PlayerUnknown's Battlegrounds (PUBG) Placement Predictor

CMPE 255 Data Mining

https://github.com/YashviDesaii/CMPE255-Final-Project

Under guidance of - Prof. Gheorghi Guzun

Pavan Nageswar Reddy Bodavarapu - 016285750

Yashvi Sanjaykumar Desai - 016420807

Vineeth Reddy Thalasani - 016422393

Sai Prasanth Guthula - 016063060

Table of Contents

1. Introduction

1.1. Objective

The goal of this project is to develop a model that predicts the placement of a player in the PUBG game using anonymized game data obtained from 65,000 games. The model is responsible for identifying key features which are essential for placement prediction and recommends effective winning strategies to the players which can help them improve their chance of winning.

1.2. Motivation

The motive of the project is to predict a player's placement in the PUBG game and help them improve their chances of winning by analyzing game data and performing exploratory data analysis on it. We analyze historical game data and recommend winning strategies to players. This helps players make informed decisions and improve their game. We aim to use data mining techniques to analyze data and select the most important features to use regressive models to predict the final placement of the player.

1.3. Literature/Market Review

Player Unknown's Battlegrounds (PUBG) is a multiplayer game which gained significant popularity since its release in 2017. It has over 70 million active users globally. Since a large amount of player data is available, various data mining techniques have been used in the past to perform analysis and provide useful insights which can help players improve their game play and increase their chance of winning. Several machine learning algorithms have been used to analyze data such as number of kills, distance traveled, and survival times to predict the outcome of the game. It has been found that a player's skill level and player's previous performance together make a great combination for predicting the player's placement. The trend suggests that there is a growing interest in using data analytics and predictive models to provide valuable insights to players to understand and improve their game play. Our project aims to contribute to this field by performing analysis on player data and using regression analysis to predict the final placement of a player.

2. System Design and Implementation Details

2.1 Algorithm(s) considered/selected

We analyzed 65,000 players' anonymized data to develop a successful predictive model. To achieve our goal, we considered several algorithms like decision trees, logistic regression and XGBoost. After hyper parameter tuning and using the best features, we narrowed it down to two machine learning algorithms - Random Forest and Gradient Boosting. Between these two, we selected the Gradient Boosting algorithm for its ability to handle complex and high dimensional data.

2.2. Technologies and Tools used

We used Python as our main programming language as it provides frameworks and libraries for data processing and machine learning. We used the Panda and the NumPy libraries for data processing and scientific calculations. We also used the matplotlib library for plotting insightful graphs and charts while performing exploratory data analysis. We used the Scikit learn library for developing our predictive model. For collaborating with each other, to perform our analysis in an interactive environment, we used Google colab.

2.3. Architecture

Our architecture of the model is based on the Gradient Boosting algorithm which can also be used for ensemble learning. We have three main components in the architecture namely, data processing, feature engineering and model training.

We performed exploratory analysis to get rid of redundant data and transform the raw data to prepare it for training. We also performed feature scaling to ensure all the features have equal importance. We performed feature engineering by selecting the most important and key features like number of kills, distance traveled, etc from the raw data and using them for our training model.

We then used Gradient Boosting on this data to predict the final placement of the player. We trained the model on the training dataset and we performed hyper parameter tuning to find the best combination,

3. Experiment / Proof of Concept Evaluation

3.1. Dataset Used

We obtained the dataset from Kaggle - https://www.kaggle.com/competitions/pubg-finish-placement-prediction/data

This Kaggle dataset was obtained using the PUBG API which contains almost 65,000 games worth of anonymized data. The dataset contains 29 features and 4446966 rows. Our target variable is the 'WinPlacePerc' which indicates the chances of winning for each player. The dataset gives game statistics such as 'distance traveled', 'longestKill', 'walkDistance' etc. These features give us insights into a player's game performance and strategy. We selected the most important features out of these for training of our model.

3.2. Methodology Followed

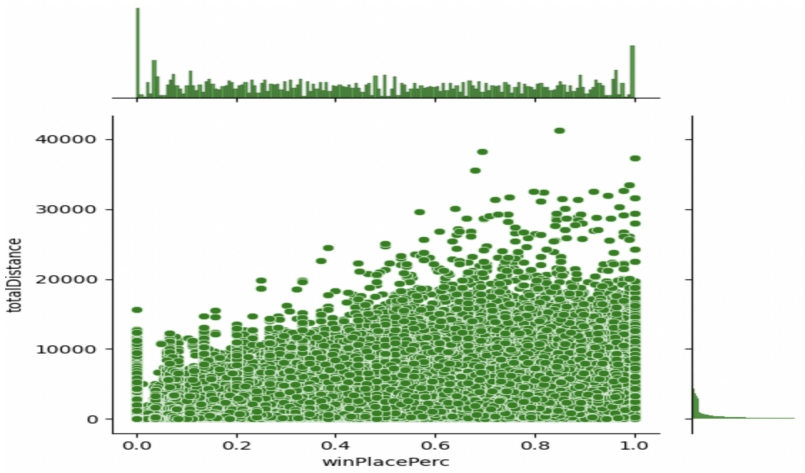
We followed 5 steps to draw insightful inferences from the data.

1. **Data Cleaning and preprocessing :-**

   We cleaned and preprocessed the data before we could start any analysis on it. We started by handling missing values, duplicates and outliers if any.

2. **Exploratory Data Analysis :-**

   The goal of performing EDA was to find correlations between the features and select the features that contribute to the target variable. We performed visualizations and calculated summary statistics.



The above figure describes the relationship between the target variable and the ‘totalDisatnce’ covered by the player. As we can see both the variables are closely correlated. However, we noticed that the reason was because the target variable was closely related to the ‘walkingDistance’ variable.

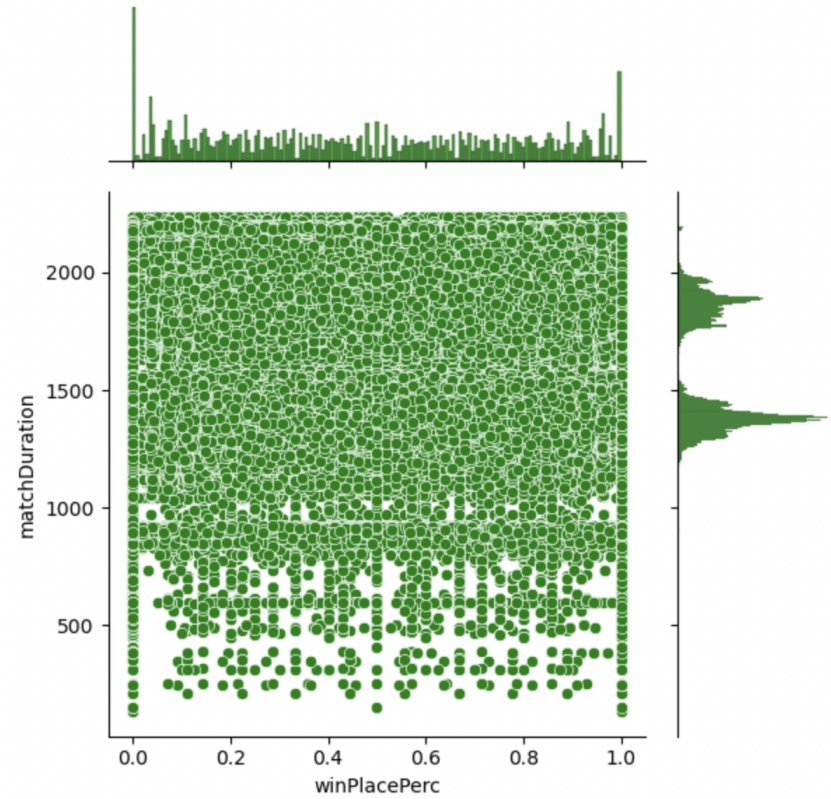


Fig. 2

Fig 2. Shows the correlation between the target variable and the matchDuration. We observe that there is a lot of variance in the

matchDuration feature and each game lasts for a varied amount of time thus we conclude that this feature does not weigh much to the target variable and hence we remove it from the final selection of features.

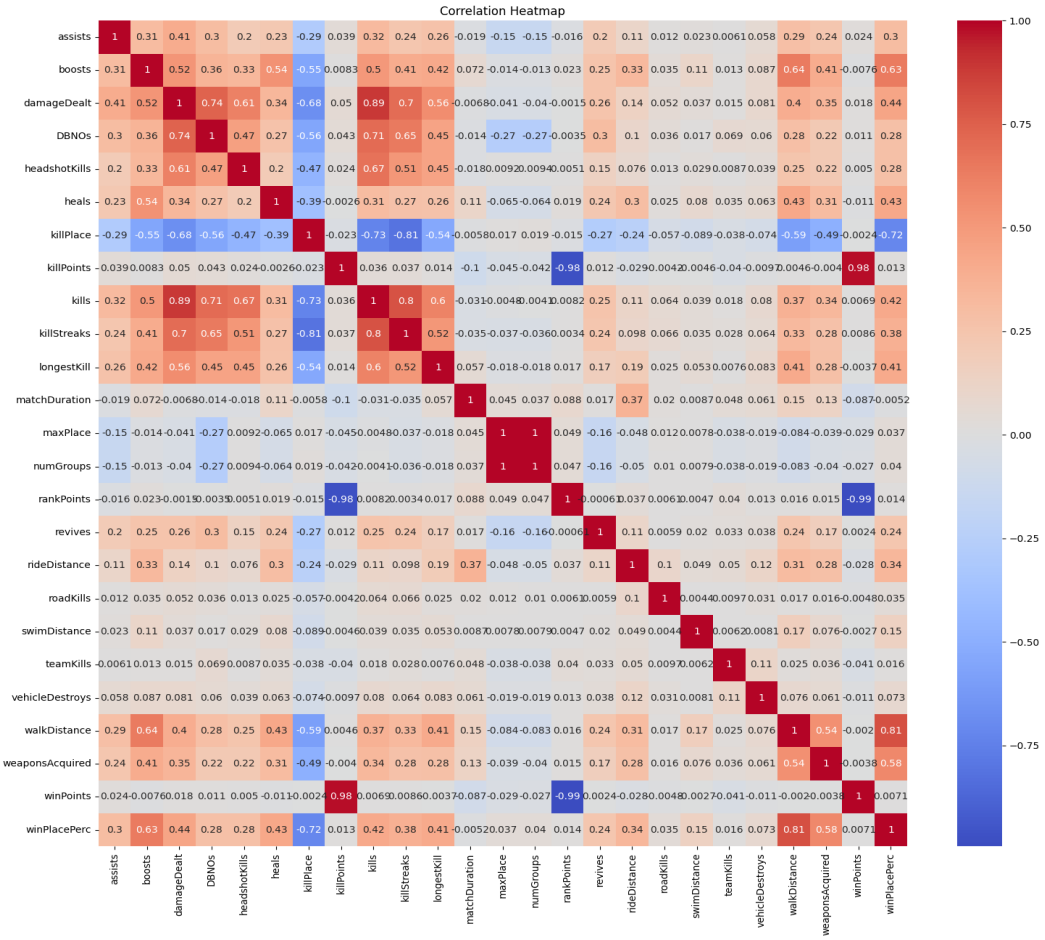


Fig. 3

Fig. 3 is the heatmap showing the correlation between all the features.

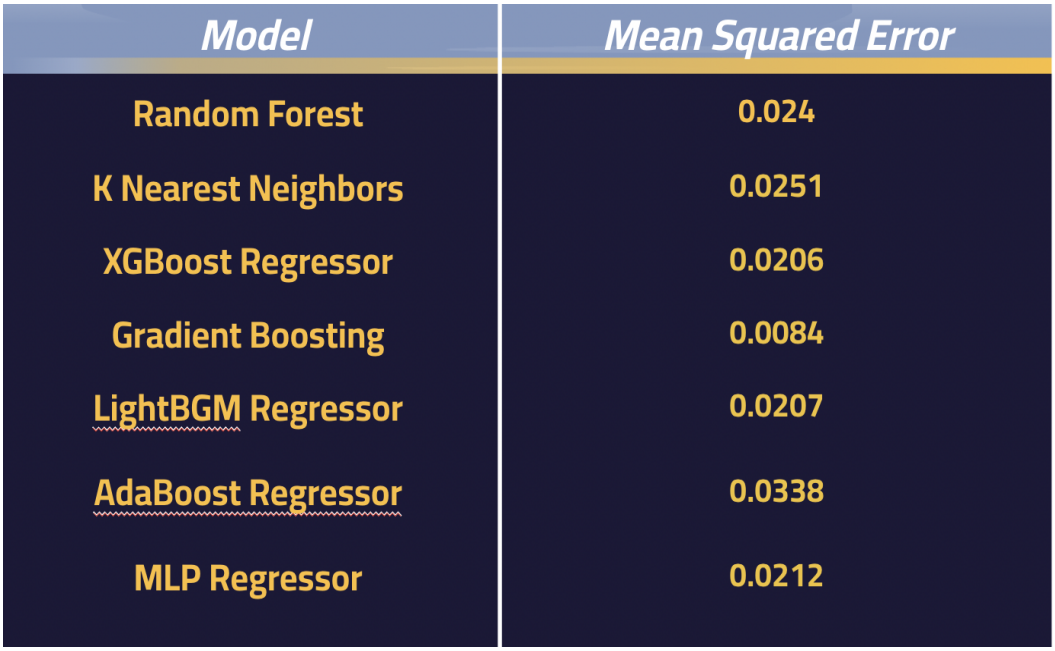
3. Model selection and evaluation :-

We chose several regression models to predict the final placement of the player. We trained the model on the following algorithms -

- Random Forest
- Gradient Boosting
- MLP
- LightBGM
- XGBoost
- KNN
- AdaBoost

We performed hyper parameter tuning on each of these models. We used the mean squared error metric to evaluate each model. The best performance was given by the Gradient Boosting algorithm which gave the least mean squared error of **0.0084494** with these parameters - n_estimators=50, learning_rate=0.1, max_depth=5, random_state=4. We chose this model because it builds a model iteratively and not just once like other decision trees or regression models. Each weak model has its results corrected by the next model. This model also handles missing data as well as outliers well. Overall it is a strong and flexible model which makes it an ideal choice for our purposes.

3.3. Results



Model	Mean Squared Error
Random Forest	0.024
K Nearest Neighbors	0.0251
XGBoost Regressor	0.0206
Gradient Boosting	0.0084
LightBGM Regressor	0.0207
AdaBoost Regressor	0.0338
MLP Regressor	0.0212

Fig. 4

Fig.4 is the comparative chart of all the models that we analyzed. It shows the comparative analysis of all the models and their mean squared errors.

Overall, we performed data processing, feature selection, model selection and evaluation, and hyper parameter tuning to achieve the desired results.

4. Discussions and Conclusions

4.1. Decisions made

We made several important decisions throughout the project. The project was focused on using regression models to predict the final placement of a game player and to suggest winning strategies to improve their performance. We explored the dataset using several visualizations techniques. We gained valuable insights from these visualizations which helped us in selecting features that were most important in predicting the target variable. We trained and evaluated the data on multiple models and selected the best performing model based on the minimum mean squared error.

4.2. Difficulties Faced

Our dataset was large and required cleaning as well as preprocessing. One of the main challenges that we faced was finding the optimal hyperparameters and fine tuning it. Using grid search to find these parameters and get the best results from the Gradient Boosting algorithm was challenging and took time and effort.

4.3. Things that worked

Our strategy to first perform exploratory data analysis worked well for us. We were able to visualize the data better and it helped us gain valuable insights from the visualizations. This consequently helped us in feature selection and engineering. As a result, we were able to use the most relevant features for our model training and get the best results. Our evaluation metric, mean squared error, was effective in evaluating the models.

4.4. Things that didn’t work

Some algorithms took a longer time to train which might have affected the hyperparameter tuning.

4.5. Conclusion

We successfully predicted the final placement of a PUBG player using regression models. Exploratory data analysis gave us valuable insights about the dataset and the correlation of the features with respect to the target variable. The Gradient Boosting algorithm worked best for us and feature selection was proven helpful. Grid search for hyper parameter tuning was helpful.

5. Task Distribution

Pavan Nageswar Reddy Bodavarapu	Performed initial exploratory data analysis and trained the model on random forest as well as gradient boosting algorithms. Worked on creating the presentation slides.
Yashvi Sanjaykumar Desai	Assisted with exploratory data analysis and worked on MLP regressor to train the model. Implemented grid search for hyper parameter tuning and worked on the report.
Vineeth Reddy Thalasani	Implemented LightBGM and XGBoost algorithms to train the model. Found the most suitable metric to evaluate the performance of the models. Also worked on resting the presentation slides.
Sai Prasanth Guthula	Trained the model on AdaBoost and Knn regressors. Created comparative analyses for all the algorithms. Worked on the report