Starcraft 2 Performance

An In-Depth Look At In-Game Telemetry And Player Rank

Abstract

In the burgeoning field of esports, the intersection of machine learning and video game analysis offers new opportunities for discovering complex patterns in player performance and strategy development. Previous research from Thomas et al. has focused on complex skill learning through video games, and Google's Starcraft AI AlphaStar has shown Starcraft II as a focal point for machine learning and analytics. In this paper, we will dissect the multifaceted nature of esports with a focus on Starcraft II. Statistical analysis via ANOVA and pairwise T-tests has highlighted the significant impact of race on Actions per Minute (APM) and Screens per Minute (SPM), with Zerg players notably achieving higher metrics. Despite linear regression's success in predicting APM, classifying player ranks posed a considerable challenge. Our findings demonstrate that while predictive modeling can illuminate aspects of in-game performance, quantifying the nuanced and comprehensive skill set of players remains a complex endeavor that requires further research.

Introduction

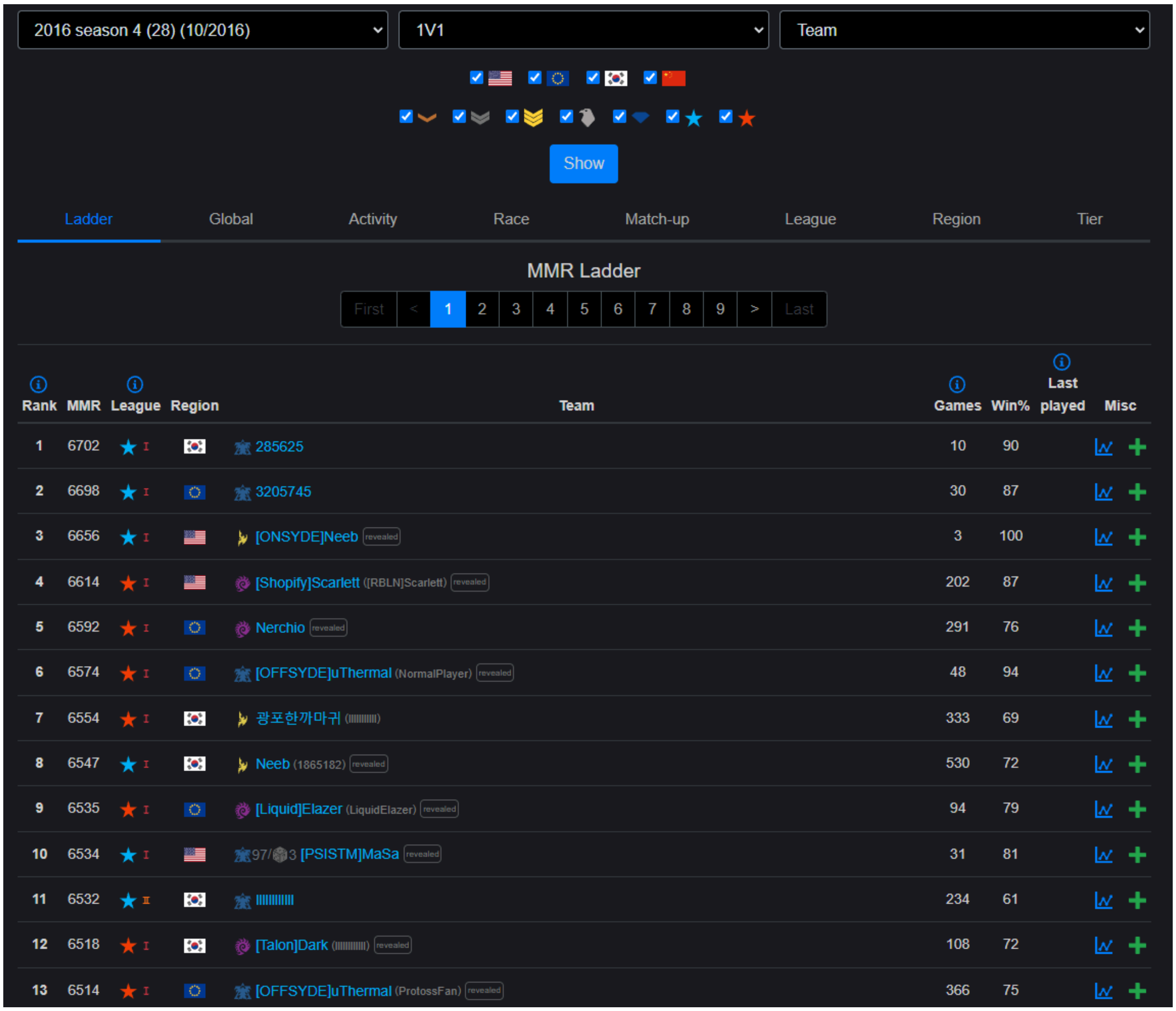
In the rapidly evolving landscape of digital entertainment, competitive gaming, commonly known as esports, has emerged as a phenomenon captivating millions worldwide. The roots of esports trace back to the early days of video gaming, where the spirit of competition found its expression in local arcades and small tournaments. Despite its humble beginnings, the 21st century marked a pivotal moment in gaming history as internet connectivity and advancements in technology catalyzed the exponential growth of esports. Today, esports encompasses a wide range of games, genres, and formats; it has fostered a global community of professional players, enthusiasts, and spectators, reaching a level of ubiquity that led to its inauguration into the Olympics in 2023. Amidst this backdrop, analyzing esports through machine learning presents a novel approach to understanding the intricacies of competitive gaming at a granular level.

This paper attempts to bridge the gap between the historical evolution of esports and the analytical depth that machine learning can provide to esports research. Our primary focus is on analyzing the godfather of esports, Starcraft. Before the analysis, we will set the stage with a high-level overview of the esports scene in the 21st century. We then perform an analysis of the SkillCraft1 Master Table Dataset. Our objective is to illuminate the multifaceted nature of player performance, strategy development, and learning processes within the competitive realm of StarCraft II (SC2), thereby offering insights into the real-time strategy (RTS) genre, as well as the broader domain of esports. We aim to contribute a novel perspective to the interdisciplinary field of game studies, one that underscores the potential of data-driven research in unlocking the complexities of digital competition.

Data

Data Collection

Most of the data used in the analysis were from the UC Irvine Machine Learning Repository and Kaggle. However, to get a better overview of the SC2 landscape, we performed web scraping of a forum that tracked every player's stats for each of the four regions since 2016 (Figure 1).



Rank	MMR	League	Region	Team	Games	Win%	Last played	Misc
1	6702			285625	10	90		
2	6698			3205745	30	87		
3	6656			[ONSYDE]Neeb revealed	3	100		
4	6614			[Shopify]Scarlett ([RBLN]Scarlett) revealed	202	87		
5	6592			Nerchio revealed	291	76		
6	6574			[OFFSYDE]uThermal (NormalPlayer) revealed	48	94		
7	6554			광포한까마귀 (IIIIIIIII)	333	69		
8	6547			Neeb (1865182) revealed	530	72		
9	6535			[Liquid]Elazer (LiquidElazer) revealed	94	79		
10	6534			97/3 [PSISTM]MaSa revealed	31	81		
11	6532			IIIIIIIIII	234	61		
12	6518			[Talon]Dark (IIIIIIIII) revealed	108	72		
13	6514			[OFFSYDE]uThermal (ProtossFan) revealed	366	75		

Figure 1. **SC2 Ladder Tracker**. Data were collected from SC2Pulse for every player in each of the four regions from season 16 (October 2016) to season 58 (April 2024).

The web-scraped data were gathered for each season, separating players by their specific regions and ranks (Figure 2).

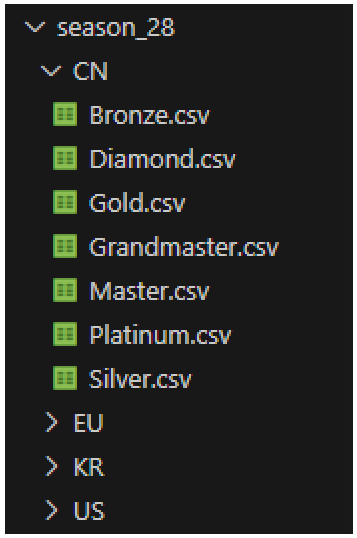


Figure 2. **Web-scraped Ladder Data**. The web-scraped ladder data was separated based on rank, region, and season.

About the Data

Based on the available filters on SC2Pulse, the fields that were saved were season, region, the player's matchmaking rating (MMR), rank, and race (Figure 3). Seasons ranged from 28 to 58, covering about eight years of data. Available regions were China, Europe, Korea, and the U.S. The player's MMR is the numeric representation of a player's skill, and is directly related to their rank. Rank refers to the different rank divisions in the ladder, which are Bronze, Silver, Gold, Platinum, Diamond, Master, and Grandmaster, ordered from worst to best. Lastly, race refers to the class a player primarily plays, with the options being Zerg, Terran, Protoss, and Random. The race Random was scraped for players who did not predominantly play a specific race. It should be noted that Random would assign one of the three other races to the player upon a match starting.

```
Season,Region,Rating,Rank,Race
28,CN,1686,Bronze,Terran
28,CN,1674,Bronze,Terran
28,CN,1654,Bronze,Protoss
28,CN,1635,Bronze,Random
28,CN,1598,Bronze,Zerg
28,CN,1557,Bronze,Zerg
28,CN,1511,Bronze,Protoss
28,CN,1449,Bronze,Random
```

Figure 3. **Ladder Data Fields**. The fields extracted from the SC2Pulse website were the season (ranging from 28 to 58), region (China, Europe, Korea, and the U.S.), the player's MMR (i.e. a numeric representation of a player's skill), rank (Bronze, Silver, Gold, Platinum, Diamond, Master, Grandmaster), and race (Zerg, Terran, Protoss, and Random).

Data Cleaning

Before applying any machine learning models to our primary dataset (i.e. SkillCraft1 Master Table Dataset), we removed any rows with missing values. Upon further inspection, these rows primarily related to players classified as "Professional." However, considering that the pros are an extreme minority of the player base, this seemed to be a pretty reasonable design choice. Additionally, uninformative columns (e.g. Game ID) were dropped.

For the statistical tests, where we used the Intel Extreme Masters (IEM) Katowice data from Kaggle, we removed any game where a player chose Random as their race. This was done to ensure that the statistical tests would provide insights relating to the three playable races.

Exploratory Data Analysis

After the data collection and cleaning, we visualized the data to gain a high-level understanding of the history of esports and the SC2 player base (Figure 4a).

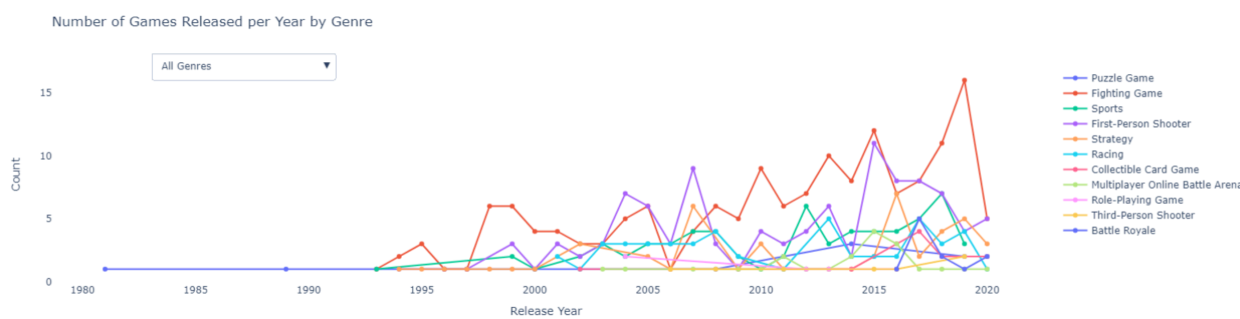


Figure 4a. **Time Series Plot of Game Releases by Genre**. Data from GGBeyond's Esports Earnings website.

The initial analysis revealed that the earliest releases of video games were in the puzzle genre (Figure 4b). However, as computing and graphic design improved, there was a rise in other genres that are still popular to this day, such as fighting games and first-person shooters (FPS) (Figure 4c). Unsurprisingly, it seems that the video game industry also experiences a sort of ebb and flow in the popularity of games as certain genres gain popularity and then lose traction again.

Figure 4b. **Time Series Plot of Puzzle Games**. Puzzle game releases have continued since 1980, peaking in 2015.

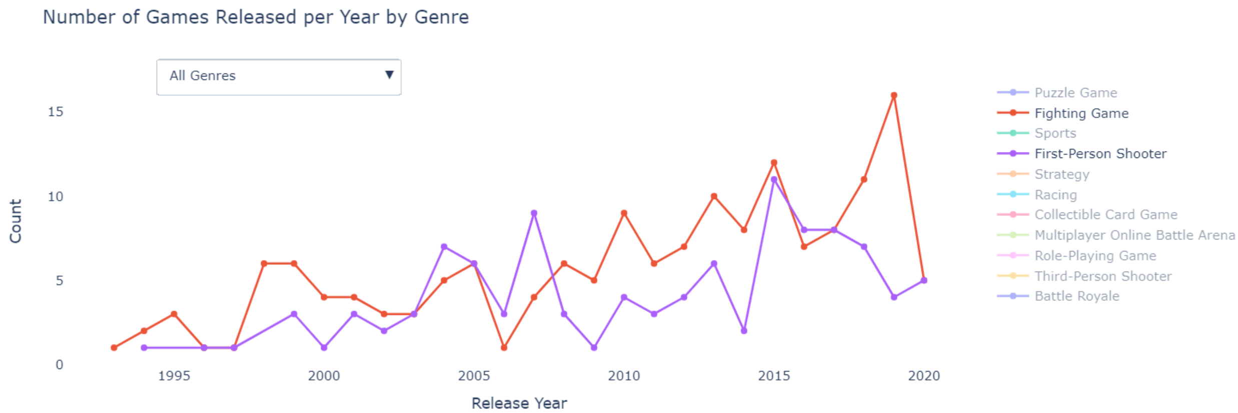


Figure 4c. **Time Series Plot of Fighting and FPS Games**. Fighting and FPS games have seen sustained popularity since their introduction in the mid-90s.

Next, we examined the popularity of the genres and specific titles (Table 1). When accounting for data until 2020, the Multiplayer Online Battle Arena (MOBA) genre had the highest tournament earnings, primarily from games like Dota 2 and League of Legends (Appendix 1 and 2). Interestingly, despite Fighting games having the greatest number of titles released per year (as depicted in Figures 4a and 4c), they don't seem to have tournaments with large prize pools. This is likely due to the relatively casual nature of mainstream titles such as Super Smash Bros Ultimate and Melee (Appendix 3 and 4).

Table 1. **Tournament Earnings based on Genre**. Data from GGBeyond's Esports Earnings website.

Genre	Total Amount of Money Earned
Multiplayer Online Battle Arena	354,299,247.51
First-Person Shooter	228,333,624.76
Battle Royale	132,922,522.56
Strategy	64,507,955.97
Collectible Card Game	34,271,397.99
Sports	26,182,617.56
Fighting Game	20,170,322.19
Racing	7,753,196.92
Role-Playing Game	5,968,597.31
Third-Person Shooter	4,901,986.38
Puzzle Game	39,082.97

Statistical Analysis

Actions per minute (APM) is a common statistic used to describe a player's capabilities in SC2. The ability to multitask and optimize each second makes APM a reliable metric for determining a player's skill and knowledge. Additionally, screens per minute (SPM) is used to measure how quickly one moves their screen around the map. A higher SPM generally means more map information for the player, which is invaluable for RTS games where decisions are made based on incomplete information. This statistical analysis will explore pro-level matches from IEM Katowice 2016 to 2023 to examine whether certain races require different levels of APM/SPM to secure a win through ANOVA tests.

In addition to investigating just the best players, we will conduct a follow-up with pairwise T-tests to investigate the effects of APM on a player's Rank using the SkillCraft1 Dataset.

For all hypothesis tests conducted below, they were performed with 95% confidence (or alpha=0.05).

APM ANOVA Test

For the APM ANOVA test, we proceeded with the following hypotheses:

Null hypothesis: The mean APM is the same across all races.
Alternate hypothesis: The mean APM is different for at least two races.

To get a baseline of how APM is distributed, we plotted boxplots for each race. As shown in Figure 5, Zerg has the highest APM, followed by Terran and Protoss. Protoss and Terran also seem to have a relatively normal APM distribution, whereas Zerg is right-skewed.

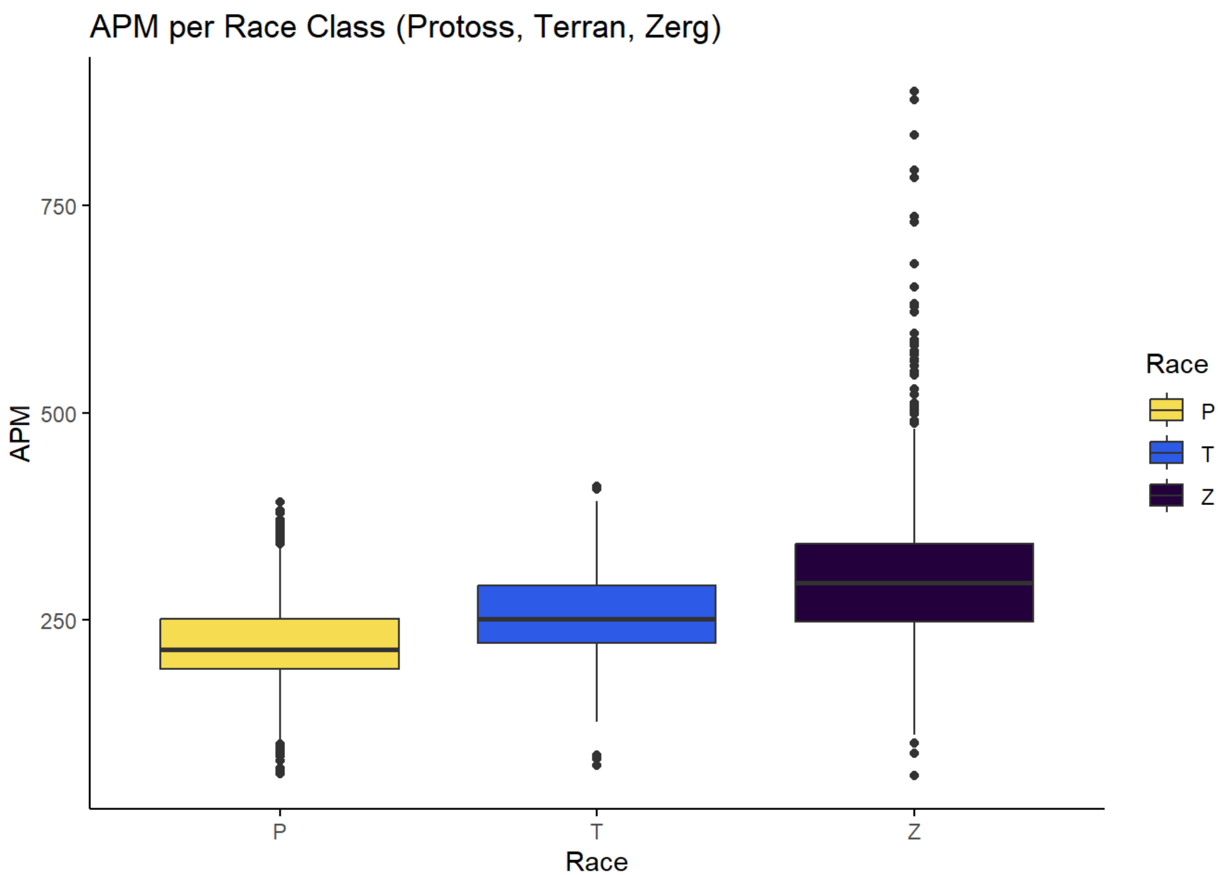


Figure 5. **Boxplot of APM by Race**. Zerg has the highest APM, followed by Terran, and then lastly Protoss.

We then checked the normality of the residuals for APM given race (Figure 6). Although the histogram isn't perfectly normal, we'll continue since ANOVA is quite robust to small deviations from normality.

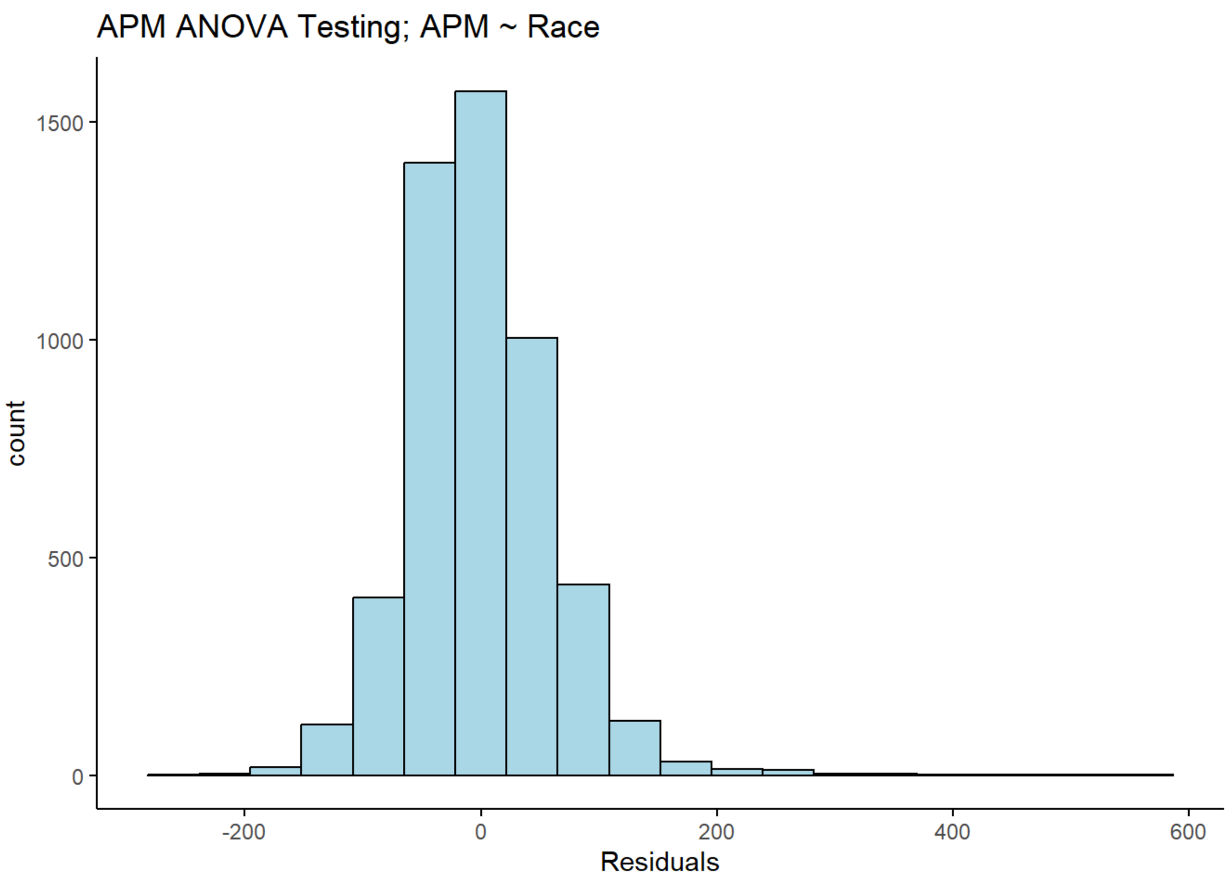


Figure 6. **Histogram of Residuals for APM by Race**. The residuals appear roughly normally distributed.

After performing the ANOVA test, we got a statistically significant effect of Race on APM, which has been detailed in Table 2. The F-statistic, with 3 levels (representing the races) and 5157 degrees of freedom, is 745.32. This large value suggests a strong difference between the mean APMs between the races. This is further supported by the p-value < 0.001, indicating very strong evidence that rejects the null hypothesis. Eta-squared, a measurement of effect size, represents the proportion of the total variance in APM that is explained by race, and a value of 22% suggests that race does have a large impact on APM.

Table 2. **ANOVA Test Results for APM by Race**. The F-statistics, with 3 levels and 5157 degrees of freedom, is 745.32, the p-value is less than 0.001, and Eta-squared is 0.22.

Metric	**Value**
F-statistic (2, 5157)	745.32
P-value	< 0.001
Eta-squared	0.22

SPM ANOVA Test

For the SPM ANOVA test, we proceeded with the following hypotheses:

Null hypothesis: The mean SPM is the same across all races.
Alternate hypothesis: The mean SPM is different for at least two races.

We followed the same workflow as the APM ANOVA test. As shown in Figure 7, Zerg has the highest SPM, followed by Protoss and Terran.

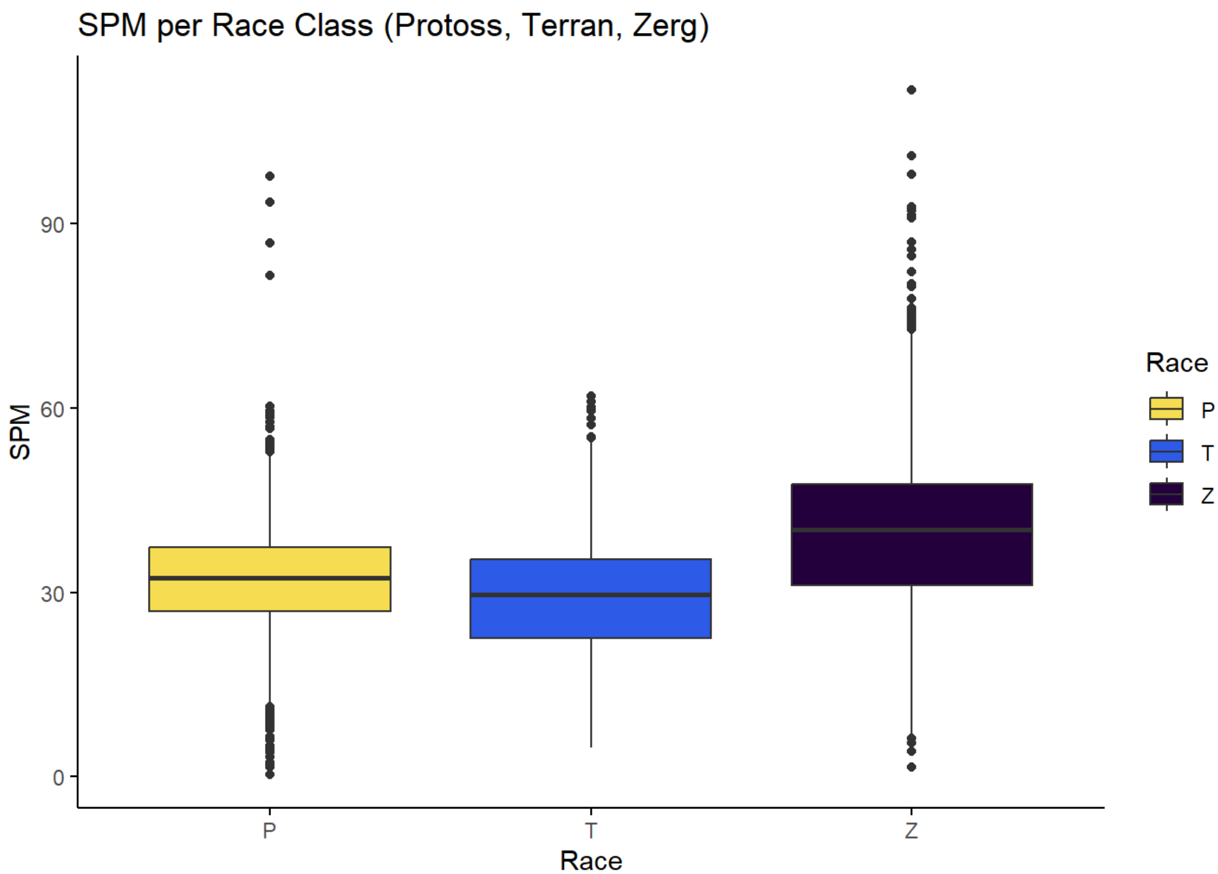


Figure 7. **Boxplot of SPM by Race**. Zerg has the highest SPM, followed by Protoss, and then lastly Terran.

We also checked the normality of the residuals for SPM given race (Figure 8).

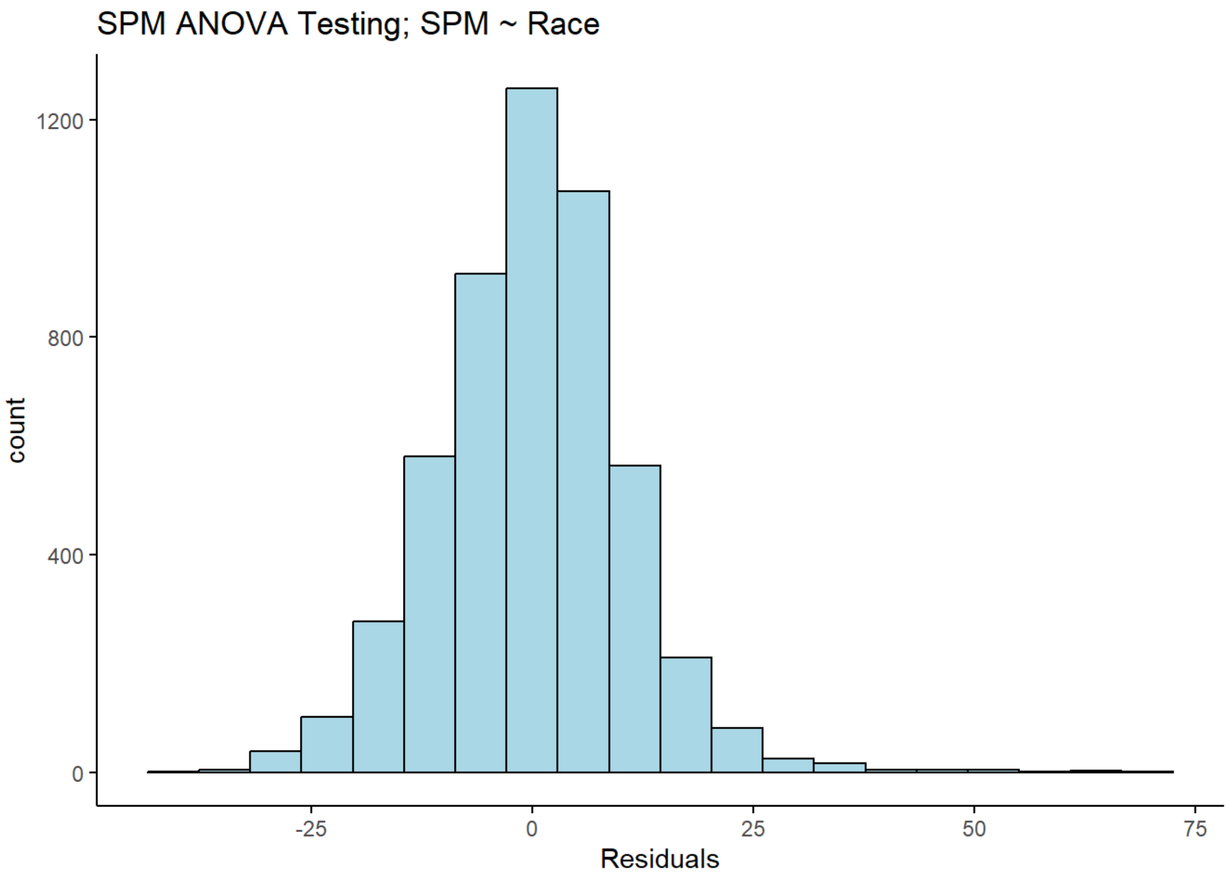


Figure 8. **Histogram of Residuals for SPM by Race**. The residuals appear roughly normally distributed.

After performing the ANOVA test, we got a statistically significant effect of Race on SPM, which has been detailed in Table 3. The F-statistic, with 3 levels (representing the races) and 5157 degrees of freedom, is 421.61. Once again, this value suggests a strong difference between the mean SPMs between the races. This is further supported by the p-value < 0.001, indicating very strong evidence that rejects the null hypothesis. Eta-squared of 14% suggests a moderate relationship between the total variance in SPM that is explained by race.

Table 3. **ANOVA Test Results for APM by Race**. The F-statistics, with 3 levels and 5157 degrees of freedom, is 421.61, the p-value is less than 0.001, and Eta-squared is 0.14.

Metric	**Value**
F-statistic (2, 5157)	421.61
P-value	< 0.001
Eta-squared	0.14

APM T-Test

For the APM pairwise T-tests, we proceeded with the following hypotheses:

Null hypothesis: The mean difference between the paired rank APMs is zero.
Alternate hypothesis: The mean difference between the paired rank APMs is not zero.

With the SkillCraft1 Dataset, we will look at how APM compared at each rank through multiple pairwise T-tests. It should be noted, however, that the SkillCraft1 Dataset contains many more variables than just APM and rank. For all of the features and their corresponding explanations, refer to Appendix 5.

Nonetheless, this will give us an idea of how relevant APM is in determining a player's rank. We first explored the average APM per rank (Figure 9). Unsurprisingly, Figure 9 shows a clear, almost linear increase in average APM as rank increases.

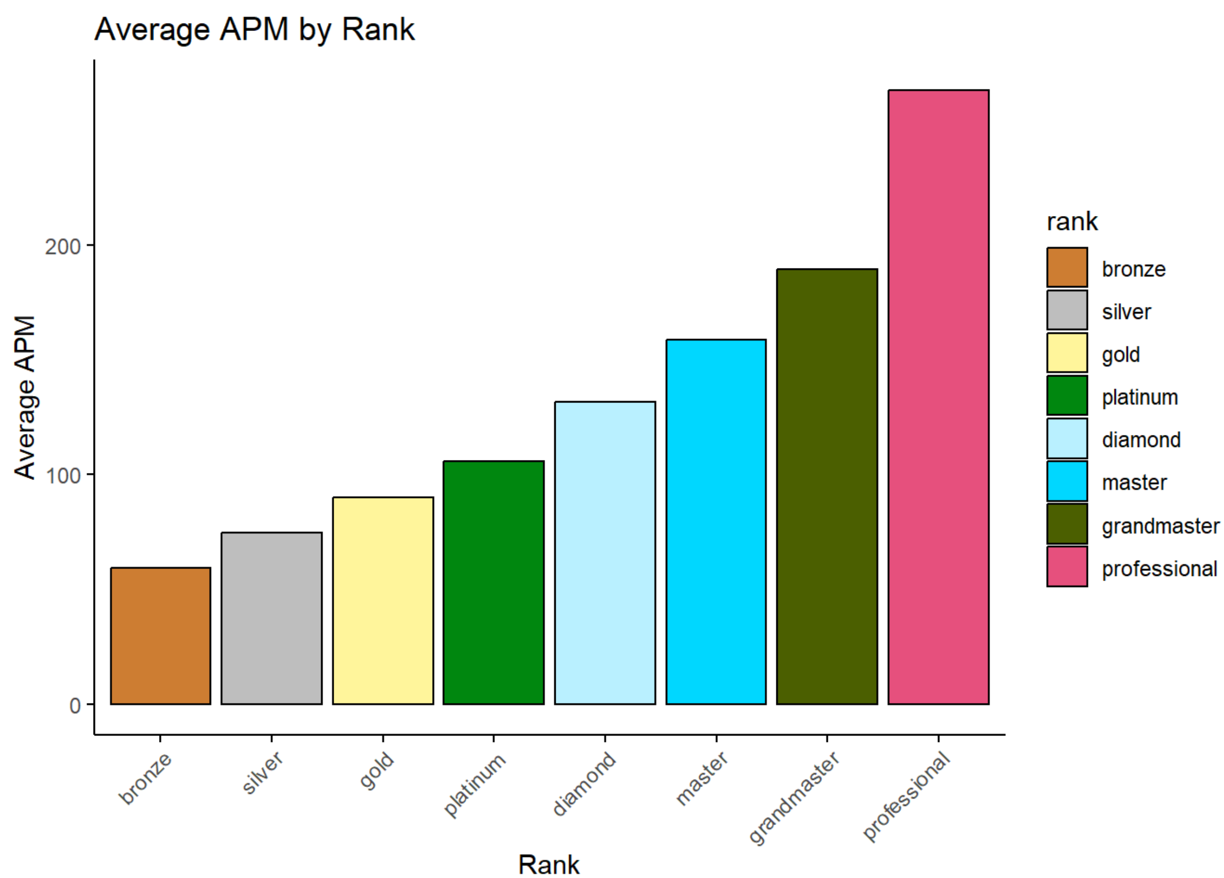


Figure 9. **Histogram of Average APM by Rank**.

Since there are eight unique ranks and we want to perform pairwise T-tests, this will give us 28 unique combinations. Figure 10 shows the p-values for each pairwise T-Test, which are all statistically significant with p-values either less than 0.001 or 0.0001.

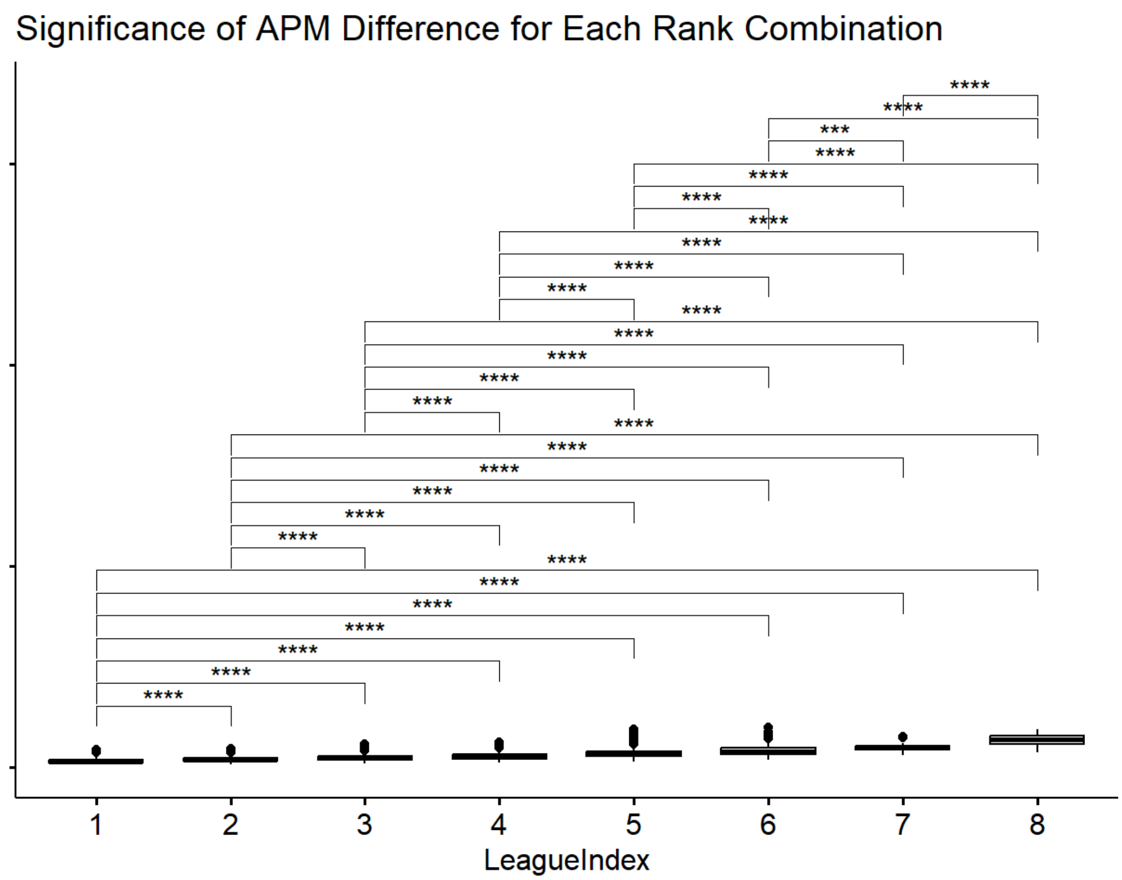


Figure 10. **Visualization of pairwise T-tests of APM by Rank.** The significance is represented as *** or ****, which corresponds to p-value < 0.0001 or p-value < 0.001, respectively.

Results

Based on these ANOVA tests, keeping the skill level consistent, we found that the average APM and SPM significantly differed from each other when considering the combinations of races from this IEM dataset. The ANOVA results are further reinforced by our insights from Figures 5 and 7, which showed Zerg having noticeably higher APM and SPM.

Additionally, Figure 11 below shows the win rates based on match-ups at the professional level. Each race has a similar win rate when considering all match-up combinations, ranging from 53 to 56%. However, when considering the ANOVA tests of APM and SPM by race, this may suggest that Zerg players have to “work harder” than Protoss or Terran players.

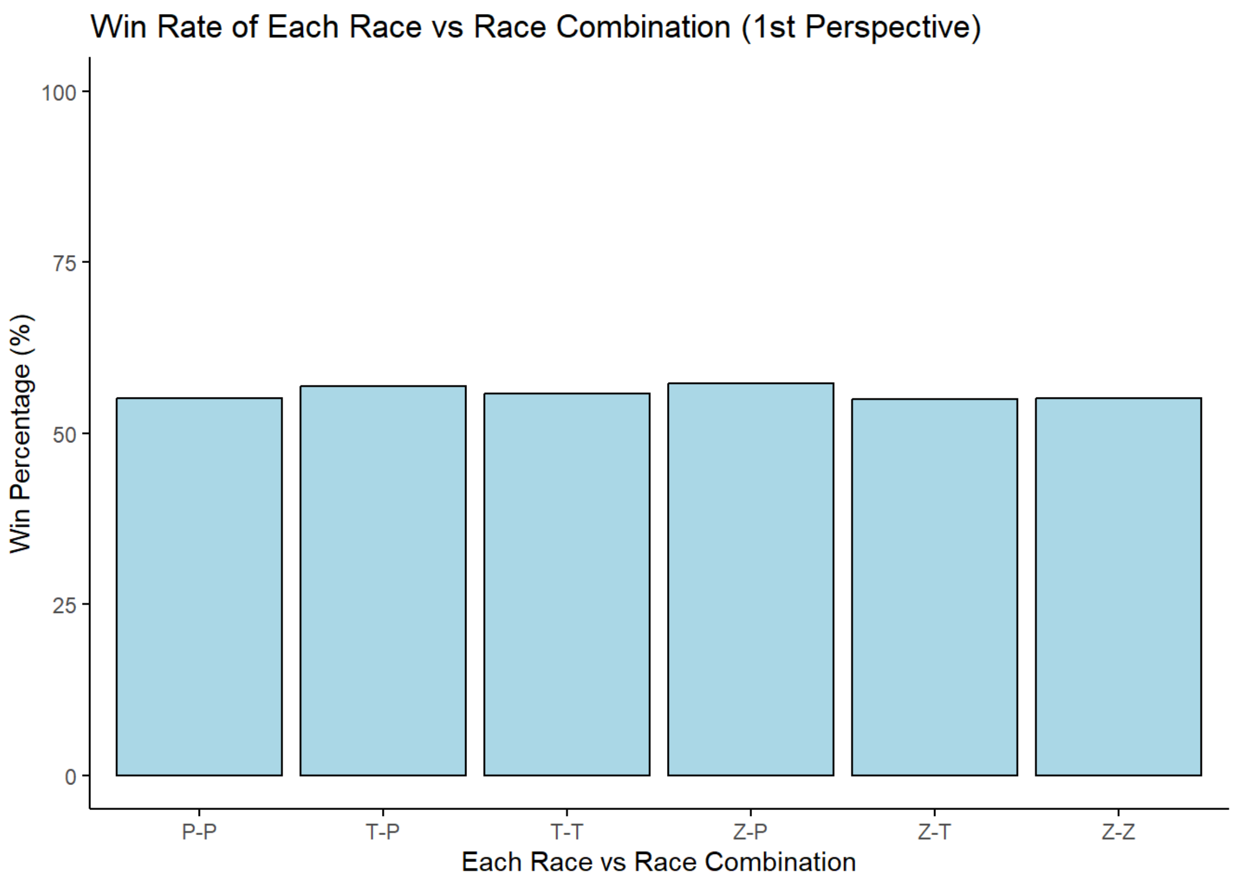


Figure 11. **Histogram of Win Rates based on Match-ups**. The win rates are from the perspective of the first race for each pair.

Similar to the findings in the ANOVA tests, we can also see that APM is also an important metric for determining a player's rank, with every pairwise T-Test resulting in a statistically significant result (i.e. every paired rank's APMs had a mean not equal to zero). There is still room for further exploration. For example, if data at user granularity can be gathered, where we have the APM, SPM, rank, and race for each player, then further testing can be done to see if the observations seen in IEM dataset are also present across different ranks. Such discovery could aid in future balancing decisions to allow for a more well-rounded gameplay experience and potentially more fair avenues for skill expression.

Machine Learning

Next, this section will delve into the application of various machine learning models to predict two primary aspects of a player's performance: APM and player rank. The first part of the analysis employs linear regression to predict APM, leveraging its continuous nature to assess how different features influence a player's ability to execute actions rapidly and efficiently.

Furthermore, we will also classify players into different ranks, a categorical outcome that encapsulates a player's overall skill. This will be done with logistic regression, support vector machines (SVMs), random forests (RFs), and XGBoost.

Exploratory Data Analysis

Before conducting any of the modeling, we performed EDA on the SkillCraft1 Dataset. First, we observed the correlation between numeric features. As seen in Figure 12, APM has a very strong positive correlation with SelectByHotkeys (0.815). This makes sense since selecting units or buildings would contribute to a player's APM. Secondly, APM has a strong negative correlation with ActionLatency (-0.722), which also is logically sound, since higher APM would require a player to make more actions, leading to less time between each action.

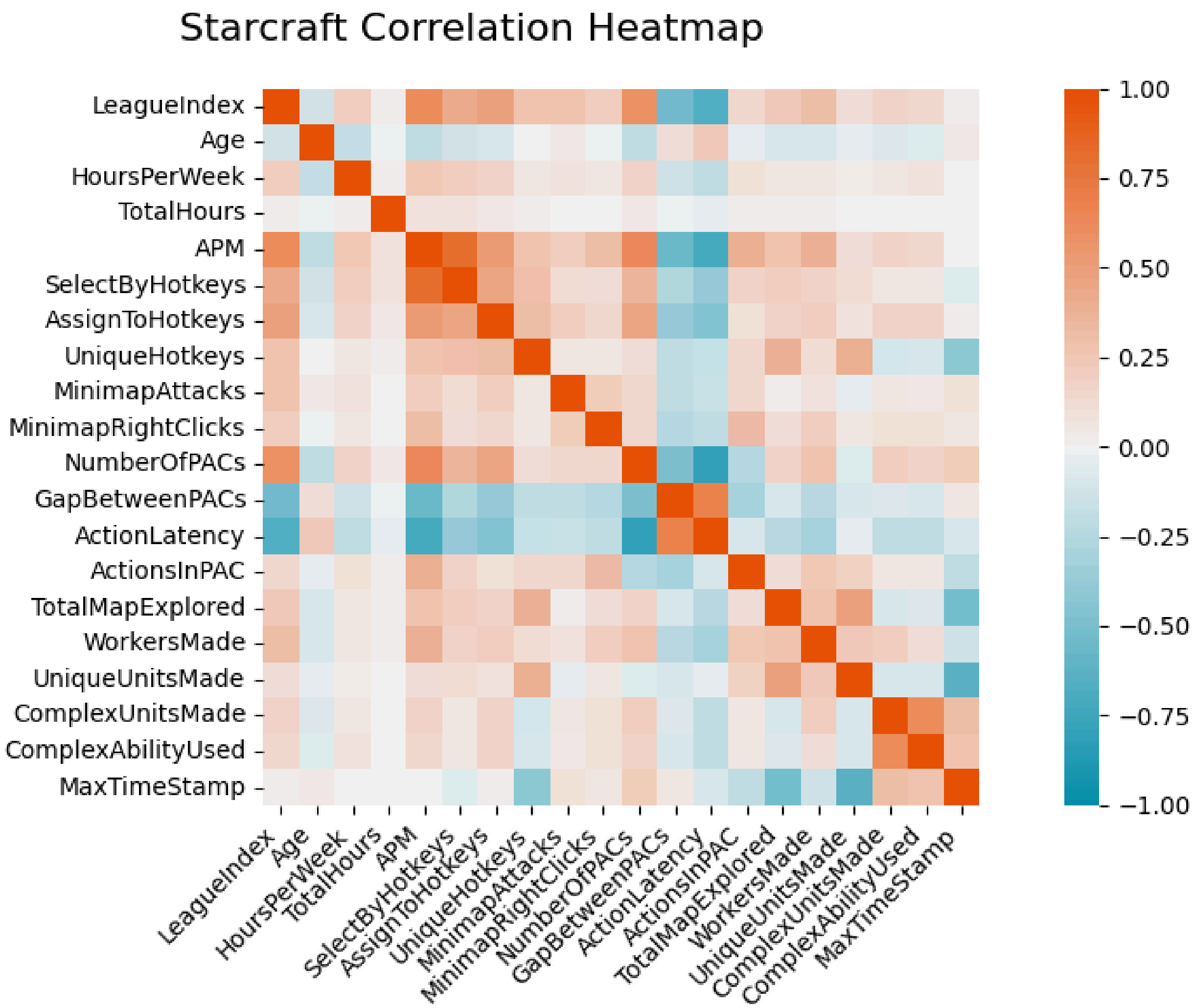


Figure 12. **Correlation Matrix of Numeric Features.** APM appears to have a very strong positive correlation with SelectByHotkeys and a strong negative correlation with ActionLatency.

We also plotted histograms to check the distributions of the features (Appendix 6-24). Most of the features appear right-skewed, though some features appear approximately normal, such as NumberOfPACs and UniqueUnitsMade. Importantly, LeagueIndex (or rank) shows that most players reside in Gold, Platinum, and Diamond.

Predicting APM: Linear Regression

In this section, our focus narrows to the intricacies of the linear regression model, with the dual objectives of discerning whether a player's APM can be accurately predicted and identifying the key features that most significantly influence the prediction of a player's APM.

The initial full model resulted in the following features detailed in Table 4, where the statistically significant features have been highlighted. Upon scrutinizing the model's summary, several predictors like HoursPerWeek and TotalHours lack statistical significance, presenting an opportunity to refine our model. Nonetheless, the model had an adjusted R-squared of 0.9755, a p-value < 2.2e-16, and an RMSE of 9.359.

Table 4. **Linear Regression Full Model**. The statistically significant features are LeagueIndex, HoursPerWeek, SelectByHotkeys, MinimapAttacks, MinimapRightClicks, NumberOfPACs, ActionLatency, ActionsInPAC, WorkersMade, UniqueUnitsMade, and ComplexAbilitiesUsed.

Feature	P-value
LeagueIndex	< 2e-16
Age	0.07678
HoursPerWeek	0.00583
TotalHours	0.57960
SelectByHotkeys	< 2e-16
AssignToHotkeys	0.18827
UniqueHotkeys	0.23226
MinimapAttacks	0.00188
MinimapRightClicks	6.37e-14
NumberOfPACs	< 2e-16
GapBetweenPACs	0.51022
ActionLatency	< 2e-16
ActionsInPAC	< 2e-16
TotalMapExplored	0.91722
WorkersMade	7.90e-14
UniqueUnitsMade	0.03079
ComplexUnitsMade	0.35672
ComplexAbilitiesUsed	9.01e-06

After employing best, forward, and backward subset selection, the optimal linear regression model consists of the features detailed in Table 5. The model had an adjusted R-squared of 0.9755, a p-value < 2.2e-16, and an RMSE of 9.358. Considering that our model complexity decreased from 18 features to 10, we achieved essentially the same level of performance in predicting a player's APM.

Table 5. **Optimal Linear Regression Model**. The statistically significant features are Age, SelectByHotkeys, MinimapAttacks, MinimapRightClicks, NumberOfPACs, ActionLatency, ActionsInPAC, WorkersMade, UniqueUnitsMade, and ComplexAbilitiesUsed.

Feature	P-value
Age	0.00685
SelectByHotkeys	< 2e-16
MinimapAttacks	0.00467
MinimapRightClicks	3.48e-14
NumberOfPACs	< 2e-16
ActionLatency	< 2e-16
ActionsInPAC	< 2e-16
WorkersMade	6.98e-14
UniqueUnitsMade	0.04833
ComplexAbilitiesUsed	6.79e-07

Predicting Rank

In this section, our exploration transitions from predicting APM as a continuous target to classifying a player's rank. We will accomplish this task by employing models, such as logistic regression, SVMs, RFs, and XGBoost. For all models, we performed a 70-15-15 train-validation-test split. For the sake of comparison, we trained a baseline model and then conducted hyperparameter tuning on the validation set to compare metrics before and after model tuning. Additionally, to account for the imbalance in LeagueIndex, as shown in Appendix 6, we also created baseline and tuned models with Synthetic Minority Oversampling Technique (SMOTE), which finds a minority class observation's k-nearest neighbors to generate new samples that combine features of the minority class sample and its neighbors. All in all, for each algorithm, there were six models, which for the sake of simplicity, will be referred to by their number mappings with the following descriptions:

1. Baseline without SMOTE balancing,
2. Tuned for accuracy without SMOTE balancing,
3. Tuned for ROC AUC score without SMOTE balancing,
4. Baseline with SMOTE balancing,
5. Tuned for accuracy with SMOTE balancing, and

6. Tuned for ROC AUC score with SMOTE balancing.

Appendix 25-28 shows the models used, respective hyperparameters, and relevant metrics, such as accuracy and ROC AUC score. For logistic regression, five of the six models performed pretty similarly to each other, with only model 4 performing noticeably worse in terms of balancing accuracy and ROC AUC score (accuracy: 0.012; ROC AUC Score: 0.619). Model 6 performed the best for SVMs (accuracy: 0.240; ROC AUC Score: 0.639), model 4 performed the best for RFs (accuracy: 0.399; ROC AUC Score: 0.802), and model 5 performed the best for XGBoost (accuracy: 0.387; ROC AUC Score: 0.802). When comparing every algorithm and the different model iterations as shown in Appendix 29 and 30, the best-performing overall model is the Random Forest model 4 (i.e. baseline with SMOTE balancing). As shown in the confusion matrix below, even the best overall model makes frequent misclassifications of a player's rank.

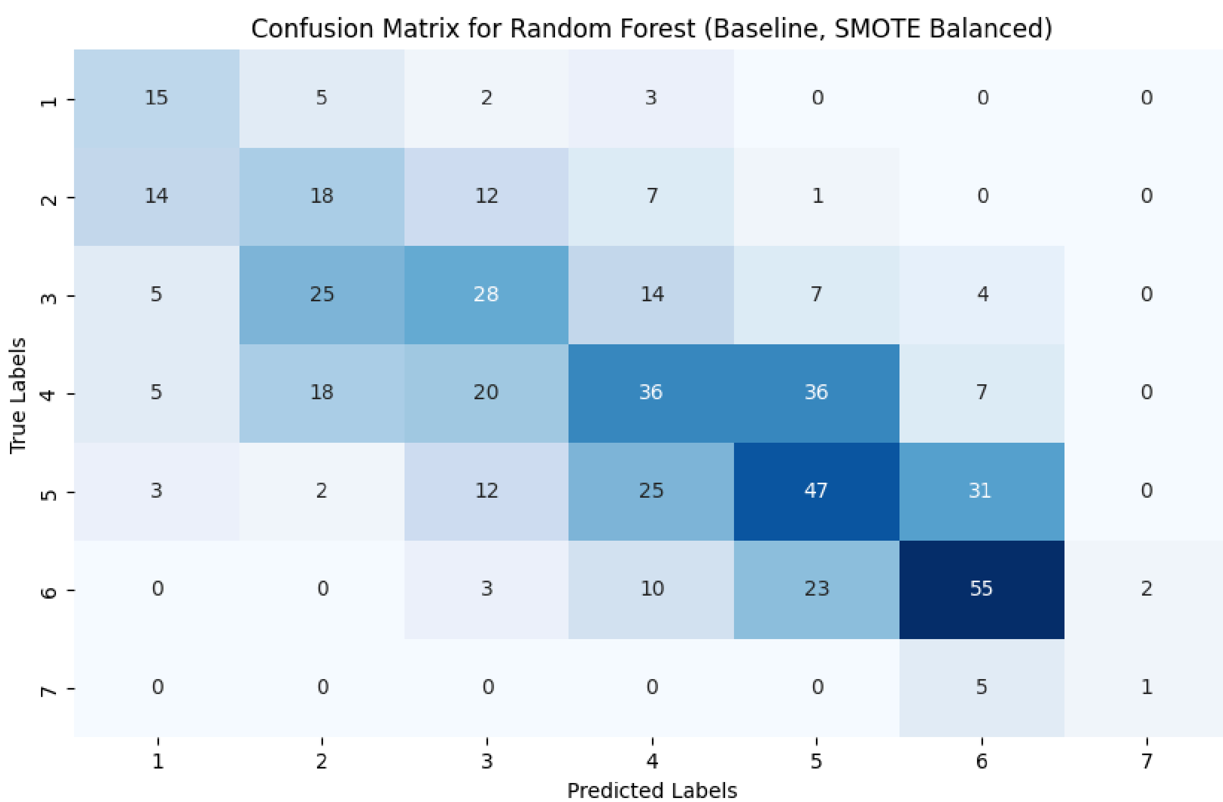


Figure 13. **Confusion Matrix of Random Forest Model 4.** The model frequently misclassifies for all classes.

Results

The linear regression model performed very well in predicting a player's APM, so much so that the pruned "optimal" version performed essentially as well as the full model. This serves as a testament to the well-defined and linearly separable nature of this particular aspect of SC2, which we initially saw in Figure 9.

However, our attempts to classify ranks revealed the challenges behind the nuanced and multifaceted nature of player skill. Despite applying a suite of machine learning models, none of the 24 models tested resulted in great performance in predicting a player's rank. Although the target variable was quite imbalanced, attempts at addressing this with SMOTE still proved unfruitful. The poor performance across all models highlights the complexity of capturing the essence of a player's rank through the gathered metrics.

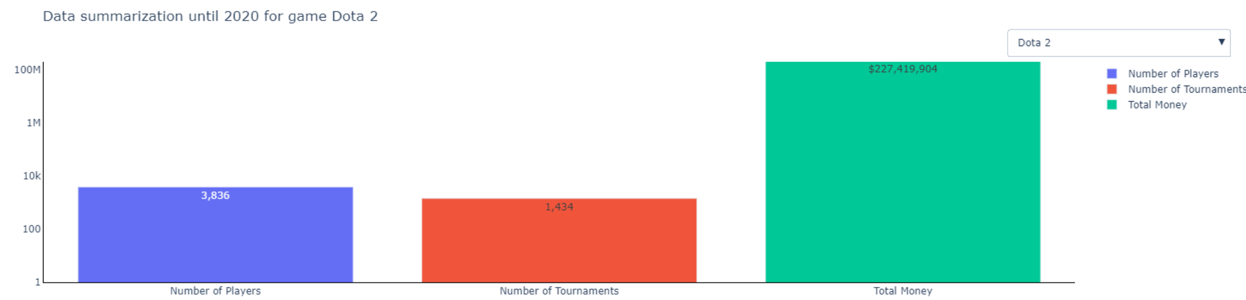
Conclusion

This paper has explored the intersection of esports analytics and machine learning with a focus on Starcraft II, a cornerstone of competitive gaming. The statistical analyses, including ANOVA and pairwise T-tests, have provided pivotal insights into the importance of APM and SPM in differentiating player capabilities across races and ranks. The ANOVA tests for both APM and SPM revealed that race significantly influences these metrics, with Zerg players exhibiting higher APM and SPM than Terran and Protoss players, suggesting inherent differences in playstyle and the strategic demands of each race. The pairwise T-tests further elucidated the relationship between APM and a player's rank, underscoring APM's role as a reliable indicator of skill expression in SC2. The analyses confirm our initial assumptions about the game's mechanics and quantify how certain in-game variables contribute to overall player performance. Potential avenues for further research could include exploring how different races' strategies or builds influence APM and SPM, as well as tracking changes in APM and SPM over time as new updates are released and meta changes occur.

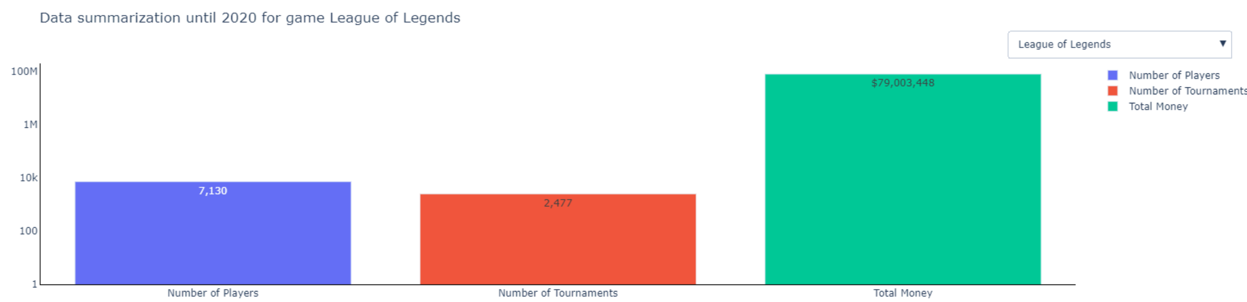
Additionally, our analysis of APM using linear regression revealed a significant, linear relationship between various gameplay metrics and a player's ability to rapidly perform actions. This model, especially the optimal version, provided a robust tool for predicting APM. Conversely, the attempt to classify player ranks proved to be quite a challenge. Despite attempts with various machine learning algorithms, hyperparameter tuning, and methods of handling imbalances in the data, fully capturing a player's performance with the given metrics alone proved insufficient. The underwhelming performance in classification shows the complexity of esports that mixes mechanical skill, decision-making, and strategy. Future studies may also try to approach the classification of player rank by gathering a more balanced dataset and applying more advanced modeling techniques, such as deep learning.

All in all, through statistical analyses and predictive modeling of Starcraft II data, this study illuminates the complex and multifaceted nature of esports. The insights garnered both advance our understanding of Starcraft II and offer a glimpse into the broader potential of data-driven research in competitive gaming.

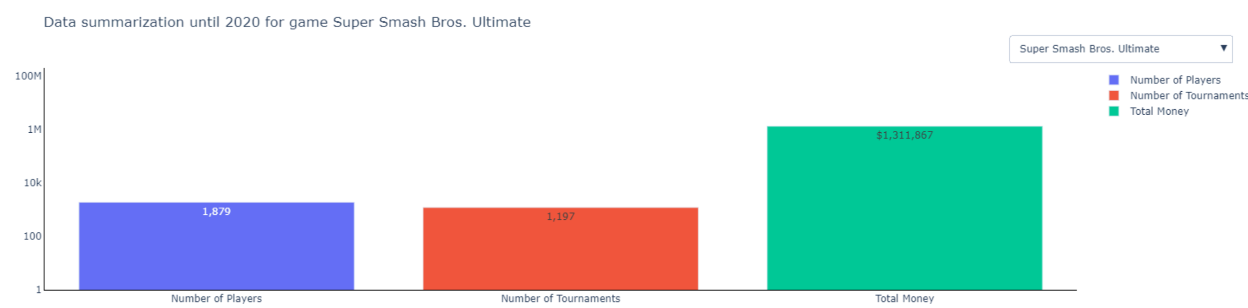
Appendix



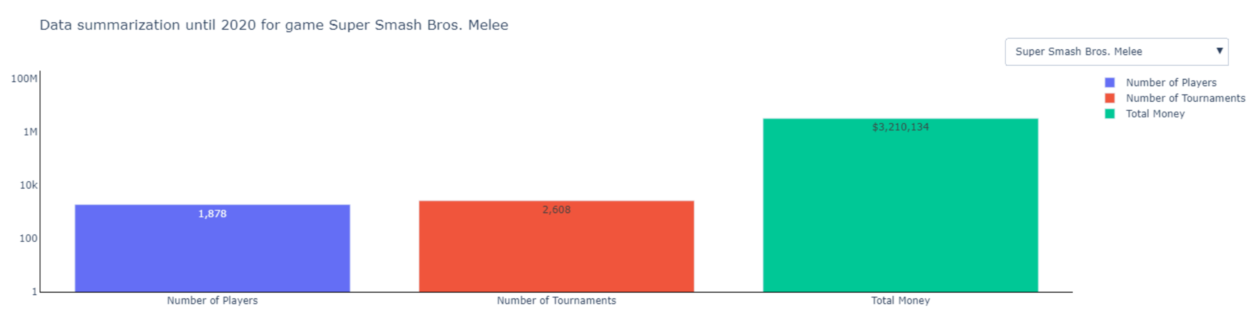
Appendix 1. **Histograms of Player Count, Tournament Count, and Total Tournament Prize Money for Dota 2**. The y-axis has been log-scaled for better visibility. Data from GGBeyond's Esports Earnings website.



Appendix 2. **Histograms of Player Count, Tournament Count, and Total Tournament Prize Money for League of Legends**. The y-axis has been log-scaled for better visibility. Data from GGBeyond's Esports Earnings website.



Appendix 3. **Histograms of Player Count, Tournament Count, and Total Tournament Prize Money for Super Smash Bros Ultimate**. The y-axis has been log-scaled for better visibility. Data from GGBeyond's Esports Earnings website.

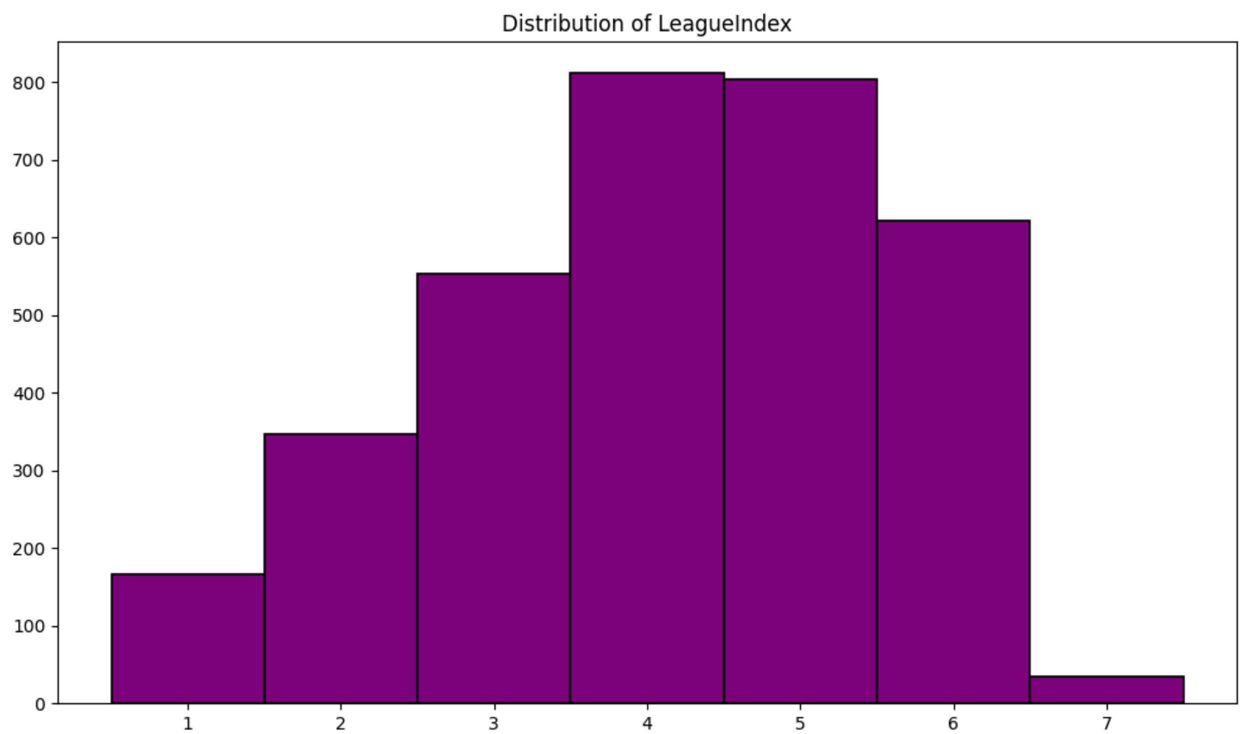


Appendix 4. **Histograms of Player Count, Tournament Count, and Total Tournament Prize Money for Super Smash Bros Melee**. The y-axis has been log-scaled for better visibility. Data from GGBeyond's Esports Earnings website.

Appendix 5. **SkillCraft1 Dataset Features and Explanations.** PAC is an abbreviation for the Perception-Action Cycle, which refers to a player switching screens for some time and performing at least one action. Explanations directly from the UC Irvine Machine Learning Repository.

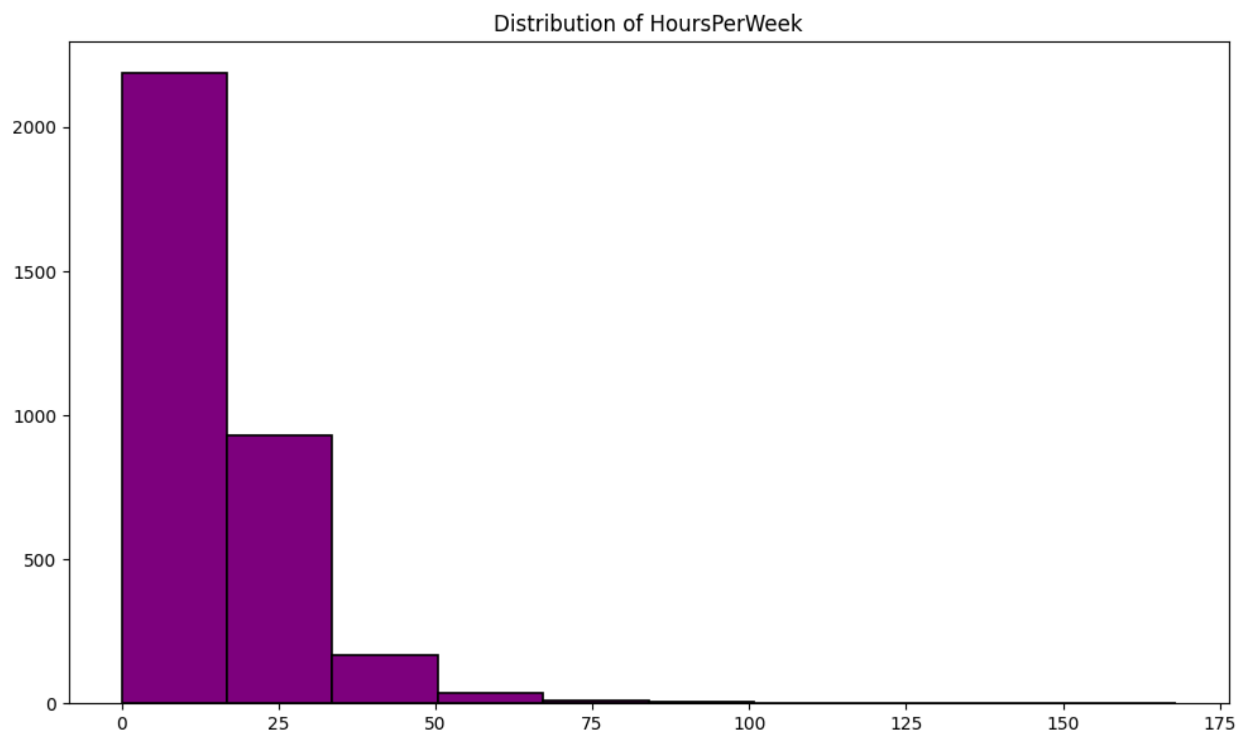
Feature	Explanation
GameID	Unique ID number for each game (integer)
LeagueIndex	Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional leagues coded 1-8 (Ordinal)
Age	Age of each player (integer)
HoursPerWeek	Reported hours spent playing per week (integer)
TotalHours	Reported total hours spent playing (integer)
APM	Action per minute (continuous)
SelectByHotkeys	Number of unit or building selections made using hotkeys per timestamp (continuous)
AssignToHotkeys	Number of units or buildings assigned to hotkeys per timestamp (continuous)
UniqueHotkeys	Number of unique hotkeys used per timestamp (continuous)
MinimapAttacks	Number of attack actions on minimap per timestamp (continuous)
MinimapRightClicks	Number of right-clicks on minimap per timestamp (continuous)
NumberOfPACs	Number of PACs per timestamp (continuous)
GapBetweenPACs	Mean duration in milliseconds between PACs (continuous)
ActionLatency	Mean latency from the onset of a PACs to their first action in milliseconds (continuous)
ActionsInPAC	Mean number of actions within each PAC

	(continuous)
TotalMapExplored	The number of 24x24 game coordinate grids viewed by the player per timestamp (continuous)
WorkersMade	Number of SCVs, drones, and probes trained per timestamp (continuous)
UniqueUnitsMade	Unique units made per timestamp (continuous)
ComplexUnitsMade	Number of ghosts, infestors, and high templars trained per timestamp (continuous)
ComplexAbilitiesUsed	Abilities requiring specific targeting instructions used per timestamp (continuous)

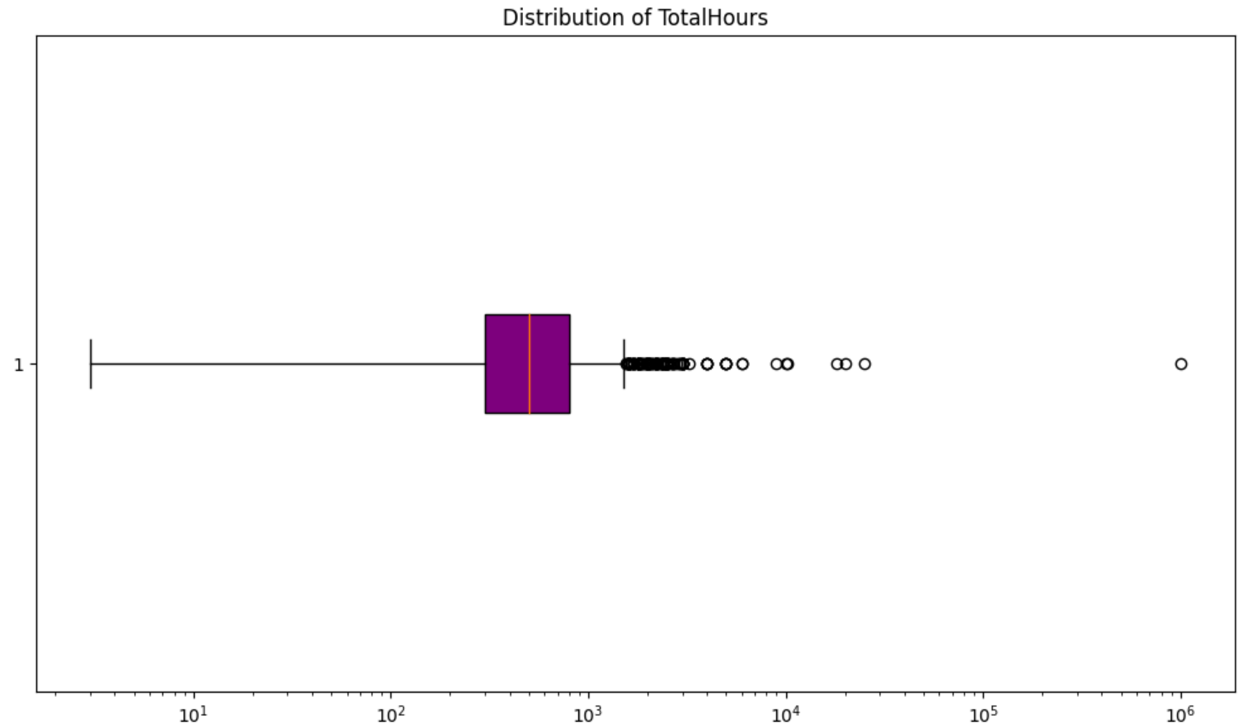


Appendix 6. **Histogram of League Index**. The numbers map to the specific ranks (i.e. 1: Bronze, 2: Silver, 3: Gold, 4: Platinum, 5: Diamond, 6: Master, 7: Grandmaster)

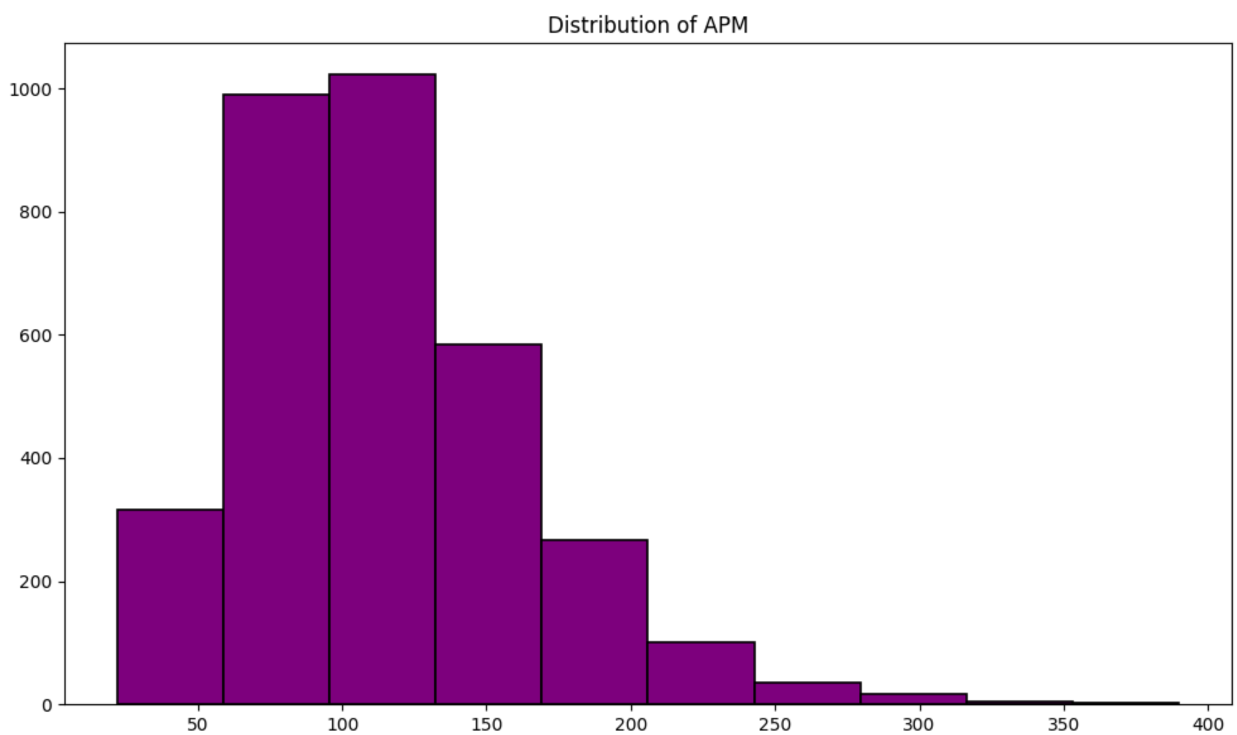
Appendix 7. **Histogram of Age**. The distribution of age is right-skewed.



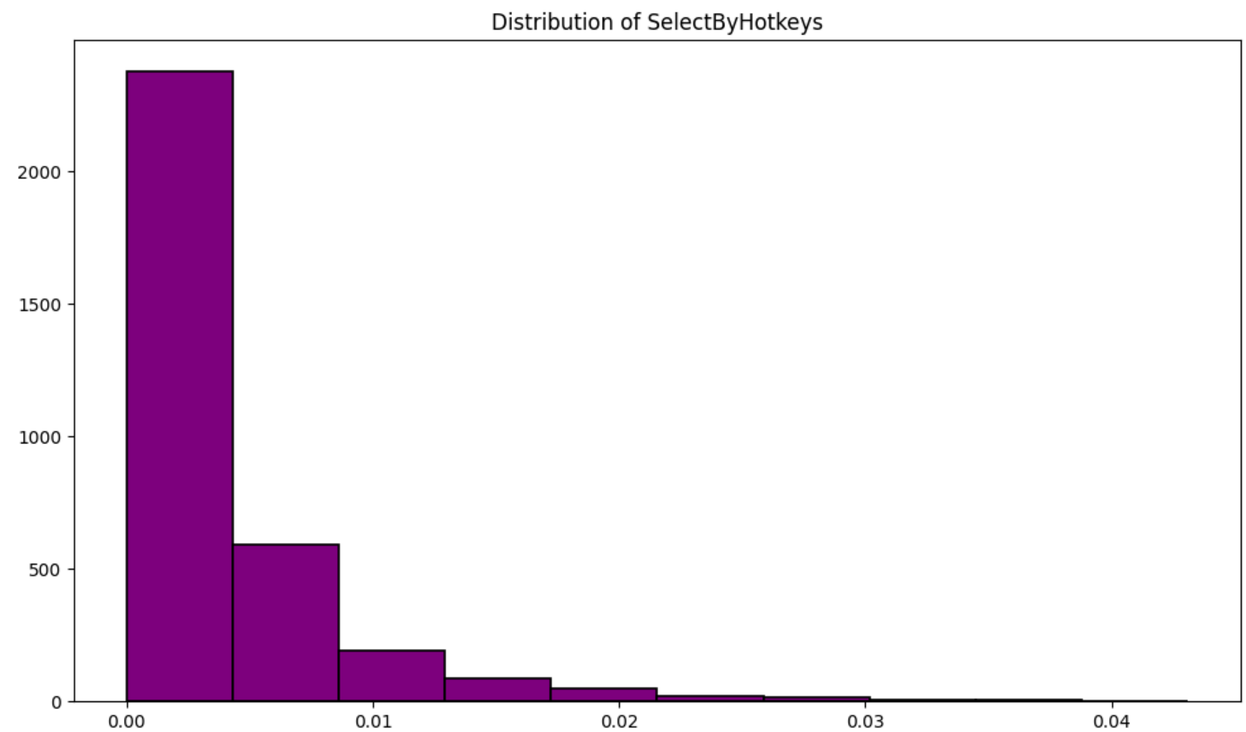
Appendix 8. **Histogram of Hours per Week**. The distribution of hours per week spent playing SC2 is right-skewed.



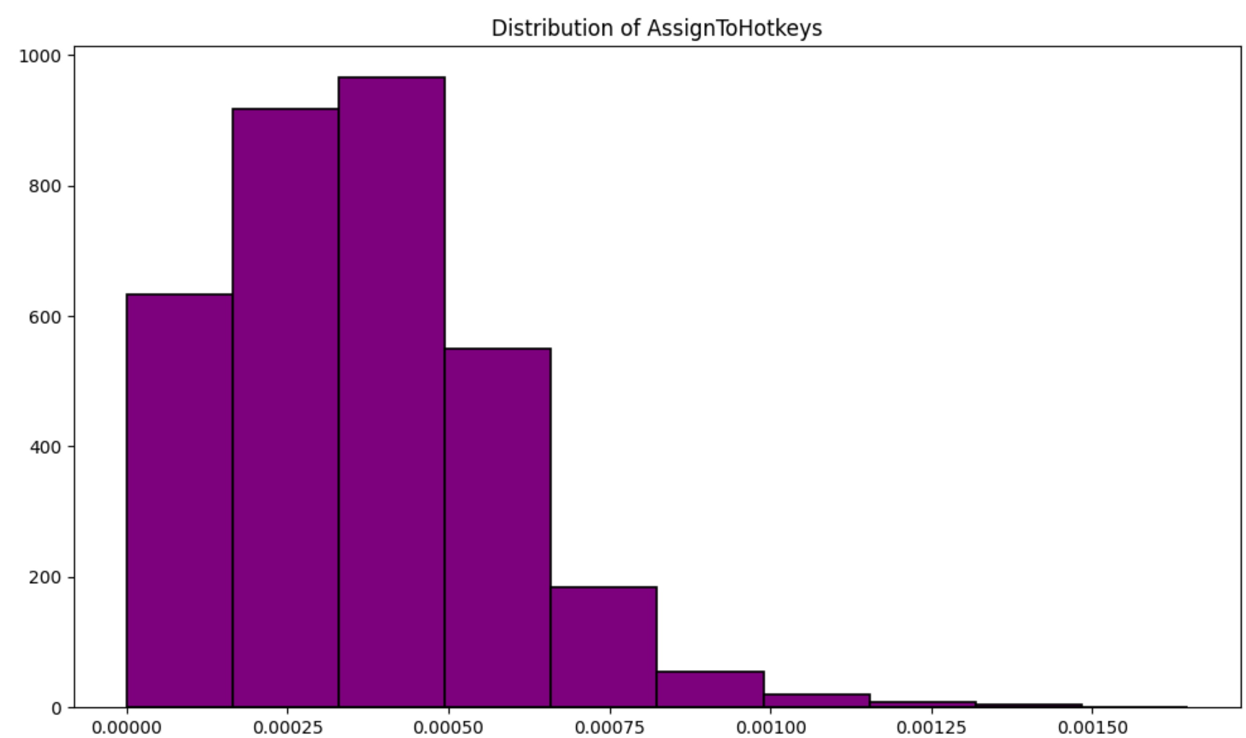
Appendix 9. **Boxplot of Total Hours**. Total Hours has a minimum value of three and a maximum of 100,000. Quartile 1 is 300 hours, the median is 500 hours, and quartile 3 is 800 hours. The average Total Hours is 960. The distribution of total hours spent playing SC2 is heavily right-skewed.



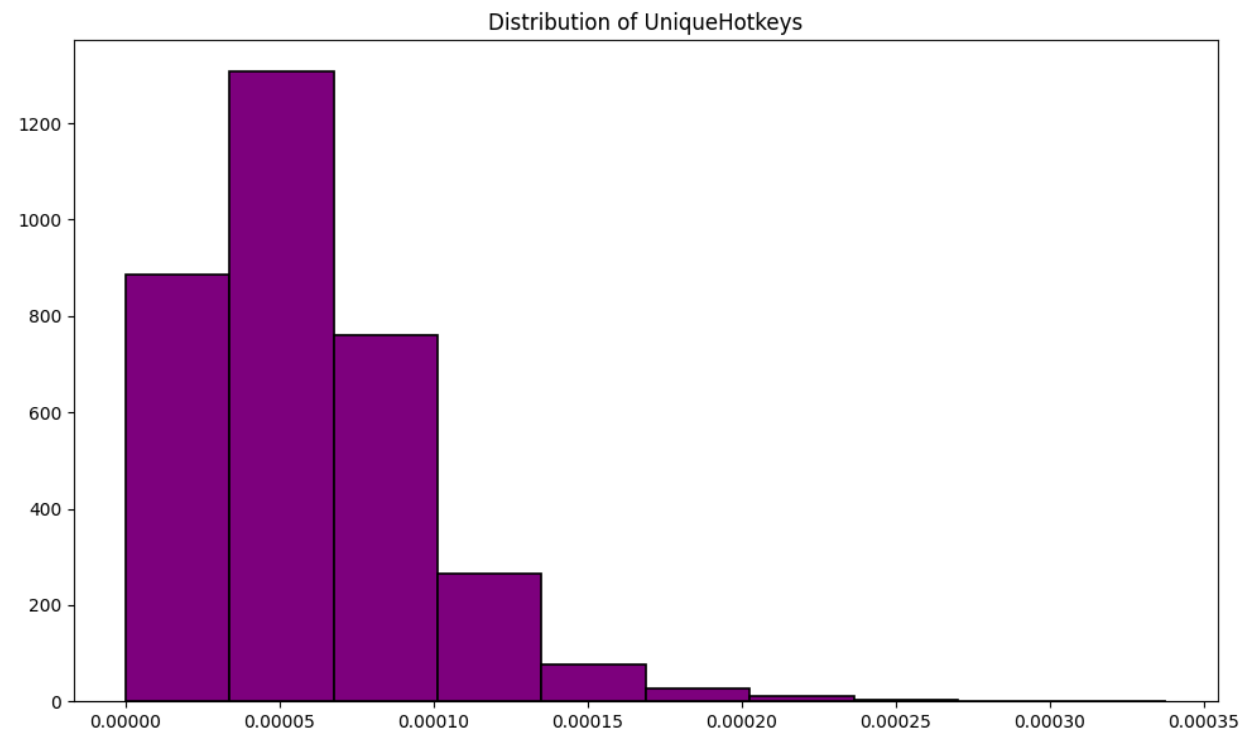
Appendix 10. **Histogram of APM**. The distribution of APM is right-skewed.



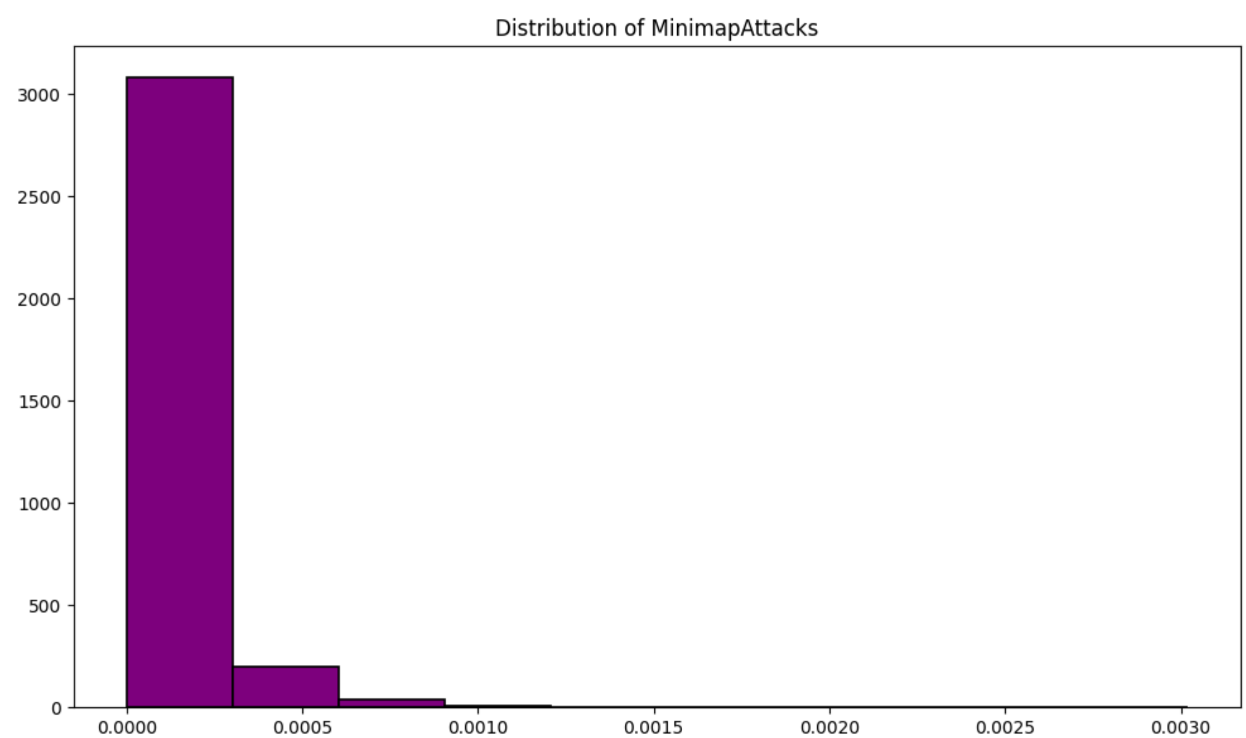
Appendix 11. **Histogram of SelectByHotkeys**. The distribution of the number of units and buildings selected with hotkeys within some timeframe is right-skewed.



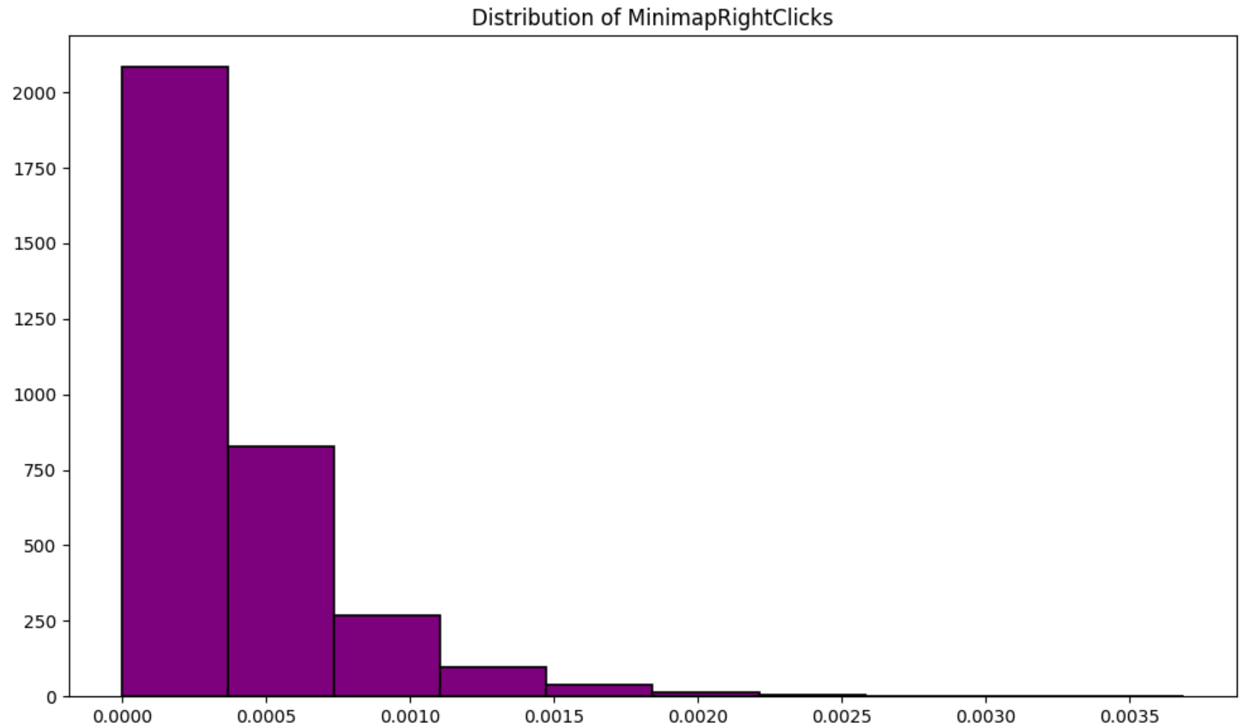
Appendix 12. **Histogram of AssignToHotkeys**. The distribution of the number of units and buildings assigned to hotkeys within some timeframe is right-skewed.



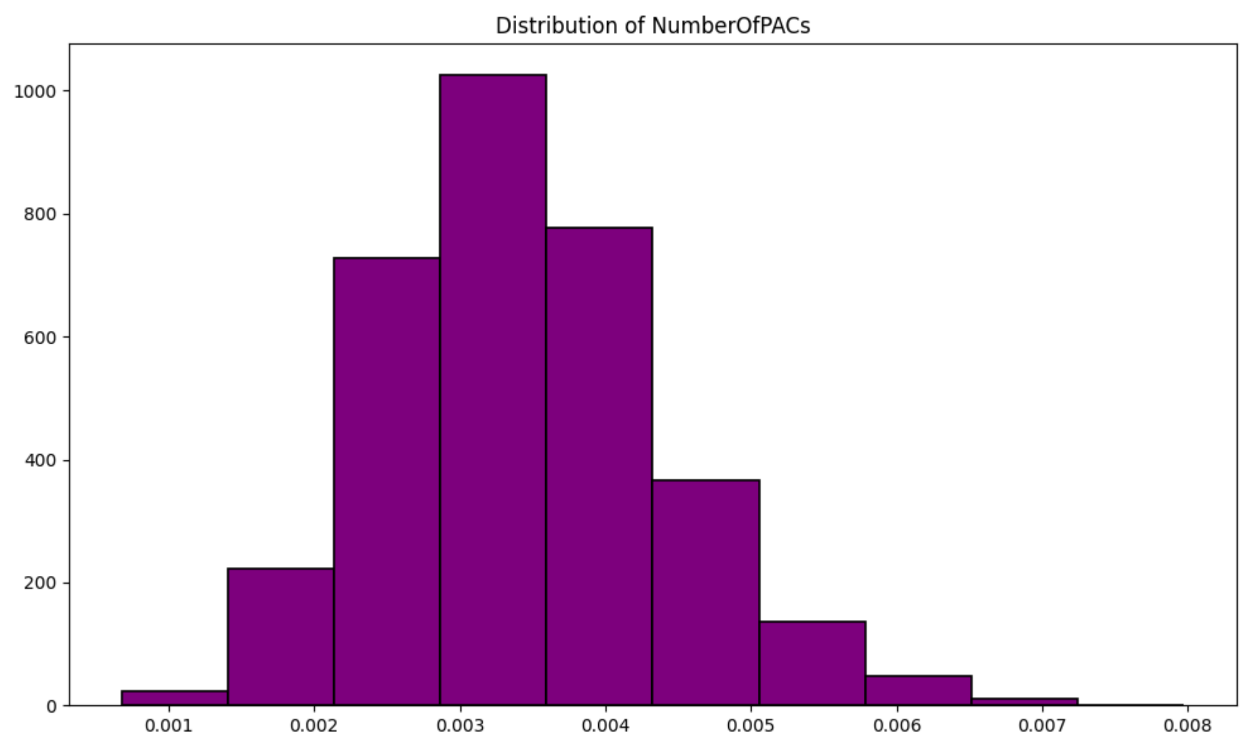
Appendix 13. **Histogram of UniqueHotkeys**. The distribution of the number of unique hotkeys used within some timeframe is right-skewed.



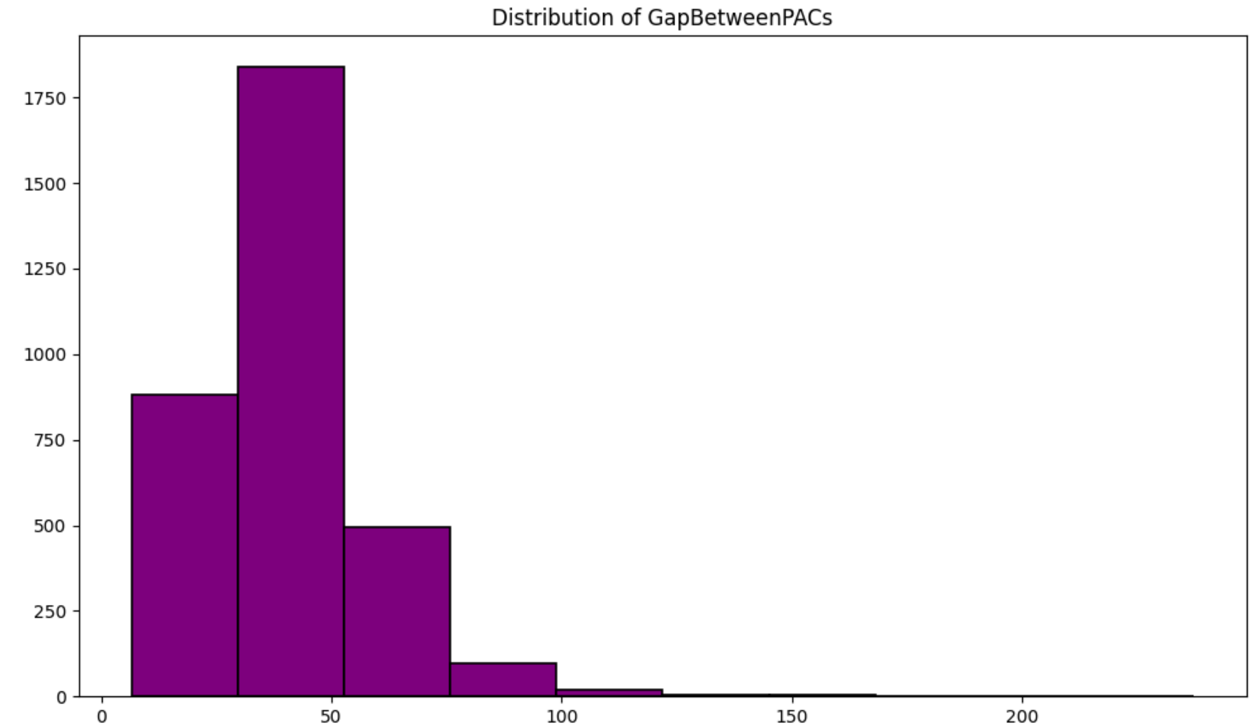
Appendix 14. **Histogram of MinimapAttacks**. The distribution of the number of attacks done by clicking on the minimap within some timeframe is right-skewed.



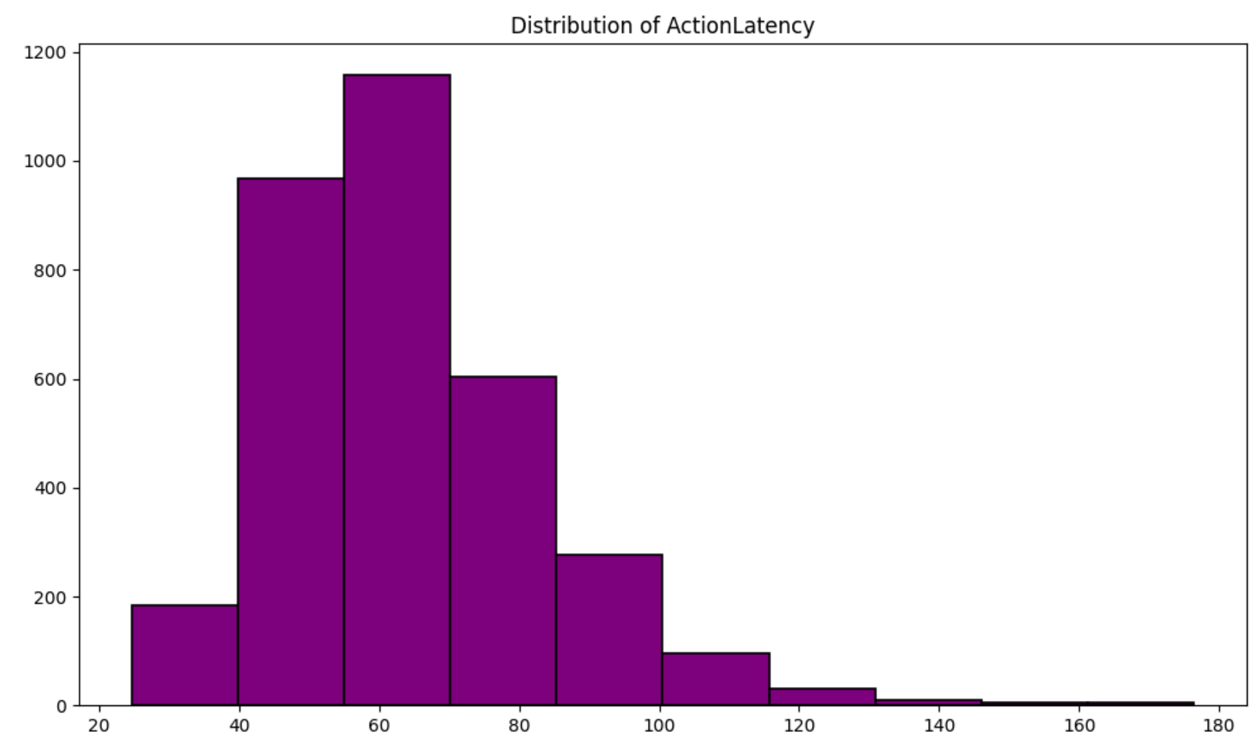
Appendix 15. **Histogram of MinimapRightClicks**. The distribution of the number of right clicks (i.e. movement commands) done by clicking on the minimap within some timeframe is right-skewed.



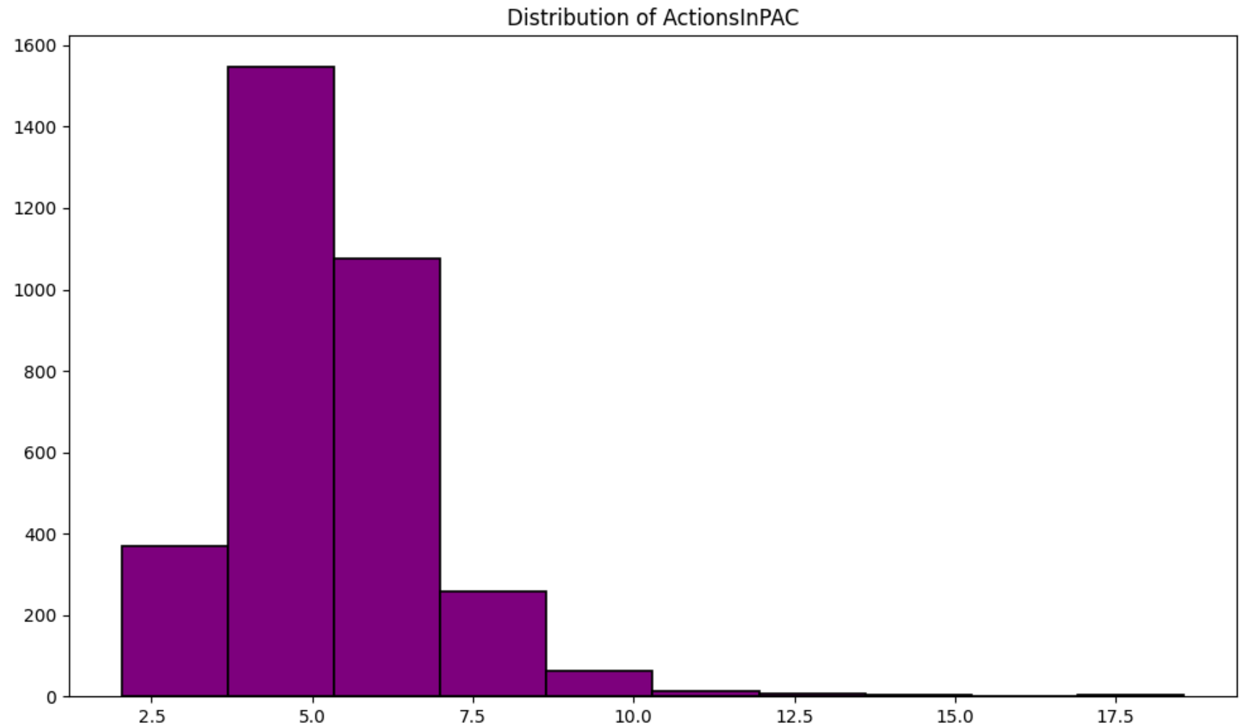
Appendix 16. **Histogram of NumberOfPACs**. The distribution of the number of perception-action cycles within some timeframe is approximately normal.



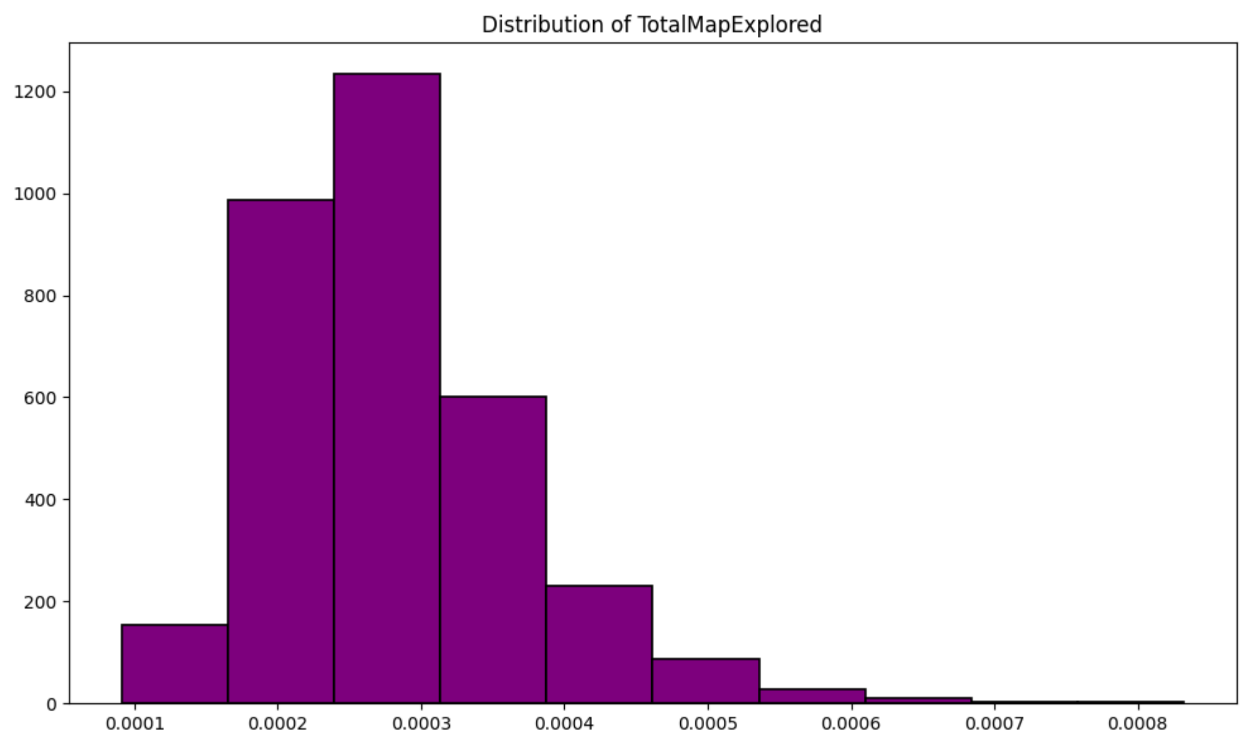
Appendix 17. **Histogram of GapBetweenPACs**. The distribution of the average delay (in milliseconds) between perception-action cycles within some timeframe is right-skewed.



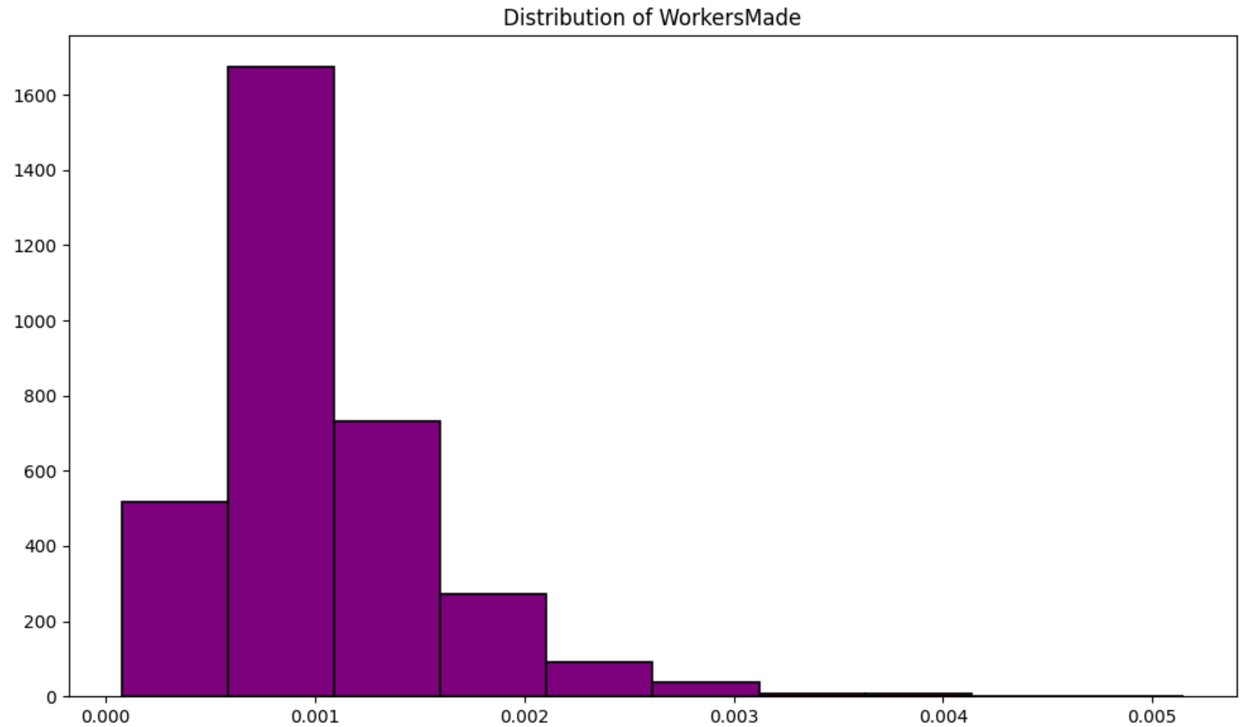
Appendix 18. **Histogram of ActionLatency**. The distribution of the average delay (in milliseconds) between actions within some timeframe is right-skewed.



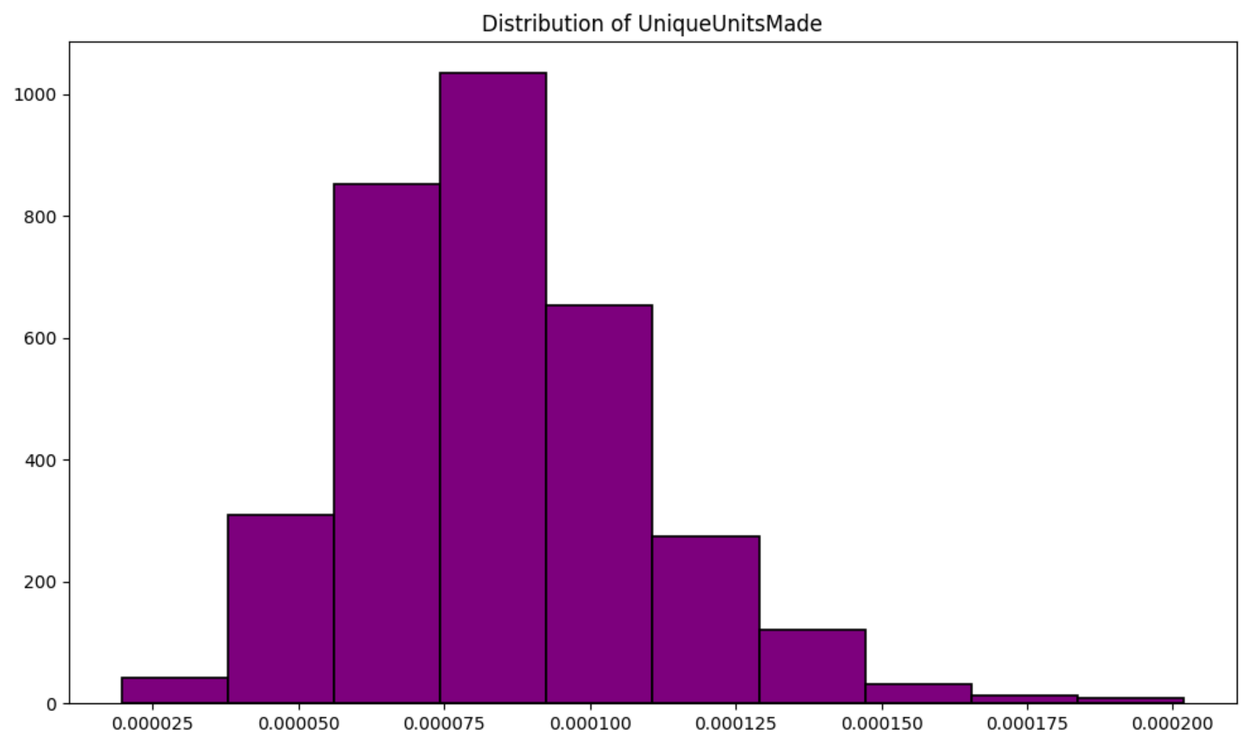
Appendix 19. **Histogram of ActionsInPAC**. The distribution of the average number of actions in each perception-action cycle within some timeframe is right-skewed.



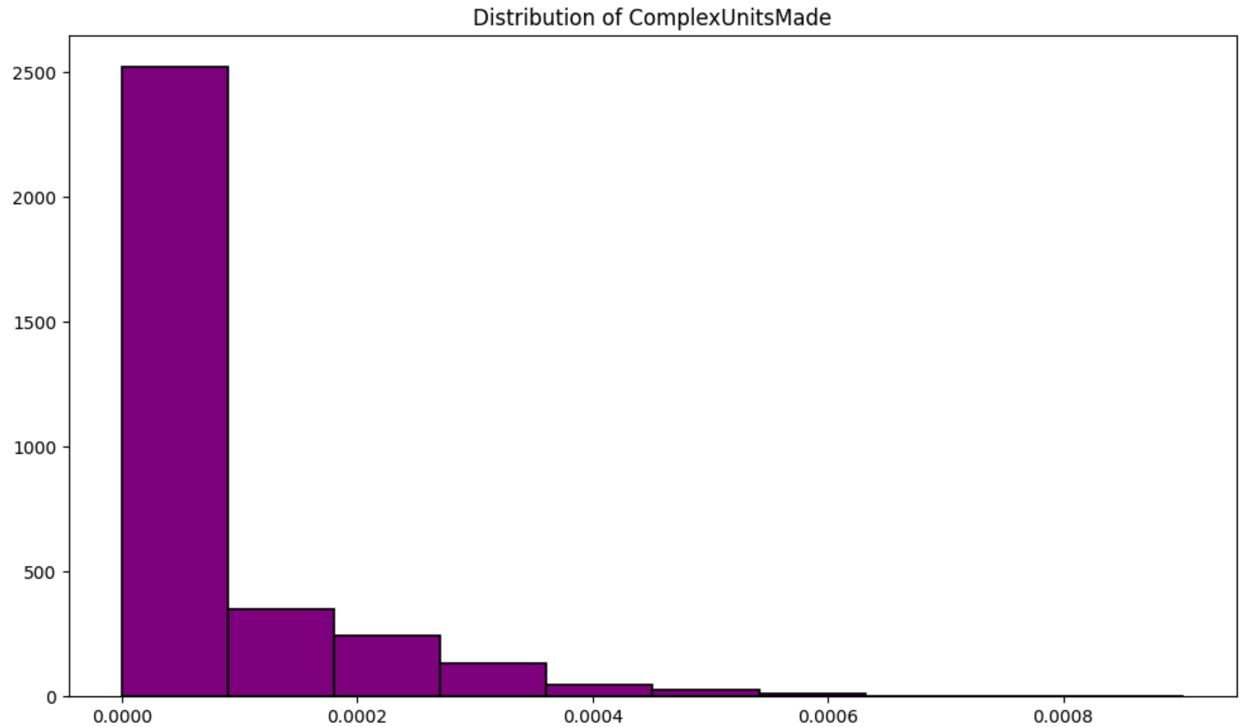
Appendix 20. **Histogram of TotalMapExplored**. The distribution of the amount of the map seen by a player within some timeframe is right-skewed.



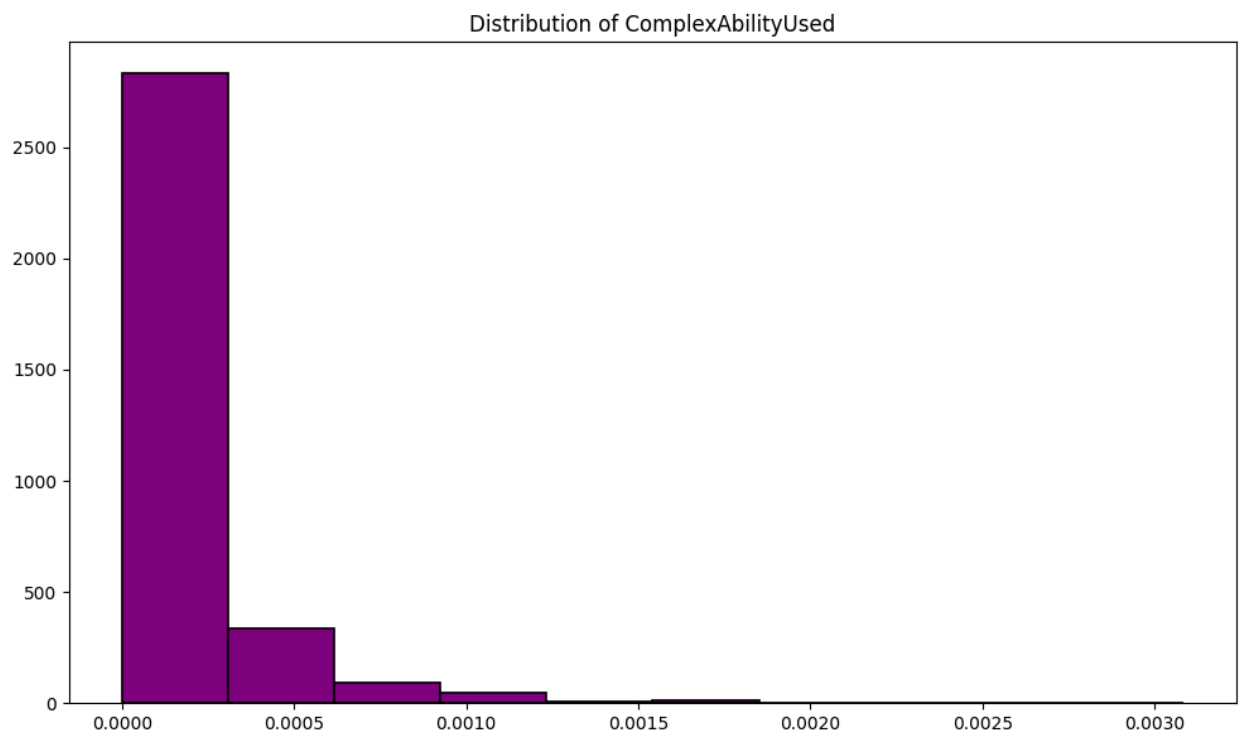
Appendix 21. **Histogram of WorkersMade**. The distribution of the number of workers (e.g. SCVs, drones, or probes) made within some timeframe is right-skewed.



Appendix 22. **Histogram of UniqueUnitsMade**. The distribution of the number of unique units made within some timeframe is approximately normal.



Appendix 23. **Histogram of ComplexUnitsMade**. The distribution of the number of complex units (e.g. ghosts, infestors, or high templars) made within some timeframe is right-skewed.



Appendix 24. **Histogram of ComplexAbilitiesUsed**. The distribution of the number of target-specific abilities (e.g. snipe, fungal growth, or feedback) used within some timeframe is right-skewed.

Appendix 25. **Table of Hyperparameters and Metrics for Logistic Regression**. The worst model is model 4.

Model	Condition	Is Balanced	Accuracy	ROC AUC Score
LR1	Baseline: C=1.0, L2 regularization, SAGA solver	No	0.257	0.536
LR2	Tuned for Accuracy: C=0.01, L2 regularization, SAGA solver	No	0.248	0.564
LR3	Tuned for ROC AUC: C=1.0, Elastic net regularization, SAGA solver	No	0.248	0.564
LR4	Baseline: C=1.0, L2 regularization, SAGA solver	Yes	0.012	0.619
LR5	Tuned for Accuracy: C=0.01, L2 regularization, SAGA solver	Yes	0.248	0.564
LR6	Tuned for ROC AUC: C=1.0, Elastic net regularization, SAGA solver	Yes	0.248	0.564

Appendix 26. **Table of Hyperparameters and Metrics for SVMs**. The best-performing model is model 6.

Model	Condition	Is Balanced	Accuracy	ROC AUC Score
SVM1	Baseline: C=1.0, RBF kernel	No	0.138	0.505
SVM2	Tuned for Accuracy: C=0.1, RBF kernel	No	0.174	0/486
SVM3	Tuned for ROC AUC:	No	0.032	0.470

	C=0.1, Polynomial kernel (degree=2)			
SVM4	Baseline: C=1.0, RBF kernel	Yes	0.112	0.506
SVM5	Tuned for Accuracy: C=1.0, RBF kernel	Yes	0.216	0.569
SVM6	Tuned for ROC AUC: C=0.1, Polynomial kernel (degree=2)	Yes	0.240	0.639

Appendix 27. **Table of Hyperparameters and Metrics for Random Forests**. The best-performing model is model 4

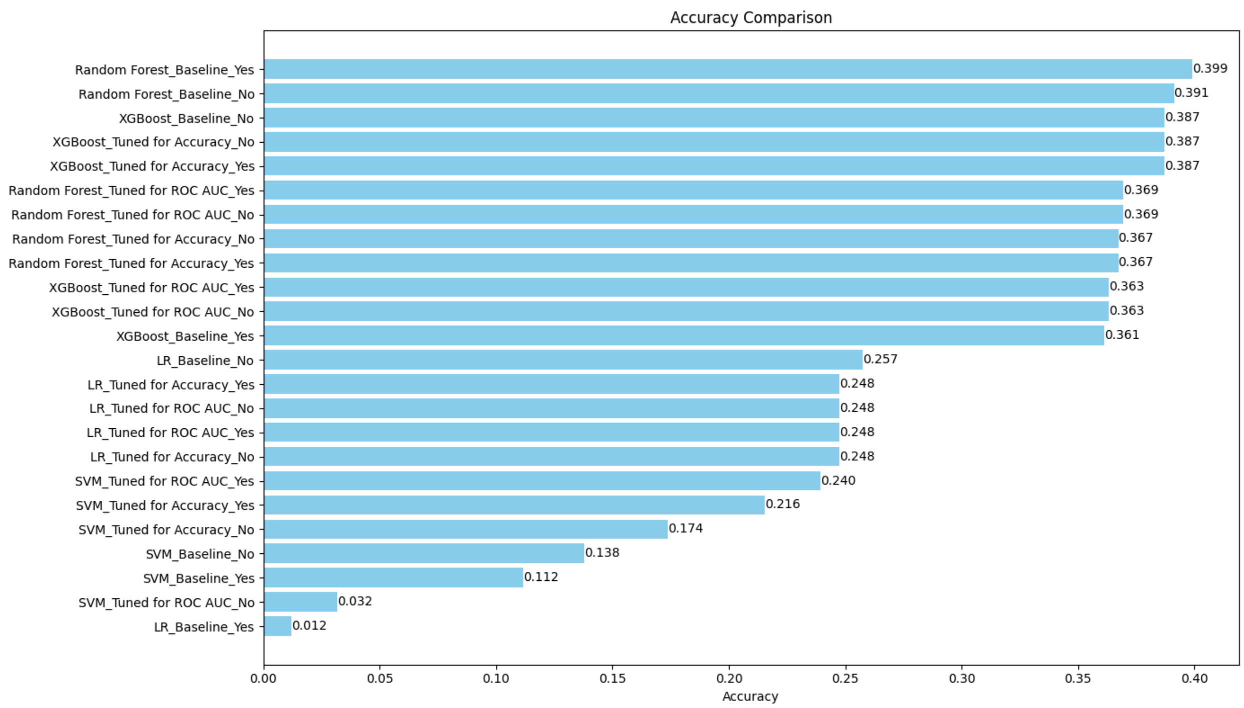
Model	Condition	Is Balanced	Accuracy	ROC AUC Score
RF1	Baseline: max_depth=None, n_estimators=100, min_samples_split=2, min_samples_leaf=1	No	0.391	0.801
RF2	Tuned for Accuracy: max_depth=None, n_estimators=100, min_samples_split=10, min_samples_leaf=2	No	0.367	0.801
RF3	Tuned for ROC AUC: max_depth=10, n_estimators=300, min_samples_split=10, min_samples_leaf=2	No	0.370	0.801
RF4	Baseline: max_depth=None, n_estimators=100, min_samples_split=2, min_samples_leaf=1	Yes	0.399	0.802

RF5	Tuned for Accuracy: max_depth=None, n_estimators=100, min_samples_split=10, min_samples_leaf=2	Yes	0.367	0.802
RF6	Tuned for ROC AUC: max_depth=10, n_estimators=300, min_samples_split=10, min_samples_leaf=2	Yes	0.370	0.802

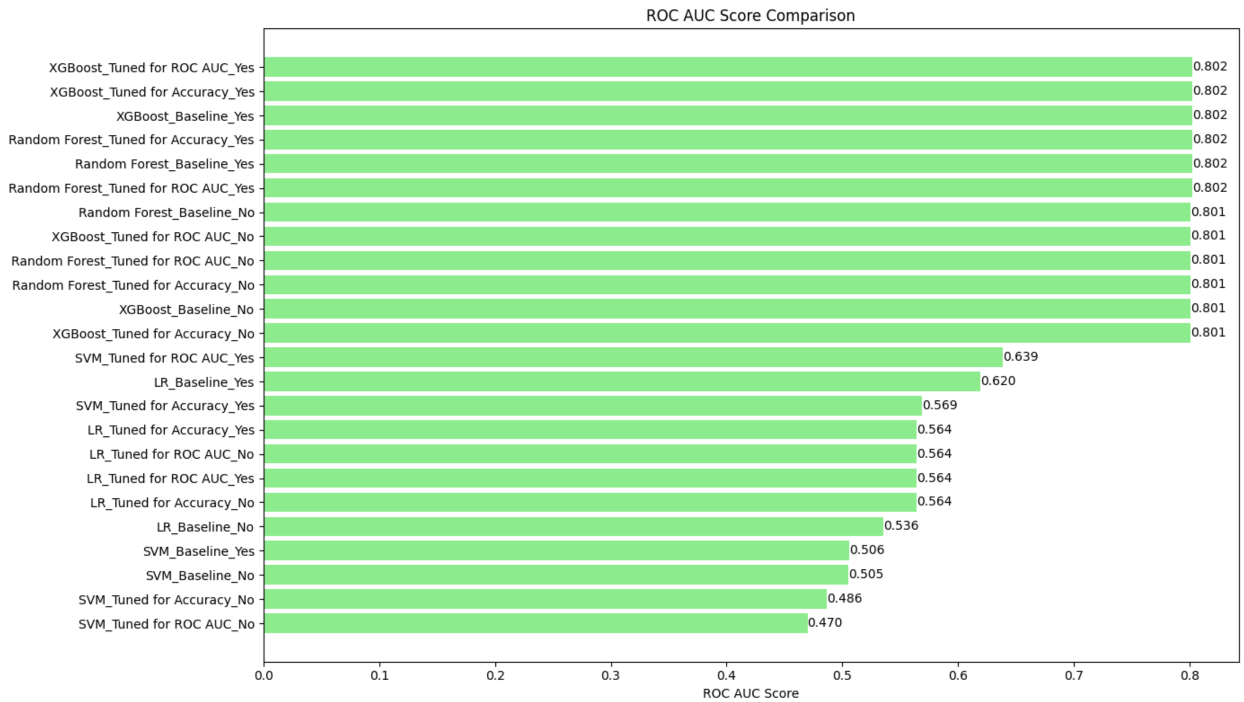
Appendix 28. **Table of Hyperparameters and Metrics for XGBoost**. The best-performing model is model 5.

Model	Condition	Is Balanced	Accuracy	ROC AUC Score
XGB1	Baseline: gamma=0, learning_rate=0.3, max_depth=6, min_child_weight=1, n_estimators=100	No	0.387	0.801
XGB2	Tuned for Accuracy: gamma=1.5, learning_rate=0.01, max_depth=3, min_child_weight=1, n_estimators=100	No	0.387	0.801
XGB3	Tuned for ROC AUC: gamma=1.5, learning_rate=0.1, max_depth=3, min_child_weight=1, n_estimators=300	No	0.363	0.801
XGB4	Baseline: gamma=0, learning_rate=0.3, max_depth=6, min_child_weight=1, n_estimators=100	Yes	0.361	0.802
XGB5	Tuned for Accuracy: gamma=1.5,	Yes	0.387	0.802

	learning_rate=0.01, max_depth=3, min_child_weight=1, n_estimators=100			
XGB6	Tuned for ROC AUC: gamma=1.5, learning_rate=0.1, max_depth=3, min_child_weight=1, n_estimators=300	Yes	0.363	0.802



Appendix 29. **Histogram of Accuracies for All Models Tested**. The baseline Random Forest model with SMOTE performs the best with an accuracy of 0.399. The baseline Logistic Regression model with SMOTE performs the worst with an accuracy of 0.012.



Appendix 30. **Histogram of ROC AUC Scores for All Models Tested**. The greatest ROC AUC score for several models is about 80%, and the worst is 47%.

References

"History of Esports." CDW.Com, CDW, 10 Sept. 2021, www.cdw.com/content/cdw/en/articles/hardware/history-of-esports.html.

Larch, Author: Florian, and Florian Larch. "Emergence of Esports: Once Ridiculed, Now a Billion-Dollar Market." ISPO.Com, 24 Oct. 2023, www.ispo.com/en/sports-business/esports-history-how-it-all-began.

Thompson, Joseph J., et al. "Video game telemetry as a critical tool in the study of Complex Skill Learning." PLoS ONE, vol. 8, no. 9, 18 Sept. 2013, https://doi.org/10.1371/journal.pone.0075129.