

Balance in Everything:

Jarvin Bayona

In video games, design can be extremely tricky. In order to make an engaging game, certain parts should be easy, and others difficult to make challenge. If you make a game too easy, you might as well be doing something else instead. If you make a game too difficult, you might just quit because running into a wall doesn't make for much fun. This phenomenon is extremely evident in multiplayer games. How do you make sure that challenge and easy is attainable for all players, especially when they face each other?

I am going to tackle this problem through the lens of the particular game 'Player Unknown's Battlegrounds (PUBG).' The first thing I will do is classify which player(s) will win the match. After having an accurate model that can predict player likelihood of winning, then I will look at the features that they most often are committed to using. If I see a particular pattern pop up, that may be a flag for a feature that needs balancing. For example, if I see that a majority of people who win are using a particular weapon type, maybe some aspects of that weapon type should be adjusted to allow for fairer gameplay. Or if I see that players who lose a lot tend to go to an area of the map and then die quickly, perhaps add something to the surrounding area to make it more difficult to die.

The data I have obtained is from Kaggle. It is almost 20gb of data with plenty of features for me to rife through. Some features are player death location, killer location, weapon used, and time in game. I will probably use logistic regression to help with the classification process. As a metric, I want to maximize recall. This will make sure that my model gets as many true positives and not be penalized by false positives (people who actually win the match.)

With these adjustments, players will want to keep playing the game. Player retention will stay high. It can also influence how newer content can be implemented. If developers know these roadblocks that currently exist, they can make sure not to add them in the next iteration of the game.