# Pokemon Final Project

Alexa Fahrer and Nick Maroulis

## Introduction and Data

As students who grew up playing Pokemon games, including Pokemon Go, Pokemon Black/White, and many others, we are interested in examining the factors that contribute to the difficulty of catching a certain species of Pokemon. It had always seemed like strong high-stat Pokemon, especially legendaries, are more difficult to catch than week low-stat Pokemon, but there were always exceptions to this logic. After searching for data about Pokemon in relation to catch rate, we came across a data set from Kaggle called "The Complete Pokemon Dataset," with includes information about more than 800 Pokemon from all seven generations (https://www.kaggle.com/datasets/rounakbanik/pokemon?resource=download). The data set has 801 observations– each a unique Pokemon– and 41 variables that are a mix of quantitative and qualitative. For this project, we decided to focus on the predictors name, Pokedex number, generation, capture rate, percentage male, the type(s) of the Pokemon, height, weight, the number of steps per egg cycle (base egg steps), experience growth (XP), base happiness, hp, attack, defense, special attack, special defense, speed, and whether the Pokemon is legendary or not.

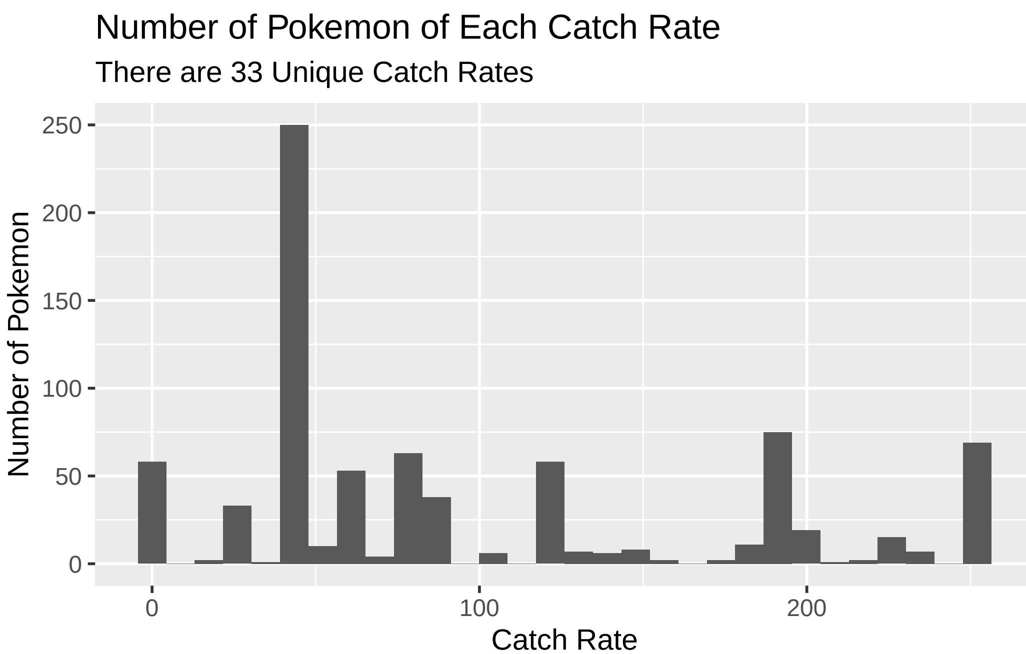
When attempting to catch a Pokemon, there are many variables that go into if the Pokemon will be caught or not. These variables include type of Poke ball used, level of the wild Pokemon, etc. All of these factors, in addition to RNG (random number generation), are factored into a specific formula to determine if a Pokemon is caught on any given attempt. Details about the formula can be found here: https://bulbapedia.bulbagarden.net/wiki/Catch_rate. Additionally, every Pokemon has its own "catch rate," which is weighted heavily in the formula. Pokemon with a higher catch rate are easier to catch. For example, Pidgey, which is a weak, unevolved Pokemon, has a catch rate of 255, which is tied for the highest, meaning that Pidgey is very easy to catch. However, Mewtwo, a strong, legendary Pokemon, has a catch rate of 3, which is tied for the lowest, meaning that Mewtwo is extremely difficult to catch. Please note that we use "catch rate" and "capture rate" interchangeably.

In this project, we are examining the following research question: What characteristics of a Pokemon influence catch rate? Our main outcome of interest is the variable "capture_rate." Our hypotheses are as follows: larger Pokemon (in terms of height and weight) will have lower capture rates than smaller Pokemon. Pokemon with higher attack, defense, special attack, special defense, HP, and speed stats will have lower capture rates than Pokemon with lower stats. A Pokemon's type will not play a role in the catch rate. Finally, legendary Pokemon will have lower capture rates than non-legendary Pokemon. In the following project, we will make visualizations to explore the data, consider different types of regression models, use variable selection techniques, consider missing data, and select a final model to examine the characteristics of Pokemon in the data set that are statistically significant in influencing capture rate.

## Methodology

### Type of Model

We decided a Linear Regression Model is the best model to predict the catch rate of a Pokemon.




In this data set, the catch rate of Pokemon is an integer from 3 to 255. However, there are only 33 unique catch rates among the 801 Pokemon. For example, 50 Pokemon have a catch rate of 60 and a whopping 250 Pokemon have a catch rate of 45. Based off of this information, it may make sense to use a multinomial regression model. Since the data are ordered, an ordinal regression model would be another good option. This would lead to a model that predicts the catch rate of a Pokemon relative to the 33 unique catch rates. However, this is a lot of outcomes for the response variable. We could simplify this ordinal regression model with some number of different “bins.” For example, bin 0 could be all Pokemon with a catch rate of 0-50, bin 1 is all Pokemon with a catch rate of 50-100, etc.

We decided against using the ordinal regression model for multiple reasons. First, 33 different outcomes for the response variable is a lot. It creates an unnecessarily complicated model. This problem can be fixed by grouping the catch rates into different “bins.” However, this is not optimal, because we don’t have a strategy for how to create the “bins.” How many “bins” should we have? Should they all be of equal length? Without the proper tools, this method does not make sense for our purposes. However, overall, we decided against the ordinal regression model because even though many Pokemon share a catch rate with many others, there are still Pokemon that have a unique catch rate, meaning the current catch rates of Pokemon are not the only possible catch rates. There could still be a new Pokemon introduced that has a fully unique catch rate. If we were tasked with predicting the catch rate of a new Pokemon with certain attributes, our model would no longer make sense if that Pokemon’s catch rate was fully unique. We would be looking at the probability that this Pokemon has a certain catch rate, but the Pokemon wouldn’t have any of the listed catch rates.

A logistic regression model also does not make sense in the context of our data because our main outcome of interest, capture rate, is a numerical predictor, and logistic regression requires the response to be categorical and binary. This leaves us with linear regression, which best fits the data we are using since our response variable is numeric. Linear regression with multiple predictors will allow us to examine the relationship between our predictors of interest and outcome while holding other variables constant. Therefore, we decided that a linear regression model would best suit our research question.

### Missing Data

Many species of Pokemon do not have a gender, leading to many NA values for the *percentage_male* variable. However, there is a disproportionate amount of legendary Pokemon that do not have a gender. This is because in a Pokemon game, legendaries are unique and special; there is only one of each legendary Pokemon in the “world.” Thus, in our data set, 63/70 of legendary Pokemon have an NA for *percentage_male*. This data is MAR (missing at random) since legendary status is an observed variable in the data set. Because these Pokemon are legendary, many of them have extremely low catch rates. Specifically, 53/70 legendary Pokemon have a catch rate of 3, the lowest possible catch rate. This means that our model will most likely not be able to accurately predict legendary Pokemon’s catch rate, and there will be worse representation among Pokemon with lower catch rates. The best and easiest solution to this problem is to eliminate the *percentage_male* predictor from the model.

Next, we noticed that while reading the .csv file, R automatically translated the *capture_rate* variable to characters. Upon inspection, Pokemon number 774, Minior, had this listed as its catch rate: “30 (Meteorite)255 (Core)”. During battle, Minior has two forms: meteor form and core form. Minior has the “shields down” ability, which means that when Minior is at half health or less, Minior changes from the Meteor form to the core form. However, Minior's catch rate also changes when it changes forms, from 30 (meteor form) to 255 (core form). Due to Minior's ability having a direct relation to the catch rate, we chose to exclude Minior as an observation. We decided to update Minior's catch rate to be NA, thus excluding this observation from the model. We also made *capture_rate* an integer instead of a character.

Finally, we noticed that there are 20 Pokemon with missing data for their heights and weights. Upon examining notes from the author of the data set on the website, we realized that these 20 Pokemon all have an alternate form (Alolan). The alternate forms of these Pokemon have different heights and weights than their normal form. As a result, the height and weight of these Pokemon were inputted into the data set as NA values. These values are MNAR (missing not at random), as they are related to unobserved data, which is whether the Pokemon has an alternate form or not. We also read that the data set scraped the Alolan base stats for these 20 Pokemon, so the rest of the information will be based on the alternate form of the Pokemon. We decided to exclude these 20 observations from the data set.

It is also worth nothing that Pokemon only have one type, and thus do not have a second type. This is not missing data, this is just the classification of the Pokemon.

### Variable Selection

After acknowledging missing data, we must next narrow down the variables in the data set to only the important predictors for our model.

For classification and readability purposes, we chose to keep the variables *name*, *pokedex_number*, and *generation*. The response variable is *capture_rate*, so we must of course keep this variable. Other important variables that may play a role in capture rate are *type1*, *type2*, *height_m*, *weight_kg*, *base_egg_steps*, *experience_growth*, *base_happiness*, *hp*, *attack*, *defense*, *sp_attack*, *sp_defense*, *speed*, and *is_legendary*.

We chose to manually eliminate some variables from the data set based on our knowledge of Pokemon. First, the *japanese_name* is just another version of *name*. The *against_?* variables are dependent on only typing (*type1* and *type2*), and it will be more useful to just look at the typing of a Pokemon rather than how effective certain moves are against it. There are hundreds of different abilities a Pokemon can have, and very little overlap of abilities between Pokemon, so we do not need the *abilities* variable. The classification of a Pokemon is almost unique for every Pokemon (there is very little overlap), and it mainly just groups Pokemon by their evolution line, yielding *classification* unwanted. Finally, the *percentage_male* variable is excluded due to missing data, as discussed above.

We now run variable selection models to determine which of the following variables are best/important for predicting catch rate: *height_m*, *weight_kg*, *base_egg_steps*, *experience_growth*, *base_happiness*, *hp*, *attack*, *defense*, *sp_attack*, *sp_defense*, *speed*, and *is_legendary*. The variables *type1* and *type2* are discussed in the variable transformation section.

First, we tried _____ We decided not to use forward selection and backward elimination methods because they often don't work well with highly correlated variables, and we suspect that some of our predictors are correlated.

```
13 x 1 sparse Matrix of class "dgCMatrix"
                          s0
(Intercept)        .
height_m           .
weight_kg          .
base_egg_steps     .
experience_growth  .
base_happiness     .
hp                -0.5594079
attack            -0.3415237
defense           -0.4295399
sp_attack         -0.3798392
sp_defense        -0.4627282
speed             -0.4346233
is_legendary1      .
```

EXPLAIN RESULTS OF VARIABLE SELECTION HERE

### Interaction Terms

We were considering using an interaction term between height and weight. This is because a tall Pokemon is usually heavier, and vis versa. However, we will not be using either of these variables in our final model, so we will not have any interaction terms.

### Variable Transformation

One of the defining features of a Pokemon is its typing. There are 18 total types, and a Pokemon can have either one or two types. In the context of the data set, if a Pokemon has a type1, but an empty type2, then the Pokemon only has one type. If the Pokemon has both a type1 and type2, then it is a dual-type Pokemon (The only exception is if the Pokemon has an alternate Alolan form, which we have already decided to remove from our dataset). While a Pokemon technically has a "primary" type and "secondary" type, the order of the typing is unimportant (e.g. a fire and ground Pokemon has the same typing as a ground and fire Pokemon). Because the Pokemon typing is organized this way, type1 and type2 can't be predictors themselves.

Our solution is to create 18 new variables (*is_[type]*), that has a '1' if the Pokemon is of the specified type, and '0' if the Pokemon is not. By using all 18 columns as predictors, we can successfully include a Pokemon's typing as a predictor of a Pokemon's catch rate.

However, we are seeking to find the most important factors of a Pokemon in determining its catch rate. Certain Pokemon types tend to be closely related to the factors of a Pokemon that we are already using as predictors. For example, dragon types typically have very high attack stats, and steel types typically have very high defense stats. By including the Pokemon's type as a predictor, the overall effect of the other predictors on catch rate will most likely be understated. However, certain types may still have an effect on catch rate for reasons other than our predictors. For example, Dragon types may be intended by the creators to have lower catch rates than other Pokemon for reasons other than the fact that they have slightly higher stats than the other Pokemon. Thus, this leads us to the creation of three different models.

### The Models

We will have two different models. The first model only uses the predictors that were selected by the LASSO model. The second model uses the same predictors as the first, but with the addition of the 18 type predictors.

## Results

```
Call:
lm(formula = capture_rate ~ hp + attack + defense + sp_attack +
    sp_defense + speed, data = pokemon_new)

Residuals:
     Min       1Q   Median       3Q      Max
-154.748  -27.961   -5.722   32.267  217.521

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 293.82136    7.44545  39.463  < 2e-16 ***
hp           -0.59752    0.08409  -7.106 2.73e-12 ***
attack       -0.34526    0.07972  -4.331 1.68e-05 ***
defense      -0.47036    0.08520  -5.521 4.61e-08 ***
sp_attack    -0.39071    0.07757  -5.037 5.90e-07 ***
sp_defense   -0.47203    0.09517  -4.960 8.67e-07 ***
```

```
speed        -0.48170    0.07967  -6.046 2.30e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 53.69 on 773 degrees of freedom
Multiple R-squared:  0.5034,	Adjusted R-squared:  0.4995
F-statistic: 130.6 on 6 and 773 DF,  p-value: < 2.2e-16



Call:
lm(formula = capture_rate ~ hp + attack + defense + sp_attack +
    sp_defense + speed + is_bug + is_dark + is_dragon + is_electric +
    is_fairy + is_fighting + is_fire + is_flying + is_ghost +
    is_grass + is_ground + is_ice + is_normal + is_poison + is_psychic +
    is_rock + is_steel + is_water, data = pokemon_new)

Residuals:
     Min       1Q   Median       3Q      Max
-155.891  -32.578   -6.222   32.716  214.339

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 289.49415    9.41959  30.733  < 2e-16 ***
hp           -0.65291    0.08704  -7.502 1.78e-13 ***
attack       -0.31207    0.08559  -3.646 0.000285 ***
defense      -0.38376    0.09284  -4.133 3.97e-05 ***
sp_attack    -0.32279    0.08547  -3.777 0.000171 ***
sp_defense   -0.49038    0.09657  -5.078 4.81e-07 ***
speed        -0.54482    0.08368  -6.511 1.36e-10 ***
is_bug       -2.11439    7.45297  -0.284 0.776720
is_dark       3.80455    8.83068   0.431 0.666712
is_dragon   -19.86610    9.29632  -2.137 0.032920 *
is_electric  12.03048    9.70298   1.240 0.215407
is_fairy      2.45313    8.94946   0.274 0.784075
is_fighting   1.22471    8.82550   0.139 0.889670
is_fire     -19.34008    8.61378  -2.245 0.025041 *
is_flying     0.08646    6.34246   0.014 0.989127
is_ghost     -4.69527    9.39412  -0.500 0.617355
is_grass     -2.68354    7.19452  -0.373 0.709254
is_ground     7.46953    8.27030   0.903 0.366721
is_ice       -3.49059   10.14962  -0.344 0.731007
is_normal    13.85158    7.74693   1.788 0.074176 .
```

```
is_poison    14.81805    7.85829   1.886 0.059725 .
is_psychic    2.25674    7.91543   0.285 0.775641
is_rock     -21.04811    8.60485  -2.446 0.014669 *
is_steel     -1.81440    9.13452  -0.199 0.842605
is_water      0.11877    6.74316   0.018 0.985952
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 53.15 on 755 degrees of freedom
Multiple R-squared:  0.5246,    Adjusted R-squared:  0.5095
F-statistic: 34.71 on 24 and 755 DF,  p-value: < 2.2e-16
```

Our first model shows that the best predictors of capture rate are the six stats of a Pokemon (HP, attack, defense, sp_attack, sp_defense, speed). In general, a Pokemon with higher stats is estimated to, on average, have a lower catch rate. Specifically, HP has the greatest impact. For every 1 point increase of the base stat HP, our model estimates, on average, a 0.59752 decrease in catch rate, while holding the other 5 stats constant. This is with a very low p-value of 2.73e-12, implying a very likely correlation. This kind of relationship can be seen with each of the stats. EXPAND? (If you want, or expand in the first paragraph of discussion)

In our second model, even with the inclusion of the *is_[type]* predictors, the stats of a Pokemon remain the best predictors. We interestingly see that *is_dragon*, *is_fire*, and *is_rock* have low p-values (and negative coefficients). This implies that there is something other than the stats of a Pokemon that causes these three types to have lower capture rates than the other types. FORMAL HYPOTHESIS TEST?

## Discussion

We hypothesized that Pokemon with higher attack, defense, special attack, special defense, HP, and speed stats would have lower capture rates than Pokemon with lower stats. Our models show that this hypothesis was correct.

We hypothesized that a Pokemon's typing would not have any effect on catch rate. However, it turns out that dragon, fire, and rock Pokemon have some external reason of having a lower catch rate. We speculate that dragon types are more rare, thus the game developers want them to be more difficult to catch. However, we are unsure why fire and rock types would be harder to catch. More research can be done into specifically why these three types have lower catch rates.

We hypothesized that larger Pokemon (in terms of height and weight) would have lower capture rates than smaller Pokemon, however, the LASSO model did not choose these variables as predictors. The capture rate of a Pokemon may not depend on the height and/or weight of the Pokemon. More research can be done prove or disprove this theory.

Finally, we hypothesized that legendary Pokemon would have lower capture rates than non-legendary Pokemon. We were very surprised that *is_legendary* was not chosen as a predictor by the LASSO model. Through our Pokemon experiences, we have always assumed that a legendary Pokemon was harder to catch because it was legendary. We speculate that legendary Pokemon have much higher stats in general, thus the reason they have low catch rates is due to their stats being so high, not because they are legendary. We ran a linear model that uses only *is_legendary* as a predictor, and there was a very likely correlation. More research can be conducted on specifically why *is_legendary* may not be the best predictor of catch rate.

## Appendix

Here is a partial look at the Pokemon data set, including the transformations we completed (manually excluding some variables, changing *capture_rate* to be an integer).

```
        name pokedex_number generation capture_rate type1  type2 height_m
1  Bulbasaur              1          1           45 grass poison      0.7
2    Ivysaur              2          1           45 grass poison      1.0
3   Venusaur              3          1           45 grass poison      2.0
4 Charmander              4          1           45  fire             0.6
5 Charmeleon              5          1           45  fire             1.1
  weight_kg base_egg_steps experience_growth base_happiness hp attack defense
1       6.9           5120           1059860             70 45     49      49
2      13.0           5120           1059860             70 60     62      63
3     100.0           5120           1059860             70 80    100     123
4       8.5           5120           1059860             70 39     52      43
5      19.0           5120           1059860             70 58     64      58
  sp_attack sp_defense speed is_legendary is_bug is_dark is_dragon is_electric
1        65         65    45            0      0       0         0           0
2        80         80    60            0      0       0         0           0
3       122        120    80            0      0       0         0           0
4        60         50    65            0      0       0         0           0
5        80         65    80            0      0       0         0           0
  is_fairy is_fighting is_fire is_flying is_ghost is_grass is_ground is_ice
1        0           0       0         0        0        1         0      0
2        0           0       0         0        0        1         0      0
3        0           0       0         0        0        1         0      0
4        0           0       1         0        0        0         0      0
5        0           0       1         0        0        0         0      0
  is_normal is_poison is_psychic is_rock is_steel is_water
1         0         1          0       0        0        0
2         0         1          0       0        0        0
3         0         1          0       0        0        0
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |