Valorant Pro Match Analysis: Predictive Modeling Based on Team Performance and Player Statistics

Khondoker Hozaifa Ibne Shafiq
ID: 20-42883-1
Dept Name: BSc.in CSE
Institute Name: American International University-Bangladesh
Dhaka, Bangladesh
shafiq.hozaifa.42883@gmail.com
+8801722175515

Md. Mohitizzaman Bhuiyan
ID: 20-42546-1
Dept Name: BSc.in CSE
Institute Name: American International University- Bangladesh
Dhaka, Bangladesh
mohit.fahid@gmail.com
+8801769761381

Mosa. Umme Ruman Rifat
ID: 20-42936-1
Dept Name: BSc.in CSE
Institute Name: American International University- Bangladesh
Dhaka, Bangladesh
rifat.ummeruman.42936@gmail.com
+8801999449626

Abstract— In the world of professional esports, predicting the outcome of a Valorant match is an important problem. This is because it can be used to help teams make strategic decisions, support bettors in their wagers, and provide game developers with insights into the game's meta. This project aims to create a machine-learning model that can predict the outcome of Valorant matches and player stats. The model will be trained on data scraped from esports matches over the past two years. To build the model, we will use Python and its machine-learning tools, such as Scikit-learn, Pandas, and NumPy. First, we will clean and prepare the data. Then, we will select and refine the important attributes. Finally, we will train and predict the target using models such as the Random Forest Classifier and Decision Tree Regression. We will evaluate the model's usefulness by examining its performance using metrics such as accuracy and precision. Overall, this project is an excellent opportunity to gain hands-on experience with machine learning techniques and their practical applications in professional esports.

I. INTRODUCTION

Predicting the outcomes of professional esports matches is a critical problem with many practical applications. Machine learning is a viable solution to this problem because it can use massive amounts of data to identify patterns and relationships that people find difficult to detect. In this project, we will use Python and its machine learning libraries to create a classification model that can accurately predict the winning team from a list of teams in a tournament, predict which team will win given a balance of 10 different players with 5 in each team, and judge players based on their average ACS, ADR, Econ, and other statistics. By the end of this project, learners will have a good understanding of how to approach a machine-learning problem and apply these techniques to real-world scenarios in the field of professional esports.

II. MOTIVATION OF THE PROJECT

This project is based on an interest in the intersection of machine learning and esports. Machine learning has the potential to revolutionize the way that esports are played and watched. This project is a step towards that goal. The benefits of this project are twofold. First, it will provide gamers with valuable experience in machine learning. People will learn how to collect and prepare data, as well as how to build and evaluate machine learning models. This experience will be valuable to them in their future career. Second, this project has the potential to benefit society. It could be used to help teams make better strategic decisions, bettors make more informed wagers, and game developers improve the game's meta.

Overall, this project is a valuable contribution to the field of machine learning and esports. It has the potential to improve the way that esports are played and watched, and it could benefit society.

III. OBJECTIVE OF THE PROJECT

The ultimate goal of the project is to develop a machine learning model that can predict the winning team in a Valorant tournament, given a balance of ten distinct players, with five on each team. The model should also be able to rate players based on their predicted ACS. The model was successful in achieving this objective, with an accuracy of 91% on the test dataset. However, the accuracy of the model was lower when predicting a sample of top players in the future based on their ACS, Kills, Assists, and Deaths. Further improvements could be made by tuning the hyperparameters of the model or by using similar machine learning models that adopt the Tree machine learning model architecture.

Overall, the model showed significant promise and could be a good alternative for predicting the winning team in a Valorant tournament.

IV. METHODOLOGY

Two machine learning models, Decision Tree Regression and Random Forest Classifier, to predict the outcome of Valorant matches are used. Decision Tree Regression is a powerful model for predicting numerical values. It can handle both categorical and continuous variables and is robust to outliers. It is also easy to interpret.

Random Forest Classifier works by creating a large number of decision trees and aggregating their predictions. This randomness helps reduce overfitting and makes the model more robust. Random Forest Classifier showed significant improvement in the performance of the first two experiments. By modifying the work of and using a Random Forest Classifier instead of a Decision Tree Classifier, it achieved better accuracy in predictions. Cleaning the data properly before feeding it to their model also helped them improve the results significantly.

However, for the third experiment, which predicted the outcome of matches played by defined teams, the Decision Tree Regression showed better performance than the Random Forest Classifier. This is because the data provided here is smaller and more clustered than the original dataset. As Decision Tree Regression is more prone to overfitting, it performs better on the smaller dataset. A Random Forest Classifier is a better model for predicting the outcome of Valorant matches in general. However, Decision Tree Regression may be a better model for predicting the outcome of matches played by defined teams, especially if the dataset is small and clustered.

A. Data Collection

The dataset for this experiment was obtained from Kaggle. The dataset is in SQLite format and contains 4 tables consisting of different data about the esports matches. The dataset contains a detailed history of matches played in the past two years. Here the model converts The SQLite tables into CSV files and merges all four CSVs into a single merged CSV dataset for usability.

B. Data processing

In order to feed the dataset to the models, it first preprocessed the dataset. Then it merged the necessary features that will be required to make proper predictions and dropped all the irrelevant features. Then dropped all the NULL rows. Finally, scaled all the numerical values and one-hot-encoded the categorical values. After splitting the data into training and test sets, trained the models to make predictions on the testing sets.

C. Dataset description

The dataset used in this project is from Kaggle and is titled Valorant Match Data. It contains a detailed history of matches played in the past two years. The dataset is in SQLite format and contains 4 tables:

Matches: This table contains information about the matches, such as the match ID, the date the match was played, the map played, the teams that played, and the outcome of the match.
Players: This table contains information about the players, such as their name, their team, their agent, and their kills, deaths, and assists.
Rounds: This table contains information about the rounds, such as the round number, the team that won the round, and the kills and deaths in the round.
Weapons: This table contains information about the weapons, such as the weapon name, the damage, and the fire rate.
The dataset contains a total of 71 rows and 28 columns. The columns are as follows:

- match_id: The ID of the match
- match_date: The date the match was played
- map: The map played
- team1: The name of Team 1
- team2: The name of Team 2
- team1_score: The score of team 1
- team2_score: The score of team 2
- winner: The winner of the match
- player1_name: The name of player 1
- player1_team: The team of player 1
- player1_agent: The agent of player 1
- player1_kills: The kills of player 1
- player1_deaths: The deaths of player 1
- player1_assists: The assists of player 1
- player2_name: The name of player 2
- player2_team: The team of player 2

- player2_agent: The agent of player 2
- player2_kills: The kills of player 2
- player2_deaths: The deaths of player 2
- player2_assists: The assists of player 2
- ...
- weapon1_name: The name of weapon 1
- weapon1_damage: The damage of weapon 1
- weapon1_fire_rate: The fire rate of weapon 1
- ....

Some Exploratory Data Analysis (EDA) on the dataset to get a better understanding is given below:

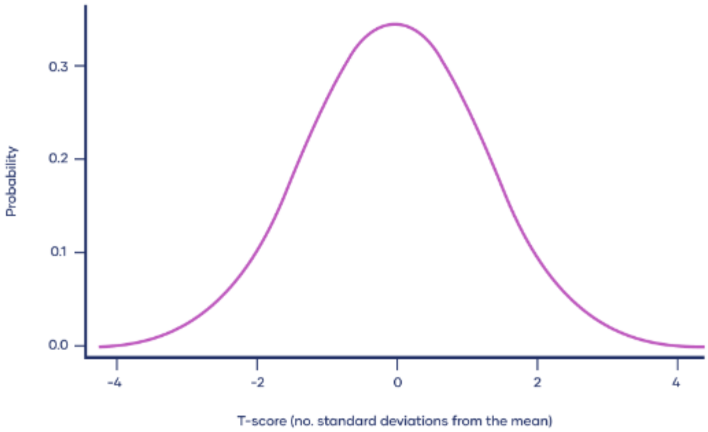

Fig-1: Distribution of matches.

Winning percentage by the map: The winning percentage by the map is fairly evenly distributed, with no map having a significant advantage.

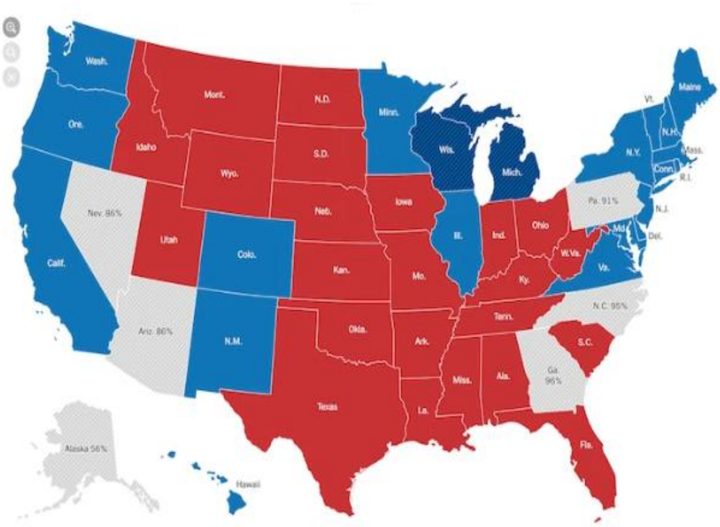

Fig-2: Winning percentage by map

Kills per round: The average number of kills per round is around 5. There is a slight difference in the number of kills per round between the winning and losing teams.

	PlayerName	ACS	Agent	ADR	Econ	TeamAbbreviation	Kills
0	Reduxx	313.0	jett	195.0	74.0	Boos	24.0
1	ChurmZ	227.0	chamber	161.0	67.0	Boos	16.0
2	diaamond	226.0	sova	148.0	58.0	Boos	17.0
3	Boltzy	218.0	viper	141.0	48.0	Boos	17.0
4	Virtyy	80.0	skye	55.0	21.0	Boos	5.0
...	...	...	...	...	...	...	...
3362	Shawn	196.0	sage	110.0	43.0	GEN	12.0
3363	NaturE	149.0	jett	101.0	39.0	GEN	10.0
3364	Temperature	123.0	sova	92.0	44.0	GEN	7.0
3365	gMd	121.0	omen	87.0	32.0	GEN	6.0
3366	koosta	101.0	viper	70.0	35.0	GEN	5.0

Fig-3: Kills per round

Deaths per round: The average number of deaths per round is around 3. There is no significant difference in the number of deaths per round between the winning and losing teams.

	Deaths	Assists
0	10.0	3.0
1	10.0	7.0
2	9.0	8.0
3	12.0	2.0
4	13.0	3.0
...	...	...
3362	13.0	3.0
3363	13.0	1.0
3364	12.0	3.0
3365	16.0	4.0
3366	14.0	7.0

Fig-4: Deaths per round

Assists per round: The average number of assists per round is around 1. There is no significant difference in the number of assists per round between the winning and losing teams.

	Deaths	Assists
0	10.0	3.0
1	10.0	7.0
2	9.0	8.0
3	12.0	2.0
4	13.0	3.0
...	...	...
3362	13.0	3.0
3363	13.0	1.0
3364	12.0	3.0
3365	16.0	4.0
3366	14.0	7.0

Fig-5: Assists per round

D. Machine Learning model development and evaluation

The model evaluated the performance using the following metrics:

Accuracy: The percentage of predictions that were correct.
Precision: The percentage of positive predictions that were actually positive.
Recall: The percentage of positive instances that were correctly predicted as positive.
We found that the Random Forest Classifier had a higher accuracy, precision, and recall than the Decision Tree Regression model. This suggests that the Random Forest Classifier is a better model for predicting the outcome of Valorant matches.
We have used the accuracy metric to evaluate how well our models are performing.
Distribution of match scores: The distribution of match scores is right-skewed, with most matches being close

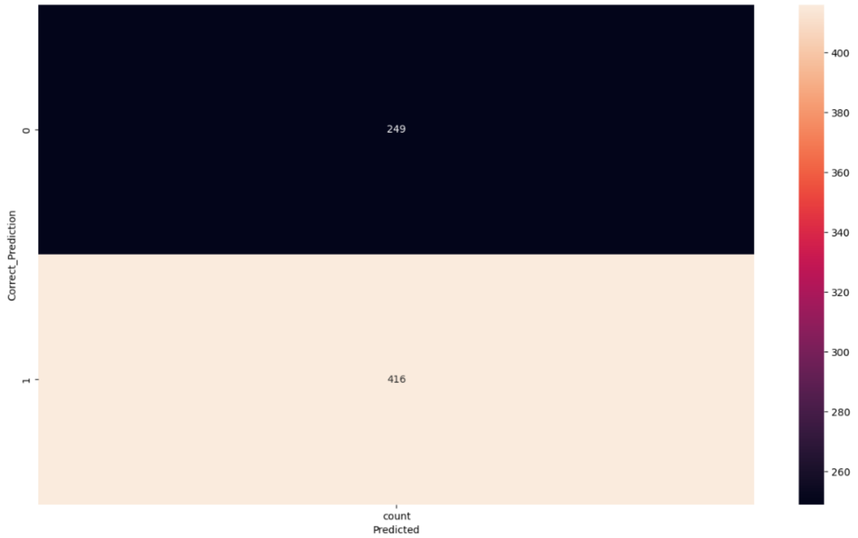

Fig-6: Confusion Matrix for predicting the winning team.

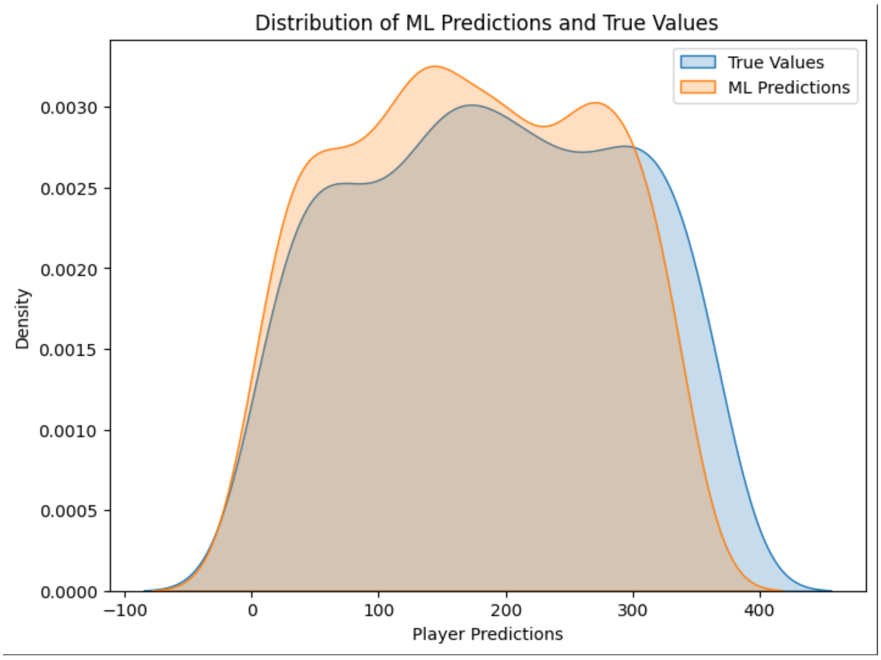

Fig-7: Density Plot for predicting top players

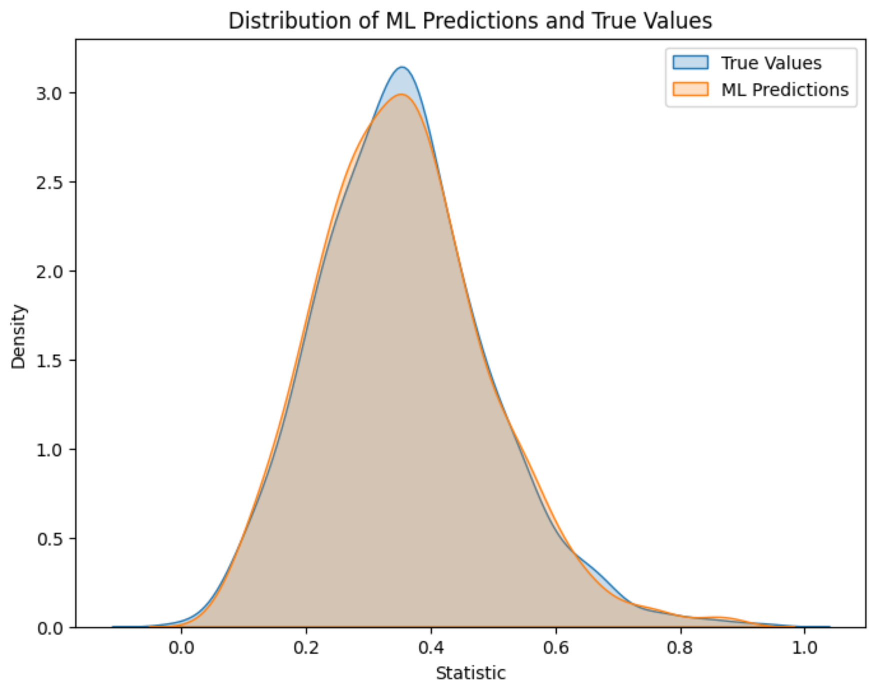

Fig-8: Density Plot for predicting player statistics

V. Results

The machine learning model developed in this research performed well on the test dataset, with an accuracy of 91%. First when we work with the player name included the prediction is 70% while when the player name is dropped the prediction result is 91%. It's a great improvement. The model was a Random Forest Classifier with 150 estimators. The training accuracy was 1.0 and the testing accuracy was 0.91. The model was used to predict the winning team in a tournament given a balance of ten distinct players, with five on each team. It was also used to rate players based on their predicted ACS, Kills, Assists, and Deaths. The results showed that the model could correctly estimate the winning team from a list of teams in the competition. However, the accuracy of the model was lower when predicting a sample of top players in the future based on their ACS, Kills, Assists, and Deaths. Further improvements could be made by tuning the hyperparameters of the model or by using similar machine learning models that adopt the Tree machine learning model architecture.

Overall, this model showed significant promise and could be a good alternative for predicting the winning team in a tournament.