

Project Proposal

PUBG Finish Placement Prediction



Team 5-

Ankit Yadav

Ritu Agrawal

Preetam Jain

Overview:

Battle Royale-style video games have taken the world by storm. 100 players are dropped onto an island empty-handed and must explore, scavenge, and eliminate other players until only one is left standing, all while the play zone continues to shrink.

We are given over 65,000 games' worth of anonymized player data, split into training and testing sets, and will predict final placement from final in-game stats and initial player ratings.

We will try to answer following question from our analysis.

What's the best strategy to win in PUBG? Should you sit in one spot and hide your way into victory, or do you need to be the top shot?

Dataset:

We are provided with many anonymized PUBG game stats, formatted so that each row contains one player's post-game stats. The data comes from matches of all types: solos, duos, squads, and custom; there is no guarantee of there being 100 players per match, nor at most 4 players per group.

Link- <https://www.kaggle.com/c/pubg-finish-placement-prediction>

File descriptions

- train_V2.csv - the training set
- test_V2.csv - the test set

Number of rows and columns:

test_V2.csv-- 1.93m x 28

train_V2.csv-- 4.45m x 29

Our Target variable is-

winPlacePerc –

The target of prediction. This is a percentile winning placement, where 1 corresponds to 1st place, and 0 corresponds to last place in the match. It is calculated off of maxPlace, not numGroups, so it is possible to have missing chunks in a match.

Data fields

DBNOs - Number of enemy players knocked.

assists - Number of enemy players this player damaged that were killed by teammates.

boosts - Number of boost items used.

damageDealt - Total damage dealt. Note: Self inflicted damage is subtracted.

headshotKills - Number of enemy players killed with headshots.

heals - Number of healing items used.

Id - Player's Id

killPlace - Ranking in match of number of enemy players killed.

killPoints - Kills-based external ranking of player. (Think of this as an Elo ranking where only kills matter.) If there is a value other than -1 in rankPoints, then any 0 in killPoints should be treated as a "None".

killStreaks - Max number of enemy players killed in a short amount of time.

kills - Number of enemy players killed.

longestKill - Longest distance between player and player killed at time of death. This may be misleading, as downing a player and driving away may lead to a large longestKill stat.

matchDuration - Duration of match in seconds.

matchId - ID to identify match. There are no matches that are in both the training and testing set.

matchType - String identifying the game mode that the data comes from. The standard modes are "solo", "duo", "squad", "solo-fpp", "duo-fpp", and "squad-fpp"; other modes are from events or custom matches.

rankPoints - Elo-like ranking of player. This ranking is inconsistent and is being deprecated in the API's next version, so use with caution. Value of -1 takes place of "None".

revives - Number of times this player revived teammates.

rideDistance - Total distance traveled in vehicles measured in meters.

roadKills - Number of kills while in a vehicle.

swimDistance - Total distance traveled by swimming measured in meters.

teamKills - Number of times this player killed a teammate.

vehicleDestroys - Number of vehicles destroyed.

walkDistance - Total distance traveled on foot measured in meters.

weaponsAcquired - Number of weapons picked up.

winPoints - Win-based external ranking of player. (Think of this as an Elo ranking where only winning matters.) If there is a value other than -1 in rankPoints, then any 0 in winPoints should be treated as a "None".

groupId - ID to identify a group within a match. If the same group of players plays in different matches, they will have a different groupId each time.

numGroups - Number of groups we have data for in the match.

maxPlace - Worst placement we have data for in the match. This may not match with numGroups, as sometimes the data skips over placements.

winPlacePerc - The target of prediction. This is a percentile winning placement, where 1 corresponds to 1st place, and 0 corresponds to last place in the match. It is calculated off of maxPlace, not numGroups, so it is possible to have missing chunks in a match.

EDA & Feature Engineering

As the data is not clean and have lot of unimportant columns. So, we need to do proper EDA and cleaning before using data for our ML models.

Training Data Values: -

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4357336 entries, 0 to 4357335
Data columns (total 26 columns):
Id                int64
groupId           int64
matchId           int64
assists           int64
boosts            int64
damageDealt       float64
DBNOs             int64
headshotKills     int64
heals             int64
killPlace         int64
killPoints        int64
kills             int64
killStreaks       int64
longestKill       float64
maxPlace          int64
numGroups         int64
revives           int64
rideDistance      float64
roadKills         int64
swimDistance      float64
teamKills         int64
vehicleDestroys   int64
walkDistance      float64
weaponsAcquired   int64
winPoints         int64
winPlacePerc      float64
dtypes: float64(6), int64(20)
memory usage: 864.3 MB
```

Training data head-

```
In [4]: train.head()
```

Out[4]:

	id	groupId	matchId	assists	boosts	damageDealt	DBNOs	headshotKills	heals	killPlace	killPoints	kills	killStreak
0	0	24	0	0	5	247.30	2	0	4	17	1050	2	1
1	1	440875	1	1	0	37.65	1	1	0	45	1072	1	1
2	2	878242	2	0	1	93.73	1	0	2	54	1404	0	0
3	3	1319841	3	0	0	95.88	0	0	0	86	1069	0	0
4	4	1757883	4	0	1	0.00	0	0	1	58	1034	0	0

We found that the data have all the elements, but it does not have a match data.

So, in feature engineering we will add all the missing columns and later use them in our prediction model.

For e.g.-

No of players in a match,
Match wise kill, etc.

Prediction:

Since, it is a Regression Problem, we will predict the final ranking of the player depending on the past data. So, for prediction we will be using the following models-

1. Linear Regression
2. Neural Network
3. Random Forest
4. Gradient Boosting Machine

We will compute the matrix to find the best model and finally, we will pickle this model so that it could be used to create a front end (Restful web app) where we can use the user interaction for taking input as player and match statistics and displaying the ranking for the same.

The whole thing will then be dockerized and deployed on cloud.