**Using Machine Learning to Predict the Outcome of VGC Pokémon Battles**

**Overview**

The Video Game Championships, or VGC, is the official format of Nintendo’s Pokémon tournament battles. Our goal for the project is to create a machine learning model which can predict the winner of a battle based on information available in an open team sheet.

**Challenges**

Pokémon VGC is a complex game to model. During team building, players choose a team from over 1000 Pokémon, each with unique stats, moves, abilities, and types. Pokémon have 4 different moves to use per turn, and the same Pokémon can perform wildly differently between matches depending on training (EV distribution), items, movesets, and synergy with other Pokémon. VGC is played in a bring-6 pick-4 doubles format, expanding the total options available per turn and before-battle multiplicatively. As for other challenges, Pokémon battles include random factors like critical hits, move accuracy, and secondary effects, which complicates modeling. Finally, rule changes called ‘regulations’ occur every 3 to 4 months, massively influencing the popularity and effectiveness of different Pokémon and strategies.

**Similar Projects**

The [Pokémon Battle Predictor](https://www.pokemonbattlepredictor.com/home) project [began](https://www.reddit.com/r/stunfisk/comments/h9t99t/pokemon_battle_predictor_a_machine_learning/?utm_source=share&utm_medium=web3x&utm_name=web3xcss&utm_term=1&utm_content=share_button) quite similarly to ours, with the intent of predicting the winner of a Pokémon battle. That project evolved into creating an agent with impressive results against human players as well as a browser extension for players. Our project is different in that it focuses on the 2 vs 2 double battle format. It is also different in that predictions will be made before a single turn has been played, and not from any choices or events that occur mid-battle. That project used [TensorFlowJS](https://www.tensorflow.org/js/), and we will consider TensorFlow, though it would be in Python.

**Part A (Supervised Learning)**

Data consists of 5000+ battle logs scraped from Pokémon Showdown, a leading online Pokémon battle hosting site. We plan to run logistic regression, random forests, and eventually neural networks, to predict the outcomes in a probabilistic range. We aim to output win probabilities over binary outcomes, as Pokémon is a stochastic game and even the best players get unlucky or have bad games. As for evaluation, we are going to focus on metrics such as ROC-AUC score and F1-Score. For our final visualizations, I would like to take aesthetics into consideration, representing Pokémon by their official artwork via [PokeAPI](https://pokeapi.co/) calls, similar to [Poképaste](https://pokepast.es/6d4003745e66f0ed). I imagine both teams on each side represented by their sprites, with a % figure indicating the likelihood of winning for both teams.

**Part B (Unsupervised Learning)**

Dataset is the same as in the supervised portion. We want to use unsupervised learning to identify common team compositions so that we can a) check the extent to which these clusters align with ladder win rates or tournament success b) add team composition variables as features to improve our supervised predictions. We will need to one-hot encode categorical features such as Pokémon species, items, and abilities. We will need to remove trainer identifiers as well as outcome labels of battles. Dimensionality reduction may be necessary for visualization and performance. We plan to use K-means clustering to identify team composition. We plan to use PCA for dimensionality reduction. These techniques will be helpful for interpreting high-dimensional data like Pokémon teams. Similar to Part A, we will likely access [PokeAPI](https://pokeapi.co/) for visualization purposes to highlight prominent pokémon within archetype clusters. To evaluate these clusters, we will manually cross reference them against existing, known archetypes and synergies. To make visualizations more attractive, we will consider t-SNE.

**Team Planning:**

By the end of **week 3**, the two of us will have collaborated on similar parts of exploratory data analysis. By the end of **week 4**, Sean will have experimented with model parameters in order to gain insights via supervised learning. Jackson will have discovered insights into common clusters. By the end of **Week 6,** our model will be at least 95% ready for production. We will focus the last two weeks on final model touchess and cleaning up visualizations. We would like to dedicate the last week to presentation creation.