Capstone Project 2: PUBG Finish Placement Predictions

Overview:

PUBG is a game where up to 100 players start in match onto an island empty handed and must explore, scavenge and eliminate other players in a free for all until only one player/team is left standing. Kaggle has given us 65,000 games' worth of anonymized data and we're asked to predict the final placement in a game given the in-game stats and initial player ratings. Kaggle has asked that we evaluate our model on Mean Absolute Error between the predicted win placement and the observed win placement.

Target Audience:

Within the last 2 years, Battle Royale (BR) styled video games have exploded in popularity, giving us new IPs such as Fortnite and Realm Royale and even affecting long standing franchises like Black Ops. This surge in numbers can arguably be traced back to PUBG, one of the first BR games to receive a substantial amount of development and sustain a large, consistent player pool. The premise of the game is simple: You and up to total of 100 players are dropped onto a large island empty handed and you must explore, scavenge and eliminate other players in a free-for-all/team vs team first person shooter styled battle until only one player or team is left standing. This project aims to use 65,000 games' worth of anonymized data to predict final placements in a game using over 20 different in-game stats. There are a couple groups of people that would be interested in the results:

- The game's developers: The data analyzed could be used to help conduct balance changes based off of impact of each feature.
- The players: The data analyzed could be used to help pinpoint weaknesses in a player's performance, giving them an objective avenue of improvement to go through.
- Tournament organizers: High impact features could be used when casting tournament games, being able to select the top 5 or 10 most impactful stats that players should keep an eye on.

Data Description:

The data I will be using can be found on Kaggle (here). The file list includes:

- A sample submission file in the correct format
- A training dataset (4.45m rows)
- A test dataset (1.93m rows)

The columns found in our data are as follows:

- DBNOs
- Assists
- Boosts
- damageDealt
- headshotKills
- Heals

- Id
- killPlace
- killPoints
- killStreaks
- Kills
- longestKill
- matchDuration
- matchId
- matchType
- rankPoints
- Revives
- rideDistance
- roadKills
- swimDistance
- teamKills
- vehicleDestroys
- walkDistance
- weaponsAcquired
- winPoints
- groupId
- numGroups
- maxPlace
- winPlacePerc - This is our prediction target

Data Initial Inspection, Cleaning and Wrangling:

After the initial importing of our training and test files into dataframes, the first step I took was to inspect the different game modes available in our data set. Looking at the list, I've split these game modes up as follows:

- Standard Game Modes:
  - Solo Game Modes:
    - solo
    - solo-fpp
  - Multiplayer Game Modes:
    - squad
    - duo
    - squad-fpp
    - duo-fpp
- Non-Standard Game Modes:
  - normal-squad-fpp
  - crashfpp
  - flaretpp
  - normal-solo-fpp
  - flarefpp
  - normal-duo-fpp
  - normal-duo
  - normal-squad
  - crashtpp
  - normal-solo

For reference, Solo games are games where the player is matched solo against up to 100 other solo players. Duo game modes are games where the player and a friend is matched with other partnered teams with a max player count of up to 100 players per game. Squad game modes are games where the player and up to 3 teammates are matched with other similar groups with a max player count of up to 100 players per game. As for FPP vs Non-FPP game types, FPP modes are modes where the players are forced into a first person shooter perspective. However, the remaining mechanics are exactly the same. With these categories in place, the statistics of the number of observations/games for each of the game modes are as follows:

- Game mode : solo, Number of observations - 181943, Number of games - 2297
- Game mode : solo-fpp, Number of observations - 536762, Number of games - 5679
- Game mode : duo, Number of observations - 313591, Number of games - 3356
- Game mode : duo-fpp, Number of observations - 996691, Number of games - 10620
- Game mode : squad, Number of observations - 626526, Number of games - 6658
- Game mode : squad-fpp, Number of observations - 1756186, Number of games - 18576

<INSERT NON STANDARD STATS HERE>

EDA (Visualizations and Inferential Statistics):

Following this, the training data was then split into separate dataframes based on the game types (with the non-standard game modes combined into one dataframe). The following procedure was then applied to each of these dataframes:

1. The .describe method was used to take a look at the general base statistics of the dataframe.
2. The .info method was used to confirm that there are no null values in the dataframe.
3. Unnecessary columns were dropped (“DBNOs”/”revives” for solo game mode types).
4. A feature list was populated.
5. A distribution plot was made for each of the features in the feature list.
6. A pearson correlation heatmap was generated.
7. A clustermap was generated.

After completing these, some general conclusions about the datasets are as follows:

- Applicable to all game mode types:
    - For the most part, the statistical distributions plotted all follow an expected non-normal distribution. From these distributions, we do see instances of outliers in our data. My opinion on how these data points came about can be attributed to hacking programs that are available online for PUBG. This gives players an advantage in game that would explain how certain players have boosted values specifically in the “damageDealt”, “headshotKills”, “heals”, “kills”, “killStreaks”, “longestKill”, “rideDistance”, “weaponsAcquired” and “winPoints” features. However, as these instances total for less than .1% of the total number of observations as well as it being relatively difficult to decide an arbitrary line at

which point to remove these outliers, these rows of data have been left in. Also, no null values were found in the data set (I believe Kaggle did an initial cleaning of the data before it was uploaded for use in the competition) and so no extra steps to clean the data were needed.
    - Looking at the heatmaps and clustermaps of all the different game modes, we see the number one most correlated value with winning placement is the distance walked by the player. This naturally makes sense: Walking distance is a function of how long the game goes on (the longer the game is the more the player needs to move to keep up with the decreasing area of play), and players that score high winning placements will necessarily have more time played in a given match.
- Solo Game Modes:
    - Along with the walking distance metric, other highly correlated values with the winning placement are:
        - Boosts
        - Kills (Kill Placement, Kill Streaks, Damage Dealt, Longest Kill Streak)
        - Weapons Acquired.
- Duo Game Modes:
    - Along with the walking distance metric, other highly correlated values with the winning placement are:
        - Boosts
        - Kills (Kill Placement, Kill Streaks, Damage Dealt, Longest Kill Streak),
        - DBNOs
        - Heals
        - Weapons Acquired.
- Squad Game Modes:
    - Along with the walking distance metric, other highly correlated values with the winning placement are:
        - Boosts
        - Assists
        - Kills (Kill Placement, Kill Streaks, Damage Dealt, Longest Kill Streak)
        - Revives
        - DBNOs
        - Heals
        - Weapons Acquired

From here, in order to determine how to progress with the machine learning algorithms, I needed to compare and test for any statistically significant differences between the feature distributions for the different game modes. I conducted two comparisons, Non-FPP and FPP modes comparison (choosing the solo game modes data sets to test for this) and solo/duo/squad matches comparison. As done previously with our EDA, for each comparison I made distribution plots of

each of the features for each game mode being tested but instead this time overlaid on top of each other. To confirm my results, a Mann-Witney U test was performed as I am working with non-normal distributions. With these comparisons done, we see no statistically significant differences between any of the standard game types. This is important, as this allows us to combined all the data for standard game modes in order to build a highly specific model.