Kaggle- PUBG Finish Placement Prediction

CS-580L Project

Bo Zhang
B00766142
bzhang73@binghamton.edu

Jie Ren
B00768706
jren18@binghamton.edu

Chentao Jiang
B00765706
cjiang21@binghamton.edu

1. Problem

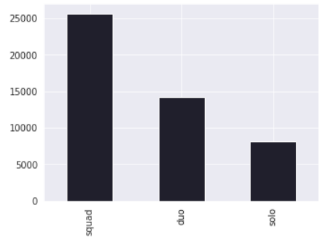
Our team focus on the PUBG finish placement prediction problem we found on Kaggle. We need to predict the possibility of every player to be the winner. PUBG is a player versus player shooter game which allows one hundred players hunt each other in a limited map, where players fight to remain the last alive. Player can choose to enter the match solo, duo, or with a small team of up to four team of up to four people. In either case, the last person or team left alive win the match. We have over 65,000 games' worth of anonymized player data, split into training and testing sets, and asked to predict final placement from final in-game stats and initial player ratings. We have over 29 attributes may contribute to the final solution such as the kills - Number of enemy players killed, rankPoints - Elo-like ranking of player and damageDealt - Total damage dealt. Using these attributes to predict who is the most likely to be the winner.

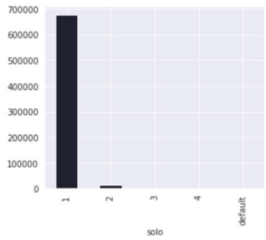
2. Dataset

We download the dataset from Kaggle. The dataset includes 47965 times of game. The total players in the whole dataset is 446965. The number of groups is 2026744. Each game has exactly data of 29 attributes such as the ID, Group ID and other attributes. It also shows the results of each player.

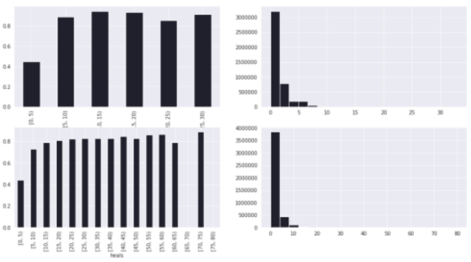
3. Data pre-processing & Choose features

We firstly clean up the whole dataset. Here in the data set, we sum all the teams by teammates numbers. We also delete some data with mistakes.

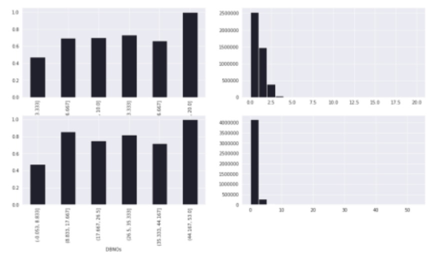




And then, we try to discover the relationship between the attributes and winning rate. For example, wo can see the healing graph. In this graph, you can see how the heal attribute can affect the final winning rate. The more players can heal themselves, the more possibility they may can be the final winner. We use different ranges and different diagrams to show the relationship between them. DBNOs is another attribute which given by the dataset.

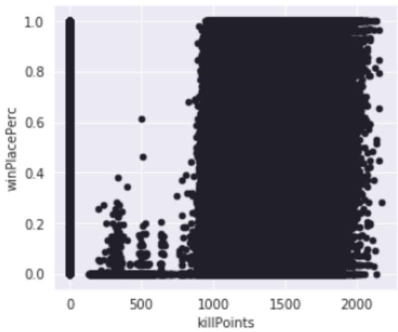


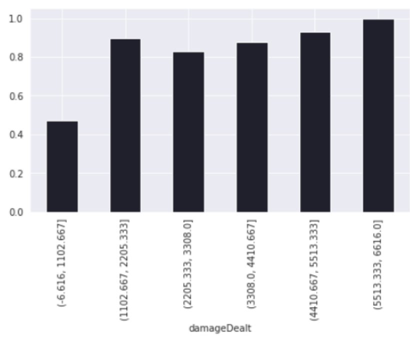
heal



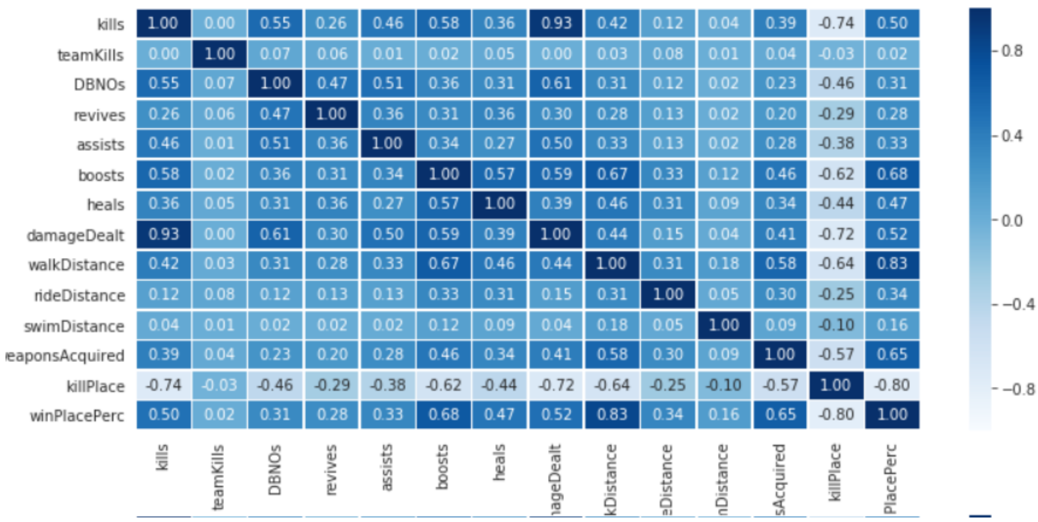
dbnos

Here, we discover many attributes to see the relationships such as kills, revives, team kills and many other attributes.





And then, we combine those attributes together to see the relationships between those attributes.

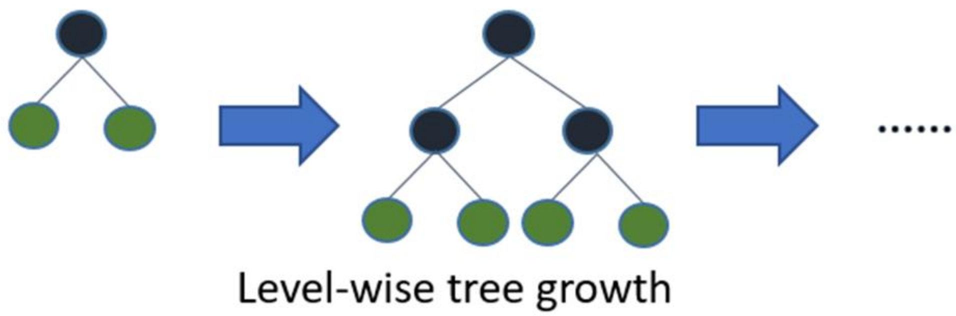


4. **Models**

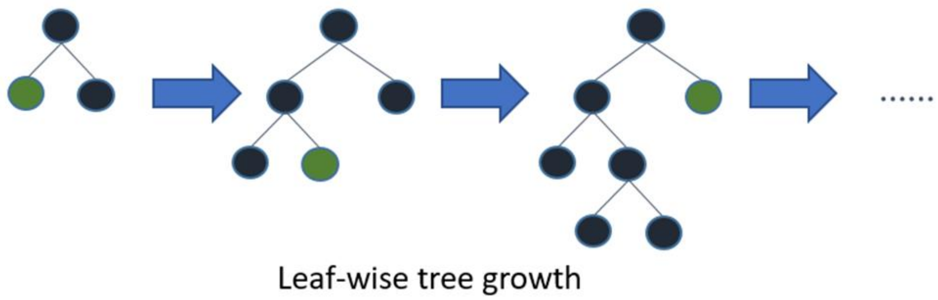
We use the light GBM model. Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking, classification and many other machine learning tasks.

Since it is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise. Therefore, when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms. Also, it is surprisingly very fast, hence the word 'Light'.

Before is a diagrammatic representation by the makers of the Light GBM to explain the difference clearly.



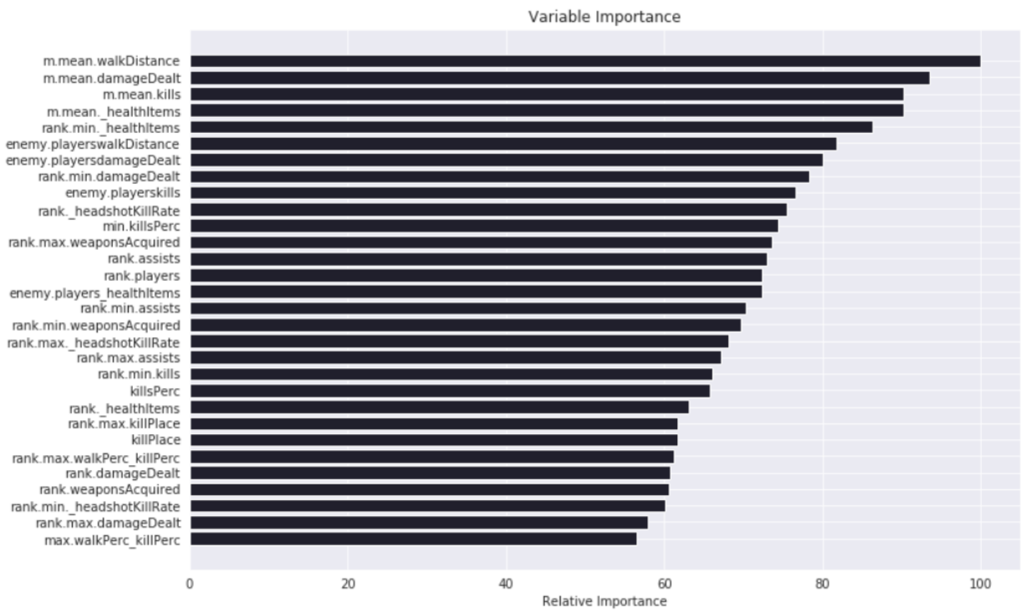
Level-wise tree growth in XGBOOST.



Leaf wise tree growth in Light GBM.

Leaf wise splits lead to increase in complexity and may lead to overfitting and it can be overcome by specifying another parameter max-depth which specifies the depth to which splitting will occur.

We also got the graph.



5. Results

Here is our result in the Kaggle. We go through the test in the Kaggle website.

Kaggle Notebook:

https://www.kaggle.com/bzhang0625/kernel73ecb80046

6. Teamwork

We divided our work into three parts.

Jie Ren: Data pre-processing, data cleaning and feature selection

Chentao Jiang: Data pre-processing, feature selection and testing

Bo Zhang: Feature generation, modeling and testing

7. Reference

https://www.kaggle.com/c/pubg-finish-placement-prediction/data

https://www.kaggle.com/bzhang0625/kernel73ecb80046/notebook