Sandy Toes: SDTS San Diego Transit Simulation

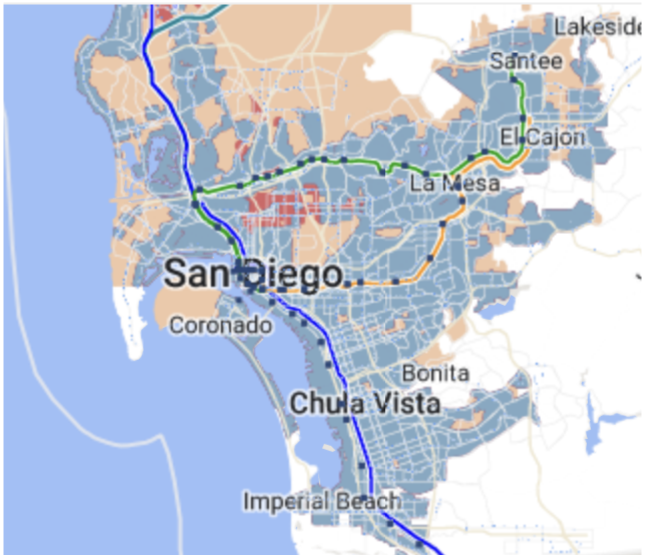
Contents

Project Description

The goal of this data science simulation project is to optimize the San Diego transit system. To achieve this goal, we will use various datasets, including but not limited to transit schedules, ridership data, and population data. We will apply machine learning and statistical models to analyze this data and uncover patterns and insights. Some of the models we will use include:

- **Clustering** models to group transit stops and stations based on similar demand patterns and geographical proximity. This will help identify potential areas for new routes and services.
- **Network optimization models** to analyze the transit system's network topology and identify the most efficient routes and connections.
- In this map, there is a red section where it is reported to be a high transit market. We should zone into this specific section and improve the transit in this area to narrow down the scope of our project.

- We should make it so that with the given routes the buses service the areas with loewst income the most while still giving adequate service to the other regions



Possible Project Directions

1. We could use the data available, create our model of the area, and use K-means to define the best places to add bus-stops, then add new routes connecting them to the current transit system.
2. We could create our model of the transit system, and do something similar to the paper and optimize the routes between the bus stops. Rather than using K-means to denote new bus stop locations, we can assign higher fitness to solutions with more routes that service these areas of high population while still giving a minimum “base amount” of service to the other bus stops. We can also attempt to minimize the edge density as in the paper. I think this would be really interesting and cool, but I don’t know how to do it quite yet.
3. We could use Linked Connections rather than writing our own algorithms to route plans, but I don’t know enough about the software to know how applicable it is.

Literature Review

This paper is an example of one way that we can approach this project. It details the use of the K-Means algorithm and a genetic algorithm to optimize a portion of a bus system in Antalya. The genetic algorithm maps chromosomes to a combination of different routes to create lines. This study takes stop density, stop layout, and passenger population of each stop into consideration.

- To improve this transit system, they wrote a genetic algorithm that minimized the occurrences of a road (edge) being used by the multiple routes. They describe this as follows: “If there is a repetitive edge use in same direction between 2 nodes in the route suggestions, there is a 1 point penalty from the fitness score for each encounter. 10 points are penalized for each node that is not included in solution due to full coverage constraint.”
    - I believe that each node here represents a bus stop, which would be the centroid of a cluster.
- Using their nodes, they made permutations of routes that span 3, 4, or 5 of those nodes which gave them an amount of routes. They discarded some routes that were too long or short, and then programmatically examined the remaining routes to make sure every node was covered.
- “The routes of the existing system were converted into routes in the proposed system domain. To do this, every stop in the current system was included to the nearest node in the proposed system. Converted routes with less than 3 nodes were eliminated because of external connections outside of the problem boundary.”

I don’t yet have a way of understanding the connections between what we are able to do and this paper. It serves as inspiration for now.

Approach

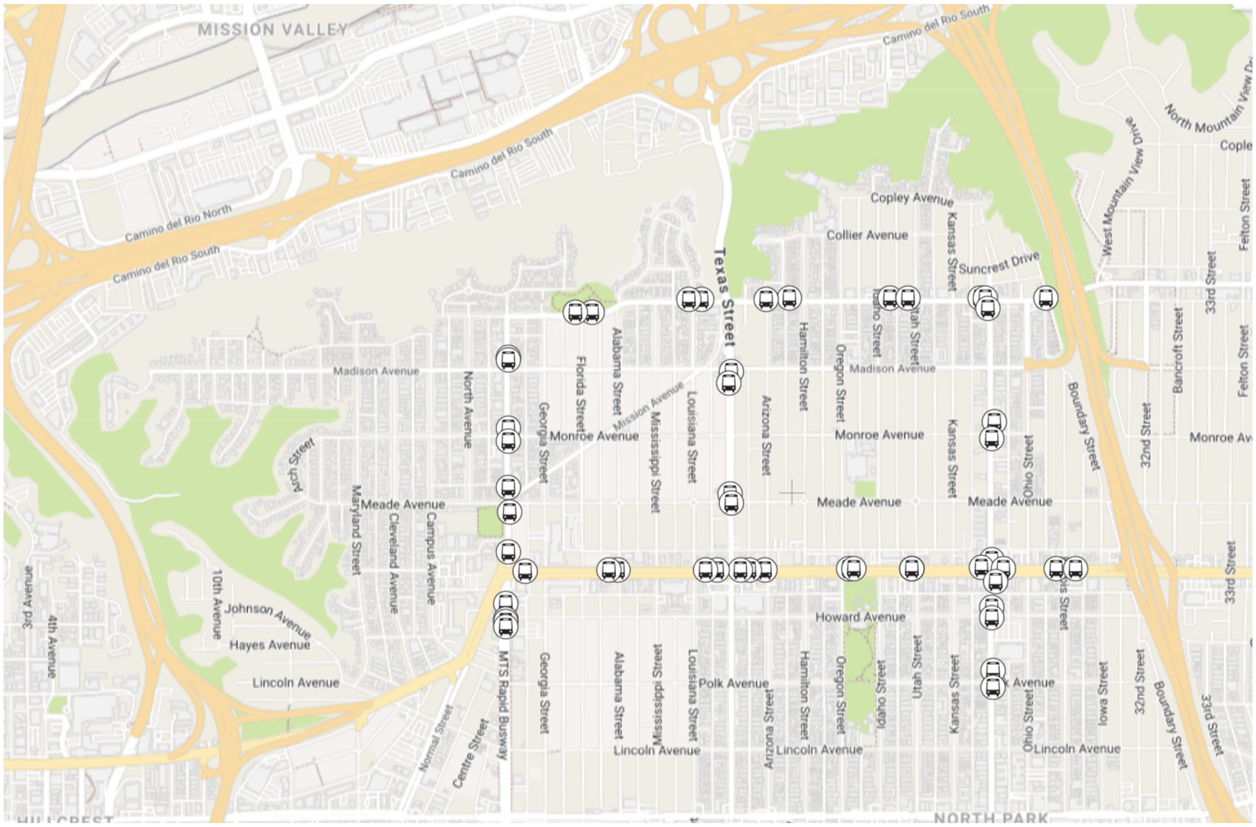
- Need to settle on a small number of data sets that can be correlated easily, say three or four at most. Examples:

    - Cities
    - Population
    - Bus lines, Routes and Stops
- Need an estimate for costs of operating a bus
- Need a simple mapping library that can take as input the data formats chosen
- Without referencing any optimizations we need to be able model and simulate the transit system with the data provided
- With the basic simulation it is then possible to generate key performance indicators(KPIs) or metrics
    - Commute time, total and average per day
    - Number of transfers, total and average per day
    - Running cost for running transit system
- Once we have a simulation of the transit system we can then use our algorithms to optimize various metrics

Getting our Stops:

All Stops, Route 2, Route 6, Route 7

Gather all stops in the area:



Average family size is 2.92, so according to this we should assign low income to be 93,700 or less.

Modeling

We need to build a model representing the system of the area we have chosen.

Potential Model classes with attributes:

- **Line**: Line Code, Line ID
- **Route**: Line, Route ID, Direction (basically a bus driving along roads)
- **Stop**: Stop name, Roads[], Area
- **Road**: Stop 1, Stop 2, Distance

We need to figure out how to make a map of the city itself that these classes will operate in. To do so, I think we can try to mirror the arcgis maps made with the routes.txt files and shape.txt files. Maybe we can try to find a way to even use those arcgis files in our simulation to cut out some work we need to do. But it may be better to write our own anyway so that we have a full understanding of how it operates. Once we have the environment built, we can then start coding in the classes and how they will behave in it.

3 parameters: population, income, and frequency of stops in comparison to others (stops with high frequency do not to be maximized)

March 8th 2023:
Using this tutorial, we have started our simulation.

Algorithm Options

K-Means: K-Means is an unsupervised machine learning algorithm that will identify clusters in a dataset. For our project, we should use K-means in order to find the centers of populations in the area where we are improving public transit. In this case, we would be using density based clustering.
Here are some questions we will need to answer:

1) For a given area, how many centroids do we want?
    a) From this article, I think using the silhouette method will be a good way to assign an appropriate amount of clusters.
    b) These are some other things we can take into consideration when determining the number of centroids and their locations:
        i) We may want to include cost as a variable in our project. The 2021 National Transit Database has lots of data on operating cost, and this pdf has more information about it. Perhaps if we can come up with a budget for the number of bus stops, we can then have our algorithm work with that budget to put as many centroids as allowed by cost.
        ii) Distance between stops is another important aspect. If there are clusters far apart from one another, people in between the clusters will have no way of accessing them without bus stops in between. One way we can handle this

is by simply adding bus stops in between at distance intervals between the two clusters to make a line.

2) How do we deal with existing bus stops?
    c) Does it even make sense to be adding in new bus stops? Perhaps it would be better to use the centroids to figure out where the population centers are, and then with the existing bus stops, focus on optimizing routes between them instead to connect as many of the clusters as possible in the most efficient way.
    d) If there are centroids that are not covered by the existing bus stops, maybe we can add them by altering or adding routes in the most efficient way.

Simulated Annealing: Simulated annealing is a machine learning algorithm that optimizes a few parameters. Assuming we decide which parameters (time, cost, etc) that we want to optimize, we may be able to run our environment through this algorithm to figure out the best way to do so. This website has step by step instructions on writing this from scratch, but we could also use the one from the linked git-hub.

- This could be an alternative to the type of genetic algorithm used in the paper mentioned above. It will be much simpler to use and might be better suited for this project.

It may be worthwhile to look into other algorithms as well.

Available Data

Data Files in the DS Project Folder (Sophia's Laptop):

transit_routes_datasd.geojson: json file of the different routes in sdmts

GTFS (Jan 29, 2023 Download): .txt files, important files: routes, stop times, stops, trips, shapes
Note: also contains: agency, lots of fare data, rider categories

- routes.txt fields: route_id,route_short_name,route_long_name,route_type, agency_id,route_desc,route_url,route_color,route_text_color,route_pattern1, route_pattern2,network_id
    - Link: arcgis representation of routes.txt and stops.txt

- stop_times.txt fields: trip_id,arrival_time,departure_time,stop_id,stop_sequence, timepoint,shape_dist_traveled,position_in_block,stop_headsign,pickup_type, drop_off_type,stop_headsign_short,stop_is_last
- stops.txt fields: stop_id,stop_name,stop_lat,stop_lon,stop_code,location_type, parent_station,wheelchair_boarding,intersection_code,stop_place,reference
- _place, stop_name_short,stop_url
    - Link; this is the data but on a gis map, that can be downloaded to excel
- shapes.txt fields: shape_id,shape_pt_lat,shape_pt_lon,shape_pt_sequence, shape_dist_traveled :: I don't fully understand this file
    - Link: this is an arc gis made with routes.txt and shapes.txt to show all the San Diego routes; it is likely something we will have to do for our project; can we piggyback off this somehow?
- transfers.txt fields: from_stop_id,to_stop_id,transfer_type,min_transfer_time, service_id
- trips.txt fields: route_id,service_id,trip_id,trip_headsign,direction_id,direction_name, block_id,shape_id,wheelchair_accessible,trip_bikes_allowed,trip_headsign_short'

These files will be critical in making our model.

2021 National Transit Database: Dataset containing excel files of operating expenses, vehicle counts, capital use, stations, facilities, revenue, basically any kind of transit data.

I have submitted a request for an API from this website, we'll see how that goes. To be clear: I don't really know what that means or what exactly I have requested.

Google Sheet Data Explained

Statistical Atlas: Represented in the first page of the data sheet. This will likely be the most useful website in terms of amount of data. This first page has the top 50 most populated neighborhoods in San Diego with associated income data, and is almost perfect for our needs. The problem is extracting the data, it is quite a pain. However, if we hone in on just one area of the city that we want to improve public transit for, it shouldn't be too difficult to manually extract the data. We can get as granular as block groups, or widen to neighborhoods.

2020 SANDAG Population Estimates: The second page of the data sheet has the population for 123 zip codes of San Diego. I am unsure about if San Diego has 123 zip codes exactly, some sources say between 70-80, others have up to 200 listed. But here, we have 123. Again, assuming we hone in on one area, this could prove useful.
Income By Zip: Here we have income by zip code, which has similar issues as above in terms of number of zip codes, but it could prove useful.

Based on these different sources, I think using statistical atlas and manually extracting the data for the area we choose is our best bet. It seems to be the most granular data and have the most relevant information for our uses.

Resources we Should Look Into:

Software for Creating APIS: A website with lots of open source platforms that may be useful to us. Here are the ones that stand out to me:

- Linked Connections: An open-source, scalable intermodal route planning engine, which allows clients to execute the route planning algorithm (as opposed to the server). Uses GTFS data.

OneBusAway: A Java app that consumes GTFS and GTFS-Realtime (along with other formats) and turns them into an easy to use REST API.

  - This seems like it can help us plot our route data, or provide us with route data? The UI isn't great and we should spend time going through this website.

- OpenTripPlanner: An open source platform for multi-modal and multi-agency journey planning, as well as returning information about a multi-modal graph (using data sources such as GTFS and OpenStreetMap).
- TransitClock - Java application that can consume raw vehicle positions and generate prediction times in formats such as GTFS-realtime. Formerly known as "Transitime ".

How to Use GTFS Data: This website details how to use the GTFS files in order to model the transit system. It would be beneficial for all of us to read it. It also seems like this website has an abundance of other articles related to working with GTFS Data. We can also look at this website, it seems to have some more information about doing this.

Githubs I am in Love with:

Gtfs Router

Tidytransit + its functions

gtfstools

Gtfs functions