DS-3000 Project | Phase 2

Source of Data

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. Therefore, we will be using this website as the source of our data so that we can make accurate predictions about the outcomes of CS:GO matches.

Prior Works

We began by searching for any pre-existing datasets that met our needs, and we were actually able to find a surprisingly apt dataset on Kaggle. Unfortunately, the Kaggle dataset only includes matches up until June 2021, and a significant CS:GO update was released that added a new core mechanic. Since it is imperative that our dataset contains the most recent data possible, we feel that this dataset will not suffice for our project. Despite not being used as our dataset, the Kaggle dataset served as an excellent guide and glimpse into what kind of statistics and information we could gather from HLTV's website.

Discussion of Web Scraper

After a thorough discussion, we decided it would be necessary to create a few web scraper scripts from scratch using python. The main reason behind this decision is that using a pre-existing dataset would prevent us from properly pursuing the main goal of this project: testing our models' predictions by having them guess actual upcoming pro matches. The custom web scraper will allow us to gather new data to use as testing data for our prediction models which are trained on recent data. The ability to gather and add data from new matches to our dataset will also give our models the additional benefit of being as up to date as possible since the model can be retrained using data from more recent matches. Additionally, by scraping the data ourselves, our dataset will be tailored exactly to our needs given we can control which types of data or variables are scraped or not. This acts as a sort of preliminary data filtering and cleaning, making our formal exploratory data analysis much simpler.

All the data will be scraped solely from HLTV's website. The python libraries needed include the **requests** library for doing HTTP GET requests to download the html for each page being scraped. Then, once the page has been downloaded, it will be scraped and parsed using the **BeautifulSoup** (bs4) library.

Data Preparation Steps

For our exploratory data analysis (EDA), we will be using the Pandas python library to help prepare and manipulate the datasets, and then the Matplotlib.pyplot module to help with visualizations. There will be virtually no data cleaning necessary, since all the data is gathered and stored virtually by HLTV so there is very little "missing" data. That being said, some matches on HLTV don't provide headshot kills and/or flashbang assists for the players stats, and these missing values will be given NaN value in the prepared dataframe (dataset). These two variables are insignificant in real world application and therefore will likely be removed from the data set entirely. All team names will be forced lowercase since they are not case-sensitive. Player names, however, are case-sensitive and so the capitalization of player names will not be modified during EDA.

Each entry in the dataset represents the statistics and data for a single map (game) played. An entry contains information about how each team performed as a whole (round wins, team rating for the map, team's current ranking when the match was played, etc.) as well as 10 statistics for each individual player on each team. And so, each team has 50 variables/columns

(10 for each of the 5 players on the team) describing how the individual players performed for this map. Currently, our dataset holds slightly over 130 variables, and 15,000 entries!

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**.