

Avery Hutchinson

ist 718  - Final Project

PokémonGO : Battle suggestions

Table of Contents

[Introduction: 2](#_Toc137393362)

[Background: 2](#_Toc137393363)

[Data Description: 2](#_Toc137393364)

[Meta Data: 2](#_Toc137393365)

[Specification: 4](#_Toc137393366)

[Observation: 4](#_Toc137393367)

[Data Cleaning and Preparation: 4](#_Toc137393368)

[Analysis: 6](#_Toc137393369)

[Basic Statistics and Viz: 6](#_Toc137393370)

[Mean, Median, Min and Max 6](#_Toc137393371)

[What’s the count? 8](#_Toc137393372)

[What’s the spread? 8](#_Toc137393373)

[Sometimes pie charts are the right answer 9](#_Toc137393374)

[Relationships, Correlation, and Word Clouds: 9](#_Toc137393375)

[What’s the correlation? 9](#_Toc137393376)

[Modeling: 11](#_Toc137393377)

[Chi-Square test 11](#_Toc137393378)

[Linear Regression 12](#_Toc137393379)

[Machine Learning: 13](#_Toc137393380)

[Random Forest 13](#_Toc137393381)

[Time to predict some winners….and losers 14](#_Toc137393382)

[Recommendation/Conclusion: 14](#_Toc137393383)

[Sources: 15](#_Toc137393384)

[Data Source: 15](#_Toc137393385)

# Introduction:

## Background:

Pokémon first debuted as a game in February of 1996, when it was launched in Japan as an electronic game series. The first two versions were called Pokémon Green and Pokémon Red. In 1998, the game went global and was introduced to the US and then to the UK in 1989 as a Red and Blue version instead of Green. There have been eight more generations of the game since: Gold and Silver, Ruby and Sapphire, Diamond, and Pearl, Black and White, X and Y, Sun and Moon, Sword and Shield and Scarlet and Violet. In between these generations there are also other games that were created with different story lines, but the above generations mark the introduction of new Pokémon species to the franchise.

While it was originally created as an electronic game to be played on a Game Boy console, the game was also turned into a trading card game (TCG). The first set was released in October 1996 and included only 102 cards. There have now been more than 30 billion cards printed. The Pokémon franchise then expanded to a smart phone app in 2016, called PokémonGO. This allowed anyone with a smartphone to play the game by walking around outside and catching Pokémon utilizing the augmented reality feature.

The franchise is well known around the word now and they have a foothold in nearly every aspect of entertainment ranging from the TCG, to television series, to electronic game cartridges, to an app. The PokémonGO game has a pokedex with approximately 800 different Pokémon available for catching, all with various data points such as name, pokedex number, type, species, generation, abilities, attack, defense, speed, etc.

## Data Description:

For this project I analyzed multiple data sets, all sourced from Kaggle, containing different statistics of all Pokémon since 1996. This includes Pokémon from the first generation through the sixth generation. All data files are csv that can be opened in excel and imported into a python program. The files contain structured and unstructured data. Each file has a different make up of rows and columns but after merging all the files together, the final data set I will be working with consists of 721 rows of individual Pokémon and 47 columns of statistics for each Pokémon. Below is a breakdown of the original csv files and their structures.

* 1 csv file with 800 rows and 13 columns of data
* 1 CSV files with 1045 rows and 51 columns of data
* 1 CSV file with 1008 rows and 45 columns of data
* 3 CSV files
  + 1 with 800 rows and 12 columns of data for machine learning purposes
  + 1 broken into training data for machine learning purposes
  + 1 broken into testing data for machine learning purposes

### Meta Data:

Sourced from: https://www.kaggle.com/code/lucasfca/pokemon-legendary-prediction

Pokedex Data:

* **pokedex\_number**: The entry number of the Pokemon in the National Pokedex
* **name**: The English name of the Pokemon
* **german\_name**: The German name of the Pokemon
* **japanese\_name**: The Original Japanese name of the Pokemon
* **generation**: The numbered generation which the Pokemon was first introduced
* **issublegendary**: Denotes if the Pokemon is sub-legendary
* **is\_legendary**: Denotes if the Pokemon is legendary
* **is\_mythical**: Denotes if the Pokemon is mythical
* **species**: The Categorie of the Pokemon
* **type\_number**: Number of types that the Pokemon has
* **type\_1**: The Primary Type of the Pokemon
* **type\_2**: The Secondary Type of the Pokemon if it has it
* **height\_m**: Height of the Pokemon in meters
* **weight\_kg**: The Weight of the Pokemon in kilograms
* **abilitiesnumber**: The number of abilities of the Pokemon
* **ability?**: Name of the Pokemon abilities
* **ability\_hidden**: Name of the hidden ability of the Pokemon if it has one

Base stats:

* **total\_points**: Total number of Base Points
* **hp**: The Base HP of the Pokemon
* **attack**: The Base Attack of the Pokemon
* **defense**: The Base Defense of the Pokemon
* **sp\_attack**: The Base Special Attack of the Pokemon
* **sp\_defense**: The Base Special Defense of the Pokemon
* **speed**: The Base Speed of the Pokemon

Training:

* **catch\_rate**: Catch Rate of the Pokemon
* **base\_friendship**: The Base Friendship of the Pokemon
* **base\_experience**: The Base experience of a wild Pokemon when caught
* **growth\_rate**: The Growth Rate of the Pokemon

Breeding:

* **eggtypenumber**: Number of groups where a Pokemon can hatch
* **eggtype?**: Names of the egg groups where a Pokemon can hatch
* **percentage\_male**: The percentage of the species that are male. Blank if the Pokemon is genderless.
* **eggcycles**: The number of cycles (255-257 steps) required to hatch an egg of the Pokemon

Type defenses:

* **against\_?**: Eighteen features that denote the amount of damage taken against an attack of a particular type

# Specification:

The main objective of this project is to use different methods of analysis such as: correlations, descriptive statistics, categorical comparisons, and of course lots of visualizations such as graphs and word clouds to gain insight into possible relationships among the various Pokémon. I will first focus on simple exploratory analysis to determine things like the average height among all Pokémon, patterns between legendary status and HP, typical Base Stat Totals for each type\_1 of Pokémon, etc.

The second part of the analysis will be more be involved and focus more on relationships and things that might point to whether a Pokémon will win a battle or not. For example, one idea to explore would be which statistics have the most influence on a Pokémon’s Base Stat Totals or Attack points. Another idea would be looking into the correlations between the number of weaknesses and the type of Pokémon to determine which Pokémon type have the least number of weaknesses and therefore might be good to use in most battles.

Finally, I will focus on implementing a machine leaning algorithm with the goal of inputting two different Pokémon and outputting the theoretical winner of a battle.

# Observation:

## Data Cleaning and Preparation:

I first started by loading the csv files into a Jupyter Notebook within a python environment. For much of the program, I worked with the 3 main csv files that included all the Pokémon statistics. Later, when it was time to program the machine learning algorithm, I imported the final files that had the Pokémon combat results, but not necessarily all the statistics.

As previously stated in an earlier section, these data files were quite but they included everything I needed so there was no need to go out and research missing data points like the different attack points or defense points for all Pokémon. Some files also have blanks for data that’s not applicable to a certain Pokémon and some files use NULL to represent this. There is also one file that uses quotes for every string category (figure 1) which will need to get fixed when joining the files. Figure 2 shows the breakdown of information and the first few columns for one of the files.

Figure1: Figure 2:

A screenshot of a phone

Description automatically generated with low confidence A screenshot of a computer program

Description automatically generated with low confidence

After the data files had all been imported, it was time to start cleaning the data. I started by dropping columns in each file that were not going to be useful for the analysis goals of this project (figure 3). I then had to rename one of the columns to the exact format as the other files so that I could use that column to merge the files on (figure 4). I ended up using the Pokedex number as that merge column since that number is the same all around the world. This means that I could join any file in the future to this project as long as I had the pokedex number column filled out.

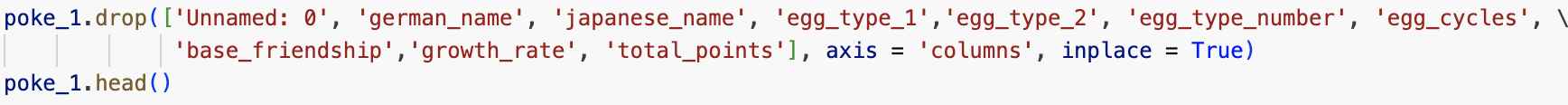
Figure 3:

Figure 4:

A screenshot of a computer code

Description automatically generated with low confidence

The end result was a data frame with 1792 rows and 59 columns(I had missed dropping a few of the duplicate columns). I began cleaning up the data some more by removing the duplicate columns, dropping rows with the word “mega” in the Pokémon name, and fixing white space and quotation mark issues. I dropped the rows with the word “mega” in them for two reasons: first, one file had the word mega listed first, and then other had it listed last which would have kept two copies of the Pokémon based on name. Second, when comparing a Mega version to a normal version of Pokémon, the fight will not be fair and the Mega will always win. Additionally, there isn’t a Mega version for all Pokémon so I felt that leaving these stats in the data would end up skewing the analysis and algorithm later on.

I then had the problem with multiple duplicate rows for each Pokémon because I joined the files to add columns of data but it left me with many duplicates. To fix this, I ran a program to compute the completeness level of each row and then drop all subsequent rows. If there was a tie of the completeness level, I had the program keep the first row it came across. This can be seen in below in figure 5. The result was a clean data frame with 721 rows/pokemon and 45 columns.

Figure 5:

A screenshot of a computer code

Description automatically generated with low confidence

# Analysis:

\*Please see Jupyter Notebook and Markdown file for full code, results, and command descriptions.

## Basic Statistics and Viz:

### Mean, Median, Min and Max

Figure 6:

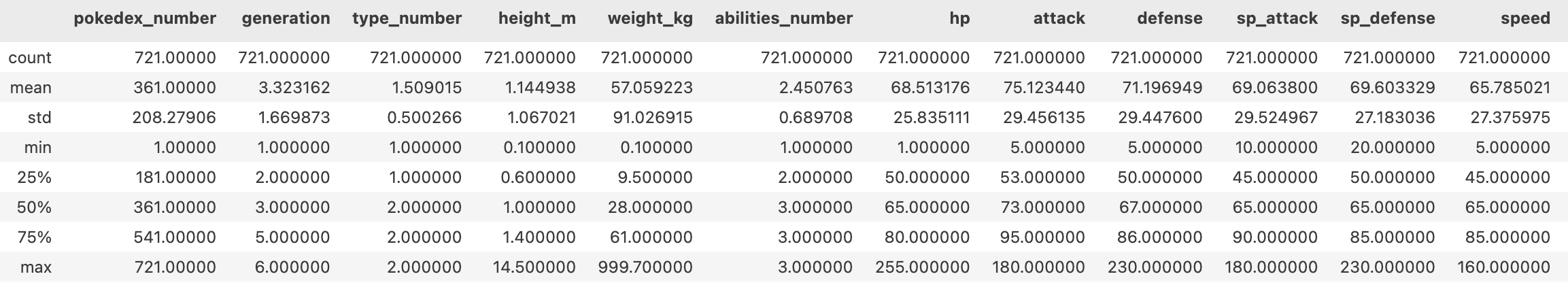


Figure 6 above shows a breakdown of the basic statistics across a few of the columns in the data set. Some things to notice are that the mean generation is 3 which makes sense since the Pokémon data sets, we are working with go from first generation and stop at the sixth generation. The average height of a Pokémon is 1.5m while the tallest is 14.5m. This might be important in determining winners of battles later. The average attack is 74 while the max is 180 and the average defense is 68 and max in 230. Attack, HP and Defense points are all very important in determining a winner of a battle and the overall strength of a Pokémon.

Figure 7:

A picture containing text, screenshot, line, diagram

Description automatically generated

The boxplot in figure 7 shows the distribution of heights among all Pokémon. We can see that the average height is about 1 to 1.5 meters, but we can also see a majority of outliers starting from around 2.5 all the way to around 14.5.

Figure 8:

A picture containing text, screenshot, diagram, line

Description automatically generated

Figure 9:

A picture containing diagram, text, screenshot, line

Description automatically generated

Figure 10:

A picture containing text, diagram, line, plot

Description automatically generated

Figures 8-10 are additional boxplots representing different distributions across categories in the data. Figure 8 shows the distribution of Pokémon across the 6 generations and their game or origin. Figure 9 represents the variation of health points related to the status of Pokémon. We can see that the average HP of legendary Pokémon vs Normal Pokémon is a great deal higher. However, there are multiple Normal Pokémon who have significantly higher HP than any of the legendary or mythical Pokémon. Figure 10 shows the spread of Base Stat points across the different types of Pokémon. It appears that Dragon types have higher Base Stats overall and might be good to have in your battle teams.

### What’s the count?

Figure 11:

A picture containing text, screenshot, diagram, plot

Description automatically generated

Figure 12:

A picture containing text, screenshot, plot, diagram

Description automatically generated

Figure 11 is an easy was to visualize the number of abilities most Pokémon have across all types of Pokémon. For example, there are only about 5 Grass types that have only 1 ability. The majority (approx. 40) have two abilities, and approx. 20 have 3 abilities. Figure 12 shows the typical status you can expect to find across all types. It appears that the approx. half of the dragon types are Legendary status.

### What’s the spread?

Figure 13:

A screen shot of a computer screen

Description automatically generated with low confidence

Figure 13 shows a scatter plot of the Pokémon data where each Pokémon is plotted on a graph based on the attack points and their defense points. They are then color coded based on which generation they are from. The plot also allows for the ability to hover over a point and the Pokémon name and stats will show up, as can be seen with Avalugg.

### Sometimes pie charts are the right answer

Figure 14:

A picture containing text, diagram, screenshot, circle

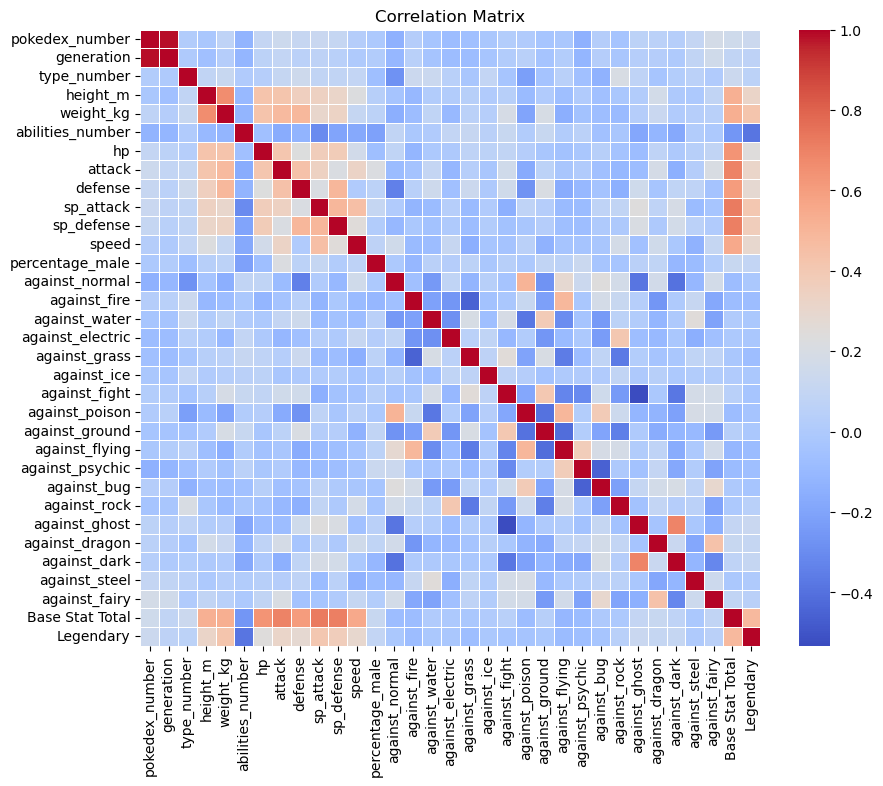
Description automatically generated

Figure 14 represents a pie chart of the percentage of all Pokémon that are from a specific generation. The exploded piece represents the generation with the highest percentage.

## Relationships, Correlation, and Word Clouds:

### What’s the correlation?

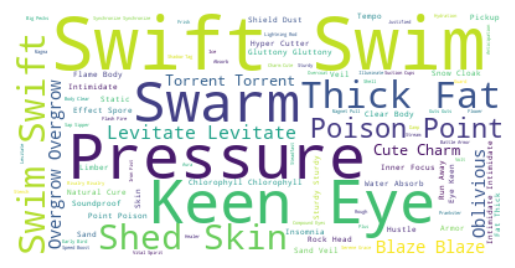
Figure 15:



The above heat map and correlation matrix shows various levels of correlation between different variables. It appears that most variables are not strongly correlated with each other but there are some that are strong that we might want to investigate later on. For example, the 'against\_' categories being correlated are interesting because this suggests that there are multiple types a Pokémon might be weak against. If a Pokémon is strong against poison, it also appears that they would be weak against normal types as well.

Figure 16: Figure 17:

A picture containing text, font, design, typography

Description automatically generated 

These word clouds are a great way to visualize the distribution of the unstructured data. The larger the word, the more times it shows up in the specified data. For example, in figure 16, we can see the most common type\_1 is water, then normal, then bug, and so on. This can be verified by figure 18 below.

Figure 18:

A picture containing text, receipt, font, screenshot

Description automatically generated

## Modeling:

### Chi-Square test

Figure 19:

A screenshot of a computer program

Description automatically generated with medium confidence

The above result suggests a significant discrepancy between the observed and expected frequencies. The p-value is very close to zero suggesting there is strong evidence to reject the null hypothesis, meaning that there is a significant association between type\_1 and legendary variables. This could mean that Legendary Pokémon fall into a small number of Type categories. Further evidence can be seen below in figure 20.

Figure 20:

A picture containing text, screenshot, font, receipt

Description automatically generated

Figure 21:

A picture containing text, font, screenshot, white

Description automatically generated

Figure 21 is the result of a chi-square test comparing Base Stat Total and the Mega evolution of a Pokémon. Interestingly, the p-value is quite high which suggests that there is no significant association between Base Stat Total and the Mega evolution of a Pokémon, which goes against what I originally hypothesized.

### Linear Regression

Figure 22:

A screenshot of a computer

Description automatically generated with low confidence

In figure 22, we see the results of a linear regression analysis run to determine what, if any, factors have a strong effect on making a string Pokémon based only off Base Stat Total points. The results are a very high R-Squared and adjusted R-squared value as well as a high F-statistic which all suggest this is a good fitting model. However, each of the independent variables has a very different p-value and not all of them are helpful to the model. The variables that we want to focus on will have p-values lower than .05. These are hp, attack, defense, speed, sp\_attack, against\_fairy, against\_dark, against\_bug, and against\_grass. With this information, we will want to try again with a new model containing just the variable with the lower p-values to get a model that really fits so we can verify what independent variable has the largest effect on the dependent variable which is Base Total Stat.

Figure 23:

A screenshot of a data

Description automatically generated with low confidence

The new model looks much better overall. We still have a high R-Squared vale and a high F-statistic, but now, all the p-values of the variables are much closer to the .05 threshold. Looking at the coefficients, we can see that against\_dark had the greatest overall effect on a Pokémon’s Base Stat Total. For every 1 unit increase in against\_dark, Base Stat Total will rise by 6.6734. It is likely then, that Pokémon that are weak against Dark types, will have a higher Base Stat Total overall and might be Pokémon we want to be collecting.

\*see Jupyter notebook for further linear regression models

## Machine Learning:

### Random Forest

For this project I used random forest to classify the Pokémon data and to assist in a machine learning algorithm that will be able to predict the winner of a battle. I chose random forest because that is what many people suggested works best for this type of data. This is where I ended up importing the other csv files to use to train the program. I did have to do a little cleaning and preparation for these files which required a lot of research into the correct syntax, but I was luckily able to find helpful hints and solutions online. I chose to use the one-hot encoding method to concatenate the data frames because I prefer working with the Pandas program and overall have better luck getting the syntax to work and make sense. Figure 24 shows the accuracy results of the model.

Figure 24:

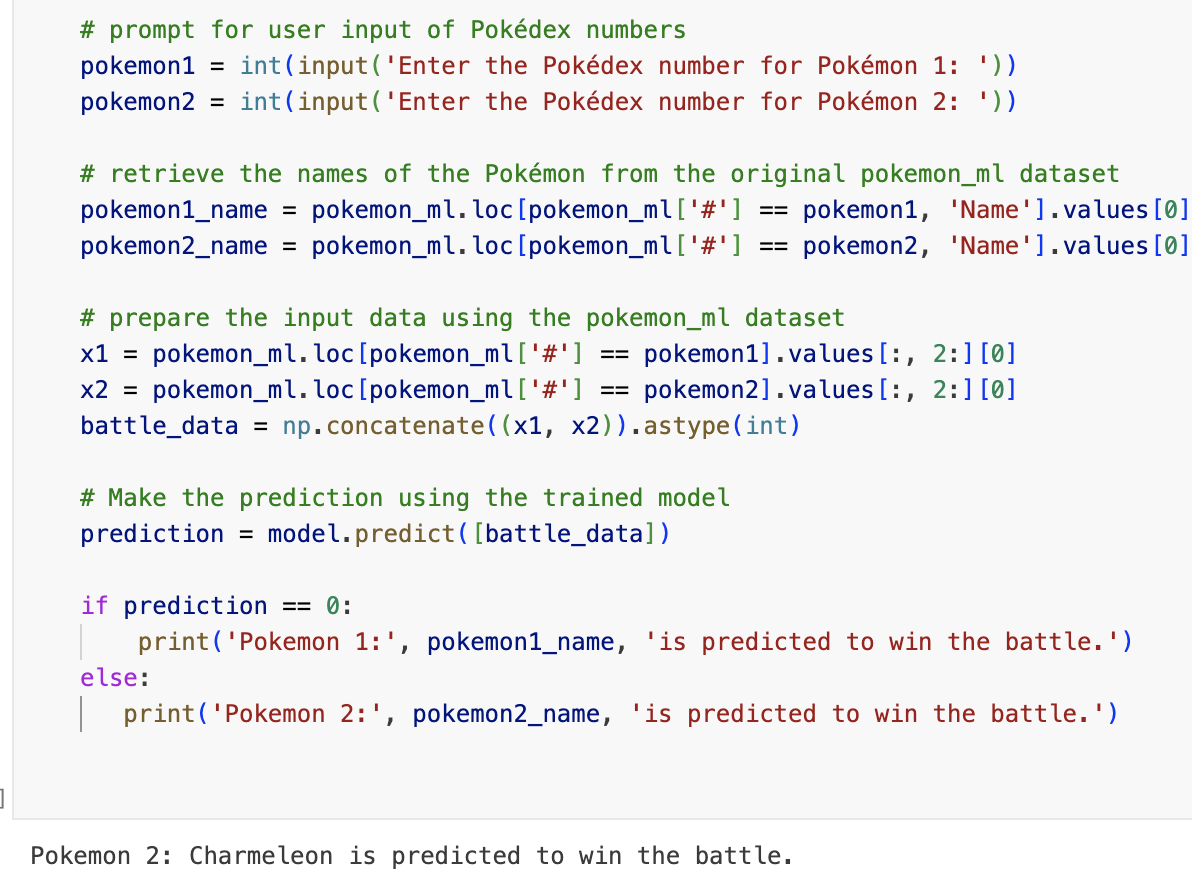
A picture containing text, receipt, screenshot, font

Description automatically generated

The model achieved an accuracy of 0.93472! This means that it correctly predicted the outcome of battles for approximately 93.47% of the Pokémon. The precision score for class 0(which was the losing Pokémon) is .93 meaning that out of all the Pokémon predicted as losers, 93% were classified correctly. The same can be said for the winners (1). Support just represents the number of instances for each class: class 0 had 5957 instances, and class 1 had 6543 instances.

### Time to predict some winners….and losers

Figure 25:



­Figure 25 shows the code to run the program against the original Pokémon data (which I copied over to a new variable called Pokémon\_ml to use specifically for the machine learning). The resulting program prompts the user to input the pokedex number for 2 different Pokémon, runs the algorithm and then outputs the calculated winner.

# Recommendation/Conclusion:

After analyzing all the data and running various programs I have a much greater insight into the world of Pokémon. There is still an enormous amount of knowledge that can still be collected from the world with more analytics and programs run. The biggest take away from this project is the ability to input two different Pokémon into the program and project a winner based on real time statistics.

On a personal note, as a player of the game and as a farther of 2 kids who play Pokémon, this has been a great project to work on and involve my kids into. We have run this program many times and it has caused a lot of great (and heated) debates between all members of the family. In the future when I have time, I would like to add to this program and adapt it to result in a team of 3 Pokémon that have the greatest chance to win the majority of battles in PokémonGO or even be able to suggest the best team given an opposing teams information. For example, being able to input 3 of my son’s Pokémon into the program and then have it suggest 3 Pokémon from my own pokedex that would beat his. The sky is the limit I think with this, and I can see the huge advantages machine learning can have not only on games like Pokémon, but even events like horse racing or soccer finals, etc.

# Sources:

## Data Source:

1: <https://www.kaggle.com/datasets/terminus7/pokemon-challenge?select=pokemon.csv>

2:<https://www.kaggle.com/datasets/mariotormo/complete-pokemon-dataset-updated-090420?select=pokedex_%28Update_04.21%29.csv>

* <https://pokemondb.net/>
* https://www.serebii.net/

3: <https://www.kaggle.com/datasets/mrdew25/pokemon-database>

4: <https://www.kaggle.com/datasets/abcsds/pokemon>

* <http://www.pokemon.com/us/pokedex/>
* <http://pokemondb.net/pokedex>
* http://bulbapedia.bulbagarden.net/wiki/List\_of\_Pokémon\_by\_National\_Pokédex\_number