Empowering Pokémon Trainers with Statistical Learning: Building Robust and Effective Teams

Group13

Bianco Valerio, Di Giammarco Susanna, Gnesi Moira,

Oddi Livia, Paoli Ilaria

June 2024

Abstract

In the world of competitive Pokémon, building a robust and effective team is a complex challenge that requires deep understanding of the Pokémon's characteristics, abilities, and synergies. This project aims to utilize statistical learning techniques to identify the key characteristics that define a competitively viable Pokémon. By constructing a comprehensive dataset from the most recent competitive Pokémon format (VGC24) and employing various predictive models, we seek to provide insights that can assist both novice and professional players in building competitive teams. The insights derived from our models not only highlight important features but also help in strategizing team compositions, thereby bridging the knowledge gap for players entering the competitive scene.

Contents

1 Introduction

Over all these years one of the main goals of Pokémon players has been to find the best team to use in competitive Pokémon tournaments. Building a competitive Pokémon team is a complex challenge that requires a deep understanding of characteristics , abilities and synergies among various Pokémon. The game knowledge needed to make all the choices necessary to build a good team is what usually keeps many player distant from a competitive environment, this problem was what convinced us to start this project.

Indeed, our main goal in this project is to identify the characteristics that define a competitively viable Pokémon and determine if these characteristics are also present in some Pokémon that are not typically used in competitive play. This type of research could serve as a valuable starting point for new players embarking on their competitive journey and can also help many professional players build their teams while avoiding biases.

2 Dataset Construction and Analysis

2.1 Data Sources and Collection

To construct our dataset, we considered all the Pokémon used in VGC24, the latest year of VGC (the competitive Pokémon format), as listed on LimitlessVGC. From this dataset, we obtained 122 Pokémon. Additionally, we randomly selected 28 Pokémon that have never been used competitively.

Then, for each Pokémon, we collected the data relative to the most common build in VCG24 from Smogon. The choice of features inside our dataset is due to games mechanics.

2.2 Features of the Dataset

In our dataset we decided to assign to each Pokémon the following 41 features:

- **Name:** the Pokémon's name.
- **Stats:** `HP`, `Attack`, `Defense`, `Special Attack`, texttttSpecial Defense, `Speed`.
- **EVs and IVs:** Features added to each stat.
- **Type 1 and Type 2:** Pokémon Types.
- **Item:** The hold Item.
- **Moves:** Each of the 4 moves corresponds to 4 features:
  - ◇ **Move power:** How much damage can be done using that move.
  - ◇ **Move Type:** The type of the move.
  - ◇ **Move effect:** Which effect is associated with the move.
  - ◇ **Move priority:** Move added speed.
- **Ability:** Pokémon special added effect.
- **Nature**: A feature that improve one stat further at the price of another stat reduction.
- **Target Variable**: The percentage use of the Pokémon inside the competitive format

To better understand the importance of each feature we will describe them and explain, in short, the impact of them in the game.

2.2.1 Stats

Each Pokémon has six stats associated to him, each of them has a particular meaning. `HP` indicates how many health points the Pokémon will have, `Attack` indicates a multiplier to the damage done by using physical attacks, `Defense` indicates the resistance to physical attacks of that Pokémon, `special attack` indicates a multiplier to the damage done by using special attacks, `Special defense` indicates the resistance to special attacks of that Pokémon and `Speed` is a value that is used to determine which Pokémon will attack before in the match.

2.2.2 Evs

`EVs (Effort Values)` are six features that are connected to the stats. Each of them can range from 0 to 255, but in competitive play, the maximum value *typically* used is 252 for game balance reasons. These values increase the power of the Pokémon in the corresponding stat, and the total value of `EVs` can add up to a maximum of 510. These 510 points can be assigned by the player to the Pokémon.

2.2.3 IVs

`IVs (Hidden Values)` have a similar function if compared to `EVs` but are generated by the game with the Pokémon when encountered. This stats are independent one from the other and can have values from 0 to 31, usually players choose only Pokémon with 31 `IVs` in all stats or 0 in some particular stats in particular cases for game reasons.

2.2.4 Types

These two variables indicate the types of our Pokémon. They play a crucial role because they influence both the damage inflicted by the Pokémon with its moves and the damage received from moves, depending on the move types. The available types are: `Normal, Flying, Fire, Psychic, Water, Bug, Electric, Rock, Grass, Ghost, Ice, Dragon, Fighting, Dark, Poison, Steel, Ground`, and `Fairy`. Every Pokémon has a `Type 1`, but not all Pokémon have a `Type 2`. Generally, Pokémon with two types are preferred by players due to their versatility, though some single-type Pokémon remain competitively viable. For Pokémon without a second type, we assign a value of “0” to `Type 2`.

2.2.5 Item

In competitive play, held items are a fundamental mechanic. These held items can vary, and each has a slightly different effect. To avoid having too many different objects with similar purposes but different names, we chose to analyze this variable as a categorical variable with the following categories:

- **Category 0:** No item (rare but possible).
- **Category 1:** Items used to boost stats (e.g., Choice items, Assault Vest).
- **Category 2:** Items used to recover `HP` (e.g., Berries, Leftovers).
- **Category 3:** Protection items (e.g., items that protect the Pokémon from terrain effects or status conditions).
- **Category 4:** Damage-enhancing items (e.g., Life Orb).
- **Category 5:** Type-enhancing items (e.g., items that boost specific types of moves).
- **Category 6:** Items affecting the terrain (e.g., Damp Rock, Grassy Seed).
- **Category 7:** Accuracy items (e.g., items that boost accuracy or critical hits or reduce the enemy’s accuracy).
- **Category 8:** Status-inducing items (e.g., items that affect the Pokémon’s status).
- **Category 9:** Items that work with the Pokémon’s ability.

2.2.6 Moves

For each move, as previously discussed, we used four features:

- **Power**: This is a numerical value that typically ranges from 0 to 200.
- **Move Type**: This feature indicates how the move will interact with the user’s and receiver’s types.
- **Move Effect**: Similar to held items, this feature describes the multitude of possible effects that each move can have. As with held items, we opted for a categorical structure for this variable, defined as follows:

  - **Category 1:** Moves whose power is affected by the status condition of the Pokémon using the move or receiving it (e.g., Facade, Hex).
  - **Category 2:** Status Moves. These moves inflict a status effect on the enemy Pokémon or reduce the user's stats as a side effect of a high damage output.
  - **Category 3:** Support Moves (e.g., Protect, healing moves, stats-altering moves).
  - **Category 4:** Utility Moves (e.g., moves that have effects related to the terrain, cause the enemy to flinch, have variable damage connected to weight, or have variable damage related to specific situations).
  - **Category 5:** Terrain Inducing Moves (e.g., Trick Room, Grassy Terrain, Rain Dance).
  - **Category 6:** Moves that hit more than one enemy.
  - **Category 7:** Moves that change the target of enemies' moves.
  - **Category 8:** Moves with no additional effect.
  - **Category 9:** Moves with recoil damage.
  - **Category 11:** Moves that require two turns.

  For some moves, two or more categories may seem appropriate. In such cases, the category was chosen based on the effect considered most valuable by competitive players for that move.

- **Move Priority**: This is an integer value that can range from -7 to +4.

We have indicated the variables described above in the constructed dataset as follows: `move1power`, `move2power`, `move3power`, `move4power` (power); `move1type`, `move2type`, `move3type`, `move4type` (move type); `statuseffectmove1`, `statuseffectmove2`, `statuseffectmove3`, `statuseffectmove4` (move effect); and finally `move1pri`, `move2pri`, `move3pri`, `move4pri` (move priority).

2.2.7 Abilities

Abilities, similar to items and move effects, are unique effects associated with each Pokémon. Due to their variety, we opted to study them as a categorical variable, called `Ability`, with the following classes:

- **Category 1:** Offensive Abilities (e.g., abilities that boost offensive output or add additional effects).
- **Category 2:** Defensive Abilities.
- **Category 3:** Status-Altering Abilities (e.g., Poison Point, Synchronize).
- **Category 4:** Abilities that affect speed in certain terrains or for specific moves.
- **Category 5:** Terrain-Inducing Abilities and abilities that boost defensive or offensive stats due to weather conditions.
- **Category 6:** Intimidate (a particularly competitive and relevant ability).
- **Category 7:** Recovery and Survival Abilities (e.g., Magic Guard, Sturdy, Regenerator).
- **Category 8:** Abilities that interact with the use of an item.
- **Category 9:** Counter Abilities (e.g., abilities that provide immunity to certain types of moves).

2.2.8 Target Variable

To help evaluate our results during modeling, we decided to add a variable, `Target`. This variable can be modeled in various ways. Our choice was to include the percentage usage of each Pokémon in competitive play from the Limitless VGC page. Other possible options were as follows:

- A binary variable $\{0, 1\}$, where 1 indicates Pokémon with a competitive presence and 0 for others.
- A categorical variable $\{0, 1, 2\}$, where 2 indicates Pokémon that won a competitive tournament, 1 for those that participated in at least one tournament but never won, and 0 for others.
- A categorical variable where each category indicates the Pokémon's ranking in the Smogon tier.

We chose to adopt our specific target variable because it is the most reasonable from our point of view and allows us to model more detailed target variables tailored to each model's requirements.

2.2.9 Excluded Features

There were other potential features that could have been included in our dataset, but we chose to exclude them. Below, we explain our reasoning for each:

- **Weakness Features:** For each Pokémon, we could have included 18 features, one for each type in the game, each expressing the relationship between each type and the Pokémon's types. We chose to exclude these variables because this information is already contained in Type 1 and Type 2 and can be easily obtained from the type chart table (2.1).

DEFENSE → ATTACK ↴	NOR	FIR	WAT	ELE	GRA	ICE	FIG	POI	GRO	FLY	PSY	BUG	ROC	GHO	DRA	DAR	STE	FAI
NORMAL													½	0			½	
FIRE		½	½		2	2						2	½		½		2	
WATER		2	½		½				2				2		½			
ELECTRIC			2	½	½				0	2					½			
GRASS		½	2		½			½	2	½		½	2		½		½	
ICE		½	½		2	½			2	2					2		½	
FIGHTING	2					2		½		½	½	½	2	0		2	2	½
POISON					2			½	½				½	½			0	2
GROUND		2		2	½			2		0		½	2				2	
FLYING				½	2		2					2	½				½	
PSYCHIC							2	2			½					0	½	
BUG		½			2		½	½		½	2			½		2	½	½
ROCK		2				2	½		½	2		2					½	
GHOST	0										2			2		½		
DRAGON															2		½	0
DARK							½				2			2		½		½
STEEL		½	½	½		2							2				½	2
FAIRY		½					2	½							2	2	½	

Figure 2.1: Type Chart Table

- **Move Precision:** Each move has a variable indicating its precision. We excluded this variable because the moves typically used in competitive play have 100% precision. It could be interesting to include it in the future and check for differences in model results.

- **Move Category:** Moves can be physical, special, or status moves. We excluded this variable because for physical and special moves, it is entirely dependent on the `Attack` and `Special Attack` stats. Depending on which stat is higher, all moves of that Pokémon will fall into that category (if `attack` > `special attack`, the moves are physical; otherwise, they are special). For status moves, this information is included in the effect features and the fact that power move is equal to 0.

- **TeraType Feature:** This feature is connected to a specific game mechanic. We chose to exclude it because its use is strictly player-dependent and, unlike other features, it will disappear with the release of new games and meta.

2.2.10 Dataset construction problems

During the construction of our dataset we encountered the following obstacles:

- For some Pokémon in our dataset, the competitive build of the most recent VGC period was not present. In those cases we insert the data relative to the most recent competitive build possible.

- Due to our choice of the target variable, the values are compressed near 0. It is important to consider this when using this variable, as the "outliers" represent the best values of our target and should be treated as such.

- During the construction of our dataset, we noticed many differences between various sites' evaluations of each Pokémon. We chose to follow the evaluations of Limitless VGC, but the same analysis conducted on Smogon's top Pokémon would yield a completely different set of Pokémon and different results.
- We decided to focus on a single build for each Pokémon instead of analyzing all possible builds. In the future, it could be interesting to add more builds to improve our analysis.

2.3 Descriptive Analysis

The dataset consists of 150 Pokémon with 41 features each, including both numerical and categorical data such as `HP`, `Attack`, `Defense`, `Sp.Atk`, `Sp.Def`, `Speed`, various `EVs` and `IVs`, and `Type` information. Detailed graphical analyses and visualizations are included in the appendix (.1) to provide a deeper understanding of the relationships between these features and the target variable.

Target Variable Analysis

From the analysis performed using boxplots, it was possible to notice strong outliers of the target variable, for example concerning types: 'Fairy' and 'Grass' Thus, we created a binary variable, **`target_close_to_one`**, to indicate whether the target value of each Pokémon is close to one (i.e., greater than or equal to 0.8). This variable helps identify Pokémon with high competitive viability.

Distribution of Pokémon Types

The frequency distribution of Pokémon types for both `Type 1` and `Type 2` was analyzed. The analysis revealed that certain types are more prevalent than others. For instance, common types like `Water`, `Fire`, and `Flying` had higher frequencies, while types like `Bug` and `Poison` were less common. This distribution is important as it influence the model's predictions and the competitive viability of different Pokémon types.

Categorical Variable Analysis

We analyzed various categorical variables, including `item effect`, `Ability`,`Nature` and and all the Moves' Status Effects. The frequency distributions of these variables provided insights into their prevalence and potential impact on the target variable. For example, certain abilities like `Protosynthesis`, `Intimidate` and `Defensive abilities` were found to be more common among competitively viable Pokémon, suggesting their significant role in competitive success. In relation to the Status effect, it was observed that for all 4 moves, status effects "Terrain Inducing Moves" and "Moves with no additional effect" were common among competitively viable Pokémon

Numerical Variable Analysis

The analysis of relationships among the numerical variables was conducted by calculating the correlation between the Target variable and variables with variance ¿0. Specifically, the analysis revealed a positive correlation between the target variable and Move Priorities, and a negative correlation with Evs Sp.atk, Evs Sp, and Ivs Atk.

The descriptive analysis provided valuable insights into the dataset's characteristics. Understanding the distribution and relationships of various features helps in building more effective predictive models. The results indicate that both offensive and defensive stats, as well as strategic use of abilities and items, play significant roles in determining a Pokémon's competitive viability. These insights will be crucial in guiding the development and refinement of our models.

3 Models

In our analysis, we explored a variety of predictive models to assess their effectiveness with our dataset. Each model has been implemented through distinct scripts, which are organized and available within a zipped folder. Each script is conveniently named after the modeling technique it employs, ensuring easy navigation and reference. In the following sections, we will explore the outcomes of these models and discuss the reasoning behind the methodological choices made during their application.

3.1 Neural Network

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 252)]             0

 dense_3 (Dense)             (None, 128)               32384

 dense_4 (Dense)             (None, 64)                8256

 dense_5 (Dense)             (None, 4)                 260

=================================================================
Total params: 40900 (159.77 KB)
Trainable params: 40900 (159.77 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 3.1: Neural Network Model Summary

As a first model we employed a Neural Network to predict the competitive viability of Pokémon based on the features in our dataset. We chose this model due to its ability to handle complex, non-linear relationships between features and target variables, which is ideal for our dataset with its mixture of numerical and categorical data.

We modeled the neural network as it follows: The architecture of our neural network model is detailed below:

Layer	**Description**
Input Layer	1 layer with input shape matching the number of features
Hidden Layers	2 layers with 128 and 64 neurons respectively, both using ReLU activation
Output Layer	1 layer with softmax activation for multi-class classification
Optimizer	Adam with a learning rate of 0.001
Loss Function	Sparse Categorical Crossentropy
Epochs	20
Validation Split	20%
Batch Size	32

Table 3.1: Neural Network Architecture and Training Details

The preprocessing stage was critical to preparing our dataset for effective model training. Initially, we removed columns with zero variance, as they do not contribute any useful information. Next, we identified each column as either numeric or categorical, and the latter were standardized by converting them to strings and encoding them using a `OneHotEncoder`. To address class imbalance in our target variable, we employed oversampling to ensure an even distribution of classes. Numeric features were then standardized using a `StandardScaler` to normalize the data. Finally, the dataset was divided into training and testing sets, with 80% of the data allocated for training and 20% for testing, using stratified sampling to maintain an even distribution of target classes.

Feature Importance Analysis Using `LIME`

In our analysis, we utilized the `LIME` library to understand the contributions of individual features to the predictions made by our neural network model. `LIME` provides local explanations by approximating the model with a simpler, interpretable model around each prediction. This allows us to gain insights into the importance of different features in driving the model's decisions.

Steps Involved in Feature Importance Analysis

1. **`LIME` Initialization**: We initialized the `LIME` explainer by passing the training data and the prediction function of our trained neural network model. This setup allowed `LIME` to generate explanations for each prediction made by the model.
2. **Generating Explanations**: For each instance in the test set, `LIME` generated a local surrogate model to approximate the predictions of the neural network. This involved perturbing the data around the instance and observing the changes in the model's predictions. The explanations provided by `LIME` included the importance of each feature in terms of its contribution to the predicted probability for each class.
3. **Aggregating Feature Importances**: The feature importances from individual instances were aggregated to obtain an overall importance score for each feature. This aggregation was done by summing the absolute values of the feature importance scores across all instances.
4. **Visualization**: The aggregated feature importances were visualized using a bar plot, shown below in Figure (3.2). The features were sorted by their importance scores, and colors were used to indicate the level of importance, ranging from green (most important) to red (least important).

Interpretation of Feature Importance

In (3.2) are shown the most important features identified by `LIME`, that included:

- Move priorities (e.g., `move 2 priority`)
- Specific attributes such as `Speed`, `Attack`, and `Sp.Atk`

Features such as move priorities and types significantly influenced the model's predictions. These features likely impact the effectiveness of the Pokémon in battles, thereby playing a crucial role in the target class determination. Attributes like `speed` and `attack` indicate the inherent strengths of the Pokémon, contributing to their classification.

3.1.1 Confusion Matrix Analysis for `TargetClass == 3`

The confusion matrix in Figure (3.3) provides a detailed evaluation of the model's performance by showing the number of correct and incorrect predictions for each class. Here, we focus on the performance of the model with respect to `TargetClass == 3`, which contains the most powerful Pokémon.

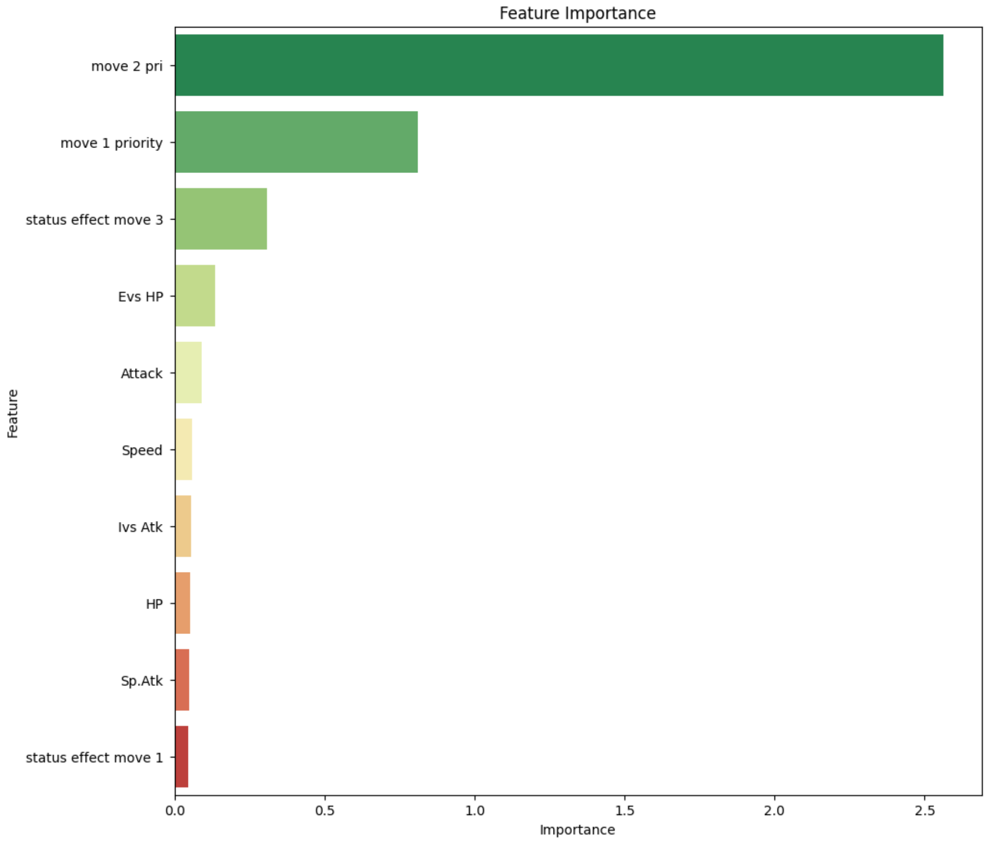


Figure 3.2: Feature importance analysis using `LIME`. This plot shows the most influential features as determined by `LIME`, with green bars indicating features that positively influence the prediction and red bars indicating features that negatively influence the prediction.

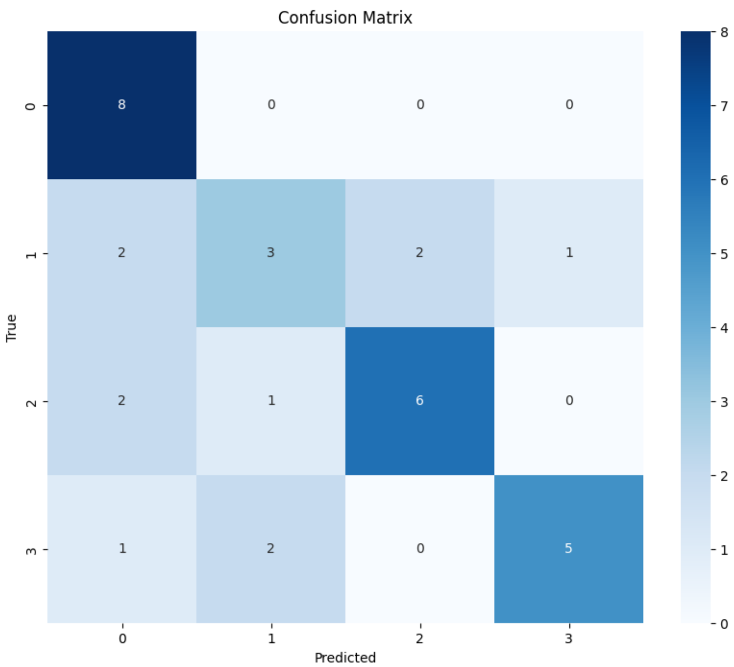


Figure 3.3: Confusion Matrix showing the performance of the neural network model on the test set. The matrix illustrates the number of correct and incorrect predictions for each class.

1. **True Positives**: the model correctly predicted 5 instances of `TargetClass == 3`. These are the

cases where the Pokémon were correctly identified as belonging to class 3.

2. **False Positives**: The model incorrectly predicted 3 instances of other classes (0, 1, and 2) as `TargetClass == 3`. These are the cases where Pokémon from other classes were mistakenly classified as class 3.

3. **False Negatives**: There were 3 instances of `TargetClass == 3` that were incorrectly predicted as other classes (1 and 2). These errors suggest that the model struggled to distinguish class 3 Pokémon from certain other classes.

The feature importance analysis using `LIME` and the confusion matrix provide valuable insights into the model's performance and the factors driving its predictions. Understanding which features most influence the model's predictions helps in refining the model and improving its accuracy. Features like move priorities, types, and specific attributes are crucial in determining the target class.
Analyzing the confusion matrix highlights the strengths and weaknesses of the model in classifying different classes. For `TargetClass == 3`, the model shows moderate performance with significant misclassifications, indicating room for improvement in distinguishing this class from others.

3.2 Support Vector Machine (SVM)

We selected the SVM (Support Vector Machine) algorithm for our dataset because of its robustness, effective handling of small sample sizes, and ability to work with non-linear data. These qualities make it an ideal model for our analysis. To apply SVM, we started by transforming all categorical variables, including those that R considered numerical but were actually categorical, into numerical variables using one-hot encoding. Then, we proceeded to delete the zero variance features (four of the IVS features were zero variance features).

After that, we divided our Target variable in 2 classes respect to their target variable. Those higher than 0.8 and those lower that 0.8. Subsequently we split our dataset into a training set and a test set with a 80/20 ratio. We trained the SVM model using the training set, with the target variable as the target. We then tested it on the test set.

Next, we checked the mean accuracy to evaluate the performance of the SVM model, our final results was 53.33%. That proves a poor performance of our method to distinguish between the classes. For this reason, and due to other methods yielding better results, we chose to concentrate on those more effective methods.

3.3 K-Prototypes

The K-Prototypes algorithm, an enhancement of the K-Means and K-Modes algorithms, was selected to manage the varied data types within the Pokémon dataset. This approach efficiently categorizes data by integrating both numeric and categorical elements.
Through the elbow method, the optimal cluster count was established at k = 3.

In the analysis, the target variable was excluded from the initial clustering phase to preserve the unsupervised nature of the methodology. Nonetheless, it was subsequently used to evaluate the clusters. A rating system was designed to pinpoint clusters with a greater proportion of "powerful" Pokémon, taking care to prevent excessively high values from unfairly affecting the score. For instance, Pokémon with a target value exceeding 5 were assigned a score of 5 plus the target value divided by 100, thereby moderating their effect on the overall assessment.

To prepare the data for clustering, categorical variables were recognized and reclassified from numeric to categorical formats. These were then merged with the original categorical variables, and all were converted to factor types prior to clustering, ensuring they were correctly processed by the K-Prototypes algorithm.

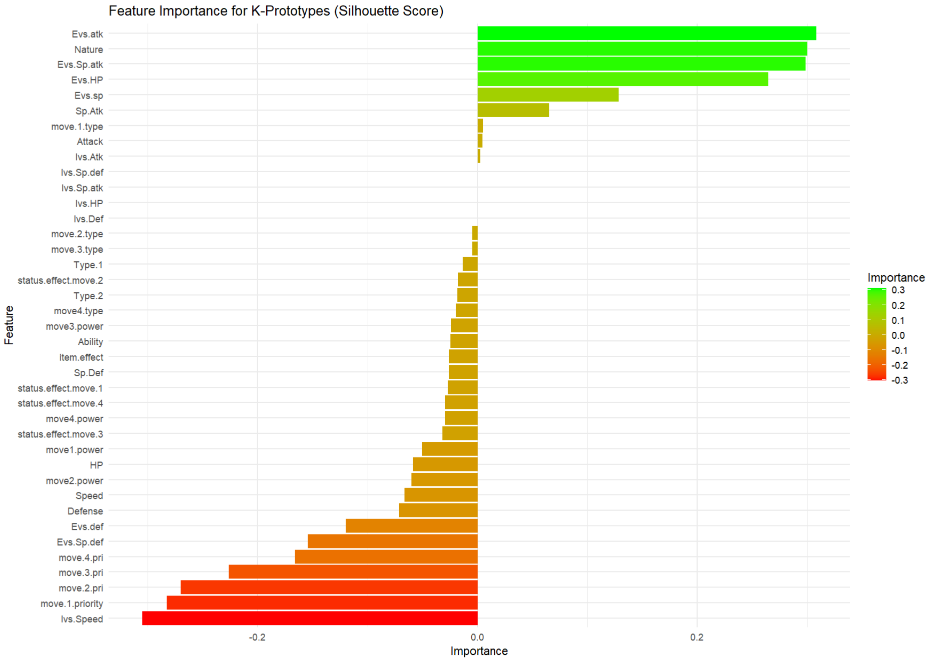


Figure 3.4

Feature Importance

Features like `Evs` and `Nature` are crucial, as their removal substantially lowers silhouette scores, highlighting their essential role in defining clusters. In contrast, the removal of features such as `Ivs speed` and `moves priority` improves silhouette scores, suggesting they might be redundant or negatively impact clustering effectiveness.
Effort Values, particularly those related to stats like Attack and Sp.Attack, significantly impact clustering, with `Evs sp.Atk` notably decreasing clustering quality when removed.
This analysis helps in making informed decisions about which features to adjust or focus on in our dataset for more accurate and effective analytical tasks.

The choice of the K-Prototypes model was motivated by its ability to concurrently deal with both numeric and categorical variables, a crucial feature considering the dataset's mixed data types. This algorithm is especially well-suited for datasets requiring the consideration of various attributes, such as Pokémon types, abilities, and stats, to create significant groups that could guide the formation of competitive teams.

3.4 Random Forest

In our analysis of the Pokémon dataset, we employed the Random Forest algorithm, renowned for its robust handling of both categorical and numerical data without extensive preprocessing. This choice was driven by the model's capability to manage mixed data types effectively and its prowess in capturing complex, non-linear relationships between features—crucial for deciphering Pokémon competitiveness.

The preprocessing phase included rigorous data cleaning where columns with zero variance were removed to enhance model performance. Categorical variables were transformed into factors, and a new target variable was created by categorizing a continuous numeric variable into classes based on quantile ranges, simplifying the classification challenge and improving interpretability.

Ultimately, the trained Random Forest model highlighted key features determining Pokémon's competitive edge through a clear visualization of variable importance:

It was evident that the types of Pokémon and their moves are the most crucial factors, indicating that strategic diversity in move types is more decisive than mere statistical power. The effectiveness of these moves (`status.effect.move`) was also significant but secondary to the type attributes. Basic stats

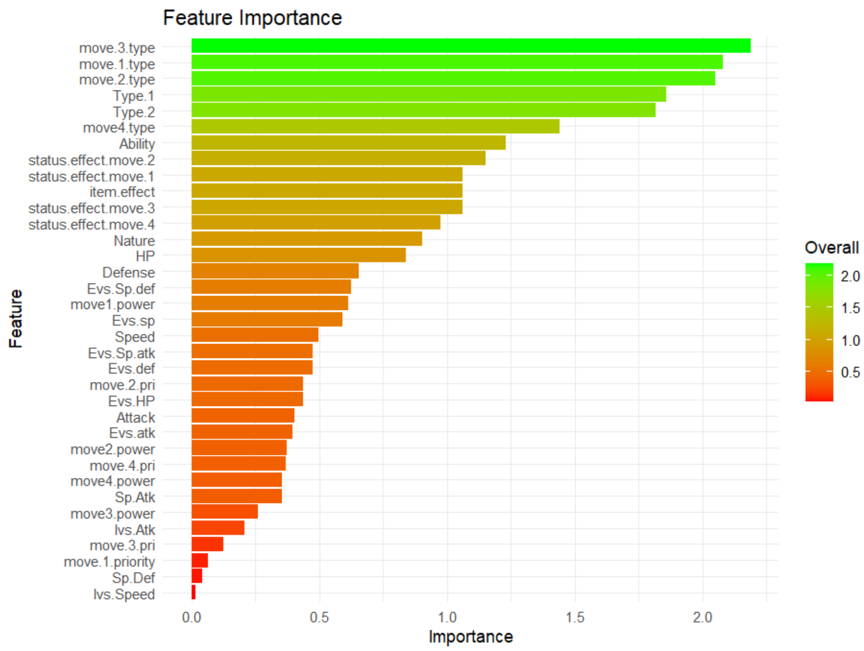


Figure 3.5

like HP, Ability, and other metrics were less influential but contributed to the overall classification. The least impactful were move priority and power and specific EVs and IVs, suggesting that these are less critical in our model's decision-making process. This empowerment through clear feature prioritization allows players to make informed decisions when building competitive teams.

We chose Random Forest because it not only provides deep insights into feature importance, aiding strategic decision-making in team composition but also excels in environments where preventing overfitting is critical, even with our dataset's moderate size. The model was tuned for optimal parameters like `mtry`, `ntree`, and `maxnodes` to boost accuracy and performance.

4 Let's Fight

4.1 Build Our Team

In this section, we discuss the rigorous process of team selection using an algorithm developed to identify the most competitive Pokémon from the output of various models. This section describes in details the implementation of our algorithm in `R`, which reads an Excel file containing Pokémon data, constructs a graph based on their battle statistics, and selects a balanced team of six Pokémon with diverse primary types.

4.1.1 Data Preparation and Graph Construction

The first step in our methodology involves reading the Pokémon data from a file which contains Pokémons the models chose. To facilitate the selection process, we construct a graph where each Pokémon is represented as a node. The edges between these nodes signify the differences in their battle statistics, calculated as the absolute difference in their overall strength. The overall strength of a Pokémon is determined by summing its battle statistics (`HP`, `Attack`, `Defense`, `Sp.Atk`, `Sp.Def`, and `Speed`).

The graph is constructed as follows:

1. **Node Representation**: Each Pokémon in the dataset is represented as a node.
2. **Edge Calculation**: Edges between nodes are weighted based on the absolute differences in their overall strength.
3. **Graph Creation**: A complete graph is created, where every Pokémon is connected to every other Pokémon, with edges representing their strength differences. Self-connecting edges (a Pokémon connected to itself) are removed to maintain meaningful connections.

With the graph in place, the next step is to select a team of six Pokémon. The selection algorithm focuses on identifying Pokémon with the highest degree of connection in the graph. The degree of connection refers to the number of edges connected to a node, indicating the Pokémon's significance in the network based on their battle statistics. The selection process is explained below:

1. **Degree Calculation**: Calculate the degree of connection for each node (Pokémon) in the graph.
2. **Sorting**: Sort the Pokémon based on their degree of connection in descending order.
3. **Diversity Assurance**: Ensure that the selected team consists of Pokémon with different primary types, `Type 1`. This step is crucial to avoid redundancy and build a balanced team capable of handling diverse battle scenarios.

Figure 4.1 shows the flowchart illustrating the process of selecting a Pokémon team. The algorithm was applied to our dataset to form a competitive team for the Pokémon Showdown simulations. By using this method, we ensured that selected Pokémon had the highest overall battle statistics, our team included Pokémon of different primary types to cover various strategic advantages and then we used graph theory to identify key Pokémon based on their connectivity and significance in the network. The algorithm was critical in forming a competitive team for our simulations. By ensuring diversity in primary types and selecting Pokémon with the highest degrees of connection, we created a robust team capable of handling various battle scenarios. This methodology underscores the utility of graph theory and network analysis in strategic decision-making and team composition in competitive Pokémon play.

So, by combining statistical learning with network analysis, we were able to build a robust and balanced team. The complete `R` code implementing this algorithm is provided in an accompanying file, allowing further exploration and application of this method.

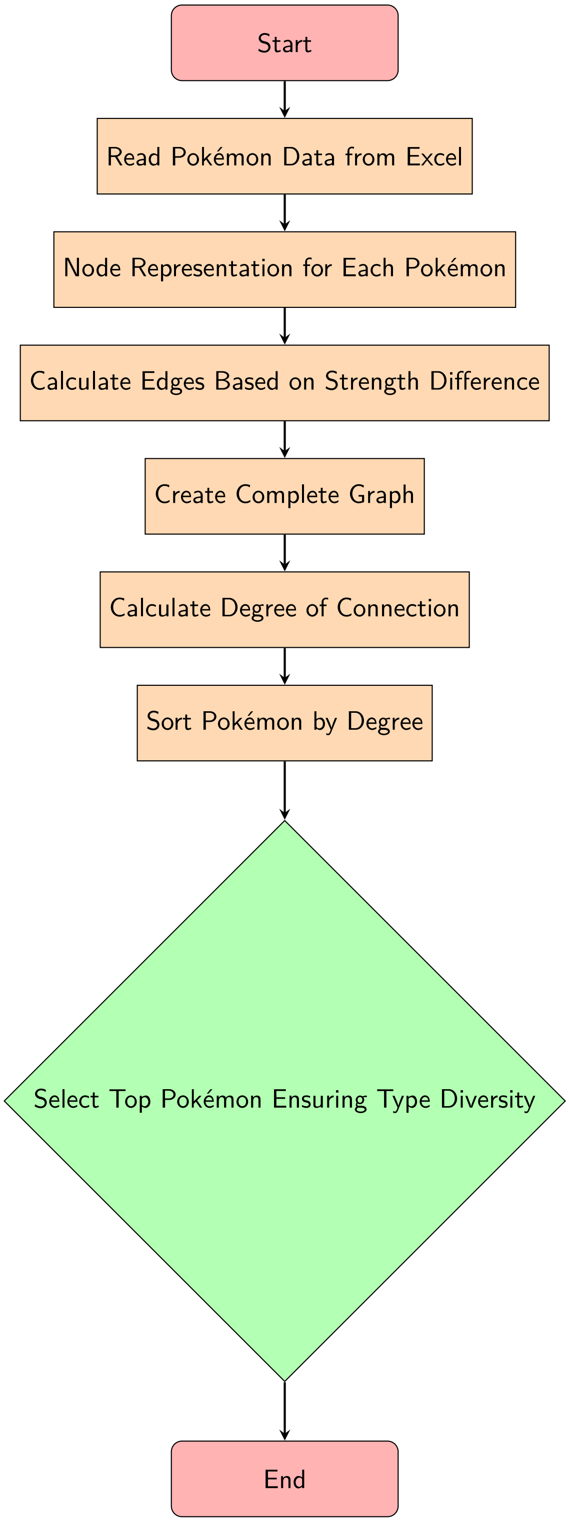


Figure 4.1: Flowchart showing the process of selecting a Pokémon team.

4.2 Battle Simulations

We conducted 50 simulations on the Pokémon Showdown website, each lasting approximately 9 minutes. These simulations were carried out between two trainers of equal skill level, with the Teratype not included. Pokémon are selected using the algorithm explained in the previous section (4.1). The simulations conducted include teams formed by the combined predictions of the Random Forest, Neural Network, and K-Prototype models. An example of a team we built is shown below:

Figure 4.2: "StatLearn" Team.

Battle Setup

- **Platform**: Pokémon Showdown
- **Number of Simulations**: 50
- **Duration per Simulation**: 9 minutes
- **Trainer Skill Level**: Equal

Our team was tested against actual competitive teams to evaluate performance. Out of 50 battle simulations, our team won 42% (21 battles) of them. The results were obtained from battles against competitive Pokémon teams. The competing trainers had roughly equal skill levels, and naturally, the results were influenced by the tactical choices made by the trainers. Conducting this experiment with high-level players could help mitigate the bias related to player skills.

Battle Results

- **Total Simulations**: 50
- **Wins**: 21
- **Win Rate**: 42%

4.3 Detailed Battle Reports

Battle 1

Teams:

- **Statlearn2's Team**: `Tornadus`, `Urshifu-*`, `Glimmora`, `Dondozo`, `Tatsugiri`, `Flutter Mane`
- **Statlearn's Team**: `Bastiodon`, `Amoonguss`, `Iron Hands`, `Ting-Lu`, `Dudunsparce-*`, `Baxcalibur`

Outcome: `Statlearn` won the battle.
Highlights:

- **Turn 1:**
  - `Dondozo` and `Tatsugiri` combination led to stat boosts for `Dondozo`.
  - `Dondozo` used `Order Up`, but `Iron Hands`' `Wild Charge` caused significant damage to `Dondozo`.
  - `Bastiodon` used `Stealth Rock`, setting up hazards.

- **Turn 2:**
  - `Dondozo` used `Protect` to avoid `Wild Charge` from `Iron Hands`.
  - Both `Dondozo` and `Bastiodon` recovered HP using `Leftovers`.
- **Turn 3:**
  - `Dondozo` increased its Attack with `Order Up` but fell asleep due to `Amoonguss`' `Spore`.
- **Turn 4:**
  - `Dondozo` fainted after continued attacks from `Iron Hands` and `Amoonguss`.
  - `Flutter Mane` entered the battle with a boosted `Sp. Atk`.
- **Turn 5-8:**
  - Various strategic switches and attacks resulted in stat adjustments and positioning, leading to a tactical battle between the teams.

Battle 2

Teams:

- **Statlearn2's Team**: `Gholdengo`, `Iron Hands`, `Amoonguss`, `Iron Bundle`, `Pelipper`, `Salamence`
- **Statlearn's Team**: `Bastiodon`, `Amoonguss`, `Iron Hands`, `Ting-Lu`, `Dudunsparce-*`, `Baxcalibur`

Outcome: `Statlearn` won the battle.
Highlights:

- **Turn 1:**
  - `Salamence` used `Tailwind` to increase team speed but was knocked out by `Baxcalibur`'s `Glaive Rush`.
- **Turn 2:**
  - `Iron Hands` and `Pelipper` worked together, with `Iron Hands` using `Drain Punch` to defeat `Bastiodon`.
- **Turn 3:**
  - `Amoonguss` used `Spore` to put `Dudunsparce` to sleep, allowing `Iron Hands` to continue its assault.
- **Turn 4-5:**
  - Strategic moves and `Rage Powder` from `Amoonguss` controlled the battlefield, with `Baxcalibur` using `Swords Dance`.
- **Turn 6-8:**
  - `Pelipper` and `Iron Hands` leveraged weather and type advantages to defeat `Ting-Lu` and `Baxcalibur`, securing the win for `Statlearn`.

These battles highlight the strategic depth and robustness of our team's composition and the models' predictions. The results demonstrate the importance of high-level competitive play in validating model performance and reducing biases related to player skills.

The detailed videos of the two example battles are attached to this report for further analysis and review.

5 Conclusions

This study has demonstrated the effectiveness of using statistical learning techniques to analyze and predict the competitive viability of Pokémon. By constructing a comprehensive dataset and applying a variety of predictive models, we have identified key characteristics that influence a Pokémon's success in competitive play.

The **Neural Network model** provided valuable insights into the complex, non-linear relationships between features. Using `LIME` (Local Interpretable Model-agnostic Explanations), we identified the most influential features, such as move priorities, types of moves, and specific attributes like `Speed`, `Attack`, and `Sp.Atk`. Despite achieving a moderate accuracy, the neural network's ability to reveal feature interactions and local feature importance makes it a powerful tool for strategic decision-making in team building. Additionally, the confusion matrix analysis for `TargetClass == 3` highlighted the model's performance in correctly classifying competitive Pokémon, albeit with some misclassifications.

The **Random Forest model** excelled in identifying the most significant features, such as Pokémon types and move types, underscoring their crucial role in competitive viability. This model's robustness and interpretability make it an excellent choice for understanding feature importance and guiding team composition decisions.

While the **Support Vector Machine (SVM)** offered less satisfactory results, it still provided useful perspectives on data segmentation within the dataset.

The **K-Prototypes model** gave us good results identifying a cluster with a high presence of competitive Pokémons and giving us an important insight on the possibility of clusterization of our dataset. In particular we saw that the main characteristics of the best cluster were higher `HP`, `Attack` and `Defense`. A surprise was the total absence of `Intimidate` from the Pokémon's ability in the best cluster.

Our analysis also included practical battle simulations conducted on the Pokémon Showdown platform. We carried out 50 simulations, each lasting approximately 9 minutes, between trainers of equal skill level. These simulations demonstrated that our team, constructed based on the model predictions, won 42% (21 out of 50) of the battles against competitive teams. This highlighted the effectiveness of our model in real-world scenarios, although it also emphasized the impact of player skill and strategic decisions on battle outcomes. Conducting similar experiments with high-level players could provide further validation and reduce bias related to player skills.

Overall, our analysis emphasizes the importance of both offensive and defensive stats, strategic use of abilities and items, and the critical role of move types in determining a Pokémon's competitive edge. These findings can serve as a valuable resource for players seeking to enhance their competitive teams and for further research in the field of competitive gaming analytics. Future work should focus on further optimizing the model and exploring additional preprocessing techniques to enhance performance. The findings underscore the potential of advanced machine learning methods in strategic planning and decision-making in the realm of competitive Pokémon.

.1 Appendix: Graphs and Visualizations

In this appendix, we present the graphical analysis and visualizations of our dataset, which help in understanding the relationships between various features and the target variable.

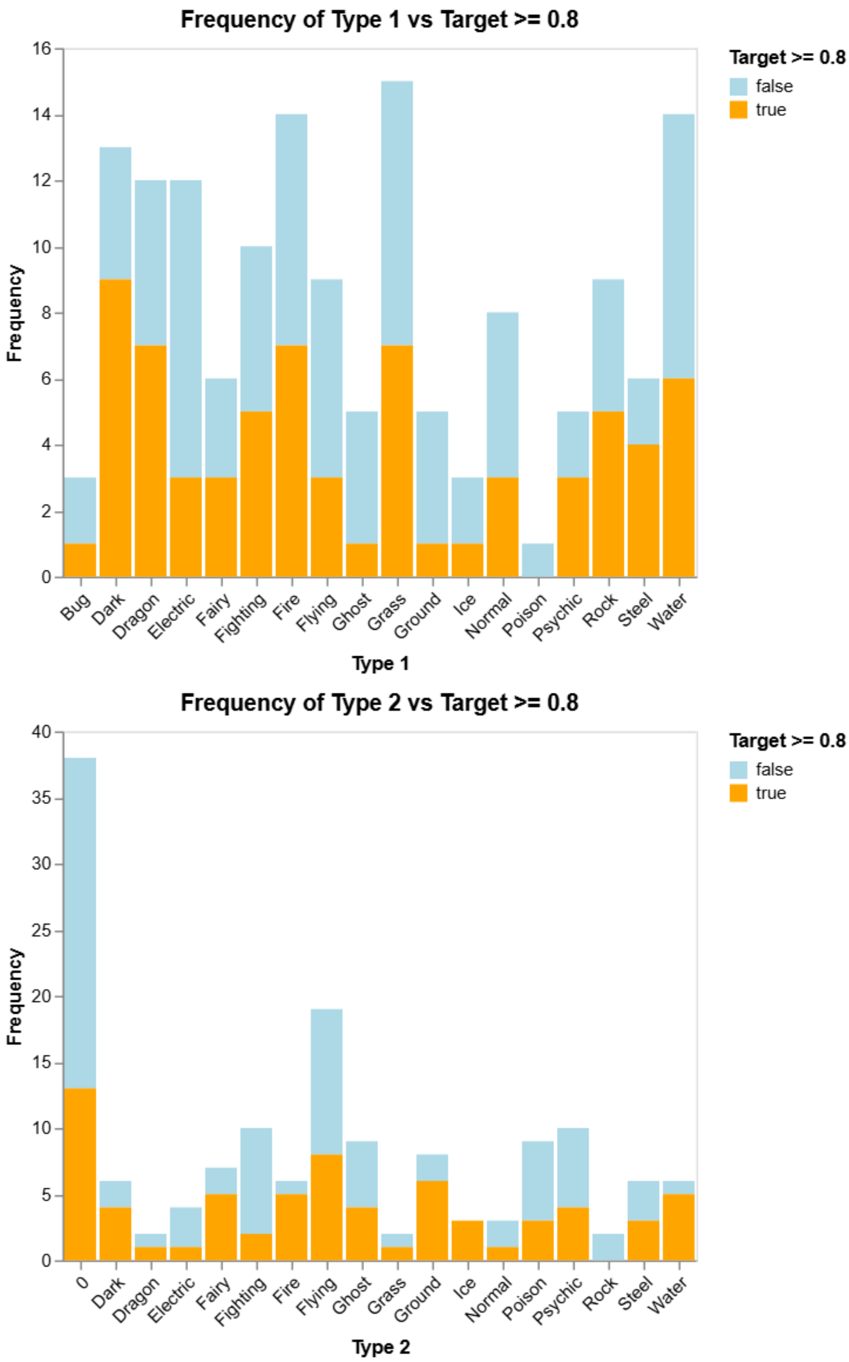


Figure 1: Frequency distribution of Pokémon types in relation to the target variable. The graphs show how the primary and secondary types of Pokémon are distributed based on their competitive viability.

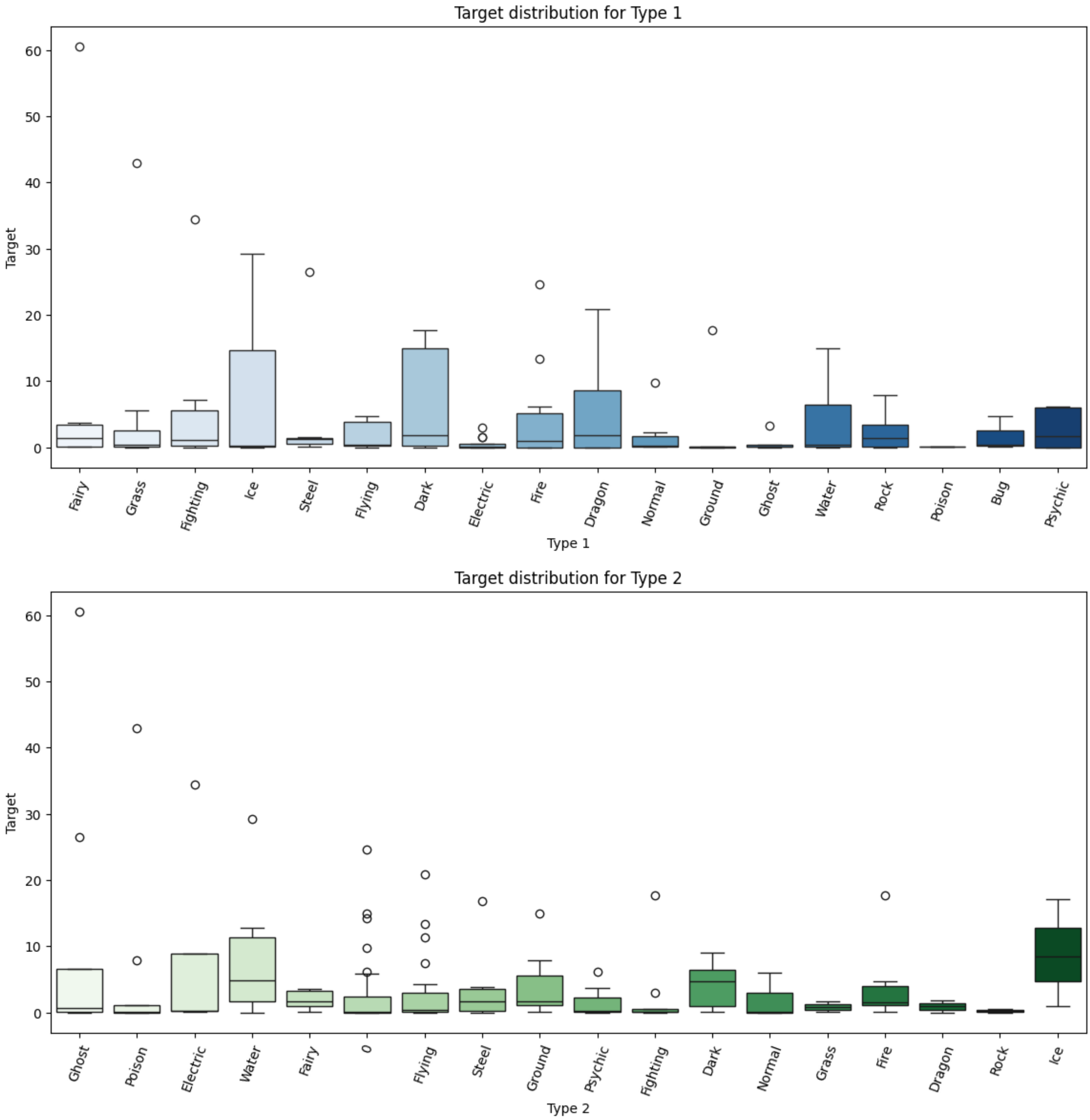
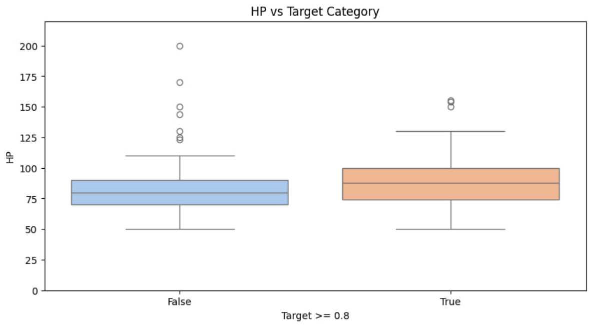
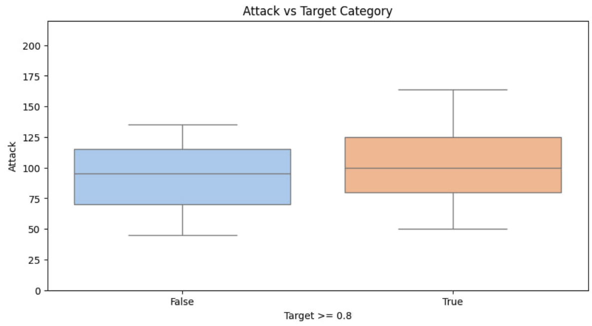


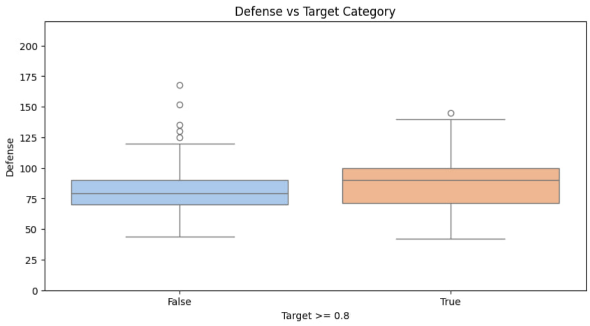
Figure 2: Target distribution across different Pokémon types (`Type 1` and `Type 2`).



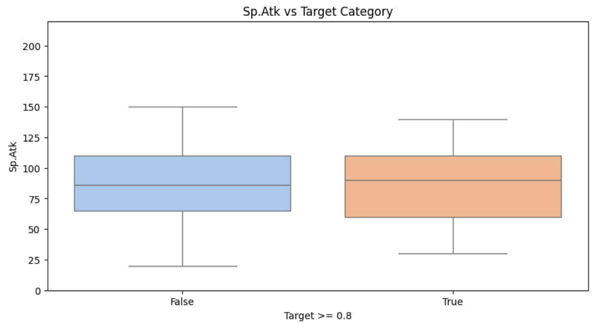
(a) Distribution of `HP` for Pokémon categorized by `Target` $\geq$ `0.8`. The boxplot shows that Pokémon with higher target values tend to have a higher median `HP`.



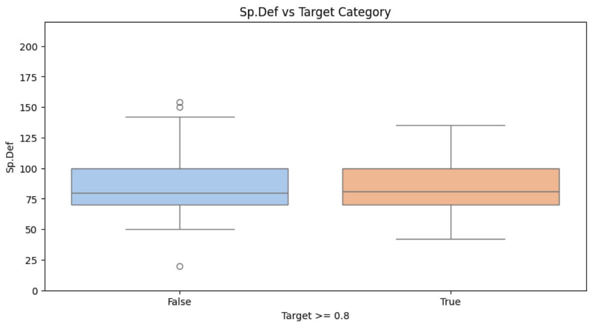
(b) Distribution of `Attack` for Pokémon categorized by `Target` $\geq$ `0.8`. The boxplot reveals that Pokémon with higher target values tend to have higher median `Attack` stats.



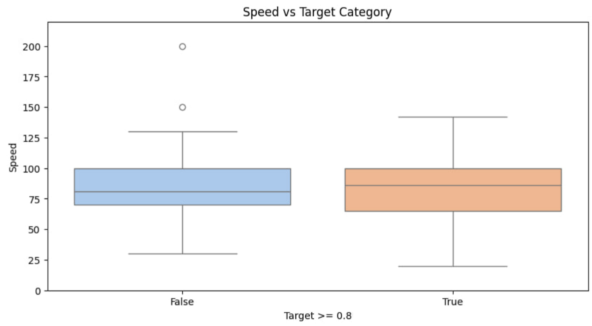
(c) Distribution of `Defense` for Pokémon categorized by `Target` $\geq$ `0.8`. The boxplot indicates that Pokémon with higher target values generally have higher median Defense stats.



(d) Distribution of Special Attack (`Sp.Atk`) for Pokémon categorized by `Target` $\geq$ `0.8`. The boxplot shows that Pokémon with higher target values exhibit slightly higher median `Sp.Atk` stats.



(e) Distribution of Special Defense (`Sp.Def`) for Pokémon categorized by `Target` $\geq$ `0.8`. The boxplot illustrates that Pokémon with higher target values generally possess higher median `Sp.Def` stats.



(f) Distribution of `Speed` for Pokémon categorized by `Target` $\geq$ `0.8`. The boxplot demonstrates that Pokémon with higher target values tend to have higher median `Speed` stats.

Figure 3: Boxplot of some (`HP`, `Attack`, `Defense`, `Special Attack`, `Special Defense`, `Speed`) characteristics of the Pokémon in our dataset

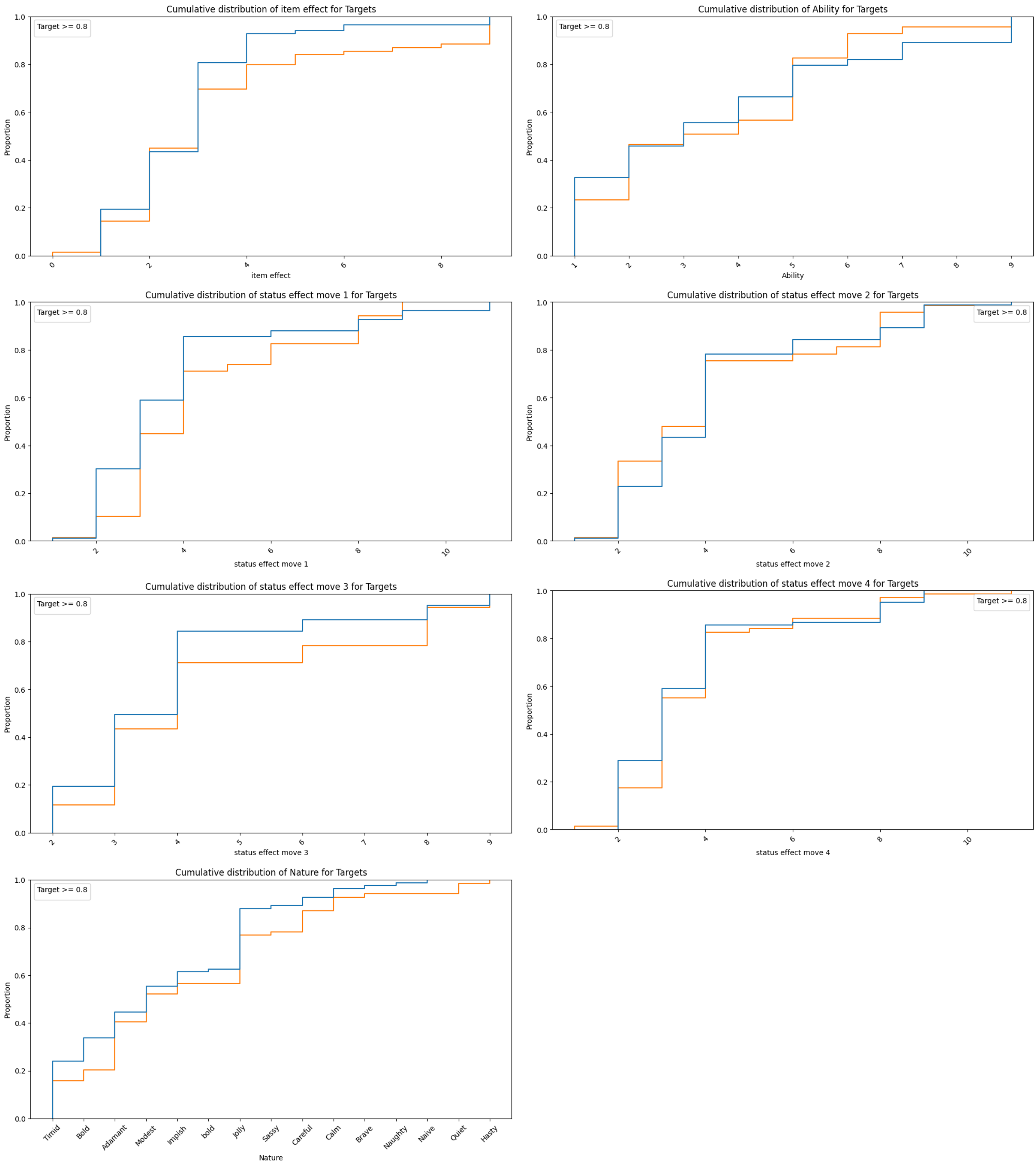
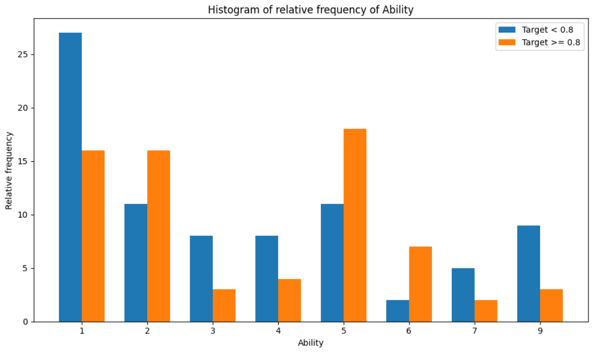
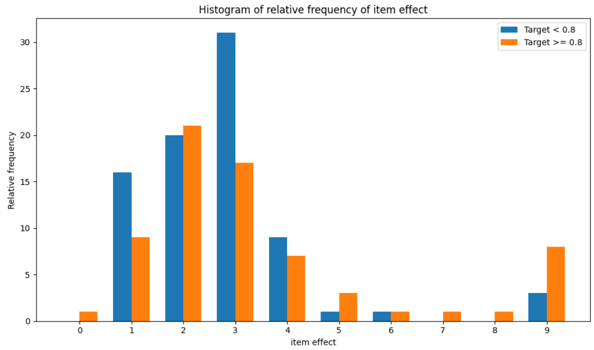


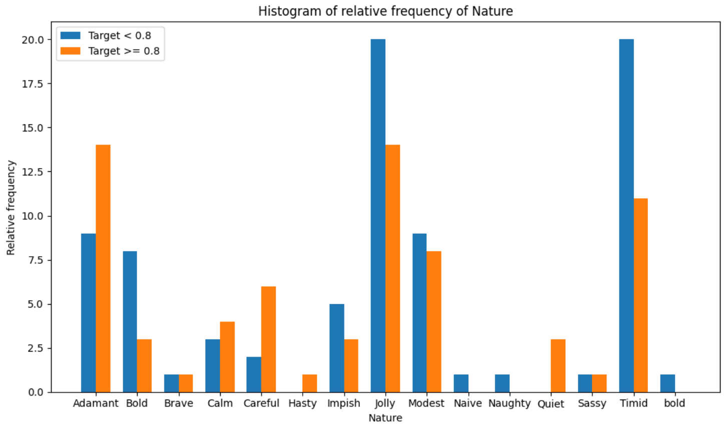
Figure 4: Cumulative distribution of various categorical features (`items`, abilities, `move effects`, and `nature`) for targets. These charts provide insights into the prevalence of different categories and their potential impact on the target variable.



(a) `Ability` percentage for targets, comparing the distribution of abilities between Pokémon with `Target` ≥ 0.8 and those with `Target` < 0.8.



(b) `Item effect` percentage for targets, showing the distribution of item effects between Pokémon with `Target` ≥ 0.8 and those with `Target` < 0.8.



(c) `Nature` percentage for targets, illustrating the distribution of different natures between Pokémon with `Target` ≥ 0.8 and those with `Target` < 0.8.

Figure 5: Various visualizations of categorical variables and their distributions in relation to the target variable.

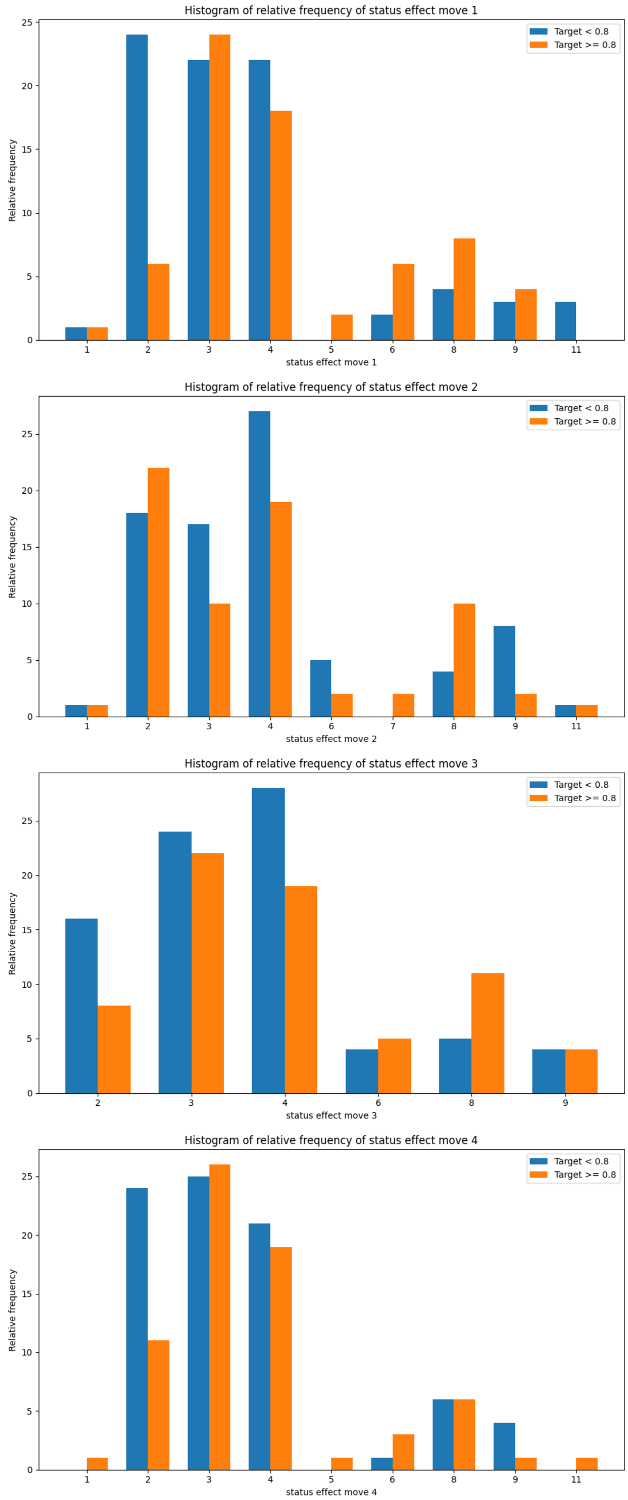


Figure 6: `Status effect move` percentage for targets, comparing the distribution of status effects for the first move between Pokémon with `Target` ≥ 0.8 and those with `Target` < 0.8.

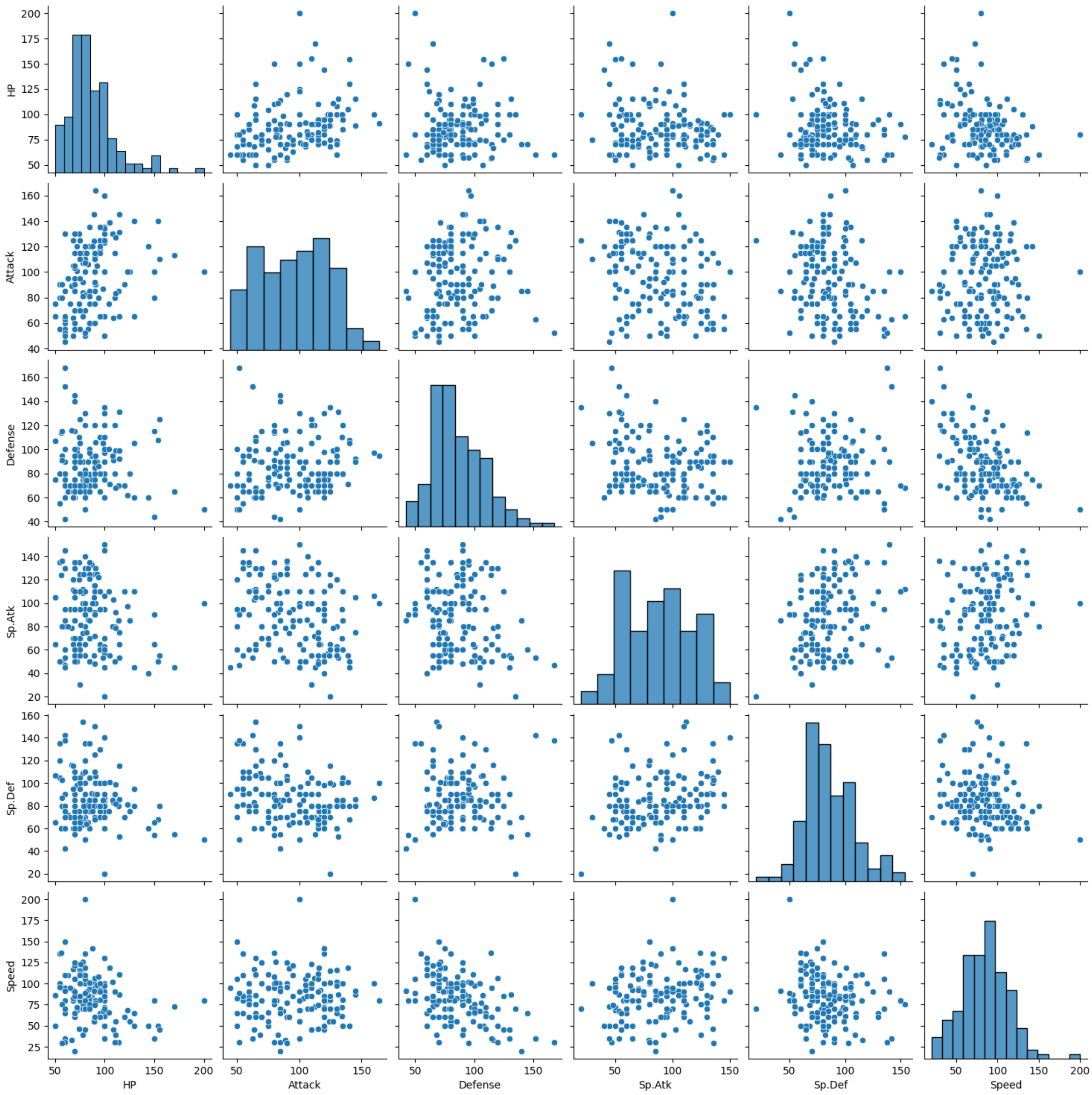


Figure 7: Pair plot showing the relationships between various numerical features such as `HP`, `Attack`, `Defense`, `Sp. Atk`, `Sp. Def`, and `Speed`. This plot helps in visualizing the pairwise correlations between these features.

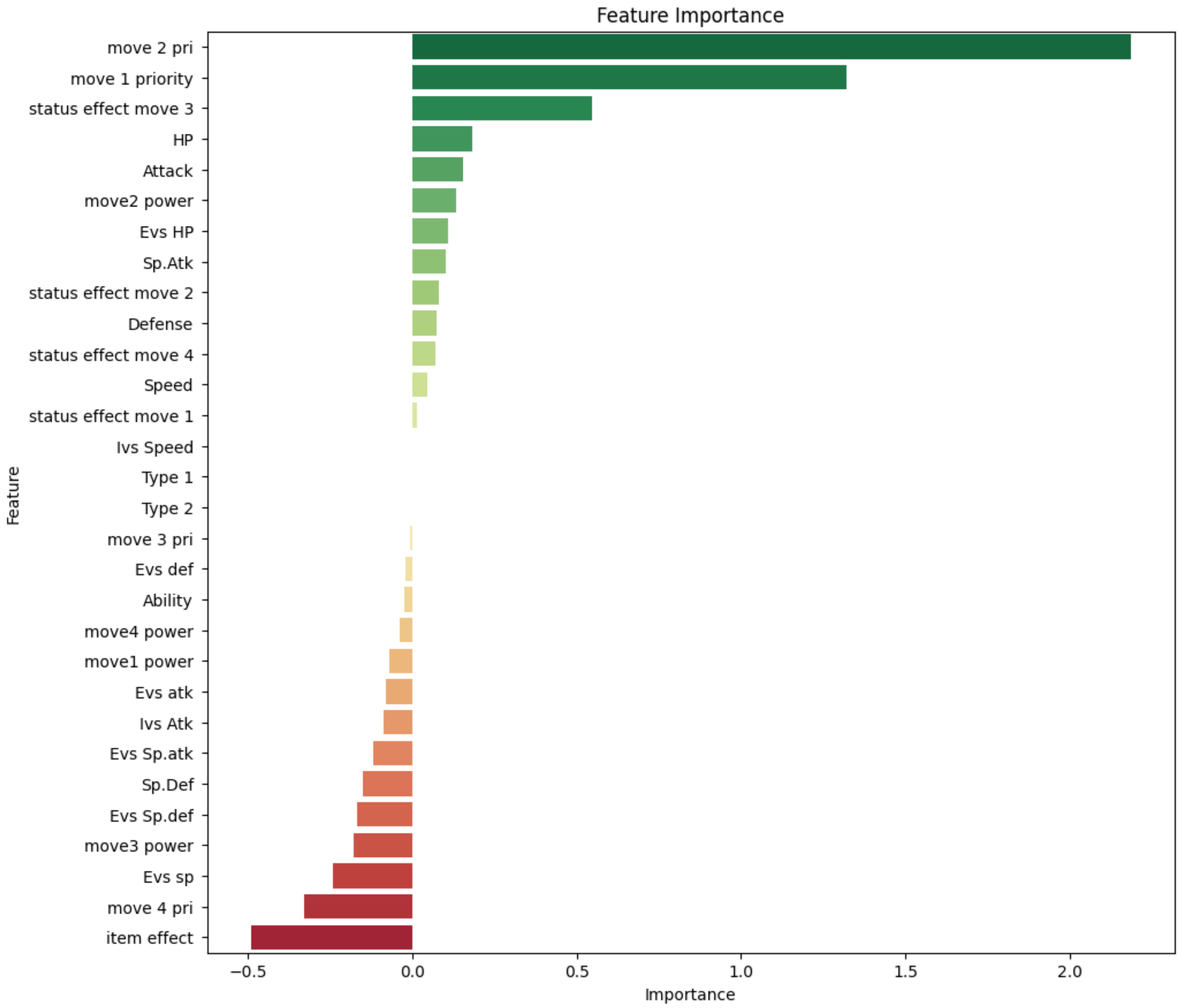


Figure 8: Feature importance analysis using `LIME`. This plot shows the most influential features as determined by `LIME`, with green bars indicating features that positively influence the prediction and red bars indicating features that negatively influence the prediction.

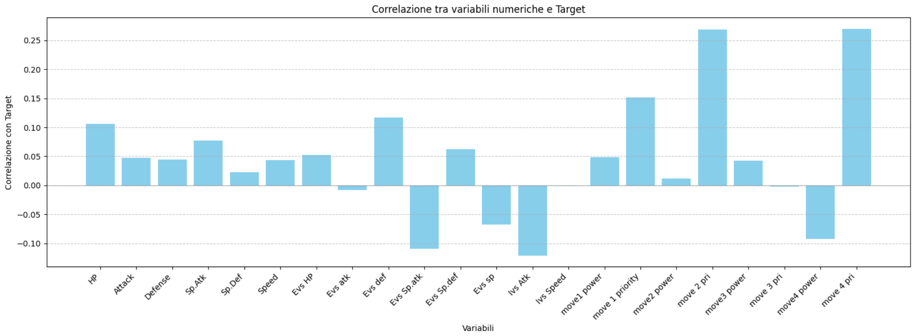


Figure 9: Correlation between each numerical variable and the Target

.2 3D Graphs

We also generated interactive 3D graphs to explore the relationships between various combinations of features and the target variable.

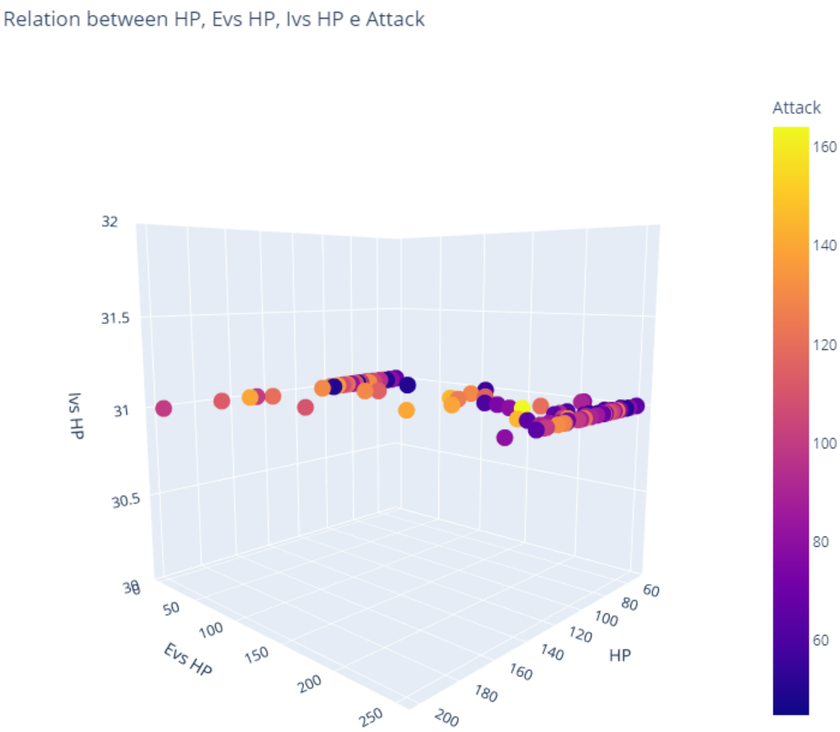


Figure 10: Relation between `HP`, Evs HP, Ivs HP, and `Attack`

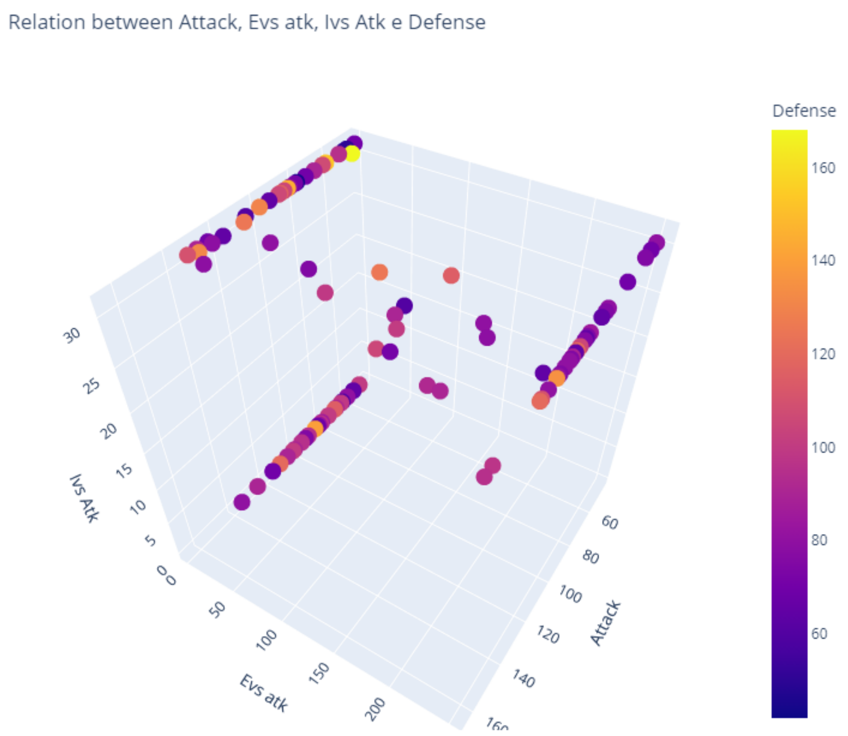


Figure 11: Relation between `Attack`, Evs Atk, Ivs Atk, and `Defense`

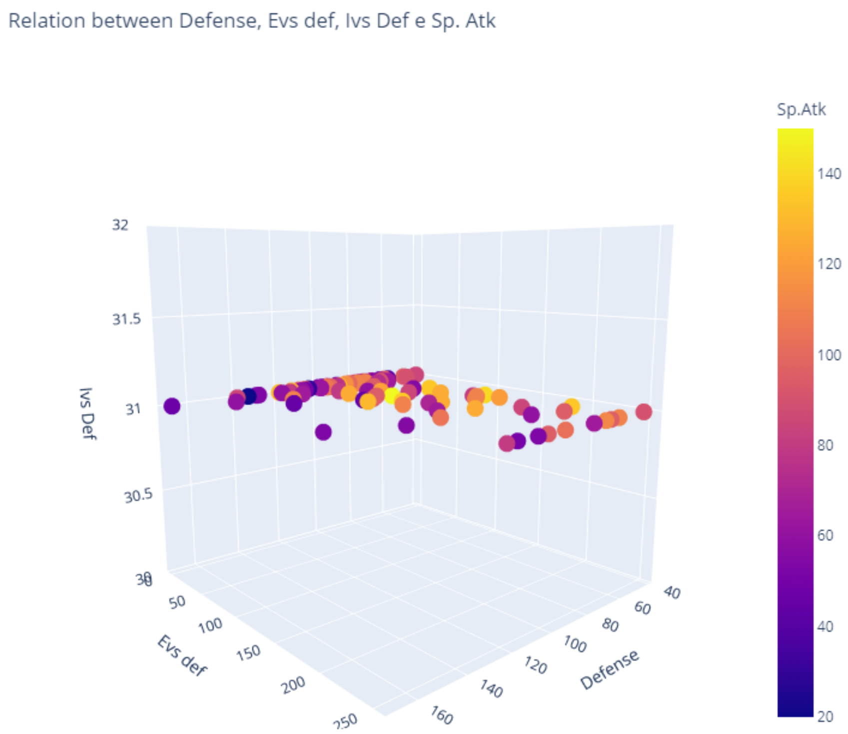


Figure 12: Relation between `Defense`, Evs Def, Ivs Def, and Sp. Atk

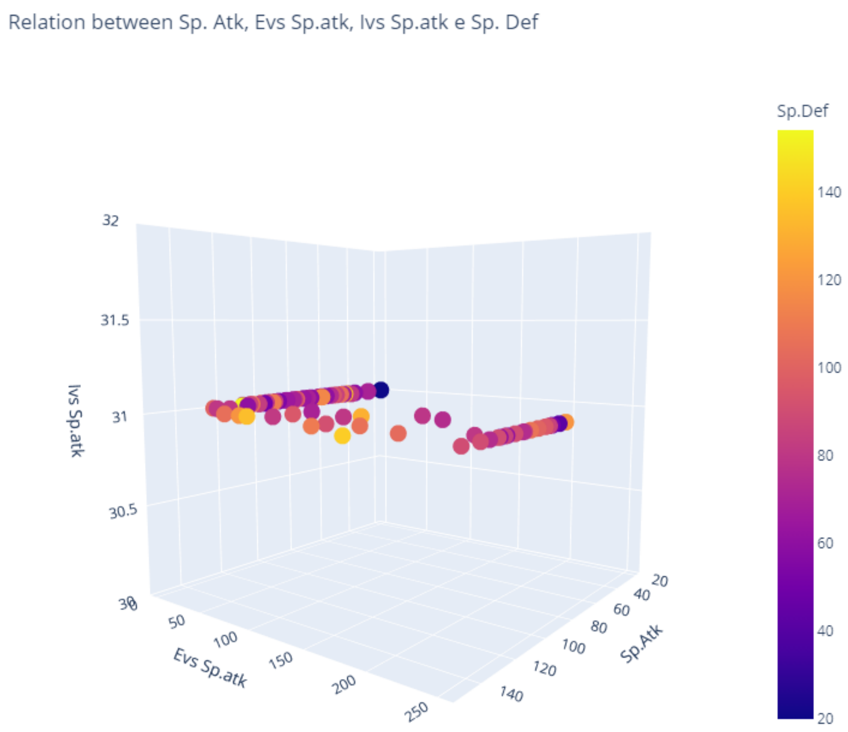


Figure 13: Relation between Sp. Atk, `Evs Sp.Atk`, `Ivs Sp.Atk`, and `Sp.Def`

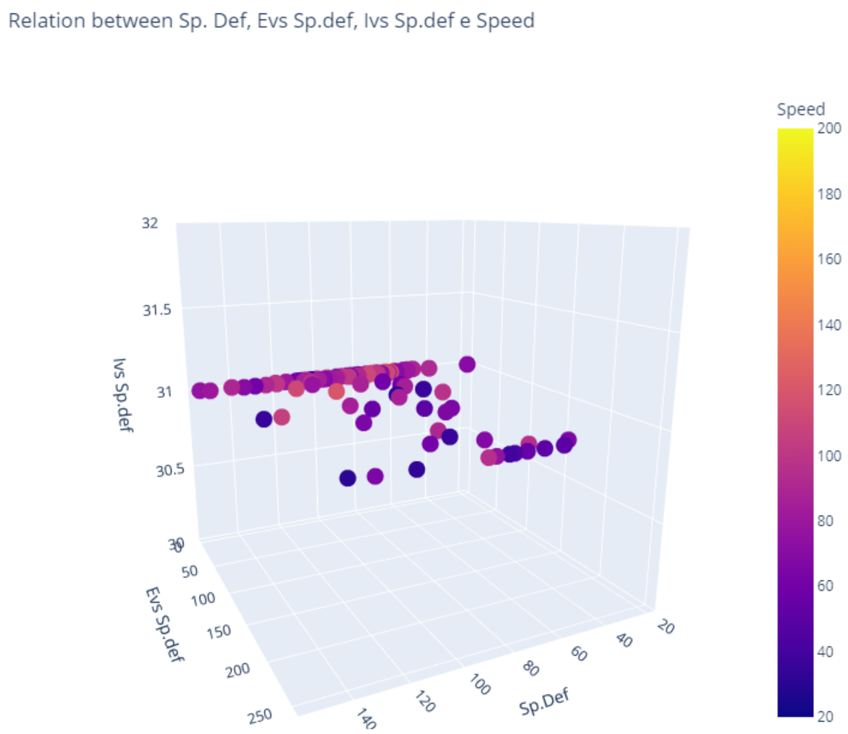


Figure 14: Relation between `Sp.Def`, `Evs Sp.Def`,`Ivs Sp.Def`, and `Speed`

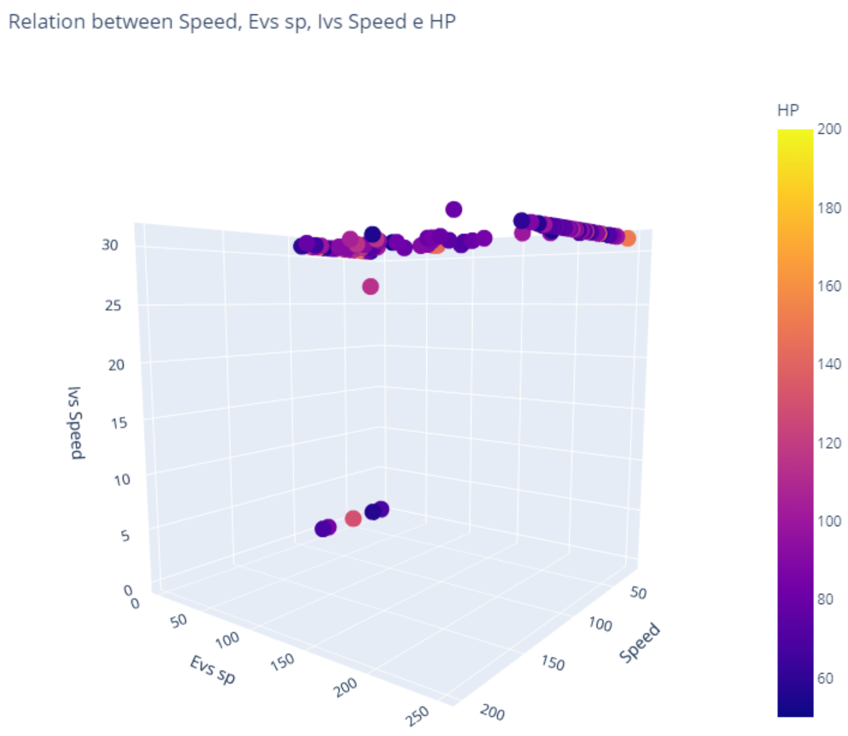


Figure 15: Relation between `Speed`, `Evs Sp`, `Ivs Speed`, and `HP`