APPLYING M.L. TO E-SPORTS GAMBLING IN CS:GO

Alec Warren, Noh Woldeyesus, Roberto Moreno, Tyler Nguyen

Conclusion from Results

- Best classifier(s) and why - any drawbacks?
- Best regressor(s) and why - any drawbacks?

Future Work & Improvements

Despite our models' performing well, and even exceeding expectations, there are still many improvements that could be made given more time. There was unfortunately insufficient time remaining to implement these, due to needing to dedicate so much time and energy to processing and aggregating our data such that it could be used to predict both past and future matches. And thus, one major 'feature' in our dataset that was not leveraged in the models is the actual map that each game is played on. Adding the ability to leverage this categorical data would definitely be the single most impactful improvement that could be done to the models in future versions of this project. Additionally, the models currently use stats from all players as features, but this could be changed such that only the players that impact the model in a significant way (i.e. the worst players on each team) are have their stats used as features.

RESULTS

Web Crawling & Scraping

For the source of our data, we have chosen to leverage HLTV's website since they have an unfathomable wealth of data and statistics, dating all the way back to CS:GO's inception. The next step was to build a custom web scraper that would allow us to gather and tailor recent data to use as testing data for our prediction models. Our initial EDA required 3 different web scraper python scripts, as well as one jupyter notebook file which used Pandas to combine the data scraped by the individual scapers into one comprehensive dataset (matches_dataset.csv).

EDA

The a significant portion of the necessary preparing of the data for EDA was performed during the collection of data via the web scrapers. Null values had to be handled properly, and many variables/columns were incorrectly typed or had mixed types. This was all extremely easy to do, however, thanks to the manner in which we scraped and stored the data for our dataset.

Processing Data for the Models

What makes our dataset unique from most normal datasets, is that we are using our model to predict matches, such that the model will has to predict the outcome of the game based on recent performance statistics for each of the two teams. This is to say, that we are not simply feeding a model the stats for a game that has already occurred and then asking it tell us who it thinks won this game, since that would be extremely easy to do based on which team had the better stats for the game and the accuracy for that model would likely exceed 95%. Instead, we have to create "profile" for each time over a given time period and aggregate the stats from all games they played in this time period's dataset. And so, this method of training and testing the data involved partitioning the data temporally into distinct 6-month chunks, and then creating the aggregated "profile" for each time during this time period, such that the performance of teams and individual players would be Individual player data spans the entirety of the period captured, effectively measuring a player's historical performance and using that as a predictor of the match outcome.

Training & Tuning The Models

Originally, three different classification models were explored: K-Nearest Neighbors, Decision Trees, and Random Forest, for both classification and regression. And then later, a Gaussian Naive Bayes Classifier and an SVM Classifier were implemented for the winner predictor task, followed by implementing an SVM Regressor for the round win difference predictor task. After doing initial tests of the models for efficacy, as well as compatibility with the dataset, the models were then put through the wringer: Using the sklearn GridSearchCV function, each model went through hyper-parameter tuning on 9 distinct 6-month-long partitions of data - each with roughly 5-7 thousand matches worth of data. GridSearchCV allowed for tuning the hyper-parameters while also performing cross-validation, ensuring the optimal settings and true performance for each model was reached, for each different partition of data.

Running the models on 9 different partitions of data allowed for testing results and efficacy across multiple years of matches, effectively creating many datasets from one. This made it so that effectively 9 completely independent "trials" were run on the various models, giving a more accurate depiction of the models' true performances.

Conclusion from Results

- Best classifier(s) and why - any drawbacks?
- Best regressor(s) and why - any drawbacks?

Future Work & Improvements

Despite our models' performing well, and even exceeding expectations, there are still many improvements that could be made given more time. There was unfortunately insufficient time remaining to implement these, due to needing to dedicate so much time and energy to processing and aggregating our data such that it could be used to predict both past and future matches. And thus, one major 'feature' in our dataset that was not leveraged in the models is the actual map that each game is played on. Adding the ability to leverage this categorical data would definitely be the single most impactful improvement that could be done to the models in future versions of this project. Additionally, the models currently use stats from all players as features, but this could be changed such that only the players that impact the model in a significant way (i.e. the worst players on each team) are have their stats used as features.