Predicting a Pokémon's Strength Using Variables Other Than Base Stats

STA 210 BDN: Nate Krall, Daniel Cohen, Brian Kim

2023-12-01

Introduction and Data:

A deeper look into the numbers in the games of Pokémon shows a game filled with relationships among several characteristics of the Pokémon. We aimed to investigate these relationships in their connection to a Pokémon's strength. Thinking about the game, each creature has its own specific set of base statistics, colloquially referred to as "base stats," including attack, special attack, defense, special defense, and speed, which indicate that Pokémon's battle prowess. Summing these stats yields a Pokémon's total base stats, the best measure of a Pokémon's overall strength – common knowledge for any Pokémon fan. We are looking to answer the following research question: **Can we predict a Pokémon's Base Stat Total from other variables?** We are analyzing how well variables such as the Pokémon's type, capture rate, growth rate, generation, height, weight, base happiness, and others can predict a Pokémon's total base stats. We hypothesize that a multiple linear regression model will be a somewhat strong predictor for base_total – thinking about the game, stronger Pokémon would seem to have certain values for these predictor variables when compared to weaker ones: for example, legendary Pokémon tend to be stronger in battle than non legendary Pokémon, so we might expect is_legendary to be a useful predictor for base_total. We retrieved the dataset from kaggle.com, a large data science online community, and the dataset is called "The Complete Pokémon Dataset" created by Rounak Banik in 2017. The dataset was retrieved via web scraper from the website serebii.net. Since it was formed in 2017, the dataset does not include Pokémon from recent games, but still includes 801 Pokémon, meaning the dataset has 801 observations. However, note that we removed one Pokémon from the original 801 Pokémon, Minior, from the dataset, since it has 2 different forms and has an uninterpretable capture rate. Minior's capture rate is: "30 (Meteorite)255 (Core)", which was recorded in the csv file as characters. We decided to exclude this observation from the model due to its uninterpretable characteristics. The dataset contains 23 variables, explanations of which can be viewed in our data dictionary. We will focus on these variables:

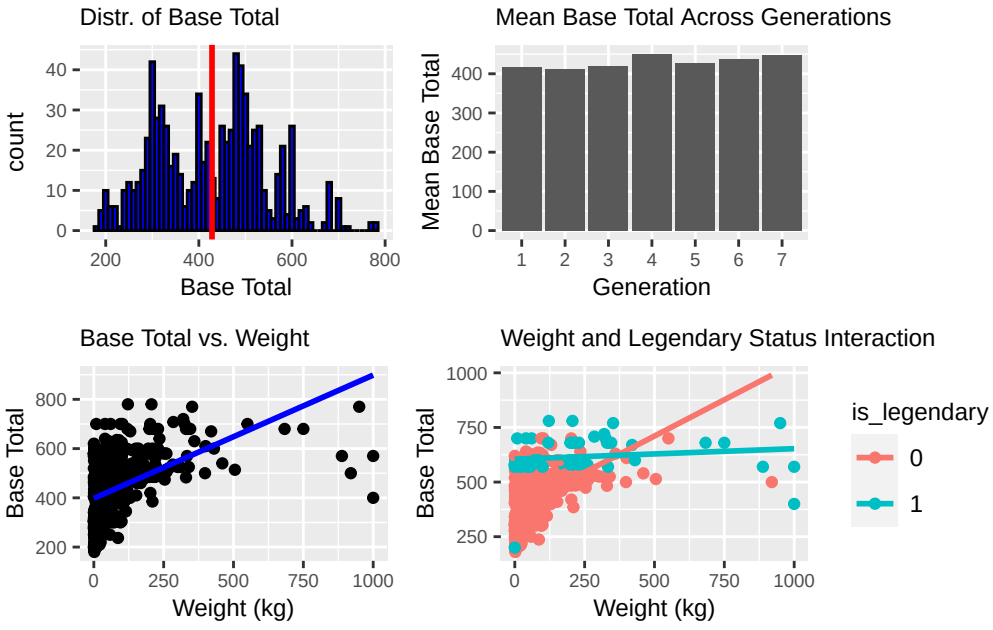
RESPONSE VARIABLE: base_total: total base stats of the Pokémon [whole number]

PREDICTOR VARIABLES

- experience_growth: The Experience Growth of the Pokémon [whole number]
- base_egg_steps: # of steps required to hatch an egg of the Pokémon [whole number]
- base_happiness: Base Happiness of the Pokémon [whole number]
- capture_rate: Capture Rate of the Pokémon [whole number]
- generation: The generation which the Pokémon was introduced [whole number 1-7]
- height_m: Height of the Pokémon [number in meters]
- percentage_male: % of the species that are male [percent, blank if no gender]
- pokedex_number: Entry # of the Pokémon in the Pokedex [between 1-801]
- type1, type2: The Primary Type and Second Type of the Pokémon, respectively
- weight_kg: The Weight of the Pokémon [number in kilograms]
- is_legendary: Denotes if the Pokémon is legendary.[0 = not legendary, 1 = legendary]

The dataset splits the base stats of each Pokémon into the individual stats, but we only need to know about the base_total. Some variables like is_legendary, may prove to be extremely important in our regression model, while others that are simply identifiers can be removed.

Exploratory Data Analysis



minimum	q1	median	mean	q3	maximum
180	320	435	428.288	505	780

p1: Distribution of Base total: The distribution of the base total seems to be roughly trimodal with three peaks (one around 300, 400, and 475). The base total points vary from about 180 to 780. Since there is an outlier around 780 base total, we use the median of 435 to describe the center of the data. The IQR is 185 (505 - 320).

p2: Mean Base Total Across Generations: This is the graph showing the relationship between different generation Pokémon and mean base total. The mean base total of generation 4 Pokémon were the highest. There is a slight positive linear relationship between the generation of the Pokémon and their mean base total.

p3: Scatter Plot of Base Total vs Weight: From the scatter plot we can see that there is no apparent correlation between the weight and the base total. There seems to be a few outliers near 900-1000kg.

p4: Interaction Between Weight and Legendary Status: This graph depicts how Pokémon weight correlates with base totals for both legendary (blue line) and non-legendary (red line) Pokémon. Legendary Pokémon typically have higher base totals (550-700), regardless of weight, showing no clear weight-related trend. On the other hand, non-legendary Pokémon exhibit a slight positive correlation; as their weight increases, their base total marginally rises. This distinction in base stat totals between legendary and non-legendary Pokémon will be reflected in our model.

Methodology:

We are conducting a multiple linear regression model to predict base_total from several other predictor variables. Any form of logistic regression would not make sense in this case as base_total is a quantitative variable, meaning we are not making classifications, and thus MLR is the model we conduct.

We randomly split our data into 75% training and 25% testing data to both train and evaluate our model.

Next, we take the training data through a recipe to ready it for our analysis.

1. We update the role of "name" to be an ID. Name is simply a label for each observation.
2. We remove irrelevant, non-predicting information like abilities, classification, and Japanese name.
    - There are hundreds of different abilities a Pokémon can have, and very little overlap of abilities between Pokémon, so we do not need the abilities variable. The classification of a Pokémon is almost unique for every Pokémon (there is very little overlap),

and it mainly just groups Pokémon by their evolution line, yielding classification unwanted.

3. We remove all against_* variables and all type variables, as these variables convey the same info.
4. We mean-center all quantitative predictors so our intercept is interpretable.
5. We address missing data in the height/weight category:
    - There are 20 Pokémon with missing height and weight values. Per the author of the database, these 20 Pokémon have alternate regional forms where their height and weights differ from their normal form, creating a disparity between these 20 Pokémon and the rest of the Pokémon in the dataset. Thus, we decided to remove these 20 Pokémon from consideration.
6. Finally, we remove percentage_male from consideration.
    - Several Pokémon do not have a gender, leading to many missing values in the dataset. Upon further examination, we found that a disproportionate 63/70 of the legendary Pokémon in the dataset to not have a gender, whereas most non-legendary Pokémon do have a gender. Percentage_male and is_legendary cannot be effective predictors in conjunction, so we decide to remove percentage_male from our model.

Note that pokédex number is in fact a unique identifier for a Pokémon, yet it also contains information on when the Pokémon was released in-game, as larger pokédex numbers correspond to later releases, which may have a tie to base_total. Also, generation may seem as if it is an arbitrary label, but it also corresponds to when a Pokémon was released in a different manner than pokédex number. We are wary of the potential of collinearity among these variables and will analyze the substantiallity of their collinearity (as well as all other potential predictors) in our analysis.

After bringing the training data through our recipe, we fit the data in a MLR model:

term	estimate	std.error	statistic	p.value
(Intercept)	420.025	3.586	117.121	0.000
base_egg_steps	-0.001	0.001	-0.688	0.492
base_happiness	0.275	0.186	1.481	0.139
capture_rate	-0.800	0.044	-18.063	0.000
experience_growth	8.931	3.201	2.790	0.005
height_m	36.300	4.253	8.536	0.000
pokedex_number	0.301	0.085	3.540	0.000
weight_kg	0.023	0.042	0.564	0.573
generation	-31.754	10.048	-3.160	0.002
is_legendary1	78.688	25.909	3.037	0.002

Base_egg_steps, base_happiness, and weight_kg are three candidates for variables to remove from the model, since we notice their p-values are all $> .05$, meaning they are potentially statistically insignificant predictors. The next step in our method is compare two models through a series of tests: the one model being our original model with all predictors after running our feature engineering, and one with those variabke removed to select a model that strikes a balance between conciseness and detail, or select the most comprehensive one available.

We create a similar recipe for the second model except that base_egg_steps, base_happiness, and weight_kg are removed. The output of running MLR is shown below:

term	estimate	std.error	statistic	p.value
(Intercept)	421.680	3.105	135.814	0.000
capture_rate	-0.806	0.044	-18.408	0.000
experience_growth	0.000	0.000	2.581	0.010
height_m	35.826	3.510	10.207	0.000
pokedex_number	0.284	0.084	3.396	0.001
generation	-30.017	9.896	-3.033	0.003
is_legendary1	57.550	12.883	4.467	0.000

We then conducted two tests to decipher which model is best to select for our final model: one comparing the AIC, BIC, and adjusted R^2 for the models, and another comparing the results of V-fold cross validation for the models. First test output:

AIC	BIC	adj.r.squared
6646.726	6694.795	0.643

AIC	BIC	adj.r.squared
6644.285	6679.244	0.643

From this test, we note that both AIC and BIC are lower for the second, reduced model. The adjusted R^2 values are very similar in size. It would be logical to select the second model, as it produces more preferable values of AIC and BIC while maintaining a very similar adjusted R^2 value. However, we run one more test, v-fold cross validation, to compare the models once again. The results of this second test are below:

Cross validation results for the 1st, full model:

.metric	.estimator	mean	n	std_err	.config
rmse	standard	71.803	15	3.619	Preprocessor1_Model1
rsq	standard	0.634	15	0.035	Preprocessor1_Model1

Cross validation for the 2nd, reduced model:

.metric	.estimator	mean	n	std_err	.config
rmse	standard	71.437	15	3.388	Preprocessor1_Model1
rsq	standard	0.636	15	0.033	Preprocessor1_Model1

We notice that the RMSE value from cross validation of the second model is less than the RMSE value from the cross validation of the first full model, while the R^2 values from both models have negligible difference, meaning the second model is a better predictor a Pokémon's base stat total. From the results of these tests, **we can confidently select model number 2, the reduced model**.

To determine there is no multicollinearity in our model, we must examine the VIF values:

names	x
capture_rate	1.265
experience_growth	1.169
height_m	1.336
pokedex_number	41.169
generation	40.427
is_legendary1	1.425

A VIF value > 10 for a variable indicates concerning collinearity. We notice that pokedex_number and generation have VIF values > 40, meaning pokedex_number and generation appear to be collinear. This makes sense – as generations of Pokémon are intervals of Pokédex numbers. Generation divides all values 1-800 of pokedex_number into different intervals (for example, generation 1 is Pokédex numbers 1-151). To fix this, we deleted generation, since it is far more discrete than pokedex_number.

Our final model is the same as the second model with generation removed.

	x
capture_rate	1.241
experience_growth	1.174
height_m	1.263
pokedex_number	1.042
is_legendary1	1.330

As we can see, the VIF values are all now satisfactorily low for the final model.

term	estimate	std.error	statistic	p.value
(Intercept)	419.982	2.774	151.379	0.000
capture_rate	-0.839	0.038	-21.811	0.000
experience_growth	2.412	2.810	0.858	0.391
height_m	28.620	2.724	10.507	0.000
pokedex_number	0.024	0.012	2.014	0.044
is_legendary1	88.596	10.629	8.336	0.000

The equation for the final model is:

$$\begin{aligned} base\_total = 419.982 - 0.839 * capture\_rate + 2.412 * experience\_growth \\ +28.62 * height\_m + 0.024 * pokedex\_number + 88.596 * is\_legendary1 \end{aligned}$$

We can tell that we expect the base total to be 419.982 for a Pokémon that is not legendary, with capture_rate, experience_growth, height_m, pokedex_number at their mean values.

For every one unit increase in capture rate, we expect base total to decrease by .839, on average, holding all other predictor variables constant.

For every one unit increase in experience growth, we expect base total to increase by 2.412, on average, holding all other predictor variables constant.

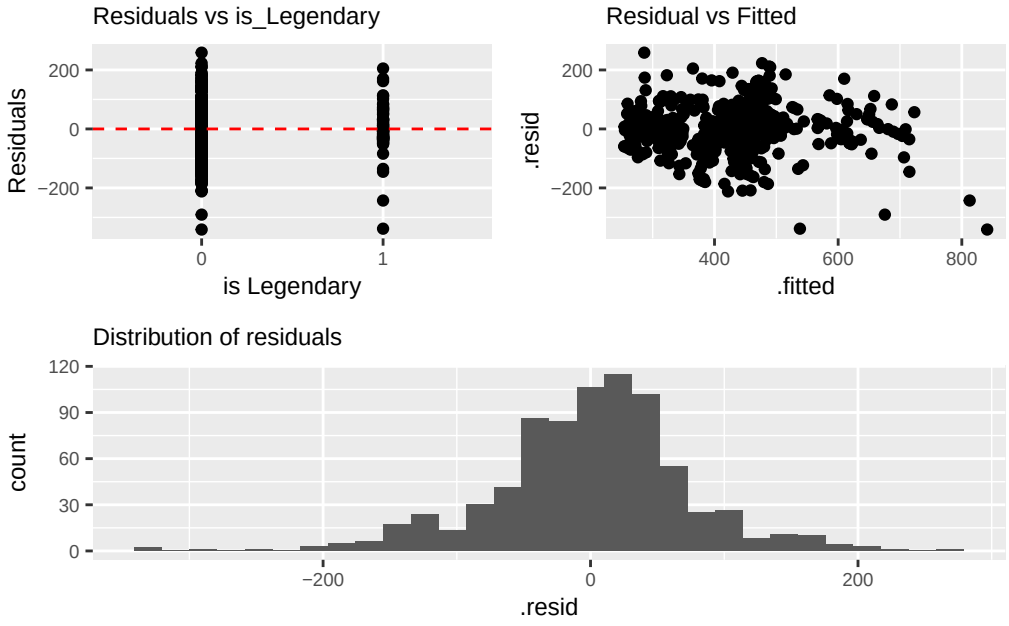
For every one meter increase in height, we expect base total to increase by 28.62, on average, holding all other predictor variables constant.

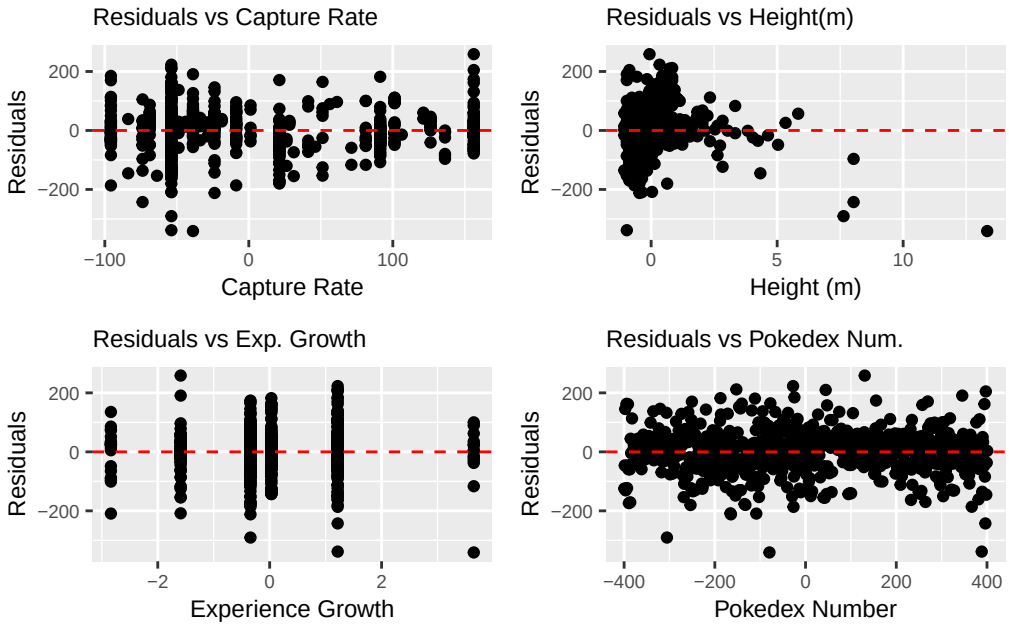
For every one increase in Pokédex number, we expect base total to increase by .024, on average, holding all other predictor variables constant.

We expect a legendary Pokémon to have a base total greater than a non legendary Pokémon by 88.596, on average, holding all other predictor variables constant.

One interesting observation from our analysis is that the visible qualities, such as height and legendary status, have the most impact on a Pokémon's base total, as indicated by their relatively large coefficients in the regression model. For players, these insights are valuable as they can make assessments of a Pokémon's strength based on physical characteristics.

Results:





Linearity condition - This condition is satisfied as there is no clear patterns in the residuals vs predictor variables such as a fanning pattern and the fitted vs residuals graph also seems to have no fanning pattern.

Constant Variance - The vertical spread of the residuals is constant across the plots of residuals versus fitted values, therefore this condition is satisfied.

Normality - The distribution of the residuals is approximately unimodal and symmetric, so the normality condition is satisfied. The sample size is sufficiently large > 30 so we can relax this condition.

Independence - The independence condition is **not** satisfied. The independence condition in our Pokémon dataset is not met due to the evolutionary relationships between Pokémon. For instance, Pokémon like Bulbasaur, Ivysaur, and Venusaur share evolutionary lines, leading to non-independent data, as these related Pokémon often have similar types, generations, and correlated base stats. Acknowledging this, our model might be less predictive for independent Pokémon, assuming independent errors, which is not the case here.

Instead of an in-depth analysis on the standard errors for the model coefficients to justify moving forward with conclusions, we choose to elaborate as to why conclusions are irrelevant if we choose one pokemon per evolution chain. We chose not to remove Pokémon from each evolutionary line to maintain dataset integrity and relevance to our research question. Excluding more than half of our Pokémon to meet the independence condition would limit the scope of our conclusions. Our aim is to analyze how various factors predict a Pokémon's base stat total, considering all Pokémon, including those within evolutionary chains, as roughly 75% of

Pokémon are part of an evolutionary chain. Limiting our analysis to only the final evolution in each chain would not accurately represent the full spectrum of Pokémon as they exist in the games, and it would detract from the purpose of our study - to understand the predictors of a Pokémon's Base Stat Total across all evolutionary stages.

Results

Here are the RMSE and R^2 given from K-fold cross validation:

.metric	.estimator	mean	n	std_err	.config
rmse	standard	71.950	15	3.206	Preprocessor1_Model1
rsq	standard	0.634	15	0.033	Preprocessor1_Model1

Here are the RMSE and R^2 given from the training set:

.metric	.estimator	.estimate	.metric	.estimator	.estimate
rsq	standard	0.641	rmse	standard	71.085

Here are the RMSE and R^2 given from the test set:

.metric	.estimator	.estimate	.metric	.estimator	.estimate
rsq	standard	0.577	rmse	standard	80.689

When we compare the R^2 values, that of the training model was .641 while the testing model had a R^2 value of .577. This is to be expected since the model will perform better on data that it has been trained on than data it has not seen. When we compare the RMSE for the training and testing data, we can see that the RMSE for the training set was 71.1 while the RMSE of the testing set was 80.69. Since lower RMSE values indicate a better fit to the model, we can see that the model once again performed slightly better on the training data than the testing data, which is to be expected. However, since the difference in the R^2 and the RMSE values between the training and the testing data wasn't too significant, this shows that our model doesn't overfit the data.

Overall, for our initial interpretations of our final model, we find it to be a relatively strong predictor for base_total of a Pokemon given the relatively low RMSE value and the relatively high R^2 value. Thus, we have confirmed our hypothesis that in fact, we *can* predict the base_total of a Pokémon with decent accuracy by using different variables.

Discussion + Conclusion:

Summary of Findings: From the equation of our final model we found out that in order to maximize the base total of a Pokémon, we want that Pokémon to have a low capture rate, high experience growth, to be tall, large pokedex number, and want our Pokémon to be of the legendary kind. Our research aimed to predict a Pokémon's base stat total. Our analysis revealed statistically significant and statistically insignificant relationships between certain characteristics such as Pokémon's capture rate, whether it is legendary, and generation

with the Pokémon's overall strength (base total). Our final model which included capture rate, experience growth, height, Pokédex number, and legendary status, was not only a good predictor of base_total, accounting for approximately 64.074% of the variability in Base Stat Total in the training data, but also was a concise model, only including statistically significant predictors. The relatively low RMSE value of 72.81 also suggested a satisfactory level of prediction accuracy, as base total ranges from 200 to 800.

Limitations and Improvement Suggestions: The dataset was drawn in 2017 does not include Pokémon in games released after this date. This means that this model should not be extrapolated past this date as the Pokémon games evolve over time and might affect our applicability of these results to future editions. Additionally, as we talked more in detail about in the previous section, the independence condition being violated limits the predictive power of this model, as evolution chains are not including in our model. While we removed variables like base_egg_steps and base_happiness due to their high p-values, further exploration could determine if any interaction effects or non-linear relationships exist that we might have overlooked.

Reliability and Validity Concerns: The linear regression model assumes linearity, independence, homoscedasticity, and normality of residuals, which we proved held except for the independence in our dataset. Even though our model satisfied all but the independence conditions, we have explained above that moving forward with this in mind is the better decision as removing non independent Pokémon would limit the scope of our conclusions and our goal of the study was to predict the base total across all Pokémon in all evolutionary stages, not just one Pokémon from each evolutionary line. Any other violation in future datasets of future Pokémon games could make the model unreliable. Additionally, the data was scraped from a fan-run website, there might be biases or errors in how the information was recorded, affecting the validity.

Future Work: To continue our research in the future, we could could include data from newer Pokémon games to keep our model relevant over time. Additionally, analyzing how a Pokémon's evolutionary stage affects its Base Stat Total could be very interesting, especially due to the great variance in Pokémon evolution. For examples, the starter Pokémon are generally pretty strong compared to other Pokémon their level at all stages in their evolution. On the other hand, Magikarp is possibly the weakest Pokémon in the game, but evolves to become Gyarados, one of the strongest Pokémon in the game. Additionally, we could have more investigation into the interactive effects between variables, like how the combination of type and legendary status impacts base stats, which could provide a deeper understanding of the underlying dynamics.