League of Legends AI Coaching Model Project Report

Kadin McWilliams

4/27/2025

kadinmcwilliams@gmail.com

Executive Summary

The analysis revealed that the most effective strategy for winning is strongly tied to two key fundamentals: protecting your own towers and successfully taking down enemy towers and inhibitors. These findings highlight that mastery of core gameplay objectives is more important to success than more complex or situational tactics. Focusing on game play that will influence these two key factors has the statistically highest probability of success for players. In conclusion, players and teams should focus on these basic, high-impact strategies to consistently improve their performance. Strong fundamentals remain the cornerstone of victory.

When the model was tested using a peers League of Legends account, the model was able to successfully predict 30/30 played games. This finding is significant as the test results show that the models learned information can be applied to other ranks and not just the best 500 players in North America.

To build on these insights, I recommend developing a real-time machine learning model that continuously learns as new patches and more games are played. This continuous learning will allow for the model to stay up to date with new strategies and patches, giving the most accurate recommendations. Additionally, creating a web application would help make these tools accessible to a broader audience, driving widespread adoption and performance improvements. Finally creating different SQL servers for each region would allow for the application to be used across the world, instead of just inside of the United States.

Introduction

The goal of this predictive analytics project is to have a machine learning model successfully predict wins and losses. Another main goal is also being able to show what the most important factors for wins and losses are in the model's predictions. The insights from what the model is able to predict about important factors will be used for player feedback. The model's recommendations will be used to identify where players are making mistakes. Identifying these mistakes will allow for players to have a more streamlined process of improvement. Highlighting the flaws in gameplay will hopefully allow players to achieve a higher rank in a shorter amount of time.

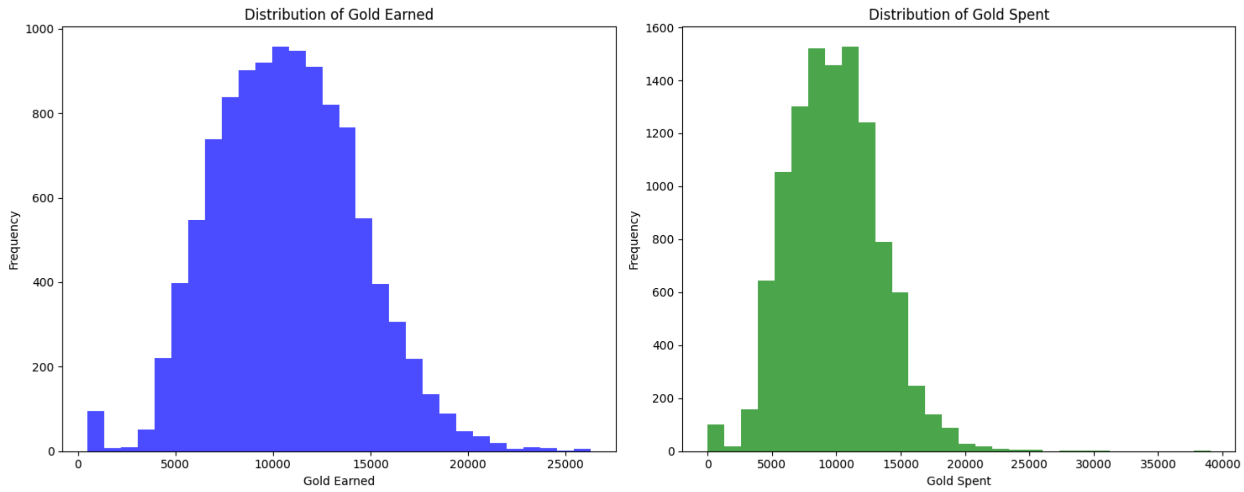
Project Summary

Project Proposal

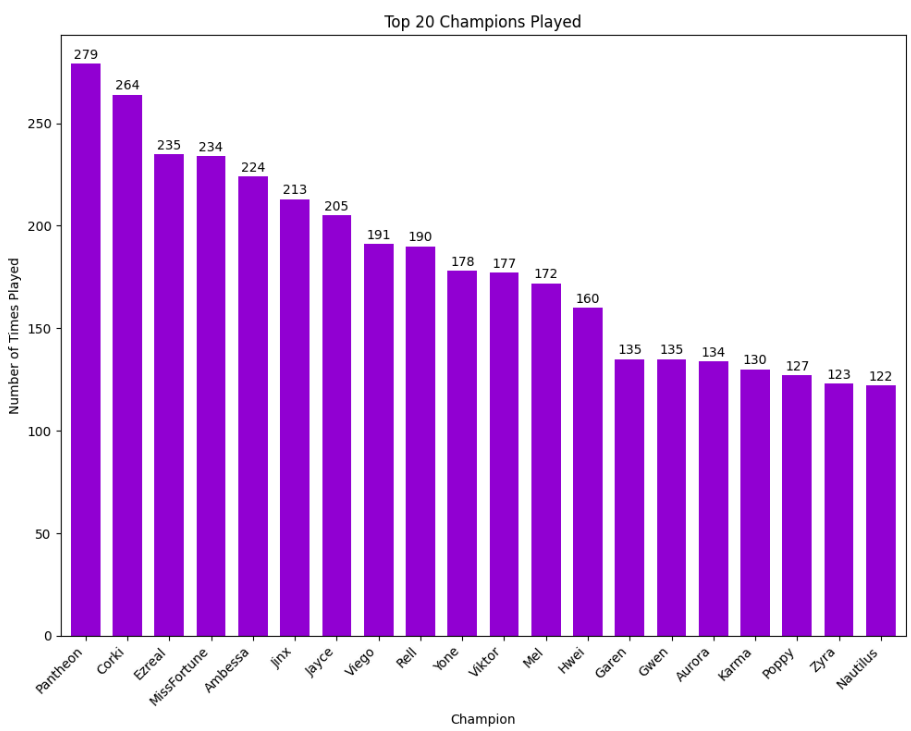
This project seeks to identify the most critical gameplay variables in *League of Legends* through quantitative analysis of 10,000+ challenger games using data collected via Riot Games' API. There are 138 gameplay variables, including objectives, damage, vision, gold income, and player behaviors, are analyzed. An exploratory data analysis is performed, followed by selecting and training a machine learning model, primarily gradient boosting machines, alongside artificial neural networks and random forest classifiers. These models will predict game outcomes and validate findings using collected game data. Python, along with libraries such as pandas, numpy, sklearn, matplotlib, and seaborn, is used for data handling, modeling, and visualization. The project ultimately aims to provide actionable coaching recommendations by highlighting the gameplay factors most strongly associated with success, offering players a structured path to improve their competitive ranking.

Exploratory Data Analysis

The dataset contains 10,957 rows and 138 columns, offering detailed statistics on player performance across multiple games. Key findings include that players average 5.6 kills, 4.87 deaths, and 8.2 assists per game, with a game duration averaging 26.5 minutes. Gold earned averages at 10,741, and turret takedowns are typically around 2 per game. Team objectives like baron and dragon kills are less frequent, with players earning fewer kills on these objectives compared to the team's performance. Significant correlations were found, such as a strong link between gold earned and gold spent, and moderate correlations between kills and turret takedowns, and assists with vision score. Notably, there were no outliers in major performance categories like CS and ward scores, but there were outliers in neutral minions, dragon kills, and baron kills, especially indicating extreme performances by jungle players. Visualizations, such as heat maps and scatter plots, confirmed these relationships, showing how metrics like gold earned and gold spent are closely tied.



Gold spent having a much closer range than gold earned could mean that efficient item building could be a potentially winning strategy. Additionally, role distributions show a relatively even spread, with Support being the least played role and Middle being the most popular. Bottom players had the highest average CS, while Jungle and Support roles had far less. Finally, the top 20 most played champions are dominated by top laners and ADCs, with Pantheon as the most played champion, despite not being in the current meta.



Turrets being linked to deaths and kills could mean that performance is heavily tied to personal deaths and kills in game. The CS column could pose problems to my prediction system if the amount of jungle camps are not being taken into account.

Methods

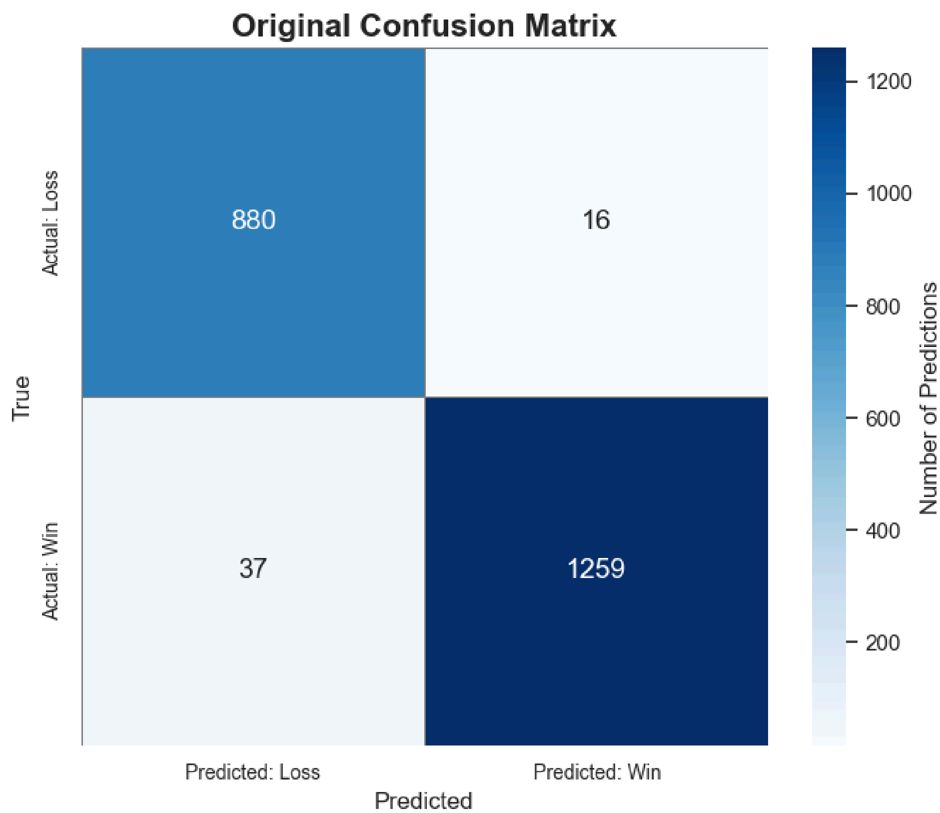
To collect, process, and store the data, a Python script was developed that accessed match data from the Riot Games server and formatted it for human readability. Once processed, the data was stored in a SQL server to facilitate access during model training. A separate script was created to construct and train the machine learning model. Data preprocessing was conducted through a pipeline, during which selected features were dropped to optimize training performance. After training, the model was serialized and saved as a pickle file for future use. This process enabled seamless integration of the gradient boosting model into the main application script. The trained model was then analyzed to identify influential gameplay features and subsequently employed to predict player win or loss outcomes, providing targeted insights for gameplay improvement.

Tools

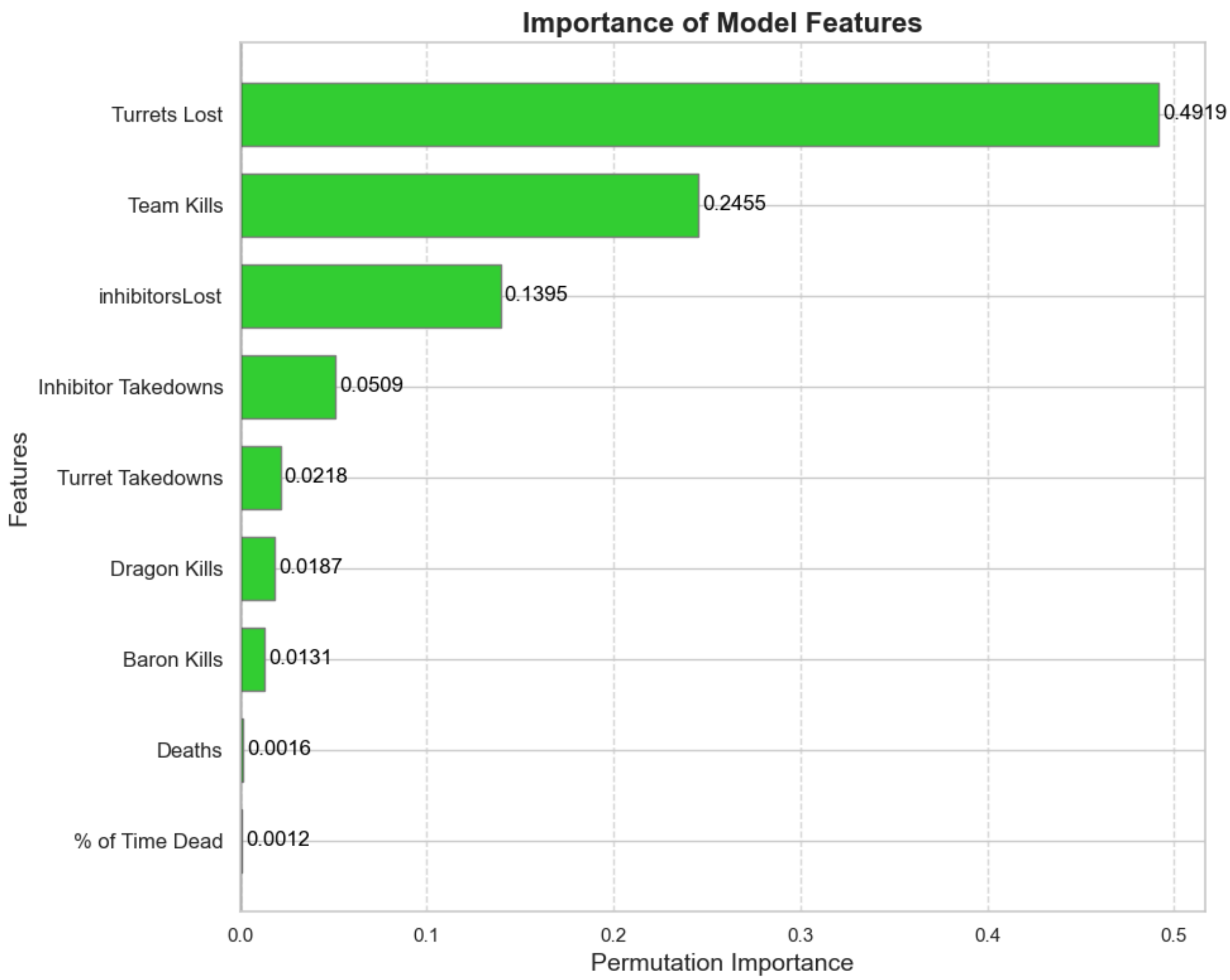
The first tool I used was the Riot Games developer tool to gather all of the data and run tests on my machine learning models. The developer tool helped with understanding what the variables are, navigating the data set and gathering new data easily from the internet. Python was used for all of the coding in this project. Python is easy to use and implement the gathering of data from the API and the machine learning models. The main python libraries used are the following: pandas, pickle, numpy, seaborn, matplotlib, sklearn, time and request. The coding of the project used various online tools such as youtube, chat gpt and various discussion forms.

Results

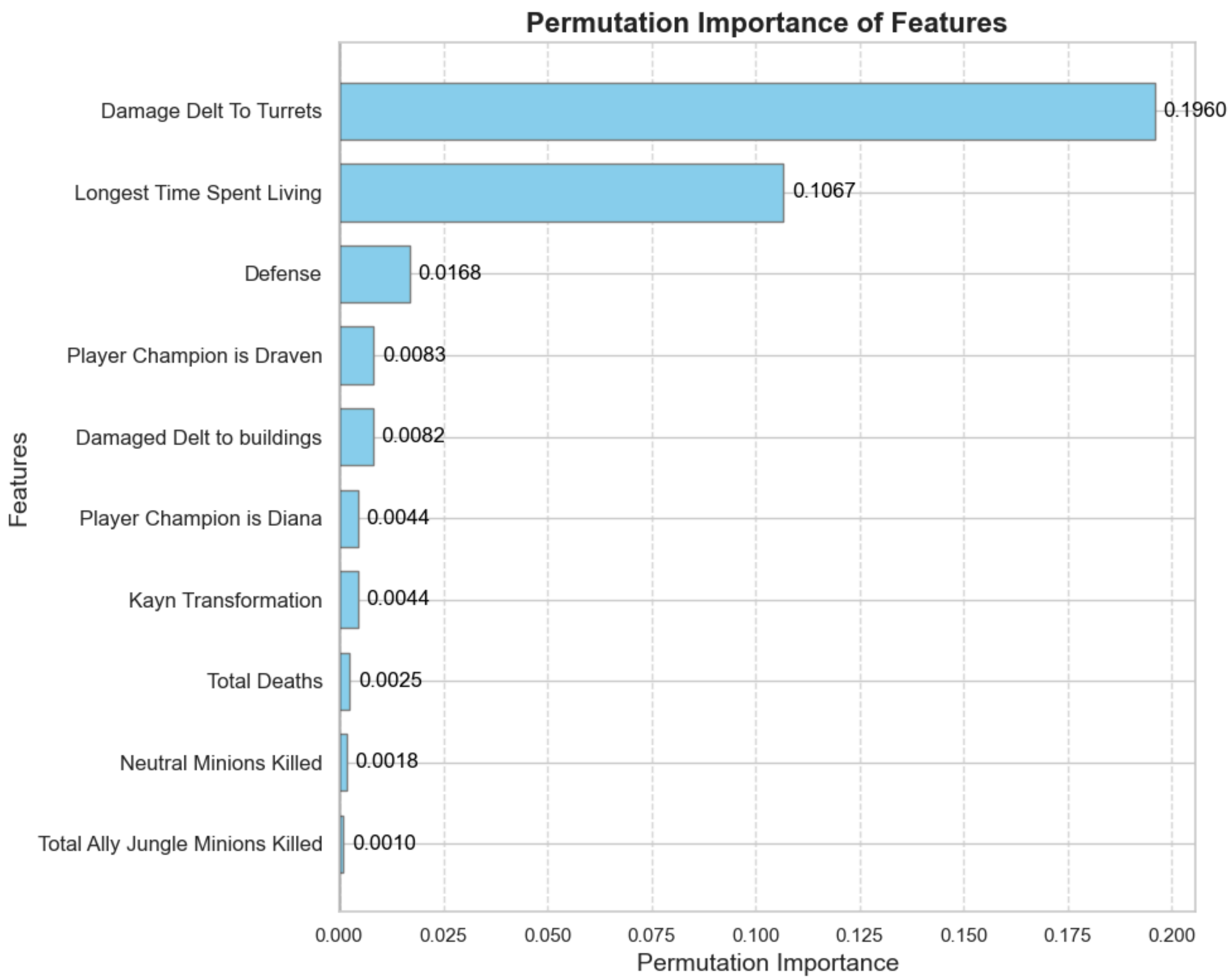
The machine learning model had an overall accuracy score of 98%. The model tended to predict actual wins and losses when it made mistakes, but overall there was no significant effect. The confusion matrix below shows the results from the test group.



The feature importance analysis of the trained model revealed several key findings regarding gameplay variables. Items, runes, and champions had minimal impact on game outcomes, with only a few champions being identified as slightly influential. Summoner spells showed no significant impact. Objectives, such as dragon and baron takes, exhibited a small positive effect, as did the number of deaths. However, the most critical variables identified were related to structural objectives: inhibitor and turret takedowns emerged as highly important factors for predicting wins. Among all features, the number of turrets lost was the single most influential variable, followed closely by total team kills, indicating that both objective control and overall team performance are pivotal in securing victory.



Another method used to show important features was permutation importance. The importance is derived from seeing how much of an effect changing each variable has on the accuracy of the model's predictions. Overall there were similar trends to the important model features. However, there were minor significance for champions and creep scores.



Reflection

Overall, this project was a rewarding and educational experience, offering valuable lessons about large-scale data science workflows, machine learning modeling, and project management. Several aspects of the project went particularly well, while others presented challenges and areas for future improvement. This section discusses what succeeded, what could have been improved, specific recommendations for future work, and the most important lessons learned throughout the process.

One of the major successes of this project was the overall performance and reliability of the machine learning models. The results from the models were strong, with key gameplay features being successfully identified. Additionally, the application of the models to predict the outcomes of future games demonstrated both consistency and scalability. It was encouraging to see the model reliably predict player wins and losses based on gameplay data, validating the initial project idea. Data collection also went very smoothly. Using Riot Games' API to gather data proved to be highly effective. The quality of the data obtained was excellent. The dataset was large, detailed, and comprehensive, allowing for robust analysis and modeling. Finding the best model for the task was another highlight. After testing several machine learning algorithms gradient boosting was the best, selecting gradient boosting as the main model balanced performance and interpretability well. Furthermore, the scalability of the model was a major success; the framework allows for easy updates as more data becomes available, ensuring that the model can continue improving without requiring a complete rebuild.

Despite the overall success, not everything went as planned. One of the major challenges was the final usefulness of the model results. Although the model was good at identifying the likelihood of winning or losing and at flagging general problem areas, it did not provide particularly detailed recommendations on how to correct mistakes. The model could point out that turret losses were critical, but it did not offer specific strategies for reducing turret losses. Additionally, the exploratory data analysis (EDA) phase did not yield as many meaningful insights as expected. While EDA is often a crucial source of understanding, in this case, much of the significant insight came from the machine learning models rather than from early data visualization and correlation studies. A more thorough or differently structured EDA process might have yielded better early guidance for the modeling phase.

Although many aspects went well, there are several things I would approach differently if I were to redo this project. First, I would work on improving the data collection system further. While the initial data gathering was effective, automating the process to update continuously and seamlessly would have improved efficiency and reduced manual workload. Setting up a more sophisticated data ingestion system would have allowed for even faster iteration cycles when refining the model. Another area for improvement would be enhancing the significance of the model findings. Although the model identified errors and important variables, the feedback it provided for correcting player mistakes was somewhat limited. In future work, I would explore more sophisticated feature importance techniques or ensemble explanations to generate more actionable feedback for players. I would also introduce a small user interface earlier in the project. Having even a basic frontend would have made testing and interpreting results easier, allowing for a better user experience in the future. Finally, while the models performed well, I would invest more time into building multiple prototypes at earlier stages. Building and testing smaller, simplified versions of the project could have uncovered some of the challenges earlier, streamlining the full development process and saving time.

For anyone embarking on a similar project, I would offer several recommendations based on this experience. First, it is crucial to develop a detailed project plan before beginning. Laying out all the stages: data collection, EDA, modeling, evaluation, deployment, ensures that no major steps are overlooked. I would also recommend building several small prototypes before attempting a full project buildout. Iterating on smaller models and processes can help catch potential problems early without requiring massive reworks. Another key recommendation is to spend more time understanding the data. The quality of the data source is critical; selecting a data set that truly captures the variables relevant to your problem is the foundation of a successful analysis. Moreover, investing time in data exploration, feature engineering, and understanding variable relationships before modeling can dramatically improve the final results. Lastly, taking the project one step at a time and not rushing through any phase is important. Data science projects benefit immensely from thoughtful planning, careful evaluation of results, and patient refinement of techniques and models.

This project offered many valuable learning experiences. Technically, I gained a strong understanding of interacting with external APIs, particularly learning how to navigate and extract large, complex datasets through Riot Games' developer tools. I also became proficient in using SQL for data storage and retrieval, allowing me to manage a large database of match data effectively. Additionally, I learned how to construct a larger-scale project using multiple interconnected Python scripts. Structuring the project into separate scripts for data collection, data processing, modeling, and prediction was critical to maintaining organization and flexibility. From a machine learning perspective, I expanded my skills by training models at a larger scale and learned how to make them easy to update as new data becomes available. Handling real-world data, addressing challenges such as missing values and noisy inputs, and ensuring that the models generalized well beyond the training set were major technical skills gained. Finally, I learned the importance of thoughtful project design and management. Developing a full pipeline from raw data to meaningful results taught me about the need for planning, organization, and continuous evaluation throughout a project. These skills will be invaluable for my future work in data science, machine learning, or any large-scale technical endeavor.

Conclusion and Future Work

In summary, this project's results successfully demonstrated that strong fundamentals, particularly protecting your own towers and destroying enemy structures, are the most important factors in winning League of Legends matches. The machine learning model achieved a high prediction accuracy of 98%, and when tested on a peer’s account, it successfully predicted the outcomes of all 30 games. These results highlight the model’s ability to generalize beyond just top-ranked players and provide valuable insights to a wider player base. The project also showcased the effectiveness of gradient boosting models for this type of structured gaming data and proved that even simple, core objectives in gameplay have the strongest statistical relationship with success. However, there is still significant room for future development. Moving forward, one key direction would be building a real-time machine learning model that continuously updates as more games are played and as game patches adjust the meta. This would ensure that the recommendations remain accurate and relevant over time. Additionally, creating a web application would allow users to easily access the model’s predictions and recommendations without needing technical knowledge, making the tool more accessible to the broader community. Setting up separate SQL servers for different geographic regions would also be an important expansion, allowing players worldwide to benefit from tailored insights based on their own regional gameplay styles and competition. Overall, this project laid a strong foundation by identifying what matters most in gameplay and demonstrating a working model, but scaling it into a dynamic, user-friendly tool represents an exciting opportunity for future work.