Part 3: Multiple Linear Regression

Alex Gui and Lathan Liou

Date Submitted: March 26, 2018

Introduction

Pokemon Go became an overnight sensation with hundreds of millions of people having downloaded the mobile game. The whole point of the game is to try to catch all the Pokemon available and train them (increase their combat power, which is abbreviated cp) so that you can battle other players with your strengthened Pokemon. A quick note about Pokemon is that they can evolve into stronger forms, so an evolved Pokemon will generally always have a higher cp than a non-evolved Pokemon. A number of people have tried to generate models in an attempt to predict the best way to maximize cp for their Pokemon. This is what we will attempt to do ourselves: create a model to predict combat power for an evolved Pokemon.

To refresh your memory, the dataset we are looking at is an original data set collected by *OpenIntro*[1], most likely by some individual who was playing Pokemon Go and decided to record data. The dataset contains 75 observations across 26 variables, with each observation representing a randomly generated Pokemon that the gamer caught. Four species are represented in this data, so the conclusions drawn from this modeling process will reflect the population of these 4 particular species: Eevee, Pidgey, Caterpie, and Weedle.

We avoid using "new" variables in our modeling process because as a user, you wouldn't have access to any of the "new" information, but if you're interested in whether your pokemon will evolve into one with a high cp (cp_{new}), you would want to know which of the Pokemon's current stats can indicate a high cp_{new}. The variables that we'd like to focus on are cp, hp, and power_up_stardust. Our intuition is that a pokemon with a higher cp might evolve into a pokemon with a higher cp. Hp, or hit points, refers to the amount of damage a Pokemon can sustain in battle before fainting. It would be interesting to see whether a Pokemon with high hp will also have high cp. Power up stardust is used to raise cp of the pokemon, but the catch here is we don't know the ideal amount of power up stardust to max out the cp_{new} of the evolved pokemon.

You can follow our work here: https://github.com/alexaaag/math158-project.

Exploratory Analysis

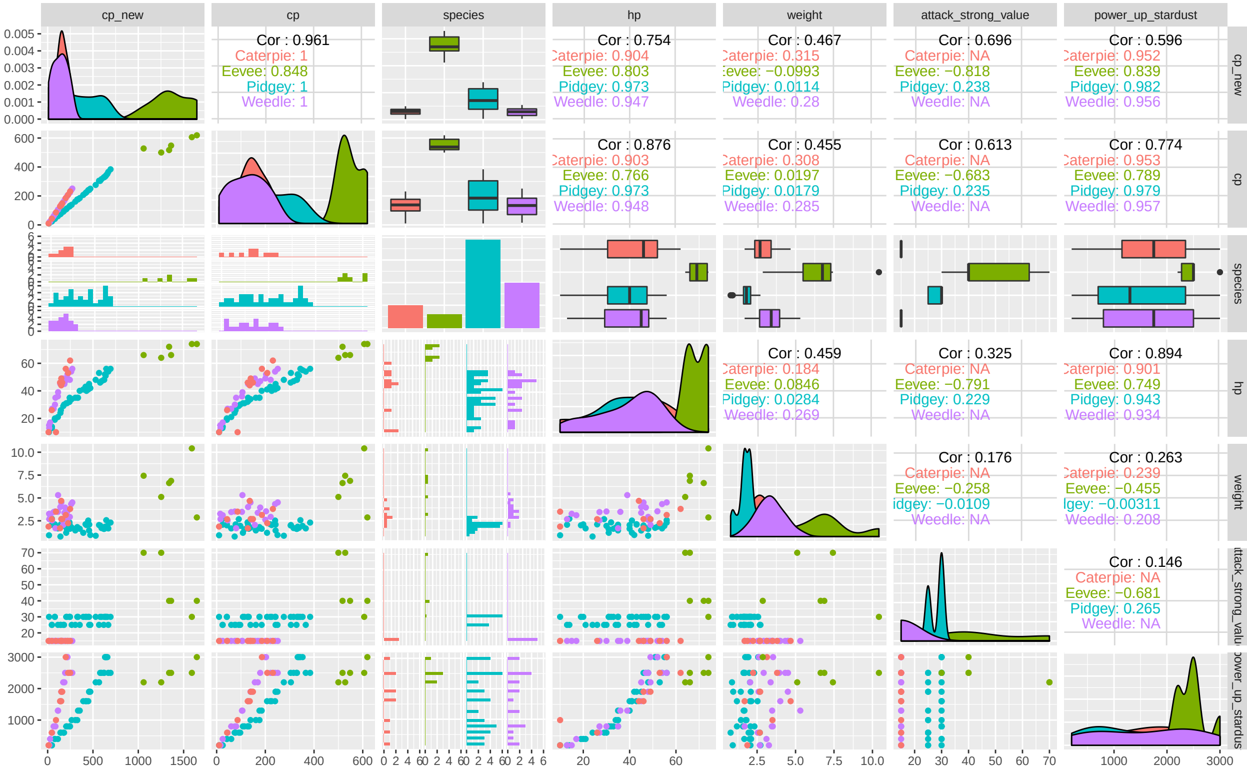
The first thing we did was comb through the dataset again to spot any outliers or weird data. As we hinted towards in Part II of our Pokemon Go regression project, all the Eevee data points seem to be outliers (their stats were way higher than the other Pokemons' stats). A question will we look into is what will the model look like if the Eevee outliers are removed.

The second thing, which we didn't catch the first time around, was that for Weedle and Caterpie, their *attack_strong_value* was lower than the *attack_weak_value*. At first glance, this didn't make sense because the *attack_strong_value* should be *higher* than the *attack_weak_value*. The name of the strong attack is "Struggle", and upon further research, we found that Struggle is actually the default second move. In other words Weedle and Caterpie don't have two attacks, one strong and one weak, but rather just a single attack. We still chose to include *attack_strong_value* in our model fitting process because it might contain information important for predictions.

The next thing we did was check the relationships between our explanatory variables as well as between the explanatory variables and the response variable.

Correlation Plots

```
ggpairs(pokemon2, aes(col = species))
```



Here, we've only included certain variables for ease of presentation; however, you can see the full correlation plot on our GitHub. When looking at the pairs plots between continuous variables, several things stood out to us.

1. cp_{new} is highly correlated with cp and hp. It's telling that cp is highly correlated with cp_{new} because that means that a pokemon with a higher starting cp can be expected to have a higher cp_{new} post-evolution.
2. Cp is highly correlated with hp, so perhaps we don't need both in the model if we are looking for a more parsimonious model. Although, since we are trying to fit a predictive model, maybe having more predictor variables might be to our benefit (more variability explained)
3. Previously, we had log-transformed hp because there were issues of non-constant variance when plotted against cp_{new}. Looking at the pairs plots, there may be the same issues of non-constant variance with cp and hp_new, which will have to be checked with residual plots.

Model Building

Note on Interactions

```
p1 <- qplot(x = hp, y = cp_new, color = species, data = pokemon) +
    geom_smooth(method = "lm")
grid.arrange(p1, nrow = 1, ncol = 1, bottom = "Figure 1: Plot of hp vs. cp_new demonstrating interactio
```

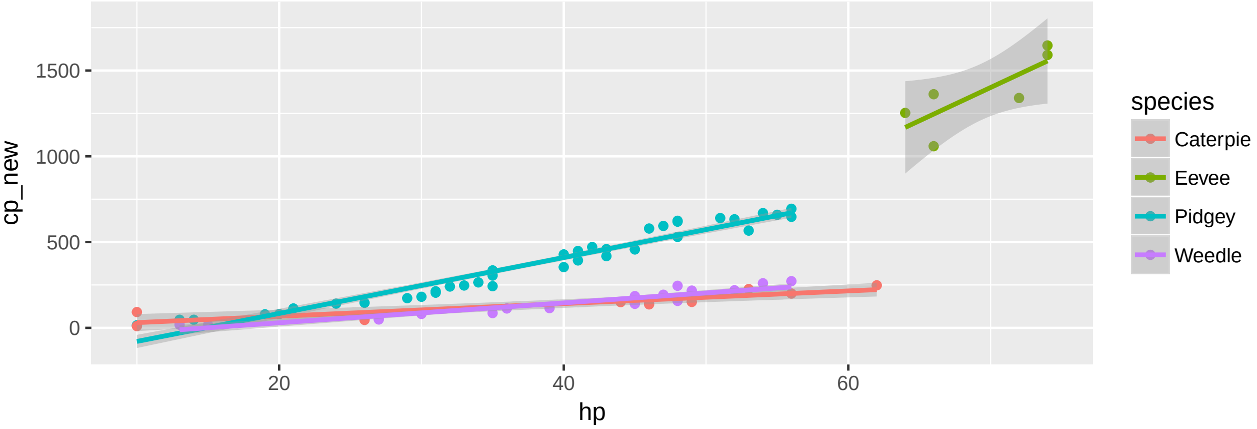


Figure 1: Plot of hp vs. cp_new demonstrating interaction by species

Since our last installment working on this data project, one of the things we noticed was that the cp_{new} vs hp model did not adequately describe the Eevee data points. Here, we decided to color by species, and from this plot (Figure 1), it's clear that species interacts with hp in predicting cp_{new} (the slopes are different). In other words, the effect of hp on cp_{new} depends on the species of Pokemon.

Based on layman knowledge of the game, it would make sense that these effects change by species since some species are naturally stronger or weaker (Eevee is a mystical mammal while Caterpie and Weedle are essentially your everyday garden worms in the Pokemon universe).

Thus, in the model building process outlined below, we've decided to provide the algorithms with the choice of including interaction terms.

Model Selection

We tried different model building procedures, specifically best subsets and forward and backward stepwise regression. Briefly, best subsets looks at all combinations of predictors and picks the best based on the following criteria that we chose: BIC, R_a^2dj, and Mallow's C_p and stepwise regression starts with an empty model and adds the best available variable to the model at each step making sure to drop variables that become less important as the model selection process moves forward.

We feed the algorithm with all variables including interaction terms. The best model from this round of selection was: cp_{new} ~ cp + species + $attack\_strong\_value$ + weight + cp:species + species:$attack\_strong\_value$ + species:weight which resulted from stepwise regression. The selection criteria we used were AIC, R^2, and R^2_{adj}.

Diagnostics

Residuals

```
p2 <- ggplot(data = pokemon_logged, aes(x = cp_new_log, y = res2),
    ylab = "Residuals", xlab = "Cp_new", title = "Pokemon Cp_new") +
    geom_point() + geom_hline(yintercept = 0, linetype = "solid")
grid.arrange(p2, nrow = 1, ncol = 1, bottom = "Figure 2: Residual Plot of Logged Model")
```

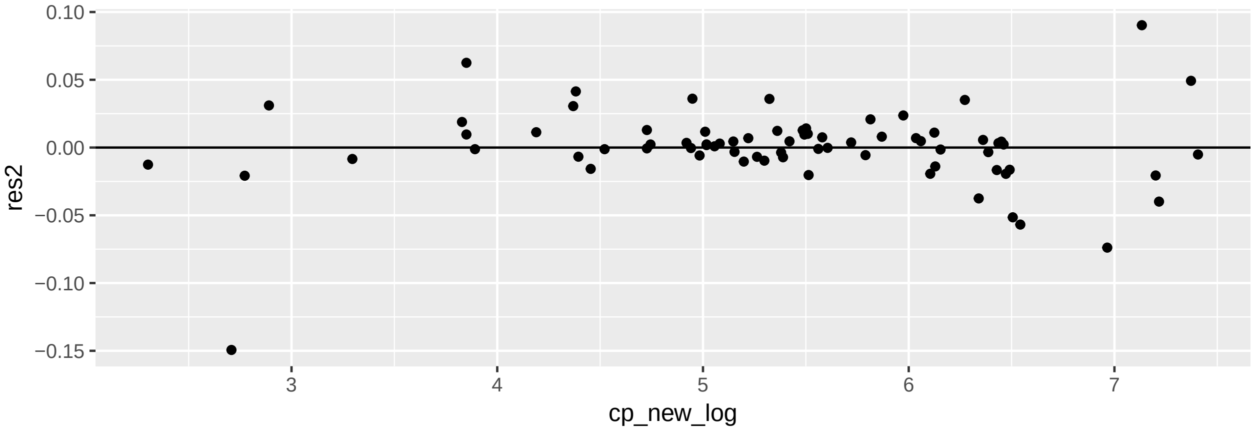


Figure 2: Residual Plot of Logged Model

We created a residual plot on our first round model cp_{new} ~ cp + species + $attack\_strong\_value$ + weight + cp:species + species:$attack\_strong\_value$ + species:weight. We observed unequal and non-normal variance which corresponded to the Eevees and issues with heteroscedasticity, so we decided to log-transform cp_{new} as well as cp, since we noticed that cp had a wide range of values during our exploratory analysis. As a result, our log-transformed values resulted in the "random scatter" that we love seeing in residual plots (Figure 2). Having log-transformed variables will improve our model appropriateness.

Outliers and Influential Points

We ran a Bonferonni test procedure to see if we had outlying Y residuals, those whose studentized deleted residuals were large in absolute value. We also ran diagnostic tests to look for influential points. We ran a test of DFFITS, which is the measure of the influence pokemon i has on model-fitted cp_{new}. From DFFITS, we saw that all the Eevees had influence on the model-fitted cp_{new}. We also ran a test of DFBETAS, which is the measure of influence pokemon i has on k^{th} regression coefficient. From DFBETAS, we saw that none of the data points had influence on the regression coefficient. Furthermore, we ran a test of Cook's Distance, which is the measure in change in regression by removing each individual point. From Cook's Distance, we saw that 4 of the Eevees had influence on the regression line. Lastly, we ran a VIF test, but we ran into an "aliased coefficients" error which indicated that our model was highly multicollinear such that VIF was essentially undefined.

Revisiting Model Selection and Diagnostics

We re-ran our model selection process on log-transformed *cp_new* and *cp* (renamed in our dataset as *cp_new_log* and *cp_new* respectively) which we discovered were necessary for inference from our residual plots. The resulting optimal model including interaction terms was cp_new_log ~ $cp\_log + species + attack\_strong\_value + hp + power\_up\_candy + species : hp + cp\_log : species + species : power\_up\_candy + species : attack\_strong\_value$. The AIC was -509 and the R^2 and adjusted R^2 were both 0.999. However, after running a VIF test, we encountered the same "aliased coefficients" error which resulted from the large number of interaction terms included in the model. Therefore, the "aliased coefficients" error indicated that our model was highly multicollinear such that VIF was essentially undefined. This led us to question if interaction terms should be eliminated to obtain a more interpretable model while not sacrificing the model's predictive power. Therefore, we re-ran the selection process excluding the interaction terms. The new best model is cp_new_log ~ $cp\_log + species + attack\_strong\_value + hp + weight$ with AIC of -502 and R^2 and adjusted R^2 of 0.999. The mean VIF of this model is 4.68.

We have strong reason to believe that we should choose the reduced model which has no interaction terms since it has a similar AIC score and equally high R^2 and adjusted R^2 as the full model with interaction terms.

Yet the reduced model is not multicollinear and is easier to be interpreted. A residual plot of this model also shows that the technical conditions for inference were satisfied. We thus selected *cp_new_log ~ cp_log + species + attack_strong_value + hp + weight* as our final model.

Interpreting Betas and the Model Overall

Just as a quick aside, we wondered what the model would look like in a dataset without Eevees, so we ran the same model selection steps. While we won't describe it here, feel free to check it out on our GitHub!

When we performed a nested F-test between the final model that we chose and our full model with all interaction terms, we obtained a p-value of 0.003, indicating that we reject H_0 that all the β's of the additional variables in the full model (namely, the betas of the interaction terms) equal 0, and conclude that at least one of the coefficients was significantly nonzero. However, we decided not to select the interaction model for reasons of overfitting that we've discussed above.

1. Holding all other variables constant, a doubling of cp is associated with a multiplicative change of $2^{1.04}$ in the median of cp_{new}, and this is extremely significant, meaning that we reject H_0 that $\beta_{cp} = 0$.
2. Holding all other variables constant, having an Eevee is associated with a multiplicative change of $e^{0.992}$ in the median of cp_{new}, and this is extremely significant, meaning that we reject H_0 that $\beta_{speciesEevee} = 0$.
3. Holding all other variables constant, having a Pidgey is associated with a multiplicative change of $e^{0.591}$ in the median of cp_{new}, and this is extremely significant, meaning that we reject H_0 that $\beta_{speciesPidgey} = 0$.
4. Holding all other variables constant, having a Weedle is associated with a multiplicative change of $e^{0.0177}$ in the median of cp_{new}, and this is not significant, meaning that we fail to reject H_0 that $\beta_{speciesWeedle} = 0$.
5. Holding all other variables constant, an increase in *attack_strong_value* of 1 unit is associated with a multiplicative change of $e^{-0.00456}$ in the median of cp_{new}, and this is extremely significant, meaning that we reject H_0 that $\beta_{attack\_strong\_value} = 0$.
6. Holding all other variables constant, an increase in hp of 1 unit is associated with a multiplicative change of $e^{-0.00197}$ in the median of cp_{new}, and this is significant, meaning that we reject H_0 that $\beta_{hp} = 0$.
7. Holding all other variables constant, an increase in *weight* of 1 unit is associated with a multiplicative change of $e^{-0.000604}$ in the median of cp_{new}, and this is not significant, meaning that we fail to reject H_0 that $\beta_{attack\_strong\_value} = 0$.

Lastly, our diagnostics still indicate that Eevees are influential, so it is important to note that out model could look very different if there were no Eevees in he data. Our mean VIF was 4.68, indicating no serious issues of multicollinearity.

Coefficient of Partial Determination

1. $R^2_{Ycp|species,attack\_strong\_value,hp,weight}$ The variability in cp_{new} remaining after modeling cp_{new} using species, *attack_strong_value*, hp, and weight, is reduced by a further 99
2. $R^2_{Yspecies|cp,attack\_strong\_value,hp,weight}$ The variability in cp_{new} remaining after modeling cp_{new} using cp, *attack_strong_value*, hp, and weight, is reduced by a further 96.1
3. $R^2_{Yattack\_strong\_value|cp,species,hp,weight}$ The variability in cp_{new} remaining after modeling cp_{new} using cp, species, hp, and weight, is reduced by a further 31.2

4. $R^2_{Y hp|cp,species,attack\_strong\_value,weight}$ The variability in cp_{new} remaining after modeling cp_{new} using cp, species, *attack_strong_value*, and weight, is reduced by a further 7.67

5. $R^2_{Y weight|cp,species,attack\_strong\_value,hp}$ The variability in cp_{new} remaining after modeling cp_{new} using cp, attack_strong_value, hp, and weight, is reduced by a further 2.83

From this analysis, we see that cp and species are by far the more important variables in the model. The other variables do contribute to the reduction in variability.

Confidence and Prediction Intervals

After we selected our model, we wanted to see it's prediction ability, so we generated a prediction interval for an individual Pidgey with characteristics we arbitrarily chose. 95% of all Pidgeys with cp = $e^{5.3}$, *attack_strong_value* of 25, hp of 41, and weight of 3 have cp_{new} between $e^{5.75}$ and $e^{6.13}$.

In addition to prediction, we were also interested in inference, that is, finding a confidence interval. For the same set of characteristics, we are 95% confident that the median cp_{new} for a Pidgey with cp = $e^{5.3}$, *attack_strong_value* of 25, hp of 41, and weight of 3 is be between $e^{5.92}$ and $e^{5.95}$.

A Brief Aside on p-values

We ran 8 hypothesis tests: the nested F-test that we performed on our final model and the t-tests for each of the β coefficients. Our p-values would be multipled by 8, which would not have changed the magnitude of any of the significances of our p-values.

Conclusion

The final model we selected was *cp_new_log* ~ *cp_log* + species + *attack_strong_value* + *hp* + *weight*, which was generated from a stepwise linear regression, based on several criteria: AIC, R^2, and $R^2_a dj$ as well as a guiding principle of avoiding overfitting without compromising the model's highest predictive potential. Although we saw hp and cp to correlated during our exploratory analysis, there were no serious issues of multicollinearity. We were not surprised by the huge importance that cp and species had in explaining the variability of cp_{new}. We were surprised that weight was kept in the model since it didn't explain a lot of variability in cp_{new}; however, given that we are attempting to generate primarily a predictive model, including weight, which provide extra predictive information, is in our best interests.

Overall, we think our model is informative in that it guides Pokemon Go players to focus on specific aspects of their Pokemon to be able to get a sense of whether their evolved Pokemon will have a high cp.

Sources

- OpenIntro (https://www.openintro.org/stat/data/?data=pokemon). This is where downloaded the csv for our data.
- baptiste (https://stackoverflow.com/questions/30299529/ggplot2-define-plot-layout-with-grid-arrange-as-argument-of-do-call). How we make nicely arranged graphs.
- sape research group (http://sape.inf.usi.ch/quick-reference/ggplot2/colour) Colors for ggplot are great