stacking model achieved the highest accuracy (~80%).

Table 1. Results for Classification

Model	Accuracy	F1	Precision	Recall
KNN	0.768027	0.767854	0.780812	0.755319
LogisticRegression	0.780697	0.774045	0.811921	0.739545
DecisionTree	0.787963	0.793616	0.784792	0.802641
SVC	0.793553	0.792665	0.809015	0.776963
RandomForest	0.797839	0.800368	0.802879	0.797872
XGBClassifier	0.797652	0.800294	0.802360	0.798239
Stacking	0.798025	0.800000	0.804751	0.795304

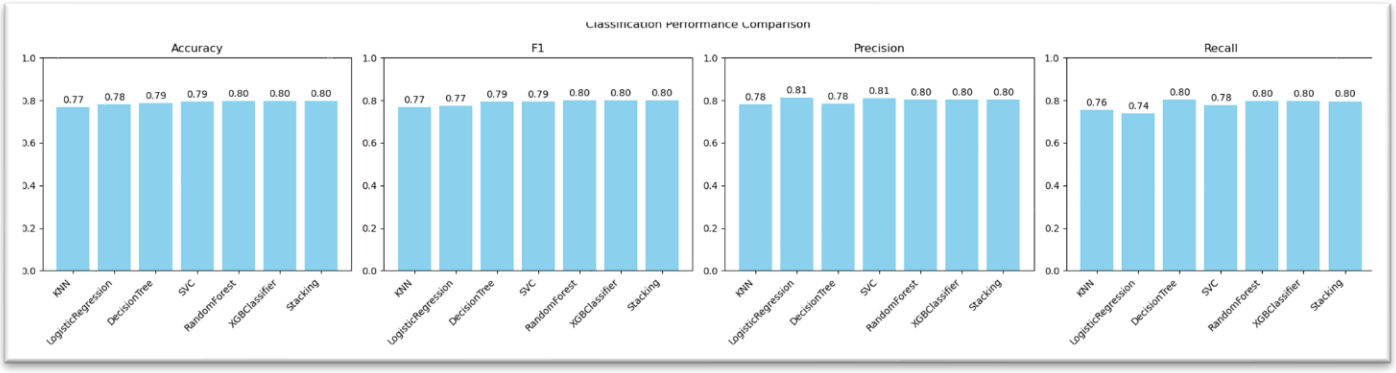

Figure 3. Result Comparison for Classification

3.3 Regression Performance

Regression models achieved strong predictive power for Blue's total gold at 15 minutes. Linear Regression explained ~96% of the variance (R^2 ~0.96) but produced higher MSE (~2.35M) and MAE (~1059). Random Forest and XGBoost reached R^2 ~0.99 with significantly lower MSE (~600–700k) and MAE (~600).

Table 2. Results for Regression

Model	MSE	MAE	R^2
LinearRegression	2,347,999	1059.137606	0.962224
RandomForestRegressor	743,927.6	603.098558	0.988031
XGBoostRegressor	773,095.5	604.368544	0.987562

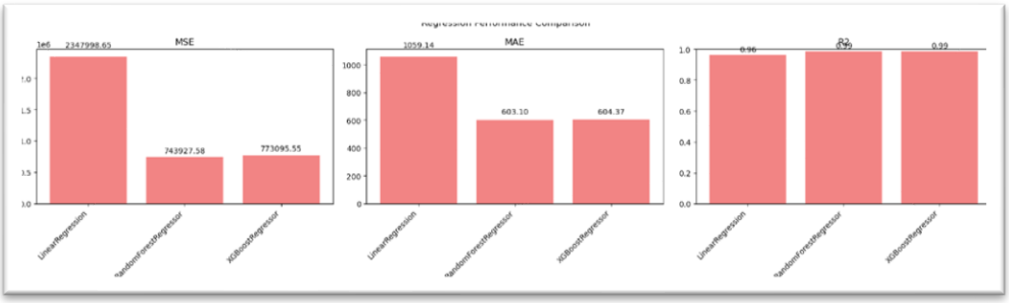

Figure 4. Result Comparison for Regression

3.4 SHAP Interpretability

It is recognized that SHAP (SHapley Additive exPlanations) is a powerful way to explain complex machine learning models, such as Random Forest or XGBoost. Game-theoretic principles are used by SHAP, and an importance value is assigned to each feature for every prediction. As a result, a clearer view is provided of how each feature influences the model's output.

It was observed from the SHAP bar plot that blueTotalLevel, blueKill, and blueTotalMinionKills have the highest average impact on predicted gold. This aligns with League of Legends gameplay, because leveling up, getting kills, and farming minions are major ways to gain gold.

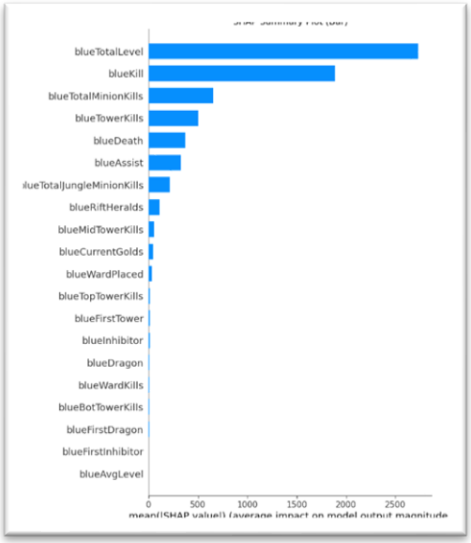

Figure 5. SHAP Summary Plot (Bar)

A more detailed view is shown by the SHAP beeswarm plot. High kill counts (marked in red) cause predictions to shift upward, while low kill counts (marked in blue) cause predictions to shift downward. By illustrating both the overall importance of each feature and how specific values change the predictions, SHAP analysis increases model transparency and builds trust in its decisions.

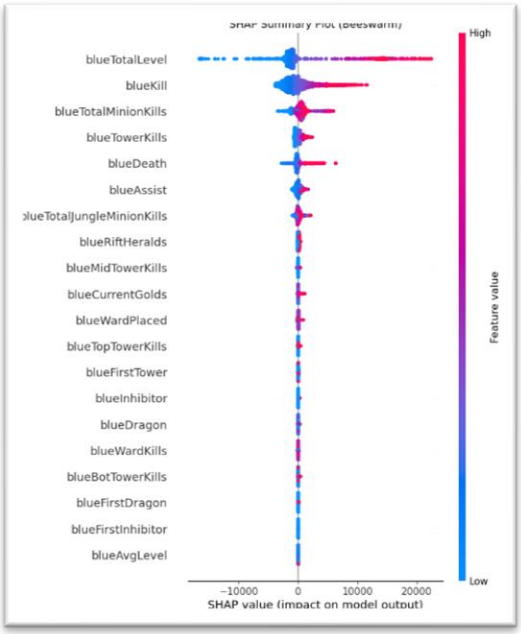

Figure 6. SHAP Summary Plot (Beeswarm)

4. Discussion

An accuracy of around 80% was achieved by the classification models. This result shows that the winning side can be predicted within the first 15 minutes of gameplay. By focusing on key features (like kills, gold, and objectives), it was possible to capture the main factors that drive early-game success. These findings suggest that even a short snapshot of match data can offer a strong signal about the final outcome. The regression models also provided further evidence of this early-game influence. It was observed that in-game metrics—such as kills, levels, and objectives—strongly affected the amount of gold collected by each side. When high kill counts, steady leveling, and early objectives were present, gold tended to grow faster. These patterns confirm that the early game is a critical phase in League of Legends, and that data gathered by the 15-minute mark can be used to make accurate predictions about both victory and overall gold progress. Through these results, the usefulness of early-game data was verified. By relying on core features and meaningful metrics, both the classification and regression tasks showed strong performance.

5. Future Work

In the future, multiple intervals could have been used to track match progression, instead of relying on a single 15-minute snapshot. Champion picks, bans, and synergy metrics could have been included, so that game strategies would have been captured more accurately. Advanced measures of map control, roam timings, or lane assignments might have been explored, and deep learning methods such as neural networks or LSTM for sequential data could have been investigated. It also would have been beneficial to combine data on skillshots, ward timing, or gold spending to create a richer modeling approach.

6. References

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*(Oct), 2825–2830.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (pp. 4765–4774).
- League of Legends API Documentation. (n.d.). Retrieved from https://developer.riotgames.com/
- Scikit-learn Documentation. (n.d.). Retrieved from https://scikit-learn.org/stable/
- Gyejr95. (n.d.). *League of Legends Challenger Rank Game 10min15min [Data set]*. Kaggle. Retrieved from https://www.kaggle.com/datasets/gyejr95/league-of-legends-challenger-rank-game10min15min

Appendices

Correlation matrices for K Nearest Neighbour, Logistic Regression, Decision Tree, Support Vector Classifier, Random Forest, relatively