Data Analysis on Pokemon GO

[1]Atul Soni, [2]Smit Shah, [3]Monik Gallani, [4]Prof. Suraj Patil

[1]Student, [2]Student, [3]Student, [4]Assistant Professor
[1]Department of Computer Engineering,
[1]MPSTME NMIMS, Shirpur Campus, India

Abstract : Pokemon Go is concept game based on Augmented Reality. For this study, a dataset having around 350,000 spawning entries was used. Each entry contained the name of the pokemon, the appearing and disappearing time of that pokemon, as well as the longitudinal and latitudinal location it spawned at. The goal of the study was to verify the prediction of the pokemon at the spawned locations and to explore the use of random forest regression model in cases having location data. The result thus proved that random forest regression performs excellently while dealing with longitudinal and latitudinal data.

***Index Terms* – Pokemon, Dataset, Classification, GridSearchCV, Machine learning, training speed, prediction**

I. Introduction

The game-play is based over a dynamic map which changes according to the user's movements in the real world surroundings. Different pokemons are present in different locations. If a pokemon is residing at the location where the user is cuurently present, an image gets displayed on the user's mobile screen as if that pokemon is in the real world. When such an encounter occurs, the user can capture that pokemon by flicking the pokeball present at the bottom of the user's screen towards that pokemon. After capturing the pokemon, we can then use that pokemon for future battles in the game. etc. The goal of our study is to verify the feasibility of the prediction of pokemon. The results of this study can be usefull not only for Pokemon Go, but also in other kinds of datasets having longitudinal and latitudinal data.

II. Related work

k-nearest neighbors is broadly used to arrange information might be in neural systems as a field or whether it’s any utilization of Biometrics, Handwriting classification or Iris detection, attainably the max. open classifier in the reserve or machine learning methods is the Nearest Neighbour Classifier in which characterization is achieved by recognizing the closest neighbours to an inquiry illustration and utilizing those neighbours to decide the class of the question KNN grouping orders cases in light of their similarity.to occurrences in the preparation information. This paper presents different yield with different separation utilized as a part of calculation and may know the reaction of classifier for the coveted application. it likewise speaks to computational problems in distinguishing closest neighbours and components for decreasing the measurement of the information. Aman Kataria , M. D. Singh[1].

The Classification method gauge the absolute and forecast models to foresee nonstop esteemed capacities. For the most part, grouping is the way toward arranging information into classifications for its most powerful and able utilize. The information grouping technique makes fundamental information that is anything but difficult to discover and recover. In this paper the execution of three Bayes classifiers calculations in particular Naive Bayes, Bayes Net also, Naive Bayes Multinomialare broke down. The coronary illness dataset is utilized for evaluating the execution of the calculations by utilizing the cross approval parameter. Lastly the relative examination is performed by utilizing the components, for example, the arrangement exactness and blunder rates on all calculations. S. Karthika* and N. Sairam[2].

Decision tree systems have been broadly used to construct order models all things considered models nearly look like human thinking and are straightforward. This paper portrays fundamental decision tree issues and flow examine focuses. Obviously, a solitary article can’t be an entire survey of all calculations (likewise known enlistment arrangement trees), yet we trust that the references refered to will cover the major hypothetical issues, controlling the specialist in intriguing exploration bearings and proposing conceivable predict position mixes that presently can’t seem to be investigated. Park, Hyeoun-Ae[3].

III. COMPARITIVE STUDY

Table 3.1 Overview of Algorithms:-

Algorithm	Application	Interpretation
KNN	Both	Easy
Linear Regression	Regression	Easy
Logistic Regression	Classification	Moderate
Naive Bayes	Classification	Moderate
Decision Trees	Both	Moderate
Random Forests	Both	Modearte
AdaBoost	Both	Difficult
Neural Networks	Both	Not Applicable

The Table 3.1 displays the comparison between application of algorithms and Interpretation of the algorithms .here both refers to regression as well as well as classification both included.

Table 3.2 Comparison of algorithms based on speeds:-

Algorithm	Training Speed	Prediction Speed
KNN	Fast	Moderate
Linear Regression	Fast	Fast
Logistic Regression	Fast	Fast
Naive Bayes	Fast	Fast
Decision Trees	Fast	Fast
Random Forests	Moderate	Moderate
AdaBoost	Slow	Fast
Neural Networks	Slow	Fast

The Table 3.2 displays comparison of the Training speed and prediction speed of the different algorithm, for KNN training speed is fast and prediction speed is moderate, for Linear regression training speed is fast and prediction speed is fast for Logistic regression speed is fast and prediction speed is fast for Naïve Bayes training speed is fast and prediction speed is fast, for Decision trees speed is fast and prediction speed is fast, for Random forest training speed is moderate and prediction speed is moderate, for AdaBoost training speed is slow and prediction speed is moderate, for Neural training speed is moderate and prediction speed is Fast.

Table 3.3: Comparison of Algorithms based on their handling of data

Algorithm	Prediction Accuracy	Separates Noise
KNN	Lower	No
Linear Regression	Lower	No
Logistic Regression	Lower	No
Naive Bayes	Lower	Yes
Decision Trees	Lower	No
Random Forests	Higher	Yes(Low Noise ratio)
AdaBoost	Higher	Yes
Neural Networks	Higher	Yes

The table 3.3 displays the comparison between accuracy of algorithms and separates noise that is noise ratio of the algorithms, for KNN prediction accuracy is lower and does not separate noise, for Linear regression prediction accuracy is lower and does not separate noise, for Logistic regression prediction accuracy is lower and does not separate noise for Naïve Bayes prediction accuracy is lower and it separate noises to an extent, for Decision trees prediction accuracy is lower and does not separate noise, for Random forest prediction accuracy is higher and low noise ratio, for AdaBoost prediction accuracy is higher and it separate noise, for Neural training prediction accuracy is higher and does not separate noise.

IV. MODEL

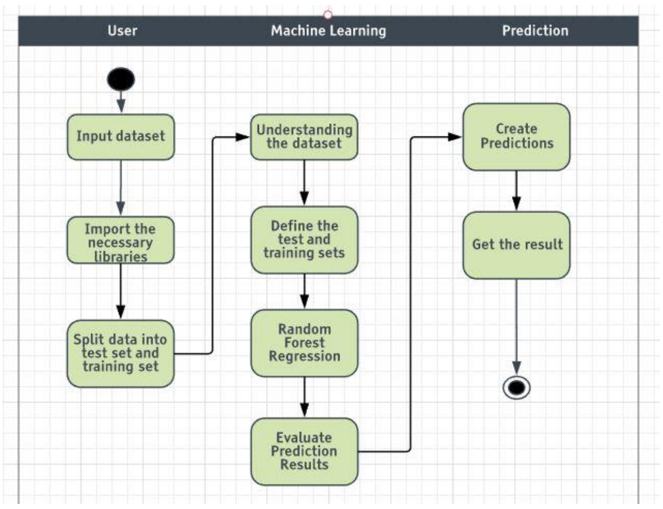

User
Machine Learning
Prediction
Input dataset
Import the necessary libraries
Split data into test set and training set
Understanding the dataset
Define the test and training sets
Random Forest Regression
Evaluate Prediction Results
Create Predictions
Get the result

Figure 1: Architecture

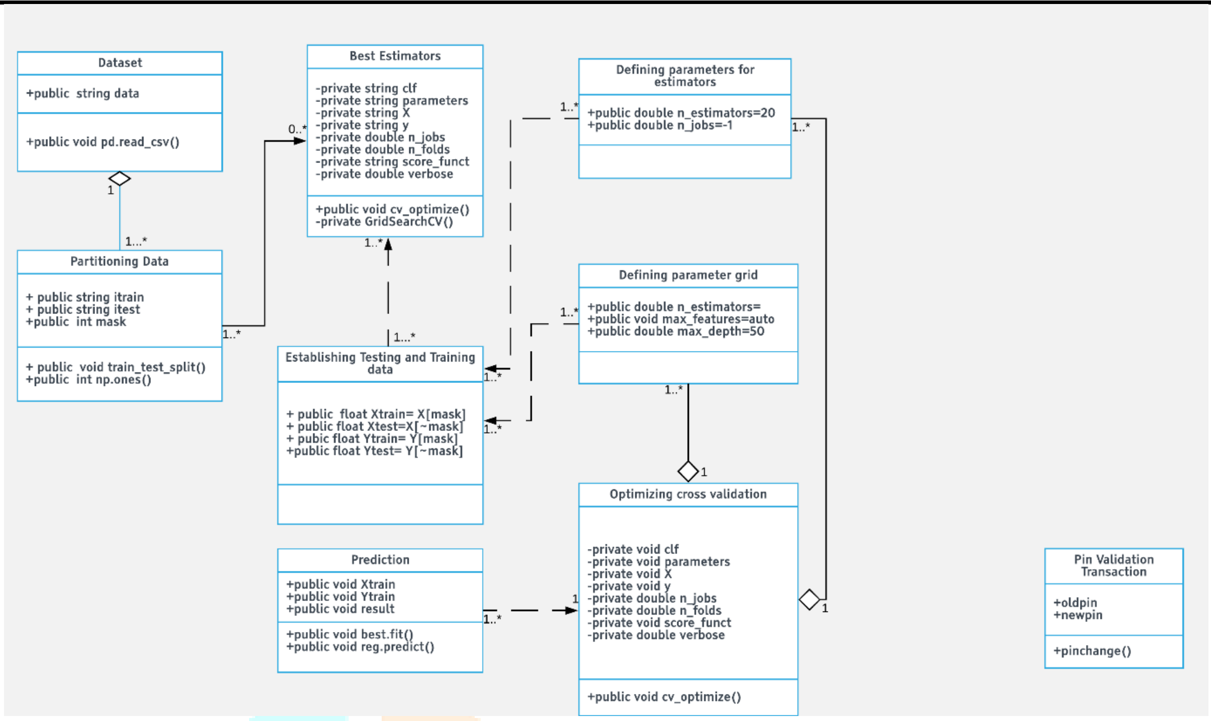


Figure 2: Class Diagram of the System

4.1 Dataset

For this study secondary data has been collected. The dataset consists of 8 attributes s2_id id, token, num(pokemon number), name(pokemon name), latitude, longitude, encounter time (appearing time), disappear time (disappearing time).but id and token are numeric variables which cannot help in prediction model so we use the 6 attributes other than id and token.

4.2 Theoretical framework

Variables of the study contains dependent and independent variable. The study used pre-specified method for the selection of variables. The study used the pokemon number are as dependent variable. These variables are used for splitting into training set and test set, which used in the process of defining the two sets into the random forest

An Random forest is an outfit (i.e., a gathering) of unpruned Decision trees. The Random forests are regularly utilized when we have extensive preparing datasets and countless factors (hundreds or even a large number of input factors). An Random forest demonstrate is a classifier that comprises of many decision trees and yields the class that is the method of the class yield by singular trees

4.3 Equations

The Random forest is a predictor based of out a group of randomized base regression trees $\{r_n(\mathbf{x}, \Theta_m, \mathcal{D}_n), m \geq 1\}$, where $\Theta_1, \Theta_2,$ .... These are the output of the randomized variable. These are together taken for aggregated random estimate by following

$$\bar{r}_n(\mathbf{X}, \mathcal{D}_n) = \mathbb{E}_\Theta \left[r_n(\mathbf{X}, \Theta, \mathcal{D}_n) \right], \quad \ldots\ldots\ldots\ldots(1)$$

Where E denotes expectation conditionally on X w.r.t random parameter

$$r_n(\mathbf{X}, \Theta) = \frac{\sum_{i=1}^{n} Y_i \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]}}{\sum_{i=1}^{n} \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X}, \Theta)]}} \mathbf{1}_{\mathcal{E}_n(\mathbf{X}, \Theta)}, \quad \ldots\ldots\ldots..(2)$$

The randomizing variable Θ is used to determine the way consecutive cuts are performed when making the individual trees, similar to the selection of the coordinate to split & position of the split.

where $\mathbb{E}_\Theta$ denotes expectation with respect to the random parameter, conditionally on X and the data set Dn. In the following, to lighten notation a little, we will omit the dependency of the estimates in the sample, and write for example ¯rn(X) instead of ¯rn(X,Dn).

Here event is

$$\mathcal{E}_n(\mathbf{X},\Theta) = \left[\sum_{i=1}^{n} \mathbf{1}_{[\mathbf{X}_i \in A_n(\mathbf{X},\Theta)]} \neq 0\right]. \qquad \text{............(3)}$$

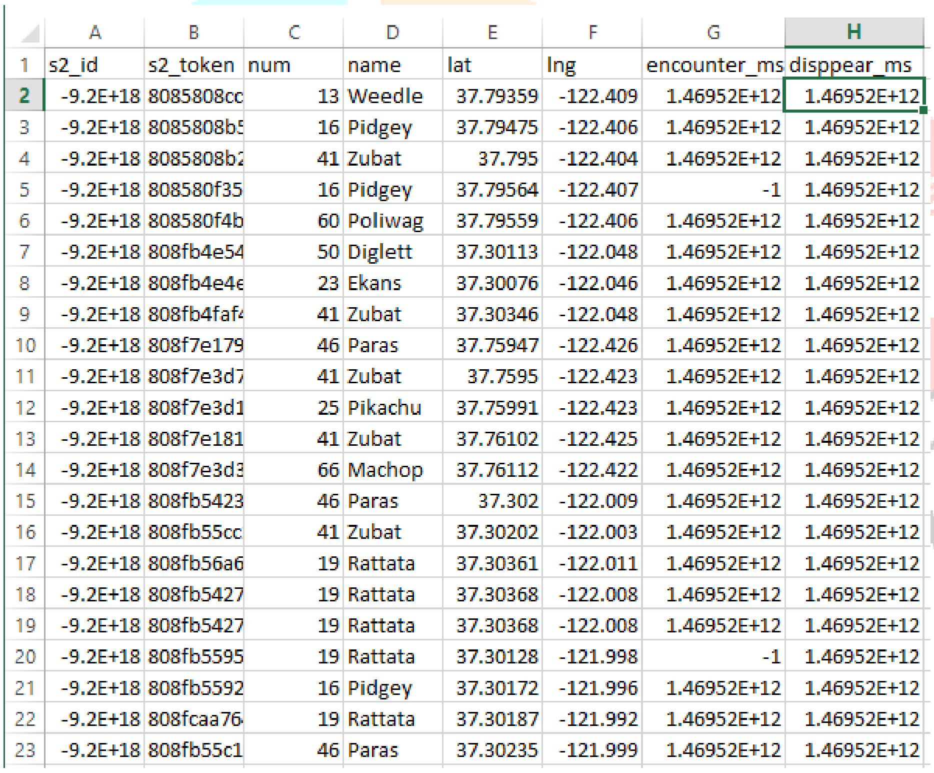
4.4 Features of Random Forest

It is unexcelled in precision among current calculations. It runs productively on substantial information bases. It can deal with a large number of info factors without variable erasure. It gives evaluations of what factors are essential in the classification. It creates an interior impartial gauge of the generalization mistake as the backwoods building advances. It has a powerful strategy for evaluating missing information and keeps up exactness when a huge extent of the information are absent. It has techniques for adjusting blunder in class populace lopsided informational indexes. Produced woods can be put something aside for later use on other information. Prototypes are calculated that give data about the relation between the factors and the classification. It offers an exploratory technique for recognizing variable connections.

As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model. GridSearchCV() is an extension of Random Search Method Instead of sampling randomly from a distribution, it evaluates all the combinations we define. To use Grid Search, we make a grid based on what we think are the best settings for searching. The model then tries out all the combinations of the specified settings. Ultimately, we then fit the model and implement our prediction function

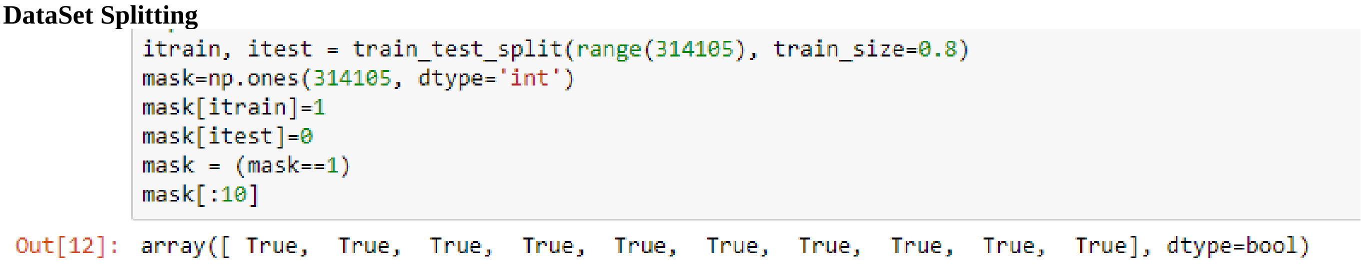
V. Implementation

Data Set



	A	B	C	D	E	F	G	H
1	s2_id	s2_token	num	name	lat	lng	encounter_ms	disppear_ms
2	-9.2E+18	8085808cc	13	Weedle	37.79359	-122.409	1.46952E+12	1.46952E+12
3	-9.2E+18	8085808b5	16	Pidgey	37.79475	-122.406	1.46952E+12	1.46952E+12
4	-9.2E+18	8085808b2	41	Zubat	37.795	-122.404	1.46952E+12	1.46952E+12
5	-9.2E+18	808580f35	16	Pidgey	37.79564	-122.407	-1	1.46952E+12
6	-9.2E+18	808580f4b	60	Poliwag	37.79559	-122.406	1.46952E+12	1.46952E+12
7	-9.2E+18	808fb4e54	50	Diglett	37.30113	-122.048	1.46952E+12	1.46952E+12
8	-9.2E+18	808fb4e4e	23	Ekans	37.30076	-122.046	1.46952E+12	1.46952E+12
9	-9.2E+18	808fb4faf4	41	Zubat	37.30346	-122.048	1.46952E+12	1.46952E+12
10	-9.2E+18	808f7e179	46	Paras	37.75947	-122.426	1.46952E+12	1.46952E+12
11	-9.2E+18	808f7e3d7	41	Zubat	37.7595	-122.423	1.46952E+12	1.46952E+12
12	-9.2E+18	808f7e3d1	25	Pikachu	37.75991	-122.423	1.46952E+12	1.46952E+12
13	-9.2E+18	808f7e181	41	Zubat	37.76102	-122.425	1.46952E+12	1.46952E+12
14	-9.2E+18	808f7e3d3	66	Machop	37.76112	-122.422	1.46952E+12	1.46952E+12
15	-9.2E+18	808fb5423	46	Paras	37.302	-122.009	1.46952E+12	1.46952E+12
16	-9.2E+18	808fb55cc	41	Zubat	37.30202	-122.003	1.46952E+12	1.46952E+12
17	-9.2E+18	808fb56a6	19	Rattata	37.30361	-122.011	1.46952E+12	1.46952E+12
18	-9.2E+18	808fb5427	19	Rattata	37.30368	-122.008	1.46952E+12	1.46952E+12
19	-9.2E+18	808fb5427	19	Rattata	37.30368	-122.008	1.46952E+12	1.46952E+12
20	-9.2E+18	808fb5595	19	Rattata	37.30128	-121.998	-1	1.46952E+12
21	-9.2E+18	808fb5592	16	Pidgey	37.30172	-121.996	1.46952E+12	1.46952E+12
22	-9.2E+18	808fcaa76	19	Rattata	37.30187	-121.992	1.46952E+12	1.46952E+12
23	-9.2E+18	808fb55c1	46	Paras	37.30235	-121.999	1.46952E+12	1.46952E+12

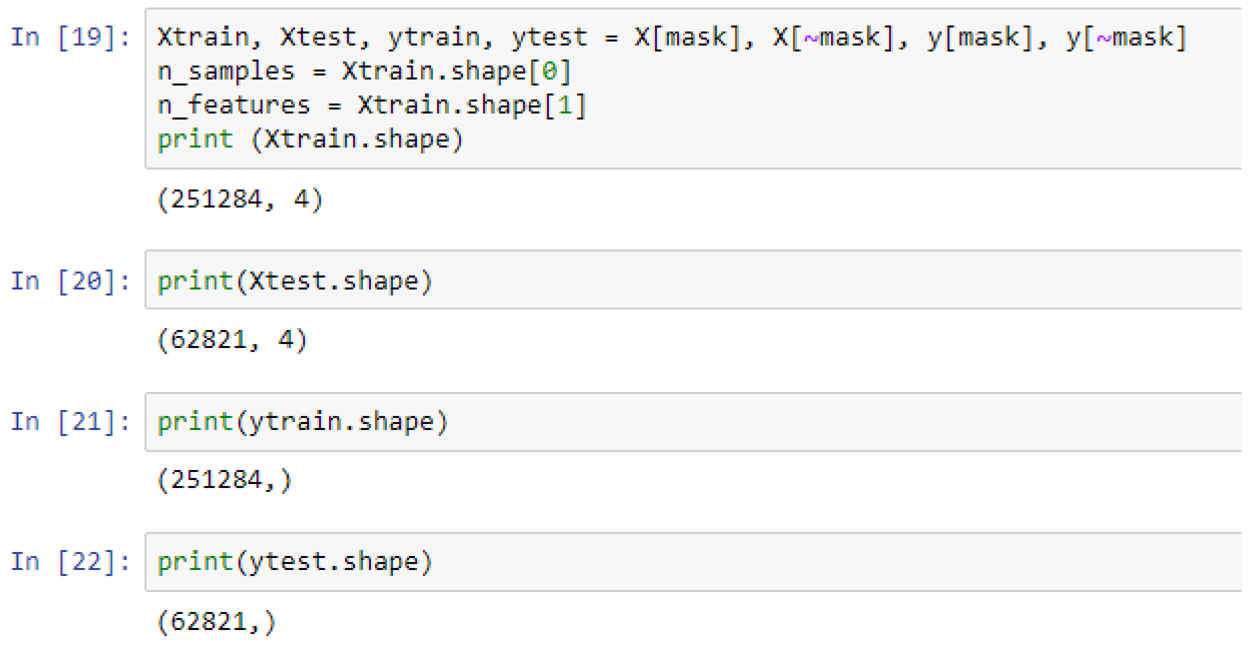
DataSet Splitting



```
itrain, itest = train_test_split(range(314105), train_size=0.8)
mask=np.ones(314105, dtype='int')
mask[itrain]=1
mask[itest]=0
mask = (mask==1)
mask[:10]
```

```
Out[12]: array([ True,  True,  True,  True,  True,  True,  True,  True,  True,  True], dtype=bool)
```

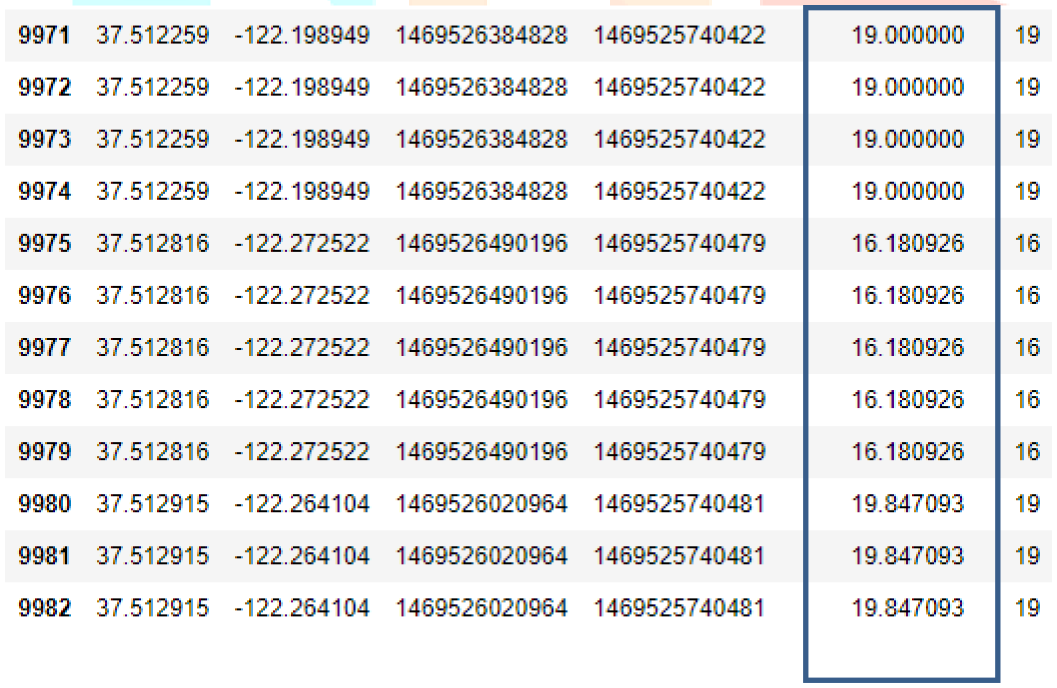
Refining the Training and Testing datasets



```
In [19]: Xtrain, Xtest, ytrain, ytest = X[mask], X[~mask], y[mask], y[~mask]
         n_samples = Xtrain.shape[0]
         n_features = Xtrain.shape[1]
         print (Xtrain.shape)

         (251284, 4)

In [20]: print(Xtest.shape)

         (62821, 4)

In [21]: print(ytrain.shape)

         (251284,)

In [22]: print(ytest.shape)

         (62821,)
```

VI. RESULTS

6.1 Predicted Pokemon

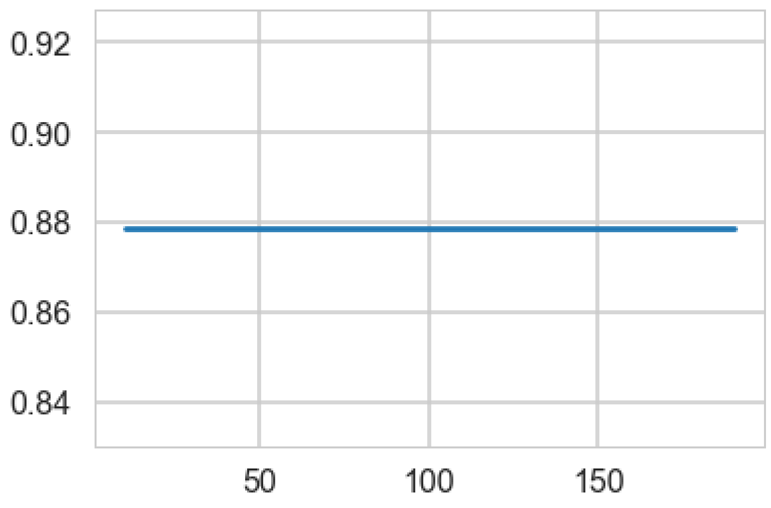


9971	37.512259	-122.198949	1469526384828	1469525740422	19.000000	19
9972	37.512259	-122.198949	1469526384828	1469525740422	19.000000	19
9973	37.512259	-122.198949	1469526384828	1469525740422	19.000000	19
9974	37.512259	-122.198949	1469526384828	1469525740422	19.000000	19
9975	37.512816	-122.272522	1469526490196	1469525740479	16.180926	16
9976	37.512816	-122.272522	1469526490196	1469525740479	16.180926	16
9977	37.512816	-122.272522	1469526490196	1469525740479	16.180926	16
9978	37.512816	-122.272522	1469526490196	1469525740479	16.180926	16
9979	37.512816	-122.272522	1469526490196	1469525740479	16.180926	16
9980	37.512915	-122.264104	1469526020964	1469525740481	19.847093	19
9981	37.512915	-122.264104	1469526020964	1469525740481	19.847093	19
9982	37.512915	-122.264104	1469526020964	1469525740481	19.847093	19

Here we get the predicted pokemon number and we compare it to the original number to find out its accuracy

6.2 Accuracy

```
In [51]: estimators = np.arange(10, 200, 10)
         scores = []
         for n in estimators:
             scores.append(reg.score(Xtest, ytest))
         plt.plot(estimators, scores)

Out[51]: [<matplotlib.lines.Line2D at 0x1ee2311d160>]
```



We find out that the prediction model has 87.89% accuracy which is approximately 88% the figure shows the validation of the same and we see that random forest has higher prediction accuracy.

VII. Conclusion

This paper documents the methodology which can be used to compare different data mining techniques for our work. We successfully implemented prediction model based on Random Forest Regression. We validated the presence of the same Pokemon at the spawned location We provided the solution for the problems we listed from pokemon go such as physical hazards due to the gameplay. Our model can be refined and integrated into a software, giving users future spawning locations. Our model can also be a guiding point for other models that work with latitudinal and longitudinal data.

References

[1] Aman Kataria , M. D. Singh., A Review of Data Classification Using K-Nearest Neighbour Algorithm. International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 6, June 2013)
[2] S. Karthika* and N. Sairam , A Naive Bayesian Classifier for Educational Qualification. Indian Journal of Science and Technology, July 2015
[3] S. B. Kotsiantis , Decision trees: a recent overview. Springer Link , April 2013
[4] Gerard Biau , Analysis of a Random Forests Model. Jmlr ,2012.
[5] Dan Gao1,2, Yan-Xia Zhang1 and Yong-Heng Zhao1, Random forest algorithm for classification of multi wavelength data. RAA Journal ,May 2012.
[6] Eesha Goel , Er. Abhilasha., Random Forest: A Review. International Journal of Advanced Research in Computer Science and Software Engineering , January 2017
[7] Leo Breiman., Random Forests , Springer link, 2001
[8] Arnu Pretorius, Surette Bierman, Sarel J. Steel., A meta-analysis of research in random forests for classification. , IEEE , 16 January 2017
[9] Anne Ruiz-Gazen, Nathalie Villa., Storms Prediction: Logistic Regression vs Random Forest for unbalanced data. Case Studies in Business, Industry and Government Statistics 1, 2 (2007) 91-101.
[10] I.Rish , An empirical study of the naive Bayes classifier. IBM T.J. Watson Research Center , 2001