Railroad Network Mapping and Analysis
Progress Report
CSE 519 – Data Science Fundamentals

Team Members:

Abhinav Jain (111495982) abjain@cs.stonybrook.edu
Chaitanya Kalantri (111446728) ckalantri@cs.stonybrook.edu
Jitendra Savanur (111491676) jsavanur@cs.stonybrook.edu

Tasks Completed:

1. **Extraction of data from kml file:**
   The data was present in the kml file. So we extracted all the fields and created a .csv file of the same. Some of the extracted fields are: OBJECTID, Shape_Leng, MILES, FRFRANODE, TOFRANODE and coordinates.

   Extract the coordinates:
   Each <Placemark> tag in the kml file denotes a segment on the railroad network. And each row in the .csv is single <Placemark> tag from the kml file. In other words, each row in the .csv file denotes a segment in the network. Each row in the.csv now contains the coordinates of all the points on that segment. We used the starting and ending coordinates of each segment for the purpose of graph creation. And the intermediate points were used when plotting the segment on the map.

2. **Construction of Graph:**
   The next step is to build a graph from the given segments. We have considered the starting and ending coordinates as the start and end nodes of an edge in the graph. The edge between two nodes will have the length of the segment(in miles) as its weight. Since more than one segment can lie between two nodes, we created a MultiGraph instead of a normal one. We used Networkx library for graph creation and related functionalities like shortest path etc.

3. **Visualizing The Complete Railway Network on the USA map:**
   From user's perspective it's easy to visualize the graph, rather than understand the path in terms of coordinates. Hence, we have used Folium library which plots the coordinates on the graph. And the joined segments could be viewed in different colors to make the path more attractive. Folium library helps to zoom in and out of the map as per the user's requirement.

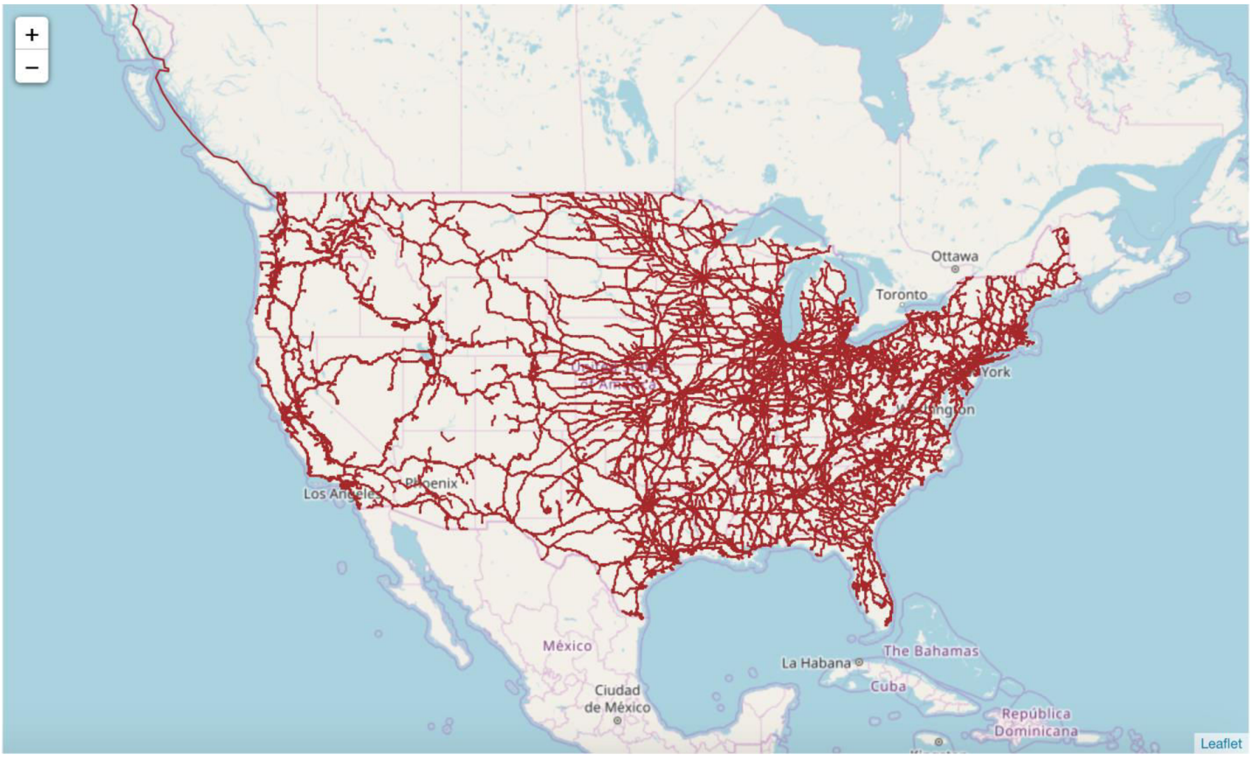


Figure 3.1: Complete Railway Network of United States of America

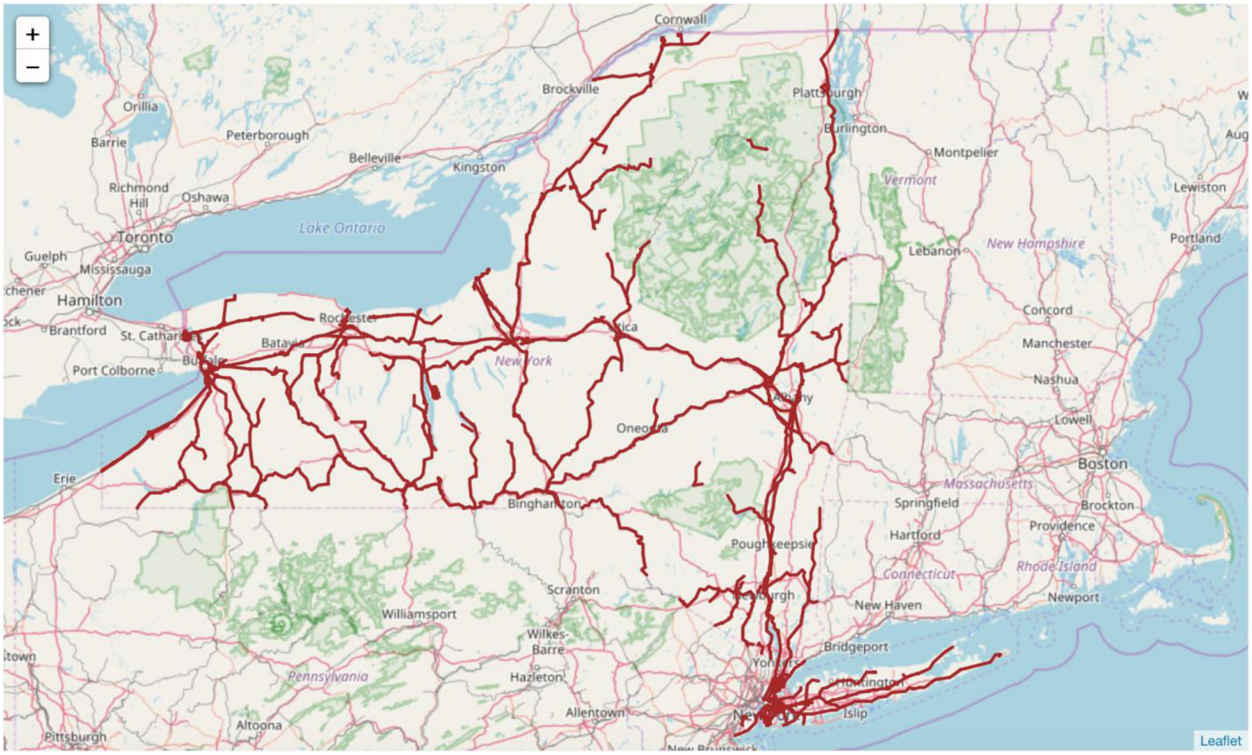
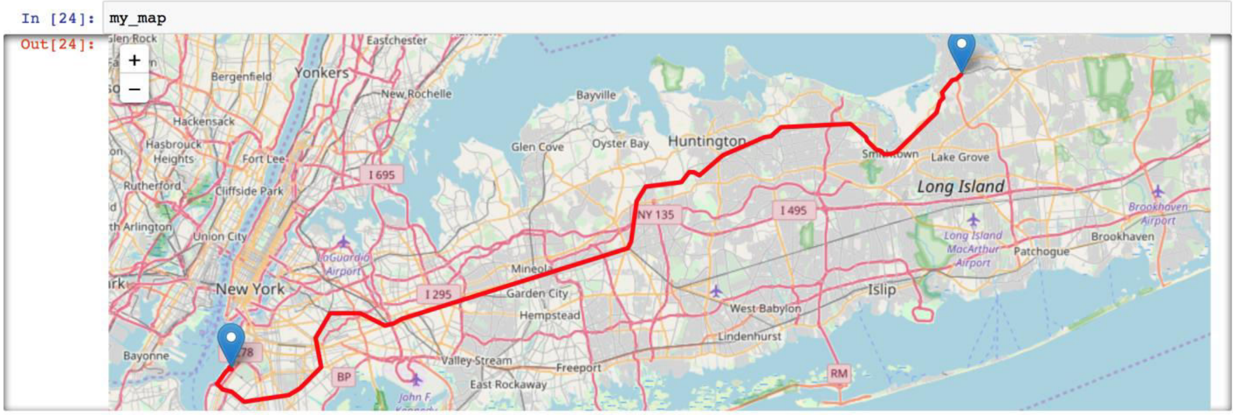


Figure 3.2: Complete Railway Network of New York

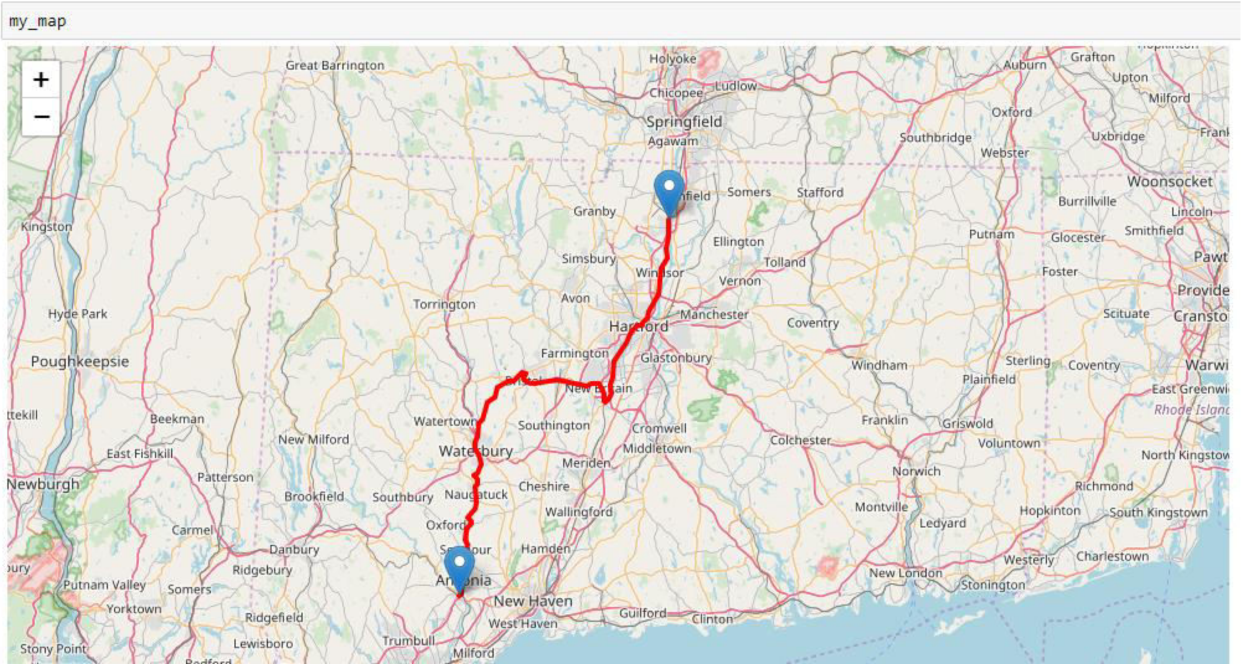
4. Shortest Path Between Source and Destination:

Now that we have all the starting, end and the intermediate coordinates of all the records. We can join the segment to form the shortest path. In order to get the shortest distance in the weighed undirected graph, we have applied Dijkstra's shortest path function available in Networkx. Hence, given source and destination, we are able to compute the shortest distance the train can take from the source to destination.



```
In [24]: my_map
Out[24]:
```

Figure 4.1: Shortest Path (in red) between East Setauket, NY to 39 St/2 Av, Brooklyn, NY



```
my_map
```

Figure 4.2: Shortest Path (in red) between Windsor Locks, CT to Shelton, CT

5. **Embed The Crossing Data On The Railroad segments:**
There is no direct mapping of the datasets for Railroad lines and crossings. Therefore, the strategy we used to embed rail crossings on the railroad segments was to compute the haversine distances of each crossing with every segment in the railroad. The crossing that was at the least distance from a segment was considered to be located on that segment.
In order to minimize the search space for the segment on which a crossing is located, we made use of the state code. We considered the segments belonging to that state only to which the crossing belongs, thus reducing the number of times the loop runs.

Haversine Function:
We used haversine library in order to find the magnitude of the distance given source and destination latitude and longitude.

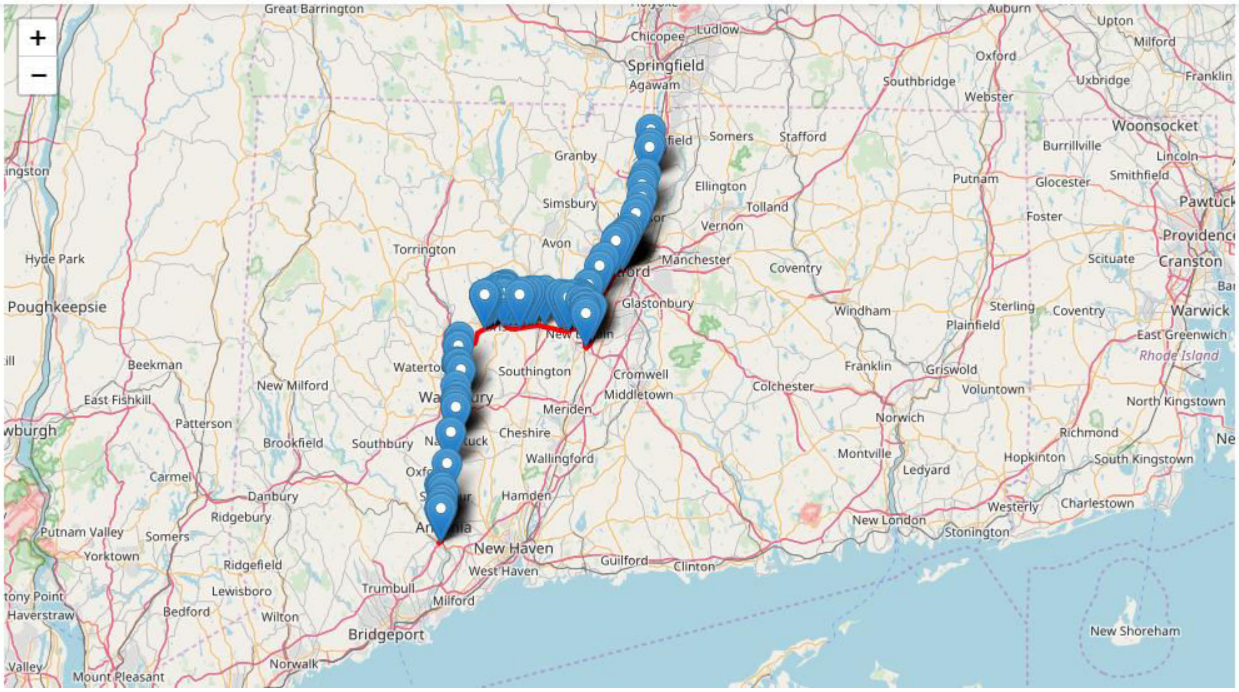


Figure 5.1: Number of crossings between Windsor Locks, CT to Shelton, CT

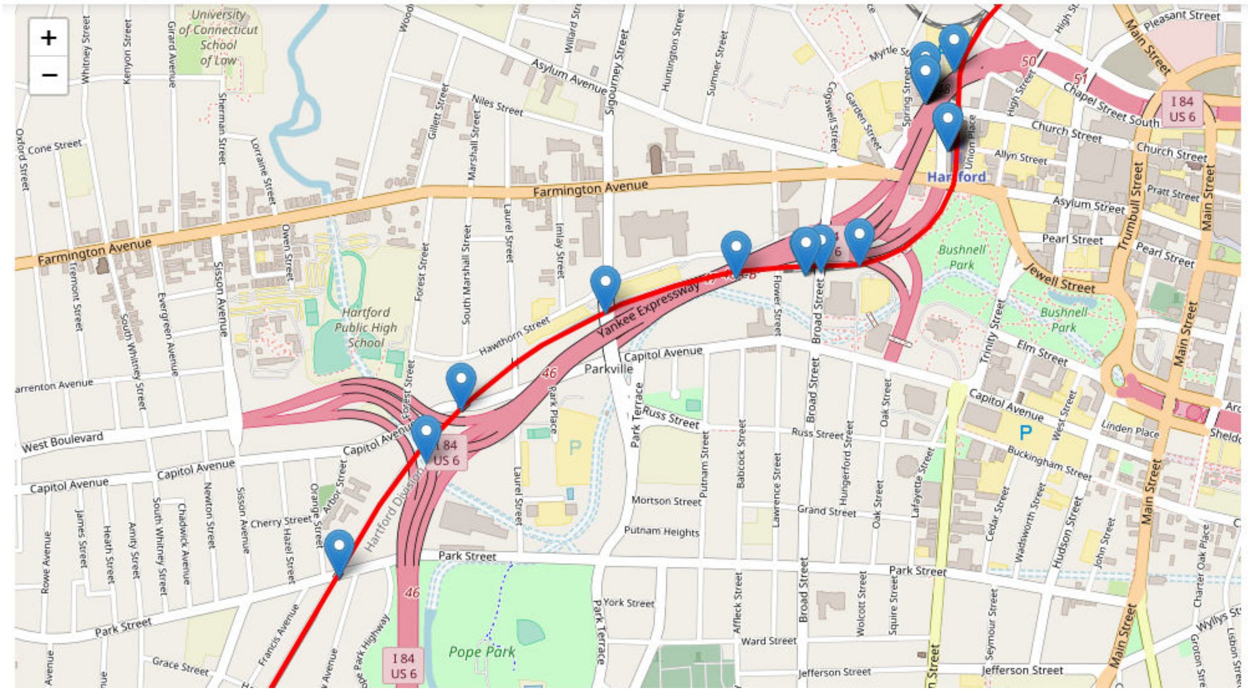


Figure 5.2: Zoomed version of Number of crossings between Windsor Locks, CT to Shelton, CT

6. Plot The Graph Between "Number of Crossing" Vs "Segments":

We are trying to predict the total number of crossing a particular segment consists of. After finding the segments in the shortest path for a source-destination pair, we use the embedded crossings data to get the number of crossings on each segment on the shortest path and plot them on a point plot.

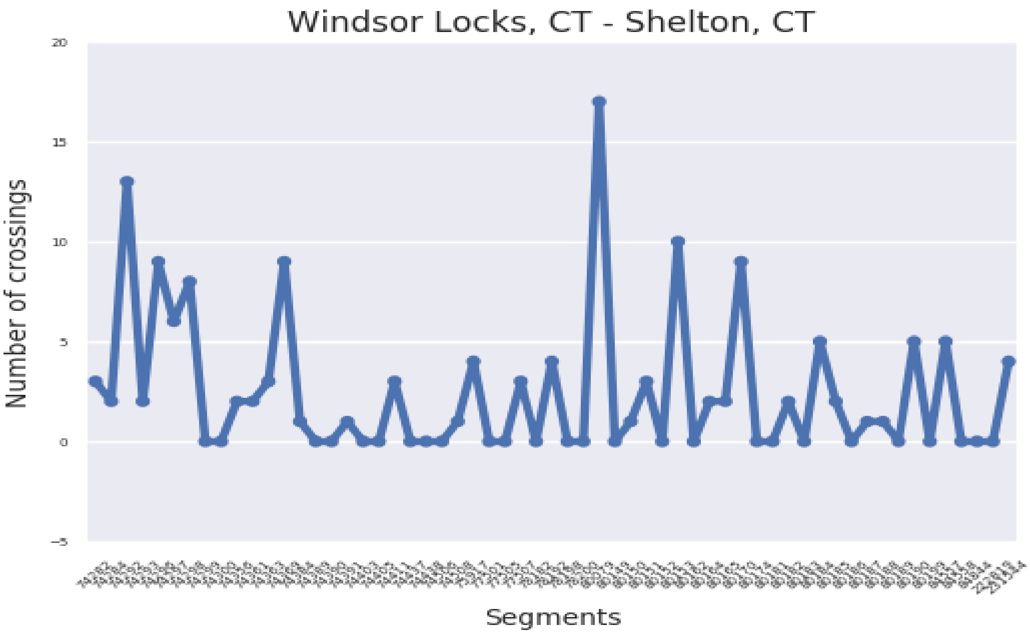


Figure 6.1: Graph between "Number of crossings" Vs "Segments" of Windsor Locks, CT to Shelton, CT

Interesting Observations:

1. During graph constrution, there were node-pairs which had more than one railway track between them. In order to accommodate all such possible paths, we made use of multigraph. And when the node-pair with multiple edges was found in the shortest path, we considered the edge with the least weight i.e. the track with shortest distance between the node-pair.

2. The Folium library internally plots maps which has the railroad network plotted. This feature of folium could be used to determine whether the path which is given by our approach is a valid one.

Issues faced:

1. Some of the crossings where quite far away from the railway tracks, which were discovered when we plotted the crossings on the shortest path.
2. Embedding crossings for the complete US railroad network is computationally very heavy. Therefore, for this phase we considered the crossings embedded for the railroad network for the states of Connecticut and New York

References

[1] https://networkx.github.io/
[2] https://github.com/python-visualization/folium
[3] https://pypi.python.org/pypi/haversine