Optimal Bus Route Selection based on Passenger Location

Sakib Ahmed[1*] **and Shamit Ahmed**[1+]

[1]Ahsanullah University of Science and Technology, Computer Science and Engineering, Dhaka, Bangladesh
[*]sakib019898@gmail.com
[+]shamitaditta380@gmail.com
[1]these authors contributed equally to this work

ABSTRACT

We often notice large local buses carrying small amount of passengers. Large number of buses cause traffic jam. Optimal route selection is necessary in order to ensure maximum passenger count. In our solution we plot passengers in a neighborhood and create clusters. By checking for large clusters at a specific time, we determine the optimal route for a bus carrying students to Ahsanullah University of Science and Technology.

Introduction

Dhaka being the capital of Bangladesh, roughly has a population of 20 million people. Transportation system and traffic jam is one of the massive problems in Dhaka. Scope of the problem is huge since it involves every single person in the city. We could not identify any current solution to this problem localized for Dhaka. Every city dweller needs to care about this problem since he or she faces constant traffic jam in their daily life. It is such a huge problem that takes away large chunks of our important time. Students of AUST have been waiting for their own transportation service forever. Our solution will come very handy if authority decides to start a bus service. Bus distribution should be calculated and depend on student count at each area. Features considered are latitude, longitude of each person and their class starting time. We get different images consisting varying cluster sizes at a specific time.

We decided to distribute buses to optimal routes by using clustering algorithm. We defined clusters based on number of plots at each location. Then we calculated cluster sizes and distance between each cluster and the destination which is AUST campus at Tejgaon, Dhaka. We could not look for related papers due to short time frame. We decided to create a solution for existing problem by ourselves. In next sections we will be discussing background,methods, results, conclusion etc.

Background

Our problem mostly involves clustering of nearby plots. We tried using classification algorithm such as K-NN to classify clusters based on their size. We decided to use K-means clustering algorithm to determine the cluster size at each area. It is an unsupervised learning method that determines groups or clusters of similar type of data.

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. K-means 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.

Tools used to create the project include python programming language, Jupiter Notebook, google map plot etc. Dash is an open-source Python framework for building web-based analytic applications developed by plot.ly.

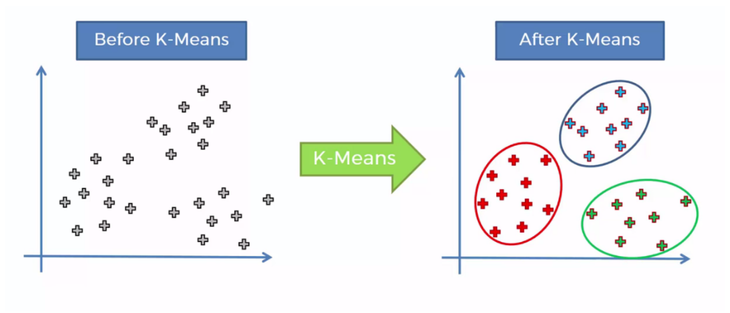

Figure 1. K-Means Clustering

Methods

0.1 Method 1

Our first method includes K-Means Clustering. Here are the main steps to reach our desired solution.

Step 1: Data Pre Processing where we processed the raw dataset and first filtered out only required columns and sorted data in required format.

step 2: Apply K-Means Clustering.

X=df8.loc[:,['place$_{l}atitude',' place_{l}ong']]$
$kmeans = KMeans(n_{c}lusters = 8, random_{s}tate = 0).fit(X)$
$id_{l}abel = kmeans.labels$
$gmap2 = gmplot.GoogleMapPlotter(23.7638, 90.4069, 13)$
$ptsymb = np.array(['b.',' g.',' r.',' c.',' m.',' y.',' k.',' c.']);$
$ptsymb2 = np.array(['yellow',' green',' red',' cyan',' megenta',' blue',' black',' white']);$
$plt.figure(figsize = (12, 12))$
$plt.ylabel('Longitude', fontsize = 12)$
$plt.xlabel('Latitude', fontsize = 12)$
$foriinrange(8):$
$cluster = np.where(id_{l}abel == i)[0]$
$plt.plot(X.place_{l}atitude[cluster].values, X.place_{l}ong[cluster].values, ptsymb[i])$
$gmap2.scatter(X.place_{l}atitude[cluster].values, X.place_{l}ong[cluster].values, c = ptsymb2[i], size = 100, marker = False)$
$plt.show()$
$gmap2.draw("C :_{m} ap2.html")$

Results

Here is the visualization of the plotting of latitude and longitudes of each data using google map.

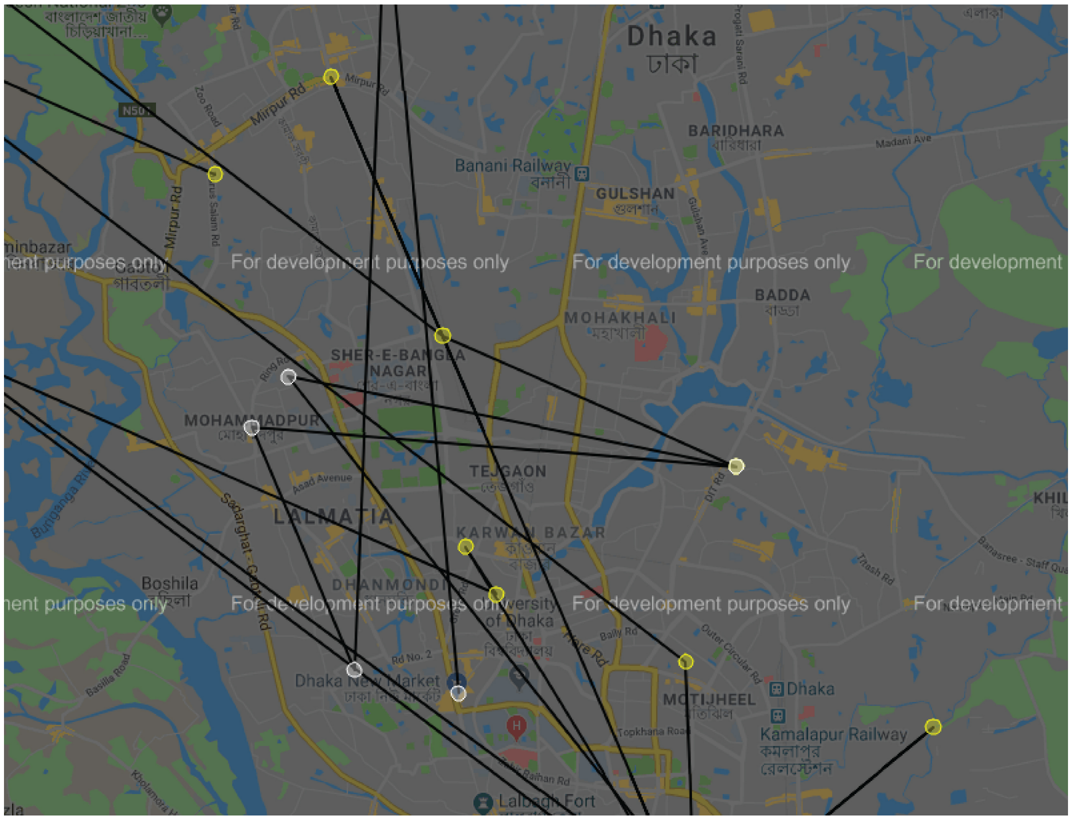

Figure 2. Plotting and Distance Measurement

Conclusion

Our future plan for this project includes applying more clustering algorithms such as DBSCAN algorithm in order to increase efficiency of the route selection and make it more realistic.

Limitations of the project are that we could not properly research the background of our process and study related reference papers. We definitely need more work on accuracy and error calculation.