MLND: Capstone Proposal

Zach Freedman
January 16, 2018

Domain Background

Overwatch is a popular first person shooter game available for PC, Xbox, and PS4. In competitive Overwatch, a matchmaker pits 2 teams of 6 similarly skilled players against one another in objective based games.

Competitive Overwatch has 1 public ranking system called Skill Rating (SR), splitting players into 1 of 7 ranks based on game skill:
* Bronze (1-1499)
* Silver (1500-1999)
* Gold (2000-2499)
* Platinum (2500-2999)
* Diamond (3000-3499)
* Master (3500-3999)
* Grandmaster (4000-5000)

Similar Work

Note: There is currently no public similar work to this problem, but my reviewer said I needed to include something. This is the closest "similar" work to the problem I'm trying to solve, but it's irrelevant and, based on the abstract, it sounds like the Stanford group pivoted away from MMR toward the public LoL ranking system. I'm not going to include this in my final work. I will not provide an incoming link for this paper, and if Udacity is not going to let me solve a *new* problem as my capstone project because there is currently no "similar" research, then the company is missing the point of its own course.

There is a lot of work being done with machine learning in sports and eSports. However, there is not much being done publicly on what separates professional physical athletes (NFL, NHL, NBA) from those at the amateur levels with respect to performance measurements. Public studies in eSports with respect to machine learning and its applications for rank separation are also sparse, but a study done at Stanford attempted to tackle a similar problem within League of Legends by predicting end of season rank based on current performance. That research is here.

Problem Statement

It's well understood that winning games increases SR while losing games decreases the rating. However, what isn't known is what/how performance statistics (eliminations, damage blocked, healing) separate *rank A* players from those in *rank B* for specific roles and characters. What separations like "Bronze rank players average 10% weapon accuracy whereas Silver rank players average 20%" exist? This project will use machine learning to analyze character performance statistics in an attempt to discover what separates PC Overwatch players at the 7 different SR classes described above.

Datasets and Inputs

Overwatch player performance is recorded and published by Blizzard Entertainment. The representative statistics are broken down on a per character basis and are updated roughly after each game played. It's important to note that although all characters share some common performance statistics, data exists that is significant only to specific characters. For example, a character like Hanzo who has only damaging abilities has no relevant information about healing.

Currently, this data can only be obtained via web scraping. Given a player_region, player_name, and player_id combination, player statistics can be found at

playoverwatch.com/en-us/career/pc/player_region/player_name-player_id.

Player battletags (player name and player ID) can be obtained from Overwatch Tracker, which maintains an SR leaderboard for the current competitive season. Roughly speaking, there are 100-200k competitive player profiles available. Each player can have a hero performance for at most 26 heroes (assuming a 27th hero is not released by the time this project begins scraping). However, as discussed later in the **Faker Removal** section, most players will only have valid hero entries for a select handful of heroes (likely 1-5 heroes). Any given database entry will correspond to the statistics for a battletag's data for a specific hero (Johnny#12345's Roadhog), where all of the features with the exception of SR class (the target) will be numerical.

With respect to the dataset distribution, the SR distribution for Overwatch is *currently* unknown. However, there are heroes in "the meta" which are picked much more frequently than others. Likewise, there are very niche characters like Mei, Torbjorn, Symmetra, and Sombra which are picked infrequently at both ends of the SR spectrum. In order to ensure that meta character statistics do not mask the performance output for off-meta characters, a database will be created for each of the current 26 heroes. This choice will benefit learning in the long run because similar characters like McCree and Soldier: 76 have unique features which could provide critical information with respect to SR evaluation. Given the current unknowns about how long the data will take to scrape, each hero database can have anywhere from 1-25k entries.

Solution Statement

This describes a multiclass supervised learning problem. Players have a set of performance statistics for each character (features) that associate them with their current SR rank (class). Training and testing machine learning classifiers (Naive Bayes, Decision Trees, Logistic Regression, SVMs) should unveil a class separation spanning Bronze to Grandmaster emphasizing significant intra-SR feature differences.

Benchmark Model

Decision trees are easy to visualize machine learning models that can scale well with high dimensional data. Therefore, as benchmark models, decision trees will be trained on a total of 20 entries for each of the 7 SR classes for each hero, meaning 26 benchmarks with 140 training elements each. Each benchmark can be tested with 5 entries from each SR range, translating to 26 benchmark test sets of 35 elements each. This gives an 80:20 train/test split. By having a benchmark of this size, proof of concept can be quickly established. However, if scraping takes longer than expected, the benchmark can be abbreviated to analyze one frequently picked character like Tracer, Genji, or Mercy instead of all 26 heroes while providing a similar level of confidence.

Additionally, it's important to note that the benchmarks are disregarding SR distribution of the scraped data.

Evaluation Metrics

The evaluation metric of choice here will be accuracy. After training classifiers, classification performance for each SR class can easily be calculated. Given some arbitrary SR class (Bronze, Silver, ...) *x*, true positive and true negative predictions TP_x and TN_x, and the size of the dataset D given by |D|,

$$acc_x = (TP_x + TN_x)/|D|.$$

The above calculation states that the accuracy for a class *x* can be calculated by summing the total number of players correctly classified at SR level *x* with the total number of players correctly classified not at SR level *x* divided by the total number of players in the dataset. With this accuracy calculation, a classifier can be evaluated on how well it separates each of the 7 SR ranks from one another.

However, as described previously, the SR distribution of the dataset, for any hero, is currently unknown. What if 95% of Pharah players are Platinum rank, and so a decision tree always outputs the SR class as Platinum? The accuracy metric will be great, but the classifier would never learn anything about Bronze or Grandmaster players. Using a balanced F1, the precision and recall of the problem can be taken into account when evaluating learners. The balanced F1 score is given by

$$F = 2PR/(P + R)$$

where P and R are precision and recall, respectively.

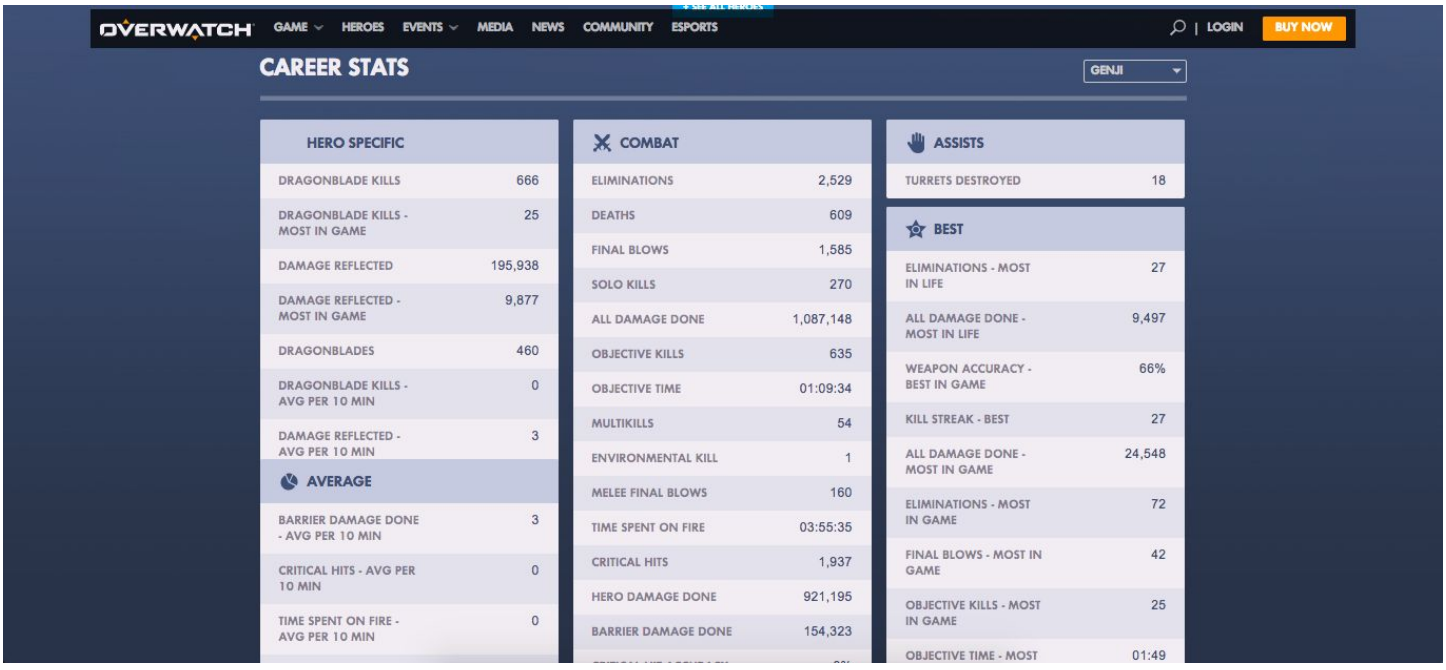
Project Design

In order to answer the proposed question, the following sequential steps will need to be taken.

Data Scraping

The required data isn't already available as a public dataset. To obtain player performance statistics, Overwatch Tracker provides an ordered leaderboard for PC players during the current competitive season (during this report competitive Overwatch is in Season 8).

However, Overwatch Tracker does not provide as detailed a statistics list as PlayOverwatch, the official Blizzard website for Overwatch information. What Overwatch Tracker does provide is an ordered listing of all competitive Overwatch players with their battletag. In the above image, the rank 1 player goes by the name Architect with a player ID of 31375. With this information, players can then be found on PlayOverwatch using their battletag to find their detailed performance statistics. The Architect#31375 player profile information for the player's most played character Genji is depicted below.



In order to collect the large dataset required for training/testing, the data scraper will be written to crawl Overwatch Tracker to collect valid and unique battle tags for players competing in competitive Overwatch Season 8. Then, with this list of battle tags, another process can be written to crawl PlayOverwatch for each specific battletag to find detailed hero performance information. This data can be written to a .csv file and later loaded into a Pandas dataframe.

Data Preprocessing

Faker Removal

Some of the player profiles collected during database assembly will be thrown away. Previous season SR ranking has a substantial influence on the following season's SR placement (initial ranking), and because of this players may place at an incorrect ranking relative to the heroes they *currently* play. To provide an example, take player *M* who mained (played most) Mercy (support/healer) in Season 7. Mercy does not require relatively impressive mechanical skill (skill involving precision with your mouse and keyboard, mouse specifically) relative to a character like Widowmaker (DPS/damage) which requires arguably one of the highest mechanical skill requirements out of the Overwatch character pool. Now also suppose *M* climbed to the rank of Grandmaster with an SR of 4000 while playing 20 hours of Mercy and 1 hour of Widowmaker during Season 7. During Season 8 placements, if *M* plays only Widowmaker and finishes their placement matches, it is possible (note this is currently speculated by the player base) that *M* places anywhere in the range [3500, 4500] SR, with larger bias towards the lower half of the range.

After placements *M* would have collected roughly 3-4 hours of Widowmaker-only gameplay for Season 8. But, it is possible that *M* is now playing a character at a skill level lower than their current SR rating. Given that the competitive system functions as Blizzard expects, if *M* continues to play Widowmaker, they will eventually fall or rise and stabilize to an SR where they play Widowmaker at a skill level equal to the skill of the other players in their matches. With this hypothesis, and note this is based on suspicion, this phenomenon, if it exists, can be averted by pruning the dataset of players with less than a specified number of hours of playtime on a character. The code would look something like:

```
T = 10
for p in players
    for h in p.heroes
        if h.hours < T
            delete h
```

Feature Scaling

It's common practice in machine learning to scale features with different ranges to fit the same range in order to improve training/learning. For example, features like damage with a real range of all [0, Infinity] and weapon accuracy with a real range of [0, 100] can all be scaled down to the range of [0, 1]. This project will implement such feature scaling.

Character Database Separation

Not all features are shared across all characters. For example, most DPS characters like McCree and Doomfist do not have the ability to heal, whereas support characters like Lucio and Ana are responsible for healing their teammates. Additionally, characters within the same role do not share an identical featureset (heroes have different abilities). Noting this, there's likely training benefits in splitting up the database into 26 hero specific databases. However, note that in the event where a player has a database entry for different characters *after* the **Faker Removal** step, each of those entries would exist in their respective databases. This means that if player *M* plays both Mercy and Zenyatta, both being support characters, the support database would have an entry for M_battletag-Mercy and M_battletag-Zenyatta.

Principal Component Analysis

Because the dimensionality of the data is sizable, PCA will be implemented after feature scaling in order to avoid the curse of dimensionality. The initial goal is to collect the first *N* principal components that explain 90% of the variance.

Supervised Learning Choice and Training

Despite PCA being used during data preprocessing, it may be the case that a feature set with more explained variance (more principal components) is required for high test accuracy. However, it may also be the case that the principal component set that explains 90% variance has 10+ principal components. Therefore, a supervised learner that handles high dimensional data will be important. To start, because their relatively quick to train and easy to visualize, a decision tree classifier will be implemented. Assuming the decision tree classifier provides accurate classification, other learners will be implemented and hyperparameters can be adjusted/compared. However, it's important to note that, given the size of the expected dataset, support vector machines will be avoided (unless necessary for exploratory purposes) due to their long processing time.

Testing and Results

After several learners have been trained, their results can be measured using the accuracy and F1 metrics described above. The expectation here is to see several learners identify similar feature sets which they regard as significant in classifying players with respect to their SR. Furthermore, because the gap between players at neighboring SR ranks (Silver-Gold, Diamond-Master) averages to 500 SR points out of the possible 5000, learners should identify relatively significant gaps between learned feature values with respect to these neighboring SR levels.