The Best Way to Win a Match of PUBG

JOSHUA RAMOS, University of Colorado at Boulder
TYLER MOORE, University of Colorado at Boulder

1 PROBLEM STATEMENT

Player Unknown Battlegrounds, better known as "PUBG", is an online multiplayer battle royal video game. The game starts with one hundred players, and the only way to win is to be the last man standing. There are many variables that may better your chances of winning. Those variables are drop location, team size, walking distance, and accuracy. We hope to discover what combinations of attributes will assist a player in ranking in a higher percentile by applying knowledge from this data mining class to the data we have collected. we hope to find what combinations will assist a player in ranking in a higher percentile.

We hope to find some interesting stats on combining unique datasets. For example, it would be interesting to find the correlation of walking certain distances and winning or things like if there is a sweet spot for team size that would increase your chances of winning. It's always interesting finding stats that many players don't even think about, and with a passion in data mining and video games we believe we'll find some useful information that is new and creative to improve community knowledge on PUBG.

2 LITERATURE SURVEY

Player Unknown Battlegrounds is a relatively new game with an initial release in 2017, so there is some research done but it hasn't been mined as much as we would've liked. From the Kaggle page and the dataset we are mining on we found two unique projects.

A user without his actual name listed but a screen name called Skihikingkevin. Skihikingkevin took the data set and created heatmaps of the final circle based on where the final people died and their coordinates. Skihikingkevin used python for his data processing using the libraries as listed matplotlib, numpy, pandas, seaborn and used my jupyter notebook to show exactly what he did. In conclusion he found, "Erangel: Although a lot more spread out, there are a few pockets of land that is more contested for final circle, primarily near very open areas between pochinki and mylta. It is worth noting that a lot of close river areas north are avoided so players based off of seeing the first circle, should be able to make an educated guess where the next circle will be.
Miramar: A lot different and more clustered. Notable clusters are between bendita and Leones, south of San Martin and west of Pecado. It is interesting to see that a large portion on the of the map is often never in the final circle" With this information it can assist a player in jumping in a location where they won't have to move as much to stay in the circle.

Another user named "Christoph Egerland" Mined the same dataset we will be using and his focus was on making maps of where players jumped the most. The tools he used were as follows python, matplotlib, numpy, pandas, seaborn and used my jupyter notebook to show exactly what he was doing. He focused on finding were the first kills were to find out where exactly the players were landing and using their coordinates to make a heat map. The heat map is useful because it can tell players where the best place to be in the final circle.

3 PROPOSED WORK

Using these datasets, we can hopefully find multiple answers to the questions we want to figure out. Before we actually begin mining, we need to make some alterations to our data so that we can use it.

The first problem we need to solve involves **cleaning our data** so that there are no NULL values. If we do not fix this problem, then we will either get errors or our answers will be skewed. The reason there are NULL values is because sometimes a player will enter the match and then leave early. When this happens, the game will enter NULL values into the data. To fix this problem, we will just need to replace all the NULL values with 0's to signify they left the match early.

Another thing we need to do before actually mining is **normalizing** our data. For example, the rules of the game involve having a team of up to 4. However, some teams might only have 3 people in their group. If we decide to look at the amount of kills a certain team has, it would not be fair to compare a team of 4 to a team of 3. This is why we need to normalize these numbers in a way that makes it fair. A solution would be to look at the kills per player on a certain team and see if a team of 4 has a higher ratio than a team of 3, or vice versa. Also, if we are going to look at team stats, one thing we will have to do is make a whole new dataset with just team information on it. Right now, our current datasets only have individual player information, including their team ID. So we can make a new dataset that finds all the unique team ID's and put their data in it.

Once we have all the datasets we need and all the data normalized, it will be time to start mining it to hopefully find answers to our questions. One of the first things we will be doing involves clustering the data. A lot of our questions involve finding patterns and one of the best ways to do that is from clustering. We plan to put all the players together who have the same type of stats, such as: same distance traveled, same amount of kills, same amount of players on a team. Once this data is all sorted out, we will sort through it to see who has the highest win percentage to hopefully figure out the best strategy to win the match.

How it is Different Than Previous Work: Our project is much different than other people's previous work because there's mostly identifies heat maps of where people started out and where they died. Ours goes more in depth because we want to know the strategy behind this game, like if a bigger team increases the chance to win or if hiding the whole game will make you less likely to die. We will have to integrate the datasets way more than just finding out where people died.

4 DATA SET

When we downloaded the datasets, we realized there are multiple attributes to look at and evaluate. In one of the datasets, it has all the information on deaths. This involves who the killer was, their position, how much time has gone by, the victims name, the victims position, etc. This dataset has over 65 million entries. There is also another dataset based purely on a specific person for an individual match, including how far they traveled, how many kills they had, the amount of time survived, their individual and team placement. Every single attribute in both tables involves integers, except for the match ID which is just a random string value.

URL : https://www.kaggle.com/skihikingkevin/pubg-match-deaths/

5 EVALUATION METHODS

The two main evaluation methods we will be using in the beginning will be clustering and association. One of the first things we need to do is put similar players together, such as number of kills or distance traveled. We currently have 65 million death entries to look at, so by grouping similar players together, we can break up our dataset into different categories and test certain ones specifically. This can significantly speed up our algorithms because we will be testing groups of players, not each one individually.

We will also be using association to figure out certain patterns. For example, do players who travel more during the match also get more kills? This will be very helpful in figuring out the best way to win a match because hopefully we can find a correlation between certain attributes that help a player increase their chances to win.

After we have found certain patterns among the dataset, the only thing left to do is to apply this knowledge and figure out if it works. One way to do that is to actually play the game and follow the patterns to see if this increases our winning percentage. Another way is to use conditional probability. We can delete all the players who do not meet the requirements we found, and then see if the remaining players have a higher winning percentage than the general group.

6 TOOLS

- Python: A general purpose programming language that is great with manipulating big datasets

-MatPlotLib: A plotting library that can produce powerful graphs for displaying dataset information
-Numpy: A library for complex matrices and array operations.
-Pandas: A library for reshaping and viewing datasets in a variety of ways.
-Seaborn: A library based on MatPlotLib that can create heat maps to better visualize our dataset

7 MILESTONES

- Data Preprocessing (removing NULL values and normalizing*) : March 12th, 2018
- Data Integration (creating new team dataset) : March 19th, 2018
- Simple Techniques : April 2nd, 2018
    - Clustering (break up dataset into different categories)
    - Association (find correlations between two or more items)
    - Other techniques we will learn later
- Back testing (once patterns are formed, look back at the data to see if our hypothesis was correct) : April 16th, 2018
- Get everything ~~formalized~~ for final presentation : May 1st, 2018

8 MILESTONES COMPLETED

We have spent a lot of time trying to learn our dataset. Data was separated into two different folders: aggregate data and deaths. Data analysis was complicated because each folder consisted of 5 csv files, and we had to interpret what made the files different.

On Keggle, there was little information as to why the files were laid out as they were, so we had to spend time opening files and comparing them. In conclusion, the data set consisted of 64 million data points. Each of the 5 CSV files split up the information to include roughly 13 million data points in each file. This course only asked that we use a dataset with more than a million data points. Therefore, we decided to only use one of the files from the aggregate set and one from the kills data.

The next problem we had was that the kills data set didn't match the length of the aggregate dataset. None of the partitioned kills datasets matched the length of any of the aggregate dataset. After comparing the datasets, we learned that kills don't include the players that won. For example, if a team of 4 players had zero deaths in the team, then none of those players would be included in the kills dataset. This explained the variable length and solved this problem.

Next, we prepared the data to be formatted in a new dataset of information that we found important. We took a sample data set of one match which was easy to do because the aggregate data laid the data out sorted by matchID. Then we used the first match in the dataset and put it into excel. Then we copied all the data to a new sample file ready to be manipulated.

The sample data was a third person match of duos with a party size of 75.* We determined our new data set would only include team_id, game_size, match_id, party_size, team_dist_ride, team_dist_walk, team_kills, and team_placement.
Team_id is a unique identification number given to a team with a unique match_ID. Game_size was included in the original file and measures how many players are in a game. Party_size shows how many players were in a particular teams party. Team _dist_ride is the average distance each team traveled.. We included this because we wanted to see how a team's unit of travel determined their placement. A summation would not be the best way to determine this because because some teams might have a player that barely walked and another player that walked an extreme distance. We chose to do mean because with small data sets of like 4 to 2 the mean would end up taking the mean.

We then began to look at values we needed to normalize. For instance, team kills compare to party size. If a team is playing 4 man squads but they only have a party size of 2, then we must normalize their kills and create some kill ratio. By doing this, we can extrapolate the data necessary to determine what distance and kill combinations will most likely result in a win. Players that cover longer distances and have higher kills tend to have more aggressive play styles than the more passive slower moving lower kill players.

9 MILESTONES TODO

Our group has done most of the work for preprocessing. We currently have three different datasets: the stats on the individual player, the information on each kill in a given match, and the integrated set which includes all the team information. This is extremely important for our team's project because one of our questions we asked asks if there is an advantage related to team size. For example, we would like to determine if a team of 4 has a higher chance to win than a team of 3. This can create some problems. When we return to the previous example, we see that if we are looking at total team kills, a size of 4 would have a higher advantage than a team of 3 because it has one more player. In response, we have to normalize our data so that we can compare different sizes of teams and decide which one is better equipped to win. To do this, we must determine the kills per player on a given team. This statistic is found by dividing the total team's kills by the number of players on the team. Once we accomplish this, it will open up many possibilities for us for the next step of the project. Following data normalization our project will mostly consist of data mining.

After normalization, the next milestone we need to complete is simple techniques for mining. The first thing we will incorporate is clustering. Another question we have is, does the distance traveled affect a players chance of winning? One thing we will cluster together is distance traveled. We will use the k-means algorithm to calculate the best distance to travel to win, because it is the perfect strategy to tackling a question like this. Once we have clustered the data, we will look at each individual cluster to identify similarities in the data sets. We will especially evaluate which cluster has the most amount of wins associated with it. This will quickly tell us if traveling less or more during a match creates a higher chance to win.

We will also use association as a technique to evaluate our data. One example we will incorporate with this technique is using our team's data to see if a certain team size creates a higher probability to win. With association, we can create support and confidence levels to test different situations.

After we have completed the techniques to find the information we needed, it will be time to backtest it. This means we can test two of our data sets against one another. For example, we can compare distance traveled versus number of kills for a team. From there, we can see which one converts to a higher winning percentage. In addition, can see if combining the two data sets creates an even better percentage, rather than just doing one alone.

10 RESULTS SO FAR

Our results thus far can be found in Table A, which was created for our sample data.We describe each attribute of our data in the milestone portion.

From this dataset, we determined there is a positive/negative correlation between the distance walked and the amount of kills. This combination was associated with higher placement on the leaderboard.

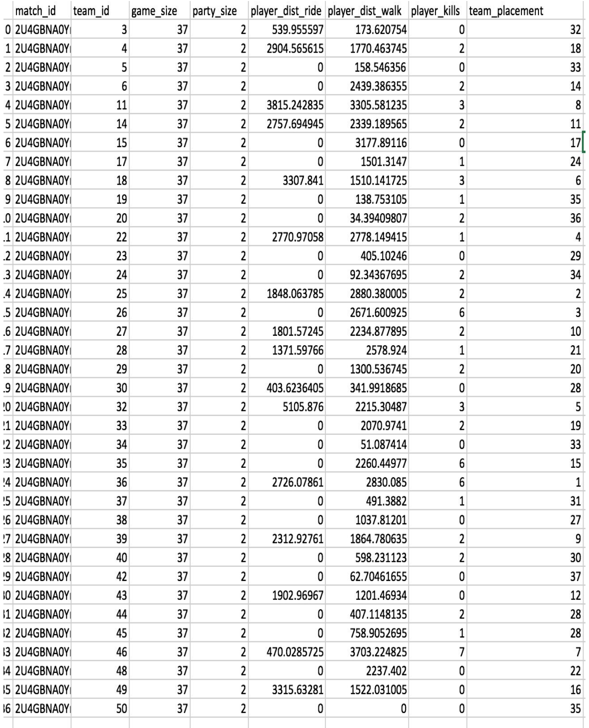
We were also able to determine that a team who rides further distances in a car tends to finish higher up in the standings. This makes sense because cars travel faster and can survive more damage. This intuitively gives the players driving them a bigger advantage, and was proven by one of our smaller data sets. In contrast, we found that the cars do not directly correlate with winning. Even though it helps players get far in the game, it does not necessarily mean they will win. In addition, we found that most of the time, the winning team did not even come close to traveling the same distance to the team with the most distance traveled by car.

In hindsight, we realized that it might be important to include the time each player played. This could be important so we can get the amount of distance traveled over time and could calculate the players average speed. We believe this would be a better measure of how much a player is actually moving during the game.

With the current results, we've also noticed we need to remove players whose data shows no movement during the game. This would indicate that the player was away from their keyboard, and they shouldn't be included in our future data files and marked to be normalized.

We don't have have all the results we'd wish to have at this time, but we do have the code to make the new larger csv file into another dataset. This code will streamline our future endeavors and allow us to manipulate the data we want more quickly. In conclusion, we are pleased with the progress we have made thus far and feel this our work will help us to continue moving toward a more conclusive way to increase our final rankings in PUBG.

Table A



	match_id	team_id	game_size	party_size	player_dist_ride	player_dist_walk	player_kills	team_placement
0	2U4GBNA0Y	3	37	2	539.955597	173.620754	0	32
1	2U4GBNA0Y	4	37	2	2904.565615	1770.463745	2	18
2	2U4GBNA0Y	5	37	2	0	158.546356	0	33
3	2U4GBNA0Y	6	37	2	0	2439.386355	2	14
4	2U4GBNA0Y	11	37	2	3815.242835	3305.581235	3	8
5	2U4GBNA0Y	14	37	2	2757.694945	2339.189565	2	11
6	2U4GBNA0Y	15	37	2	0	3177.89116	0	17
7	2U4GBNA0Y	17	37	2	0	1501.3147	1	24
8	2U4GBNA0Y	18	37	2	3307.841	1510.141725	3	6
9	2U4GBNA0Y	19	37	2	0	138.753105	1	35
.0	2U4GBNA0Y	20	37	2	0	34.39409807	2	36
.1	2U4GBNA0Y	22	37	2	2770.97058	2778.149415	1	4
.2	2U4GBNA0Y	23	37	2	0	405.10246	0	29
.3	2U4GBNA0Y	24	37	2	0	92.34367695	2	34
.4	2U4GBNA0Y	25	37	2	1848.063785	2880.380005	2	2
.5	2U4GBNA0Y	26	37	2	0	2671.600925	6	3
.6	2U4GBNA0Y	27	37	2	1801.57245	2234.877895	2	10
.7	2U4GBNA0Y	28	37	2	1371.59766	2578.924	1	21
.8	2U4GBNA0Y	29	37	2	0	1300.536745	2	20
.9	2U4GBNA0Y	30	37	2	403.6236405	341.9918685	0	28
20	2U4GBNA0Y	32	37	2	5105.876	2215.30487	3	5
21	2U4GBNA0Y	33	37	2	0	2070.9741	2	19
22	2U4GBNA0Y	34	37	2	0	51.087414	0	33
23	2U4GBNA0Y	35	37	2	0	2260.44977	6	15
24	2U4GBNA0Y	36	37	2	2726.07861	2830.085	6	1
25	2U4GBNA0Y	37	37	2	0	491.3882	1	31
26	2U4GBNA0Y	38	37	2	0	1037.81201	0	27
27	2U4GBNA0Y	39	37	2	2312.92761	1864.780635	2	9
28	2U4GBNA0Y	40	37	2	0	598.231123	2	30
29	2U4GBNA0Y	42	37	2	0	62.70461655	0	37
30	2U4GBNA0Y	43	37	2	1902.96967	1201.46934	0	12
31	2U4GBNA0Y	44	37	2	0	407.1148135	2	28
32	2U4GBNA0Y	45	37	2	0	758.9052695	1	28
33	2U4GBNA0Y	46	37	2	470.0285725	3703.224825	7	7
34	2U4GBNA0Y	48	37	2	0	2237.402	0	22
35	2U4GBNA0Y	49	37	2	3315.63281	1522.031005	0	16
36	2U4GBNA0Y	50	37	2	0	0	0	35

Egerland, Christoph. "PUBG Data Analysis." PUBG Data Analysis | Kaggle, www.kaggle.com/chegerland/pubg-data-analysis.

KP. *Final Circle Heat Map*. www.kaggle.com/skihikingkevin/final-circle-heatmap.