Predicting Win Rates in League of Legends:

A Comparative Study of Machine Learning Algorithms

Amitanshu Ghosh

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COGNITIVE SCIENCE AND ARTIFICIAL INTELLIGENCE

AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES OF TILBURG UNIVERSITY

STUDENT NUMBER

2027054

COMMITTEE

Dr. Eric Postma

Dr. Sharon Ong

LOCATION

Tilburg University School of Humanities and Digital Sciences

Department of Cognitive Science & Artificial Intelligence

Tilburg, The Netherlands

DATE

Jan 15th, 2024

ACKNOWLEDGEMENTS

I would like to thank Dr. Eric Postma for his time and always simplifying concepts that were hard to grasp. His valuable feedback and check-ins have helped me improve and complete this thesis.

DATA SOURCE, ETHICS, CODE AND TECHNOLOGY STATEMENT

The data for this thesis is publicly available and was sourced from the Riot Games API. All changes to the data were made by me in order to facilitate the experimentation process.
All the tables and images were made by me.
There was occasional use of grammarly to check for grammatical errors in this thesis and chat gpt was used to generate a skeleton format for this thesis.

Predicting Win Rates in League of Legends: A Comparative Study of Machine Learning Algorithms

Amitanshu Ghosh

Abstract

This paper explores the application of various machine learning models in prediction win rates in League of Legends (LoL), a popular Multiplayer Online Battle Arena (MOBA). It specifically examines the effectiveness of Logistic Regression, K Nearest Neighbors, Gradient Boosting Classifier and Random Forest Classifier. With a dataset consisting of 10,000 games played by Diamond level players, this study evaluates the performance of these models based on their accuracy, training time and their F1 score. The findings of this study demonstrate that simpler models, post feature selection, can offer high accuracies and low training times. This study also highlights the importance of feature selection and the role it plays in determining model efficiency.

1. Introduction

In recent years, as video games have become a widespread form of entertainment, League of Legends (LoL) has emerged as one of the most played Multiplayer Online Battle Arena (MOBA) games with around 180 million monthly active players. This rise in popularity for video games led to a thriving esports scene with professional teams competing for millions of dollars in prize money. As the esports scene is continuously growing, teams and coaches must implement newer methods and metrics to analyze games in order to help their players achieve victory. Some of these methods involve creating player profiles and analyzing previous games to help predict the factors that contribute to a victory.

League of Legends (LoL) is a MOBA developed by Riot games and released in 2009. The game's premise revolves around two teams, each team consisting of five players, competing in a virtual arena with the primary objective being to destroy the opposing team's "Nexus", a core building which is located within the opposing base.

Each player can select and control a unique character, also known as “Champions”, selected from a roster of 150+ champions each with their own unique playstyle and mechanics. These champions gain strength over the course of a game by accumulating experience, gold and purchasing items which enhance their combat abilities.
The map in League of Legends, known as “The Summoner’s Rift”, is a symmetrically divided arena consisting of two halves with three main paths or “lanes” and an area known as the “jungle” consisting of trees and monsters.

Figure 1: A LoL map with the highlighted red and blue sides.

Players engage in combat against enemy players in their lanes, AI controlled minions and neutral monsters all in an effort to gain more experience, gold and control over the game and eventually destroy the opposing Nexus.
The strategic depth of the game is evident when one takes into consideration all the champions, their unique playstyles and the balance of power levels among the different champions. Each decision made by a player during a fight, which item they purchase and when they perform certain actions can lead to a multitude of different outcomes.
Accurately predicting win rates in LoL is a difficult task but it can result in certain players and teams identifying factors that contribute to a victory and those that are less important. This paper aims to examine the application of different Machine Learning (ML) models, ranging from

simplistic models to more complex ones, and drawing comparisons between them. Specifically, we will be investigating the performances of different ML models and explore the trade offs between model complexity and predictive accuracy. Finally, we will identify the key features that significantly contribute to a higher win rate.

However, predicting win rates is a challenging task due to the dynamic nature of the game and the multitude of factors that can influence the outcome of a match. There are certain milestones, when achieved by a team, that can drastically increase their chances of winning a game. Some of these milestones include having a gold lead, which is where a team with more money (gold) gains a significant advantage; a kill lead, where a team that has eliminated more enemy champions and in turn gain more control over the game; and a turret difference, where a team that has destroyed more turrets gain control of the game and get closer to destroying the enemy Nexus. By identifying these milestones and understanding their significance, we can develop ML models that are more accurate when it comes to predicting win rates.

Furthermore, this paper will explore the nuanced relationship between ML model complexity and their predictive accuracy. By comprehending the trade offs, we will guide others in selecting the appropriate models for their respective tasks. The feature analysis performed in this paper also greatly contributes to our understanding of the underlying factors that influence LoL game outcomes which provide developers, coaches and players with more insight regarding the game.

In summary, this paper endeavors to address questions in the field of predictive modeling, in particular for win rates in League of Legends. The research question guiding this study is as follows:

What machine learning models demonstrate the highest predictive accuracy in forecasting win rates in League of Legends?

This research question will guide the following chapters and help provide a focused and systematic approach to answering the intricacies of predictive modeling in the ever changing context of League of Legends and esports.

2. Related Work

League of Legends, a leader in the MOBA genre, presents unique challenges for machine learning and predictive modeling applications. The complexity of the game, due to its dynamic nature and the multitude of factors that influence outcomes, makes it a challenge for those who wish to predict win rates and understand the factors that influence win rates. This section of the paper will review and discuss existing literature that have delved into predicting win rates in LoL and what methods have been already employed to accurately predict win rates.

Silva et al. (2018) employed Recurrent Neural Networks to help them predict the outcomes of LoL matches. Their data consisted of minute-by-minute game data from a large sample of competitive matches. Afterwards, they examined the accuracy of the winning prediction with respect to the time frame from which the data was drawn. This resulted in them comparing win rate accuracies at the ten and twenty minute mark respectively. They achieved similar win rate predictions with an accuracy of up to 83.54% across these two time frames. However, their accuracy of winning predictions decreased as the time windows were shifted forward. They based their study on Deep Learning Methods in contrast to the Machine Learning Methods that this paper will be exploring. Their dataset also consisted of minute to minute data whereas this paper will focus on predicting win rates at the 10 minute mark given the features that are available to us in that time frame.

Another study by Arik (2023) used Light Gradient Boosting Machines (LightGBM) in an attempt to optimize on computational load while still retaining the high accuracy. They also simultaneously used Logistic Regression, Support Vectors and Gradient Boosting Classifiers to compare performances and results. The LightGBM model performed the best with an accuracy of 96.8% in 4.88 seconds. Their other models performed similarly when predicting accuracies however, there were large differences in the time taken by each model. Arik (2003) concluded that since the predictive accuracy is relatively similar across a multitude of models it is better to choose algorithms that are less time consuming. This study emphasized on models that are already known to have high accuracy ratios and a significant aspect of the research was on the player and in-game metrics.

Another study utilized a Deep Neural Network model system and processed around 26,000 matches to predict outcomes in League of Legends No et al. (2021). They employed features such as “Dragon Gap”, “Level Gap”, “Blue Rift Heralds” and “Tower Kills” and the model achieved an accuracy of 93.75% in predicting the 2020 League of Legends world championship. This study demonstrates the potential and upsides of integrating deep learning into esports analytics. However, this study utilized mid game data from the 15 minute mark rather than real time predictions and individual data which highlights the effectiveness of a more holistic approach to predicting match outcomes.

Bahrololloomi et al (2023) predicted LoL outcomes by focusing on player roles rather than match features. This study employs a series of Gradient Boosting Models in an attempt to compute an overall score based on player variables rather than match variables. This study concluded that the differences in player roles have no significant influence on the outcome of the game but their heuristic performance metric can predict game outcomes with a 86% accuracy.

A study by Lin (2016) investigated predicting match outcomes with the use of pre game knowledge such as player experience and mastery alongside in game statistics. They created feature vectors for Gradient Boosting Trees and Logistic Regression models and concluded that pre game knowledge alone is a weak predictor. The incorporation of in game statistics significantly improves the predictive strength of a model and it highlights how different types of data affect predictive models for League of Legends.

Ani et al. (2019) also investigated the possibilities of win rate prediction in LoL with the help of a random forest model. They took into consideration pre match features along with within match features for their predictions. Some of the features used were Champion picks and bans, summoner spells and player history. With the random forest model, they achieved an accuracy of 95.52% with the pre match features and an accuracy of 98.18% with the within match features.

Another similar study by Khromov et al. (2019) on the game Counter Strike: Global Offensive (CSGO) compared professional player’s biometric skill rather than any particular pre game or post game features. This study considered features such as reaction times and dynamics of pressing certain keys. Their model was able to predict a player’s skill with an accuracy of 90% but this failed to capture player knowledge and only considered raw skill. Both studies were carried out on professional players who are at the pinnacle of skill and therefore the results cannot be attributed to the general population.

Work done by Hodge et al. (2021) uses real time match evaluations for a game known as Dota 2 which is also a MOBA and similar to LoL. This real time evaluation is also present in other sports such as chess with the use of engines such as *Stockfish*. These real time evaluation methods help casters and spectators with gaining an understanding of the game state and the direction it may progress in.

These studies highlight how important predictive modeling is in the field of e-sports and the multitude of factors and variations one might encounter when attempting to build a model of their own. Each study offered us new insights regarding modeling and which features contribute significantly and those which do not.

This paper contributes to this field by performing an analysis of multiple machine learning models and determining which model provides us with the best accuracy while not being as computationally demanding. It also aims to highlight the features that significantly affect win rate in a game of League of Legends.

3. Dataset and Methods

3.1 Overview

The methods section of this thesis covers the approach that was adopted for the exploration and analysis of the dataset. Pandas and NumPy were instrumental in data preprocessing while SciPy simplified the chi squared test in order to determine the importance of the features in the dataset. A random forest classifier was also used to determine and visualize the significance of the features against each other. Finally, PyCaret was used in order to streamline the experimentation process.

The primary objectives of this section are to preprocess the data, identify the significant features and visualize their interactions which provides us with insights regarding their influence on predicting win rates. These steps will result in a dataset that is well structured and also contains significant features which makes it appropriate for the predictive modeling tests.

3.2 Dataset

The data acquisition process for this study involved sourcing player game data from the Riot Games API. The dataset particularly focuses on 10,000 games consisting of players ranked Diamond in League Of Legends ranked modes. The riot games API is free to use for all players and provides us with intricate details such as player statistics and game statistics sorted by teams for easier data processing. The choice of Diamond plus games was made in order to collect data from players who have demonstrated a certain level of skill and understanding when it comes to in-game mechanics and therefore can provide us with reliable data about the diverse game related features. Leveraging the data set from the Riot Games API ensures a lever of authenticity and accuracy to the dataset and no missing values as it records information directly from authentic games that have been played by players.

3.3 Analysis and Feature Selection

The dataset consists of 10,000 ranked games of Diamond rating and above. The data covers various features related to League of Legends (LoL) games, including Blue and Red team statistics, outcomes and other relevant features.

The dataset initially contained 40 features which are evenly split between the Red and Blue teams. While all features do contribute to predicting win rates, we are identifying features that significantly contribute to a victory. To help us identify the significant features, we perform a chi square test on the dataset to determine which features will remain for the predictive modeling section of the experiment.

A chi- square test is a statistical test which is used to determine if a significant association exists between two categorical variables. If we acquire a p <= 0.05 we can consider the results to be statistically significant which in turn suggests a significant association between the variables. This helps us retain the features which are more closely associated with winning the game ("blueWins"). The chi square test indicated that out of the 40 total features, only 22 demonstrated significant contribution towards win rate. These features are, as usual, split evenly between the Red and Blue teams. The features we will be retaining for the predictive modeling section are as follows:

Blue Team Features: blue wins, blue wards destroyed, blue first blood, blue kills, blue deaths, blue assists, blue elite monsters, blue dragons, blue heralds, blue towers destroyed, blue average level.

Red Team Features: blue wins, red wards destroyed, red first blood, red kills, red deaths, red assists, red elite monsters, red dragons, red heralds, red towers destroyed, red average level.

The column containing "gameID" was dropped because it was not a contributing factor to predicting win rates. The data was then divided into two sets. One set contained features from the Red team and the other set contained features from the Blue team. Both sets, however, contained "blueWins" because this column displays if a game resulted in a victory or a loss for the Blue team.

A Random Forest Classifier (RFC) model was also employed to help evaluate the importance of the features when predicting victories. RFCs are a form of machine learning algorithm that leverages an ensemble of decision trees for classification tasks. The model was trained twice: once using Blue team features and then using Red team features. In both cases, predicting "blueWins". The graphs below illustrate the outcomes of the RFC and provide information regarding which features are determined to have a more significant impact on the model's predictions.

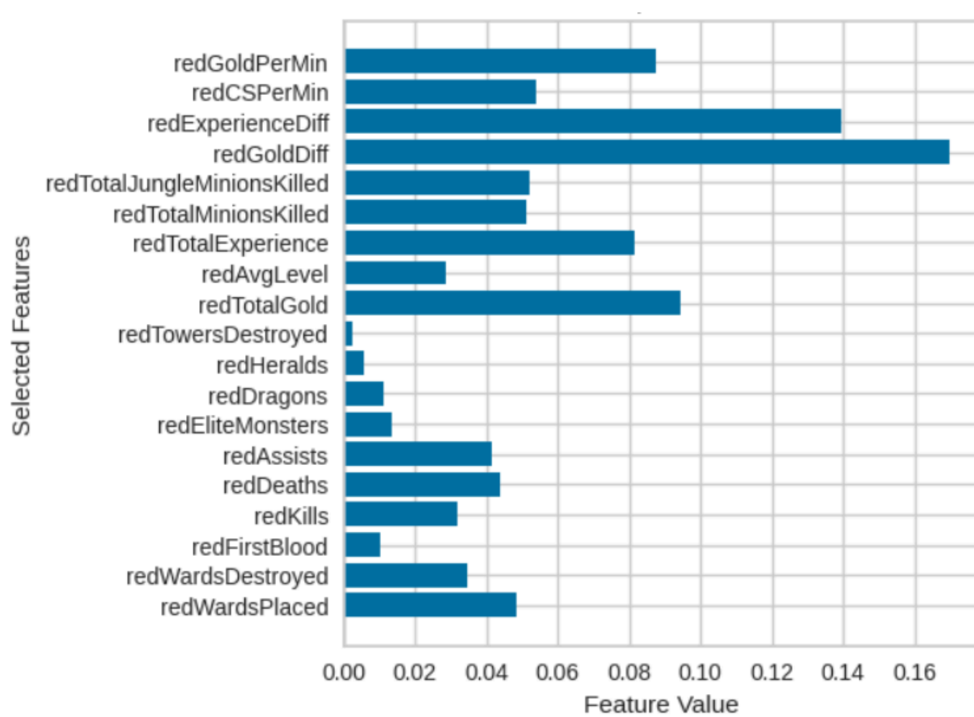

Figure 2: Red team features and their Feature Values

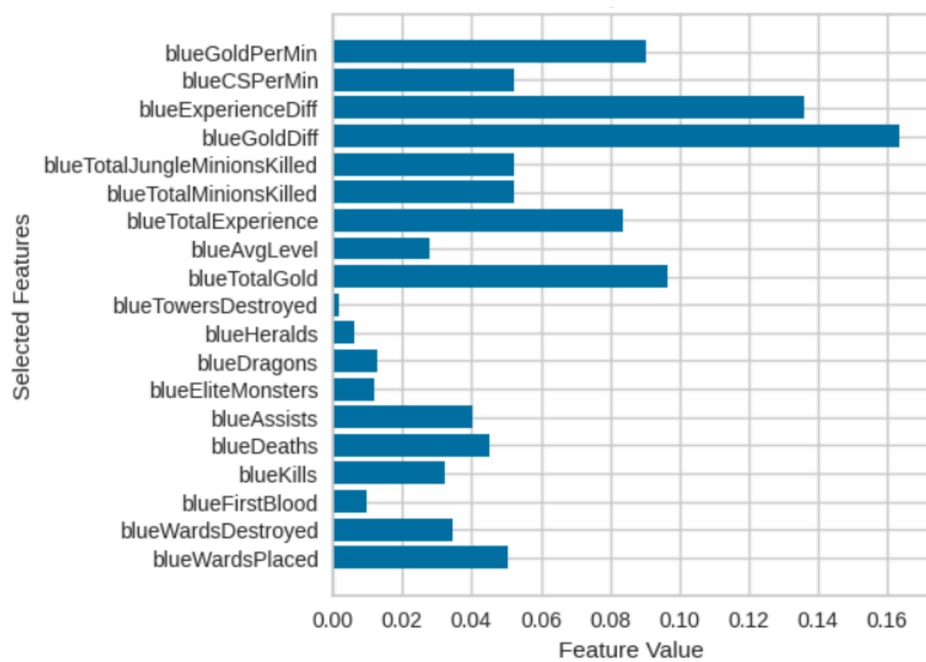

Figure 3: Blue team features and their Feature Values

The Feature Values indicate how significant a feature is in predicting “blueWins”. In this case, both teams have very similar Feature Values attached to their features and the ranking of features for both Red and Blue teams are the same.

3.4 Predictive Modeling

The PyCaret library was used for efficient and effective machine learning model development. The target variable was set to “blueWins” and three experiments were carried out. The first experiment used the dataset containing both red and blue team features to predict “blueWins”. The second experiment was carried out with a dataset containing only blue team features. The third experiment utilized a dataset with only red team features. This was done to see if there would be any significant changes in the outcome of “blueWins” when the dataset contained different types of features.

PyCaret enables us to run multiple ML models one after another with minimal setup. This is possible due to the *compare_models* function and it provides us with the top ten best performing models for our dataset.

After discovering the best performing model, we are able to evaluate the model through metrics such as accuracy, area under the curve, recall, precision and F1 score with *finalize_model.* Finally, we utilize *predict_model* to make predictions for the respective datasets based on the finalized model. These results were then analyzed to help us draw meaningful insights into the predictive capabilities of the chosen model.
The models that we chose to examine for this study were Logistic Regression, K Nearest Neighbors, Gradient Boosting and Random Forest classifier. This was done in order to examine the benefits and drawbacks of using simpler models in contrast to more intricate ones.

4. Results

This section of the paper will focus on the results generated by each of the Machine Learning models. Each model will be run once on three data sets: the Dual feature (DF) set containing features from both Red and Blue teams, the Blue features (BF) set which contains select features from the Blue team and the Red features (RF) set which contains select features from the Red team.

This was done so that we could easily compare performance within the same model across different data sets and make appropriate comparisons and conclusions.

4.1 Logistic Regression

Logistic regression is a statistical method employed when our dataset contains categorical dependent variables and it is commonly used for linear classification tasks. It is one of the more accessible and interpretable algorithms due to its linear nature which results in it being faster and less computationally demanding than other algorithms.

Log Reg	**Accu.**	**AUC**	**Recall**	**Prec.**	**F1**	**Kappa**	**TT (sec)**

Dual Features	0.7289	0.8056	0.7276	0.7285	0.7279	0.4577	0.8780
Blue Features	0.7319	0.8106	0.7337	0.7312	0.7321	0.4638	0.0430
Red Features	0.7296	0.8098	0.7334	0.7281	0.7303	0.4592	0.1180

Table 1: Results of the Log Reg Across all sets.

Using Logistic Regression, the results from our analysis show varied performance across the different feature sets. The DF set, which includes features from both teams, had an Accuracy of 72.89%, with an Area Under the Curve (AUC) of 0.8056. This suggests that the model has a strong ability to differentiate between the classes that are present. The Recall was 0.7276, Precision was 0.7285 and the F1 score was 0.7279. However the Training Time (TT) for this set was 0.8780 seconds.

In contrast, the BF set which only contained features from the Blue team achieved an Accuracy of 73.19%. The AUC of 0.8106 was the highest among the three sets which demonstrates that it had the best discriminatory power. The model also reported a Recall of 0.7337, a Precision of 0.7312 and a F1 score of 0.7321. This was also the most computationally efficient set with a TT of 0.0430 seconds.

Finally, the RF set, which has features from the Red team, achieved an accuracy of 72.96% which is relatively close to the other two sets. The AUC of 0.8098 still indicates a strong capability of distinguishing between classes. The Recall was 0.7334, Precision was 0.7281 and the F1 score was 0.7303. The TT for the Red features set was 0.1180 seconds which is faster than the DF set but slower than the BF set.

The BF set provided us with the highest Accuracy, AUC and F1 score which indicates a good balance between model sensitivity and precision. It also had the lowest TT of the three sets. The DF set did not significantly outperform the BF and RF and yet it had a substantially longer TT.

These findings highlight the efficiency of using team specific features for Logistic Regression models for this particular classification task.

4.2 K Nearest Neighbors

The K Nearest Neighbors (KNN) algorithm is a non parametric and supervised learning classifier which makes use of proximity to help make classifications or predictions about the positions of individual data points.

KNN	Accu.	AUC	Recall	Prec.	F1	Kappa	TT (sec)
Dual Features	0.6821	0.7451	0.6766	0.6842	0.6799	0.3643	0.3690
Blue Features	0.6548	0.7065	0.6462	0.6573	0.6515	0.3096	0.092
Red Features	0.6646	0.7198	0.6618	0.6652	0.6633	0.3293	0.1700

Table 2: Results of the KNN Across all sets.

The same three sets were utilized to help evaluate the performance of the KNN classifier.

The DF set yielded an accuracy of 68.21% with an AUC of 0.7451. Recall, Precision and F1 were 0.6766, 0.6842 and 0.6799 respectively. The TT for the DF set was 0.3690 seconds which is the longest among the three sets.

The BF set had an accuracy of 65.48% with an AUC of 0.7065. Recall, Precision and F1 were 0.6462, 0.6573 and 0.6516 respectively. The TT for the BF set was 0.092 which makes it the most efficient in terms of computational time.

The RF set had an accuracy of 66.46% and an AUC of 0.7198. Recall, Precision and F1 were 0.6618, 0.6652 and 0.6633 respectively. The TT for the RF set was 0.1700 which makes it the most efficient in terms of computational time.

The RF and BF sets had comparable Accuracy and other metrics but the BF set demonstrated a significantly lower TT than the other two sets. The DF set had the longest training time yet this did not yield improved performance metrics. This highlights that a combined feature set may not necessarily result in the highest accuracy predictions in the context of this KNN algorithm.

4.3 Gradient Boosting Classifier

Gradient Boosting Classifiers are a ML technique used to combine several weak learners into stronger learners. Each new model is trained to minimize the loss functions using gradient descent. This process is repeated until we meet a certain stopping criteria.

GBC	Accu.	AUC	Recall	Prec.	F1	Kappa	TT (sec)
Dual Features	0.7271	0.8080	0.7163	0.7315	0.7236	0.4542	3.1420
Blue Features	0.7105	0.7876	0.7160	0.7080	0.7117	0.4210	0.4930
Red Features	0.7093	0.7833	0.7039	0.7114	0.7074	0.4186	0.5840

Table 3: Results of the GBC Across all sets.

The DF set for the Gradient Boosting Classifier yielded an Accuracy of 72.71%. This set had the highest AUC at 0.8080. The Recall, Precision and F1 metrics were 0.7163, 0.7315 and 0.7236 respectively. The DF set however, required a substantial TT at 3.1420 seconds.
The BF set achieved an accuracy of 71.05% with an AUC of 0.7876. The Recall, Precision and F1 metrics were 0.7160, 0.7080 and 0.7117 respectively. The TT for this set was the lowest at 0.4930 seconds.
The RF set achieved an accuracy of 70.93% with an AUC of 0.7833. The Recall, Precision and F1 metrics were 0.7039, 0.7114 and 0.7074 respectively. The TT for this set was 0.5840 seconds.
While the DF set had the highest accuracy and AUC, it also required the longest TT. While the BF set was a little less accurate the TT was significantly lower which might make it better to use if computational resources are limited.

4.4 Random Forest Classifier

Random Forest Classifiers are ensemble learning methods and work by creating a multitude of decision trees at the time of training and making predictions by aggregating the results from its diverse trees.

RFC	Accu.	AUC	Recall	Prec.	F1	Kappa	TT (sec)
Dual Features	0.7216	0.7957	0.7024	0.7297	0.7156	0.4432	2.9470
Blue Features	0.6762	0.7380	0.6792	0.6750	0.6769	0.3524	0.7950
Red Features	0.6791	0.7434	0.6673	0.6833	0.6749	0.3582	0.6620

Table 1: Results of the RFC Across all sets.

The DF set for the RFC yielded an Accuracy of 72.16%. This set had the highest AUC at 0.7957. The Recall, Precision and F1 metrics were 0.7024, 0.7297 and 0.7156 respectively. The DF set had the longest TT at 2.9470 seconds.

The BF set achieved an accuracy of 67.62% with an AUC of 0.7380. The Recall, Precision and F1 metrics were 0.6792, 0.6750 and 0.6769 respectively. The TT for this set was 0.7950 seconds.

The RF set achieved an accuracy of 67.91% with an AUC of 0.7434. The Recall, Precision and F1 metrics were 0.6673, 0.6833 and 0.6749 respectively. The TT for this set was the lowest at 0.6620 seconds.

The DF set had the best accuracy and AUC however, it also had the longest TT which may be a trade off when resources and time are limited. The BF and RF sets, while having lower accuracy, offered much better computational efficiency which could make them suitable choices in the right circumstances.

4.5 Evaluation

In order to determine which Machine Learning Model Performed the best we will take into consideration the Accuracies, TT and F1 scores of the models across the sets. This is because we want to find the model that has the highest accuracy and F1 and lowest TT.

Upon reviewing the performances of all the models across the feature sets, the model that stands out with the highest accuracy and F1 score while having the lowest TT is the Logistic Regression model when utilizing the BF set. It had an accuracy of 73.19%, a F1 score of 0.7321 and a TT of 0.0430 seconds. The high predictive accuracy paired with the low TT makes it an efficient model to utilize for this given classification task when using the BF set.

The Gradient Boosting Classifier was the model that comparatively underperformed when analyzing the DF set. Despite an Accuracy of 72.71% and a F1 of 0.7236, the TT was notably high at 3.1420 seconds.

DF	**Accu.**	**TT**	**F1**
LR	0.7289	0.8780	0.7279

KNN	0.6821	0.3690	0.6799
Average	**0.7055**	**0.6235**	**0.7039**
GBC	0.7271	3.142	0.7236
RFC	0.7216	2.947	0.7156
Average	**0.72435**	**3.0445**	**0.7196**

Table 5: Average Acc, TT & F1 on the DF set.

BF	**Accu.**	**TT**	**F1**
LR	0.7319	0.043	0.7321
KNN	0.6548	0.092	0.6515
Average	**0.69335**	**0.0675**	**0.6918**
GBC	0.7105	0.493	0.7117

RFC	0.6762	0.795	0.6769
Average	**0.69335**	**0.644**	**0.6943**

Table 6: Average Acc, TT & F1 on the BF set.

RF	**Accu.**	**TT**	**F1**
LR	0.7296	0.118	0.7303
KNN	0.6646	0.17	0.6633
Average	**0.6971**	**0.144**	**0.6968**
GBC	0.7093	0.584	0.7074

RFC	0.6791	0.662	0.6749
Average	**0.6942**	**0.623**	**0.69115**

Table 7: Average Acc, TT & F1 on the RF set.

When comparing the simpler models (LR and KNN) to the more complex models (GBC and RFC) we see that the simpler models, across all sets, have an average Accuracy of 69.865%, a TT of 0.278 seconds and a F1 score of 0.6975. The complex models have an average Accuracy of 70.39%, a TT of 1.4372 seconds and a F1 score of 0.702.
The simpler models seem to vary in Accuracy and F1 by an insignificant amount while taking significantly less time than the more complex models.

5. Discussion

The results acquired from the comparative analysis of the K Nearest Neighbors, Logistic Regression, Gradient Boosting Classifier and Random Forest Classifier models provided us with a multifaceted view of model efficiency in relation to their accuracy, computational efficiency and their F1 scores.
The Logistic Regression model, when applied to the BF set, provided us with the highest accuracy and F1 score combined with a low TT. This finding is noteworthy because it suggests that a model's real world utility should not be dismissed due to the fact that it is a simpler model and aligns with the findings of Silva et al. (2018) who employed RNNs for predicting win rates but emphasized on efficiency. For tasks that require speed and accuracy, simpler models would be the preferred models with drastically lower TT times and slightly lower accuracies in comparison to complicated models.
In contrast, the Gradient Boosting Classifier yielded solid accuracy and an F1 score but was significantly more computationally demanding which was reflected by its longer TT. This finding is similar to the conclusions by Arik (2003) about the importance of computational efficiency when it comes to model selection, particularly when the accuracies are similar across most models.

The trade offs which were observed were in line with the broader literature. The studies by Bahrololloomi et al (2023) and Lin (2016) suggest that complex models can potentially offer slightly better predictive accuracy but the additional computational cost will usually not justify their utilization over simpler and more efficient models.

This study also highlights the importance of feature selection in predictive modeling as seen in the study carried out by Ani et al. (2019) and Khromov et al (2019). The models generally took significantly more time when presented with the DF set in comparison to when they were presented with the BF or RF sets. We also highlighted the features that significantly contribute to accurate predictions in League of Legends.

Within the scope of this study, we can conclude that Logistic Regression demonstrates the highest predictive accuracy in forecasting win rates in League of Legends. There were no significant trade offs between model complexity and predictive accuracy however, there were significant trade offs between model complexity and Training Time which was usually higher for the more complex models.

Finally, the most important features that help predict win rates in League of Legends are: blue wins, red wards destroyed, red first blood, red kills, red deaths, red assists, red elite monsters, red dragons, red heralds, red towers destroyed, red average level, blue wards destroyed, blue first blood, blue kills, blue deaths, blue assists, blue elite monsters, blue dragons, blue heralds, blue towers destroyed, blue average level.

5.1 Low Accuracies

This study produced lower accuracies in comparison to other studies so it is important that the results are accurate only within the given context of this study. In the future, using tailored Machine Learning algorithms instead of PyCaret may yield better results. PyCaret is a low code library that enables us to run simultaneous Machine Learning models, this could be the reason for the low accuracies we are presented with across this study.

6. Conclusion

This thesis has contributed to the vast field of predictive modeling in the field of esports analytics, especially in predicting win rates in League of Legends. The comparative analysis of the models employed provided us with valuable insights regarding their precision, accuracy and computational demands and can help future researchers explore simpler or more complicated models with more confidence. The importance of feature selection was also seen as the data sets that underwent feature selection performed significantly better in the Training Time category while having minimal differences in accuracy.

In conclusion, this thesis provides valuable insights into the application of machine learning algorithms in esports.

References

Ani, R., Harikumar, V., Devan, A. K., & Deepa, O. S. (2019). Victory prediction in league of legends using feature selection and ensemble methods. *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 74–77. https://doi.org/10.1109/ICCS45141.2019.9065758

Arık, K. (2023). Machine learning models on moba gaming: League of legends winner prediction. *Acta Infologica*, *7*(1), 139–151. https://doi.org/10.26650/acin.1180583

Bahrololloomi, F., Klonowski, F., Sauer, S., Horst, R., & Dörner, R. (2023). E-sports player performance metrics for predicting the outcome of league of legends matches considering player roles. *SN Computer Science*, *4*(3), 238. https://doi.org/10.1007/s42979-022-01660-6

Hodge, V. J., Devlin, S., Sephton, N., Block, F., Cowling, P. I., & Drachen, A. (2021). Win 2022prediction in multiplayer esports: Live professional match prediction. *IEEE Transactions on Games*, *13*(4), 368–379. https://doi.org/10.1109/TG.2019.2948469

Khromov, N., Korotin, A., Lange, A., Stepanov, A., Burnaev, E., & Somov, A. (2019). Esports athletes and players: A comparative study. *IEEE Pervasive Computing*, *18*(3), 31–39. https://doi.org/10.1109/MPRV.2019.2926247

Lin, L. (2016). League of legends match outcome prediction. Comput. Sci. Dept., Univ. Stanford, Stanford, CA, USA, Rep.

No, S.-J., Moon, Y.-J., & Hwang, Y.-H. (2021). Quantitative analysis for win/loss prediction of “league of legends” utilizing the deep neural network system through big data. *Journal of the Korea Society of Computer and Information*, *26*(4), 213–221. https://doi.org/10.9708/jksci.2021.26.04.213

Shen, Q. (2022). A machine learning approach to predict the result of League of Legends. *2022 International Conference on Machine Learning and Knowledge Engineering (MLKE)*, 38–45. https://doi.org/10.1109/MLKE55170.2022.00013

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