

An Analysis of Competitive Pokémon Usage on Pokémon Showdown

JSC370 - Final Project (<https://github.com/tirangol/JSC370-Project>)

Richard Yin

30 April 2025

Introduction

Pokémon (or Pokemon) is a role-playing video game series wherein player characters travel across a fictional region, capturing and training teams of creatures called Pokemon to use in battles against other Pokémon and Pokémon trainers to accomplish various objectives. Central to its gameplay is the turn-based battle system which has fostered a wide-reaching competitive scene, with tournaments like the Pokemon World Championships boasting prize pools of over \$2,000,000 USD (The Pokémon Company, 2025).

There are many competitive formats to a Pokemon battle with various rules and clauses. In the common singles format, two opposing trainers field 1-6 Pokemon, which possess:

- An ability
- A level (which is usually set to the same for all Pokemon)
- An optional held item
- 1-4 unique moves
- 1-2 unique types (there are 18 total types, see Figure 1)
- Base statistics for hit points (HP), attack, special attack, defence, special defence, and speed.

As a battle begins, trainers send out a single Pokemon. Each turn, players can use a move from their current Pokemon or switch to a different Pokemon; they are unaware of their opponent's choice during this time. Once both sides have decided to act, events play out as such: switching occurs first, followed by moves with higher priority levels, then finally the Pokemon with higher speed acts first. If a Pokemon loses all of its HP ("faints") at the end of a turn, its trainer sends out another Pokemon and the next turn begins; if no more Pokemon can stand, the trainer loses the battle.

To win a battle, it is important for competitive players to build teams with a wide range of types and moves rather than simply those with the highest base statistics, since nuances in moves and Pokemon typing add significant complexity to the battle system. For instance, moves are labelled by types and may deal extra ("super effective") or reduced damage based on the defending Pokemon's type(s) (see Figure 1 below). Moves also have base power statistics which are amplified if the move type matches the user (called same-type attack bonus, or STAB). "Physical" moves use the attacking Pokemon's attack against the defender's HP and defence to calculate damage, while "special" moves use special attack against HP and special defence. Finally, non-damaging moves have a variety of effects like temporarily changing a Pokemon's stats, healing, causing weather or status conditions (e.g. hail, rain, poison, burn) which affect Pokemon on the field, and creating hazards that damage or weaken Pokemon upon switching in.

Defender	Normal	Fire	Water	Grass	Electric	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon	Dark	Steel	Fairy
Attacker	Normal													1/2	0		1/2	
Fire	1/2	1/2	2		2							2	1/2		1/2		2	
Water	2	1/2	1/2				2					2		1/2				
Grass	1/2	2	1/2				1/2	2	1/2		1/2	2		1/2		1/2		
Electric		2	1/2	1/2				0	2						1/2			
Ice	1/2	1/2	2			1/2		2	2						2		1/2	
Fighting	2				2		1/2		1/2	1/2	1/2	2	0		2	2	1/2	
Poison				2			1/2	1/2				1/2	1/2			0	2	
Ground		2	1/2	2			2			0		1/2	2				2	
Flying			2	1/2		2					2	1/2						1/2
Psychic					2	2				1/2						0	1/2	
Bug		1/2		2			1/2	1/2	1/2	2				1/2		2	1/2	1/2
Rock		2				2	1/2		1/2	2		2					1/2	
Ghost	0										2			2		1/2		
Dragon															2		1/2	0
Dark							1/2				2			2		1/2		1/2
Steel		1/2	1/2		1/2	2						2					1/2	2
Fairy		1/2					2	1/2							2	2	1/2	

Figure 1. Pokemon type chart. Cell numbers (blank = 1) are damage multipliers for one type against another (e.g. the cell with row Fire and column Grass is 2, so Fire does x2 damage on Grass types). If a Pokemon has two types, the multipliers for both types are used (e.g. Fire does x2 damage on Grass types and Steel types, so it does x4 damage on Steel/Grass types). Source: [Wikipedia](#)

This project aims to uncover insights into the effectiveness of a Pokemon in a competitive setting by answering the research question, “**What properties in a Pokemon most significantly influence its usage rate in competitive singles battles?**” Generation 7, the Alola region introduced by Pokemon Sun/Moon, was selected as the study’s main focus because it has the largest pool of playable Pokemon. Furthermore, this analysis only considers Pokemon, base statistics, types, and moves, ignoring items and abilities. This is because items and abilities have wildly-differing effects, making it difficult to collapse into simpler categories; moreover, items are not unique to a Pokemon, while many abilities can uniquely identify individual Pokemon, which can lead to an overfitted analysis.

Data in this study were collected largely from [Smogon University](#) (or Smogon), a website specializing in competitive Pokemon resources. Smogon has forums, damage calculators, as well as a [Pokedex](#), a table of all Pokemon with base statistics, abilities, movesets, and community team-building strategies for each generation. Smogon also compiles [monthly statistical analyses](#) of Pokemon battles in the form of TXT files, which include raw counts of Pokemon usages as well as common abilities, items, and moves (data from January 2019 was used in this study). These battle statistics are sourced from [Pokémon Showdown!](#), a free Pokemon battle simulator that allows for building Pokemon teams, playing against random opponents online, and competing with a ranked Elo system. Pokemon evolution data (explained later) were collected from [Veekun](#), an independently-created Pokedex website with a more complex search system.

Methods

Data Retrieval and Cleaning

Pokemon and Moves Data

To obtain generic Pokemon data including base statistics, move details, and type effectiveness, Smogon’s generation 7 Pokedex was scraped. This site was selected because its user interface was simplistic, its move descriptions were more standardized and technical than official in-game descriptions, and its Pokemon naming conventions would hopefully match the monthly battle statistics (since Smogon created both resources).

The displayed tabular data was dynamically injected from a Javascript object in a `<script>...</script>` tag and thus did not show up in the fetched HTML. Thus, the object was isolated and parsed into JSON, where it became clear the JSON contained a set of dataframes for Pokemon types and base statistics, moves, items, abilities, and offensive type matchups; only the first two dataframes were important for this study. Nonstandard Pokemon (i.e. fanmade Pokemon from Smogon’s Create-A-Pokemon project) were filtered out, pokemon type was converted from a variable-length array to a string, then the type effectiveness dataframe was refitted into a matrix so both offensive and defensive matchups could be more easily indexed. Features were then extracted from the moves and Pokemon dataframes (see next section).

It should be noted that some Pokemon abilities (e.g. Levitate, Thick Fat) change defensive type matchings, but these are not covered by this analysis because there also exists abilities which can nullify these effects (e.g. Mold Breaker) and the uncertainty around these changes make it difficult to incorporate meaningfully into analysis.

Usage and Common Moves Data

To acquire the response variable – usage – one of Smogon’s monthly battle statistics were scraped. The usage number of a Pokemon in a TXT file is defined as a Pokemon’s number of appearances across all matches in the TXT file’s specified generation, tier, and month; teams with duplicate Pokemon or opposing teams with matching Pokemon can potentially result in > 1 Pokemon appearance per match. Data from January 2019 were selected in hopes that being one of the last months before the next Pokemon generation, the usage rates would be relatively consistent.

This analysis includes the tiers Ubers, OU (overused), UU (underused), RU (rarely-used), NU (never-used), PU, ZU (zero-used), and LC (little cup), which are defined in descending Pokemon strength by Smogon (n.d.) based on usage rates and community adjudication; Pokemon from lower (i.e. worse) tiers may be used in higher tiers, but not vice versa. Tiers were incorporated to create a richer analysis, since without them the vast majority of Pokemon have no utility and thus would never be used at all.

After web scraping the TXT file, regex was used to extract the rows corresponding to names, raw counts, and most common moves (a variable-sized list). Moves representing “Nothing” (blank slots that appear if < 4 moves were chosen for the Pokemon) and “Other” were removed. For each Pokemon, its raw counts were summed across tiers, while the most common moves from the most-played tier were extracted. It was initially attempted to use the most common moves from the Pokemon’s intended tier (specified by the Pokemon dataframe in Table 1), but some Pokemon had no usage for their specified tier, suggesting that the tier list changed since January 2019. This analysis focuses on commonly-used moves instead of entire move pools because the former is more informative of a Pokemon’s competitive strategy.

Pokemon Evolutions

Pokemon has an evolution mechanic in which certain Pokemon can change into new Pokemon under certain conditions (typically when their level reaches a threshold). Evolution chains are at most three Pokemon long;

Pokemon can thus be categorized as first-evolutions, middle-evolutions, final-evolutions, or non-evolving. As a potential confounding variable, evolution was included in this study as a control.

For every possible evolution status, a query was made to fetch the table returned by [Veekun's Pokedex search](#), then the Pokemon names were combined with the evolution status to form a dataframe. To match Smogon naming conventions, many manual corrections were made:

- Adjusting capital letters
- Changing apostrophes to fit UTF encoding
- Removing unique names for “default” forms (e.g. “Normal Deoxys” to “Deoxys”)
- Removing accents (e.g. “Flabébé” to “Flabebe”)
- Hyphenating names (e.g. “Alolan Dugtrio” to “Dugtrio-Alola”).
- For Pokemon with forms for each type (e.g. Arceus-Bug, Arceus-Grass, etc.), a row was created for each unique type

Data Wrangling

The moves dataframe was merged with the top 5 most common moves of each Pokemon; the 5 top was used instead of top 4 in order to account for small variations in play-style. Once feature engineering was performed (see next section), the move features were merged with the Pokemon dataframe and evolution dataframe. Inconsistent naming conventions were identified in the process and manually resolved:

- Hidden Power is a unique move whose type can vary across even the same Pokemon. It was notated in the TXT file by its actual type (e.g. ‘Hidden Power Ice’) but only as ‘Hidden Power’ in the moves dataframe, where new rows for each type had to be manually added.
- Differences in writing conventions existed between dataframe and TXT files for some multi-form Pokemon like ‘NidoranM’ and ‘NidoranF’ (to ‘Nidoran-M’ and ‘Nidoran-F’), as well as ‘Necrozma-Dusk Mane’ and ‘Necrozma-Dawn-Wings’ (to ‘Necrozma-Dusk Mane’ and ‘Necrozma-Dawn Wings’)

The Pokemon dataframe had 988 observations, which became 936 after merging. Three Pokemon genuinely saw zero usages – Tranquill, Gothita, and Scatterbug – so they were excluded from analysis due to a lack of common move data. Most removed Pokemon were either not introduced in the mainline Pokemon game (e.g. Melmetal, Eevee-Starter, Pikachu-Starter) or were a specific Pokemon form. Some forms were only cosmetic in nature (e.g. Pikachu-Hoenn, Pikachu-Unova), so they were merged into the base form; others forms were related to abilities that change a Pokemon’s form in the middle of a game (e.g. Aegislash, Wishiwashi), which can affect base statistics, resulting in extra rows created for this Pokemon’s forms. This analysis did not consider these edge cases for the sake of simplicity.

No outliers and implausible values were detected in this final dataset.

Feature Engineering

Relevant initial features in all the scraped dataframes are shown in Table 1. Features that were manually engineered from the Pokemon dataframe’s features are shown in Table 2. It should be noted that the `best_stat` feature is calculated based on the best statistic out of `atk`, `spa`, `bulk`, `spbulk`, and `spe`:

- ‘spe’ if it is `spe`
- ‘atk’ if it is `atk` or `spa`, where `atk` and `spa` differ by at most 10
- ‘phatk’ if it is `atk` and `atk > spa + 10`
- ‘spatk’ if it is `spa` and `spa > atk + 10`
- ‘bulk’ if it is `bulk` or `spbulk`, where `bulk` and `spbulk` differ by at most 10
- ‘phbulk’ if it is `bulk` and `bulk > spbulk + 10`
- ‘spbulk’ if it is `spbulk` and `spbulk > bulk + 10`
- ‘none’ if the difference between the best and worst stat is at most 10

Table 3 lists the features engineered from the move dataframe Apart from `priority` and `atk_{type}`, the logical-based features used string matching on the move description to filter for for a specific effect.

Table 1. Relevant initial features in the Pokemon (left) and moves (right) dataframe

Feature	Type	Description	Feature	Type	Description
name	chr	Pokemon’s name	name	chr	Move’s name
hp	int	Hit points	category	chr	Either ‘Physical’, ‘Special’, or ‘Non-Damaging’
atk	int	Attack	power	int	Move’s damage, if it is damaging (0 otherwise)
def	int	Defence	accuracy	int	Probability of a move successfully executing
spa	int	Special Attack	priority	int	Move’s priority level (-7 to 7)
spd	int	Special Defence	description	chr	Move’s effects
spe	int	Speed	type	chr	Move’s type
types	chr	Pokemon’s types, comma-separated			
formats	chr	Pokemon’s Smogon tier			

Table 2. Engineered features from the Pokemon dataframe

Feature	Type	Description
total	int	Sum of all base statistics
stdev	int	Standard deviation of all base statistics
bulk	num	$(hp + def) / 2$, a more well-rounded measure of defence
spbulk	num	$(hp + spd) / 2$, a more well-rounded measure of special defence
typecount	factor	Number of unique types, either 1 or 2
best_stat	chr	Estimation of the Pokemon’s primary strength. Can be ‘atk’, ‘phatk’, ‘spatk’, ‘bulk’, ‘phbulk’, ‘spbulk’, ‘spe’, ‘none’
weaknesses	num	Number of defensive weaknesses (where x2, x4)
resistances	num	Number of defensive resistances (where x0.5, x0.25, x0)
is_{type}	logi	Whether the pokemon is {type} (for all 18 types)
def_{type}	factor	The pokemon’s defensive multiplier against {type} (for all 18 types). Can be 0, 0.25, 0.5, 1, 2, 4

Table 3. Engineered features from the moves dataframe. These features apply to the top 5 most common moves of a Pokemon.

Feature	Type	Description
prop_physical	num	Proportion of moves that are physical
prop_special	num	Proportion of moves that are special
prop_nondamaging	num	Proportion of moves that are non-damaging
num_inaccurate	int	Number of moves with accuracy < 100%
priority	logi	If there is a move with priority > 0
stat_change	logi	If there is a non-damaging move that raises/lowers stats
multisetup	logi	If there is a non-damaging move that does something for multiple turns
heal	logi	If there is a move that heals
status	logi	If there is a non-damaging move that inflicts a status effect (i.e. burn, poison, paralysis, sleep, confusion)
hazards	logi	If there is a hazard-related move (hazards damage Pokemon on switch-in)
switch	logi	If there is a move that forces/prevents a Pokemon switch
recoil	logi	If there is a move, typically a powerful move, that also hurts the player
ohko	logi	If there is a move that involves instantly fainting the opponent or player
flinch	logi	If there is a move that can flinch the opponent, skipping their move

Feature	Type	Description
selfweaken	logi	If there is a move, typically a very powerful move, that lowers the player's stats after usage
multihit	logi	If there is a move that hits multiple times
protect	logi	If there is a move that protects the player from something for one turn
atk_{type}	logi	If there is a super-effective move against {type} (for all 18 types)
num_super_effective	int	Number of unique types that can be hit super-effectively from the moveset; a Pokemon's offensive coverage

Data Exploration

The following tools were utilized for data exploration:

- **dplyr** to calculate means and proportions of variable values.
- **ggplot2** to create histograms of Pokemon usages, box plots of usage against multiclass categorical variables (e.g. tier, best stat, number of types), scatter plots of usage against continuous variables (e.g. base statistics), 2D heatmaps/histograms of usage against small count variables (which would otherwise clump together in a regular plot, e.g. number of weaknesses/resistances), and a correlation matrix of base statistics to each other.
- **patchwork** to combine topically-related **ggplot** diagrams into a single figure.
- **kable** to display the dataset and results of ANOVA tests conducted with **anova**.
- **lm** to fit simple linear regression models of base statistics to usage.

Models and Tests

The models used for regression include:

- *Linear Models (LMs)*: The fitting of a line to data that is assumed to follow $Y|X \sim \mathcal{N}(\mu, \sigma^2)$. This response variable was **log(usage)**, equivalent to a generalized linear model with a log-normal distribution.
- *Generalized Linear Models (GLMs) (MASS)*: An extended form of linear regression that allows $Y|X$ to follow any distribution. The quasi-poisson and negative binomial distributions are considered.
- *Generalized Additive Models (GAMs) (mgcv)*: A GLM or LM with an additional non-linear smooth function for select coefficients to allow for fitting non-linear trends.
- *Random Forests (randomForest)*: The averaged predictions of many decision trees, a type of machine learning model that iteratively splits features based on threshold values into “purer”, less varied subgroups in order to make predictions.
- *XGBoost (caret)*: A type of gradient boosting algorithm (which iteratively builds decision trees that correct the previous iteration's errors) that uses regularization and pruning to increase performance.

The decision tree models are trained on the same 80/20 train/test split. The random forest consisted of 50 trees, while XGBoost was trained on 100 rounds of max depth 3, with $\eta = \gamma = 0.5$ and 10-fold cross-validation. To compare the models across training and testing set, root mean squared error (RMSE) is used, defined as $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2}$, where y_i and t_i are the model predictions and actual logged usages for Pokemon i .

Plots are created of model coefficients and residuals as well as feature importance plots for the decision tree models. p -values are used to determine significance and decide the features to keep in the models.

Results

Exploratory Data Analysis

Usage

The distribution of usages is extremely right-skewed, with a mean of 37,986 far exceeding the median of 8511, and a logarithmic-scale histogram of usages appearing vaguely symmetrical (Figure 2). Being an example of count data, it seems appropriate to model the counts as a Poisson distribution (where $E[X] = \text{Var}[X]$) or an overdispersed distribution (to account for extra variance) like the quasi-poisson or negative binomial. However, the distribution is also vaguely normal, with very high counts and no zeroes, which may make the log-normal distribution a reasonable approximation.

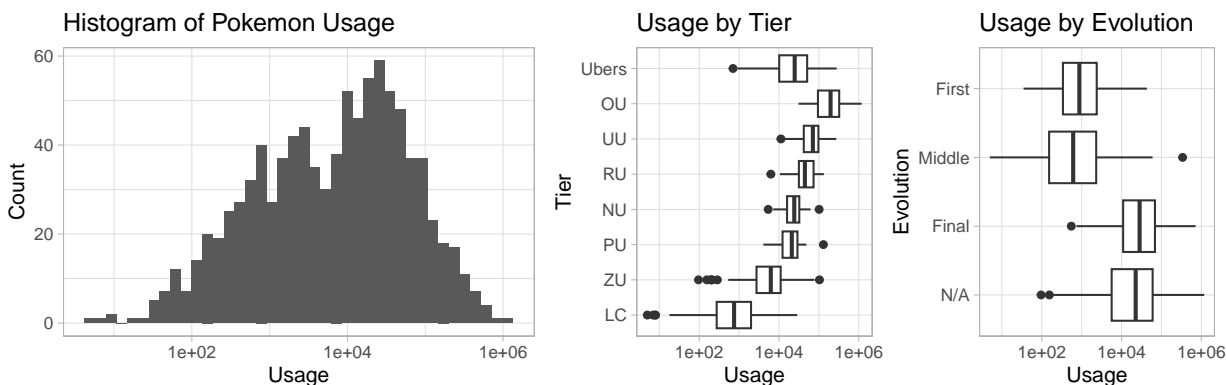


Figure 2. Histogram of Pokemon usage in the dataset independently (left), by tier (middle), and by evolution stage (right). The distribution of Pokemon is extremely right-skewed, with Pokemon in higher tiers being more commonly used. Non-evolving or final-evolution Pokemon are used more than not fully-evolved Pokemon.

Table 4. Most and least commonly-used Pokemon in the dataset.

Name	Types	Total	Tier	Evolution	Usage
Landorus-Therian	Ground,Flying	600	OU	Non-Evolving	1177996
Greninja-Ash	Water,Dark	640	OU	Final Stage	717556
Ferrothorn	Grass,Steel	489	OU	Final Stage	684332
Heatran	Fire,Steel	600	OU	Non-Evolving	671964
Tapu Koko	Electric,Fairy	570	OU	Non-Evolving	592173
...
Skiploom	Grass,Flying	340	LC	Middle Stage	18
Luxio	Electric	363	LC	Middle Stage	8
Silcoon	Bug	205	LC	Middle Stage	8
Cascoon	Bug	205	LC	Middle Stage	7
Spewpa	Bug	213	LC	Middle Stage	5

Pokemon usage is also heavily dependent on tier and evolution: mid-to-high-level tiers like OU, UU, and RU are the most common tier by far and low-level tiers like LC and ZU are the least common, while fully-evolved or non-evolving Pokemon are more popular than not fully-evolved Pokemon. Though a Pokemon's tier and evolution does not affect how it battles, the model may need to adjust for these confounders. This trend also manifests in Table 4, which lists the dataset's extremes. The top 5 most common Pokemon are all fully-evolved or non-evolving dual-types in OU with high totals, while the top 5 least common Pokemon are monotypes with low totals in LC. This trend is further explored in Figure A1/A3 in the report website, which show positive correlations between tier, evolution stage, total base statistics, and usage.

Base Statistics

Every base statistic appears to follow the same trend (Figure 3), being mostly distributed between 40 and 120 with more large outliers than small outliers. Plotted against usage, there is a noisy but somewhat nonlinear positive correlation that is roughly linear when the statistic is < 100 , but flattens as the statistic increases, suggesting a GAM could model these variables well. Usage also correlates positively with the total and standard deviation of all base statistics.

Fitting a linear model between logged usage and each base statistic, all slope parameters are significant ($p < 2 \cdot 10^{-16}$), with R^2 varying from 0.16 to 0.28, confirming the observed positive correlation.

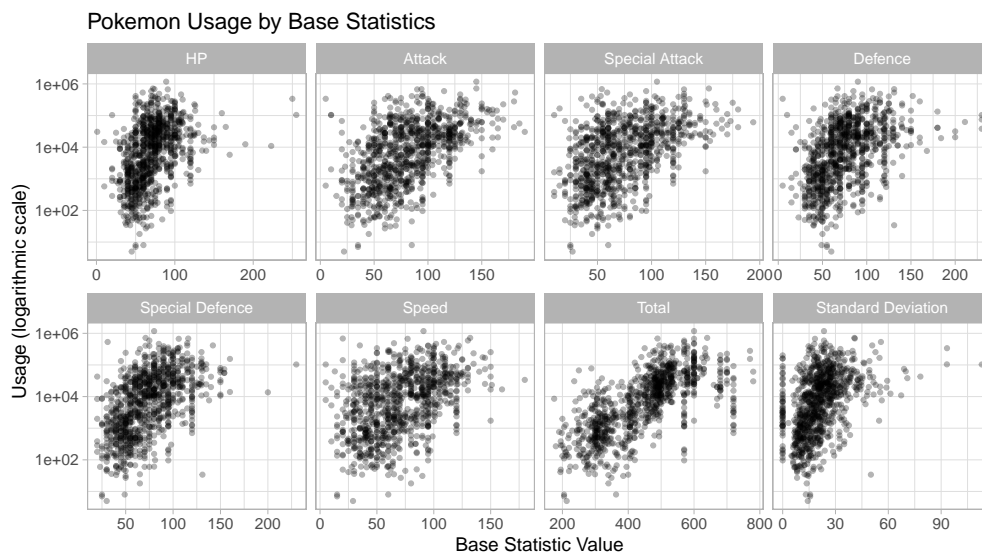


Figure 3. Scatter plots of a Pokemon's base statistics against usage. All follow a similar noisy but positive correlation.

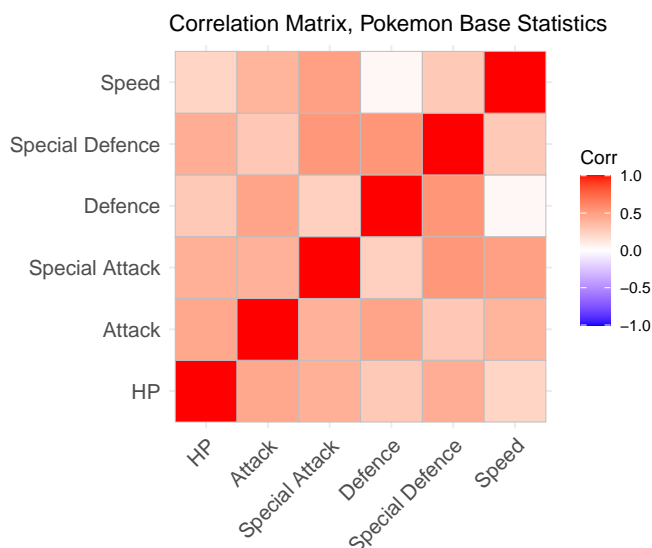


Figure 4. Correlation matrix of all Pokemon base statistics. All variables are correlated positively with each other, except for speed and defence which have no correlation.

A possible reason for the identical trends across base statistics could be that they are highly-correlated. Plotting a correlation matrix in Figure 4 confirms this hypothesis, which shows positive correlations between

every value with each other (with the exception of speed to defence, with no relationship). This correlation is likely explained by the fact that evolutions almost always result in increases to base statistics, raising the need to control for evolution status.

Out of the 936 Pokemon in the dataset, 497 (53%) can be said to be attack-based, 211 (23%) are defence-based, 168 are speed-based (18%), and 60 specialize in none (6%); attackers and defenders are split 75/254/168 and 113/57/41 as general/physical/special attackers and defenders respectively. Analyzed against usage (Figure 5 and A2), it appears that well-rounded and fast Pokemon are the least likely to be used. Special-based Pokemon are more likely to be used than physical-based Pokemon, and physical/special defenders/attackers are more likely to be used than general defenders/attackers. However, the box-and-whisker plots are wide and the categories significantly overlap, raising the question of statistical significance. An ANOVA fit between logged usage and `best_stat` reveals a p -value of 1.813×10^{-5} , confirming there is potentially some evidence for predictive power in this feature.

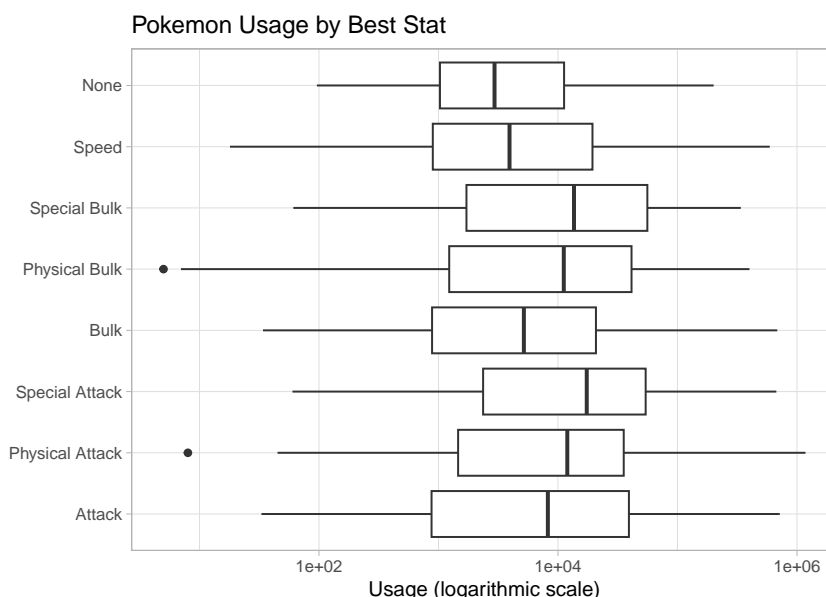


Figure 5. Box and whisker plots of a Pokemon’s best stat type against usage. Special-based Pokemon are used most, followed by physical-based Pokemon, followed by general attackers/defenders, followed by well-rounded and speed-based Pokemon.

Typing

Figure 6 shows various 2D histograms and a box-whisker plot to assess the predictive power of shallow type-related features. It is hypothesized that Pokemon with less weaknesses, more resistances, and more super-effective coverage would be more useful competitively and thus used more often. However, apart from the number of resistances, there is little to no correlation between usage and coverage, or usage and weakness count. Perhaps poor coverage is not problematic for defensive-based Pokemon and many weaknesses is not problematic for attack-based Pokemon. It may be necessary to combine `best_stat` with model weaknesses and coverage to properly model the hypothesized effect.

Finally, dual-types are more-used than monotypes, which is likely another byproduct of evolution as Pokemon commonly gain an additional type as they evolve.

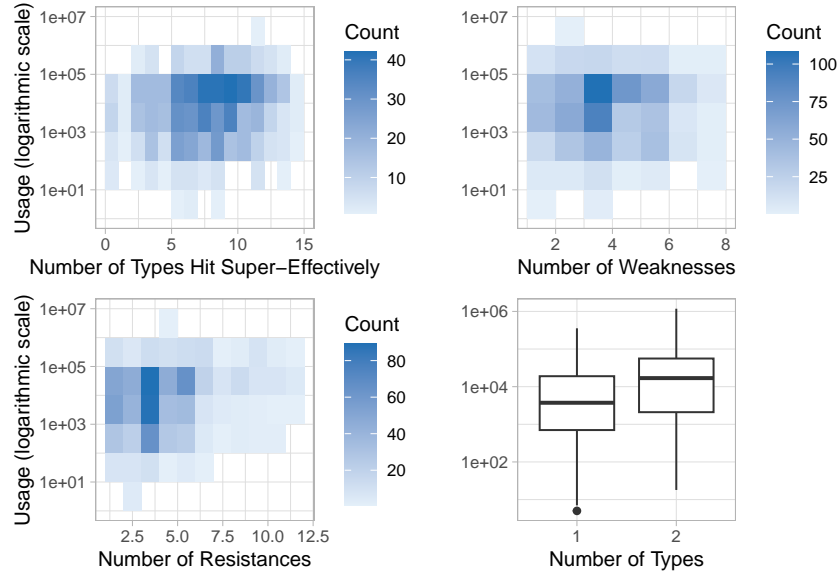


Figure 6. 2D histogram plots and box-plot of usage against shallow type-related features: number of types, weaknesses, resistances, and types that the Pokemon’s top 5 most common moves can hit super-effectively. Only the type and resistance count correlate positively with usage.

Table 5. Significance tables for three separate ANOVA tests for logged usage against the presence of a type in a Pokemon (left), the defensive type multiplier against an attacking type (middle), and the offensive type multiplier against a defending type (right).

Feature	F value	Pr(>F)	Feature	F value	Pr(>F)	Feature	F value	Pr(>F)
is_normal	6.9345	0.0086	def_normal	7.2390	0.0001	atk_normal	13.5078	0.0003
is_fire	1.2131	0.2710	def_fire	3.0437	0.0166	atk_fire	6.5559	0.0106
is_water	0.2633	0.6080	def_water	1.0281	0.3917	atk_water	5.3301	0.0212
is_grass	11.8726	0.0006	def_grass	4.1875	0.0023	atk_grass	0.0379	0.8458
is_electric	0.2435	0.6218	def_electric	5.3380	0.0001	atk_electric	12.6499	0.0004
is_ice	0.2463	0.6198	def_ice	3.3818	0.0093	atk_ice	8.2042	0.0043
is_fighting	8.4880	0.0037	def_fighting	1.8527	0.1168	atk_fighting	1.9805	0.1597
is_poison	0.1282	0.7204	def_poison	2.8561	0.0145	atk_poison	0.2963	0.5864
is_ground	1.8917	0.1693	def_ground	3.9528	0.0015	atk_ground	3.5091	0.0614
is_flying	3.4481	0.0636	def_flying	8.0489	0.0000	atk_flying	3.6751	0.0555
is_psychic	3.8630	0.0497	def_psychic	4.9282	0.0002	atk_psychic	3.7264	0.0539
is_bug	11.2178	0.0008	def_bug	2.3008	0.0571	atk_bug	1.5447	0.2142
is_rock	0.0128	0.9100	def_rock	1.1786	0.3187	atk_rock	0.2038	0.6518
is_ghost	2.7052	0.1004	def_ghost	1.3071	0.2656	atk_ghost	3.0885	0.0792
is_dragon	9.4229	0.0022	def_dragon	0.8217	0.4820	atk_dragon	5.4831	0.0194
is_dark	14.8223	0.0001	def_dark	0.5565	0.6439	atk_dark	4.1375	0.0422
is_steel	49.0728	0.0000	def_steel	1.6276	0.1652	atk_steel	1.1909	0.2754
is_fairy	34.8815	0.0000	def_fairy	1.5069	0.1981	atk_fairy	0.0775	0.7807

Pokemon typing can be additionally be analyzed on a per-type level. Table 5 shows the results of three ANOVA tests performed on features for the presence of a type, defensive matchups against a type, and offensive coverage against a type, indicating that to a 5% level,

- 10 types’ presence/absence affects usage: normal, grass, fighting, flying, psychic, bug, dragon, dark, steel, fairy.
- Defence against 7 types affects usage: normal, grass, electric, poison, ground, flying, psychic

- Offense against 11 types affects usage, including all types except grass, fighting, poison, bug, rock, steel, and fairy.

At this level of granular detail, it is difficult to explain why specific type matchups affect usage more. Densities of logged usage are plotted in Figures [B1](#)/[B2](#)/[B3](#) to further examine type-level effects, where it appears that the presence of steel, fairy, and dragon types, resistance to dragon types, and coverage against electric types correlate with higher usage. To prevent overfitting to type combinations, many of which may uniquely identify a Pokemon (see Figure [C1](#)) only one the three sets of variables should be used.

Move Details

Specific move mechanisms in a Pokemon’s common movepool are analyzed by logged usage using ANOVA. Table 6 implies that special offensive moves, multi-turn setup moves, status-inflicting moves, hazard-related moves, switching-related moves, self-weakening moves, and multi-hitting moves significantly impacts usage.

Table 6. ANOVA test for a Pokemon’s usage against whether their common movepool contains moves with specific properties.

Feature	Df	Sum Sq	Mean Sq	F value	Pr(>F)
prop_physical > 0	1	11.8101	11.8101	2.4112	0.1208
prop_special > 0	1	36.6091	36.6091	7.4742	0.0064
prop_nondamaging > 0	1	2.0897	2.0897	0.4266	0.5138
num_inaccurate > 0	1	7.6851	7.6851	1.5690	0.2107
priority	1	0.2023	0.2023	0.0413	0.8390
stat_change	1	0.6772	0.6772	0.1382	0.7101
multisetup	1	37.0086	37.0086	7.5557	0.0061
heal	1	5.8081	5.8081	1.1858	0.2765
status	1	21.6991	21.6991	4.4301	0.0356
hazards	1	21.9889	21.9889	4.4893	0.0344
switch	1	26.6060	26.6060	5.4319	0.0200
recoil	1	1.1408	1.1408	0.2329	0.6295
ohko	1	0.6372	0.6372	0.1301	0.7184
flinch	1	6.5915	6.5915	1.3457	0.2463
selfweaken	1	76.0087	76.0087	15.5180	0.0001
multihit	1	28.8187	28.8187	5.8836	0.0155
protect	1	4.8617	4.8617	0.9926	0.3194
Residuals	917	4491.5540	4.8981	NA	NA

Model

Distribution

Three linear models with different data distributions were tested – log-normal, quasi-poisson, and negative binomial. Each model incorporated base statistics (totals, standard deviations, best base statistic type) and shallow type information (number of types, weaknesses, resistances, and offensive type matchups), with coefficients displayed in Figure 7. The quasi-poisson model and negative binomial model have additional dispersion parameters θ of 25171.35 (the estimated $\frac{\text{Var}[Y|X]}{\text{E}[Y|X]}$) and 1.056 (where $\frac{1}{\sqrt{\theta}} \approx 0.973$ is the relative excess standard deviation) respectively, providing strong evidence of overdispersion.

All three models generally agree on the effect of statistically significant terms. Adjusting for tier and evolution status, dual-types have a higher rate of usage (≈ 1.2 times) than monotypes. Speed-based Pokemon now have the highest usage rates, while well-rounded Pokemon have the lowest usage rates. Usage increases

with less weaknesses and more resistances, as well as with increasing standard deviation and total total base statistics, showing that specialized Pokemon are more-used competitively.

It should also be noted that the negative binomial model suffers from poor optimizer convergence, raising concerns over the validity of the results.

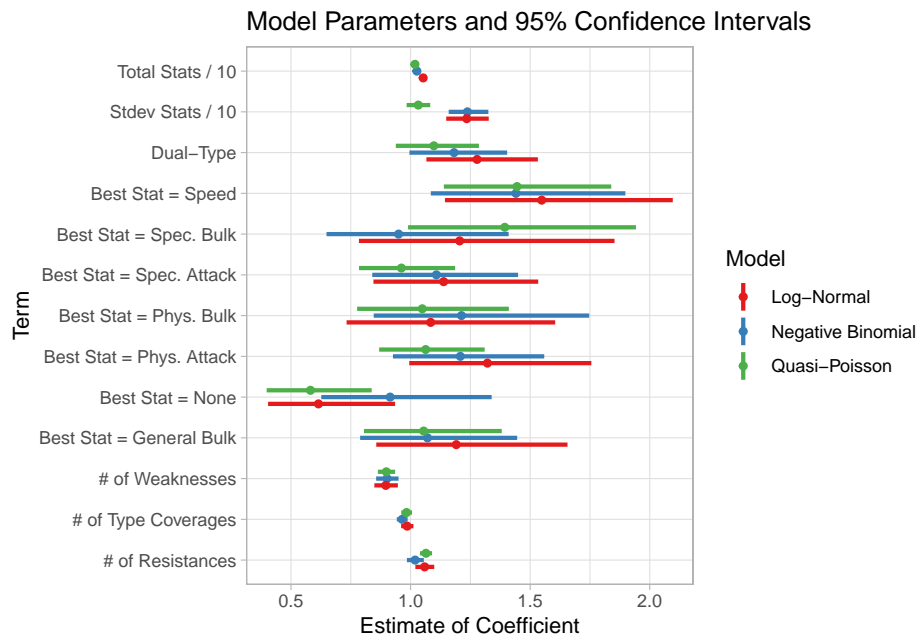


Figure 7. Plot of coefficient estimates and 95% confidence intervals for GLMs with log-normal, quasi-poisson, and negative binomial distributions and a logarithmic link function, using features based on base statistics and type information, and adjusting for evolution status and tier. The numbers represent rate ratios of a Pokemon being used relative to a monotype, general attack-based Pokemon.

The residual plots in Figure 8 reveal that competitive tiers form the basis of model predictions. Pokemon are clustered by tier into diagonal lines, with small amounts of variance from the features of interest, whose effects are weak. The negative binomial’s residuals do not appear unusual, suggesting the numerical instability issue was minor. Thus, in the absence of strong evidence for any model, the negative binomial was chosen as the most conventional option to model this overdispersed count data.

Residual Plots of GLMs of Competitive Pokemon Usage

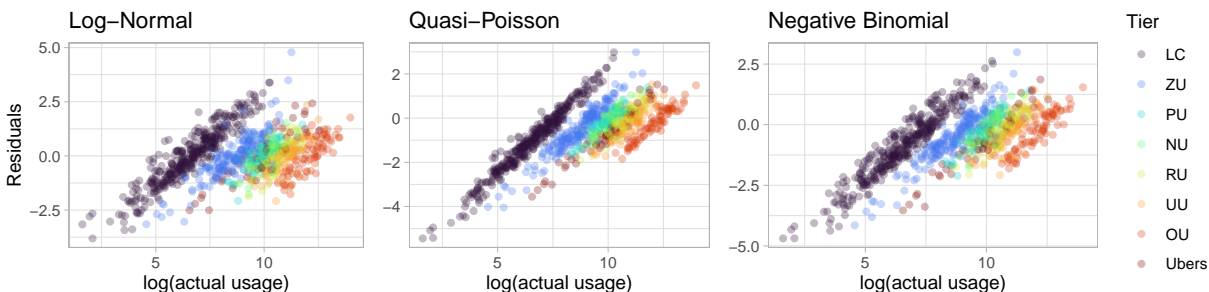


Figure 8. Plots of residuals (= actual logged Pokemon usage – predicted logged Pokemon usage) for the models in Figure 7, coloured by tier.

Base Statistics

To find the most informative base statistics, the negative binomial model from the previous section was modified to replace the total base statistics with attack, special attack, bulk, special bulk, and speed, as well as with the bulk parameter. Table 7 shows these new coefficients, where speed, special attack, and special defence show significant (to a 5% level) positive correlations with usage. Bulk does not appear to correlate significantly with usage.

Table 7. Estimated multiplicative change to a Pokemon’s rate of usage per 10-point increase in individual base statistic type, with p -values, after adjusting for evolution status and tier, for two negative binomial GLMs with one using default base statistics (left) and the other using the bulk feature (right).

Term	Estimate	p-value	Term	Estimate	p-value
Speed	1.0464	0.01453	Speed	1.0480	0.01191
Attack	1.0261	0.22663	Attack	1.0209	0.33320
Special Attack	1.0663	0.00157	Special Attack	1.0670	0.00145
Defence	0.9965	0.85308	Bulk	0.9676	0.36824
Special Defence	1.0454	0.03249	Special Bulk	1.0376	0.31934
HP	0.9731	0.11933			

Based on the nonlinear positive correlations observed in Figure 3, two GAMs (Figure 9) were constructed:

- A “shallow model” with smooth functions for the total and standard deviation
- A “deep model” with smooth functions for all six base statistics and standard deviation.

For the shallow model, total statistics behave similarly to Figure 3, but usage consistently increases with standard deviation. For the deep model, attack and special attack correlate positively with usage, while defence and special defence have no clear correlation. HP appears to correlate negatively with usage, while speed appears to have an optimal region around 80-100, but no positive correlation as noticed in Table 7, which is difficult to explain.

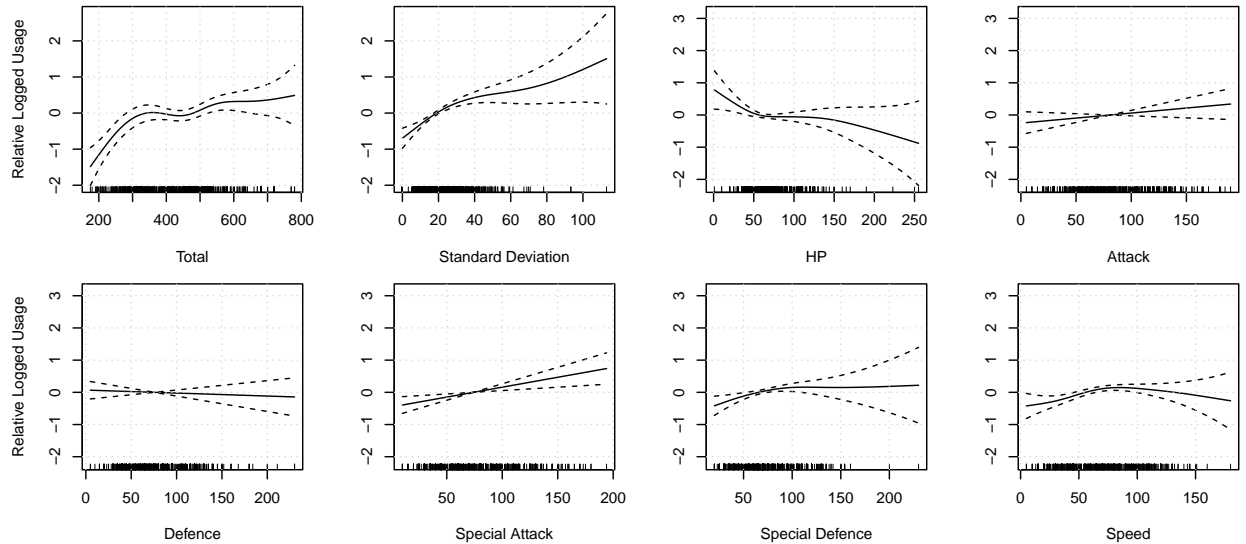


Figure 9. Trendlines of a “shallow GAM” for the total and standard deviation of base statistics (top-left quarter), and a “deep GAM” for individual base statistics and the standard deviation (bottom, top-right quarter). The GAMs have the same features as models in Figure 7.

Types

Three sets of type-level boolean features were separately added to the deep GAM model: the presence of, super-effective coverage of, and resistance to a type. Note that one type was excluded as it was redundant information (since Pokemon must have at least one type) that created perfect multicollinearity.

Figure 10 shows that after adjusting for evolution status, tier, number of weaknesses and resistances, and base statistics, the presence of steel, ground, poison, fire, fighting, fairy, and dark types correlate positively with usage to a significant level, while the presence of bug types contributes negatively to usage. The trend is especially prominent for steel types, which is reasonable as it is the best defensive type (see Figure 1). Resistance to bug, flying, grass, and water (i.e. bug or grass types) negatively correlates with usage, albeit barely significantly ($p \approx 0.05$). Type coverage does not appear to correlate with usage apart from psychic type coverage, which has a marginal negative effect. The results suggest that type presence is the most informative type information.

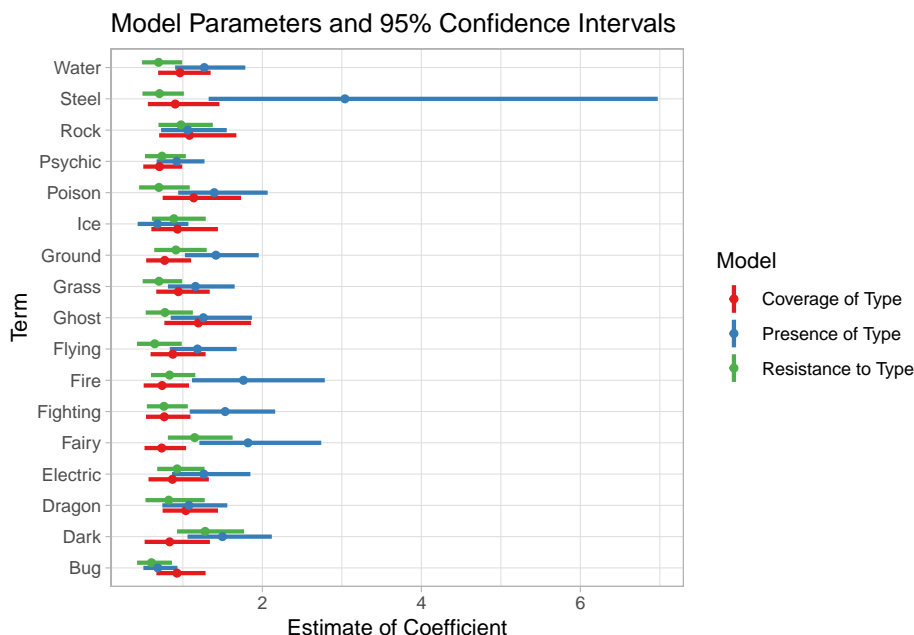


Figure 10. Plot of estimates and 95% confidence intervals for a subset of coefficients for three deep GAM models with additional boolean features for either the presence of, offensive coverage of, or resistance to each Pokemon type except for the normal type. The coefficients represent the rate ratios of a Pokemon’s usage, given the Pokemon is/covers/resists a specific type.

Moves

A set of move-based features were added to the type-presence GAM model, with coefficients for the new features displayed in Table 8. Non-damaging and priority moves significantly contribute to greater Pokemon usage. This is reasonable as priority moves allow slow Pokemon to overcome low speeds, and more non-damaging moves imply a Pokemon has more possible play-styles other than offense/defence. On the other hand, the presence of inaccurate, stat-changing, and multiturn setup moves negatively impact usage. Inaccuracy is often added to balance powerful moves – out of 115 moves with power ≥ 100 , 54 of them non-perfect accuracies – so it seems that trading reliability for power is unpopular. Moves affecting base statistics are similarly risky as they sacrifice a turn for a future advantage that is contingent on the attacker not switching out (which removes the base statistic change); however, multiturn setup moves have effects persisting across turns, making them less risky in theory, so it is interesting to see that Pokemon with these moves are more popular.

Table 8. Estimated multiplicative change to a Pokemon’s rate of usage per the presence of a move with a specific effect (or the proportion of non-damaging moves and number of inaccurate moves), with p -values, for the type-presence GAM in Figure 10 with additional move-based features.

Move Feature	Coefficient	p-value
Prop. of Non-Damaging	2.50931	0.00139
# of Inaccurate	0.86875	0.00321
Has Priority	1.23449	0.01738
Has Stat Change	0.69041	0.00001
Has Multiturn Setup	0.70318	0.01897
Has Heal	1.11196	0.22022
Has Status Effect	1.01111	0.91223
Has Hazards	0.99312	0.93901
Has Switch	1.10583	0.23882
Has Recoil	1.16218	0.21536
Has OHKO	0.75481	0.05283
Has Flinch	0.89594	0.21228
Has Self-Weaken	1.00331	0.97597
Has Multihit	0.92123	0.63463
Has Protect	0.82987	0.19858

Decision Trees

Many of the tested covariates were binary in nature and could potentially interact with each other, but the sheer size of the feature sets made analyzing interactions extremely time-consuming and difficult. As such, an alternate approach was attempted with a random forest and XGBoost model trained on the same feature set as Table 8 with a 80/20 training/testing dataset split. The response variable of $\log(\text{usage})$ assumed an underlying log-normal distribution, since the models cannot train on non-normal distributions.

Figure 11 shows feature importance plots for both models, which generally agree with each other. Adjusting for evolution status and competitive tier, base statistics are the most important to determining usage, with the defences more important than the offensives. Shallow type details form a clear 2nd place, followed by a mix of move-related and type-related covariates. The most informative type is bug type, which was previously shown to correlate negatively with usage.

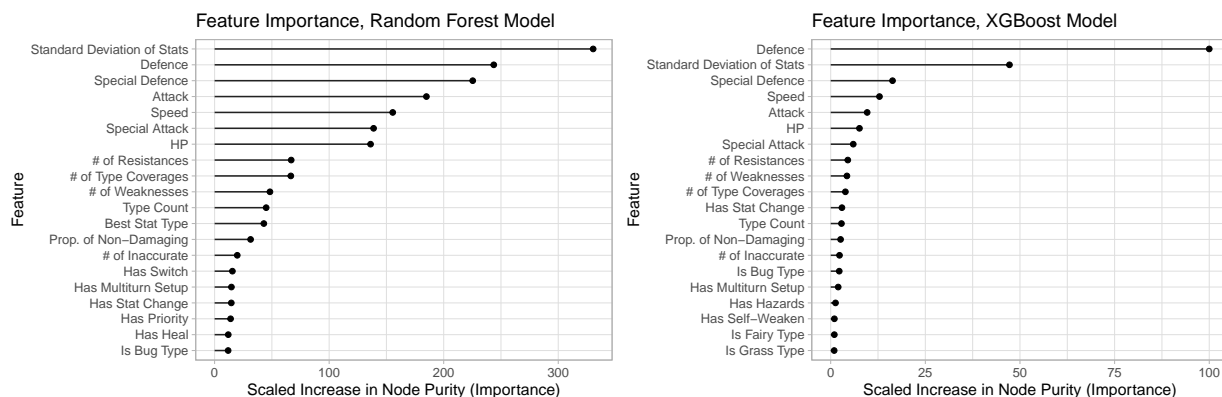


Figure 11. Feature importance plots showing the top 20 significant features of a random forest (left) and XGBoost (right) model trained on the same features as Table 8, after adjusting for evolution status and tier.

The random forest model has logged RMSEs of 0.49/1.15 on the training/testing data, while the XGBoost model has 0.41/1.26, a clear sign of overfitting that is corroborated by a closer analysis of the residuals in Figure 12 revealing considerable outliers in the testing set. Overfitting most likely occurred in continuous

variables such as the base statistics, where there are more possible splits and thus more ways to overfit; consequently, this would amplify their importance values in the feature importance plots.

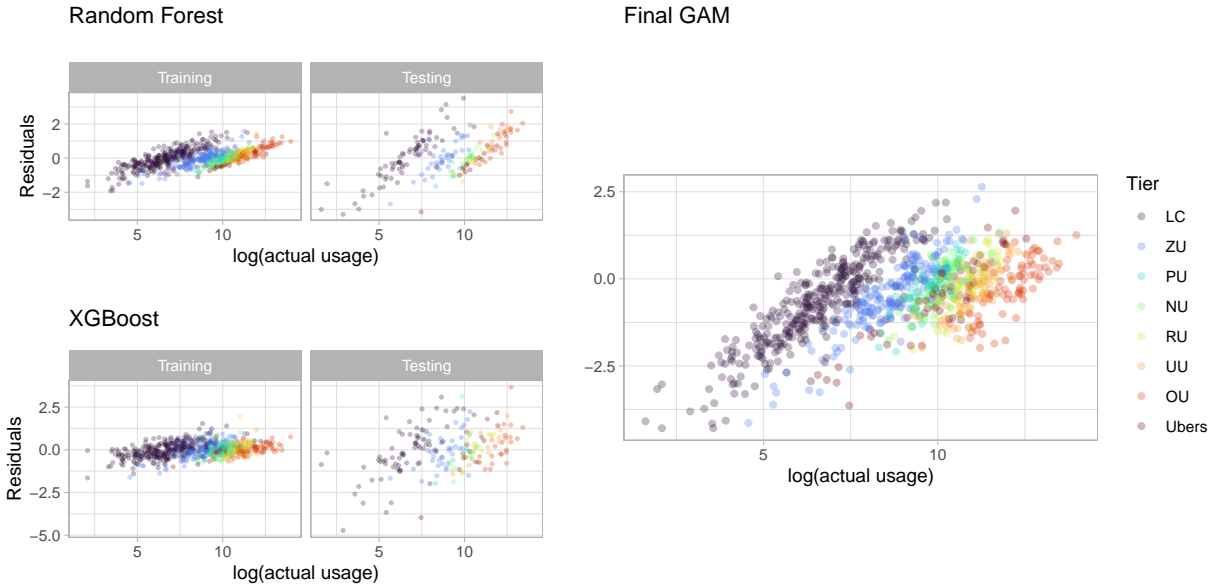


Figure 12. Residual plots of the difference in a Pokemon’s logged usage versus a model’s predicted logged usage for the decision tree models (left, split by training/testing set) and the final GAM model (right) from Table 8.

All models, even the final GAM model, still show diagonal line cluster artifacts in Figure 12, indicating there remains much unexplained variance in the data. Comparing the decision trees to the GAM, the trees find all base statistics important while the GAM only finds significance for a subset, possibly due to overfitting. The GAM also finds significance in move types and typings that the decision trees seem to consider unimportant. All models find importance in the standard deviation of base statistics, as well as the proportion of non-damaging moves.

Table 9. Coefficients for the final GAM model in Table 8, representing the estimated multiplicative change to a Pokemon’s rate of usage for a given Pokemon property or increase to it, relative to a monotype, general attack-based Pokemon.

Feature	Value	p-value	Feature	Value	p-value
(Intercept)	2432.726	2.03e-185	Is Fighting Type	1.490	0.027
Evolution = Final	0.530	0.005	Is Poison Type	1.375	0.112
Evolution = Middle	0.355	5.86e-16	Is Ground Type	1.313	0.124
Evolution = Non-Evolving	0.488	0.003	Is Flying Type	1.075	0.693
Tier = NU	14.310	3.22e-28	Is Psychic Type	0.958	0.796
Tier = OU	85.398	1.32e-64	Is Bug Type	0.703	0.032
Tier = PU	13.141	5.31e-29	Is Rock Type	1.181	0.407
Tier = RU	26.338	2.74e-39	Is Ghost Type	1.278	0.232
Tier = Ubers	14.224	2.30e-18	Is Dragon Type	1.351	0.127
Tier = UU	33.591	2.97e-44	Is Dark Type	1.546	0.015
Tier = ZU	6.471	1.47e-17	Is Steel Type	3.151	0.006
Dual-Type	1.181	0.212	Is Fairy Type	1.886	0.002
Best Stat = General Bulk	1.386	0.055	Prop. of Non-Damaging	2.509	0.001
Best Stat = None	0.939	0.76	# of Inaccurate	0.869	0.003
Best Stat = Phys. Attack	1.339	0.049	Has Priority	1.234	0.017
Best Stat = Phys. Bulk	1.718	0.009	Has Stat Change	0.690	1.21e-05

Feature	Value	p-value	Feature	Value	p-value
Best Stat = Spec. Attack	1.088	0.58	Has Multiturn Setup	0.703	0.019
Best Stat = Spec. Bulk	1.080	0.734	Has Heal	1.112	0.22
Best Stat = Speed	1.437	0.022	Has Status Effect	1.011	0.912
# of Weaknesses	0.990	0.846	Has Hazards	0.993	0.939
# of Resistances	0.880	0.017	Has Switch	1.106	0.239
# of Type Coverages	1.005	0.785	Has Recoil	1.162	0.215
Is Fire Type	1.737	0.018	Has OHKO	0.755	0.053
Is Water Type	1.354	0.081	Has Flinch	0.896	0.212
Is Grass Type	1.044	0.821	Has Self-Weaken	1.003	0.976
Is Electric Type	1.229	0.285	Has Multihit	0.921	0.635
Is Ice Type	0.717	0.153	Has Protect	0.830	0.199

Conclusion

After adjusting for competitive tier and evolution status, this analysis finds relatively modest conclusions:

- In order of decreasing importance, base statistics, the presence of types, and the presence of moves all have statistically significant effects on a Pokemon’s usage. This holds even across different assumptions of data distributions as well as model architectures.
- Usage increases by a factor of $\approx 1.67/1.20$ per 10-point increase in the total/standard deviation of base statistics respectively, when not controlling for specific base statistics.
- Usage correlates positively with special attack and special defence, but results are inconclusive for other base statistics. Table 9 and Figure 7 agree that speed-based Pokemon have high usage rates, but disagree on the other type specializations.
- Dual-types are have usage rates 1.2 that of monotypes, and usage rates decrease by a factor of 0.8 per additional weakness, when not controlling for specific types.
- The most popular type is steel, whose presence boosts usage by a factor of 3. Other types with significant positive effects are fairy ($\times 1.9$), fire ($\times 1.7$), dark ($\times 1.5$), and fighting ($\times 1.5$). The least popular type is bug, whose presence lowers usage by a significant factor of 0.7.
- Usage increases by roughly 25% both for Pokemon with priority moves and per a 50% increase in the proportion of non-damaging moves. Usage decreases by a factor of 0.7 for Pokemon with stat-changing moves and multiturn setup moves, as well as by 0.9 for each inaccurate move it has.

A large proportion of variance in usage data remains unexplained by these variables, as seen in the residual plot artifacts of Figure 12. Nevertheless, these findings provide an adequate starting point for understanding the popularity of a Pokemon in any given tier. Extensions on this study might include:

- Compiling a wider range of usage data across more months to obtain stabler numbers
- Looking at a Pokemon’s usage on winning teams (across many Pokemon matches) as opposed to usage numbers
- Factoring in a limited amount of item or ability information – solely that which affects base statistic, typing, or some common effects
- Including information regarding the Pokemon’s entire movepool, not just the top 5 moves
- Analyzing moves in a way that considers move popularity, team strategy, or individual moves. For instance, some moves are learnable by most Pokemon (e.g. Protect, Toxic) and some are indicative of wider team strategies (e.g. Trick Room, Baton Pass, Belly Drum) which may be better analyzed individually

Bibliography

The Pokémon Company. (2025). *2025 Pokémon World Championships*. Pokémon. Retrieved 16 March, 2025, from <https://www.pokemon.com/us/play-pokemon/pokemon-events/championship-series/2025/world-championships>

Smogon University. (n.d.) *An Introduction to Smogon's Tier System*. Retrieved 16 March, 2025, from https://www.smogon.com/bw/articles/bw_tiers