1) Introduction and Motivation

Pokemon Go is this year's highest grossing mobile game released this Summer by Niantic Labs (a Google subsidiary). Termed "immersive reality", the game relies on users moving throughout areas and catching virtual "Pokemon" that appear on the screen.

I frequently play this game walking to and from class. As a computer science major, I naturally looked into the online developer community for the game. Pokemon Go turned out to be much less random than it appeared. Using an application from a github programming team (Christopher 2016), I was able to create and use bot accounts to scan areas I would otherwise be unaware of. As I used the application, I began to notice trends and patterns in spawns.

2) Problem definition

Not all Pokemon spawns are created equal. Pokemon, by definition, vary in name, shape, combat power, and other attributes. Therefore, certain species of Pokemon are more desirable to acquire than others. Knowing this disparity, it becomes valuable to know spawn points and patterns. Scanning revealed two main quantitative structures, spawn points and nests.

Spawn points are the methodology on which Pokemon spawn. Throughout the scan area we are able to identify various location points that spawn Pokemon. 'Spawning' Pokemon occurs at an exact minute of every hour the spawn point is active and produces a catchable specimen for 15 minutes. Spawn points are "active" for 15 of every 60 hours, or about 25% of the time.

Nests consist of a collection of spawn points in specific areas. These spawn points are extremely likely to spawn a specific species of Pokemon. The scanning program revealed that Audubon Park's spawns were unlike other areas on the map.

The goal of this project is to provide users with information about their chances to find a particular species of Pokemon in a specific location. Knowing this information, they can improve their likelihood of acquiring desirable Pokemon.

3) Solution?

Seeing Audubon Park's trends and patterns of spawn points I wanted to find a quantifiable way of defining a nest. Whenever I would visit Audubon Park I would notice a strange density of Squirtles. In an immersive reality game, it would make sense for a turtle to spawn by the water, but there was such a unique density to it. In my attempt to define a nest, I knew there were squirtles, but specifically I wanted to know how many squirtles and when over a period of time.

This disparity in spawns are only notable inside the Audubon Park area. Fifteen steps off of the green section of your map pointed to spawns returning to normal. To prove the existence of a nest at Audubon Park, I would compare data from spawn locations inside Audubon Park versus data from spawn locations outside of Audubon Park.

4) Data?

Initial input data collection

To collect data for this experiment, I scanned 1600m in every direction from McAlister stadium for a week. For simplicities sake, I decided to focus upon 10 spawn points. Every spawn point has two characteristics; a longitude latitude location, and a 15 hour spawn timer based on an hour and minute combination specific to each spawn point. This combination operates on a 60 hour timer that resets back to 00:00 at 12:00AM Sunday every week.

The first five of these spawn points are within Audubon Park. The sixth spawn point sits on the LBC quad. The seventh is located near our very own Stanley Thomas. Our eighth point is contained at Devlin Fieldhouse. Our ninth point greets local patrons at Bruno's. Lastly our tenth point is located over by Felipe's.

Data reveals that three of the five spawn points will be active on any given day. Spawns frequently passing over multiple days due to the 15 hour active duration. All spawn points are in consistent with a 15 hour active duration followed by a 45 hour sleep duration. These amounts never changed throughout the experiment.

Format

Each spawn event is represented as a record containing three numbers: hour of the day, Pokemon Species Type(Eevee vs Pidgey vs Squirtle), and ID of the point where the spawn occurred.

Training data

Our training data set is stored in a .csv format. Each .csv file contains data from the same 10 spawn points, 5 within Audubon Park, and 5 outside of Audubon Park. Each .csv file contains the spawns over 24 hours of a particular day of the week. Each hour has a number corresponding to the spawn, 0 if no spawn, 1 if squirtle, 2 if bulbasaur, 3 if eevee, 4 if pidgey, or 5 if other. Training data is read into the LogisticRegressionModel and will predict ratio of spawns for a 24 hour period based on the inputted data set. We have 7 total files .csv that can be used as training data.

Test data

Our testing data is stored in a .csv format. Each .csv file contains data from the same 10 spawn points, 5 within Audubon Park, and 5 outside of Audubon Park. Each .csv file contains the spawns over 24 hours of a particular day of the week. Each hour has a number corresponding to the spawn, 0 if no spawn, 1 if squirtle, 2 if bulbasaur, 3 if eevee, 4 if pidgey, or 5 if other. This testing data's actual ratio is compared against the prediction ratio of a LogisticRegressionModel built upon data from the training set. We have 7 total .csv files that can be used as testing data.

Output

The output of this model represented in two numbers.The first represents the ratio of Squirtles to all other pokemon spawns for our 5 spawn points inside Audubon Park. The second represents the ratio of Squirtles to all other spawns for the 5 spawn points outside of Audubon Park for our 5 spawn points outside Audubon Park. The difference between these numbers tells us the difference in density and number of predicted squirtle spawns for inside and outside of Audubon Park. These ratios are further explained in the results section.

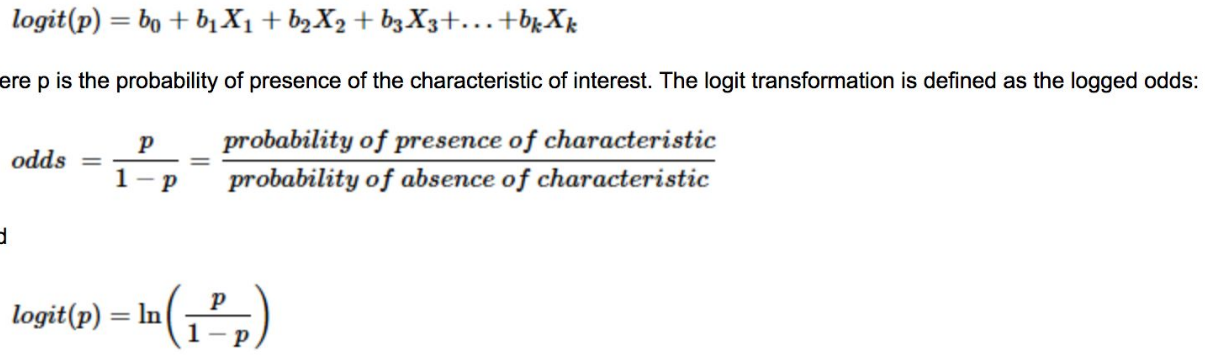
5) Machine learning design

Due to the nature of the 60 hour timer, the weekly midnight Sunday reset to 00:00, and making exact Pokemon spawn/despawn times, we are able to accurately predict spawn points. I created a learning program to predict Pokemon spawns for our ten spawn points.

To help read the data, I created a PokeCsvReader converts data from a .csv file(PokeInput) into ArrayLists of a custom class called Pokemon. Each Pokemon stored in these arraylists have a ID that corresponds to their species (1=Squirtle, 2=Bulbasaur, 3=Eevee, 4=Pidgey, 5=Other), an integer value that stores when in the spawn cycle they spawned, and a classifier of whether that Pokemon is in Audubon Park(sourced from spawn points 0-5) or not from Audubon Park(sourced from spawn points 6-10).

For our seven days of data, I was able to create a learning program that can predict future spawn locations and times across different days. For our training and testing, I compared our days of spawn data against each other. The LogisticRegressionModel class represents a Java representation of logistic regression. The constructor contains a double, rate, to represent the rate of change in training test data. The weights array in the constructor represents our eventual output. The weights array in our experiment contains two doubles, one to represent the number of squirtles as a ratio of all spawns in Audubon Park, and outside of Audubon Park. The intent of this model is to show that Audubon Park operates as a squirtle nest and spawn points inside will have significantly higher ratios than outside the park.

Logistic Regression Definition:



$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k$$

where p is the probability of presence of the characteristic of interest. The logit transformation is defined as the logged odds:

$$odds = \frac{p}{1-p} = \frac{probability\ of\ presence\ of\ characteristic}{probability\ of\ absence\ of\ characteristic}$$

and

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

In our model, “probability of presence of characteristic” is defined as the ratio of squirtle spawns within Audubon Park for that particular day of data. Correspondingly, “probability of absence of characteristic” refers to the ratio of squirtle spawns outside of Audubon Park. These two numbers will always add together to 1.

Lets consider an example of 60 total spawns where 12 of those spawns were Squirtles. In addition we know that 9 squirtles spawned inside Audubon Park and that Audubon Park had a total of 28 spawns. We also know that the remaining 3 squirtles were outside Audubon Park and spawns outside Audubon Park total’d 32 for the day.

Our P for Audubon Park squirtles should be 9/28 or .3214. Therefore our odds equation is as follows:

Odds(SquirtleinAud) = (.3214 / .6786) = .47362216

The "logit transformation" is defined as the logged odds, therefore:

LOGIT TRANSFORMATION(squirtlesVSotherInAud) = ln(.47362216) = -0.74734541

Using the logit transformation table (https://www.medcalc.org/manual/logit_transformation_table.php)

We can see for logit(p)=-0.74734541 , the probability of p having a positive outcome, or p being a squirtle inside of Audubon Park, is just above .32 or 32%.

We also know P for Non Audubon Park squirtles will be 3 / 32 or .09375. Therefore our odds equation is as follows:

Odds(SquirtleNotAud) = (.09375 / .90625) = .10344828

The "logit transformation" is defined as the logged odds, therefore:

LOGIT TRANSFORMATION(squirtlesVSotherNonAud) = ln(.10344828) = -2.2686835

Using the same logit transformation table we can see that for logit(p)=-2.2686835, the probability of p having a positive outcome, or p being a squirtle spawn outside of Audubon Park, is about .095 or 9.5%

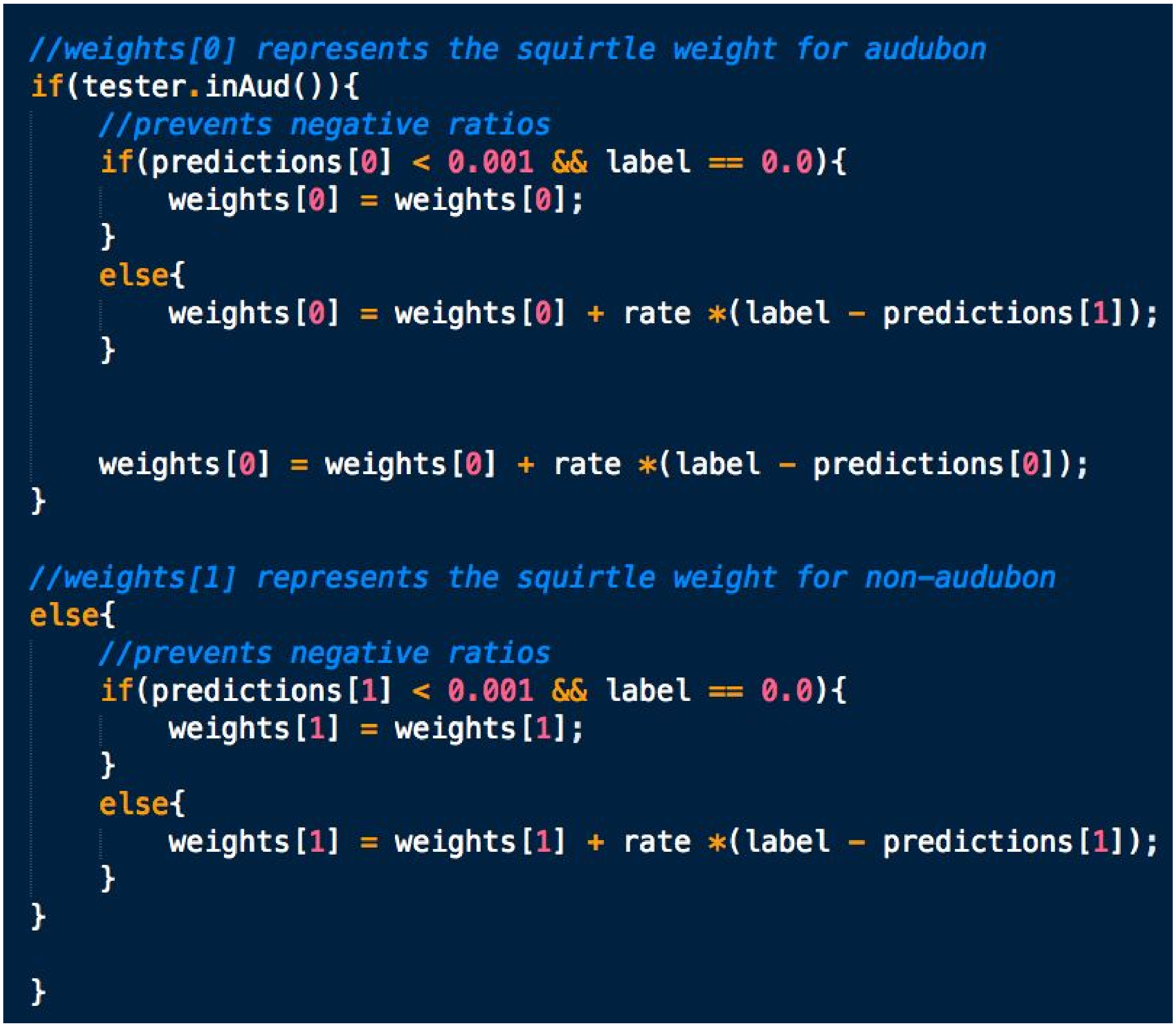
The model follows this design to show the difference between the two results. Differences between these two area ratios will show the difference in spawn distribution of the specified species between the two distinct areas.

6) Machine learning algorithm

The base model for implementation of this Logistic Regression Model was borrowed from a github project wishing to implement a logistic regression model in Java (Peng 2016). The first step of this model is to pick a training set of data and a testing set of data. Our model uses the attached .csv files and files may be used both for training and testing. Once these data sets are designated, the model is constructed using LogisticRegressionModel(2). The (2) refers to the number of areas being compared in this model, the first being Audubon Park and the second being Non-Audubon Park.

Once created, the LogisticRegressionModel may use the train command with the

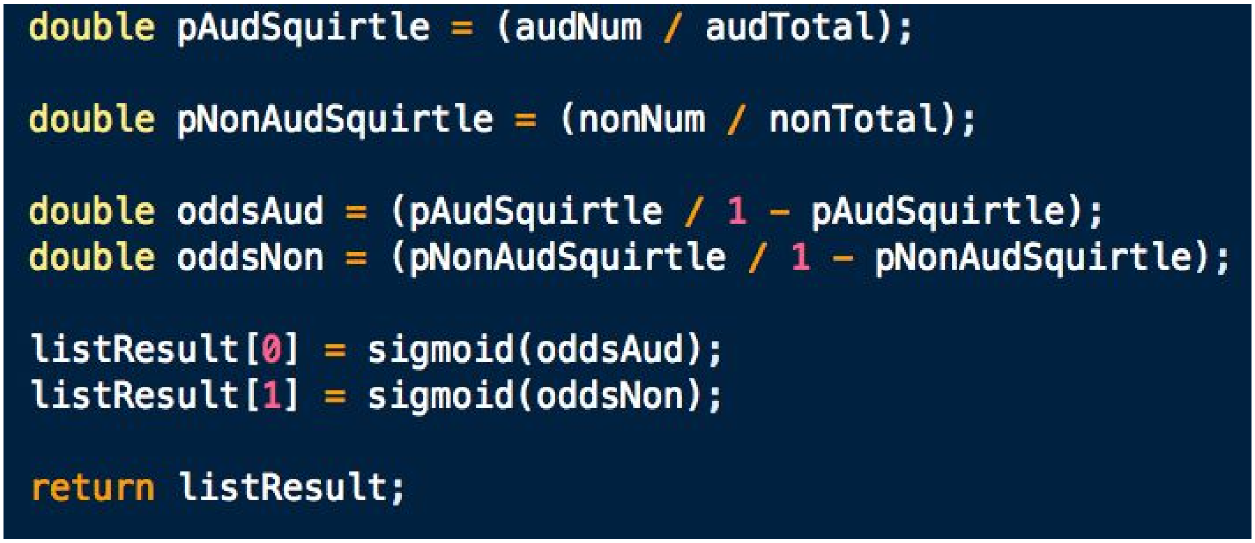
training data set. The train command works breaking down the inputted training data into individual Pokemon units and recording the three characteristics of spawn hour, spawn species, and ID of spawn point. After recording statistics from the training set, the model changes the projections of the two ratios based on the newly learned information using an adjustable weight and rate formula implemented in the model. The model will now be able to generate a prediction of squirtles spawns inside and outside of Audubon Park for a 24 hour period based on the learned training data. The code below shows a single (of 3000) iteration of the LogisticRegressionModel learning from the training data through changing the weight[0] or weight[1] variables.



```
//weights[0] represents the squirtle weight for audubon
if(tester.inAud()){
    //prevents negative ratios
    if(predictions[0] < 0.001 && label == 0.0){
        weights[0] = weights[0];
    }
    else{
        weights[0] = weights[0] + rate *(label - predictions[1]);
    }


    weights[0] = weights[0] + rate *(label - predictions[0]);
}

//weights[1] represents the squirtle weight for non-audubon
else{
    //prevents negative ratios
    if(predictions[1] < 0.001 && label == 0.0){
        weights[1] = weights[1];
    }
    else{
        weights[1] = weights[1] + rate *(label - predictions[1]);
    }
}

}
```

Once the model is able to predict these ratios, we perform a similar squirtle ratio analysis of data for the testing data set. The model tests the data by going through each

individual Pokemon unit and recording their three characteristics. After recording data from the entire training set, the model finds the actual ratio of Audubon squirtles to all other species spawns inside Audubon and the actual ratio of Non-Audubon squirtles to all other species spawns outside Audubon Park.



```
double pAudSquirtle = (audNum / audTotal);

double pNonAudSquirtle = (nonNum / nonTotal);

double oddsAud = (pAudSquirtle / 1 - pAudSquirtle);
double oddsNon = (pNonAudSquirtle / 1 - pNonAudSquirtle);

listResult[0] = sigmoid(oddsAud);
listResult[1] = sigmoid(oddsNon);

return listResult;
```

Now that we know the actual results of our testing data, we compare it against our training data and predictions from our model to find our results. The model displays the ratio difference between the actual result from our training data set, and our predicted result using our training data. A positive result such as 0.03 means that our model predicted a 3% higher ratio of squirtle spawns than the actual result. This means that the model overestimated the number of squirtles. A negative result such as -0.05 would mean that our model predicted a 5% FEWER ratio of squirtle spawns than the actual result.

The importance of this comparison is to show the accuracy of our model. By comparing the differences in predictive ratios versus actual ratios, we can show consistency of a species across different days. By comparing difference in ratios inside and outside of Audubon Park, we can show species density of different areas.

The user designates the specific day of the training data and has our LogisticRegressionModel instance learn from that day's spawn data through the train command. The model then produces predictions about that data through the predict function. The user then has the data we will test our trained model against designated in the testList variable. By using the arrayClassify command, the model produces two separate ratios output meant to represent the predicted ratio of squirtles inside and outside of Audubon park for the set of data. These two ratios are then compared against the actual ratios of squirtles inside and outside of audubon park and presented to the user. [0] represents the ratio of Audubon squirtle spawns compared to all other areas in Audubon and [1] the ratio of Non-Audubon squirtles against every other spawn. Next, the difference of these two ratios is then calculated and presented as output to the user as a single variable that shows the correlation of Audubon Park to squirtle spawns as compared to all other areas.

7) Your Results

Model accuracy shows the extent at which we can predict spawn species and spawn quantities for two distinct areas. The results of this program are the differences between our predicted results generated from our training data and the actual results from our training data. By comparing the differences in predictive ratios versus actual ratios, we can show consistency of a particular species across different days. By comparing difference in ratios inside and outside of Audubon Park, we can show species density of different areas.

The value of these ratios characterize users chances of catching a specific pokemon (in our model, squirtle) in the two areas of interest. Based on these ratios, the user can make an informed decision of which area the user should go if they is looking for a particular species.

Training and testing our data showed consistently high ratios of Squirtles in Audubon Park across different days. Our model predicted a low of 75.2% of squirtle spawns being within Audubon Park and a high of 88.9%. The actual ratios of squirtles within Audubon Park never dropped below 75% and rose as high as 97%. As a result, our predictions showed the highest “wrongness” at .22 or 22% between the lowest and highest ratio days. In addition, the ratio of Squirtles outside of Audubon Park remained below 25% in every result and always showed a lower variance than the inside difference.

Training and testing our data showed that slightly more than .75 or ¾ of squirtle spawns occurred within Audubon Park. The average ratio, about .7757 or 77.57% can be used for another example of how our logistic regression model runs. For this example, the dependent characteristic is spawn location inside of Audubon Park (1) or outside (0).

We also know P for Audubon Park squirtles versus non Audubon Park squirtles will be approximately .77.57 or 77.57%. Therefore our odds equation is as follows:
Odds(SquirtlesinVSoutAud) = (.7757 / .2243) = 3.45831476
In other words, we have 3.4x better "odds" of finding a squirtle in Audubon compared to outside of it, assuming equal number of spawn points considered.
The "logit transformation" is defined as the logged odds, therefore:
LOGIT TRANSFORMATION(squirtlesVSotherNonAud) = ln(3.45831476) = 1.24078
Using the same logit transformation table we can see that for logit(p)=1.24078, the probability of p having a positive outcome, or p being a squirtle spawn outside of Audubon Park, is about .7757 or 77.57%. Knowing this across all of our data sets, we can say that squirtles in Audubon Park spawn at about 3:1 ratio compared to those outside of Audubon Park.

Knowing that Audubon Park is approximately three times more likely to spawn a Squirtle than outside of it, the user is able to make a more informed decision of traveling to Audubon Park if they wish to capture larger quantities of squirtles than can be found in other locations. This model can be used by users seeking to catch a particular species of pokemon, and attempting to decide between different areas to catch it. By generating prediction ratios and testing those predictions, users are able to make informed decisions of which areas contain the Pokemon they seek.

This model can be replicated and used to fit any two designated areas and a particular species. For example, City Park is a well known Bulbasaur nest. This experiment could be made to fit City Park by comparing 5 City Park spawn points versus 5 spawn points outside of City Park. By specifying the desired species as Bulbasaur instead of Squirtle, the user would be able to use the model and predict bulbasaur ratios for the two areas. Since Bulbasaur and Squirtle share almost identical game rarity, it is a comparable study and would be very replicable.

By comparing spawn diversity and times across different days, patterns began to emerge. Because Sunday resets the 60 hour spawn clock, there is a schedule every week that spawn points hold to. Due to the patterning of 24 hour days into a 60 hour cycle, 2.8 iterations of the 60 hour cycle pass per week. This means that the last 12 spawns of a 60 hour cycle are not spawned three times in a week, but rather only two, regardless of location. This makes spawn points that do not occur in the 48-60 hour segment of the cycle inherently more valuable.

The data showed high correlation between Audubon Park spawn similarities across days, but less so for spawn points outside the park. In addition, data shows that Tuesday, Wednesday, and Thursday all have unique spawning patterns and times. Data from these three days produce similar spawn diversity when compared to test overall data from different days, but do not share spawn timings or patterns with any other day of the week.

This means that there exists two pairs of spawn cycles, Friday paired with Sunday and Saturday paired with Monday.. Friday and Sunday have matching spawn times and locations and therefore produce identical data. This means that using Friday to train our PokeLearner will always produce predictions that perfectly match Sunday's testing data. The same can be said if we train with Sunday's data to test against Friday's data. This property also exists for the pairing of Saturday and Monday.

Our model predicted a Squirtle correlation to Audubon Park of about .75 or 75%. In other words, our model predicted a ratio of about 3 Squirtles spawns inside of Audubon Park for every Squirtle spawn outside of it across an equal number of spawn points representing inside and outside of Audubon park.

8) Related work, are there other results?

The next logical step would be to find another nest and run a similar test. City Park is widely known as a nest, and a similar train and test to predict spawn times, locations, and diversity would produce similar, repeatable results. If these results are confirmed, we can reproduce these ratios for different species dependent on the nest. If City Park is a Bulbasaur nest, we could perform a similar test to mine checking for the ratio of Bulbasaur spawns inside and outside of City Park.

9) Conclusion

By using patterns in our data, we were able to perfectly predict a spawn's time, place, and species. Since Friday/Sunday and Saturday/Monday share spawn schedules, we are able to train data from one and test it against another with perfect accuracy. If we use data from other, unconnected days, we able to get a good guess at the spawn diversity and their locations, but unable to get a good grasp on predicting when those spawns will occur.

Our Logistic Regression Model produces a prediction output that three squirtles would spawn inside Audubon Park for every Squirtle outside of it, assuming an equal number of spawn points. This clustering of a specific species proves our intention of quantifying a nest.

Sources:

Christopher, M. (2016, August). Mchristopher/PokemonGo-DesktopMap. Retrieved December 05, 2016, from https://github.com/mchristopher/PokemonGo-DesktopMap/releases

Peng, T. (2016). tpeng/logistic-regression. Retrieved December 05, 2016, from https://github.com/tpeng/logistic-regression/blob/master/src/Logistic.java