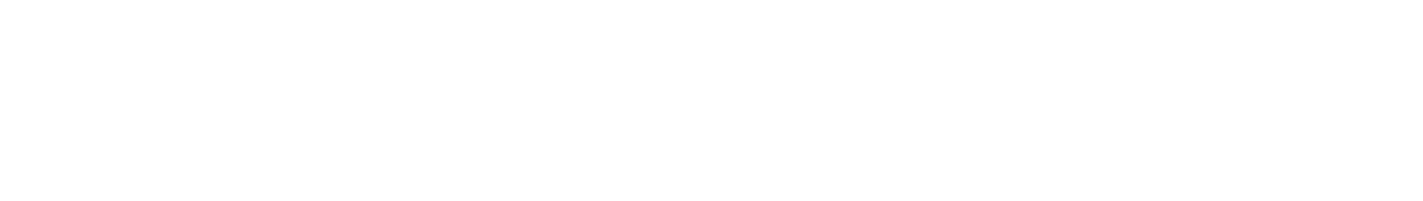
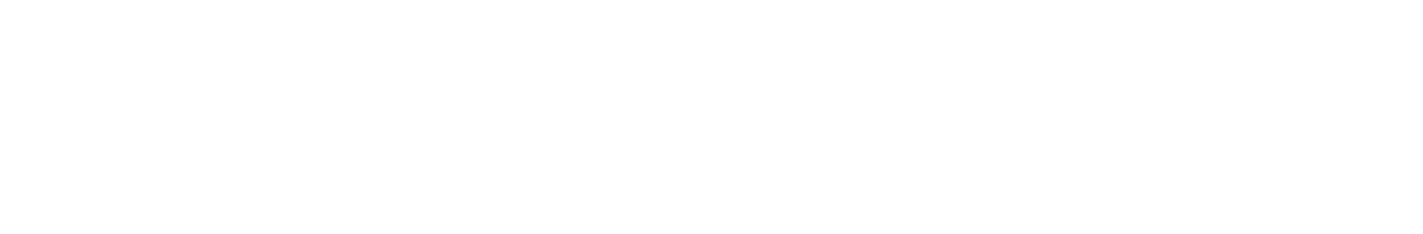


**SENTIMENT ANALYSIS**



Submitted to: Yogendra Sir,Sylvester Sir

# Submitted by: Abhinav Kothari, Ankit Gokhroo, Ayushi Agrawal

TABLE OF CONTENT

* Certificate
* Acknowledgement
* Abstract
* List of figures
* Introduction
* Theory
* Methodology
* Data Pre-processing
* Classification Method used
* Data visualization
* Result Analysis
* Annexure

**Certificate**

Date: 27/06/17

This is to certify that Mr. **Abhinav Kothari**, student of 3rd Year from Department of Computer Science, JECRC Foundation, has undergone a Project work from June 3, 2019 to July 17, 2019 in **Data Science & Machine Learning** titled “**Hotel Review Analysis**”

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Incharge Seal

**Certificate**

Date: 27/06/17

This is to certify that Mr. **Ankit Gokhroo**, student of 3rd Year from Department of Computer Science, JECRC Foundation, has undergone a Project work from June 3, 2019 to July 17, 2019 in **Data Science & Machine Learning** titled “**Hotel Review Analysis**”

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Incharge Seal

**Certificate**

Date: 27/06/17

This is to certify that Ms. **Ayushi Agrawal**, student of 3rd Year from Department of Computer Science, JECRC Foundation, has undergone a Project work from June 3, 2019 to July 17, 2019 in **Data Science & Machine Learning** titled “**Hotel Review Analysis**”

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Incharge Seal

## Acknowledgement

For making this project we would like to thank our course instructor, Yogendra Sir and Sylvester Sir .They helped us in learning the various concepts of the Data Science. Here we would also like to thank Kunal Sir and Forsk Team for mentoring us throughout this project.

## Abstract

Rapid growth of data comes with a challenge of sorting and analyzing them and finding some great insights, where raw data exists in graphical form, textual form or in images(structured or unstructured). Data science, machine learning and Deep Learning finds its application in various fields like finding sentiments of tweeter reviews etc ,stock market, recommendation systems, image processing, aerial photography, military, weather forecasting etc.

This report is about our project on “Sentiment Analysis of Hotel Reviews” that addresses about data preprocessing and post processing which includes plotting, classification and prediction of sentiment of review (positive or negative) in real time and the ability of machine learning algorithms to deal with different set of data. In this project, we have tackled a classification problem of predicting where the sentiment of review is positive or negative. We have tested and used Support Vector Machine(SVM) ,Naïve Bayes along with (GaussianNB, BernoulliNB, MultinomialNB), Random Forest to determine the results. In addition to this, we have also made use of Natural Language Processing(NLP) to deal with human language(in reviews).

## List Of Figures

Fig. 1 Data Science

Fig. 2 Decision Tree

Fig. 3 Logistic Regression

Fig. 4 Linear Regression

Fig. 5 Pokemon Activity

Fig. 6 Pokemon Activity Count in Various Cities

Fig. 7 Mapping of Pokemon Names with the ID’s

## Introduction

Sentiment Analysis of Hotel Reviews is NLP based project whose main aim is to deal with the reviews of user and predict its sentiment as Positive or Negative. Like many reviews for particular hotel written by user on website about quality of hotel service,food,surrounding etc, then role of our model is to predict the sentiment of those reviews, which help users in finding better quality of the hotel according to the reviews analysis in future.

Our approach to the problem is very simple, we first retrieve the dataset that consists of roughly 12,000 json files, we only include approximately 500 json files in training our model which is sufficient. When we load json files, in each json file we have a dictionary, in each dictionary we have two keys “HotelInfo” and “Reviews”. we skip the key “HotelInfo” which is of no use in training the model . And in “Reviews” key, we have reviews information including ['Author', 'AuthorLocation', 'Content', 'Date', 'Ratings', 'ReviewID',’Title’] as columns in a dataframe.

We have no labels in our dataset,which we made by help of library TextBlob .In TextBlob library on the basis of “Polarity” and “Subjectivity” we labelled 0 and 1 in our dataframe.

For features case,we choose ‘Title’ column from dataframe containing reviews and before fit it to the model we encode features by

short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

Than by preprocessing the data,appling various NLP techniques we train our model with different algorithms of machine learning, Finally, we compare the performance of each of these algorithms with the help of accuracy matrices, score matrices and then chooses the one that gives more accuracy.

## Theory

Data science is a "concept to unify statistics, data analysis and their related methods" in order to "understand and analyze actual phenomena" with data.[3] It employs techniques and theories drawn from many fields within the broad areas of mathematics, statistics, information science, and computer science, in particular from the subdomains of machine learning, classification, cluster analysis, data mining, databases, and visualization.

Data science – discovery of data insight

This aspect of data science is all about uncovering findings from data. Diving in at a granular level to mine and understand complex behaviors, trends, and inferences. It's about surfacing hidden insight that can help enable companies to make smarter business decisions.

For example:

Netflix data mines movie-viewing patterns to understand what drives user interest, and uses that to make decisions on which Netflix original series to produce

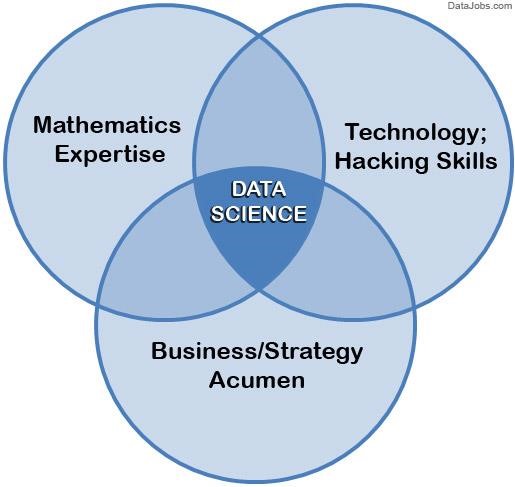
Data science – development of data product

A "data product" is a technical asset that: (1) utilizes data as input, and (2) processes that data to return algorithmically generated results. The classic example of a data product is a recommendation engine, which ingests user data, and makes personalized recommendations based on that data.

For example:

Amazon's recommendation engines suggest items for you to buy, determined by their algorithms. Netflix recommends movies to you. Spotify recommends music to you.

Machine learning and statistics are part of data science. The word learning in machine learning means that the algorithms depend on some data, used as a training set, to fine-tune some model or algorithm parameters. This encompasses many techniques such as regression, naive Bayes or supervised clustering.



**Fig. 1 Data Science**

Supervised and unsupervised learning describe two ways in which machines algorithms can be set loose on a data set and expected to learn something useful from it.

**Supervised:**

If we are training our machine-learning task for every input with corresponding target, it is called [supervised learning,](https://en.wikipedia.org/wiki/Supervised_learning) which will be able to provide target for any new input after sufficient training. Our learning algorithm seeks a function from inputs to the respective targets. If the targets are expressed in some classes, it is called *classification* problem. Alternatively, if the target space is continuous, it is called *regression* problem.

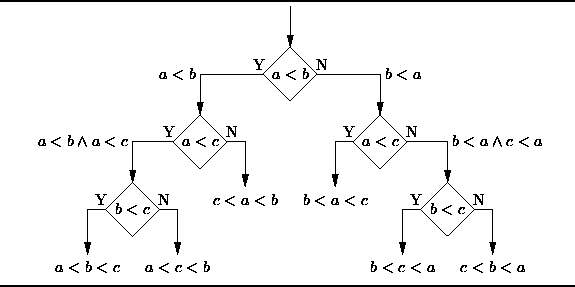
* **Regression** analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer causal relationships between the independent and dependent variables.

* **Classification** model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. Outcomes are labels that can be applied to a dataset.

**Unsupervised:** If we are training our machine-learning task only with a set of inputs, it is called unsupervised learning, which will be able to find the structure or relationships between different inputs. Most important unsupervised learning is *clustering*, which will create different cluster of inputs and will be able to put any new input in appropriate cluster.

* **Cluster** analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory [data mining,](https://en.wikipedia.org/wiki/Data_mining) and a common technique for [statistical](https://en.wikipedia.org/wiki/Statistics) [data analysis,](https://en.wikipedia.org/wiki/Data_analysis) used in many fields, including [machine learning,](https://en.wikipedia.org/wiki/Machine_learning) [pattern recognition,](https://en.wikipedia.org/wiki/Pattern_recognition) [image analysis,](https://en.wikipedia.org/wiki/Image_analysis) [information retrieval,](https://en.wikipedia.org/wiki/Information_retrieval) [bioinformatics,](https://en.wikipedia.org/wiki/Bioinformatics) [data compression,](https://en.wikipedia.org/wiki/Data_compression) and [computer graphics.](https://en.wikipedia.org/wiki/Computer_graphics)

1. **Decision Trees:** A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance-event outcomes, resource costs, and utility.

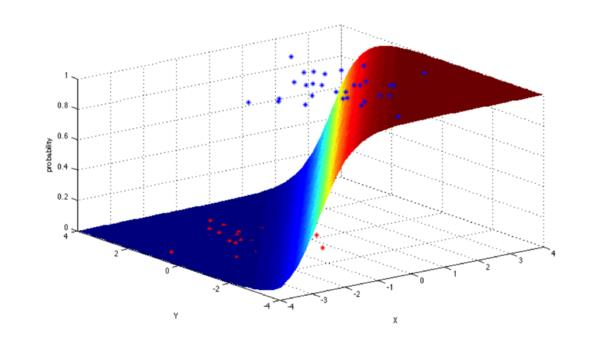


**Fig. 2 Decision Tree**

From a business decision point of view, a decision tree is the minimum number of yes/no questions that one has to ask, to assess the probability of making a correct decision, most of the time. As a method, it allows you to approach the problem in a structured and systematic way to arrive at a logical conclusion.

1. **Logistic Regression:** Logistic regression is a powerful statistical way of modeling a binomial outcome with one or more explanatory variables. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

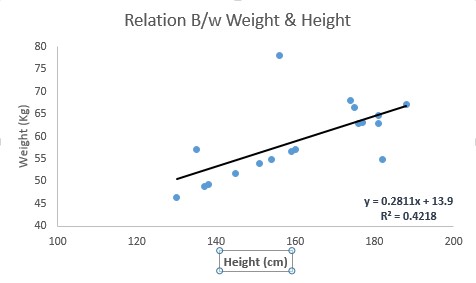
PAGE



**Fig. 3 Logistic Regression**

1. **Linear Regression**

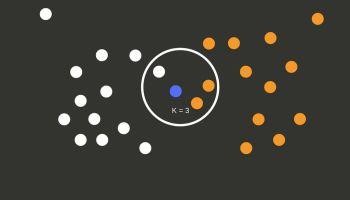
It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a \*X + b.



**Fig. Linear Regression**

1. **KNN (K- Nearest Neighbors)**

It is also a lazy algorithm. What this means is that it does not use the training data points to do any generalization. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function. These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables. If K = 1, then the case is simply assigned to the class of its nearest neighbor. At times, choosing K turns out to be a challenge while performing KNN modeling.



**Fig. 4 KNN**

PAGE

## Methodology

To understand the methodology adopted, we first understand our dataset and the variables.

**Dataset**

We scrap the data from the website “Trip Advisor” whose link is { http://times.cs.uiuc.edu/~wang296/Data },after scrapping we get approximately 12000 json files ,each json file contains a dictionary having two keys, “HotelInfo” and “Reviews”.

We only load approximately 1000 json files to train our model as it is sufficient dataset. We skip “HotelInfo” key as it contain only hotel information which makes no use in training of our model. From “Reviews” key we load the data in our dataframe which contains ['Author', 'AuthorLocation', 'Content', 'Date', 'Ratings', 'ReviewID',’Title’] as columns in a dataframe.

**Column Description:**

> ’Author’ : Name of the user who writes the review.

> ’AuthorLocation’ : Location where user live or where he/she reviewed.

> ’Content’ : Review written in descriptive way

> ‘Date’ : Date when review is written

> ‘Ratings’ : contain ratings of different features related to hotel in this format:

{'Service': '5', 'Cleanliness': '5', 'Overall': '5.0', 'Value': '5', 'Sleep Quality': '5', 'Rooms': '5', 'Location': '5'}

> ' ReviewID ': generated unique id for each review written

> ‘Title’ : contains review in conclusive review

**Features:**

> ’ **Title** ’: After analysing data we conclude “Title” column as our feature.

Than for extracting important words from the each review present in ‘Title’ column we make use of “Natural Language Processing (NLP)”.To deal with this data we apply following approach:

* Import NLP package as nltk.
* Cleaning the reviews by removing all the things except alphabets by substituting them by space(‘ ’).
* Than we convert the each word of the review in lower case.
* Than we split the review and convert it into a list.
* Than we download the stopwords  ( A stop word is a commonly used word (such as “the”, “a”, “an”, “in” etc) that a model has been programmed to ignore, when it takes the text as a input ) and than import from nltk by following code

nltk.download('stopwords')

from nltk.corpus import stopwords

* Than we remove the words from the list which we make above and remove the wods which are in the stopwords.
* Than we do a stemming(**stemming** is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form) by making portstemmer object.Basically it is finding the root word of each word present in the list.
* Than we again join all the words present in the review and get a final reviews for training purpose.

**Encoding of Features by Term Frequency Method and Inverse Document Frequency Method**

**Term Frequency - Inverse Document Frequency** it makes sure that less importance is given to most frequent words and also considers less frequent words.

**Term Frequency** is number of times a **particular word(W)** occurs in a review divided by totall number of words **(Wr)** in review. The term frequency value ranges from 0 to 1.

**Inverse Document Frequency** is calculated as **log(Total Number of Docs(N) / Number of Docs which contains particular word(n))**. Here Docs referred as Reviews.

**TF-IDF** is **TF \* IDF** that is **(W/Wr)\*LOG(N/n)**

Using scikit-learn's tfidfVectorizer we can get the TF-IDF.

**Dependent Variables:** latitude, longitude, appearedLocalTime, closeToWater, city, continent, weather, population, windspeed, closetowater, urban-rural, terraintype.

**Independent Variables:** pokemonid, class

## Data Pre-processing

For pre-processing, we considered Python as our options for the project. After some experimentation, we found that while R was easier for statistics and analysis of the data, the lack of uniformity among the various ML packages made Python our preference. The ML algorithms provided by the scikit-learn package do not function if the input data has missing values. Hence we either had to impute data at the missing slots or remove the instances that had these missing fields. Upon examining the data, we realized that in this project we don’t have any missing data but have categorical data that needs to be converted in integer values to make the model work. And to do that we used pd.get\_dummie() and LabelEncoder().

Additionally, the statistical correlation between some features (for instances where they were available) were appeared to be not related. Hence, we dropped those features. In addition to this, sampling was necessary for machine learning algorithms.

## Libraries used

* numpy
* pandas
* matplotlib.pyplot
* mpl\_toolkits.basemap
* geopy.geocoders
* seaborn
* math
* pylab
* pickle
* scikitlearn o sklearn.cross\_validation o sklearn.preprocessing

o sklearn.linear\_model o sklearn.neighbors o sklearn.ensemble o sklearn.metrics

## Regression Method Used

Here in this project we had used Logistic Regression. Here we had used Logistic Regression 2 times, first to train the data for Latitude part and another time for the Longitude part. However as the data is very large and we had many attributes so the training part of the data took very long time. So we had trained the data single time only and had saved the trained data in the pickle file, from where we can use that train data to make the predictions. Here we had used Logistic Regression only because we were very much familiar with this method, so it was easy for us to use it and our requirements of doing predictions completely met with the features of the Logistic Regression.

## Classification Method used

Here for making classification of the pokemons on the basis latitudes and longitudes we had used various algorithms like K – Nearest Neighbors, Logistic Regression in Classification mode, Decision Tree Classifier, Random Forest Classifier, SVM, but we got the highest accuracy that of 19.65% from the K - Nearest Neighbors, that too with 40 nearest neighbors.

## Data visualization

For doing visualization of the data we had used libraries like Basemap, Seaborn and matplotlib.

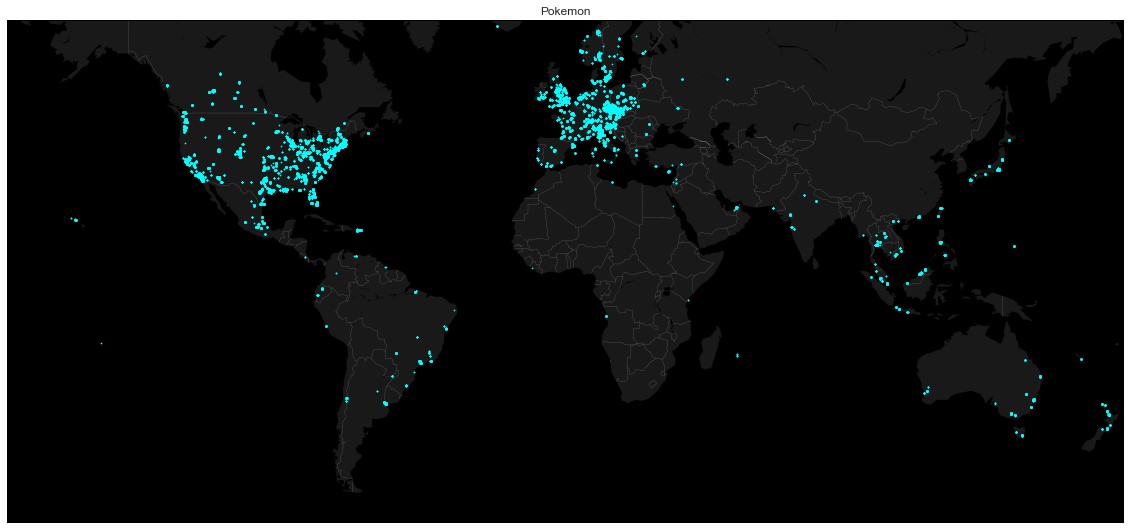
Through basemap we were able to easily map the different countries and continents and then on the basis of the longitude and latitude we were able to plot the pokemon through scatter plots.

And for making the plot for counting the number of pokemons county wise we had used seaborn library and with its help we had use matplotlib to make the bar chart.

## Result Analysis

### Part 1

Here we had to plot the pokemons based on latitudes and longitudes.

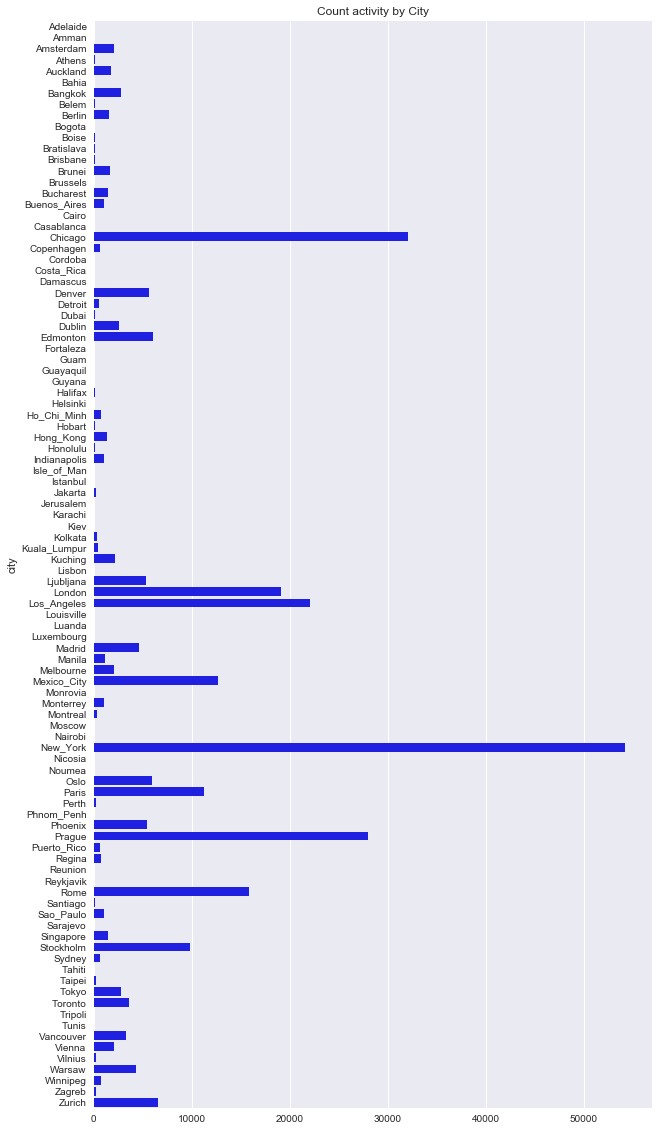


**Fig. 5 Pokemon activity**

Here red dots represent the pokemons present.

### Part 2

In this part we had to make a bar chart of the number of pokemons present in the different cities.



**Fig. 6 Pokemon activity count in various citites**

### Part 3

In this part we had to make the classification of the pokemons based on latitudes and longitudes, so here we had used K – Nearest neighbors to classify them with 40 nearest neighbors, but here we got the best accuracy score of 19.65% only as the number of features used to classify the data was very less, resulting in low accuracy score.

### Part 4

In this part we had to map the names of the pokemons with their ID’s. So to do this task we had used the map function to map names with the ID’s



**Fig. 7 Mapping of names with ID**

### Part 5

In this part we had to predict the location of the pokemon of our choice. So for this purpose we had used Logistic Regression, here we got the accuracy score of around 74.5 % in total. But training time was very high but when using the trained data from the pickle file the computation time for giving the results was less than 1 minute.

## Conclusion

Here in this project we were able to correctly visualize the pokemons found at various places around the world and also their activities in various cities. We were able to correctly classify the pokemons based on latitudes and longitudes although accuracy score for classification was not that good, but good enough with the amount of data present with us and the features used to make classification. We were able to correctly map the names of the pokemons with their ID’s. We were also able to make the predictions of the location of a particular pokemon with the accuracy of about 72.5%.

Annexure

**Code for visualization :-** import pandas as pd import matplotlib.pyplot as plt from mpl\_toolkits.basemap import Basemap

plt.figure(1, figsize=(20,10)) m1 = Basemap(projection='merc', llcrnrlat=-60, urcrnrlat=65, llcrnrlon=-180, urcrnrlon=180, lat\_ts=0, resolution='c')

m1.fillcontinents(color='#cc9966',lake\_color='#99ffff') m1.drawmapboundary(fill\_color='#99ffff') m1.drawcountries(linewidth=0.1, color="w")

x, y = m1(df.longitude.tolist(),df.latitude.tolist()) m1.scatter(x,y, s=3, c="red", lw=0, zorder=5)

**Code for training the data with Logistic Regression** from sklearn.linear\_model import LogisticRegression clas = LogisticRegression(random\_state = 0) clas1 = LogisticRegression(random\_state = 0) print 'Training data with latitudes...' clas.fit(X\_train, Y\_train) print 'Training data with longitudes...'

clas1.fit(X\_train, Y1\_train)