AITA
ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

■ RESEARCH ARTICLE

A Machine Learning Based Predictive Analysis Use Case For eSports Games

Atakan Tuzcu [a], **Emel Gizem Ay** [a†], **Ayşegül Umay Uçar** [a], **Deniz Kılınç** [a]

[a] Department of Computer Engineering, University of Bakırçay, İzmir, Turkey
[†] aemelgizem@gmail.com, corresponding author

RECEIVED MARCH 5, 2023
ACCEPTED APRIL 25, 2023

CITATION Tuzcu, A., Ay, E.G., Uçar, A. U., & Kılınç. D. (2023). A machine learning based predictive analysis use case for eSports games. *Artificial Intelligence Theory and Applications, 3*(1), 25-35.

Abstract

League of Legends (LoL) is a popular multiplayer online battle arena (MOBA) game that is highly recognized in the professional esports scene due to its competitive environment, strategic gameplay, and large prize pools. This study aims to predict the outcome of LoL matches and observe the impact of feature selection on model performance using machine learning classification algorithms on historical game data obtained through the official API provided by Riot Games. Detailed examinations were conducted at both team and player levels, and missing data in the dataset were addressed. A total of 1045 data were used for training team-based models, and 5232 data were used for training player-based models. Seven different machine learning models were trained and their performances were compared. Models trained on team data achieved the highest accuracy of over 98% with the AdaBoost algorithm. The top 10 features that had the most impact on the prediction outcome were identified among the 47 features in the dataset, and a new dataset was created from team data to retrain the models. After feature selection, the results showed that the accuracy of Logistic Regression increased from 89% to 98% and the accuracy of Gradient Boosting algorithm increased from 96% to 98%.

Keywords: league of legends; riot game; machine learning; random forest; gradient boosting

1. Introduction

Multiplayer Online Battle Arena (MOBA) games are a genre of games that offer a team-based combat experience, requiring strategy, coordination, and skill. The primary objective in MOBA games is to destroy the opponent team's main base. Sports analytics is a method used in analyzing player performance, team strategies, and predicting competitive outcomes by utilizing data obtained from such games. This study was conducted using data from one of the MOBA games, League of Legends (LoL). Similar to other MOBA games, LoL follows a 5v5 game style, where teams consist of 5 players in roles such as top lane, mid lane, jungle, marksman, and support. The tasks of players based on these roles vary according to different strategies. Due to the combination of limited parameters in the game, many possible game strategies can be formed, as the items obtained during the game can elicit different reactions from the characters.

League of Legends (LoL) is a team game, and the data of all five players in the team should be taken into consideration. Poor performance of some players can be

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2023 University of Bakırçay

compensated to a certain extent, and there is still a possibility of the team winning. Individual player evaluations can lead to inaccurate predictions of game outcomes. When creating the dataset, match data was retrieved using an API, specifically focusing on recent matches that are closer to the current date. The dataset includes data for each player in the match. To perform team-based analysis, the data was grouped by teams and transformed into a new dataset. The analysis was conducted on these two datasets. Another important aspect of this study is to identify the game criteria that most significantly impact the match outcome. To achieve this, feature selection was performed on the dataset to identify the most influential features. These features can assist the team in forming a strategy and put the team in a more advantageous position. The problem in this study is a classification problem. The most used machine learning algorithms for classification problems in the literature were utilized in this project. The goal is to predict the outcome of a match (win or lose) based on team data.

The remainder of the paper is structured as follows: Section two provides a comprehensive literature review on game analytics. Section three briefly describes the materials and methods used in the study, including the dataset collection process, preprocessing techniques, machine learning algorithms, and model performance evaluation criteria. In section four, we present the results of our experimental study, discuss the findings, and analyze the impact of feature selection on model performance. Finally, the conclusion summarizes the entire study.

2. Related Works

Even before the advent of computers and digitalization, data was generated from sports competitions, much like in all activities today. Analysis based on this data allowed for inferences to be made about game strategies that would give teams an advantageous position over others in these competitions.

The study conducted by Y. Yang et al. [2] stands apart from previous studies by incorporating data obtained during the game, in addition to pre-game data. This approach resulted in changes in the expected winning team, based on the in-game data. The researchers chose a logistic regression model as their prediction model and conducted their study on Dota 2. They used their trained model with real-time data and presented their results graphically. Their findings revealed that the team expected to win until the 7th minute of the game was different from the team that eventually won the game. This study illustrates how the use of in-game data can influence the accuracy of the output. However, by solely relying on logistic regression in their trained model, the researchers overlooked other models that could potentially have resulted in higher accuracy.

In their study, A. Silva and colleagues [3] aimed to compare RNN [4] models by leveraging the inherent characteristics of the data. They tested simple RNN, LSTM [5], and GRU [6] models and found that the simple RNN model had the highest accuracy rate. The researchers utilized a dataset where each row represented a minute of the game, with the goal of capturing changes in the data as the game progressed. Their results showed that the simple RNN model achieved a consistent accuracy rate of 76.29%. However, the researchers acknowledged that the model's performance may be affected by game updates and may not work as consistently.

In a separate study, Hitar-Garcia and colleagues [7] utilized pre-game data to predict the winning team in professional matches. They created new features with the aim of revealing the dynamics of player-to-player matchups and relationships. Classification

algorithms were employed in alignment with their defined objectives. However, the most critical factors for team success were not addressed.

In another related study, Q. Shen [8] conducted research on data from diamond-ranked games. Popular machine learning algorithms were employed, and a voting classifier was built to predict game outcomes. The accuracy of the voting classifier was found to be 72.68%. However, feature selection was not applied, and the impact of features on game outcomes was not elucidated.

In a study conducted by F. Bahrololloomi and colleagues [9] individual players were evaluated considering their positions and roles in the game. They developed a simple win prediction model that could predict match outcomes when given the names of ten players divided into two teams. They obtained scores for players and teams overall. By considering the highness of the team score, they made predictions and recorded an accuracy rate of 86%.

In the study conducted by T.D. Do and his colleagues, [10] they predicted game outcomes based on the champions chosen by players within the game using deep learning. They achieved a prediction accuracy of 75.1% when predicting game outcomes even before the start of the game, based on the champions chosen by the players.

A project on live professional match prediction was conducted by Victoria Hodge and her colleagues [11] using data from a different MOBA game, DotA 2. They utilized Random Forest and Logistic Regression algorithms. After a 5-minute game of DotA 2, an accuracy of 85% was achieved in the prediction of match outcomes.

In this study, data was collected from the last matches via the API platform. The dataset was analyzed for both teams and players. The classification algorithms were trained on both datasets. In addition, feature selection was performed to identify the most important factors affecting the outcome of the match. Seven different machine learning algorithms, which are commonly used in the literature for classification problems, were employed, and as a result of the trainings, a success rate of 98% was achieved.

3. Materials and Methods

3.1. Dataset

The dataset used in this study was created by obtaining game data from an online platform through the Riot API, which is provided by the game's developer. The Riot API is a tool used by developers to integrate Riot Games into their applications. Although Riot Games offers numerous APIs to researchers, only two were utilized in this project. Figure 1 illustrates the data extraction steps for the API used in this study.

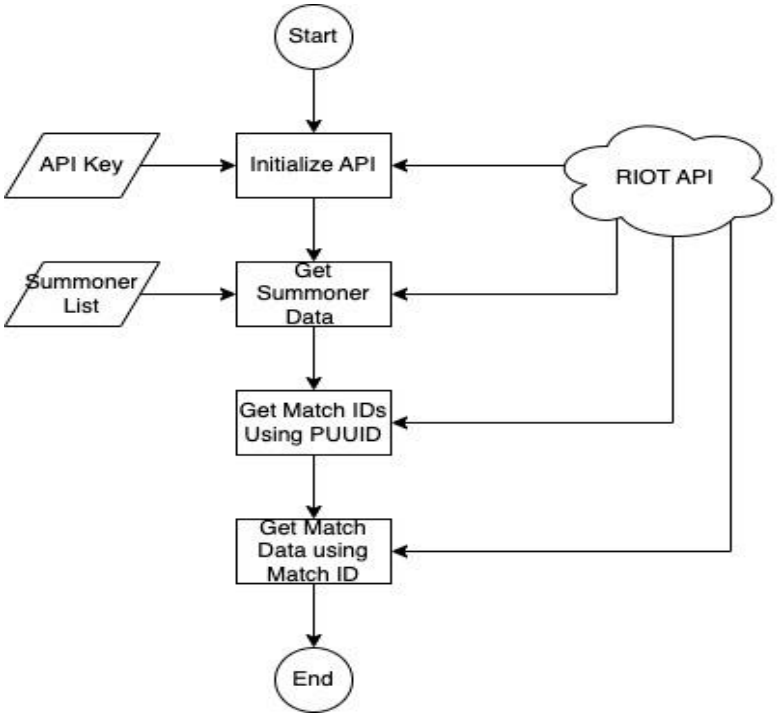

Figure 1. Data Collection with RIOT API

Summoner-v4: "Summoner v4" refers to the fourth version of the API that provides access to user account-related data in the game, League of Legends. This API version allows access to user account information, champion statistics, match history, and other account-based data. The API used in this study offers 6 different methods for obtaining summoner information. The method used in this research retrieves summoner data using the summoner name and stores the response value for retrieving the PUUID, a unique value for each summoner. This method is a GET method that requires the summoner name and region as input and returns a summoner object as response.

Match-v5: "Match v5" refers to the fifth version of the League of Legends API provided by Riot Games. This version allows access to in-game match data and enables retrieval of detailed information about matches. This API offers three methods that developers can use to retrieve information on match games. In our study, we utilized two of these methods to obtain game IDs and subsequently access each game's data. These methods are both GET methods, with one taking the PUUID as a parameter and responding with game IDs, while the other takes game IDs as a parameter and responds with the corresponding game data.

The datasets are labeled with a binary label, where 0 indicates losing team and 1 indicates winning team. The numeric features of the datasets are presented in Table 1.

Table 1. Numerical Information of The Classes in The Datasets

Dataset	Data Groups (Labels)	Data Counts	Feature Count	Total Instance Count
DS1. Team-Dataset	Loser	587		
	Winner	591	47	1,045
DS2. Player-Dataset	Loser	2,594		
	Winner	2,638	47	5,232

The dataset contains a total of 47 features. Some important attributes in the dataset and their definitions are shown in Table 2.

Table 2. Description Of Some Features

Feature Name	Description
turretsLost	The number of towers lost
turretKills	The number of destroyed towers.
inhibitorKills	The number of destroyed inhibitors.
inhibitorTakedowns	The number of inhibitors destroyed by the player.
largestKillingSpree	The highest killing spree count.
deaths	The number of deaths of the player.
damageDealtToObjectives	Damage dealt to objectives.
totalTimeSpentDead	The time spent dead in the game.
kills	The number of kills.

When examining the data distribution based on classes in the dataset, it is known that the current dataset is balanced, meaning that the data is evenly distributed among different classes. The dataset was split into training and test data with a test data ratio of 20%. The data used in the test set was not used in any way in the training set. The 80/20 ratio [12] is often used because it provides a reasonable balance between having enough data for training a machine learning model and having enough data for evaluating the model's performance.

The allocation of 80% of the data as the training dataset allows the model to learn the underlying patterns and relationships. The remaining 20% serves as an independent test dataset to evaluate the model's performance and assess its ability to generalize to unseen data. This ratio was chosen based on the size of the dataset.

3.2. Pre-processing

In this stage of the study, the game data collected with the RIOT API was pre-processed to ensure that the classification models would produce accurate results. Firstly, the attributes in the dataset were examined separately, and missing values were detected in some of the attributes; if these missing values exceeded 80%, they were deleted. The "platform id" and "game id" attributes in the dataset were combined into a single column, and the dataset was grouped based on this column to obtain a team-based dataset. As a result, the DS1 dataset based on teams and the DS2 dataset based on players were obtained for model training.

3.3. Machine Learning Algorithms

The use of machine learning algorithms in game analytics has been increasingly prevalent in recent years. In this study, after preprocessing steps were completed on the dataset, various categories of machine learning classification algorithms were applied to DS1 and DS2 datasets. The selection of classification algorithms for this study was

based on a literature review of commonly used algorithms in the field. Ensemble learning algorithms were also included among the chosen algorithms. The classification algorithms used in this study were as follows: Random Forest [13], Decision Tree [14], Logistic Regression [15], LightGBM [16], Naive Bayes Classifier [17], Gradient Boosting [18], and AdaBoost [19]. These algorithms have different approaches and advantages to solve classification problems. Random Forest is an ensemble learning algorithm and one of its features is feature selection, which measures the impact of each feature on prediction. The system workflow is illustrated in Figure 2.

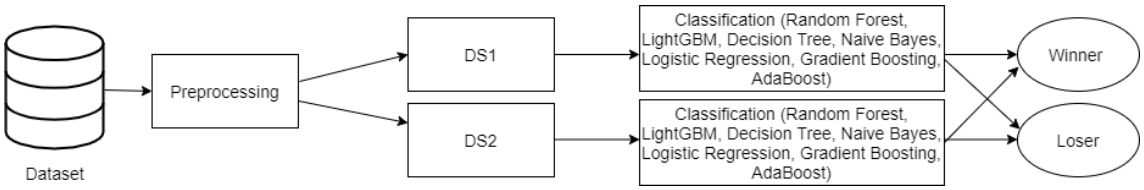

Figure 2. Operating schema of the system

3.4. Evaluation Criteria

To evaluate the accuracy of the system that performs classification using machine learning algorithms, a confusion matrix was used. The confusion matrix is a commonly used evaluation matrix in classification problems to assess the performance of a model. It aids in evaluating the performance of a model by comparing the true class labels with the predicted class labels. Table 3 shows the structure of a two-class (positive, negative) confusion matrix [20].

Table 3. Confusion Matrix

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

In this study, the performance evaluation metric of accuracy, which can be calculated from the confusion matrix, was utilized to assess the performance of the models. One of the reasons for using this metric is that the dataset is balanced. Accuracy is a commonly used metric to measure the performance of a model. The accuracy value is calculated by the ratio of the total number of correctly predicted classes in the model to the entire dataset. True Positive and True Negative refer to the areas where the model correctly predicted, while False Positive and False Negative refer to the areas where the model incorrectly predicted. The equation for the accuracy metric used to evaluate the model's performance is shown in Equation 1.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad [1]$$

4. Experimental Study and Results

The selected machine learning algorithms were trained with the preprocessed datasets. When examining the results of this training, the most successful model among the team-based dataset was the AdaBoost algorithm with an accuracy rate of 0.9847. On the other hand, the most successful models among the player-based dataset were the Gradient Boosting and LightGBM algorithms with an accuracy rate of 0.95. The accuracy rates of the trained machine learning models are provided in Table 4.

Table 4. Performance Comparison of Models

Algorithm Name	DS1	DS2
Random Forest	0.9732	**0.9533**
Decision Tree	0.9503	0.9388
Naive Bayes	0.8015	0.7265
Logistic Regression	0.8969	0.7624
Gradient Boosting	0.9656	0.9541
LightGBM	**0.9770**	**0.9541**
AdaBoost	**0.9847**	0.9526

The confusion matrices of the top 2 models with the highest accuracy rates for both DS1 and DS2 datasets have been shown. Figures 3 and 4 represent the confusion matrices for the player dataset, while Figures 5 and 6 represent the confusion matrices for the team dataset.

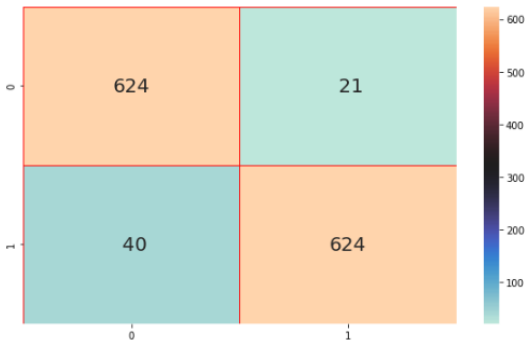

Figure 3. Random Forest

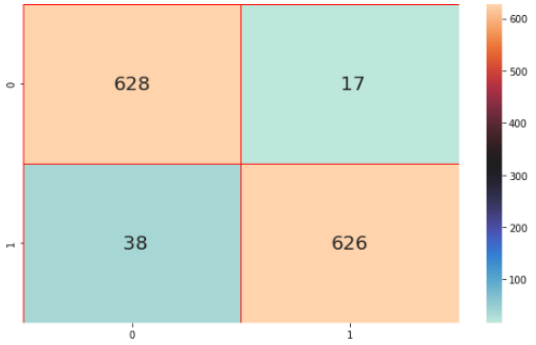

Figure 4. Gradient Boosting

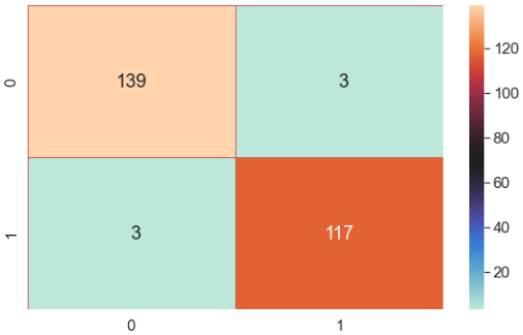

Figure 5. LightGBM

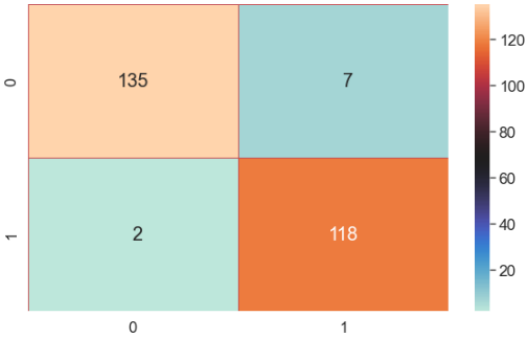

Figure 6. Gradient Boosting

Although the current dataset is not a very large dataset, it is a dataset with a high concentration of numerical data. Upon examining the structures of the algorithms used, their performance, and considering the current dataset, Random Forest and Gradient Boosting algorithms have emerged as prominent options in the study. Both of these models are ensemble models. Random Forest is an ensemble method that combines

multiple decision trees to create a prediction model. It offers resistance to noise in the dataset and provides high prediction accuracy with low training time. On the other hand, Gradient Boosting is an ensemble method that progressively improves prediction models and achieves high prediction accuracy. It also provides resistance to noise and is tailored for numerical data.

4.1. Feature Selection

When examining the team dataset, it is known that the total number of data points is 1045 and the dataset contains 47 features. Feature selection is the process of reducing the number of input variables when developing a prediction-based model. It is desirable to reduce the number of input variables to decrease the computational cost of modeling and, in some cases, improve the model's performance [21]. The decision tree algorithms used in the study prune the branches of the tree based on the importance of the input variable.

In this study, a Gini score-based algorithm was used for feature selection, and the top 10 features that have the most impact on classification (Figure 7) were selected to train models and calculate their accuracy values.

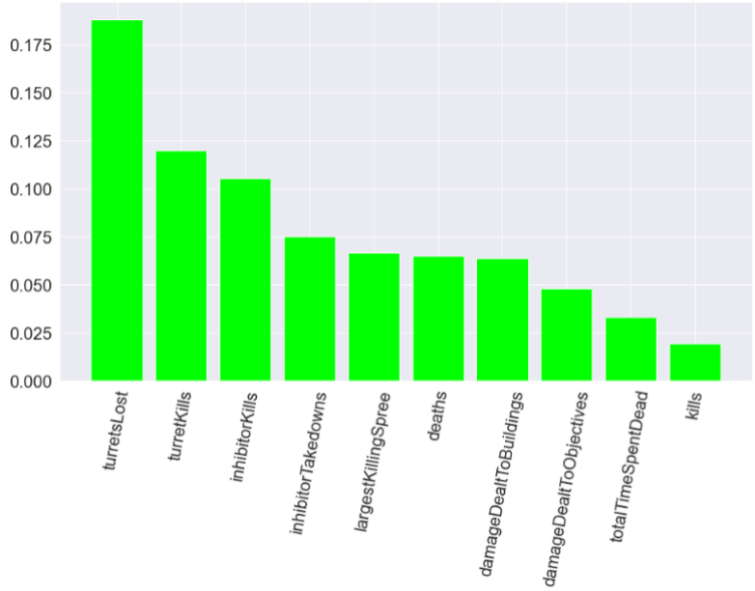

Figure 7. The Results of Gini Score-Based Feature Selection

When analyzed, 10 factors that have the most impact on the outcome of the game can be observed. These factors indicate the qualities that a team should possess against their opponents during the match. Teams can devise strategies based on these qualities.

4.2. The Effect of Feature Selection

In this study, feature selection was performed on a data set with 47 attributes to aim for model training with fewer features. Out of the 7 models trained with the team data set, performance improvement was observed in 5 models. The most significant performance

increase was observed in the Naïve Bayes and Logistic Regression algorithms. According to the accuracy values obtained after the feature selection process, the most successful models were Logistic Regression and Gradient Boosting, as seen in Table 5.

Table 5. Performance Comparison of Algorithms After Feature Selection

Algorithm Name	Accuracy Value Before Feature Selection	Accuracy Value After Feature Selection
Random Forest	0.9732	0.9809
Decision Tree	0.9503	0.9618
Naive Bayes	0.8015	0.9656
Logistic Regression	0.8969	0.9847
Gradient Boosting	0.9656	0.9847
LightGBM	0.9770	0.9770
AdaBoost	0.9847	0.9809

4.2.1. Comparison of Confusion Matrices for Naïve Bayes

After the feature selection process, it was observed that the accuracy value of the Naïve Bayes model increased by 0.16%. The confusion matrices of the algorithm before and after feature selection are shown in Figure 7 and Figure 8, respectively.

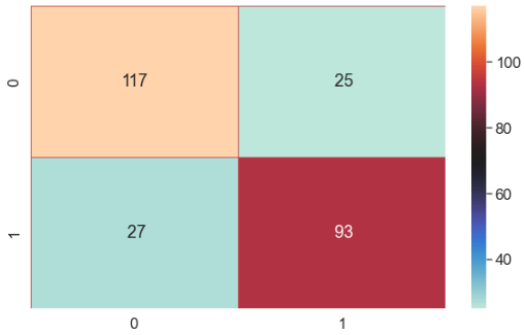

Figure 7. Before Feature Selection

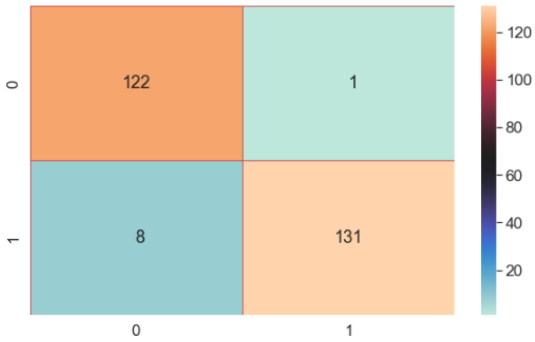

Figure 8. After Feature Selection

It can be observed that the Naïve Bayes algorithm made successful predictions on 210 out of 262 test data before feature selection. After feature selection, it made successful predictions on 253 out of 262 test data. This indicates that the performance of the model has improved after feature selection, as evident in the results.

4.2.2. Comparison of Confusion Matrices for Logistic Regression

After the feature selection process, it was observed that the accuracy value of Logistic Regression model increased by 0.16%. The confusion matrices of the algorithm before and after feature selection are shown in Figure 7 and Figure 8, respectively.

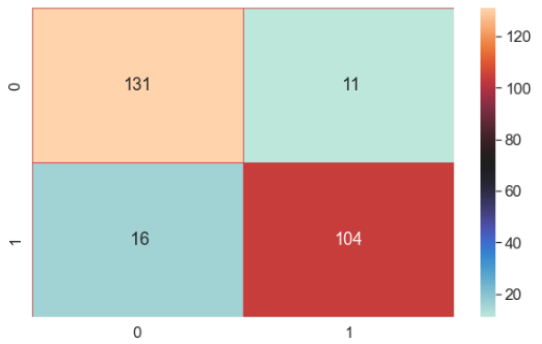

Figure 9. Before Feature Selection

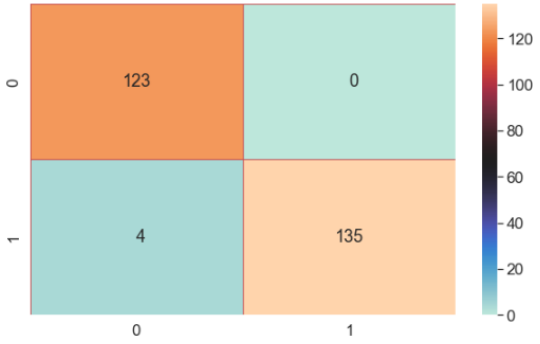

Figure 10. After Feature Selection

4.2.3. Comparison of Confusion Matrices for Gradient Boosting

It has been observed that the accuracy value of the Gradient Boosting algorithm, which is one of the most successful models, increased by 0.01% after the feature selection process. The confusion matrices before and after the feature selection are shown in Figure 11 and Figure 12, respectively.

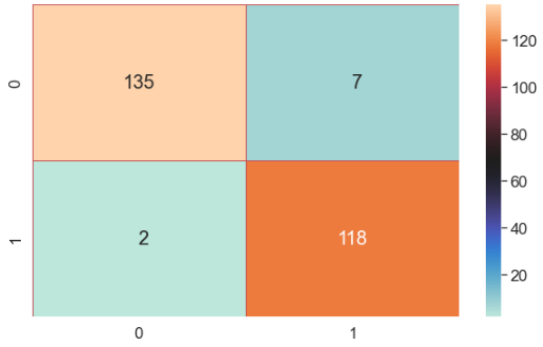

Figure 11. Before Feature Selection

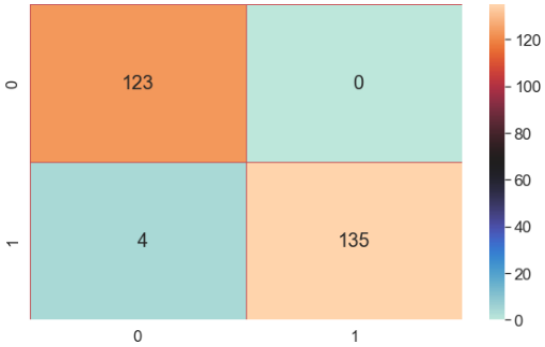

Figure 12. After Feature Selection

5. Conclusion and Future Works

The aim of this study is to predict the outcome of League of Legends games using historical game data obtained through the official API provided by Riot Games. The game data presents a classification problem, and machine learning models including Random Forest, Decision Tree, Logistic Regression, Light GBM, Naive Bayes Classifier, Gradient Boosting, and AdaBoost algorithms were used for classification. The highest accuracy rate of 98.41% was achieved with the AdaBoost algorithm on the team dataset. It was observed that selecting important features and training models with these features can result in high performance and using only 21% of the features in the dataset reduces the workload of the model. After the feature selection process, Logistic Regression and Gradient Boosting were identified as the most successful algorithms with an accuracy rate of 98.41%. It was also observed that the same accuracy rate was achieved with the AdaBoost algorithm without the feature selection process.

The result of this study clearly shows that identifying the most influential features on the game outcome through feature selection provides teams with insights for planning their

strategies. Moreover, higher accuracy scores were obtained in machine learning with the support of the feature selection process.

In the future, deep learning models can be constructed and optimized to achieve higher success rates for classification. Moreover, more comprehensive and complex models can be trained with real-time data flow to improve the accuracy of game outcome predictions. Strategies can be provided to players during gameplay.

References

[1] Mora-Cantallops, M., & Sicilia, M. Á. (2018). MOBA games: A literature review. Entertainment computing, 26, 128-138.

[2] Yang, Y., Qin, T., & Lei, Y. H. (2016). Real-time e-sports match result prediction. arXiv preprint arXiv:1701.03162.

[3] Silva, A. L. C., Pappa, G. L., & Chaimowicz, L. (2018). Continuous outcome prediction of league of legends competitive matches using recurrent neural networks. In SBC-Proceedings of SBCGames (pp. 2179-2259).

[4] Medsker, L. R., & Jain, L. C. (2001). Recurrent neural networks. *Design and Applications*, *5*, 64-67.

[5] Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586.

[6] Dey, R., & Salem, F. M. (2017, August). Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS) (pp. 1597-1600). IEEE.

[7] Hitar-Garcia, J. A., Moran-Fernandez, L., & Bolon-Canedo, V. (2022). Machine learning methods for predicting league of legends game outcome.

[8] Shen, Q. (2022, February). A machine learning approach to predict the result of League of Legends. In 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE) (pp. 38-45). IEEE.

[9] Bahrololloomi, F., Sauer, S., Klonowski, F., Horst, R., & Dörner, R. (2022). A Machine Learning based Analysis of e-Sports Player Performances in League of Legends for Winning Prediction based on Player Roles and Performances. In VISIGRAPP (2: HUCAPP) (pp. 68-76).

[10] Do, T. D., Wang, S. I., Yu, D. S., McMillian, M. G., & McMahan, R. P. (2021, August). Using machine learning to predict game outcomes based on player-champion experience in League of Legends. In Proceedings of the 16th International Conference on the Foundations of Digital Games (pp. 1-5).

[11] Hodge, V. J., Devlin, S., Sephton, N., Block, F., Cowling, P. I., & Drachen, A. (2019). Win prediction in multiplayer esports: Live professional match prediction. IEEE Transactions on Games, 13(4), 368-379.

[12] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21(1), 1-13.

[13] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[14] Freund, Y., & Mason, L. (1999, June). The alternating decision tree learning algorithm. In icml (Vol. 99, pp. 124-133).

[15] LaValley, M. P. (2008). Logistic regression. Circulation, 117(18), 2395-2399.

[16] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

[17] Murphy, K. P. (2006). Naive bayes classifiers. University of British Columbia, 18(60), 1-8.

[18] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

[19] Schapire, R. E. (2013). Explaining adaboost. In Empirical inference (pp. 37-52). Springer, Berlin, Heidelberg.

[20] Liang, J. (2022). Confusion Matrix: Machine Learning. POGIL Activity Clearinghouse, 3(4).

[21] Fashoto, S. G., Mbunge, E., Ogunleye, G., & den Burg, J. V. (2021). Implementation of machine learning for predicting maize crop yields using multiple linear regression and backward elimination. Malaysian Journal of Computing (MJoC), 6(1), 679-697.