Framingham Bus Project: Optimal Bus Routes Project Report

Partner: Lincoln Lynch

Student: Tri Hoang

The Problem

Budgets across multiple departments of the Framingham Public School System are getting tighter as costs are increasing. It is getting harder to afford the same high level of academic support that has been always the standard of our system. It is urgent for the system to cut unessential spending and operate at a higher efficiency. The saving will be redirect to essential functions of the system that ensure the success of our students.

Proposal

The Framingham Public School's Bus System is very expensive to run. It is thought to be possible to increase the efficiency of the system, and thus, reduces the number of buses needed. We are hoping to do that with the help of data science tools, historical data of students' location, traffic data, etc. Even with a reduction of just one bus, it would provide a tremendous help to other departments of the system. Otherwise, if no bus can be removed, updated efficient routes would prove useful to require less travel time of buses. And thus, it reduces gas consumption and ensures students' on-time arrival.

Data

Framingham Public School System has provided us with the most updated data of the 2018-2019 school year. We were also able to obtain data from the past couple years (2015-2016 to 2017-2018). Thus, we have a dataset cover continuously from 2015 up until now. The dataset consists of students' address, their current bus stop, their current school, their eligibility status.

Modified Data

The original data is stored in a way that is hard to be interpreted by data science tools. For example, students' locations are stored as a house number and street name, instead of latitude and longitude coordinates. Schools are identified by their names, instead of an address or coordinates. And all students from different schools, which would require a different bus route, a merged within one document.

To overcome those issues and allow the data to be interpreted by the data science tools, some augmentation was done to it. Street name were converted to coordinates, using a third-party geocoding API. The documents were split into multiple data frames, grouped by the schools. The addresses of schools were looked up online and substituted by its coordinate. And various other augmentations were performed. As the result, the documents are now finally readable by our data science tools.

Approach

First of all, we need to convert the addresses into a latitude and longitude coordinates for easy interpretation. It was done through using the HERE Geocoding API, with techniques such as rate limiting to overcome the service's usage limitation.

Afterward, we plotted the coordinates as an overlay over a map. Even though this plotting doesn't help with the actual algorithm itself, it shows us the general layout of the student locations. And thus, we can perform some visual analysis, and see ahead of what the result will look like.

Through the map, I notice that there are some regions with a large amount of student all clustered within it. And its proximity to the school varies. Through this, I thought of using a clustering algorithm to cluster students that are close to each other. Besides that, I also need to set weights to each student address, based on their distance from school. The reason behind that is because students who live further away will require more time to get to school, and thus, require earlier pickup time.

However, with the current algorithm, distance between an address to the school is based on flying distance. It doesn't take into consideration of the road layout, walkability, etc. I will need to use a third-party API such as Google Maps or Bing Maps to find the shortest path between the address and the school, and use that as a feature in our analysis.

Besides the logistics analysis, we can allow some improvement by determine whether a student's walking distance from school is acceptable. With cases that the distance is acceptable, according to weather and schedule data, we can remove those nodes and ask the students to walk.

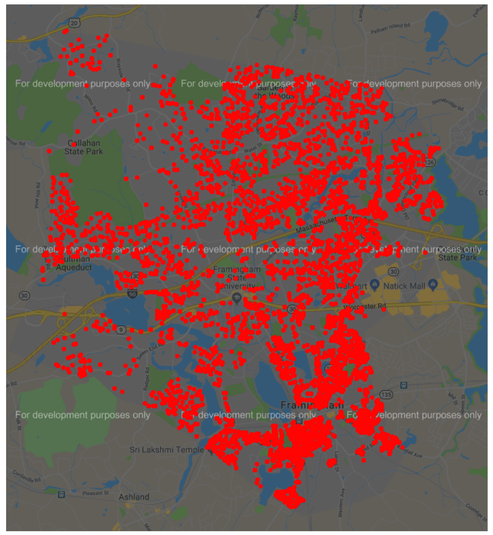

Figure 1: plotting of all students' addresses

Algorithm

In this project, I will work with the Framingham High School (FHS) first, since it has the largest students body and thus, has high potential of reducing the number of buses.

1. Benchmarking

To benchmark each clustering algorithm, I have a **score_of_cluster()** function that consumes HERE API to calculate the total amount of time it would takes to traverse all the nodes in the cluster, in a door-to-door fashion, and then to the destination. This scoring mechanism does not represent the true time it would take for the bus to traverse through that cluster, since students can walk to a bus stop and thus, reduced the amount of travel required for the bus. However, my reasoning behind this is that since we are trying to accommodate students by reduce amount of walking they have to take. And since we are using the same benchmark for the entire program, it has become standardized and thus allow good comparison between clusters. There are some other problems with this benchmark method such as one-way road, street size to accommodate bus, etc. However, it would increase the complexity of this project and makes it not fit in the semester timeframe.

2. Current routes

The current real-world bus clusters were extracted by using filtering and grouping functionalities of pandas. With the data frame of FHS itself separated from preprocessing, I grouped it by **am_busnumb**; thus, each of the groups is a cluster. Benchmarking this current routes yielded a score of **121703**. And the current routing uses **34 buses**.

3. K-Means

My first approach was to apply K-Means clustering algorithm. K-Means seems like a perfect fit for this problem since it works solely on clustering nodes by its distant to each other. It should be able to effectively group each cluster of students that live with in an area together. However, in practice, it does not perform as well as I expected. I tried to run the algorithm with **n_clusters** equal to the number of buses in the current system, and it yielded a score of **221800**, which is way worse than what the current system can do. The reason behind this poor performance could be that K-Means, or any other clustering algorithms, ignore all the small details of road layouts, streets rules, etc. And thus, these small details added up and caused a large deteriorate in performance.

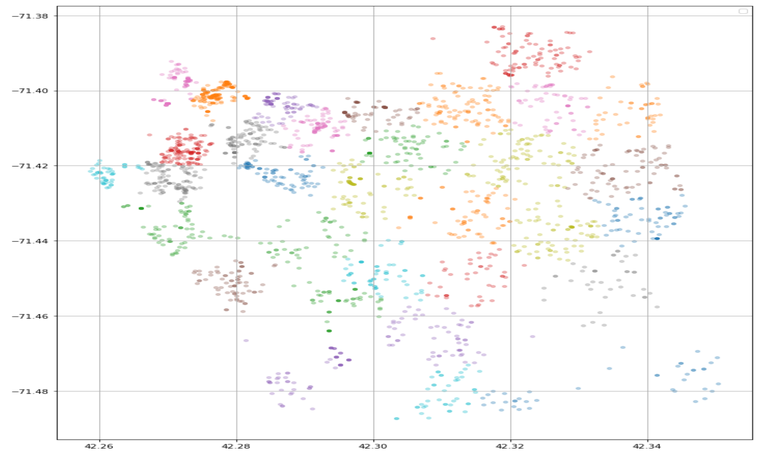

Figure 2: Clustering by K-Means into 34 clusters

4. DBSCAN

My second approach was to apply DBSCAN clustering algorithm. DBSCAN were built specifically for geo-clustering, and thus, fits very well into this project. However, in real practice, it requires a lot of fine tuning to get the clustering to work correctly. I need to manually set the **eps** values, which is the maximum distant between two nodes to consider those two nodes neighbors. It works well as a clustering tools to group some close neighborhood. However, it mistakenly groups a large chunk of the dataset as noise. This algorithm has potential to produce good result. However, it will require much further investigation and variable configuration.

Result & Summary

Within the scope of this semester, I was not able to find the clustering that would provide a more efficient bus routes. K-Means provided a clustering but it is far from optimal. DBSCAN provided some good clustering, however, it leaves out a lot of nodes since it sees those as noise. In general, the problem with clustering itself is that it ignores environment variables such as road layouts, street rules, etc. These small details could add up times required for the bus to travel, and thus, decrease its efficiency. This is something that needed to be address in future work.

Future Work

The most promising approach for future work would be to continue fine tuning the DBSCAN algorithm. By optimizing its eps distance, and other various DBSCAN settings, I think it should be possible to find a better clustering. Besides, I must find some way to introduce route features into the clustering model.

There are some possible improve that we can apply once the basic routing has been figured out. For some routes, we can allow student to transfer from a bus to one school to another bus that head to another school, at some pre-defined central location. This allows more efficient pick up in case where a region that has students from multiple schools. They can all board one bus, then some of them can get off at a transfer station and get on the bus to their school. It is very likely that this method will greatly reduce number of buses needed. However, this would require a scheduling algorithm, and thus increase the problem's complexity greatly.