CICS 397A Final Project Write Up

By Jarrod Daniels

Intro

Ever since I was a kid, video games have always had a place in my heart. They have been a fun pass time as well a way to meet and engage with new people. Growing up, one of my favorite games was Pokémon. Pokémon is a series of RPG styled games developed by Game Freak and published by Nintendo. The series has been a massive hit, with the franchise having made an estimate of over $92 billion in revenue and being a common household name. Within the game, there are hundreds of creatures known as Pokémon (also referred to as “mons” or “pocket monsters”). Each Pokémon has a unique set of classifiers including their type (the elemental properties associated with both the Pokémon and their moves. All Pokémon have a primary type but not all Pokémon have a secondary type.) as well as individual numerical stats such as health points, attack, defense, special attack, special defense, speed, height, weight, and the base total of their battle stats. Within the series, there are also Pokémon that are classified as being legendary. As the title may imply, Legendary Pokémon are considered to be extremely rare and powerful as well as being prominent figures with legends and myths of the game’s lore.

My goal for this project can be split up into three main parts. First, I wanted to conduct basic exploratory analysis to get an idea of what the distribution of stats across all Pokémon looked like as well as see who some of the front runners were for specific statistics. Second, I wanted to see if I could build various classifiers in an attempt to build a model that would predict if a Pokémon was legendary based upon its stats. Finally, I wanted to make my own distance function that could look at a single Pokémon and determine the 5 Pokémon that are most similar to it.

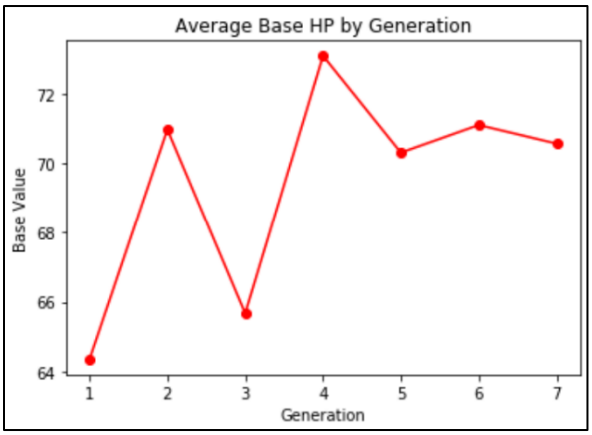
(Since this was a solo project, each portion of the project was done by myself)

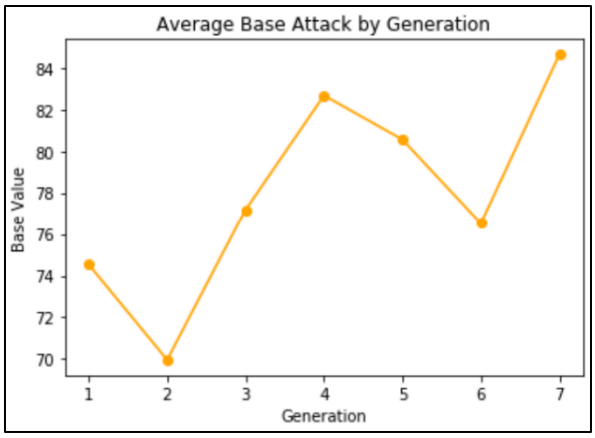
Data

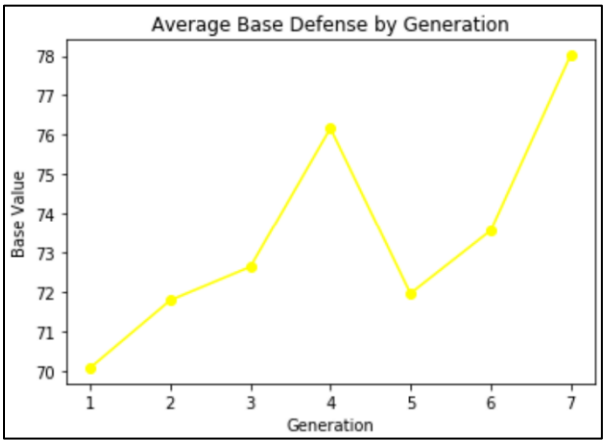
For this project, I was able to find a dataset off of the website https://www.kaggle.com/ That included every Pokémon from generations 1-7 (a generation is essentially the iteration of the game) as well as plethora of their data. The data set however was not very clean (there were many extra columns that included irrelevant data and columns that were out of order) and as a result I had to remove many of the columns and order the remaining data into a clean and usable csv file. This final file is labeled as simply ‘pokemon.csv’ and can be viewed in the folder that this document is attached to. Once I had cleaned up the csv file, I was able to create a pandas data frame with it in Python which I then used for the duration of the project.

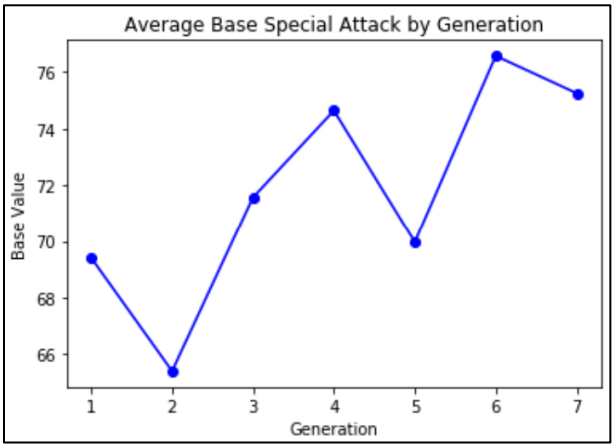
Plots/Visualizations and Learning Outputs

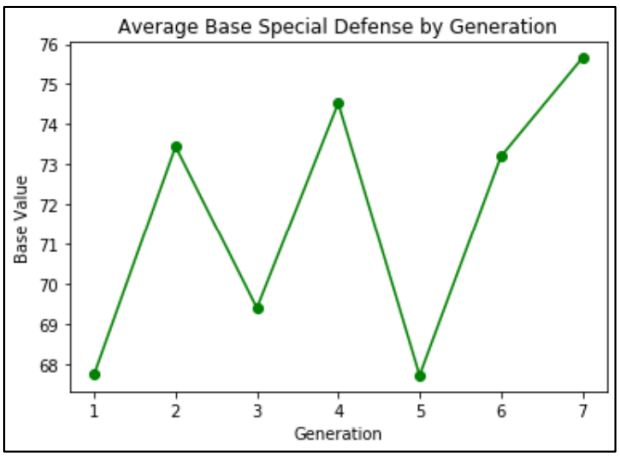
As mentioned before, for the first part of the project I wanted to conduct exploratory data analysis on the data in order to better understand the distribution of Pokémon stats. The first set of plots seen below is the average base stats (base stats are one of the underlying values that help determine the growth rate for a given Pokémon) for the 6 primary battle stats across each generation of Pokémon (all of these graphs were made using the matplotlib and seaborn library).

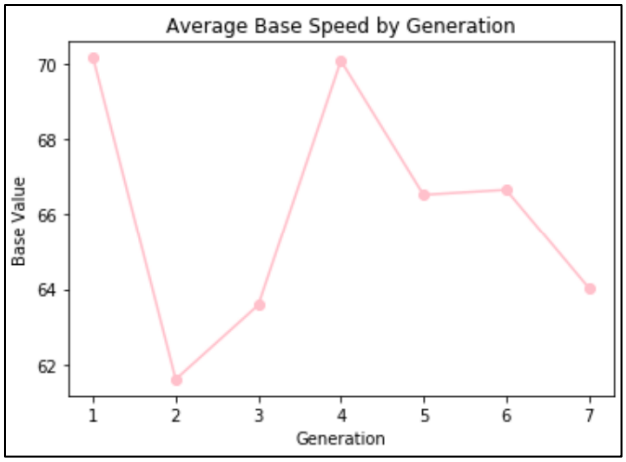




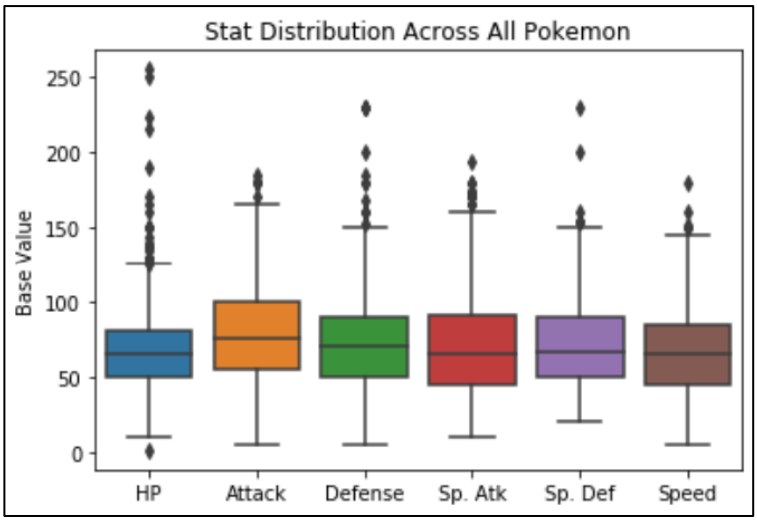


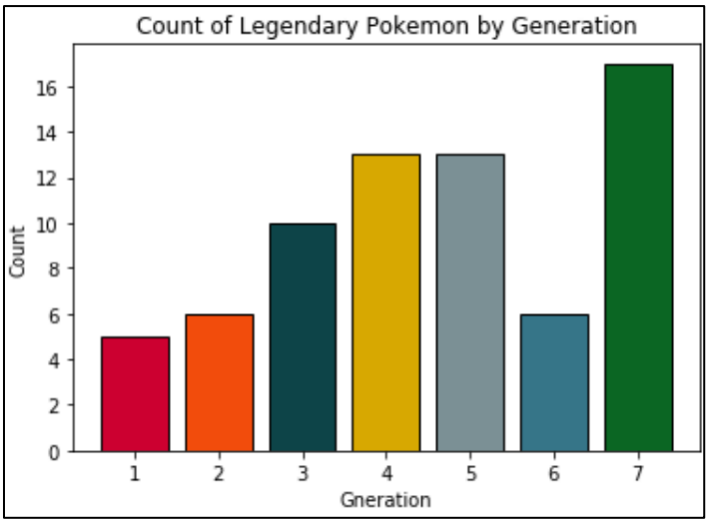






Next, I wanted to see what the overall spread of stats was across all Pokémon, regardless of which generation they were from as well as the number of legendary Pokémon from each generation (this will be relevant for the next section).





Moving onto the next portion of the project I wanted to use various types of learning models to see if it would be possible to predict whether a Pokémon was a legendary based upon its total stats, individual battle stats, and its weight/height. To do this, I used the Scikit-learn library and constructed a classification tree, nearest neighbors' classifier, and a classification tree that utilized a 5-fold cross validation. The results of all three are seen below as well as the features that the learning algorithm deemed to be the most important.

```
decision tree training acc: 0.9783, test acc: 0.9652
```

(Decision Tree accuracy)

```
k-nn 1 test acc: 0.9652
k-nn 3 test acc: 0.9652
k-nn 5 test acc: 0.9602
k-nn 7 test acc: 0.9602
k-nn 9 test acc: 0.9602
k-nn 19 test acc: 0.9602
k-nn 39 test acc: 0.9453
k-nn 79 test acc: 0.9154
k-nn 159 test acc: 0.9055
k-nn 319 test acc: 0.9055
```

(K-nn accuracy scores at various n neighbors)

```
decision tree 5-fold cross-validation acc: 0.9688
```

(5-fold cross validation decision tree results)

For the final part of the project I created my own distance formula that attempts to find the 5 most similar Pokémon of any given Pokémon. I will go into more detail about this in the code description later. However, the results for the input of the Pokémon Bulbasaur is seen below.

```
In [327]: main3('Bulbasaur')
['Oddish', 'Budew', 'Bellsprout', 'Foongus', 'Chikorita']
```

Code Overview

(All code was written and run in Spyder and the code was compiled by calling functions in the console rather than the terminal)

All the code for this project is contained within the python file PokeProject.py which is contained within the same zip file as this document. The python file has 3 different main functions that you can run. The function main() contains all of the code for exploratory data analysis portion of the project (including the graphs). It primarily utilizes pandas, seaborn, and matplotlib and primarily involves looking at basic stats such as averages across the data set. It can be run by calling “main()” in the console.

The second function main2() contains all the code for the model building portion of the project. For this portion of the code, I primarily used functions from the sklearn library as well as other libraries such as pandas and numpy. For this portion of the project, I was able to modify my code from the 4th homework assignment as well as utilize code provided to us in the skeleton code from homework 4. This portion of the code can by ran by simply calling “main2()” in the console

The final function main3() contains all the code for the distance function of the project. For this portion of the code, I modified preexisting code from homework 3 in order to successfully read in the data in a way that it could be used to calculate distance function. I then modified my distance function, so that distance would be calculated based upon factors like The Pokémon’s primary and secondary type, the difference in their overall stats as well as the difference in their individual stats. Once a distance number was calculated for a given Pokémon, the knn function would sort all the distances for a given Pokémon and produce the 5 lowest (making sure to not include itself of course). This function can be called by calling main3(‘Pokémon Name’) into the console.

Challenges:
I would say the biggest challenges for this project came from cleaning/preparing the data for each portion of the project as well as creating my own Pokémon distance function. When originally loading in the csv, the data was all out of order and some of the functions were rejecting the data initially. I also ran into issues working with pandas and iterating through data to manipulate it. With the distance function, the challenge came with making sure that the distance metrics were being calculated correctly (it was dealing with strings and number) as well as making sure that the distance numbers were weighted well. After overcoming all these however, the project really came together, and all the functions worked as I had intended to.

Insights:
One interesting insight I received on this data was when creating the graphs of average stats across each generation. Often fans complain that certain generation of Pokémon are to slow or are too squishy (this means low hp stat). One complaint that is brought up often is that generations 6 & 7 are stereotyped as having very slow Pokémon but when we look at the averages, we can see that they are about in the middle with generations 2 & 3 being slower. Another piece I found to be quite interesting is that there are some Pokémon types that are more frequently seen as primary types rather than secondary types. A good example of this is if we look at the output of main(), we can see the there are only 3 Pokémon with a primary type of flying but 95 that have it as their second type (something I never really picked up while playing these games).

Future and Conclusion:

In the future I would love to build off and improve the distance function I made with other data that is available on these Pokémon. For instance, some Pokémon have type advantages over others (water beats fire, fire beats grass, grass beats water etc.) as well as special abilities and natures that were not considered during this project. It would also be interesting to look at stat distribution across types (how does HP compare when looking at normal and dark types). Finally, I would want to look more in depth at stat distribution and classification by evolutions (many Pokémon have evolutions that boost their stats significantly. Some Pokémon evolve twice while other do not evolve at all). Overall, this has been an interesting and fun project that has allowed me to apply what I've been learning in the classroom this semester to something that has been a large part of life outside of school.

Note:
I am not sure how this code is being run but I would heavily suggest it be run using the Spyder 4.0 (I programmed the whole project in it). I have created a 4th function called main4() that will run all three parts of the code at once with part 3's input being ('Squirtle'). I have also put this function at the bottom of my code so that it should run when you hit the run button. Please contact me if there are any issues with running the code.