CHARIZEFF

Members: Francesca Di Criscio, Francesco La Piana, Emidio Veccia

1. Introduction

The main goal of our work is to develop a model that could predict the winner of a Pokemon battle. The first step was the analysis of a given dataframe. After the analysis it was pretty clear that we had enough data to extract all the possible informations needed.

2. Feature Engineering

The first features we were able to extract were static: for each battle, we created a set of variables based only on the composition of Player 1's team, meaning values that do not depend on how the turns unfold. Although these features were useful for obtaining an initial representation of the problem, we soon realized that they were not sufficient on their own to capture the complexity of battle dynamics.

We later extracted a set of dynamic features from the battle_timeline, with the goal of giving the model information about the real progression of the fight. The dedicated function generates characteristics such as, for example, the damage dealt by each player, negative statuses inflicted and received, the number of Pokémon switches, KOs made, the number of turns in which Player 1 is under one or more negative statuses at the same time. These additions led to improved performance of our model.

The initial results obtained with the model were satisfactory but not particularly impressive. This led us to think that a more targeted use of the information contained in the battle_timeline could improve the model performance. Because of this, we decided to extend the feature set by including strategic characteristics, designed to capture tactical decisions made by the players.

We introduced the feature **momentum_score**, a synthetic measure of the dynamic advantage of Player 1 throughout the entire battle. The feature combines three key elements for each turn:

HP difference, Status inflicted or received and KO events. For each turn we compute:

$$M(t) = (hp_1(t) - hp_2(t)) + 0.5\, S_2(t) - 0.5\, S_1(t) + 3\, KO_1(t) - 3\, KO_2(t)$$

where $S_1(t)$ and $S_2(t)$ indicate the presence of a status condition, and $KO_1(t), KO_2(t)$ indicate KO events.

The final score is the average across all turns:

$$\text{momentum\_score} = \frac{1}{T} \sum_{t=1}^{T} M(t)$$

This feature summarizes the battle flow and, in our experiments, was among the most informative for tree-based models, since it directly captures HP advantage, statuses, and KOs, all correlated with victory. A second important feature is **damage_slope**, which measures how the damage inflicted by the Pokémon changes during the battle. It reflects the real balance of power between the two players. A positive slope means that Player 1 is gaining control over time, while a negative slope indicates a loss of momentum.

A variable initially considered but later discarded was the analysis of players' move choices. Because experienced players consistently rely on the same four or five moves, variability is low, and the move distribution adds no meaningful information to the battle's dynamics. Therefore, it was excluded from the final model.

3. Model

Once the first set of features was built, we chose **Logistic Regression** as the predictive model. This choice was motivated by the fact that our problem is a binary classification task (victory of Player 1 or Player 2), and logistic regression is simple, fast to train, and able to provide well-calibrated class probabilities. The integration of new (more complex) features led to a clear improvement in model performance. However, it also became evident that the features were becoming increasingly complex and non-linear. To overcome this limitation, we decided to adopt a **Random Forest** model, for several reasons: it naturally handles non-linearities and complex interactions; it reduces variance and limits overfitting; it provides useful feature importance values for interpretation. This model allowed us to obtain a more accurate representation of the phenomenon and, as a result, better predictive performance. After Random Forest, we also tried **XGBoost**, which builds trees sequentially, each one correcting the errors of the previous one. The motivation was to better capture complex patterns and weak predictive signals. Finally, thanks to the ensemble method, we combined the models used to exploit the strengths of each and obtain more stable and robust predictions compared to individual models.

4. Validation

For validation, we used **k-fold Cross-Validation** to evaluate the model in a reliable way, reduce dependence on a single train/validation split, and ensure that the obtained performance was truly generalizable. In the k-fold Cross-Validation we chose k = 5 because this number gives stable estimates with low compute cost.

5. Conclusions

The project showed that extensive feature engineering, combined with non-linear models and robust evaluation techniques, can significantly improve predictive performance on a dataset of Pokémon battles. The combined use of different models made it possible to capture the strategic complexity of the battles and to achieve reliable and competitive results.