Facebook V: Predicting Check Ins

Sabrish Ramamoorthy,Tushar K.,Ankur B.

August 8, 2016

Contents

Chapter 1

Introduction

The goal of the Kaggle competition is to predict which place a person would like to check in to. For the purposes of this competition, Facebook created an artificial world consisting of more than 100,000 places located in a 10 km by 10 km square. Data was fabricated to resemble location signals coming from mobile devices, giving you a flavor of what it takes to work with real data complicated by inaccurate and noisy values.

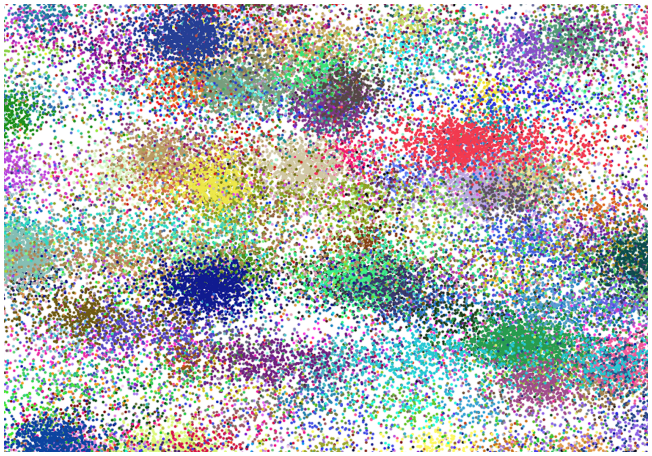

Figure 1.1: Sample coordinates plot

1.1 Objective

For a given set of coordinates, the task is to return a ranked list of the most likely places.

1.2 Challenge

Inconsistent and erroneous location data can disrupt experience for services like Facebook Check In.So the challenge is the usage of the noisy data and predict the most accurate place id on the coordinate system

Chapter 2

Dataset description

The train and test dataset are split based on time. There is no concept of a person in the dataset. All the row_id's are events and not people.

Descriptions of the variable

- **row_id-** id of the check-in event
- **x y-** coordinates
- **accuracy-** GPS accuracy of the location
- **time-** timestamp
- **place_id-** id of the business, this is the target we are predicting

The column such as time and accuracy, are intentionally left vague in their definition by facebook.

summary(fb1)
where *fb1* is the data stored from the csv

```
> summary(fb1)
     row_id              x                y             accuracy          time          place_id
 Min.   :       0   Min.   : 0.000   Min.   : 0.000   Min.   :   1.00   Min.   :     1   Length:29118021
 1st Qu.: 7279505   1st Qu.: 2.535   1st Qu.: 2.497   1st Qu.:  27.00   1st Qu.:203057   Class :character
 Median :14559010   Median : 5.009   Median : 4.988   Median :  62.00   Median :433922   Mode  :character
 Mean   :14559010   Mean   : 5.000   Mean   : 5.002   Mean   :  82.85   Mean   :417010
 3rd Qu.:21838515   3rd Qu.: 7.461   3rd Qu.: 7.510   3rd Qu.:  75.00   3rd Qu.:620491
 Max.   :29118020   Max.   :10.000   Max.   :10.000   Max.   :1033.00   Max.   :786239
```

Figure 2.1: R code sample of the data

Sample plot of place_id with respect to x and y coordinate.

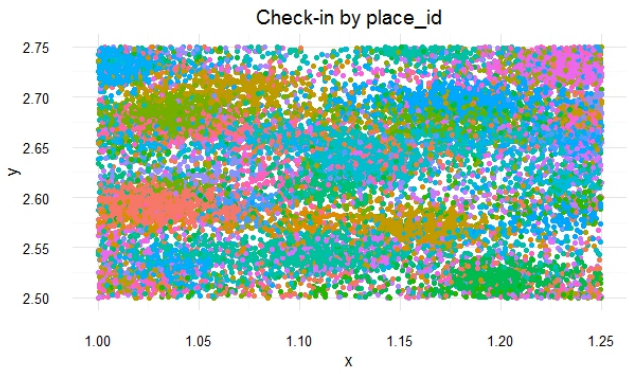

Figure 2.2: Plot of place id with respect to x and y coordinate

Chapter 3

Data preparation and analysis

The following code is used to convert the variables such that they can be properly summarized in the following sections.

3.1 Time Analysis

Approach 1: To find how time is related we first converted time to Epoch time for analysis.

```
128  fb$hr <- as.numeric(format(as.POSIXct(fb$time, origin="1970-01-01"),"%H"))
```

Figure 3.1: Conversion of time stamp to Minutes

When we plotted a 3D Scatter plot of the converted time in R, we got the below graph of position coordinates vs hour.

```
44  plot_ly(data = small_trainz, x = x , y = y, z = hr, color = place_id,
45  type = "scatter3d", mode = "markers", marker=list(size= 5)) %>%
46    layout(title = "Place_id's by position and Time of Day")
```

Figure 3.2: Script to display time variance

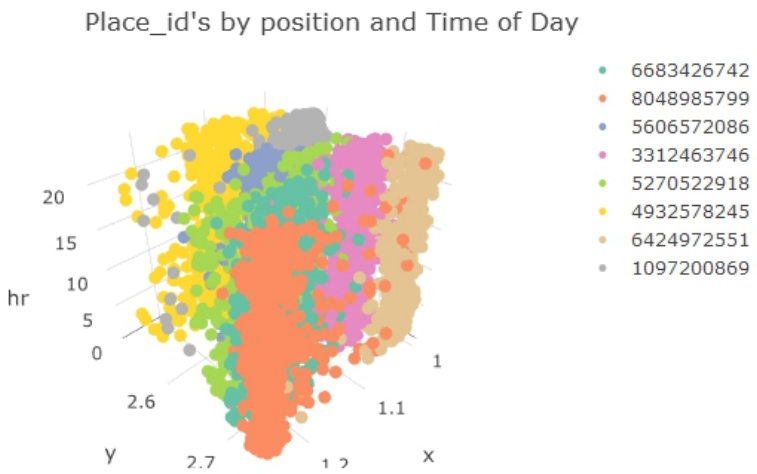

Figure 3.3: Checkin with respect to timestamp as second

We see that the even though the data is clustered but it is spread across time and no cluster is formed to do any analysis.

Approach 2: Considering time stamp given as minutes.

```
fb$hour = (fb$time/60) %% 24
```

Figure 3.4: Conversion of time stamp to Minutes

When, we plotted this new 3d graph with new time stamp there was clustering in a specific time slot. Indicating closing and and opening of museum, pub or a park.

```
plot_ly(data = small_trainz, x = x , y = y, z = hour, color = place_id,
type = "scatter3d", mode = "markers", marker=list(size= 5)) %>%
  layout(title = "Place_id's by position and Time of Day")
```

Figure 3.5: Script to display time variance

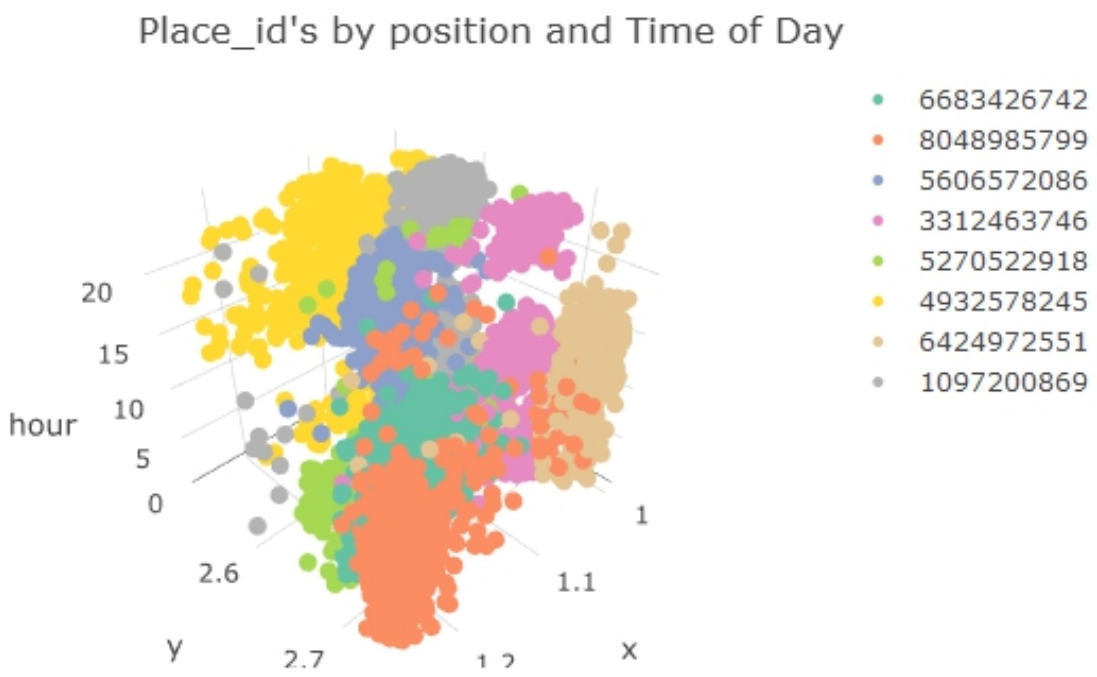

Figure 3.6: Checkin with respect to timestamp as minutes

Since there was clustering for a given id, we used this time for classification the data set based on hour.

3.2 Accuracy Analysis

For most end users, accuracy is heavily influenced by having usable signals from several GPS satellites instead of just a few.In the dataset for every checkin, we have accuracy which may affect the coordinates of the checkin. To find how the accuracy may affect the prediction model, we plotted a box plot (accuracy v place_id).

```
62  p <- plot_ly(small_trainz, x = accuracy, color = place_id, type = "box")
```

Figure 3.7: Boxplot Script

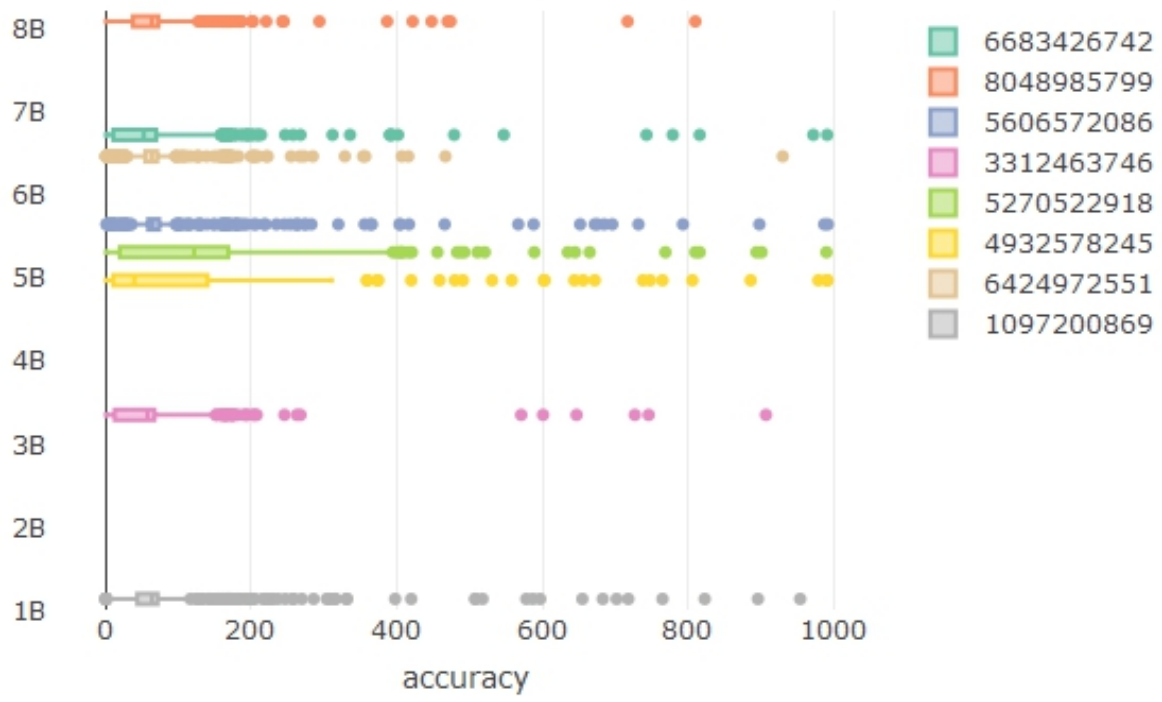

Figure 3.8: Boxplot Accuracy and Time

To get better perspective of how accuracy may effect the data we plotted the 3D scatter of the coordinate with accuracy to get better perspective of the data.

```
plot_ly(data = small_trainz, x = x , y = y, z = accuracy, color = place_id,  type = "scatter3d", mode = "markers",
        marker=list(size= 5)) %>% layout(title = "Accuracy vs position coordinate ")
```

Figure 3.9: Accuracy vs PositionID Script

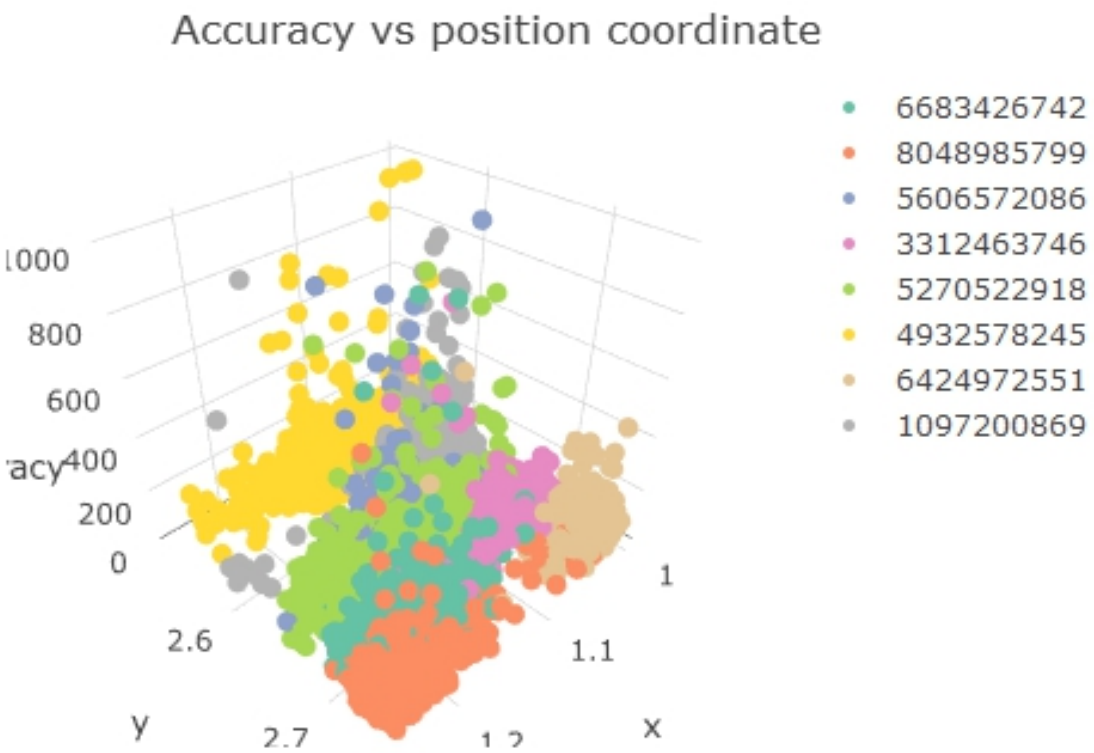

Figure 3.10: Accuracy vs PositionID X and Y

Here, we see that for a given set of place id the data set lies in the range of below 300 accuracy mark. To make the model more accurate we considered the data set that is below the 300 mark. So that the outlier and have excluded them to train the data. So that we can get a more accurate result.

Chapter 4

Prediction

4.1 Algorithms and use-cases

4.1.1 K-means Clustering

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells - reffered from Wikipedia.

Definition Given a set of observations (x1, x2, , xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k (n) sets S = S1, S2, , Sk so as to minimize the within-cluster sum of squares (WCSS) (sum of distance functions of each point in the cluster to the K center).

Standard Algorithm Reason :

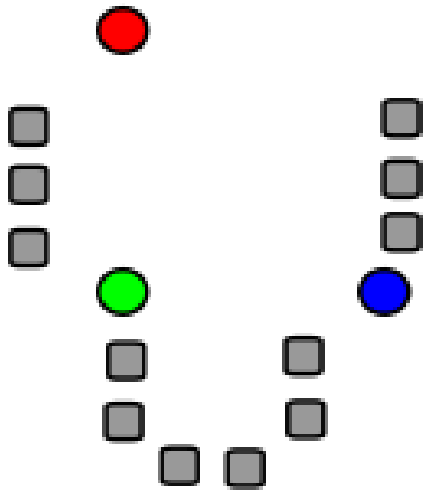

Figure 4.1: k initial "means" (in this case k=3) are randomly generated within the data domain (shown in color).

Variables considered for the Kmeans: Timestamp

Following are the Map Reduce job implementation outputs:

2. Location based clustering: **Introduction** : It is a Location based clustering on the output from K-Means step. The output from K-Means clustering had a large amount of data for each centroid value. **Reason** Applying KNN on large dataset is difficult and hence we divided the clusters according to 10x10 into 1x1 blocks. The final output had 600 rows, 10x10x6.

3. KNN **Definition**In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point.

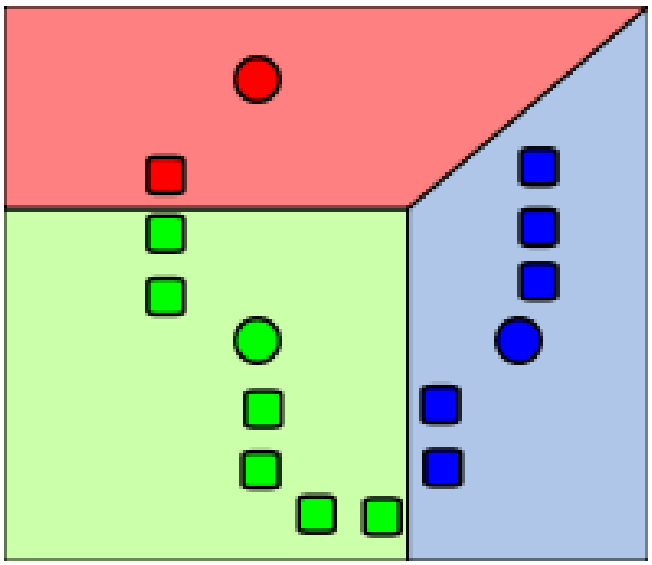

Figure 4.2: k clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

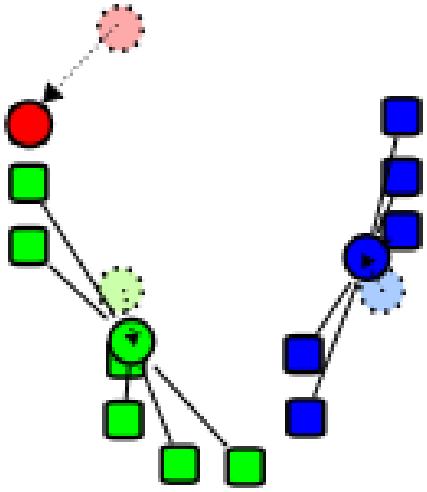

Figure 4.3: The centroid of each of the k clusters becomes the new mean.

The accuracy of the k-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification.

In our case each 1x1 block had multiple location, so we selected the nearest location based on Users co-ordinates.

We took two datasets R:Training dataset and S:Test Dataset.

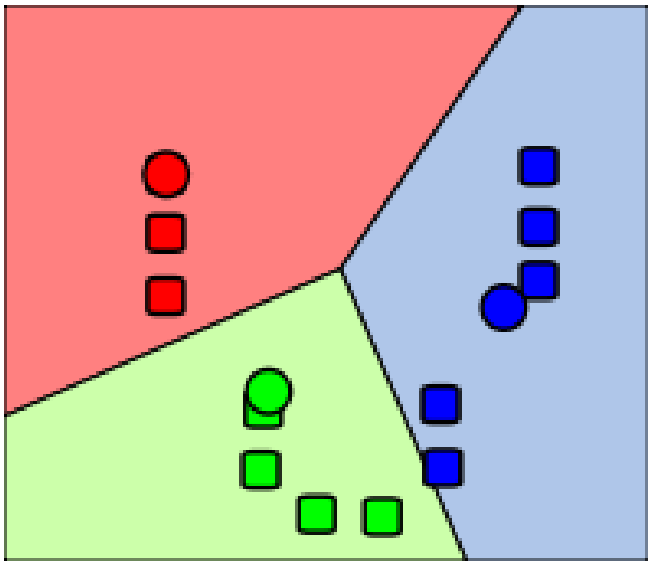

Figure 4.4: Steps 2 and 3 are repeated until convergence has been reached.

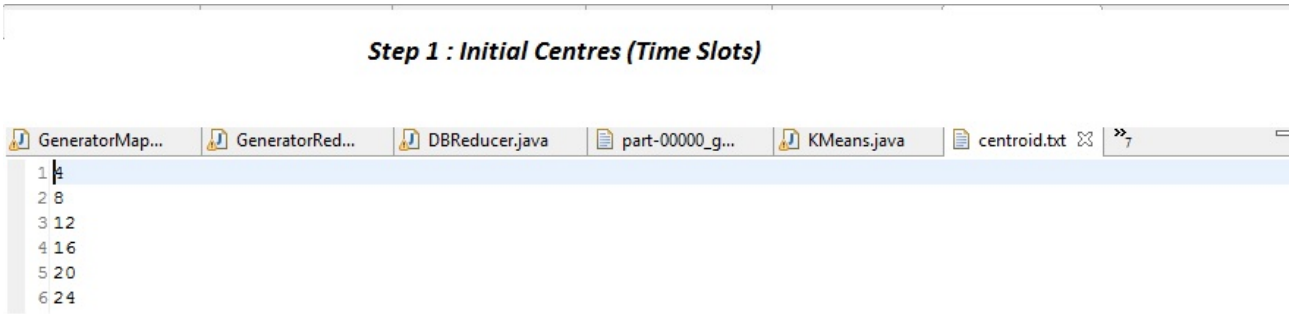

Figure 4.5: Considering Centroids as 4,8,12,16,20 and 24

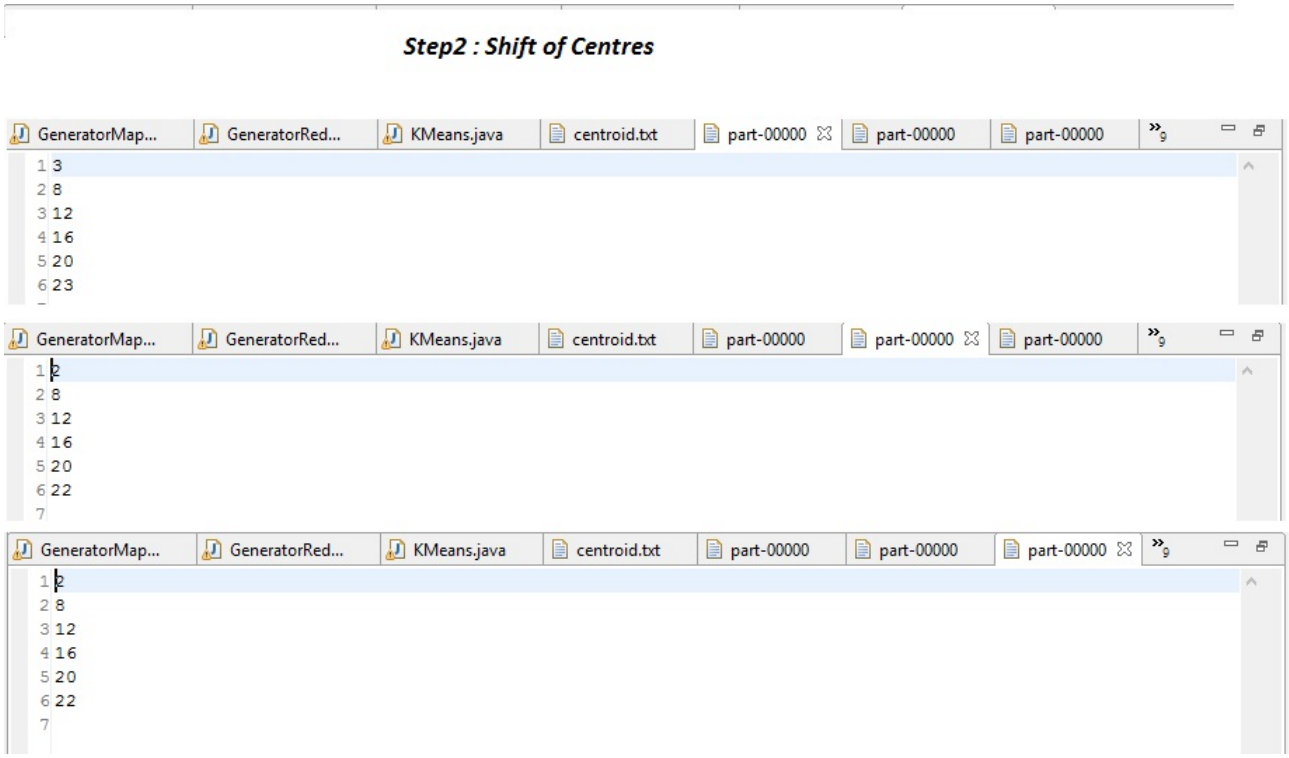

Figure 4.6: Supervised learning.Calculating mean based on input(Centroid.txt)

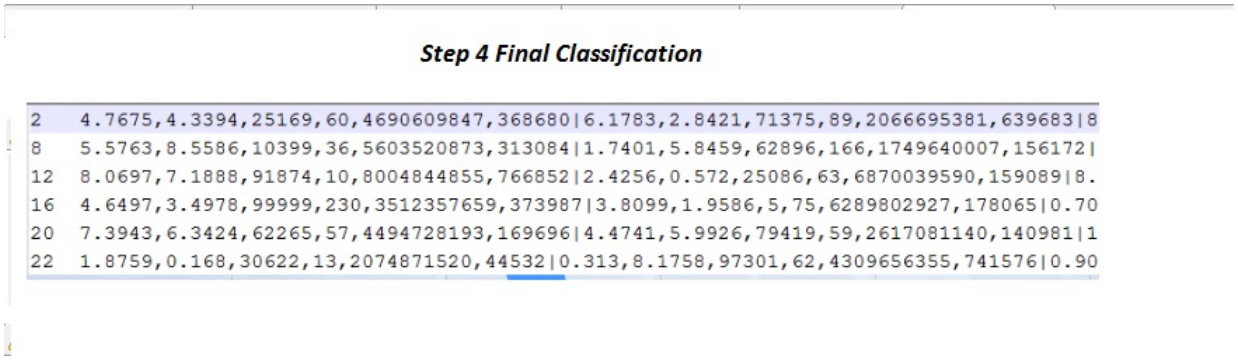

Figure 4.7: Clustering based on mean centroids

Train Dataset

row_id	x	y	accuracy	time	place_id
0	0.7941	9.0809	54	470702	8523065625
1	5.9567	4.7968	13	186555	1757726713
2	8.3078	7.0407	74	322648	1137537235
3	7.3665	2.5165	65	704587	6567393236
4	4.0961	1.1307	31	472130	7440663949
5	3.8099	1.9586	75	178065	6289802927

Figure 4.8: This is the training dataset supplied

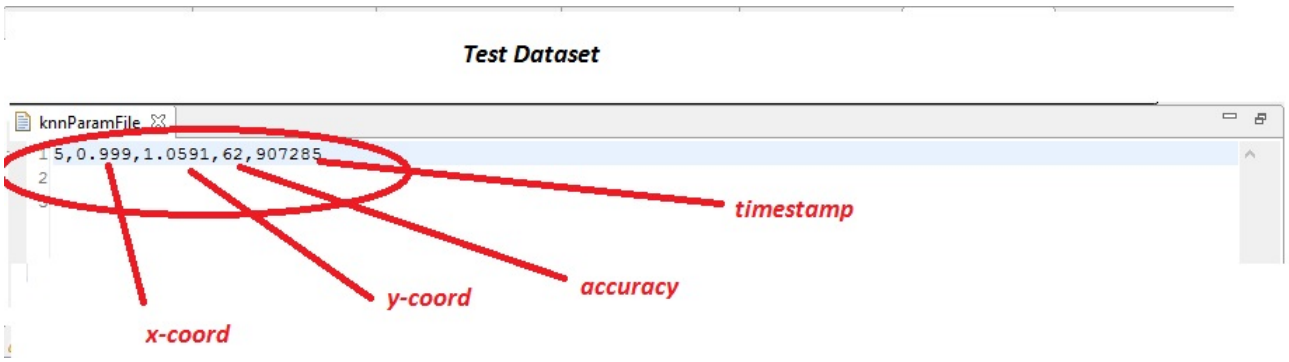

Figure 4.9: This is the test dataset supplied

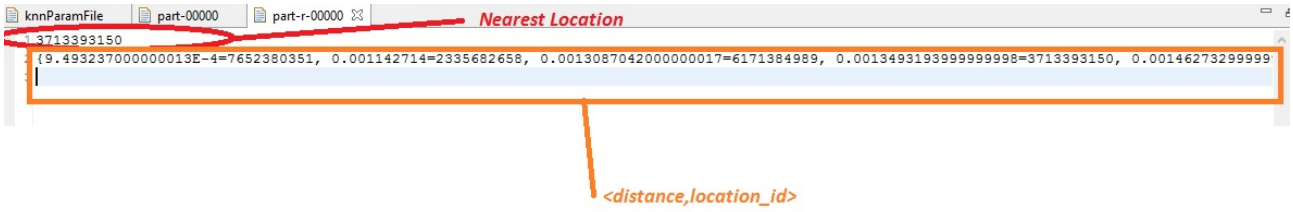

Figure 4.10: This is the output

Chapter 5

Problems Faced

- **Block data set approach-** During the initial analysis we considered the location as a region rather than a point in map. So for a given placed ID we planned to draw a region based on the coordinate given by the facebook. And when we received a point to predict its location ID we just need to check whether the point lies in the region. The problem with this approach was that it was time consuming and was not suitable for a large dataset. Then we changed the approach to point based and classification based on time stamp.
- **Division of Plots into 100 blocks-** We divided the 10x10 map into regions of 1x1 block because knn has a drawback that it cannot handle huge set of data.Each block is given to a mapreduce operation so that nearest neighbor could be identified. The problem with this approach was that it was difficult to include time factor and accuracy parameter for better prediction.
- **Clustering-** Identifying the number of cluster so that the correct data set could be incorporated and which would aid to get accurate results. Initial approach was to divide the time into 3 sets of 8 hrs,16 hrs and 24 hrs. However this data set has too many data and the classification process was taking too much time. Then doing trial and error we finalized on the 4, 8,12,16,20 and 24.
- **Identifying the time stamp-** We were required to identify the vagueness of timestamp. The timestamp was not clear whether it was in seconds, mins or hours. So we started by considered timestamp as seconds from epoch time i.e seconds from 1-1-1970. But this approach did not created any relation between checkin for a location id. Then we did analysis assuming that the given time stamp is in minutes. And by doing this and plotting the graph we understood that for each place_id there were clusters of checkin this made sense as for given place like museum, park and pub there will be a specific time slot when there would have lot of checkin and this would aid in the prediction.
- **data.table-** reading in the data
- **FNN-** k nearest neighbors algorithm

Chapter 6

Technologies used

6.1 System description

The following symbols characters are reserved by LATEX because they introduce a command and have a special meaning.

6.2 JSP-Servlet Web Service

JSP-Servlet web service is used to retrieve the data from MongoDB and also was used for creating a interactive User Interface.

6.3 D3.js

D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS.

D3.js was used for creating data visualization and for creating UI.

6.4 MongoDb

MongoDb
MongoDB (from humongous) is a free and open-source cross-platform document-oriented database. Classified as a NoSQL database, MongoDB avoids the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster.
In our system MongoDb was used to store the data in the database. The data is stored in the following format: loc_key: '0_0', loc_id:'8328367769', x_coord: '0.6522', y_coord: '0.96505', checkin_count:'2', shift1:'0', shift2:0, shift3:2

6.5 Hadoop

MapReduce
MapReduce is the heart of Hadoop. It is this programming paradigm that allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster.
The term MapReduce actually refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job.

The MapReduce job performed the k-mean and knn operations on the data set. So that the fast computation was possible.

6.6 AWS

Amazon Web Services (AWS), offers a suite of cloud computing services that make up an on-demand computing platform. These services operate from 13 geographical regions[3] across the world. The most central and best-known of these services arguably include Amazon Elastic Compute Cloud, also known as "EC2", and Amazon Simple Storage Service, also known as "S3". AWS now has more than 70 services that span a wide range including compute, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools and tools for the Internet of things.

- **Amazon EMR-** With Amazon EMR (Amazon EMR) you can analyze and process vast amounts of data. It does this by distributing the computational work across a cluster of virtual servers running in the Amazon cloud. The cluster is managed using an open-source framework called Hadoop
- **Amazon EC2-** Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

6.7 R

Data analysis done in R

Library used:

- **dplyr-** Dataframe manipulation
- **ggplot2-** Graph Plotting
- **plotly-** 3D plotting
- **tidyr-** Dataframe manipulation
- **data.table-** reading in the data
- **FNN-** k nearest neighbors algorithm

Chapter 7

Conclusion

As data clustering has attracted a significant amount of research attention, many clustering algorithms have been proposed in the past decades. However, the Parallel K-Means Clustering Based on MapReduce enlarging data in applications makes clustering of very large scale of data a challenging task. In this paper, we propose a fast parallel k-means clustering algorithm based on MapReduce, which has been widely embraced by both academia and industry. We use speedup, scaleup and sizeup to evaluate the performances of our proposed algorithm. The results show that the proposed algorithm can process large datasets on commodity hardware effectively

Chapter 8

References

- **Dataset-** www.kaggle.com/c/facebook-v-predicting-check-ins
- **Algorithm-** en.wikipedia.org/wiki
- **Papers-** Parallel Clustering Algorithms. Parallel Computing 11, 275290 (1989)
- **Algorrithm-** k nearest neighbors algorithm