**PUBG Players’ Stat Analysis**

Yijie Wang

yw3156@columbia.edu

Electrical Engineering, Columbia University

Yihang Ding

yd2459@columbia.edu

Electrical Engineering, Columbia University

Abstract

PUBG is a popular online game. To better analyze the game pattern and help players improve their skills, we did feature engineering to find out the most important features w.r.t. win place and did EDA to visualize the result. We found out cheaters in game with anomalies detection, and we predict the win place with players’ game statistics.

***Keywords-component: EDA, Anomalies Detection, Win Place Prediction, Gradient Boosted Tree***

Introduction

PlayerUnknown’s BattleGrounds (PUBG) is an online multiplayer battle royale game populated in the past year. In the game, up to one hundred players parachute onto the map and search for weapons and equipment to kill others while avoiding getting killed. The available safe area of the game's map decreases in size over time, directing surviving players into tighter areas to force encounters. The last player or team standing wins the round.

With its big maps(up to 8 km\*8 km), complicated terrain and abundant game mechanics, this game is easy to learn, but hard to master.

With more than 20 kinds of game statistics, considering the complexity of the game, we analyzed the most important features and visualize them; It’s widely agreed that there are many cheaters in this game, so we did anomalies detection to find out the cheaters. Lastly, we used the cleaned data to predict the win place.

1. Related Works

As digital games market size has reached more than $125 billion in 2018[1], gamers and e-sports pro players have the demand for game statistics for skills improvement. There are several game statistics websites and mobile applications, such as www.op.gg[2], which provides insights to gamers for several popular video games such as PUBG, with 30M monthly active users. In this website, players can track their game statistics and career results, it also provides performance of different weapons and information about maps such as vehicle spawns. While such websites provides very detailed statistics for each player, they do not focus too much on data analysis based on huge amount of game data.

1. Overview

Our project consists of 3 parts.

In the first part, we would explore our data and find out the most correlated features with winning place. Applying feature engineering on our over 4 millions line dataset, we find some important feature with winning place. Three different modes are also compared in our analysis.

In the second part, we focus on finding cheaters in the game. With null hypothesis testing, cheaters are found and disposed from our dataset.

In the third part, with clean data after anomalies detection, we use 3 different regression algorithms to predict the final win place for solo game mode. And we analyze the feature importances in our best model.

Detailed description of our work will be presented below separately.

1. Implementation
2. Dataset

We use dataset from Kaggle to do our project. The dataset is stored in CSV file which contains 4446966 entries and 29 features.

|  |
| --- |
| Id object groupId object matchId object assists int64 boosts int64 damageDealt float64 DBNOs int64 headshotKills int64 heals int64 killPlace int64 killPoints int64 kills int64 killStreaks int64 longestKill float64 matchDuration int64 matchType object maxPlace int64 numGroups int64 rankPoints 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(19), object(4) memory usage: 983.9+ MB |

In order to reduce memory usage, we iterate through all the columns of a dataframe and modify the data type, which reduces 70% memory usage.

1. Preprocessing

Dirty data is the first problem we need to deal with. In this section we find all the data that is invalid by checking their form and detecting anomalies.

From dataset itself, we find all the invalid data in the dataset. From their reality meaning, we find cheaters and AFKs.

Also, we did a normalize to kill and damage in order to make the data more reasonable.

1. EDA

We make some exploratory data analysis to the dataset and find several interesting facts from that. The kills(damage dealt), walk distance, swim distance, ride distance and heal are our main target features.

1. Win Place Prediction

We make a win place prediction based on cleaned dataset. As in game mode *Duo* and *Squads*, the level of teammates can affect the win place significantly which cannot be fully revealed by the basic statistic in our dataset, so we only consider *Solo* mode in this task. So those features that are only related to *Duo* and *Squads*, such as TeamKills and Revives, can be removed.

The label we want to predict is WinPlacePerc, which is a percentile winning placement, where 1 corresponds to 1st place, and 0 corresponds to last place in the match.

By drawing a correlation heatmap, we can drop features with very low correlation to WinPlacePerc, after the feature selection, the data has 173521 entries and 18 features.

We use three algorithm including Linear Regression, Random Forest and Gradient Boosted Decision Tree(GBDT) to do the regression.

We use Linear Regression to see how good a simple model can do in this task, then we use Random Forest and GBDT, as these two models are different ways of ensemble learning, we want to compare the effectiveness of bagging and boosting.

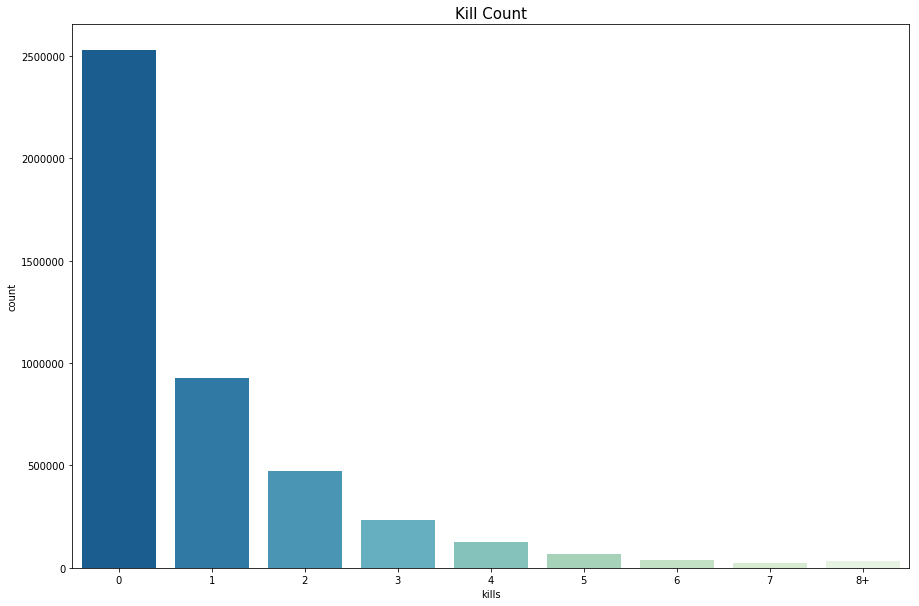
This task is implemented with Scikit-learn.

1. Experiment Results
2. Exploratory Data Analysis

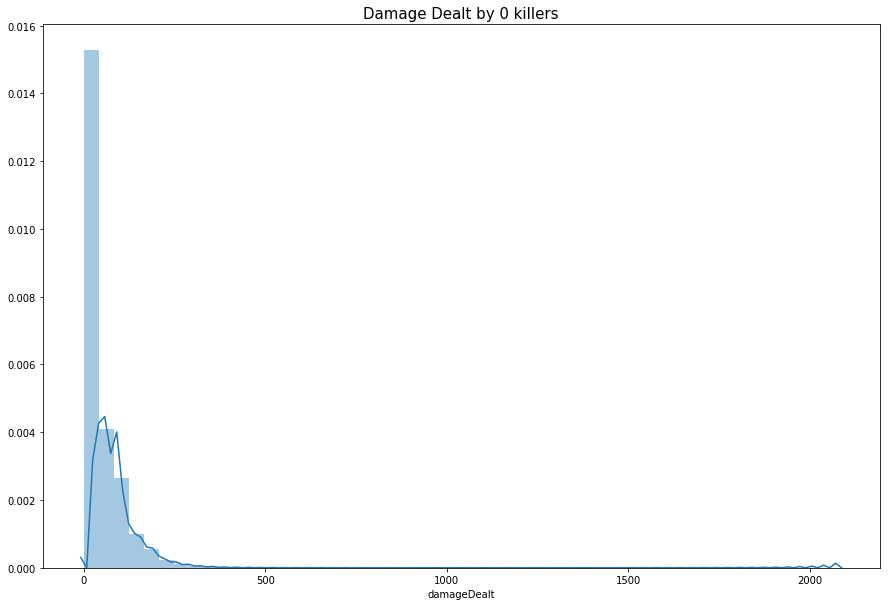
1.1 Killers

From interviews with players and our gaming experiences, killers tend to be the most consider fact. First of all, we would like to look through killers to find out some useful information.

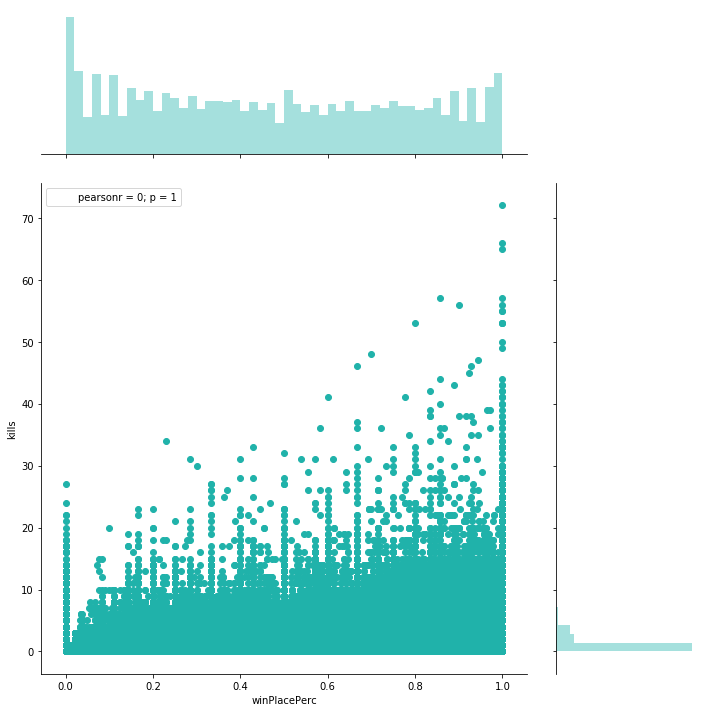
The average person kills 0.9248 players, 56.8% of people kill nobody, 90% of people have 3.0 kills or less, 99% of people have 7.0 kills or less, while the most kills ever recorded is 72.



From kill count diagram, we can find out less than half players can make a kill in a game. If they can’t make a kill, we would like to look at their damages.



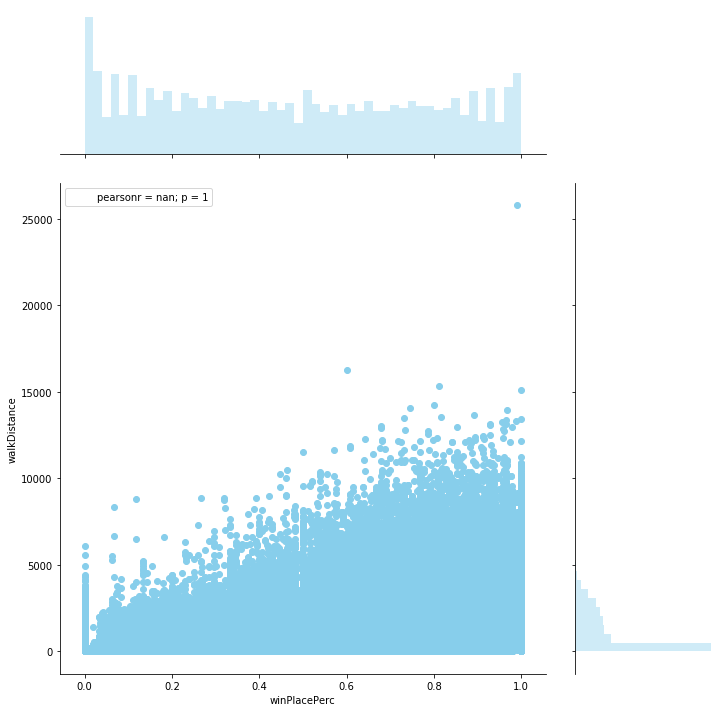
We can find out some players even make more than 1000 damages, which is enough for killing more than 10 players, without killing a person.



From this joint graph, we can see the correlation between kills and win place is quite obvious.

1.2 Runners

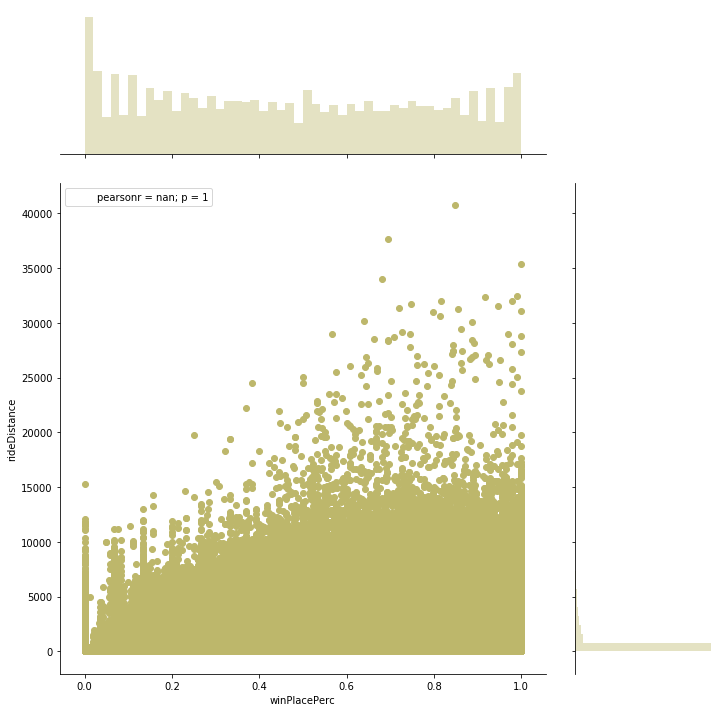
We can find out that the walking distance distribution is comparable smooth, while a large number of players walk a very short distance.



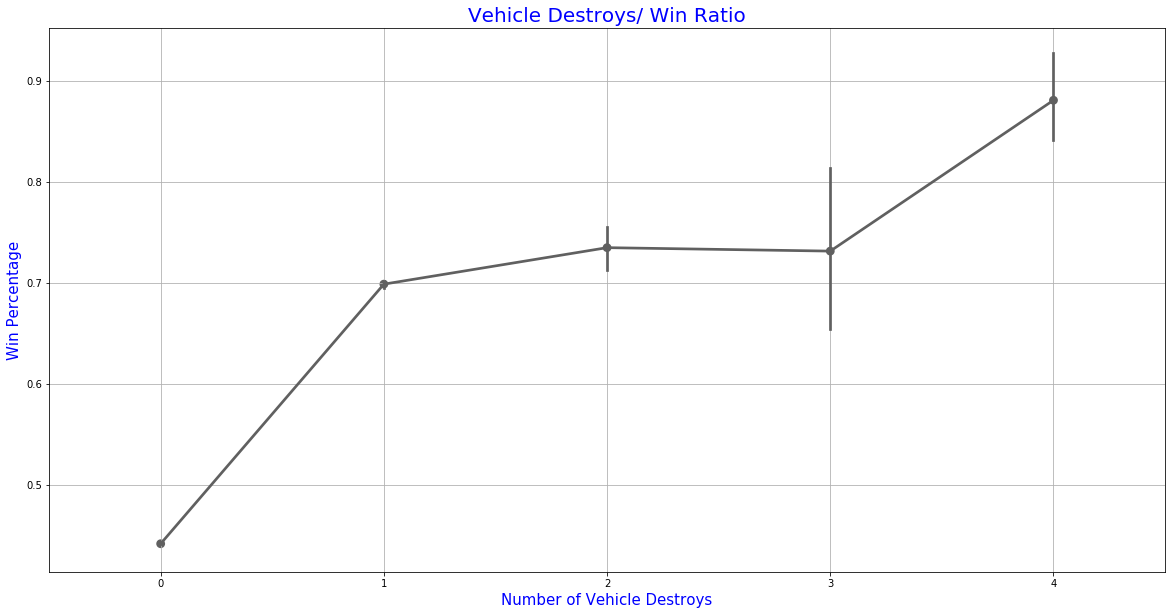
We make this joint graph again to find out the connection between walk distance and win place.

1.3 Drivers

Ride distance distribution is quite different than the diagram before. Not everyone would choose to drive a car to move in a game. There are mainly two reasons. Firstly, car is hard to handle in the game and may cause a hurt or even a death if you can’t drive in a right way. Secondly, vehicles would make a loud noise and expose you as a target.



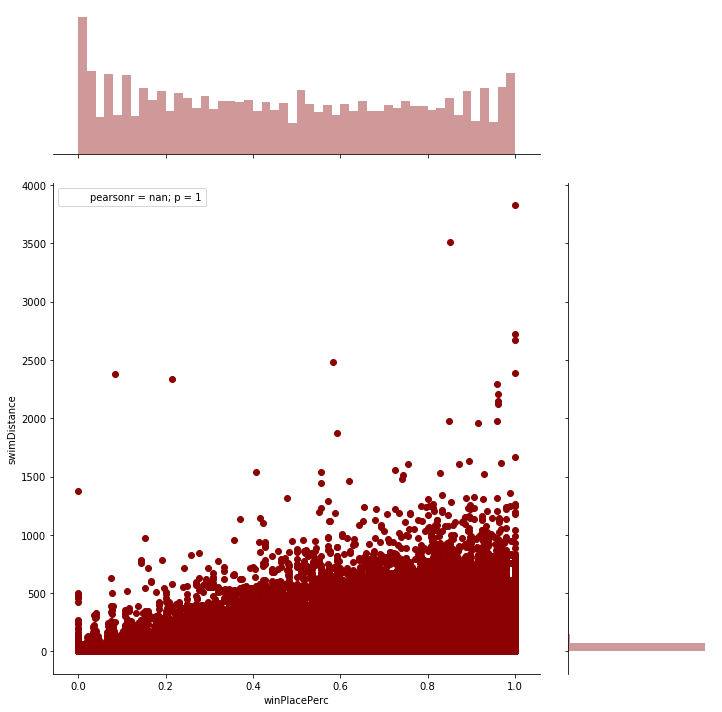
With this joint graph, we find out a less direct but still clear correlation.



One interesting thing is that vehicle can be destroyed in this game, which means we can destroy the vehicle to kill everyone in the car and close to the car just like what would happen in the real would.

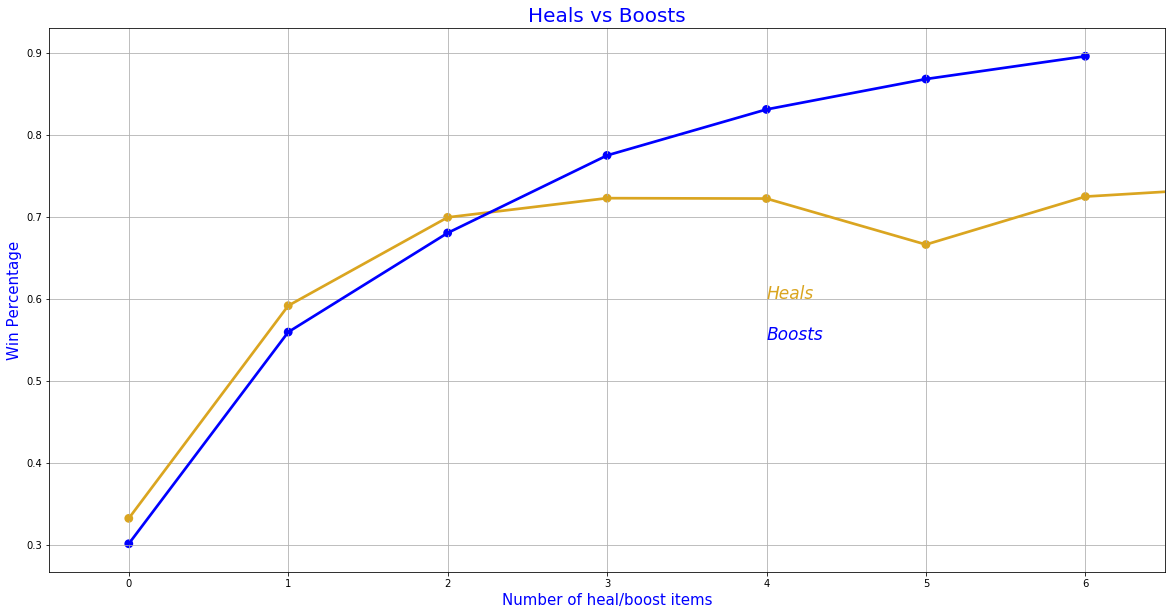
1.4 Swimmers

Swim distance is quite likely to ride distance. We don’t necessary need to swim in a game. Swim is even more dangerous than drive because you can’t shot in the water and you are quite slow.

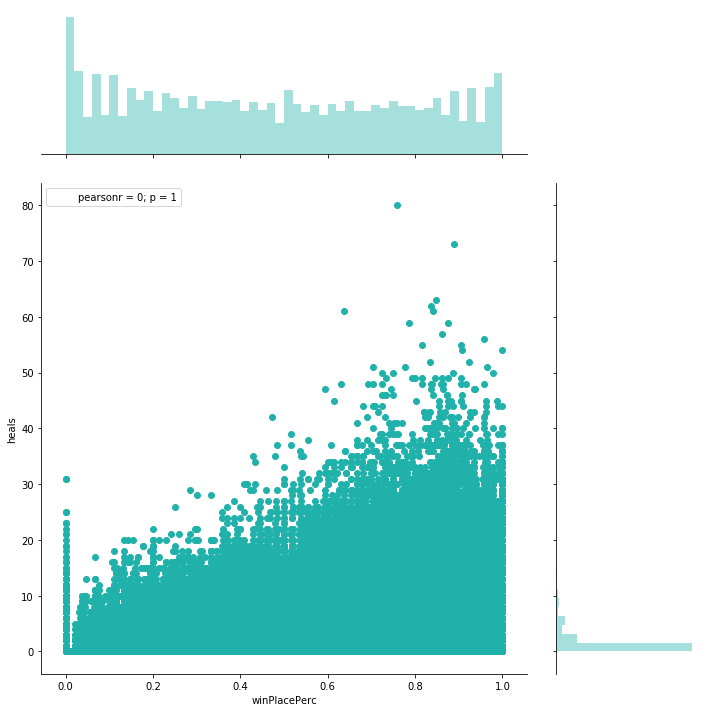


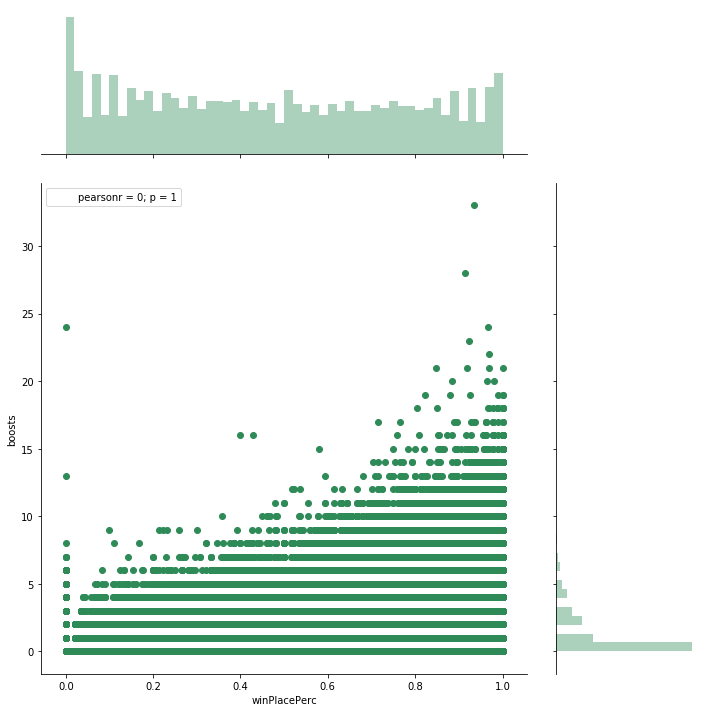
1.5 Healer

In this game, we have two kinds of healer, which are heals and boosts. Heals can heal you immediately after using them, while boosts heal you gradually and make you run faster during the heal process.



From this graph, it is clear that using boosts is a common sense for those who can win this game or get a high place.

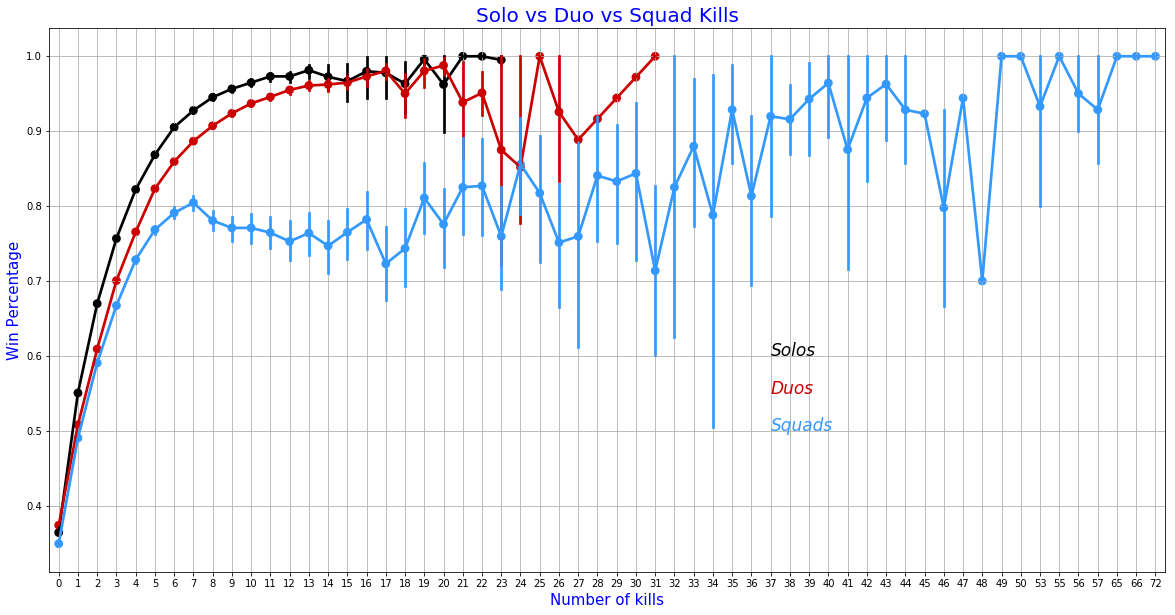




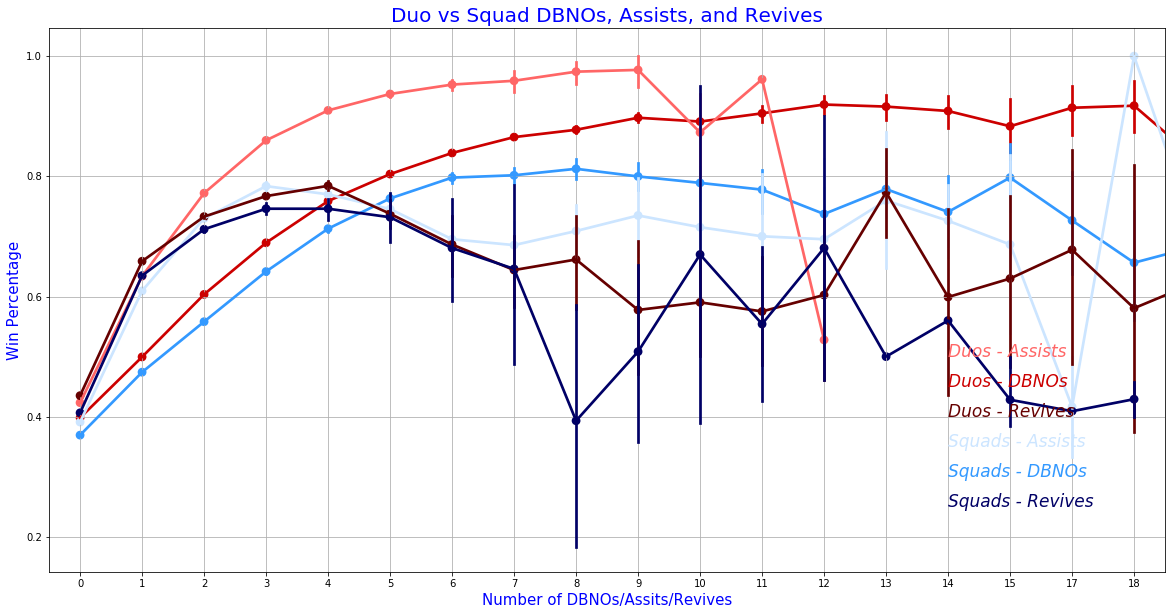
From the two graph above, we can notice a both clearly correlation.

1.6 Solos, Duos and Squads

Solos, duos and squads are three models you can choose. If you choose solos, you would play alone. Duos means you could play with one teammate and squad is a four man team. There are 709111 (15.95%) solo games, 3295326 (74.10%) duo games and 442529 (9.95%) squad games.Since we didn’t find a significant connection between kills and win place, we would find it out in different model.

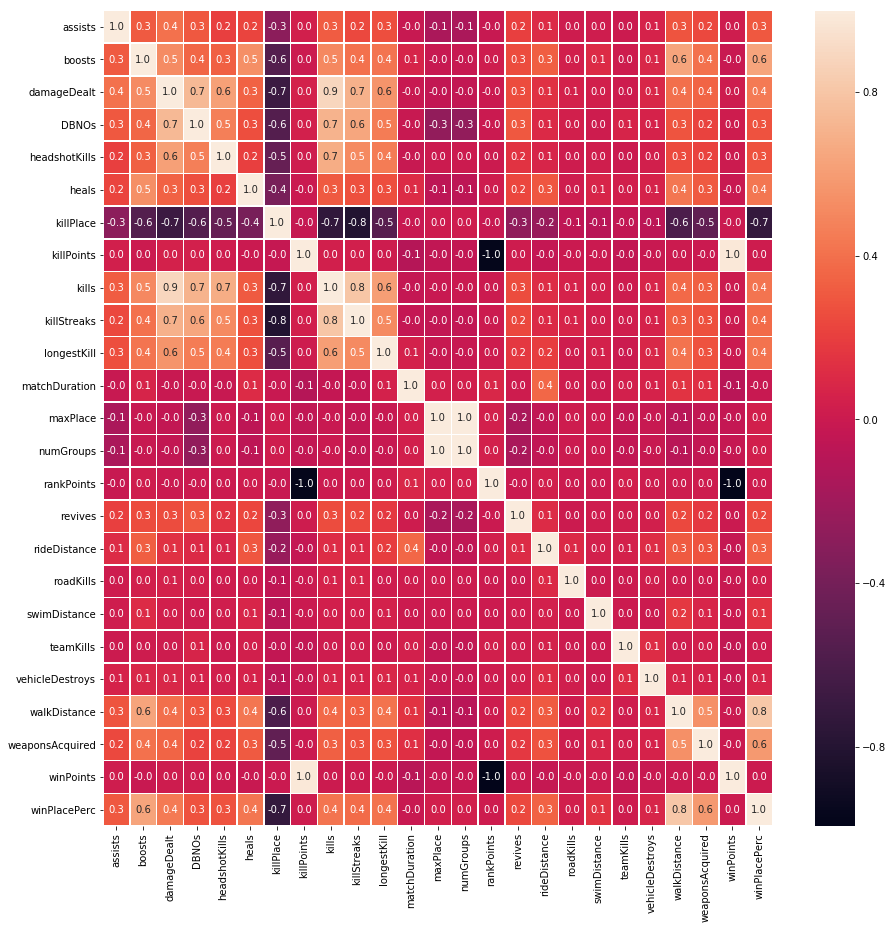


In the solo and duo games, kills are more important in compare with squads. Kills nearly stop to boost the chance to win after 7 kills in the squads.

In the other hand, team works are much more important. In this graph, DBNO means Down But Not Out. If your teammate is DBNO, you still have a chance to revives them and have them back in the battle. Only a some amount of cooperation would make a huge difference.

1.7 Correlation Matrix

Consider about the number of features, we use correlation to find out the most correlation features and drop some useless features.



1.7 Joined Players and Normalize

In this game, the maximum number of players is 100 every game. However, the number may not always be 100, so we need to take a look at how many players in a game.



It is quite obvious that you would be hard to find others in a game with less players. With this situation, we use normalize to make our data more comparable.

train['killsNorm'] = train['kills']\*(100/train['playersJoined'])

train['damageDealtNorm'] = train['damageDealt']\*(100/train['playersJoined'])

Those are what I used to normalize.

1. Anomalies Detection

We would focus on finding out cheaters in this part. When deal with 100% headshot, we would use null hypothesis testing to make sure we can classify them as cheaters.

2.1 Illegal data

Find records that the win place perception is null, which return one result.

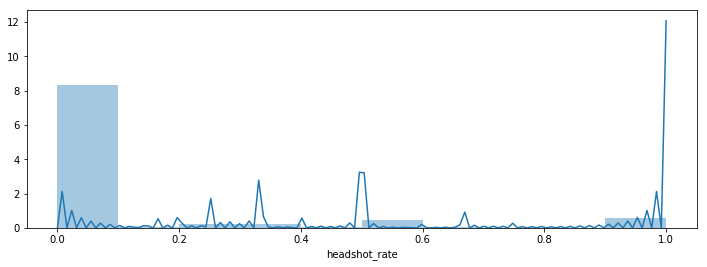
2.2 Cheaters

* Find killing without moving.

Sum the total moving distance of walking, swimming and driving as totalDistance. In this game, we need to move around to find weapons and equipment, so a player kill others without moving is definitely a cheater.

* 100% headshot

We find out some players kill more than 10 in a game with 100% headshot. It is very difficult but still theoretically possible. 100% headshots rate may not mean they are cheaters because we may have that by chance if we have a few kills. However, that may be anomalous if we have a lot of kills.



It is reasonable to use a null hypothesis testing to make sure this.

We'll set up the null and alternative hypotheses and try to find out.

1. Null hypothesis.

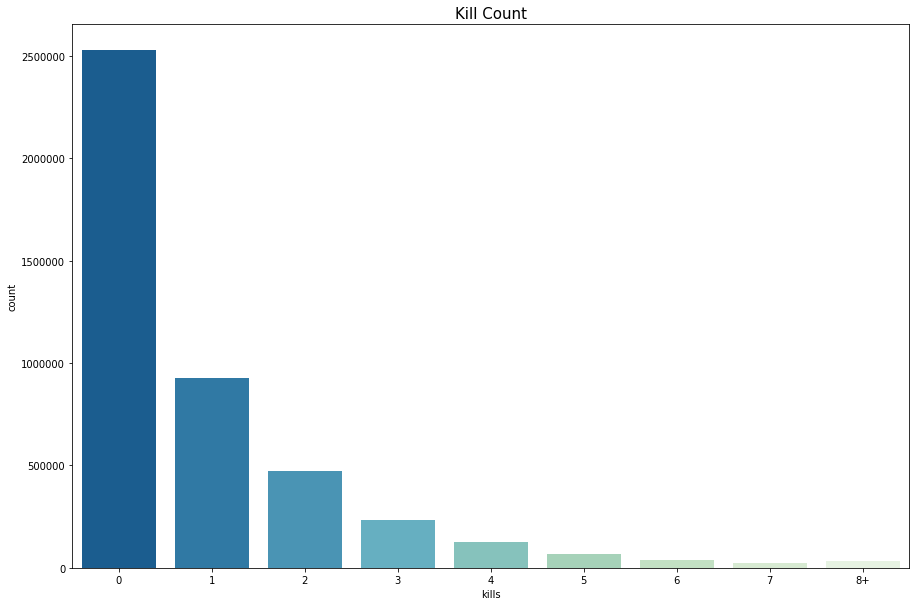
The kills number of 100% headshots are like a random sample out of all kills numbers. The average came out higher than that of overall kills due to chance variation.

2) Alternative hypotheis.

The kills number of 100% headshots are too large to be the result of chance variation alone.

If the null hypothesis were true, then the headshoters kill would be comparable to those drawn at random without replacement from all kills number. So let's create an array of all kills number and draw at random from it.

We simulate the difference of numbers of kills between 100% headshots killer and overall players 100 times. The Simulated Statistic is 0.24956 while the observed Statistic is 8.3. So, we got a p-value of 0.13. The p-value could be considered low, which means the null hypothesis could be rejected. However, they may be some quite good players. We could take another feature into our consideration.



From kill counts, we found most of players make 0 or 1 kill in a game. To be specific, the average person kills 0.92 players, 99% of people have 7.0 kills or less. We choose 7 kills to get rid of players who make a 100% headshot rate by chance in this place.

There are 99% players kill 7 or less people in a game. It is quite rare to kill more than 7 people in a game. If a player have a 100% headshot when they made such an incredible number of kills, we could consider this as our null hypothesis.

1. Null hypothesis.

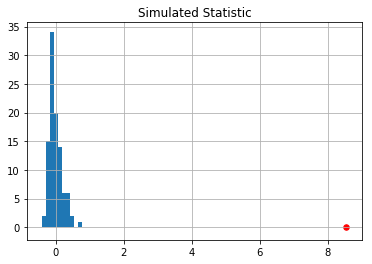
The kills number of 100% headshots and more than 7 kills are like a random sample out of all kills numbers. The average came out higher than that of overall kills due to chance variation.

2) Alternative hypotheis.

The kills number of 100% headshots and more than 7 kills are too large to be the result of chance variation alone.

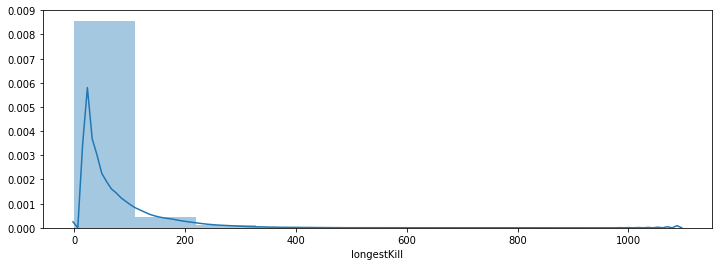
If the null hypothesis were true, then the headshoters kill would be comparable to those drawn at random without replacement from all kills number. So let's create an array of all kills number and draw at random from it

We simulate the difference of numbers of kills between 100% headshots killer who kill more than 7 people and overall players 100 times. The Simulated Statistic is 0.1956 while the observed Statistic is 8.3. So, we got a p-value of 0.0034 and deny our null hypothesis.



* Longest kill

First, I find out some players who have a very long shot, which is over 1000 meters. They are quite close to 1000 meters. From official document, a shot around 1000 meters are still possible. Personally, my longest shot is around 1000. So I decide to keep them too.



* Walk distance

From official document, the fast running speed in the game is 6.3 m/s, which means if you are running all the time at ground with empty hand and make it to be the last survivors, you can run 11340 meters in a 30 minutes game. For those who run over 11340 meters, they are absolutely cheaters.



* Ride distance

The longest ride distance is 40704 m, which is still possible. We would keep all records.

* Swim distance

The longest swim distance is 3824 m, which is weird but also possible. So we would keep them.



* Weapons acquired

In this game, you need to search rooms to find weapons and all you can take is two main weapons with a pistol and a melee weapon, which makes it impossible to acquire a lot of weapons in a game. In this case, I delete records with more than 50 weapons acquired.



2.3 AFK

AFK(Away From Keyboard) means players who are online but have no action at all. We find it by finding all people with killing noone and never moving.

2.4 Summary

In this section, we drop records fit those features below:

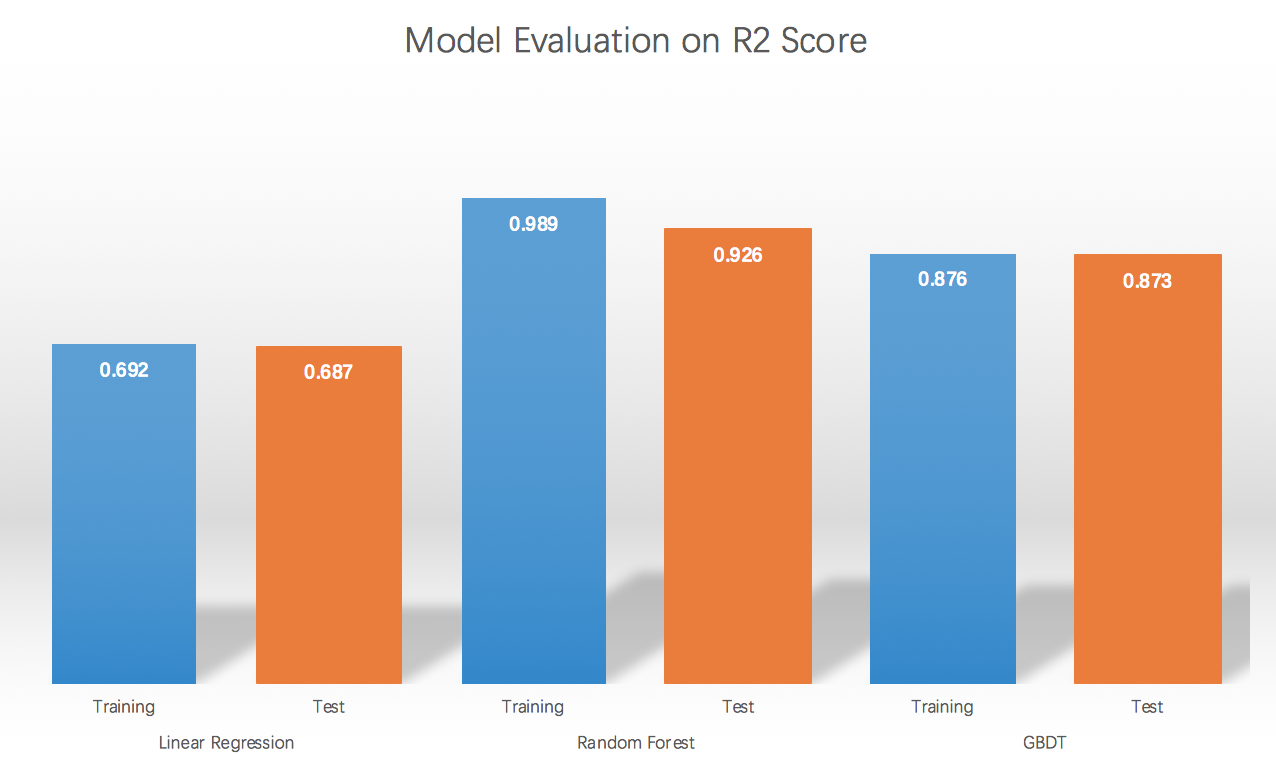
* Killing without moving;
* Walk distance over 11340 m;
* Weapons acquired over 50 in a game;
* Killing noone and never moving;
* Invalid data.

We remove 1 invalid data, 251 cheaters and 88,205 AFK in the dataset.

1. 3. Win Place Prediction

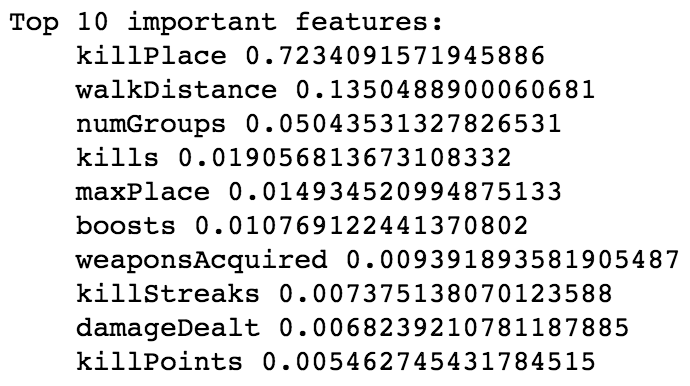
We use R2 coefficient of determination as the metric to evaluate our model. The best model could have a score of 1 and the worst model could have a score of 0.

Below is the R2 scores for 3 models on training and test dataset.



From the scores we can see Linear Regression has the lowest scores, Random Forest has highest scores and GBDT lies between them. There is a drop between training score and test score for the Random Forest classifier, indicating overfitting, but it still outperforms the other two models. Though having a lower score than Random Forest, GBDT is good at generalization, as there is little gap between training and testing.

Based on the score on test the set, the Random Forest is the best model in out task. Below is the top 10 important features in our model:



The most important feature is KillPlace, which is Ranking in match of number of enemy players killed. It is reasonable, as kill place shows the relative level between this player and other players in this game. It also shows that the most correlated feature isn’t necessarily the most important feature in a non-linear model.

1. Conclusion

In this project, we use a PUBG players’ statistic dataset to analyze and visualize features, detect cheaters with anomalies detection. We use 3 regression algorithms to predict win place. The best model is Random Forest.

References

1. T. Wijman and T. 2018, "Global Games Market Revenues 2018, Per Region & Segment, Newzoo", Newzoo, 2018. [Online]. Available: https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/.
2. "About OP.GG :: LoL Stats, Record Replay, Database, Guide, MMR - OP.GG", OP.GG Korea, 2018. [Online]. Available: http://www.op.gg/about/. [Accessed: 16- Dec- 2018].