Bachelor Datascience and Artificial Intelligence

The Possibilities of Applying Machine Learning to a Limited Esports Dataset

Jesse Nieland

Supervisor:
Mike Preuss

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

15/01/2025

Abstract

This thesis explores if its possible to build predictive models using a limited esports dataset. Professional Valorant match data was collected using a web scraper, which provided information on the issues that come with small and incomplete data. Initial attempts to identify a problem to analyze for this thesis showed problems, which shifted the focus to the current research question: evaluating the potential of predictive models, using models like decision trees, random forests and Support Vector Machines. The following exploratory process revealed the importance of generalizability and choosing the right features to achieve meaningful predictions. This thesis provides guidance on how to tackle data challenges in esports analytics and explores how machine learning can still be useful with such limitations.

Contents

1 Introduction

Valorant is a tactical first-person shooter released by Riot Games in 2020. Two teams of five compete to achieve specific objectives, such as planting or defusing a bomb, comparable to Counter-Strike. Although only being released for a few years, Valorant has gained significant popularity, with simultaneous viewership peaking at about 1.7 million viewers all-time. This places the game just below highly successful games such as Fortnite, Dota 2 and Counter-Strike [Espa]. In 2024, Valorant reached its highest viewership, ranking fourth among all games [Espb]. This ranking excludes Chinese broadcasts, which significantly gained popularity since Valorant's Chinese release in Summer 2023 [VAL23]. VCT Champions 2024 reached 9.1 million simultaneous viewers, over five times that of VCT Champions 2023, showing the game's fast growth [Far24].

As esports continues to grow, data analysis in this field becomes increasingly important, as it helps to identify patterns, analyze strategies and improve game development. Some games, such as League of Legends and Counter-Strike, have been the focus of multiple studies in this field. However, Valorant lacks these studies, likely because of its relatively recent release and its policy on player data. Unlike some other games, Valorant gives players ownership over their own data, requiring players to opt-in to a third-party application if it wants to access their data [Hen21].

These limitations create challenges throughout the data analysis process, but also present a unique opportunity. This thesis explores the possibilities of machine learning on a limited esports dataset, investigating what is possible and identifying areas of improvement. The findings inform about the challenges of data analysis in Valorant and guide future work in esports analytics.

1.1 Thesis Overview

Section 1 is the introduction; Section 2 explains the mechanics of Valorant; Section 3 describes the problem identification process; Section 4 defines the research question; Section 5 explores related work; Section 6 describes the research approach; Section 7 explains the modeling process and explores the results; Section 8 concludes and discusses possible future research; The Appendix contains figures and tables used throughout the thesis.

2 Game Overview

This section gives a brief overview of the game and its mechanics. This should give the reader a better understanding of the game as we start to look at its data during this thesis.

2.1 Game Basics

Valorant is a tactical first-person shooter in which two teams of five players compete to win 13 rounds. One team starts as attackers and the other as defenders, with the sides swapping after 12 rounds. If the game is tied at 12-12, overtime begins, where both teams play one round as attackers and defenders. Overtime continues until one team wins both rounds.

Both sides have different objectives: the attackers need to plant a bomb, known as the spike, on one of the defenders' sites, while the defenders need to stop the spike from exploding. This can be done by defusing the spike after it has been planted, stopping the attackers from planting it before the rounds ends, or eliminating all attackers. The attackers win when the spike explodes or by eliminating all defenders.

2.2 Unique Mechanics

Players can choose different guns during the game, allowing them to fight their opponents in their preferred way. Valorant also includes a character system, where each agent has their own unique abilities that let them specialize in different roles. Every agent has an ultimate ability, which can be charged by getting kills, dying, planting the spike, or collecting ultimate orbs that are placed around the map. Combining the abilities of different agents allows players to create interesting strategies. Guns and abilities are bought using money earned during rounds. Money is earned in multiple ways, such as kills, planting the spike and winning or losing rounds. The economy system helps teams recover if they lose multiple rounds in a row by giving them extra money. This keeps the match competitive, as teams can not snowball of an economic lead. Lastly, the game includes multiple maps with varying design (Figure 1). This introduces different strengths and weaknesses per map, giving certain sides, agents, or weapons an advantage.

Figure 1: Minimap of Lotus [Wik]. The yellow part indicates the sites the spike has to be planted on. Lotus is relatively big, features 3 sites and has wide areas teams can fight over for control.

2.3 Professional Circuit

Valorant's professional circuit, the Valorant Champions Tour (VCT), consists of four different regions: EMEA, Americas, Pacific and China. Teams from these regions compete in regional events to secure spots for international events known as Masters. Both regional and Masters events award points to the top teams, bringing them closer to qualifying for Champions, the Valorant world championship.

3 Problem Identification

The first phase of the thesis focused on identifying a research question. Discussions with the supervisor led to researching the possibilities of developing a time series model. The goal is to find out how long it takes to accurately predict if an agent was well balanced by looking at playerbase data after its release. This meant looking into possible data sources.

3.1 Exploratory Model

The Official Valorant API does not allow personal projects as use cases [Rio21a], and unofficial API's have severe limitations, such as high rate limits and needing players to sign in to use their specific player data [Hen21]. This severely limits the options for analyzing playerbase data.

3.1.1 Web Scraper

As Riot put specific limitations on the data from the game itself, the research focus shifted to professional match data from other sources. These matches are documented at multiple sites, such as `https://www.vlr.gg` and `https://www.rib.gg/`. Although limited in size and specifications, these sites provide flexible and consistent data. As these are different data than intended, the goal changed to designing a random forest model that predicts the outcomes of matches based on historical data. VLR is the only platform with an API, although it is an unofficial API [Sad]. After reading the documentation, I concluded that the output provided would not be sufficient for the requirements, resulting in the need for a web scraper. Permission from the website owners was required for the scraping, and as VLR was the only platform to reply and give permission, the decision was made to scrape data from `https://www.vlr.gg`. A rate limit was put in place of 10 requests per second, on the owners' request.

The web scraper is written using Python and the requests and BeautifulSoup libraries. It only covers the 2024 season, since huge changes in team composition, such as players and coaches switching teams, are made between seasons. Additionally, Riot created a franchised league in 2023, which meant that there are only 2 seasons with a set group of teams competing in VCT. The previous seasons had a significantly different format and the participating teams were not set.

The scraper focuses on general match details, outputting the following columns:

- **Event**: The name of the event for this match.
- **Team 1** and **Team 2**: The competing teams.
- **Date**: The date and time the match took place.
- **Match Format**: The format of the match, such as best-of-3 (Bo3) or best-of-5 (Bo5).
- **Maps Picked/Banned**: The map veto for the match.
- **Score per Map**: The score by each team for every map played.
- **Match Durations**: The total duration of each map.
- **Score per Half**: The score by each team for each half of every map.
- **Match URL**: A link to the match page.

The data is saved as a CSV file using the Pandas library so it can be cleaned and analyzed in the following stages.

3.1.2 Data Cleaning

The data collected through scraping needs to be cleaned before it can be used for the model. The cleaning tasks included:

- **Type Conversion:** Converting all string-based data into their proper types, such as integers for scores and dates for timestamps.
- **Splitting Data:** Separating map-specific statistics (e.g., scores per map, durations, map name) from match-specific statistics (e.g., bans/picks, teams, map format) to create two structured datasets.
- **Handling Missing Data:** Handling missing values by marking them as incomplete.
- **Splitting Combined Values:** Extracting useful information from more complex entries, such as breaking down scores or map picks/bans into individual entries.
- **Standardizing Data:** Ensuring consistency in data representation, such as consistent naming conventions for teams (e.g., Giants Gaming and GIANTX are the same team).
- **Manual Error Correction:** Manually fixing errors caused by issues in the HTML, such as maps having the wrong order in the HTML.

Using Python and Pandas, the scraped data is transformed into a structured and usable format for modeling.

3.1.3 Preliminary Results

With the data cleaned, we wanted to determine the possibilities of a model predicting match outcomes for this dataset. Implementing a basic random forest model on the current data with some additional feature engineering gives preliminary results. With these results, we can analyze the potential of the model. This uses Pandas and Sklearn. The engineered features used in the model were as follows:

- **Pick Team 1:** Is the map a pick from Team 1 (True or False).
- **Pick Team 2:** Is the map a pick from Team 2 (True or False).
- **Remaining Map:** Is the map not picked by either team (True or False).
- **Recent Form Difference:** The difference in recent performance between the two teams, calculated as the average score difference over the last 3 matches.
- **Win Rate Difference:** The difference in match win rates between Team 1 and Team 2.
- **Team 1 Attack Win Rate:** The win rate of Team 1 attack rounds.
- **Team 1 Defense Win Rate:** The win rate of Team 1 defense rounds.
- **Team 2 Attack Win Rate:** The win rate of Team 2 attack rounds.
- **Team 2 Defense Win Rate:** The win rate of Team 2 defense rounds.
- **Total Rounds Played Difference:** The difference in total rounds played by the two teams, showing experience throughout the season.

The basic random forest regressor model that was built gave the results found in Table 1 and Figure 2. The results show that the model performs only slightly better than guessing randomly. The low value of R-squared indicates that the model does not explain any of its variability. Discussions with the supervisor concluded that this match outcome prediction model based on historic data will not work as the data size is too small. Focusing on team statistics spreads the data across 44 different teams, severely limiting the data points per team. Furthermore, significant changes in team rosters between seasons make it difficult to gather more data. This meant that other research questions needed to be explored.

Metric	**Value**
R-squared (R^2) [Mil05]	-0.008
Accuracy of Predicted Winners	55.95%

Table 1: Preliminary Results of the Random Forest Model

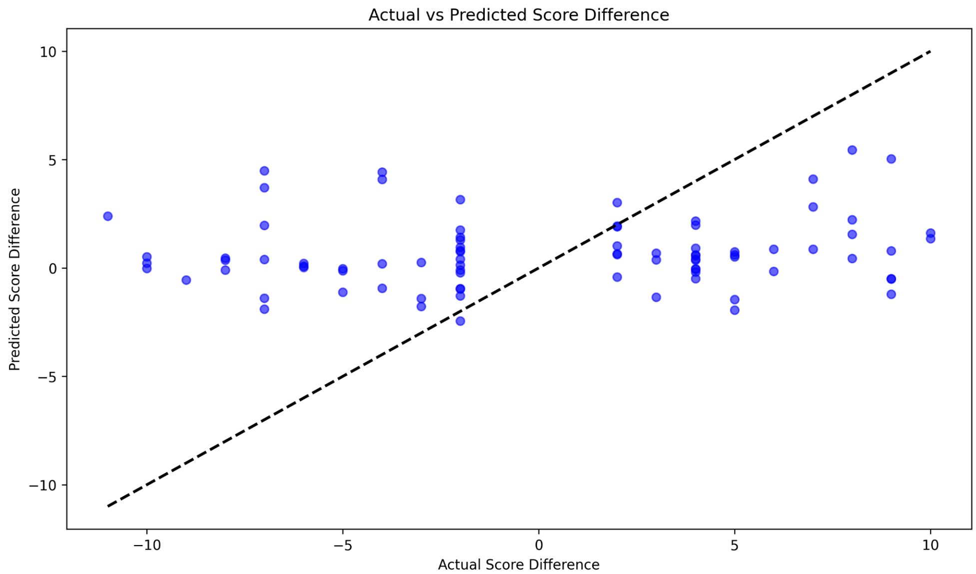

Figure 2: A scatterplot showing the distribution of actual versus predicted score difference

3.2 Exploratory Analysis

As we had to identify a new focus for the research, I performed differential analysis to find interesting patterns to model for the research question.

3.2.1 Data Processing

This analysis focused more on maps. We reasoned that this would be easier to model if we could not focus on the teams, and it should give interesting data, such as side preference and map preference. For this analysis, the web scraper is modified to collect additional match data from VCT 2023. The difference in team strengths between years would not significantly matter for analysis of the maps, and this would give twice as much data. Additionally, a column is added to track the sides teams first played on per map.

Using this column, the picked starting side can be determined when combined with the map veto rules. This is based on the official rules for VCT, which can be found in the official rulebooks for VCT 2023 [Rio23] and VCT 2024 [Rio24]. The detailed map veto rules used for the analysis are provided in Appendix A. By following these rules and combining them with the data from the Map Veto column, the new columns Side Picker and Side Picked were created, allowing analysis of side-specific statistics.

3.2.2 Differential Analysis

As we wanted to analyze differences in the data, with the focus being more on map-specific statistics, plots were created to visualize these statistical differences. To get a better picture of the map-specific data we were working with, plots were created to show the distributions of the maps played. These figures can be seen in Figures 3 and 4.

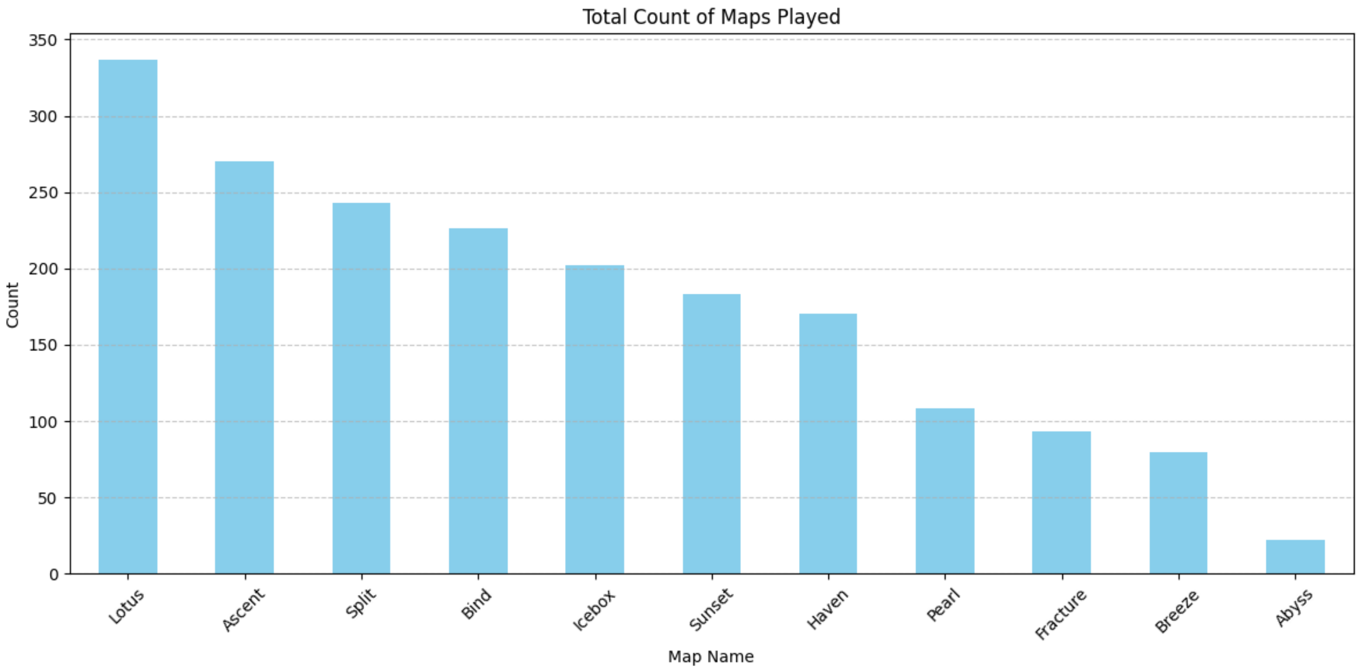

Figure 3: Total count of maps played, highlighting the popularity of each map within the dataset.

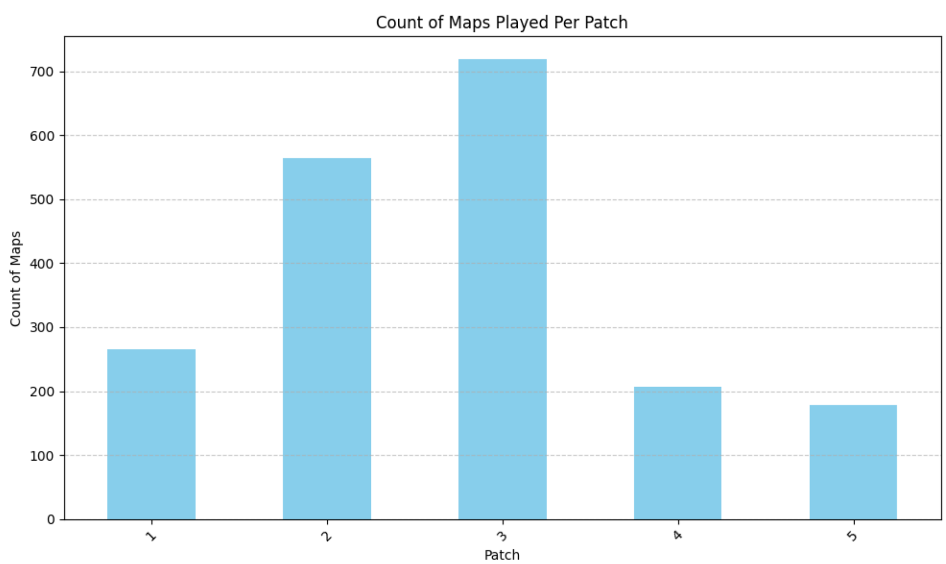

Figure 4: Distribution of maps played across different patches, showing the number of maps played in each patch over time.

Patches are defined as the periods with different map pools played in these matches. They are based on general updates to the game, but do not account for every update, only changes to the map pool. This also means that changes to the map pool made in between seasons are not visible in the data, as there are no matches being played in the VCT circuit.

In addition, plots were created to visualize the preference for sides and maps, the strength of each side, and the competitiveness of the games. Using these plots, we can observe patterns, helping us to define a research question. These figures can be found in Appendix B.

3.2.3 Observations & Conclusions

Observing the count of maps played per patch, as seen in Figure 4, unfortunately shows a significant imbalance between patches. Patch 2 and 3 clearly have a lot more matches compared to 1, 4 and 5. This can be partly explained by the addition of a new map, Abyss, which happened at the start of the fifth patch. To add Abyss, Split had to be rotated out, which is the only map change between patch 4 and 5. Normally, map pool changes have two maps rotating out, so this is only a small update. However, these differing match counts introduce a problem. Analysis per patch gets hampered by a lack of data, as a count of circa 200 maps results in an average map count of less than 30, as there are 7 maps in each map pool. The already small data set with which we are working would only be broken down into smaller parts.

Furthermore, if we look at Figure 3, there is a significant difference between the total number of maps being played. Maps such as Pearl, Fracture, and Breeze have been removed from the map pool because of issues with map design and popularity, while Abyss and Sunset are maps that have only been added recently. This gives an unbalanced distribution, with some maps giving more information than others.

By looking at all the figures, including those in Appendix B, some patterns can be observed. Figure 7 shows that teams prefer to play Lotus, while avoiding playing Fracture and Breeze. Figure 8 shows that teams have a slight preference for attack. This can be explained as a result of the attackers having more control over their actions. Often, they are the side that needs to take action, while the defenders are more reactive. In addition, attackers often have the advantage when it comes to taking ultimate orbs, which can give them a slight edge. Furthermore, the figure shows that the teams specifically prefer to play defense on Ascent and Split. While Ascent has always been defense-sided, Split is, according to the data, a neutral map. This can be seen in Figures 11 and 13. This could be a consequence of the old version of Split being defense-sided. This version is not played in any matches in the dataset as it was changed before VCT 2023. This could introduce a bias, as even though the data suggests that Split is a neutral map, teams might still think it favors defenders.

Figures 11 and 13 show the actual side balance per map. According to the figures, Lotus is very attack-sided, while Ascent is defense-sided. In general, maps slightly favor attack, which could be explained by the previously mentioned increase of control. Figure 12 visualizes a distribution of the competitiveness of each map. Closely Contested and Blowouts are, respectively, the top and bottom 25% of all games. While this shows some patterns, such as Bind having fewer blowouts than all the other maps, the figure does not show anything to base a research question on.

Figures 9 and 10 visualize the side strength per map from the perspective of the side picker. We can see that picking attack on the new map Abyss and defense on Fracture result in a worse score difference compared to picking the opposite. As Abyss has only been added recently, attackers might struggle with the map if they lack experience on it, as attackers often execute plays more than defenders, who play more reactive. Fracture has a significantly different design compared to other maps. The defenders start in the middle of the map and have to defend their sites from opposite sides. This might make it harder to defend the map without proper preparation. Additionally, these figures suggest that newly added and rarely played maps favor the map picker more than regular maps. Abyss, Fracture, Pearl and Sunset show a slightly worse score difference compared to maps that have been in the map pool for a lot longer. Breeze is an outlier here, showing an average score difference.

Although some patterns are observed, we concluded that it is hard to define a specific focus for this thesis using this dataset. As mentioned, the data we are working with are small in size and limited in features. The analysis shows that there is an unbalanced map distribution, and the map pool changes show issues with working in a temporal setting. This brings us to the conclusion that instead of focusing on a specific issue for this research, it is better to explore the possibilities of modeling the current limited data.

4 Research Question

We want to experiment with the potential of machine learning models on the limited dataset we could gather from Valorant esports data. The research question can be formulated as follows:

"Can we build any meaningful predictive models with a limited Valorant esports dataset?"

4.1 Specifications

For this thesis, we consider a meaningful predictive model one that shows a clear improvement in accuracy over a baseline. This improvement is rather subjective, but should show that the model outperforms simple rules, such as an heuristic or random guessing. Additionally, it is important that the model generalizes well, which can be evaluated using the standard deviation. Achieving these results would show that meaningful predictive models are possible. However, if we cannot achieve these results, it would not mean that meaningful predictive models are impossible. It can inform about the problems encountered and the potential future work that needs to be done. Both results would satisfy us.

4.2 Motivation

The data collection issues that we encountered, such as limited and incomplete data, give a unique opportunity. Investigating what we can do with this limited data allows us to find what models are possible. In addition, we can find the potential improvements needed in this field.

5 Related Work

Studies that try to apply machine learning to esports data are a more recent development, but start to become more frequent as the field grows. In this section, we briefly look at some related studies and review where this study can be placed.

Studies have shown the ability to predict match outcomes in games like League of Legends, Dota 2 and Counter-Strike. Hodge et al. (2021) studied live match prediction in Dota 2, focusing on real-time data during professional games [HDS+21]. Using data gathered from replays using an API and community-developed tools, they accurately predicted 85% of match outcomes after 5 minutes. League of Legends seems to be the most popular game for machine learning studies, likely because of the accessibility of its API [Rio21b]. Ani et al. (2019) use feature selection and ensemble methods to predict match outcomes in League of Legends [AHDD19]. For their study, random forests seemed to be particularly effective. Do et al. (2021) investigated the impact of player's experience on a champion using a deep neural network, which showed that it had significant predictive power [DWY+21]. Hitar-García et al. (2023) use feature engineering and selection to improve the data needed to predict match outcomes, achieving similar accuracies as other approaches but using fewer data [HGMFBC23]. Silva et al. (2018) focused on continuous outcome prediction using recurrent neural networks (RNNs) [SPC18]. By looking at specific time intervals in the game, it is possible to analyze and identify the actions that teams should take. Schmidt (2020) examined match predictions for Counter-Strike: Global Offensive to develop a decision support system [Sch20]. This master thesis encounters similar data issues, having to settle for suboptimal data for improved accessibility.

In contrast to these well-established games that have been released for over a decade, Valorant was released less than 5 years ago. Additionally, Valorant data are not as readily available compared to other games, as the developers want players to have ownership of their data [Cho20]. This means that research specifically focused on Valorant is still lacking. The few existing projects for Valorant have mainly been developed by enthusiasts [Lam][Fon]. This thesis fills this gap, while specifically focusing on the possibilities of machine learning on a limited dataset.

6 Research Approach

Given the small data set and its limitations, the focus of this thesis will be on its potential for predictive modeling. The goal is not to create high performing models but to determine if it is possible to create models that give insights into the data. The design of heuristic rules could help in the model building, potentially increasing prediction accuracy and helping as a benchmark. To evaluate the results, performance metrics and diagrams are generated.

6.1 Data

As input for the models, the cleaned and processed data from the 2023 and 2024 VCT seasons is used. Games without map veto data were excluded as they lack data on the maps and sides chosen by the teams. This results in the removal of 15 matches from the Champions China Qualifier event for VCT 2023, which we consider an acceptable and reasonable loss.

6.2 Heuristic

As we need a target to predict, I looked at heuristics that could be used as a benchmark. This gives a baseline that the models should aim to surpass. An interesting heuristic to look at is predicting who will win based on the halftime score. The heuristic works as follows:

1. **Teams and Roles**:
   - **Team X**: The opposite of Team Y, usually the map picker unless the map is neutral.
   - **Team Y**: The side picker.
2. **Heuristic Rules**:
   - **If one team leads at halftime**:
     - Predict the leading team as the winner.
   - **If the halftime score is tied**:
     - **If the map is picked by one of the teams**:
       * Predict **Team X** (the map picker) as the winner.
     - **If the map is neutral**:
       * Predict **Team Y** (the side picker) as the winner.

For games with a tied halftime score the heuristic predicts team X if they picked the map, as we assume that the map picker has the advantage on their chosen map. However, on neutral maps, the heuristic predicts team Y, since we assume that the side picker has the edge because they can pick their preferred side. Although this does make sense, as leading at halftime means you are a few rounds closer to winning than the opponent, it does not take into consideration potential map strengths and side biases. In addition, games with a tied halftime score are hard to predict using a heuristic. Using models with the right features could help improve on this heuristic.

6.3 Models

The following three models will be used to improve on the heuristic:

- **Decision Tree**: A simple and interpretable model that splits data based on feature thresholds.
- **Random Forest**: An ensemble of decision trees that combines their outputs [Bre01].
- **Support Vector Machine (SVM)**: A model that separates data by finding the best boundary [CV95].

Starting with a decision tree, a simple model that is also easy to interpret is used to improve the heuristic. Following this up with random forest gives a more robust method, that is based on the decision tree model. Lastly, by using SVM with different kernels, a different method is tried that is more complex and may give different insights.

The same train and test splits are used for each model to ensure consistency. 80% of the datas will be the train set, the other 20% the test set. The random forest model uses 100 trees for each prediction. The default values are used for the SVM models.

6.4 Metrics

The mean test accuracy is calculated to evaluate the models and the heuristic. This metric is calculated using 10 different training and test sets, based on random seeds, to reduce the risk of overfitting and to evaluate the model stability using the standard deviation. For tree-based models, a range of depths is tested to identify the depth that achieves the best accuracy. Alongside accuracy and standard deviation, plots visualize the effect of the maximum depth on the performance of tree-based models. Additionally, the feature importance is calculated using the random forest model. This helps identify the most influential features, which gives important information that can be used to test different scenarios. Lastly, visualizations of decision trees are generated to look at specific splits, giving insight in the choices made by the model.

7 Modeling Process & Results

In this section, we discuss the iterative modeling process and its results. Because this research is exploratory, we use the results to make decisions on how to continue. The code for this has been written in Python and uses the Pandas and Sklearn libaries.

7.1 Heuristic Process & Results

To start, I wrote a script to analyze the performance of the heuristic. Looking into the different scenarios of the heuristic allows for a more complete analysis of the heuristic performance. The results can be seen in Table 2.

Table 2: Accuracy of Heuristic and Predictors in Various Scenarios

Scenario	**Accuracy (%)**	**Number of Games**
Halftime Score Heuristic (including ties)	76.40	1898
Halftime Score Heuristic (excluding ties)	81.85	1592
Map Picker Accuracy (tied halftime)	48.64	257
Side Picker Accuracy (neutral map, tied halftime)	44.90	49

The results show that the halftime heuristic is a significant predictor, especially in games that exclude halftime ties. This shows that the halftime lead feature has strong predictive power. However, the accuracies for the heuristic drop significantly for games with tied halftime scores. Both of the heuristic predictions for tied halftime games perform worse than random, indicating room for improvement. The disappointing performance for tied halftime games could be explained by the lack of a strong predictor. For a tied halftime score, a lot of dynamics might have an impact. For example, if Team Y picks defense on Ascent, a defense-sided map, and the halftime score is tied, we might expect Team X to win as they play the second half on defense. However, if Team Y picks defense on Lotus, an attack-sided map, a tied halftime score suggests an advantage for Team Y. Unfortunately, the limited number of tied halftime games makes modeling these data more challenging.

7.2 Model Setup

For the implementation of the three models, the following features are created:

- **Target**: The team that wins (0 if Team X wins, 1 if Team Y wins).
- **Halftime Lead**: The difference in scores at halftime (used in every model, except SVM).
- **Halftime Lead Scaled**: The scaled difference in scores at halftime (used only in SVM).
- **Neutral Map**: Indicates if the map did not get picked by either team (True or False).
- **Picked Attack**: The side picked by the side picker (1 if they picked attack, 0 if they picked defense).
- **New/Rare**: Indicates if the map is new or rarely played (True or False).
- **Side Strength**: The side bias of the map (1 if attack-sided, -1 if defense-sided, 0 if neutral).
- **Pick Map {map_name}**: The map that is being played (One-hot encoded, True or False).

While every other feature takes only one column, the Pick Map {map_name} column uses 11 columns, one for each map. This feature gives a more detailed distinction, which could help capture map-specific patterns. The New/Rare and Side Strength features categorize the maps based on my expert knowledge combined with the results seen in Appendix B. These features aim to generalize the data more effectively. New/Rare is a binary indicator of whether a map is new or rarely played. This could potentially explain performances on maps, as teams might lack experience on the new or rarely played maps. The Side Strength feature categorizes maps by their side bias. Combined with the Picked Attack feature, this could help the model predict using side balance. However, these features introduce redundancy: while the features New/Rare and Side Strength categorize maps, Pick Map identifies the map. Because of this, it is valuable to look at the results with and without these features. Analyzing the difference can provide insight into the importance of using generalized features versus specific features when building a model on a limited dataset.

7.3 Initial Model Results

To start, two scenarios are evaluated. First, we train the models on the full dataset, excluding all columns except New/Rare and Side Strength. Second, we train on the same dataset, but exclude the one-hot encoded map feature, the Pick Map columns. In this way, we can observe the results without having to worry about redundancy. The results can be seen in Table 3, Figure 5 and Appendix C.

Table 3: Model Mean Test Accuracy Results for All Games

Model	**Without New/Rare and Side Strength**	**Without Pick Map**
Heuristic	75.68% ± 1.74%	75.68% ± 1.74%
Decision Tree	76.45% ± 1.80% (depth: 1)	76.45% ± 1.80% (depth: 1)
Random Forest	76.32% ± 1.52% (depth: 5)	76.05% ± 1.98% (depth: 1)
SVM: linear	76.26% ± 1.48%	75.76% ± 1.86%
SVM: rbf	76.24% ± 1.63%	76.53% ± 1.25%
SVM: poly	75.71% ± 1.54%	74.11% ± 1.60%
SVM: sigmoid	70.03% ± 1.52%	68.87% ± 2.22%

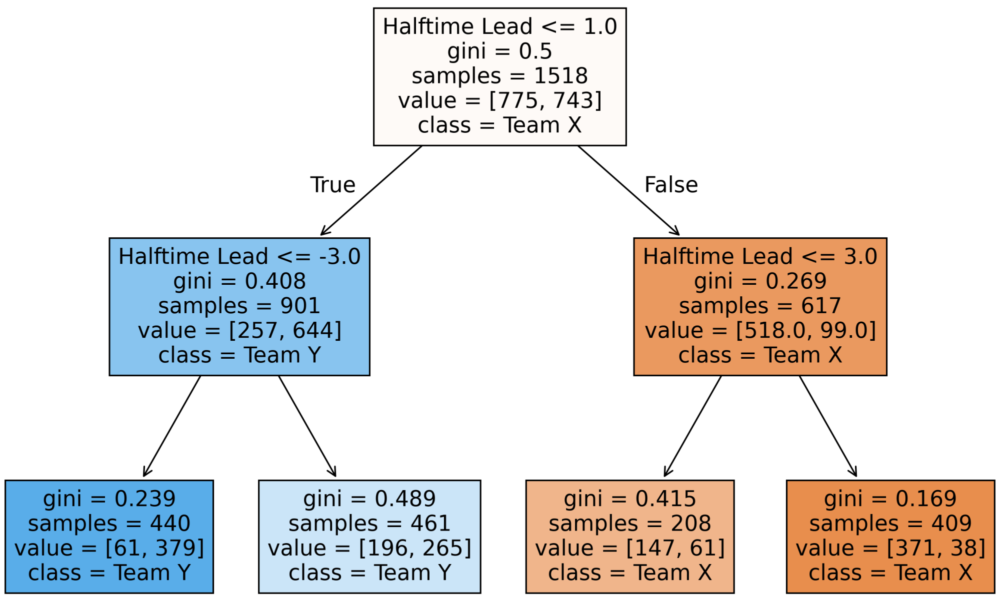

Figure 5: Decision tree visualization with a maximum depth of 2.

As seen in Table 3, there is a slight improvement in the averaged accuracies across all the models compared to the heuristics. However, the tables and figures also show that the models tend to overfit as they become too complicated for this prediction. The decision tree achieves its optimal test accuracies in both scenarios after the first split. The random forest performs slightly worse than the decision tree, probably because of the small dataset size. It also shows optimal test accuracy at very early splits. The accuracy vs. depth plots confirm this as well, showing that the tree-based models consistently do not make significant improvements before overfitting across the test splits. The SVM model performs worse for the sigmoid kernel when compared to the linear, rbf and polynomial kernels.

The decision tree and the feature importance tables shows that the halftime lead feature is by far the strongest predictor. The decision tree splits the root node based on the Halftime Lead at the value 1, which implies that the model predicts Team Y when the halftime score is tied.

Furthermore, the feature importance shows that the Halftime Lead significantly outperforms the other features. However, the two scenarios show no significant differences. Although there are slight changes in accuracy across the board, there are no significant improvements, which can be explained by the strength of Halftime Lead. As the Halftime Lead is directly tied to the match outcome, it is expected that it significantly outperforms the other features.

7.4 Model Performance on Tied Halftime Games

Since the Halftime Lead is the most important feature for the models, we focus on scenarios where the halftime score is tied. In these cases, the Halftime Lead feature should have no predictive power, which gives the opportunity to evaluate other features. However, the smaller dataset for tied games significantly reduces the reliability and generalizability of the results compared to previous scenarios.

The models are evaluated on the subset of tied halftime games using the following two scenarios to avoid redundancy: in the first, the New/Rare and Side Strength features are excluded, and in the second, Pick Map is excluded. The results can be seen in Table 4 and Appendix D.

Table 4: Model Mean Test Accuracy Results for Tied Halftime Games

Model	**Without New/Rare and Side Strength**	**Without Pick Map**
Heuristic	48.23% ± 4.23%	48.23% ± 4.23%
Decision Tree	55.65% ± 8.48% (depth: 7)	50.97% ± 6.17% (depth: 5)
Random Forest	55.00% ± 6.94% (depth: 5)	50.81% ± 5.87% (depth: 2)
SVM: linear	56.45% ± 5.73%	48.39% ± 4.78%
SVM: rbf	54.19% ± 5.64%	50.48% ± 6.25%
SVM: poly	55.16% ± 4.66%	51.61% ± 6.77%
SVM: sigmoid	50.48% ± 5.50%	45.32% ± 4.86%

Compared to models trained on the full dataset, these models show a substantial drop in mean test accuracy. The models perform only slightly better than random guessing, particularly in the scenario that excludes the Pick Map feature, where most accuracies hover around 50%. In contrast, the first scenario achieves a mean test accuracy of around 55%, showing a slight improvement. However, both scenarios have high standard deviations, reflecting the smaller dataset size. The SVM models generally perform the best, achieving the highest accuracies with relatively moderate standard deviations.

The feature importance tables in Appendix D show a wider spread of important features. The Halftime Lead feature has 0% importance, confirming that it has no predictive power. In the scenario excluding Pick Map, two features stand out. The Side Strength and Picked Attack feature stand out as the top two features, likely because of how they interact, as they reflect the balance between sides on certain maps.

The decision trees, found in the appendix, give further insight. In the first scenario, the tree mainly splits on the Pick Map feature. It is interesting to see that splits after the Pick Map splits seem to make the model overfit, with predictions that do not change or leaf nodes with very few samples.

The decision tree from the second scenario highlights the dynamic between Side Strength and Picked Attack. When the map is defense-sided (according to Side Strength) and Team Y picked attack, 9 out of 11 maps in the training set resulted in Team Y winning. This makes intuitive sense, as Team Y tying the score at halftime while playing on the weaker side should give them the advantage. However, similar overfitting issues as in the first scenario are observed.

7.5 Simplified Model Performance on Tied Halftime Games

Because of the observed overfitting in the previous models it is interesting to look at a simplified model. Figure 18 shows that splits after the Pick Map splits contribute little to the model's performance. It might be valuable to only look at the Pick Map feature, simplifying the model. The results for this approach can be seen in Table 5 and Figure 6.

Table 5: Model Mean Test Accuracy Results for Tied Halftime Games including only Pick Map

Model	**Mean Test Accuracy (%)**
Heuristic	48.23% ± 4.23%
Decision Tree (depth: 7)	58.06% ± 5.00%
Random Forest (depth: 9)	56.13% ± 3.80%
SVM: linear	57.10% ± 5.01%
SVM: rbf	56.77% ± 5.34%
SVM: poly	57.10% ± 5.01%
SVM: sigmoid	56.77% ± 5.34%

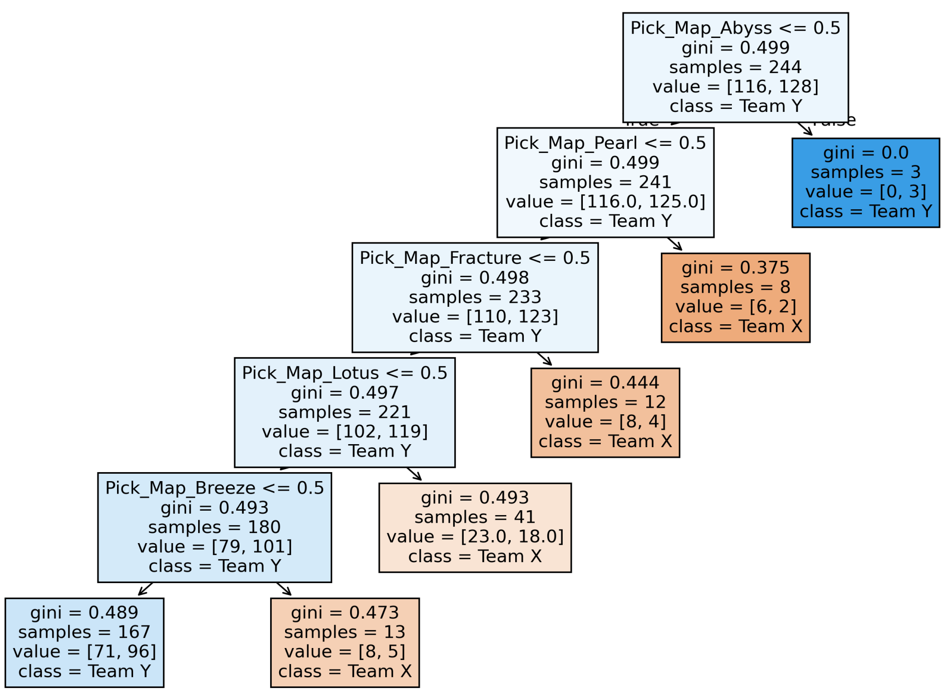

Figure 6: Decision tree visualization for tied halftime games including only Pick Map (max_depth=5)

Simplifying the model by including only the Pick Map feature improves both the mean test accuracy and the standard deviation. All models achieve an accuracy above 56%, with standard deviation consistently around 5%. Although the tree-based models show a large depth, this is explained by the way the trees are build, which can be seen in Figure 6. The model splits on the map played, with every depth corresponding to a specific map. This results in a simple and interpretable model. Despite the smaller dataset for tied halftime games, this simplified approach avoids overfitting and shows improvements.

8 Conclusion & Further Research

In this thesis, the possibilities of applying machine learning models on a limited esports dataset were investigated. Although the models achieved notable accuracies of around 76.5%, this is only a slight improvement over the basic heuristic used as a benchmark. Given that Halftime Lead dominated the predictive power, an approach that excluded this feature was explored. This shifted the focus to modeling the subset of tied halftime games, which further reduced the already limited dataset and further limits confidence in the generalizability of the model results. These models showed a substantial drop compared to models trained on the full dataset, at best averaging around 56% with a high standard deviation. Simplifying the model to include only one feature, Pick Map, slightly improved its performance. In this simplified scenario, all models achieved an average test accuracy of around 57%, with a reduced standard deviation compared to the other models. This appears to be a meaningful model, as it suggests that the map feature could provide insight into map-specific dynamics. Additionally, previous models using the Side Strength and Picked Attack features suggest there is predictive power behind the side dynamics on a map. Although the average accuracies remain close to random guessing, the results suggest that slight improvements are possible using a simple model.

Throughout the thesis, we explored different sides of the machine learning process. First, the necessary data for data analysis had to be gathered. The data had to be processed so that it could be used as input for analysis and models. Data analysis helped identify patterns that could potentially be used as a focus for predictive modeling. Finally, the modeling process demonstrated the potential of machine learning on a limited dataset.

This thesis explored the challenges associated with the analysis of limited esports data. This iterative process highlighted several areas of improvement necessary in this field and for similar research, such as:

- **Dataset Size**: Since we worked with a limited dataset, expanding it is important. This can be achieved through additional data collection via web scraping, accessing the Valorant API or using external data sources. Unfortunately, I missed a Kaggle dataset that contains detailed match data from 2021 to 2024 during this research, but it could be a valuable data source for future studies (Kaggle Dataset).
- **Data Collection**: While the dataset was limited in total size, it was also limited in detail. In future work, additional data should be gathered per match, such as the team compositions or round-specific data.

- **Data Storage and Management**: Ensuring data are stored in a structured and consistent format encourages future research by making it easier to collect the necessary data.
- **Feature Optimization**: With an improved dataset, new features can be engineered, potentially improving the model performance. Additionally, automatic feature selection can further optimize the model's input.
- **Additional Models**: Implementing other models than tree-based algorithms and Support Vector Machines could give further insights.
- **Optimizing Model Performance**: Although this thesis explored different parameters and techniques, a more systematic approach could improve model performance. For example, optimizing hyperparameters for SVMs might give better results. In addition, future work could look at pruning the decision tree for improved performance.

In situations where data is limited, it is especially important to manage the data carefully. Keeping the data structured, free of errors and consistently updated allows for more studies to be carried out. By following these points, we can improve the field of esports data analytics and potential future work.

References

[AHDD19] R. Ani, Vishnu Harikumar, Arjun K. Devan, and O.S. Deepa. Victory prediction in League of Legends using Feature Selection and Ensemble methods. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 74–77, 2019.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Cho20] Gene Chorba. VALORANT API Launch and Policies. https://www.riotgames.com/en/DevRel/valorant-api-launch, July 2020. Senior Developer Relations at Riot Games.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[DWY+21] Tiffany D. Do, Seong Ioi Wang, Dylan S. Yu, Matthew G. McMillian, and Ryan P. McMahan. 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*, FDG '21, New York, NY, USA, 2021. Association for Computing Machinery.

[Espa] Esports Charts. Top Games by Peak Viewership – Esports Charts. https://escharts.com/top-games?order=peak&year=all-time.

[Espb] Esports Charts. Top Games by Peak Viewership – Esports Charts. https://escharts.com/top-games?order=peak.

[Far24] Leo Faria. Twitter post. https://x.com/lhfaria/status/1865033516813074744, 2024. Global Head of Valorant Esports.

[Fon] Junior Fonseca. Valorant Predictor. https://github.com/Juniorffonseca/valorant-predictor.

[HDS+21] Victoria J. Hodge, Sam Devlin, Nick Sephton, Florian Block, Peter I. Cowling, and Anders Drachen. Win Prediction in Multiplayer Esports: Live Professional Match Prediction. *IEEE Transactions on Games*, 13(4):368–379, 2021.

[Hen21] Henrik-3. Unofficial Valorant API. https://github.com/Henrik-3/unofficial-valorant-api, 2021.

[HGMFBC23] Juan Agustín Hitar-García, Laura Morán-Fernández, and Verónica Bolón-Canedo. Machine Learning Methods for Predicting League of Legends Game Outcome. *IEEE Transactions on Games*, 15(2):171–181, 2023.

[Lam] Adrian Lam. Valorant Map Predictor. https://github.com/adclama9/Valorant_map_predictor.

[Mil05] Jeremy Miles. *R-Squared, Adjusted R-Squared.* John Wiley & Sons, Ltd, 2005.

[Rio21a] Riot Games. Valorant API Documentation. https://developer.riotgames.com/docs/valorant, 2021.

[Rio21b] Riot Games, Inc. Riot Games Developer Portal. https://developer.riotgames.com/, 2021.

[Rio23] Riot Games. VCT23 Official Competition Ruleset, April 2023. Pages 19–21.

[Rio24] Riot Games. Valorant Official Competition Ruleset, January 2024. Pages 19–21.

[Sad] Andre Saddler. vlrggapi. https://github.com/axsddlr/vlrggapi.

[Sch20] Zachary Schmidt. Esports Match Result Prediction for a Decision Support System in Counter-Strike: Global Offensive. Master's thesis, Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Leiden, The Netherlands, July 2020. Supervised by Dr. Mike Preuss and Dr. Rens Meerhoff.

[SPC18] Antonio Luis Cardoso Silva, Gisele Lobo Pappa, and Luiz Chaimowicz. Continuous outcome prediction of league of legends competitive matches using recurrent neural networks. *SBC-proceedings of SBCGames*, pages 2179–2259, 2018.

[VAL23] VALORANT. UNITED TOGETHER // China Launch Official Cinematic - VALORANT. https://www.youtube.com/watch?v=F5qEtdVtsvo, July 2023.

[Wik] Valorant Wiki. Lotus Minimap. https://valorant.fandom.com/wiki/Lotus?file=Lotus_minimap.png.

A Map Veto Rules

The following tables outline the map veto rules used in the analysis.

A.1 BO3 Format

Table 6: Map Veto Rules for BO3 Format

Map Veto Rules for BO3 Format
Team A bans one map.
Team B bans one map.
Team A picks map 1.
Team B picks side for map 1.
Team B picks map 2.
Team A picks side for map 2.
Team A bans one map.
Team B bans one map.
Map 3 is the only map remaining.
Team A picks side for map 3.

A.2 BO5 Format

Table 7: Map Veto Rules for BO5 Format

Map Veto Rules for BO5 Format
Team A bans one map.
Team B bans one map.
Team A picks map 1.
Team B picks side for map 1.
Team B picks map 2.
Team A picks side for map 2.
Team A picks map 3.
Team B picks side for map 3.
Team B picks map 4.
Team A picks side for map 4.
Map 5 is the only map remaining.
Team B picks side for map 5.

A.3 Double Elimination Grand Final Format

Table 8: Map Veto Rules for Double Elimination Grand Final

Map Veto Rules for Double Elimination Grand Final
Upper Bracket Team bans two maps.
Team A picks map 1.
Team B picks side for map 1.
Team B picks map 2.
Team A picks side for map 2.
Team A picks map 3.
Team B picks side for map 3.
Team B picks map 4.
Team A picks side for map 4.
Map 5 is the only map remaining.
Team B picks side for map 5.

B Differential Analysis

The following figures visualize the differential analysis performed in search of a research question.

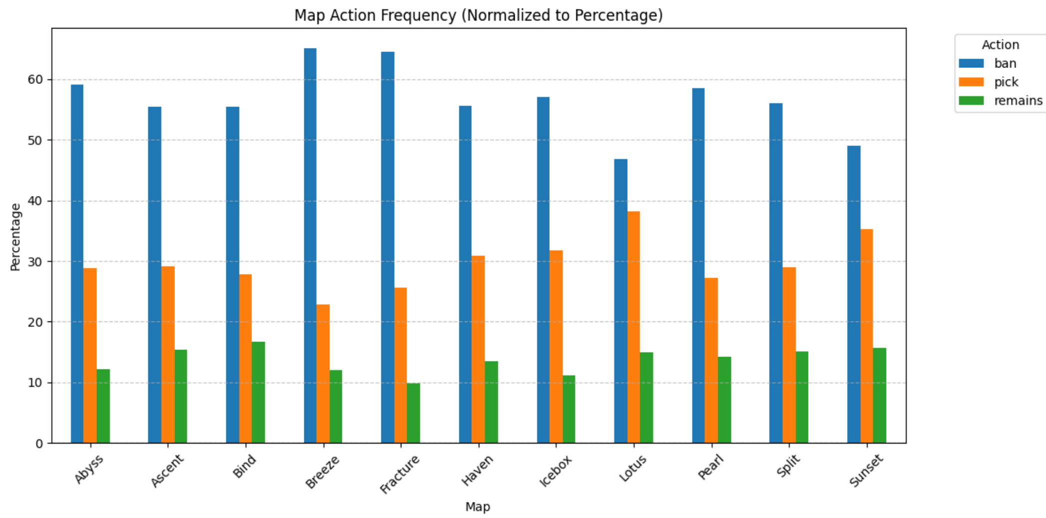

Figure 7: Normalized percentage breakdown of map actions (ban, pick, remain) for each map.

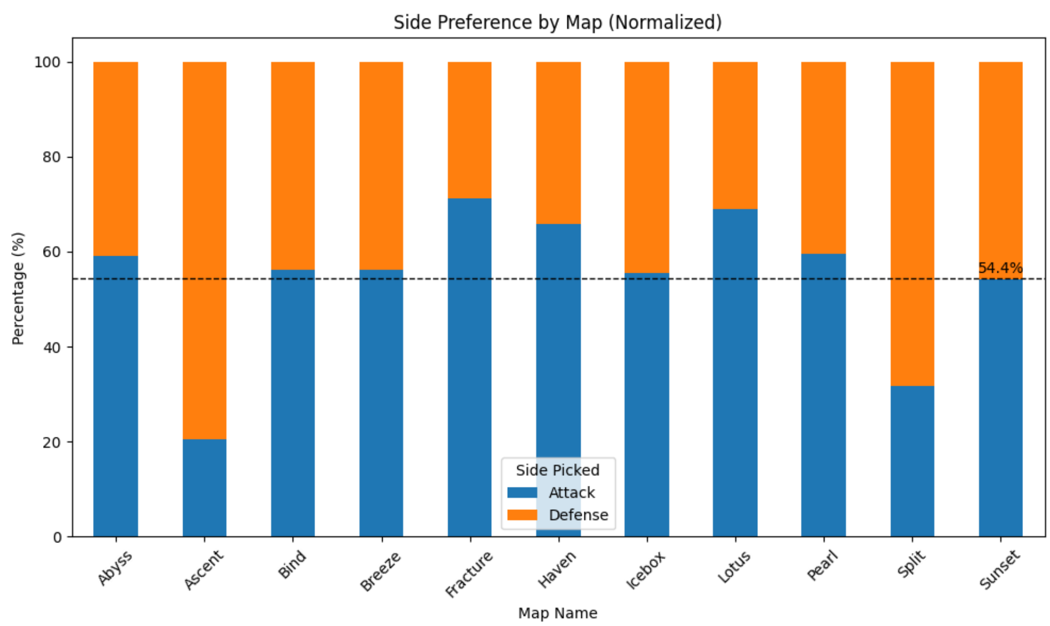

Figure 8: Side preferences (Attack vs. Defense) for each map, normalized to percentage values.

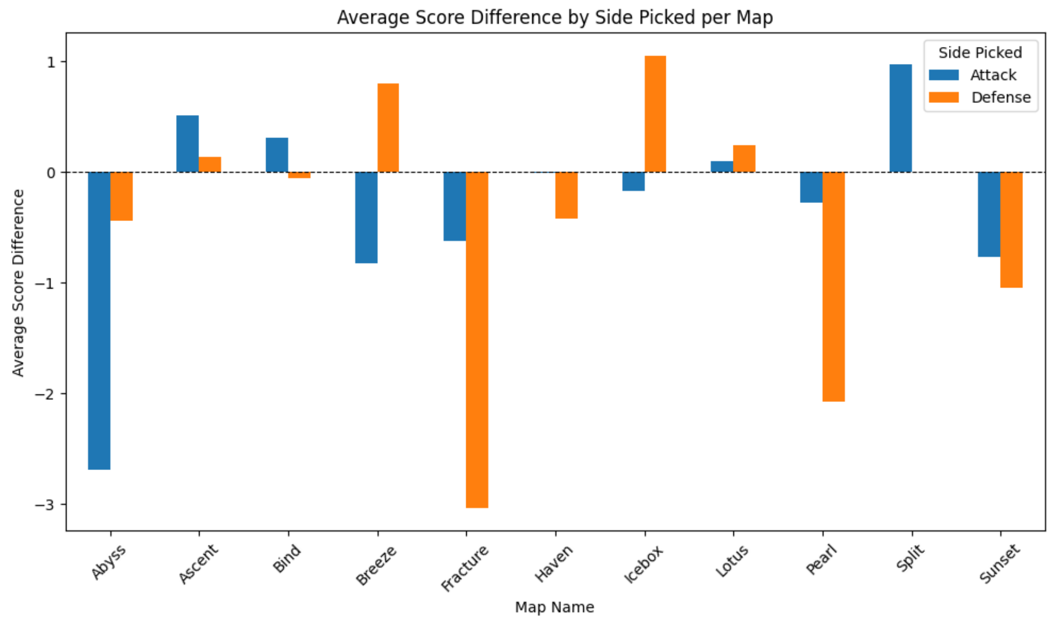

Figure 9: Comparison of average score differences when Attack or Defense is picked for each map.

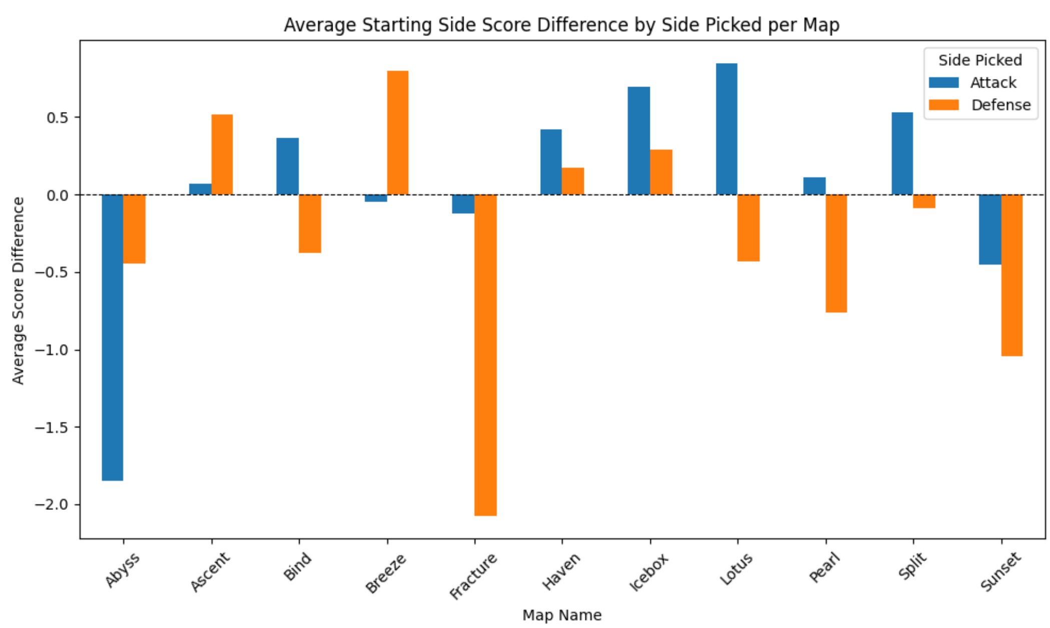

Figure 10: Average first half score difference when Attack or Defense is picked for each map.

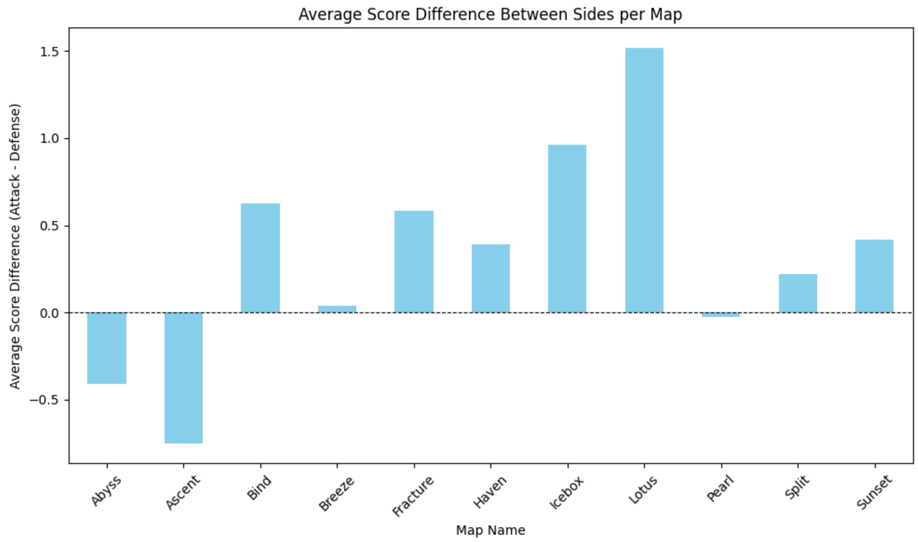

Figure 11: Average score difference (Attack - Defense) for each map, indicating map-specific balance.

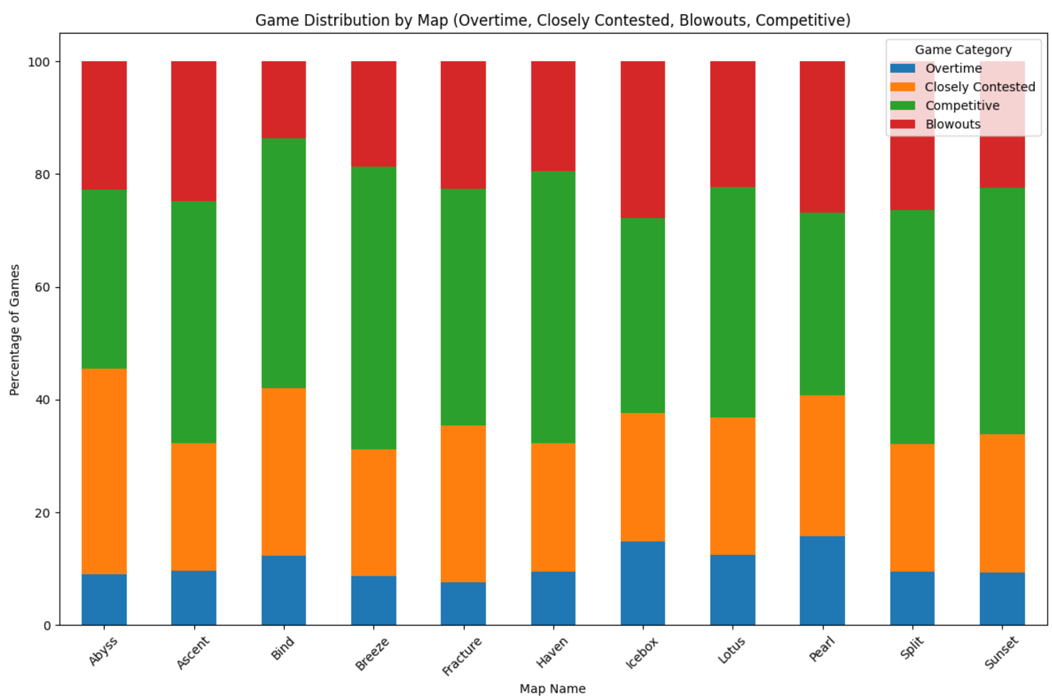

Figure 12: Percentage breakdown of game outcomes (Overtime, Closely Contested, Competitive, Blowouts) across maps.

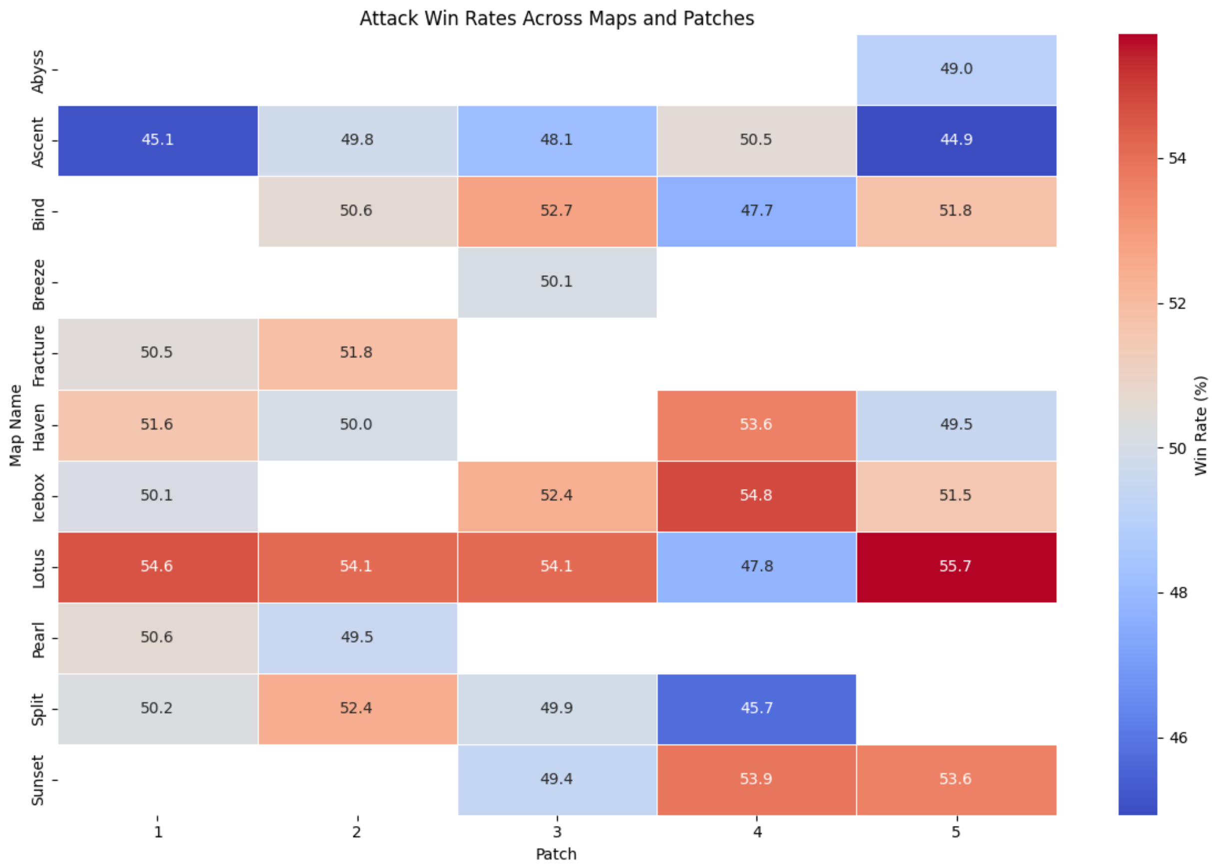

Figure 13: Heatmap showing attack win rates (%) across maps and patches, highlighting trends in attacker performance over time. The defense win rate heatmap is its opposite.

C Initial Model Results

This section presents results generated during the initial modeling process.

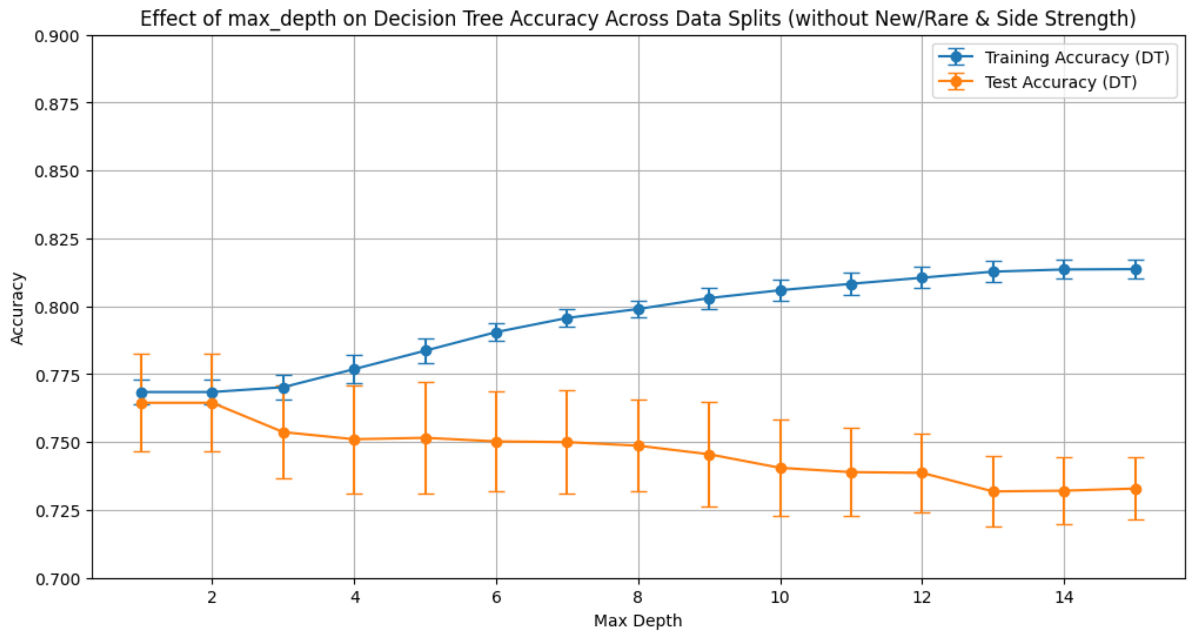

Figure 14: Train and test accuracy vs. max_depth for the Decision Tree model (excluding New/Rare & Side Strength).

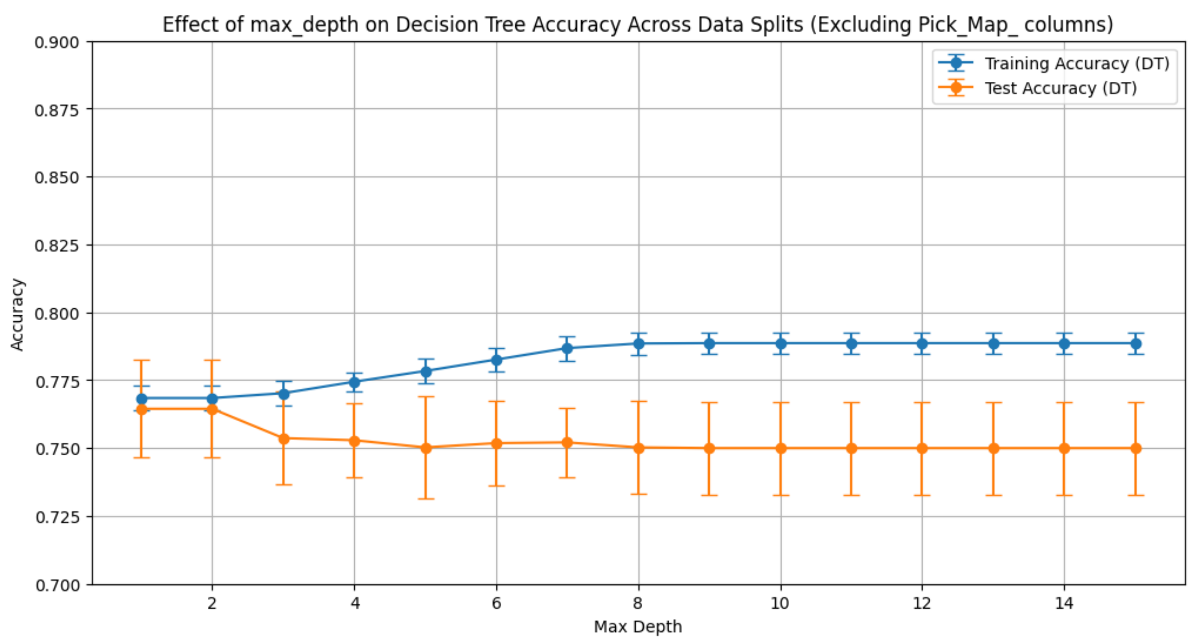

Figure 15: Train and test accuracy vs. max_depth for the Decision Tree model (excluding Pick Map feature).

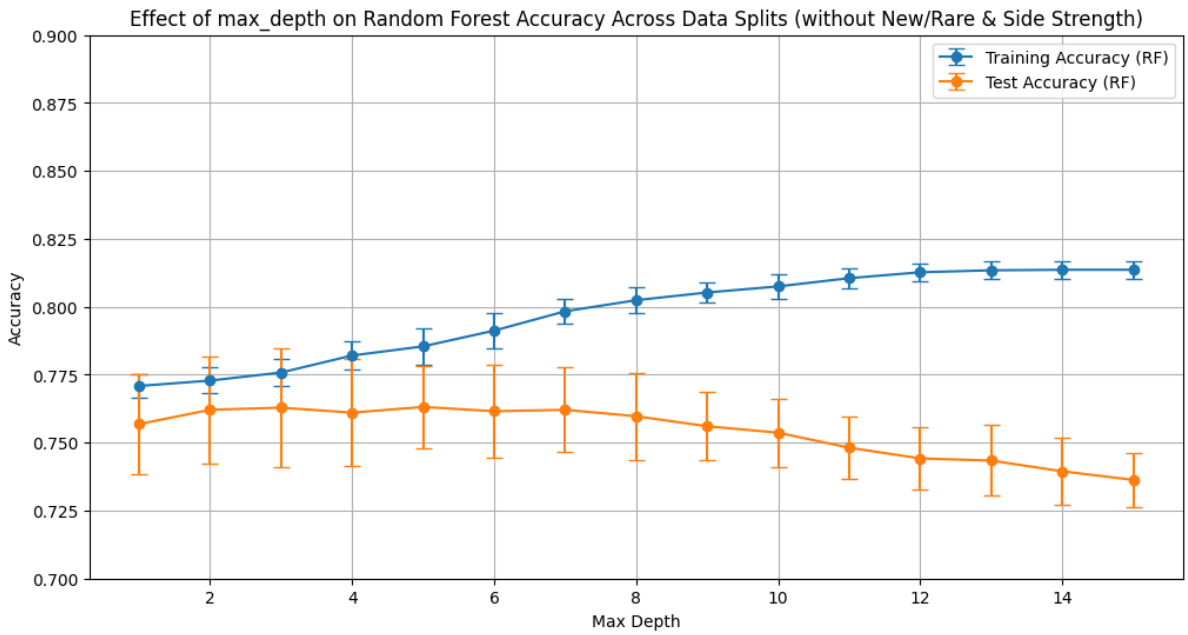

Figure 16: Train and test accuracy vs. max_depth for the Random Forest model (excluding New/Rare & Side Strength).

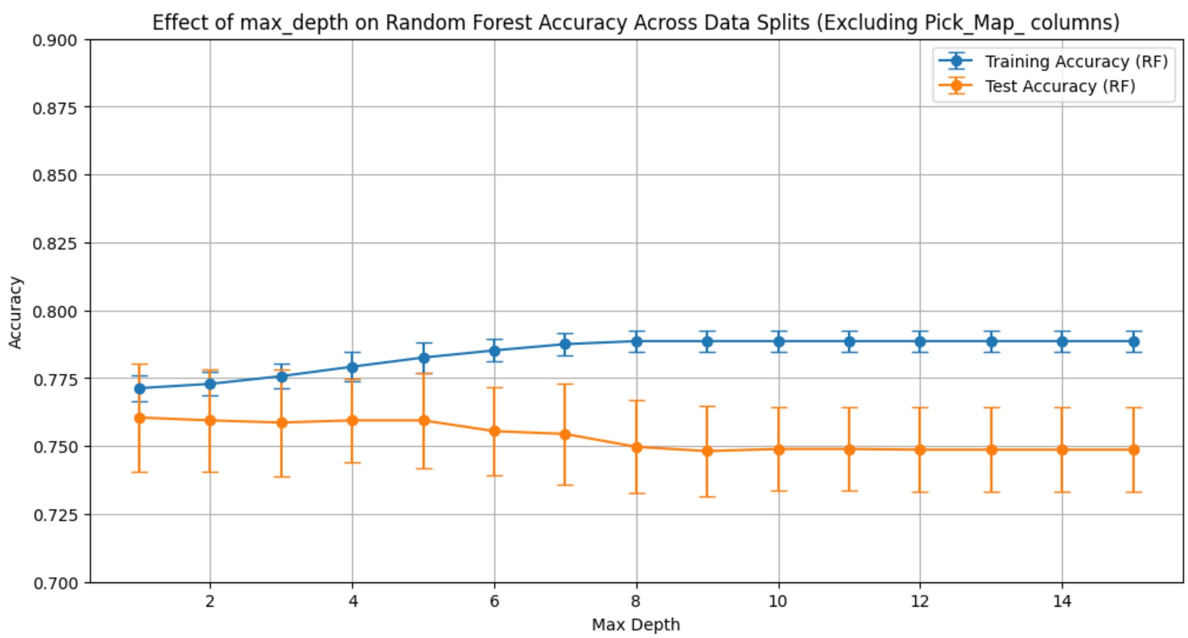

Figure 17: Train and test accuracy vs. max_depth for the Random Forest model (excluding Pick Map feature).

Table 9: Feature Importance excluding New/Rare and Side Strength

Feature	**Importance**
Halftime Lead	0.608543
Picked_Attack	0.070925
Pick_Map_Sunset	0.066856
Pick_Map_Pearl	0.060414
Neutral Map	0.044264
Pick_Map_Fracture	0.042516
Pick_Map_Bind	0.019166
Pick_Map_Breeze	0.018780
Pick_Map_Lotus	0.015382
Pick_Map_Icebox	0.011923
Pick_Map_Abyss	0.011389
Pick_Map_Split	0.010687
Pick_Map_Haven	0.010647
Pick_Map_Ascent	0.008508

Table 10: Feature Importance excluding Pick Map

Feature	**Importance**
Halftime Lead	0.749966
New/Rare	0.102663
Picked_Attack	0.060456
Side Strength	0.055911
Neutral Map	0.031003

D Model Results for Tied Halftime Games

This section presents results generated during the modeling process of tied halftime games.

Table 11: Feature Importance for Tied Halftime Games excluding New/Rare and Side Strength

Feature	**Importance**
Pick_Map_Pearl	0.124565
Picked_Attack	0.116987
Pick_Map_Sunset	0.115483
Neutral Map	0.102564
Pick_Map_Fracture	0.100429
Pick_Map_Abyss	0.097802
Pick_Map_Breeze	0.080775
Pick_Map_Lotus	0.070889
Pick_Map_Icebox	0.063851
Pick_Map_Bind	0.047462
Pick_Map_Split	0.045686
Pick_Map_Haven	0.021852
Pick_Map_Ascent	0.011654
Halftime Lead	0.000000

Table 12: Feature Importance for Tied Halftime Games excluding Pick Map

Feature	**Importance**
Side Strength	0.341562
Picked_Attack	0.258460
New/Rare	0.209114
Neutral Map	0.190865
Halftime Lead	0.000000

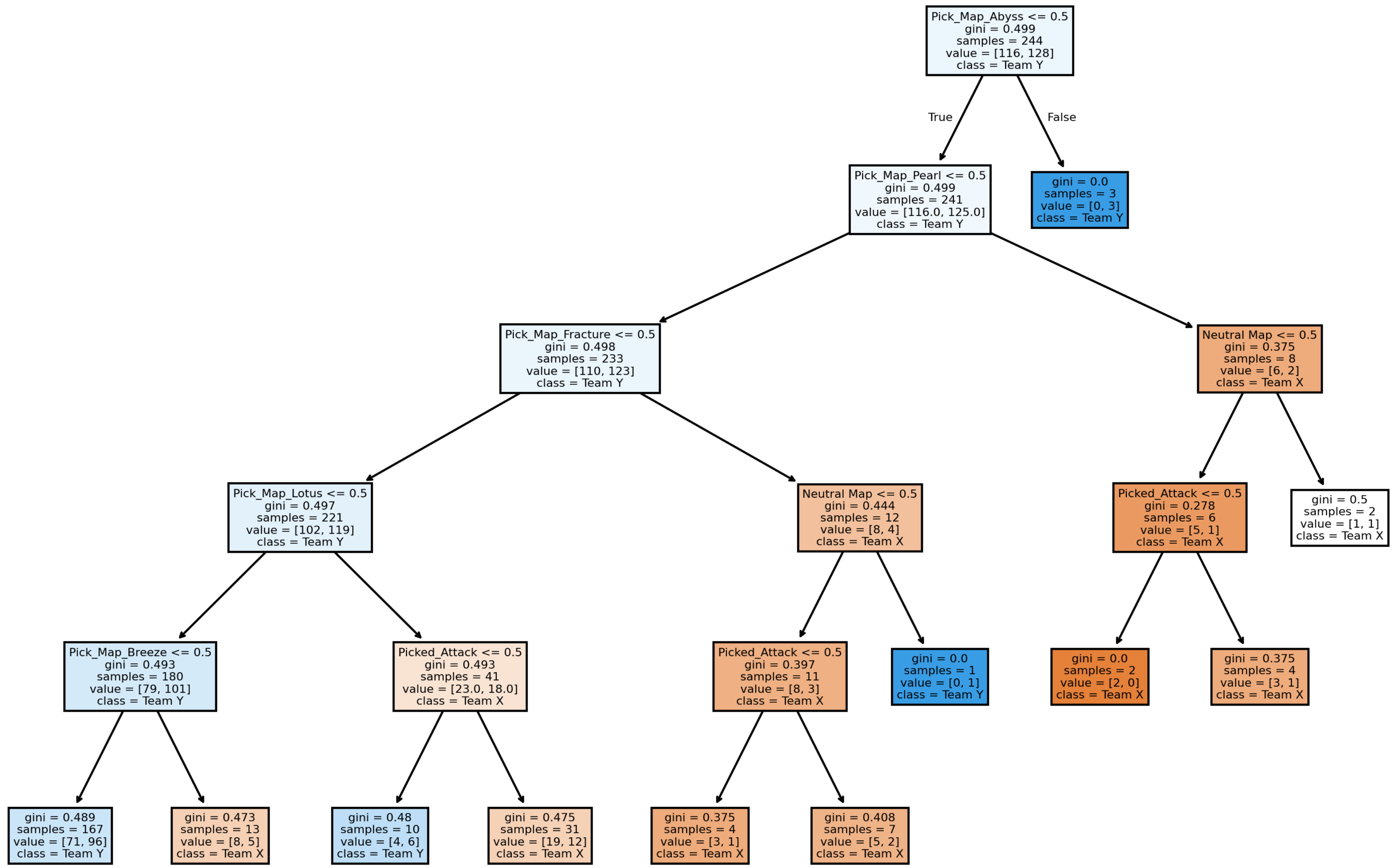

Figure 18: Decision tree visualization for tied halftime games excluding New/Rare and Side Strength (max_depth=5).

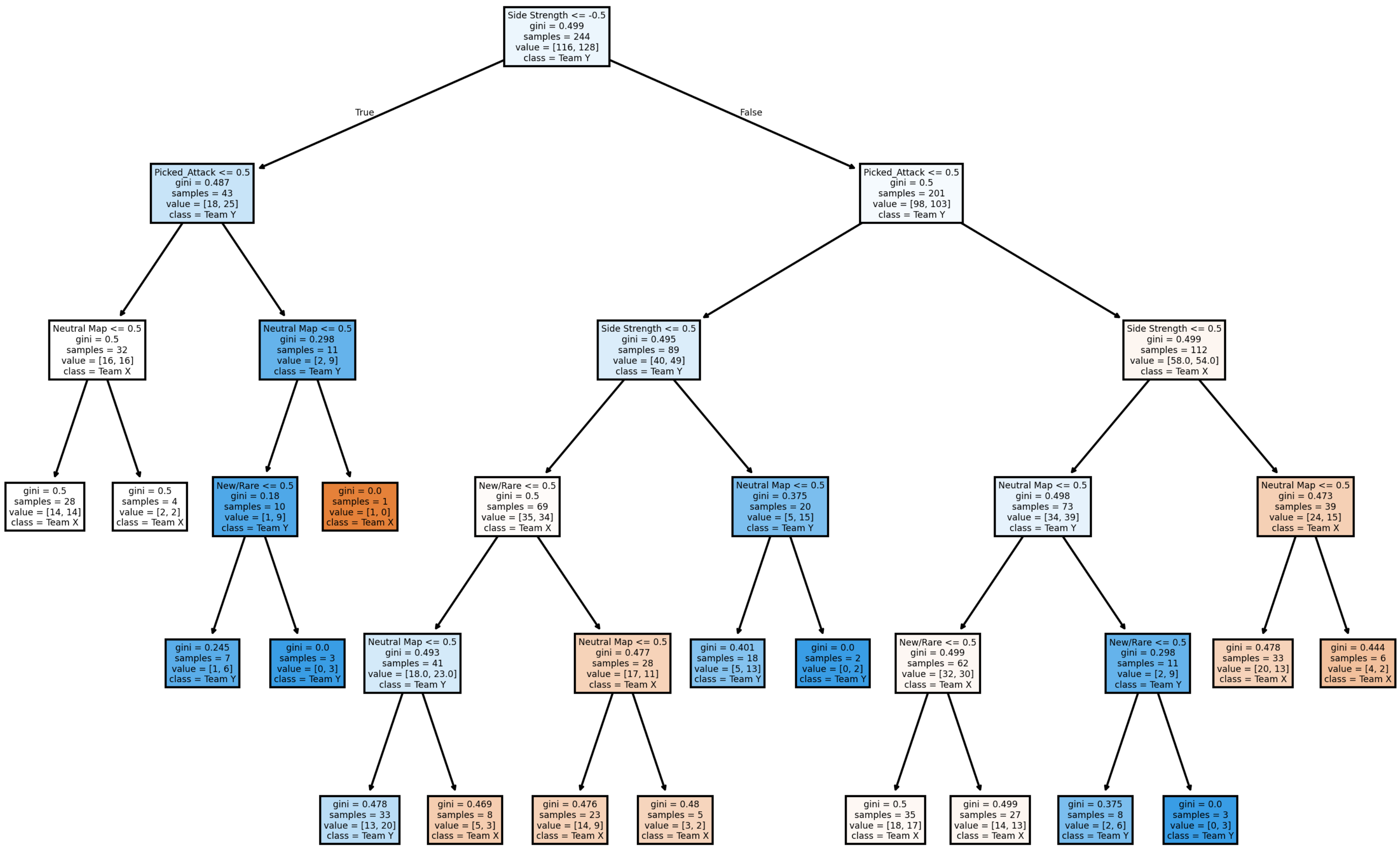

Figure 19: Decision tree visualization for tied halftime games excluding Pick_Map_ (max_depth=5).