Pokémon G1 OU – Battle Outcome Prediction

Gabriel Gaitanaru Tommaso Montuori

Marco Autieri

Master's Degree in Data Science, Sapienza University of Rome

November 14, 2025

1 Overview

The objective of this project was to predict the outcome of competitive Pokémon Generation 1 OU battles using only the information contained in the `train.jsonl` and `test.jsonl` files. We developed three independent models—Logistic Regression, Random Forest, and XGBoost—each implemented in a separate Python script while sharing the same feature–engineering pipeline to ensure consistency, reproducibility, and compliance with the challenge rules.

2 Feature Engineering & Selection

We extracted two main families of features. **Static features** captured Player 1's team composition (average and summed base stats) and the opponent's lead Pokémon. **Dynamic features** were derived from the first 30 turns of the battle timeline, including HP trajectories, early-turn HP differentials, cumulative damage and recovery, move frequencies and base powers, inflicted statuses, KO counts, timing of the first KO, and advantage windows across early, mid, and late game phases. Relative features (*P1 - P2*) further improved separability. These engineered variables match exactly those computed in our three scripts.

3 Model Experimentation & Refinement

Logistic Regression served as a linear baseline, tuned through different values of the regularization parameter C. Random Forest experiments varied the number of trees, depth constraints, and split thresholds. XGBoost was optimized via learning rate, tree depth, subsampling, column sampling, and regularization parameters. While Logistic Regression provided stable but limited performance, both tree-based models captured non-linear interactions among timeline-derived features. Extending the timeline beyond 30 turns or using deeper Random Forests introduced overfitting, consistent with cross-validation results.

4 Validation Strategy

We adopted stratified 5-fold cross-validation across all experiments to prevent data leakage and to ensure stable comparisons between models. Validation results showed a strong correspondence with the public leaderboard. XGBoost achieved the highest performance with a public score of **0.8173**, outperforming Random Forest (~ 0.8146) and Logistic Regression (~ 0.809). Given its robustness and superior exploitation of timeline-based features, **XGBoost was selected as our final submission**.