A Machine Learning System for Predicting Team Win Rates, Character Builds, and Individual Performance in Eternal Return

Zyrus Crispino

College of Engineering, Architecture, and Mathematics

*De La Salle University - Dasmarinas* Dasmarinas City, Philippines [crispino.zyrus@gmail.com](mailto:crispino.zyrus@gmail.com)

***Abstract*—This paper presents a Python-based Multi-Model machine learning system specifically for the Metrics of Team and Character Performance and Prediction in *Eternal Return*. The tool employs probability, pattern ranking, and common trends when it comes to common teams and character builds that consistently rank high in every match. Additionally, it accounts for a certain team or character’s additional metrics such as team performance like team kills, kills, deaths, assists, as well as the total damage to other players and damage to other monsters, otherwise known in other games as creeps.**

**A distinguishing feature of the system is its capability to parse user-inputted data through the three models and output the according predictions. In one model, the user will input three names of *Eternal Return* characters to form a team, and the model will output the predicted win rate of the team. In another model, the user will simply input the name of an *Eternal Return* character and receive information on their top 3 builds. And for the last model, the user will input the name of a single character from *Eternal Return* and receive that character’s individual predicted win rate, as well as show the historical win rate of the character.**

**The primary objective of this system is to demonstrate the capabilities of Machine Learning algorithms in providing reliable statistics regarding Team Placement, Character Builds, and individual Character Winrate that can serve as the foundation for future work regarding similar subjects.**

***Index Terms*—Machine Learning, Eternal Return, Team Per- formance Prediction, Character Build Prediction, Win Rate Prediction, Independent Models.**

1. Introduction

The use of Machine Learning when it comes games has been on the rise since it’s conception. Particularly, the use of machine learning and artificial intelligence had existed way back when. It is only now that machine learning is being used due to the booming technology that it is now today. Similarly, machine learning has been stated to have use in the domain of game development[1]. Projects that use video game data or analytics and use machine learning to process the said information are now abundant and come in different yet similar structures. All process data, and all predict to varying degrees of success. Said data more often than not seems to be regarding the game’s own definition of their statistics. This is possibly

because statistics are how most gameplay mechanisms func- tion as most of it stems from statistical elements to create creative gameplay[2]. One genre of games that make use of such elements are Multiple Online Battle Area games, or better know and simply put as MOBAs, and battle royales. MOBA games are games where players are commonly grouped in teams in an online battle area controlling fighting units, most commonly from a top-down view of varying angles [3]. Teams go toe-to-toe in a fight for objectives that differ between video games, but are mostly the same. There are objectives that can give the players who acquire it massive boons or advantages over their opposition, but not enough to completely overpower them. While for Battle Royales, it is a different story. Players can be grouped, or not at all. Once in game, it’s an absolute free for all where everyone is playing for survival, and the only winner is the last one standing in a map continuously shrinking for some reason or another[4]. Eternal Return, meanwhile, is a mix of both. It features the qualities of a MOBA in terms of character-based combat, distinct abilities, item builds that can change a character’s playstyle as the user sees fit, as well as strategic gameplay in teams of three. But it also incorporates battle royal elements such as eight teams of three per game, all fighting for survival on the area that Eternal Return calls Lumina Island. There are currently 80 characters in the game, and that number will only continue to increase as time goes on. Multiply the 80 characters in the game, to the many number of items each character can craft and equip that will influence their strengths and weaknesses per game, the amount of choices in the game are staggering, and even become cause for decision paralysis. Luckily, there is one thing Eternal Return has in spades, and that is data. Particularly, match data, such as what characters won, which items did they use, how well did they perform, and practically everything each character had done in a match is readily available in the form of raw, unparsed, and undecoded numerical data. With past match data, it is possible to predict the trend of team win rates, popular character builds, and individual character’s historical win data as well as their win rate. All of which can help unfamiliar players get a decent idea of what to do

for their first time, or even for developers to see trends in recent matches and apply hotfixes, or character balancing to improve gameplay experience[5]. As mentioned, this can be done with sufficient data. Data which was not available, or at least to my search, was not publicly available. Therefore, I had to get my own with the help of Nimble Neuron, the company that developed Eternal Return, and their publicly available API upon request. With a sufficiently abundant dataset on hand, model training can begin.

1. Related Work

Several systems that are fundamentally similar to what my work suggests have already been developed and implemented for similar games ranging from similar genres to genres of the opposite spectrum, and some are even implemented in websites. Such projects or systems include, but are not limited to the following;

1. *Building a Machine Learning Model with Linear Regres- sion to Predict Dota 2 Teams Winning*

By utilizing team compositions and in-game statistics like hero selections and performance metrics, Cholig investigates the use of linear regression to forecast match outcomes in Dota 2 [6]. The main focus is on developing a single predictive model that generates the probability of a team winning based on the lineup chosen. Despite its simplicity, the method demonstrates the fundamental use of machine learning methods in competitive multiplayer games.

However, the study does not address build optimization or individual character metrics, and it is restricted to a single model and a single prediction type (win/loss). My suggested system, on the other hand, goes beyond this by presenting three distinct models that are uniquely adapted to the mechanics of Eternal Return and can forecast not just team win rates but also individual character win rates and optimized character builds and consider more character specific variables rather than simple win rate and pick rate.

1. *League of Legends Win Analysis*

Using League of Legends public match data, Zijie Mei does a statistical win rate study with an emphasis on identifying elements that increase the likelihood of winning [7]. The study highlights trends among top-performing players and teams by examining champion selections, performance indicators, and game length. The essay demonstrates how MOBA game data can yield valuable insights using very simple statistical techniques.

Mei’s method focuses on correlation-based insights rather than predictive modeling, despite the fact that it is instructive. By employing real machine learning algorithms to generate predicted probabilities and applying these techniques to the structurally similar but distinct game Eternal Return, my methodology expands on this kind of data exploration.

1. Methodology

This project uses a three-model machine learning system, each implemented independently in Python notebooks. The models focus on team win prediction, character win predic- tion, and character build recommendation, respectively. All models were trained using a custom-built dataset collected via the Eternal Return API, pre-processed and evaluated using standard classification and recommendation metrics.

1. *Approach*

The project uses three individual models to accomplish three separate tasks. First, to identify Team Placement, second, to predict character builds, and third, to predict individual character win rate. Classifiers and regression based models were used to best identify the model that gave the best rated outputs.

Important data pre-processing operations including address- ing class imbalance, transforming categorical variables, and normalizing inputs are all part of the modeling pipeline[6]. The system uses stratified 3-fold cross-validation to evaluate dependability. The top-performing models were chosen based on precision, recall, and F1 score after each model was trained and assessed using a variety of machine learning algorithms.

1. *Algorithms*

For the three models in this work, a variety of machine learning methods were used for both classification and re- gression tasks [7]. In order to identify the optimal performer for every predicting task, it was necessary to investigate performance across different algorithm families.

Here’s a breakdown of the algorithm used:

1. *Regression Models*

Regression methods were used to approach the Character Win Rate Prediction and Team Win Rate Prediction models[8]. With the help of features including kills, fatalities, assists, team damage, and more, these models sought to estimate a con- tinuous win rate number. We tested the following regression algorithms[9]:

* 1. Linear Regression
  2. Ridge Regression
  3. Lasso Regression
  4. ElasticNet
  5. Decision Tree Regressor
  6. Random Forest Regressor
  7. Gradient Boosting Regressor
  8. Histogram-based Gradient Boosting
  9. AdaBoost Regressor
  10. Extra Trees Regressor
  11. XGBoost Regressor[11]
  12. LightGBM Regressor[12]

1. *Classification Models*

Predicting the best build (equipment + augments) for a par- ticular character based on past match performance was the aim of the Character Build Recommendation model, which was

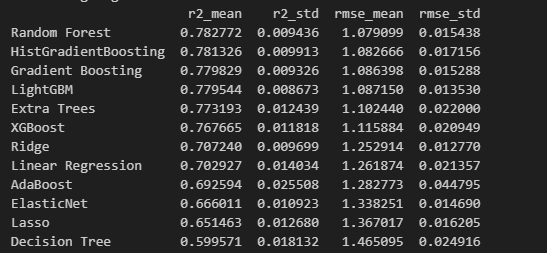


Fig. 1. Multi Model Test

structured as a classification problem[10]. The classification algorithms listed below were tested:

* 1. Random Forest (with class weights balanced)
  2. Gradient Boosting
  3. Logistic Regression
  4. AdaBoost
  5. Extra Trees
  6. XGBoost[11]
  7. LightGBM[12]
  8. Support Vector Classifier (SVC)
  9. K-Nearest Neighbors (KNN)
  10. Naive Bayes

1. Results
2. *Team Win Rate Prediction (Regression)*

In this program, the model takes data from each game and compiles them into the proper dataframe. Taking into teams of 3 per game, and simplifying the team’s stats, the model trains through the abundant dataset to learn.

After testing through the models, the model that was iden- tified to be the best was RandomForest: r2mean: 0.782772 rmsemean: 1.079099 Test Program:

1. *Character Build Recommendation (Classification)*

This classification model recommended the top 3 builds for a given character. It evaluated build effectiveness based on features like match outcomes, weapon types, and augment choices.

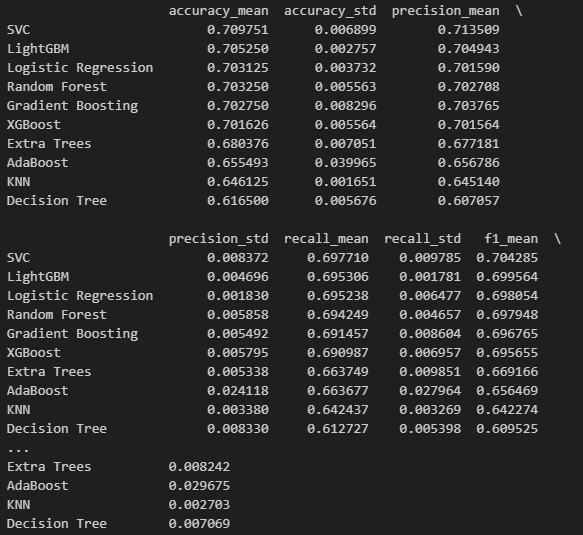
After testing through the models, the model that was identified to be the best was SVC: accuracymean: 0.705250 precisionmean: 0.704943 recallmean: 0.695306 f1mean: 0.699564 Test Program:

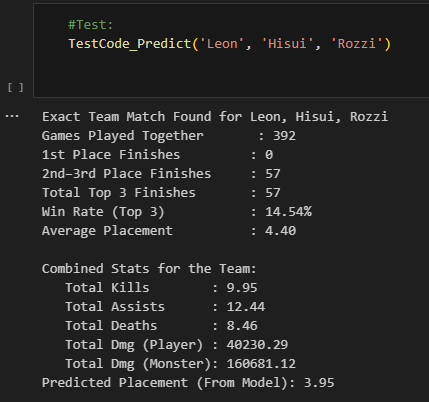
1. *Character Win Rate Prediction (Regression)*

Lastly, this model predicts the individual win rate of a single character using features like personal performance stats, damage dealt, and frequency of use throughout the dataset

After testing through the models, the model that was identified to be the best was HistGradientBoosting: r2mean: 0.680878 rmsemean: 1.261386 Test Program:

Fig. 2. Sample Testing



Fig. 3. Character Build Predictor Model Test

1. *Real-World Application*

Players and teams looking to gain an advantage in Eternal Return can benefit from the machine learning models used in this project. Players can make better selections throughout the planning and drafting stages by forecasting win rates based on team configurations and individual characters. By helping novice players find high-performing item and augment combinations, the Character Build concept speeds up their learning curve and enhances performance without depending entirely on trial and error. Furthermore, the framework and

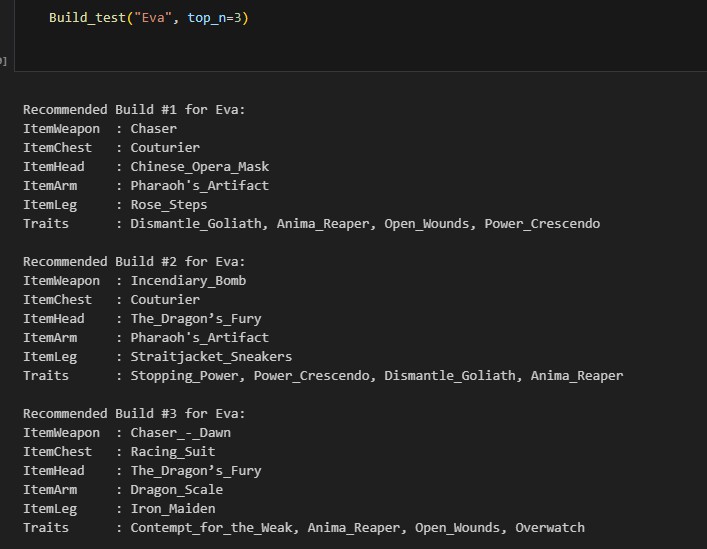


Fig. 4. Sample Testing

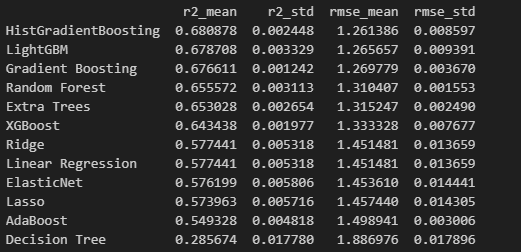


Fig. 5. Multi Model Test Results

dataset offer a starting point for further study on team-based strategy games, allowing for further in-depth investigations into adaptive matchmaking, player behavior prediction, and synergy modeling.

1. Conclusion and Future Work

This study shows that it is feasible to forecast team win rates, individual character win rates, and suggest the best character builds in Eternal Return by employing a multi- model machine learning approach. The models demonstrated encouraging outcomes in terms of prediction accuracy and use- fulness after undergoing a thorough preprocessing procedure and evaluating more than ten machine learning techniques for each task. In competitive situations, athletes, strategists, and analysts can use these instruments’ insightful information to make better informed choices.

In order to make the framework more usable in real-time, future research might concentrate on combining the three models into a single application or interface. The dataset might be expanded to include player skill level and patch changes, or more sophisticated deep learning methods could be used to increase accuracy. Furthermore, adding more specific match data, including positions, goals, or item timings, may result in even more thorough analysis and tailored suggestions.

1. Acknowledgments

First and foremost, I want to sincerely thank my instructor, Engr. Montances, Mark O., M.Sc. - ”Sir Mong,” for giving the direction and inspiration that enabled this endeavor. Even though I had been considering this undertaking for a while, I just did not have enough information or motivation to start.

Finally, I would like to extend my thanks to Eternal Return, a game I hold near and dear to my heart, and sometimes a game I wish to throw to the deepest pit there is on and off earth.

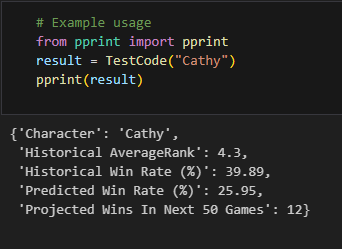


Fig. 6. Sample Testing

References

1. The role of Machine Learning in Game Development Domain
   * A review of current trends and future directions. (2021, November 1). IEEE Conference Publication — IEEE Xplore. https://ieeexplore.ieee.org/document/9647261
2. The role of Machine Learning in Game Development Domain
   * A review of current trends and future directions. (2021, November 1). IEEE Conference Publication — IEEE Xplore. https://ieeexplore.ieee.org/document/9647261
3. What are MOBA games? A complete guide. (n.d.). [https://www.konvoy.vc/blogs/what-are-moba-games](http://www.konvoy.vc/blogs/what-are-moba-games)
4. Moba vs Battle Royale: What are some popular Battle Royale games? — Lenovo Philippines. (2023, May 28). [https://www.lenovo.com/ph/en/glossary/moba-vs-battle-royale-games/](http://www.lenovo.com/ph/en/glossary/moba-vs-battle-royale-games/)
5. Eternal Return Review The Unspoken Diamond of Battle Royales Game8. (2024, August 14). Game8 the Top Gaming and App Walk- throughs Straight From Japan! https://game8.co/articles/reviews/736
6. Google for Developers. (2016, May 11). *Let’s write a pipeline - Machine learning Recipes #4* [Video]. YouTube. [https://www.youtube.com/watch?v=84gqSbLcBFE](http://www.youtube.com/watch?v=84gqSbLcBFE)
7. Data Professor. (2022, May 27). *Build your first ma- chine learning model in Python* [Video]. YouTube. [https://www.youtube.com/watch?v=29ZQ3TDGgRQ](http://www.youtube.com/watch?v=29ZQ3TDGgRQ)
8. *What is a regression model?* (2021, June 16). IMSL by Perforce. [https://www.imsl.com/blog/what-is-regression-model#:](http://www.imsl.com/blog/what-is-regression-model)˜:text=A% 20regression%20model%20provides%20a,by%20a%20linear%20 regression%20model.

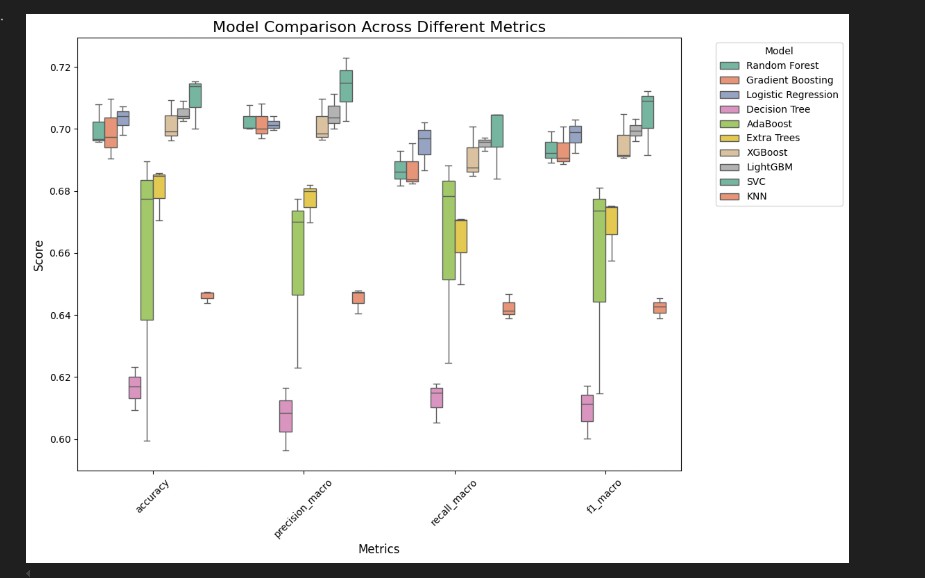


Fig. 7. Enter Caption

1. *scikit-learn: machine learning in Python — scikit-learn 1.6.1 documen- tation*. (n.d.). https://scikit-learn.org/stable/
2. *Choosing the best machine learning classification model and avoiding overfitting*. (n.d.). MATLAB & Simulink. [https://www.mathworks.com/campaigns/offers/next/choosing-the-best-](http://www.mathworks.com/campaigns/offers/next/choosing-the-best-) machine-learning-classification-model-and-avoiding-overfitting.html
3. *Python Package Introduction — xgboost 3.0.0 documentation*. (n.d.). https://xgboost.readthedocs.io/en/stable/python/python intro.html
4. *Python-package Introduction — LightGBM 4.6.0.99 documentation*. (n.d.). https://lightgbm.readthedocs.io/en/latest/Python-Intro.html
5. *NumPy*. (n.d.). https://numpy.org/
6. *pandas - Python Data Analysis Library*. (n.d.). https://pandas.pydata.org/
7. “Matplotlib — Visualization with Python.” Available: https://matplotlib.org/
8. Appendix

# Appendix: Use of AI Tools

* + **Debugging Assistance**: ChatGPT was utilized to trou- bleshoot and debug issues related to drop rate calculations and the overall simulation logic. It provided valuable suggestions for identifying errors and optimizing code performance.
  + **Code Snippets**: The AI assisted in generating code snip- pets for implementing specific features, such as random number generation for simulating item drops.
  + **Customization**: All code suggestions provided by Chat- GPT were thoroughly reviewed, customized, and refined to meet the specific requirements of the simulation project. This process includedadjusting logic, and opti- mizing the performance of the generated code.
  + **Learning and Understanding**: ChatGPT served as a learning resource, explaining the functionality of various code components. This ensured a comprehensive under- standing of the underlying logic before integrating it into the project.

# Appendix: Graphs and Charts

This appendix section includes any visual representations of the data.

# Appendix: Terminologies

This section provides definitions and explanations of key terms used throughout the paper.

* + **API (Application Programming Interface)** – Used to retrieve raw match data directly from Eternal Return’s backend service, enabling dataset construction.
  + **Augments** – Passive abilities or buffs equipped before a match that influence a character’s performance.
* **Character Build** – A combination of weapon, equip- ment, and augments optimized for a specific character to maximize performance in matches.
* **Classification Model** – A machine learning model used to categorize data points, such as predicting the top 3 builds for a given character.
* **Creeps (Monsters)** – Non-player units in the game that players can fight for experience or items; tracked in damage statistics.
* **Cross-Validation** – A technique for evaluating ML mod- els by dividing data into multiple training and testing subsets to ensure generalizability.
* **Eternal Return** – A multiplayer online survival game combining battle royale and MOBA elements, where players choose characters with unique abilities and com- pete to be the last one standing.
* **F1 Score** – A performance metric combining precision and recall to measure a model’s accuracy, especially useful for imbalanced datasets.
* **Individual Winrate** – The expected win percentage of a single character, based on historical performance data.
* **Match Data** – Information collected per game session, including kills, deaths, assists, team placement, and dam- age dealt.
* **Regression Model** – A model used to predict continuous values, such as estimating win rates or team placements.
* **Team Winrate** – The predicted probability or percentage chance that a three-character team will win a match.