Pokemon Go and Tulane/Loyola Universities

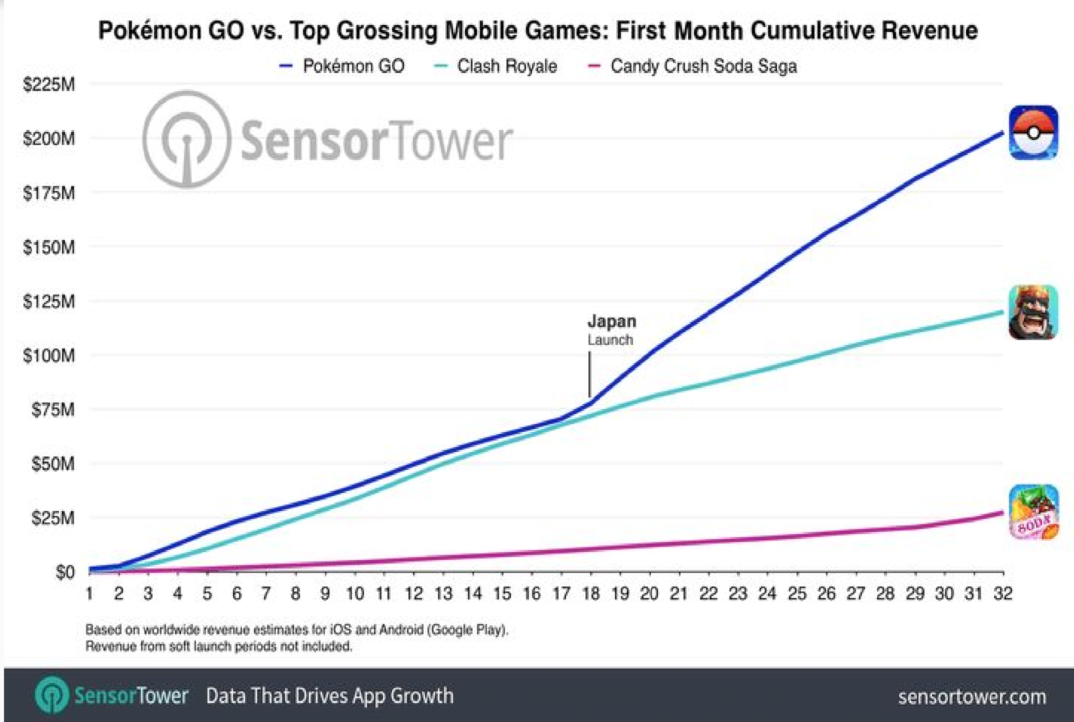
Taylor Huntington
Machine Learning
Fall 2016

Itinerary

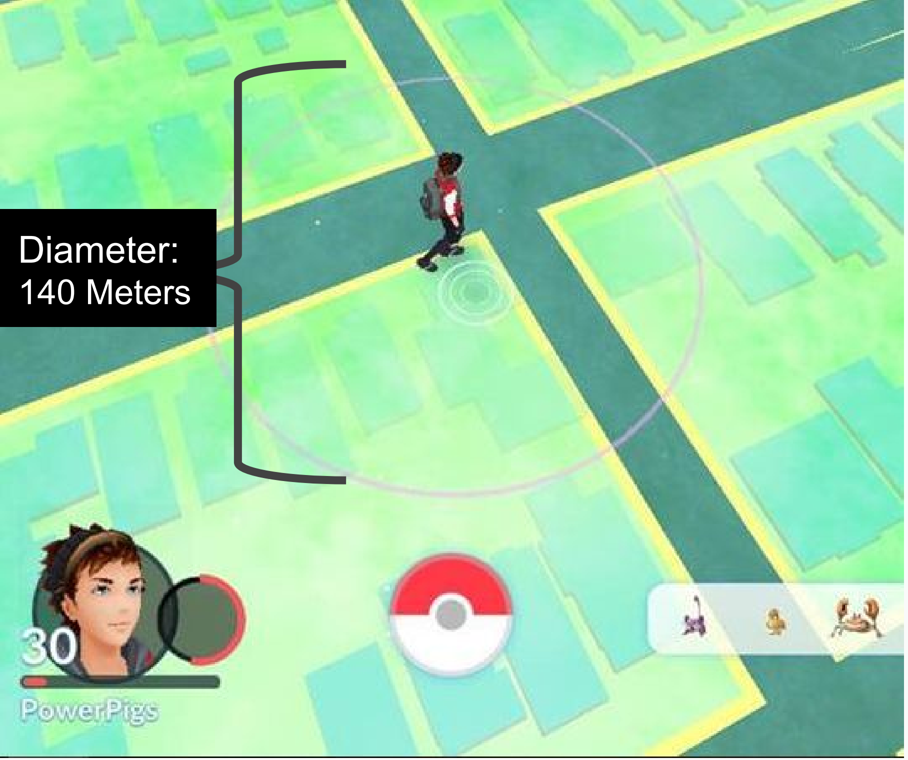
- What is Pokemon Go?
- Scanning Mechanics
- Map Mechanics
- Map Features
- Machine Learning Applications
- Training and Testing Data
- Logistic Regression Model
- Pidgey / Eevee Map
- Bulbasaur / Squirtle Map
- Conclusions

What is Pokemon Go and Why Does It Matter?

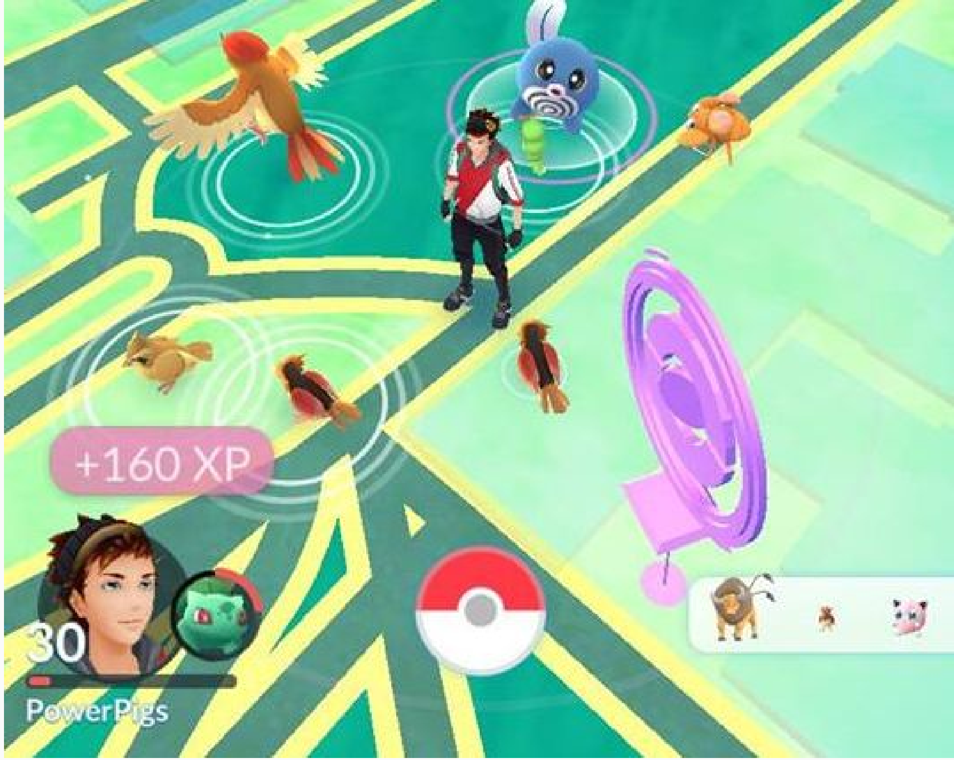
- Highest first month revenue for a mobile game, ever.
- Eclipsed $200 million dollar revenue after just ONE MONTH of release.
- Since release (July 6, 2016), has never dropped out of the top 25 apps in daily revenue for iOS OR Android.



Based on worldwide revenue estimates for iOS and Android (Google Play).
Revenue from soft launch periods not included.



Scans 70 meters in all direction from your current GPS location every 10 seconds.

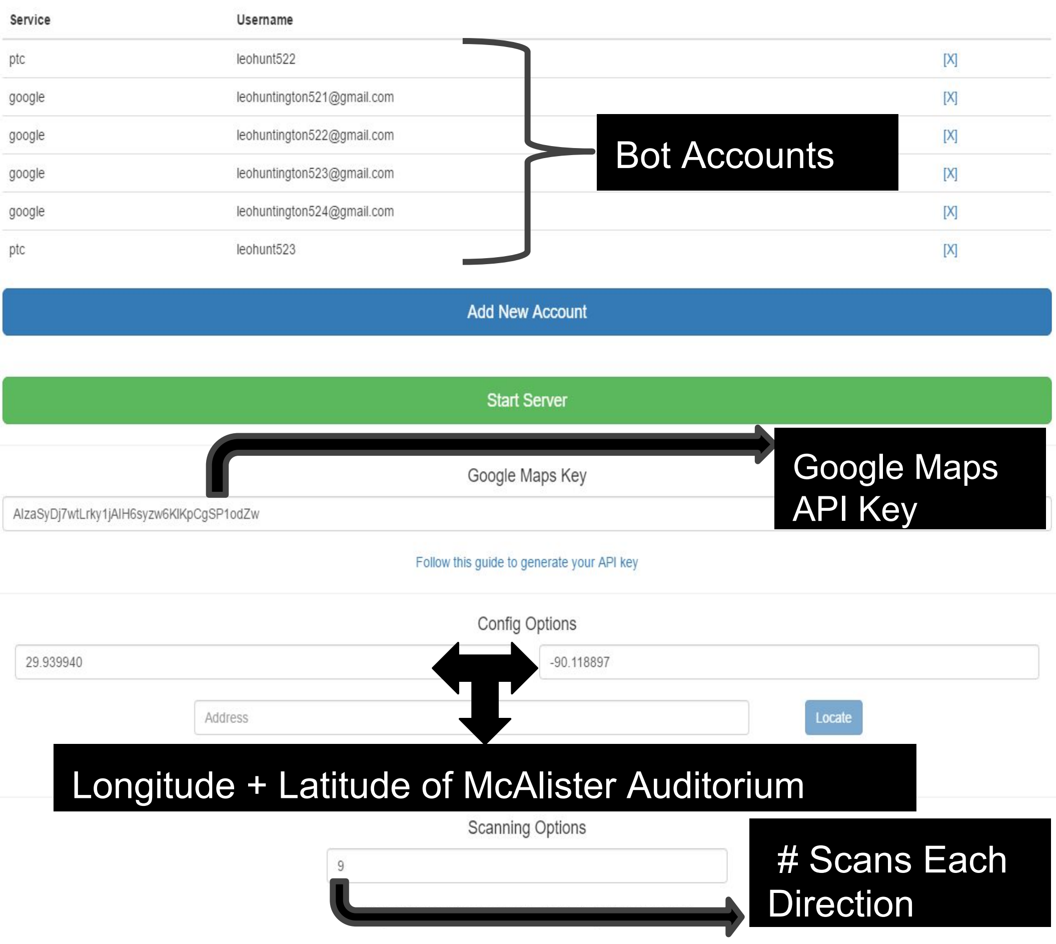


When nearby Pokemon are detected, they show up and may be “tapped on” to engage in the Pokeball fight.

Pokemon Go Scan Mechanics

Setting Up The Map

Source: https://github.com/mchristopher/PokemonGo-DesktopMap



Service	Username	
ptc	leohunt522	[X]
google	leohuntington521@gmail.com	[X]
google	leohuntington522@gmail.com	[X]
google	leohuntington523@gmail.com	[X]
google	leohuntington524@gmail.com	[X]
ptc	leohunt523	[X]

Bot Accounts

Google Maps Key

AIzaSyDj7wtLrky1jAIH6syzw6KlKpCgSP1odZw

Google Maps API Key

Follow this guide to generate your API key

Config Options

29.939940

-90.118897

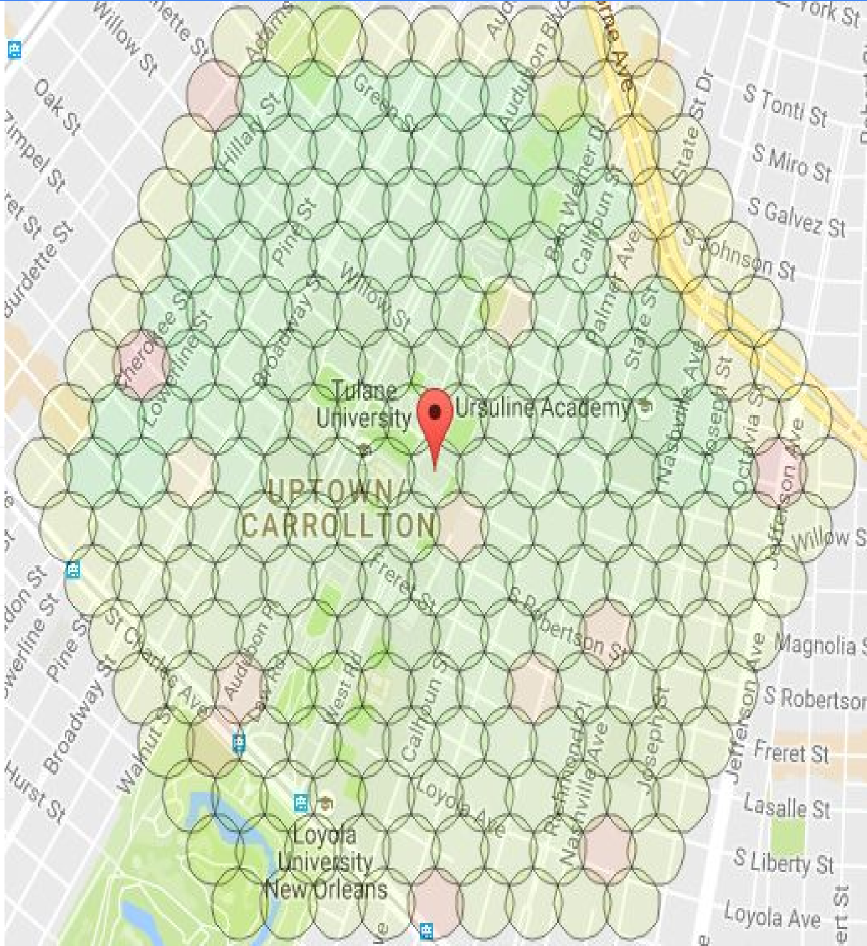
Address

Longitude + Latitude of McAlister Auditorium

Scanning Options

9

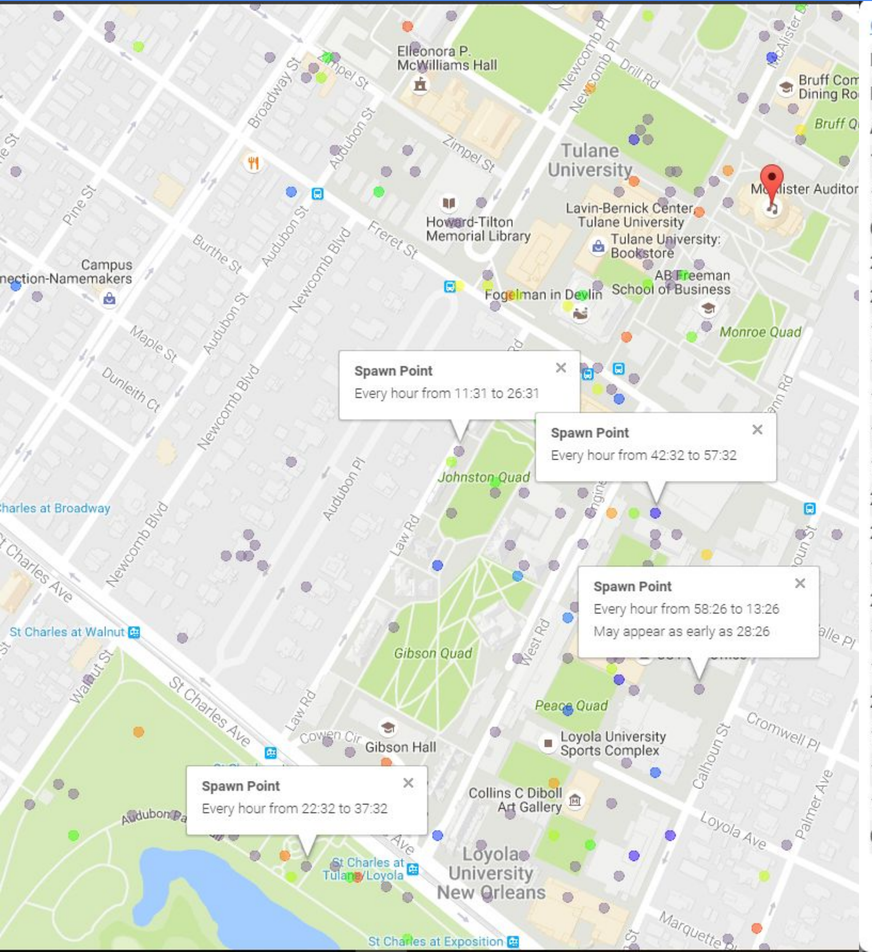
# Scans Each Direction



1 Week or 168 Hours of Scanning Later, Two Main Features;

1. Spawn Points

2. Nests

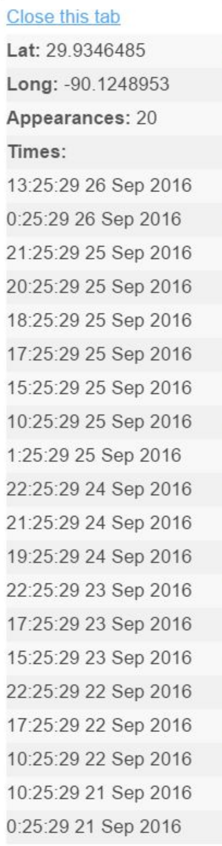


Spawn Point
Every hour from 11:31 to 26:31

Spawn Point
Every hour from 42:32 to 57:32

Spawn Point
Every hour from 58:26 to 13:26
May appear as early as 28:26

Spawn Point
Every hour from 22:32 to 37:32



Close this tab

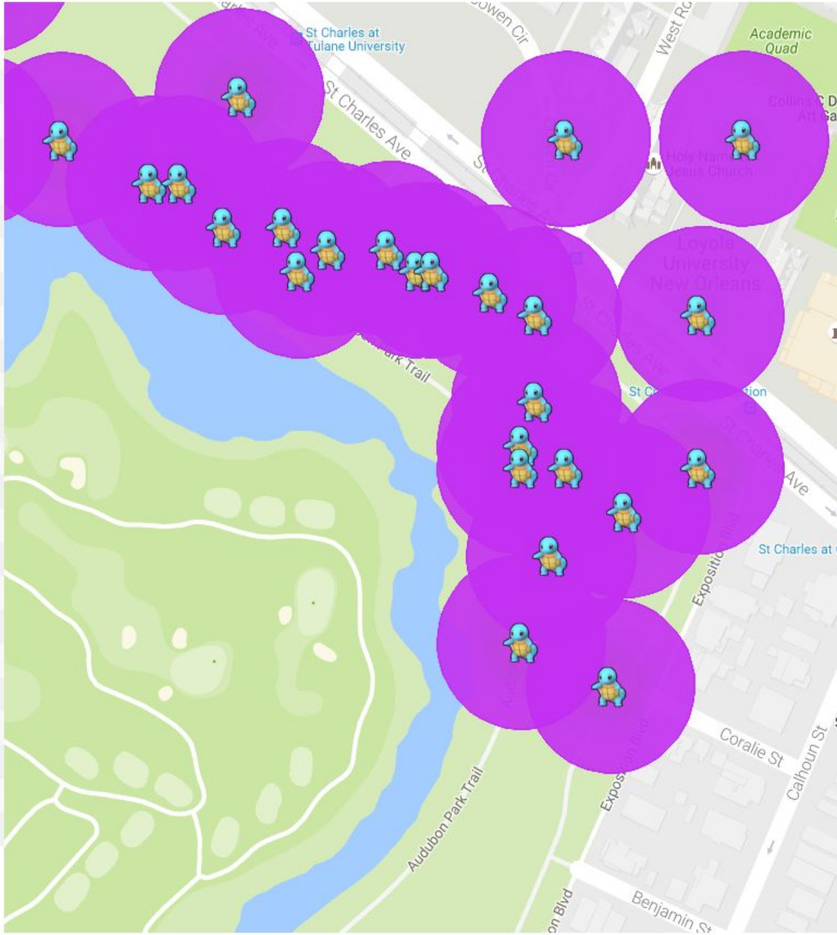
Lat: 29.9346485

Long: -90.1248953

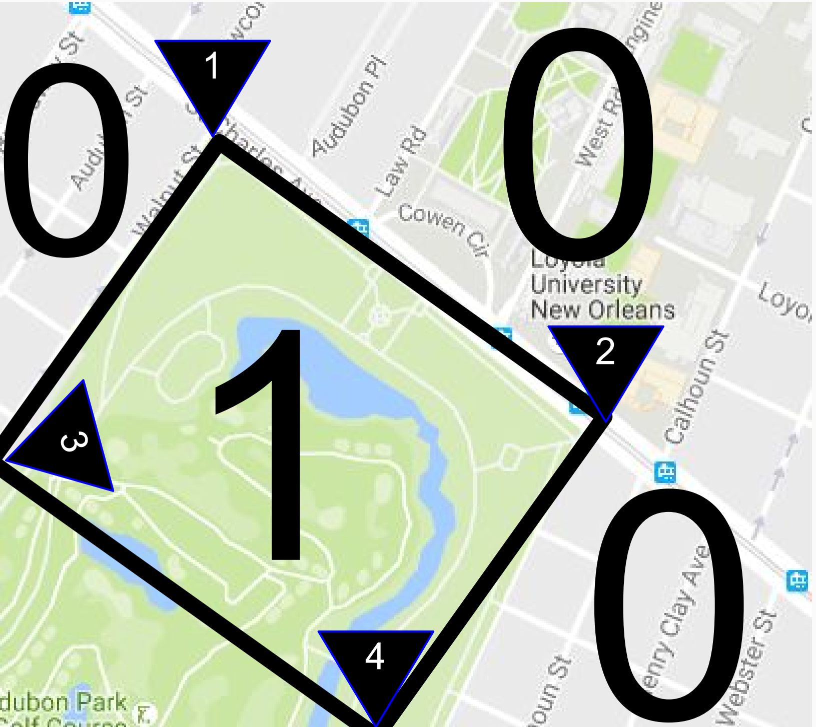
Appearances: 20

Times:

13:25:29 26 Sep 2016
0:25:29 26 Sep 2016
21:25:29 25 Sep 2016
20:25:29 25 Sep 2016
18:25:29 25 Sep 2016
17:25:29 25 Sep 2016
15:25:29 25 Sep 2016
10:25:29 25 Sep 2016
1:25:29 25 Sep 2016
22:25:29 24 Sep 2016
21:25:29 24 Sep 2016
19:25:29 24 Sep 2016
22:25:29 23 Sep 2016
17:25:29 23 Sep 2016
15:25:29 23 Sep 2016
22:25:29 22 Sep 2016
17:25:29 22 Sep 2016
10:25:29 22 Sep 2016
10:25:29 21 Sep 2016
0:25:29 21 Sep 2016



Dependent Variable (DV) = 1 IF Inside Audubon
= 0 IF Outside Audubon



Defining "Audubon Park" Longitude/Latitude

1 - Latitude: 29.936078, Longitude: -90.125425

2 - Latitude: 29.933144, Longitude: -90.121228

3 - Latitude: 29.932750, Longitude: -90.128169

4 - Latitude: 29.30600, Longitude: -90.122590

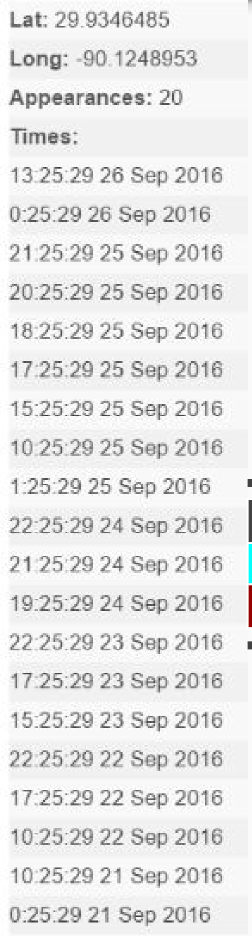
ROUGHLY

If(Latitude Input) between {29.30600, 29.936078}

AND(Longitude Input) between {-90.121228, -90.127169}

DV = 1 Else DV=0

Training and Testing Data



Lat: 29.9346485

Long: -90.1248953

Appearances: 20

Times:

13:25:29 26 Sep 2016
0:25:29 26 Sep 2016
21:25:29 25 Sep 2016
20:25:29 25 Sep 2016
18:25:29 25 Sep 2016
17:25:29 25 Sep 2016
15:25:29 25 Sep 2016
10:25:29 25 Sep 2016
1:25:29 25 Sep 2016
22:25:29 24 Sep 2016
21:25:29 24 Sep 2016
19:25:29 24 Sep 2016
22:25:29 23 Sep 2016
17:25:29 23 Sep 2016
15:25:29 23 Sep 2016
22:25:29 22 Sep 2016
17:25:29 22 Sep 2016
10:25:29 22 Sep 2016
10:25:29 21 Sep 2016
0:25:29 21 Sep 2016

3
2
1

Spawn Point Audubon 1 or “A1” Squirtle spawns for our test data.

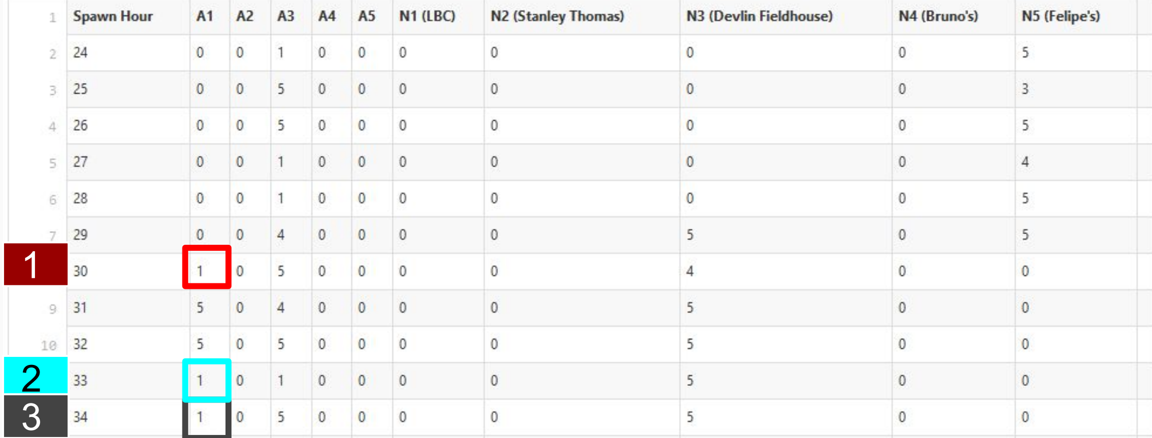
Testing Data (Saturday)

Training Data (Thursday)

Sample data for this project was collected from 12:00AM Wednesday September 21st until 11:59PM September 27th. For this project, we noted data from 10 spawn points, 5 from within Audubon Park, and 5 outside of Audubon Park at popular destinations. Spawn points at the LBC, Stanley Thomas, Devlin Fieldhouse, Bruno’s, and Felipe’s were chosen as our “non-audubon” spawn points due to their high traffic and probably density comparable to Audubon Park.

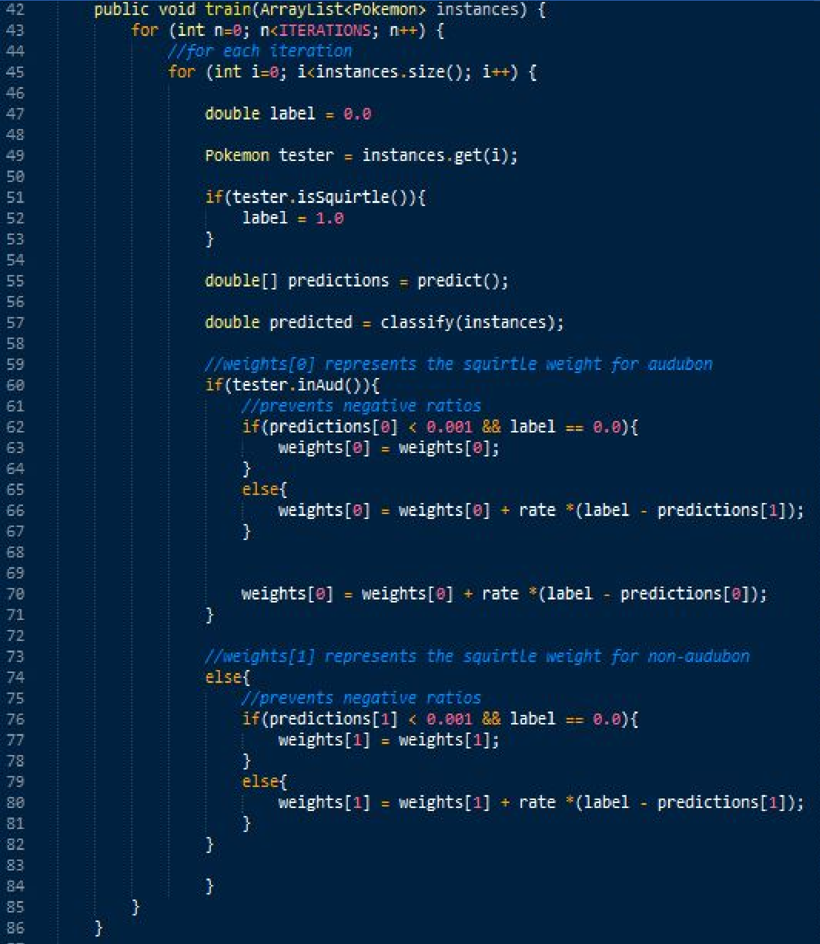
Spawn point data was represented by an integer value with their codes listed below. Each day of the week was sectioned into 24 hours and spawns recorded for all 10 spawn points over that day.

PokeInputSaturday.csv
0=No Spawn, 1=Squirtle, 2=Bulbasaur, 3=Eevee, 4=Pidgey, 5=Other

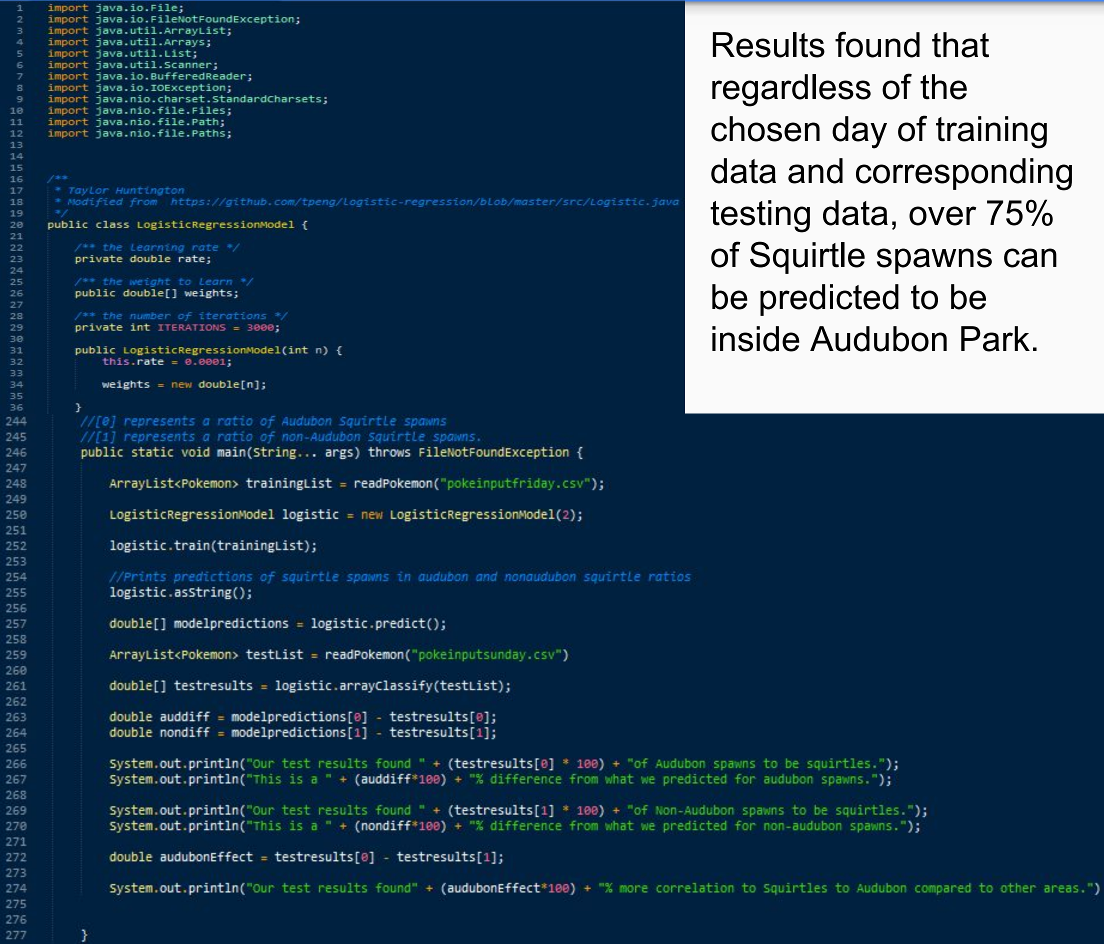


1	Spawn Hour	A1	A2	A3	A4	A5	N1 (LBC)	N2 (Stanley Thomas)	N3 (Devlin Fieldhouse)	N4 (Bruno's)	N5 (Felipe's)
2	24	0	0	1	0	0	0	0	0	0	5
3	25	0	0	5	0	0	0	0	0	0	3
4	26	0	0	5	0	0	0	0	0	0	5
5	27	0	0	1	0	0	0	0	0	0	4
6	28	0	0	1	0	0	0	0	0	0	5
7	29	0	0	4	0	0	0	0	5	0	5
1	30	1	0	5	0	0	0	0	4	0	0
9	31	5	0	4	0	0	0	0	5	0	0
10	32	5	0	5	0	0	0	0	5	0	0
2	33	1	0	1	0	0	0	0	5	0	0
3	34	1	0	5	0	0	0	0	5	0	0

Logistical Regression Model (Learning Algorithm) FEATURES: Species, spawn hour, inAud DEPENDENT VARIABLE (DV) is location (in or OUT of audubon as binary)

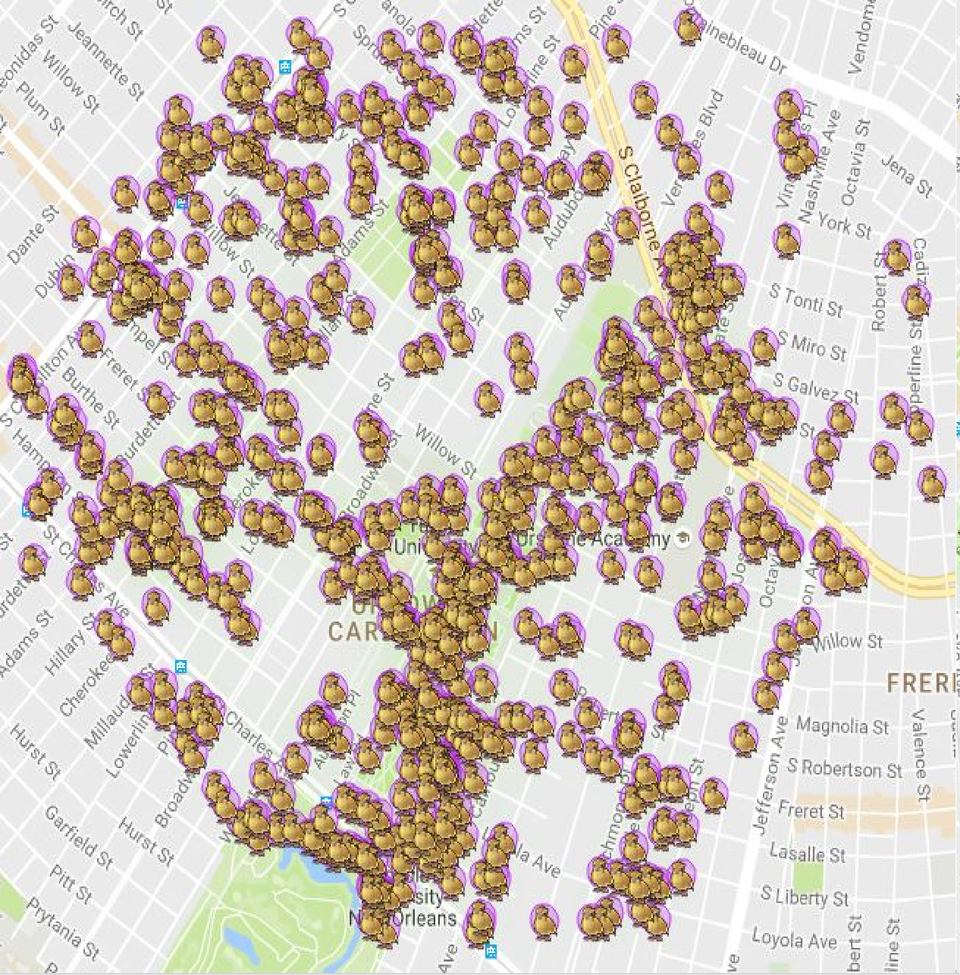


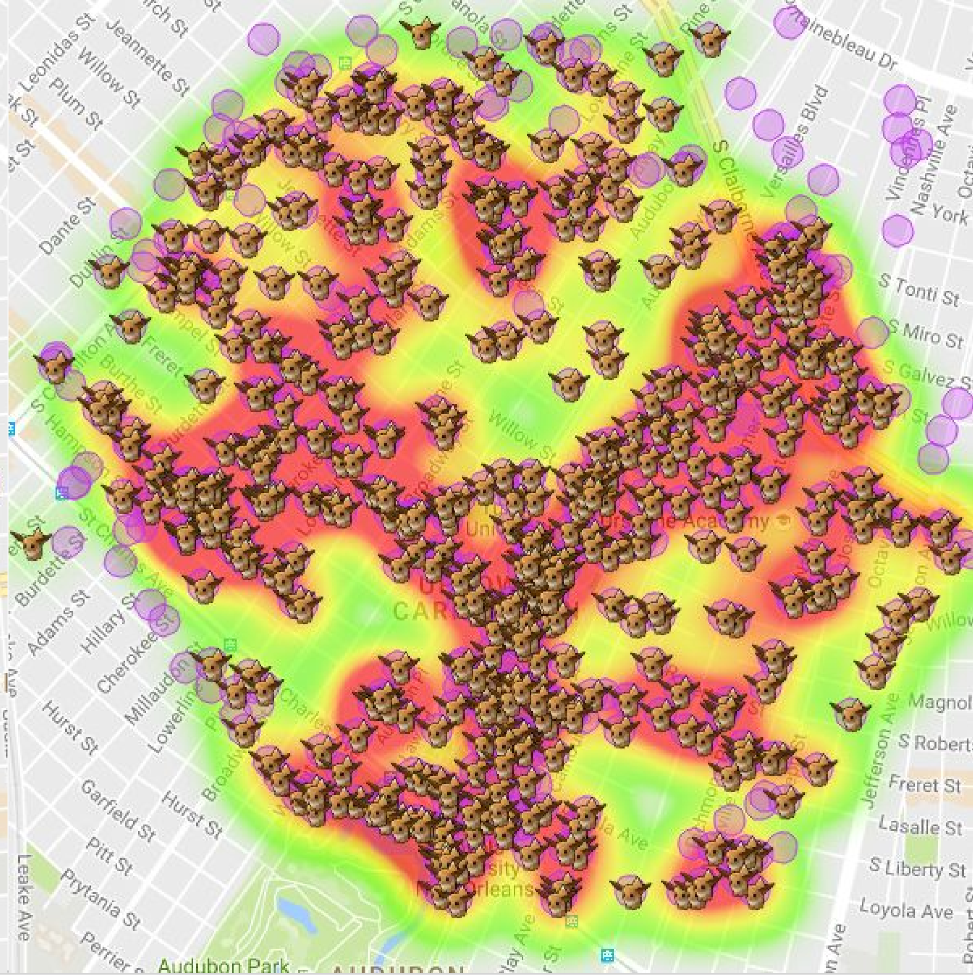
```java
public void train(ArrayList<Pokemon> instances) {
    for (int n=0; n<ITERATIONS; n++) {
        //for each iteration
        for (int i=0; i<instances.size(); i++) {

            double label = 0.0;

            Pokemon tester = instances.get(i);

            if(tester.isSquirtle()){
                label = 1.0;
            }

            double[] predictions = predict();

            double predicted = classify(instances);

            //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]);
                }
            }

        }
    }
}
```

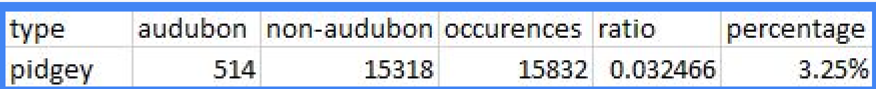


```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;



/**
 * Taylor Huntington
 * Modified from  https://github.com/tpeng/logistic-regression/blob/master/src/Logistic.java
 */
public class LogisticRegressionModel {

    /** the learning rate */
    private double rate;

    /** the weight to learn */
    public double[] weights;

    /** the number of iterations */
    private int ITERATIONS = 3000;

    public LogisticRegressionModel(int n) {
        this.rate = 0.0001;

        weights = new double[n];

    }
     //[0] represents a ratio of Audubon Squirtle spawns
     //[1] represents a ratio of non-Audubon Squirtle spawns.
     public static void main(String... args) throws FileNotFoundException {

         ArrayList<Pokemon> trainingList = readPokemon("pokeinputfriday.csv");

         LogisticRegressionModel logistic = new LogisticRegressionModel(2);

         logistic.train(trainingList);

         //Prints predictions of squirtle spawns in audubon and nonaudubon squirtle ratios
         logistic.asString();

         double[] modelpredictions = logistic.predict();

         ArrayList<Pokemon> testList = readPokemon("pokeinputsunday.csv")

         double[] testresults = logistic.arrayClassify(testList);

         double auddiff = modelpredictions[0] - testresults[0];
         double nondiff = modelpredictions[1] - testresults[1];

         System.out.println("Our test results found " + (testresults[0] * 100) + "of Audubon spawns to be squirtles.");
         System.out.println("This is a " + (auddiff*100) + "% difference from what we predicted for audubon spawns.");

         System.out.println("Our test results found " + (testresults[1] * 100) + "of Non-Audubon spawns to be squirtles.");
         System.out.println("This is a " + (nondiff*100) + "% difference from what we predicted for non-audubon spawns.");

         double audubonEffect = testresults[0] - testresults[1];

         System.out.println("Our test results found" + (audubonEffect*100) + "% more correlation to Squirtles to Audubon compared to other areas.")

     }
```

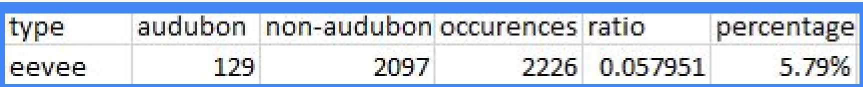
Results found that regardless of the chosen day of training data and corresponding testing data, over 75% of Squirtle spawns can be predicted to be inside Audubon Park.



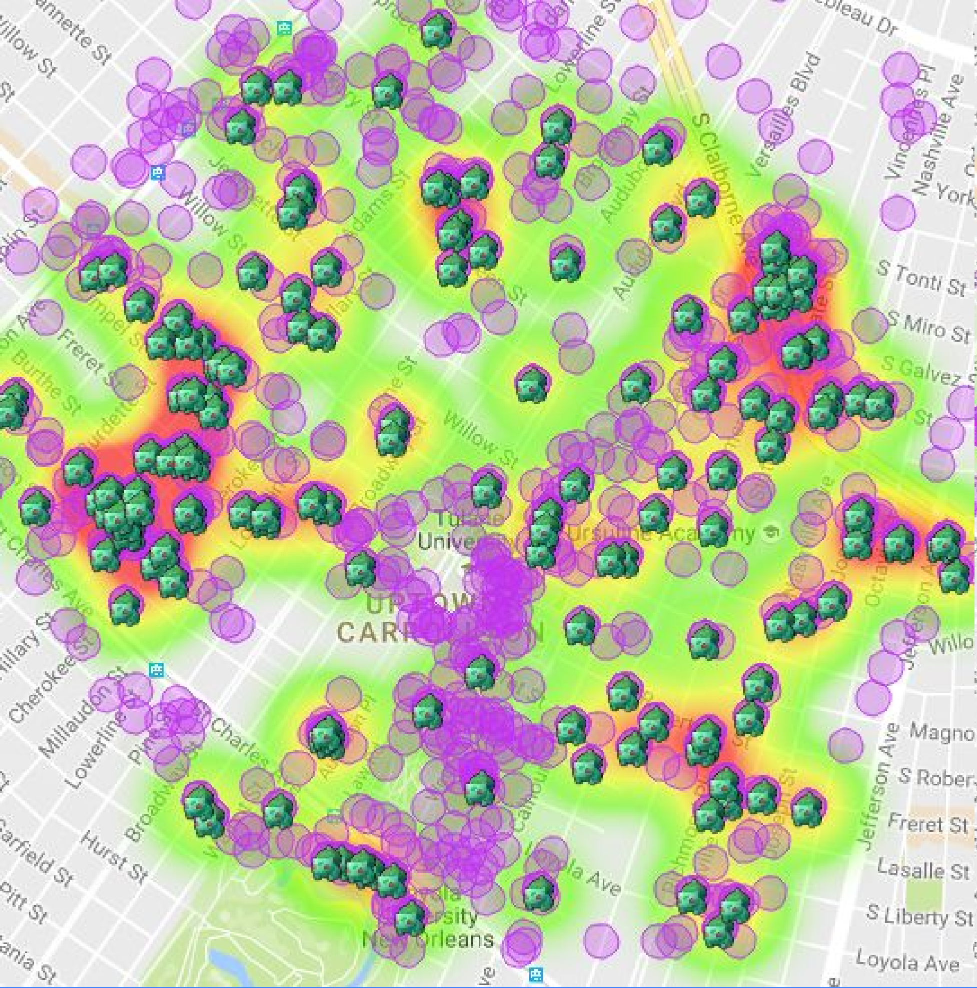


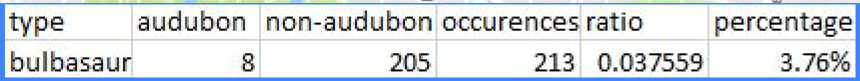


type	audubon	non-audubon	occurences	ratio	percentage
pidgey	514	15318	15832	0.032466	3.25%

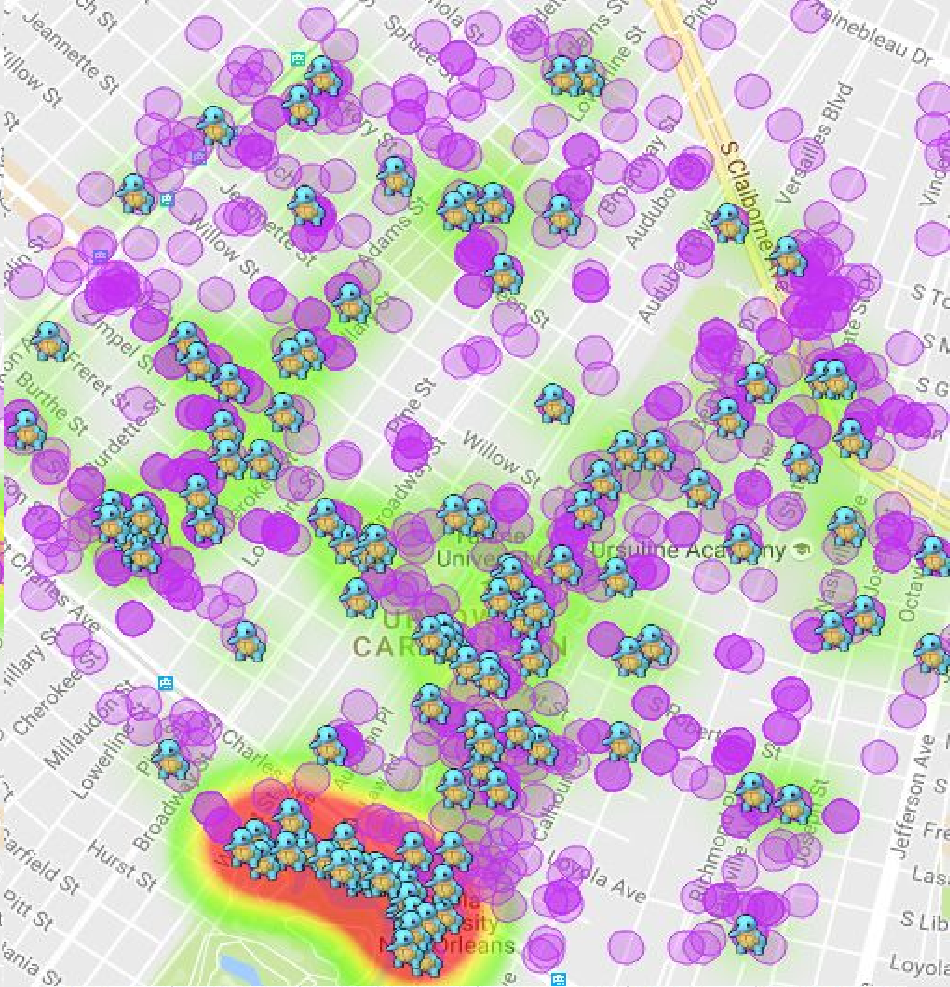


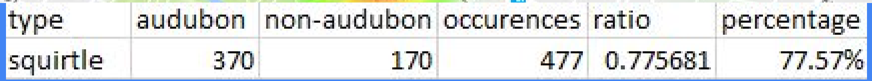
type	audubon	non-audubon	occurences	ratio	percentage
eevee	129	2097	2226	0.057951	5.79%





type	audubon	non-audubon	occurences	ratio	percentage
bulbasaur	8	205	213	0.037559	3.76%

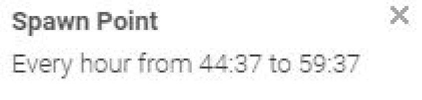




type	audubon	non-audubon	occurences	ratio	percentage
squirtle	370	170	477	0.775681	77.57%

Results/Conclusions

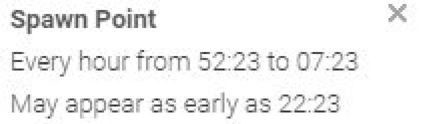
Example A



Spawn Point

Every hour from 44:37 to 59:37

Example B



Spawn Point

Every hour from 52:23 to 07:23

May appear as early as 22:23

1. Pokemon spawn via an hourly timer at specific locations known as “spawn points”.
   a. All spawn points operate on a 60 hour timer.
   b. Each spawn point will spawn a creature at a specific minute combination every hour for fifteen hours straight when it “starts”.
   c. Spawn points can be “confident” (example A) or “random” (example B).
   d. Certain spawn points have different tendencies to spawn different creatures.
   e. Spawn points in an area tend to spawn similar species.
2. Created Logistic Regression Model predicts that AT LEAST 75% of Squirtles spawns to occur within Audubon Park, regardless of chosen days of test and training data.
   a. Low: .752 or 75.2%   b. High: .889 or 88.9%
3. Confirms Audubon Park being a Squirtle Nest.
   a. 77.57% (370 / 477) of all Squirtles spawns occurred inside the “Audubon Park” defined area.
   b. Of 61,246 NON-SQUIRTLE spawns, no species exceeded 10% of their occurrences in the Audubon Park defined area.