MODELING TRAFFIC CONDITIONS TO DETERMINE SHORTEST PATH

NOAH DAHLE and KRISTINA WILSON*
PROJECT ADVISOR: DR. OLIVIA PROSPER†

Abstract. It is common for a mapping platform to relay information to the user about the traffic in the area. Traffic conditions have different degrees of traffic intensity that are represented by colors. We have studied Google Maps traffic data through a simulation to model these conditions. Our methodology explores the prospect of how different traffic condition colors will imply different travel times. This modeling was a means to develop an user interface to determine the fastest path between two points in a particular area of Knoxville, Tennessee. We accomplished this by using Dijkstra's Algorithm. The final product is a program that accepts and corrects user input for day of week, time, and two points, and the output is the path the user should take, along with the travel time according to the Google Maps traffic conditions at that day and time. This model can be applied to different cities to study their traffic patterns or in city planning.
Keywords. Dijkstra's Algorithm, mapping platform, graph theory, data fitting, travel time

1 Introduction

In recent years, web mapping platforms such as Google Maps have become popular for users to determine travel routes, due to the platforms' convenience and reliability. These platforms use a variety of resources to determine the shortest travel route, such as using tools to identify traffic patterns in the area. This service allows the platform to offer the most time-efficient or gas-efficient route to the user by measuring and weighing the traffic in the area.

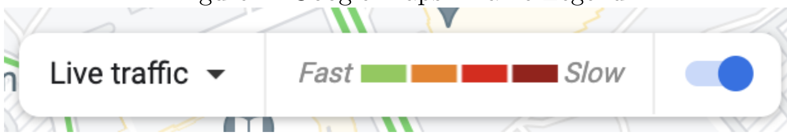
Other studies have set the foundation for using a GPS as a data source. Rito, Lopez, and Biona collected data from a GPS to model the amount of emissions produced in a certain area [7]. They used traffic conditions to measure the amount of traffic a road in the Philippines experienced and street view to see what kind of vehicles were driving on the road to estimate the emissions produced per day. Friaswanto, Lisangan, and Sumarta used GPS and Dijkstra's algorithm in collaboration to help firemen find the quickest path to a fire [6]. They optimized the time it takes firemen to extinguish a fire. Khatri modeled route preferences for bikers on a University of Arizona campus [5]. He used a GPS to track the riders around campus and then modeled their route patterns. Building onto this foundation, we can study data collection from a GPS and algorithms to analyze traffic data.

Google Maps has a color-coded system to classify the level of the traffic. The color coded traffic legend is seen in Figure 1 [3]. The legend features green, orange, red, and brown keys,

*Department of Mathematics, University of Tennessee at Knoxville, TN 37996 (zrd939@vols.utk.edu, kwils119@vols.utk.edu)

†Department of Mathematics, University of Tennessee at Knoxville, TN 37996 (oprosper@utk.edu)

Figure 1: Google Maps' Traffic Legend



in order of increasing traffic intensity. That is, green implies quick traffic and brown implies very slow traffic.

Google has a traffic simulation that predicts the traffic conditions for an area at a specific time for a typical day of the week. For example, this simulation could answer a question such as "How long will it take me to get home from work at 5:00pm on a **typical** Friday?" An example of the simulation controls is shown in Figure 2 [4]. The user is able to choose a day of the week and a time to analyze. The color-coded legend demonstrates the volume of traffic on a particular street. This simulation does not represent the traffic on a specific day – such as June 2, 2022 – but the typical traffic on an average Thursday.

We study the area outlined by the map in Figure 3. Our goal is to create a model that can accurately predict travel times in this map. We hypothesized that travel time changes depending on the traffic condition (green, orange, red, brown traffic) in the map. We collected data for green, orange, red, and brown traffic conditions, developed models based on this data, and used these models to predict travels times in the map. We precisely define the term travel time, along with other terms used, in Table 1.

1.1 Outline

Table 1 provides terms and definitions used throughout the paper. Section 2 provides a description of the methods we used to collect data. Section 2.1 details the data collection from Google, section 2.2 details the model selection for each traffic condition (i.e., green, orange, red, and brown), and section 2.3 supplies pseudo code for algorithms we use in the project. Section 3 provides documentation of how well our models and algorithms perform.

2 Method

We make use of three main techniques in our project: data collection using Jupyter Notebook and Google Maps, model derivation in Microsoft Excel, and C++ programming to

Figure 2: Example of Google Maps Traffic Simulation

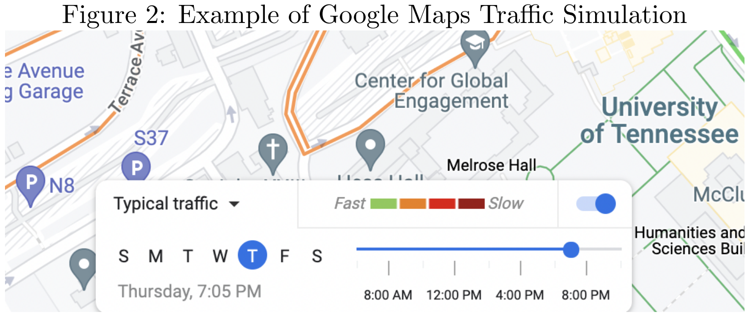
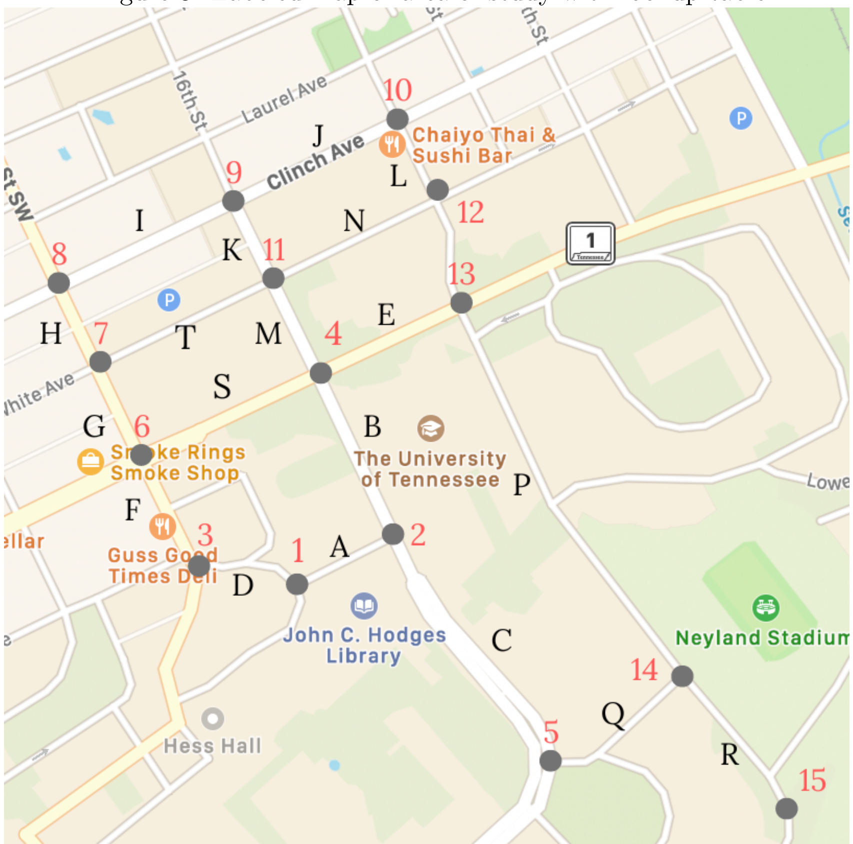


Figure 3: Labeled map of area of study with lookup table



Label	Street Name	Intersections	Nodes
A	Melrose Ave	Melrose Pl to Volunteer Blvd	1, 2
B	Volunteer Blvd	Melrose Ave to Cumberland Ave	2, 4
C	Volunteer Blvd	Melrose Ave to Peyton Manning Pass	2, 5
D	Melrose Pl	Melrose Ave to Melrose Pl	1, 3
E	Cumberland Ave	16th St to James Agee St	4, 13
F	Melrose Pl	Melrose Pl to Cumberland Ave	3, 6
G	17th St	Cumberland Ave to White Ave	6, 7
H	17th St	White Ave to Clinch Ave	7, 8
I	Clinch Ave	17th St to 16th St	8, 9
J	Clinch Ave	16th St to James Agee St	9, 10
K	16th St	Clinch Ave to White Ave	9, 11
L	James Agee St	Clinch Ave to White Ave	10, 12
M	16th St	Cumberland Ave to White Ave	4, 11
N	White Ave	16th St to James Agee St	11, 12
O	James Agee St	White Ave to Cumberland Ave	12, 13
P	Phillip Fulmer Way	Cumberland Ave to Peyton Manning Pass	13, 14
Q	Peyton Manning Pass	Volunteer Blvd to Phillip Fulmer Way	5, 14
R	Phillip Fulmer Way	Peyton Manning Pass to Neyland Stadium	14, 15

Table 1: Terminology

Symbol	Term	Definition	Units
$\mathcal{S}$	Speed Limit	The speed limit on a particular street; used to calculate ideal time	ft/s, mph
$\mathcal{D}$	Distance	The distance between two nodes	ft
$\mathcal{I}$	Ideal (Travel) Time	The travel time between two nodes, assuming no traffic	s
$\mathcal{A}$	Actual (Travel) Time	The measured travel time between two nodes, including traffic	s
$\mathcal{F}$	Difference Time	$\mathcal{A} - \mathcal{I}$	s

compute the shortest path using Dijkstra's algorithm.

2.1 Data Collection

We make use of Google Maps' traffic simulation to collect our data. Google has four key colors for identifying traffic conditions: **green, orange, red,** and **brown**. If there is no color, we call this condition **N/A**.

First, we analyzed the simulation of a typical week from Sunday to Saturday from 6:00 am to 10:00 pm, moving in 15 minute increments. The simulation analyzes the typical behavior of a typical day of the week, not a specific date. Thus we collected data for an average Sunday, an average Monday, continuing for every day of the week for Sunday through Saturday. Using the labels from our map (Figure 3), we identified the changes for each edge during each day for each time increment (Tables 2, 3, 4, and 5). We reference this data when computing the shortest path at a particular time.

Next, we collected travel times for each condition (green, orange, red, brown) in real time, not using the simulation. We collected actual travel time ($\mathcal{A}$), distance traveled ($\mathcal{D}$), and speed limit ($\mathcal{S}$) of the area. We also calculated the ideal travel time by $\mathcal{I} = \mathcal{S} \times \mathcal{D}$. We compared $\mathcal{A}$ and $\mathcal{I}$ to study how the traffic condition has affected the actual travel time.

We collected about 30 travel times for each color key: 32 for green, 30 for orange, 30 for red, and 30 for brown. We identified 3-5 outliers per color. Copies of this data are found at Tables 2, 3, 4, and 5 respectively. Outliers are in italics.

The map we chose to study (Figure 3) has limited red traffic and no brown traffic, so we used various locations and roads to collect the data for red and brown. For the red traffic condition, points 1-4, 12, and 14-16 derived from the map in Figure 3. We analyzed interstates I-40, I-640, TN-170, TN-62, I-24, I-65, I-440, and TN-254, and streets Kingston Pike, Henley St, and Locust St to collect additional data for the red traffic condition. We collected actual travel time, distance, speed limit, and ideal travel time for these locations. We measured various distances on the same roads. For example, data points 5 and 8 are the same road but different distances, since we collected data at different mile markers on this road.

For the brown traffic condition, we analyzed interstates I-5, US-101, CA-110, I-405, I-10, CA-91, I-24, I-65, I-40, I-69, I-710, I-45, TX-288, I-5, I-278, I-H-1, I-275, I-610, and MacArthur Cwy, and streets Sutherland Ave, Kingston Pike, and Battery Park to collect data. We collected actual travel time, distance, speed limit, and ideal travel time for these locations. We measured various distances on the same roads. For example, data points 25 and 27 are both on I-610, but they record different distances.

Note: * implies the speed limit varied on the route, so we use the average speed limit as $\mathcal{I}$.

Table 2: Data Collected for Green Traffic Condition

Area	$\mathcal{S}$ (mph)	$\mathcal{D}$	$\mathcal{I}$ (Ideal Time)	$\mathcal{A}$ (Actual Time)
1. Cumberland Ave	*35*	*3168 ft*	*1.0 min*	*4 min*
2. Volunteer Blvd	*25*	*5808 ft*	*2.6 min*	*5 min*
3. Clinch Ave	35	2640 ft	.86	3 min
4. Neyland Dr	45	11088 ft	2.8	4 min
5. 17th St	30	3696 ft	1.5	3 min
6. Highland Ave	35	2640 ft	.9	2 min
7. 22nd St	25	2112 ft	.9	2 min
8. Lake Ave	20	2112 ft	1.2	2 min
9. Phillip Fulmer Way	20	3168 ft	1.8	3 min
10. Sutherland Ave	35	5808 ft	1.9 min	3 min
11. I-140 E	65	70752 ft	12.4 min	15 min
12. I-140 E	65	28512 ft	4.98	6 min
13. I-140 E	65	17424 ft	3.05	4 min
14. US-129 N	55	26928 ft	6.55	7 min
15. US-129 N	55	12144 ft	2.51 min	3 min
16. Cumberland Ave	35	5280 ft	1.714 min	3 min
17. US-441 S	*	7392 ft	2.057 min	3 min
18. Neyland Dr	*	23760 ft	5.785 min	6 min
19. TN-62 E	35	14256 ft	4.63 min	5 min
20. James White Pwky/I-40E	55	27984 ft	5.78 min	6 min
21. Cumberland Ave	35	5280 ft	1.71 min	3 min
22. Volunteer Blvd	25	5280 ft	2.4 min	3 min
23. 17th St	30	3168 ft	1.2 min	2 min
24. US-129 S	55	57024 ft	11.78 min	12 min
25. 22nd St	25	2112 ft	.96 min	2 min
26. 17th St	30	4752 ft	1.8 min	4 min
27. Cumberland Ave	*35*	*7920 ft*	*2.57 min*	*5 min*
28. Clinch Ave	35	4752 ft	1.54 min	4 min
29. Phillip Fulmer Way	20	3168 ft	1.8 min	2 min
30. Cumberland Ave	*35*	*7920 ft*	*2.57 min*	*5 min*
31. Cumberland Ave	*35*	*11616 ft*	*3.77 min*	*8 min*
32. 17th St	30	3696 ft	1.4 min	3 min

Table 3: Data Collected for Orange Traffic Condition

Area	$\mathcal{S}$ (mph)	$\mathcal{D}$	$\mathcal{I}$ (Ideal)	$\mathcal{A}$ (Actual)
1. Cumberland Ave	35	3686 ft	1.2 min	5 min
2. 17th St	30	1584 ft	0.6 min	3 min
3. White Ave	25	4752 ft	2.6 min	7 min
4. Volunteer Blvd	25	5280 ft	2.4 min	5 min
5. Lake Ave	20	2640 ft	1.5 min	3 min
6. Clinch Ave	35	3168 ft	1.0 min	3 min
7. 16th St	35	2112 ft	0.7 min	3 min
8. Joe Johnson Dr	20	2640 ft	1.5 min	3 min
9. James Agee St	20	2112 ft	1.5 min	3 min
10. 21st Steet	25	2112 ft	.96 min	3 min
11. Henley Street	35	4224 ft	1.371 min	3 min
12. Cumberland Ave	*35*	*4752 ft*	*1.54 min*	*6 min*
13. Pellissippi Pwky	65	18480 ft	3.231 min	6 min
14. Kingston Pike	*40*	*7392 ft*	*2.1 min*	*7 min*
15. Sutherland Ave	35	5280 ft	1.71 min	4 min
16. I-40 E	65	24288 ft	4.246 min	8 min
17. I-40 N	65	79100 ft	13.829 min	24 min
18. Kingston Pike	*40*	*7392 ft*	*2.1 min*	*6 min*
19. University Commons Way	25	2112 ft	.96 min	3 min
20. 16th St	35	2112 ft	.686 min	2 min
21. White Ave	25	3168 ft	0.9 min	4 min
22. Cumberland Ave	*35*	*4224 ft*	*1.37 min*	*6 min*
23. Lake Ave	20	2640 ft	1.2 min	3 min
24. Joe Johnson Dr	20	2112 ft	.96 min	2 min
25. Cumberland Ave	*35*	*5808 ft*	*1.886 min*	*8 min*
26. 17th St	*30*	*2640 ft*	*1 min*	*5 min*
27. 22nd St	25	2640 ft	1.2 min	5 min
28. White Ave	25	2112 ft	.96 min	3 min
29. Phillip Fulmer Way	20	1584 ft	.9 min	2 min
30. Grand Ave	25	2112 ft	.96 min	2 min

Table 4: Data Collected for Red Traffic Condition

Area	$\mathcal{S}$ (mph)	$\mathcal{D}$	$\mathcal{I}$ (Ideal)	$\mathcal{A}$ (Actual)
1. 18th St	35	528 ft	.17 min	2 min
2. Laurel Ave E	20	1056 ft	.6 min	2 min
3. Laurel Ave W	20	1056 ft	.17 min	2 min
4. White Ave	25	1056 ft	.48 min	2 min
5. I-40 W	65	13728 ft	2.4 min	9 min
6. I-40 E	65	3168 ft	.55 min	2 min
7. I-640 W	65	2640 ft	.46 min	2 min
8. I-40W	65	6864 ft	1.2 min	4 min
9. Kingston Pike	35	3696 ft	1.2 min	3 min
10. I-40E	65	23760 ft	4.15 min	12 min
11. I-40W	65	17424 ft	3.05 min	13 min
12. Grand Ave	*25*	*2112 ft*	*.96 min*	*2 min*
13. I-40E	65	1584 ft	.2769 min	1 min
14. 21st St	25	528 ft	.24 min	1 min
15. College St	25	328 ft	.1491 min	1 min
16. 22nd St	25	528 ft	.24 min	2 min
17. 1-40 W	65	6864 ft	1.2 min	5 min
18. Henley St	35	2112 ft	.686 min	3 min
19. Locust St	25	528 ft	.24 min	2 min
20. I-40W	65	3168 ft	.5538 min	2 min
21. I-40E	65	36960 ft	6.4615 min	18 min
22. I-40W	65	8448 ft	1.4769 min	6 min
23. TN-170	55	5280 ft	1.0909 min	4 min
24. TN-62	*45*	*8448 ft*	*2.133 min*	*6 min*
25. I-24E	*70*	*28512 ft*	*4.629 min*	*19 min*
26. I-65N	55	7392 ft	1.5273 min	6 min
27. I-440	*	6336 ft	1.433 min	7 min
28. I-24E	55	31152 ft	6.436 min	21 min
29. I-24W	55	5280 ft	1.0909 min	3 min
30. TN-254	45	1584 ft	.4 min	2 min

Table 5: Data Collected for Brown Traffic Condition

Area	$\mathcal{S}$ (mph)	$\mathcal{D}$	$\mathcal{I}$ (Ideal)	$\mathcal{A}$ (Actual)
1. I-5 S in L.A.	55	30624 ft	6.33 min	45 min
2. US-101 S	55	8448 ft	1.75 min	12 min
3. CA-110	55	2640 ft	.545 min	4 min
4. I-405S	55	16368 ft	3.38 min	27 min
5. I-10 E	55	7392 ft	1.53 min	11 min
6. CA-91 E	*65*	*13728 ft*	*2.4 min*	*25 min*
7. Sutherland Ave	35	377 ft	.1224 min	1 min
8. Kingston Pike	45	384 ft	.09697 min	1 min
9. Kingston Pike	45	367 ft	.0927 min	1 min
10. I-24W	55	3696 ft	.7636 min	5 min
11. I-65N	55	1056 ft	.2182 min	2 min
12. I-40E	55	2112 ft	.4364 min	3 min
13. I-69	75	4224 ft	.64 min	6 min
14. I-69	75	6864 ft	1.04 min	11 min
15. I-710 N	*65*	*6864 ft*	*1.2 min*	*5 min*
16. I-45N	75	3168 ft	.48 min	5 min
17. TX-288 N	55	3696 ft	.7636 min	6 min
18. I-5 S	55	26928 ft	5.5636 min	30 min
19. I-278W	50	6336 ft	1.44 min	11 min
20. Battery Park	25	3168 ft	1.44 min	8 min
21. I-45N	75	3168 ft	.48 min	5 min
22. I-69N	75	213 ft	.0323 min	1 min
23. I-10E	45	2640 ft	.6667 min	4 min
24. I-H-1E	*55*	*9504 ft*	*1.9636 min*	*23 min*
25. I-275N	55	11088 ft	2.2909 min	15 min
26. I-610E	60	3696 ft	.7 min	5 min
27. I-610N	60	2640 ft	.5 min	4 min
28. I-610E	60	528 ft	.1 min	1 min
29. I-610W	60	528 ft	.1 min	1 min
30. MacArthur Cwy	45	7920 ft	2 min	13 min

2.2 Model Process and Selection

For each of the four traffic conditions (green, orange, red, brown) we considered three different models: a quadratic model, a linear model, and an exponential model. Each model was fit to the data for green, orange, red, and brown traffic conditions in order to estimate their parameters.
In our process we made the following assumptions:

1. The N/A condition will have no effect on the actual travel time.
2. Ideal travel time = speed limit * distance. This is the minimum travel time possible.
3. The data from the Google simulation is accurate.
4. Traffic conditions in the Knoxville area change around every 15 minutes (thus we collected data from the Google simulation in 15 minute intervals)

We estimated parameters for the following equations.
$\mathcal{A}_A = a_2 I^2 + a_1 I + a_0$
$\mathcal{A}_B = b_1 I + b_0$
$\mathcal{A}_C = c_0 e^{c_1 I}$
We estimated parameters considering and not considering the outliers of each of the four traffic conditions' data sets. Our estimations for considering outliers and not considering outliers are found in Figures 4 and 5 respectively.

Figure 4: Estimated parameters for each model, with outliers included

	Quadratic	**Linear**	**Exponential**
Green	$a_0 = 1.877$ $a_1 = .7259$ $a_2 = .0191$	$b_0 = 1.492$ $b_1 = .9577$	$c_0 = 2.3648$ $c_1 = .1582$
Orange	$a_0 = 1.626$ $a_1 = 1.7786$ $a_2 = -.012$	$b_0 = 1.8366$ $b_1 = 1.6104$	$c_0 = 2.9632$ $c_1 = .173$
Red	$a_0 = .2412$ $a_1 = 3.8488$ $a_2 = -.1199$	$b_0 = .7191$ $b_1 = 3.1197$	$c_0 = 1.8277$ $c_1 = .4525$
Brown	$a_0 = .0873$ $a_1 = 8.4883$ $a_2 = -.3241$	$b_0 = 1.0232$ $b_1 = 6.6632$	$c_0 = 2.5026$ $c_1 = .6131$

Figure 5: Estimated parameters for each model, with outliers removed

	Quadratic	**Linear**	**Exponential**
Green	$a_0 = 2.0385$ $a_1 = .5133$ $a_2 = .0363$	$b_0 = 1.3312$ $b_1 = .9541$	$c_0 = 2.2626$ $c_1 = .1579$
Orange	$a_0 = 1.4881$ $a_1 = 1.5105$ $a_2 = .0085$	$b_0 = 1.3444$ $b_1 = 1.6301$	$c_0 = 2.618$ $c_1 = .1775$
Red	$a_0 = .459$ $a_1 = 3.6857$ $a_2 = -.1159$	$b_0 = .9105$ $b_1 = 2.9636$	$c_0 = 1.8561$ $c_1 = .4382$
Brown	$a_0 = .6252$ $a_1 = 6.8238$ $a_2 = -.0706$	$b_0 = .8515$ $b_1 = 6.4177$	$c_0 = 2.418$ $c_1 = .5868$

Graphs of data with the models with and without outliers are shown in figures 6 and 7 respectively.

Model Selection.

We evaluated how well each of the three models fit the traffic data using the Akaike information criterion (AIC). The AIC is an equation that is calculated from the number of parameters from the model, the number of data points that the model fits, and the residual sum of squares (RSS) of the fitted data [1] [8] [2]. The equation we used is $AIC = 2k + n \ln RSS$, for k is the number of parameters of the model and n is the number of data points modelled [1]. The RSS is calculated by summing the square of the difference of each projected result from the model and the actual value. AIC is a helpful tool to evaluate the models we considered, as a lower AIC implies a better fit of the data [1]. The calculations are as follows in Figure 8.

Figure 6: Graph of Collected Data with Each Models' Fit

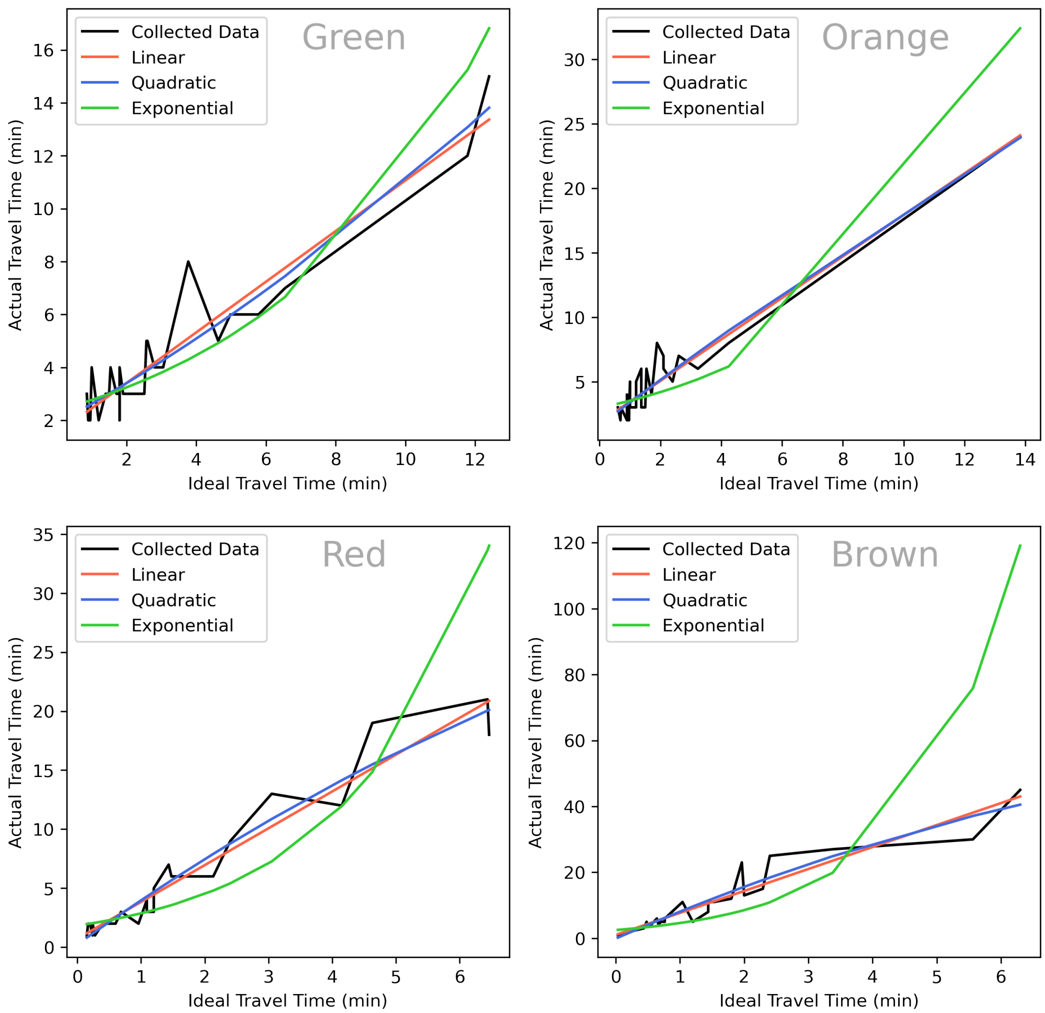


Figure 7: Graph of Collected Data (Removing Outliers) with Each Models' Fit

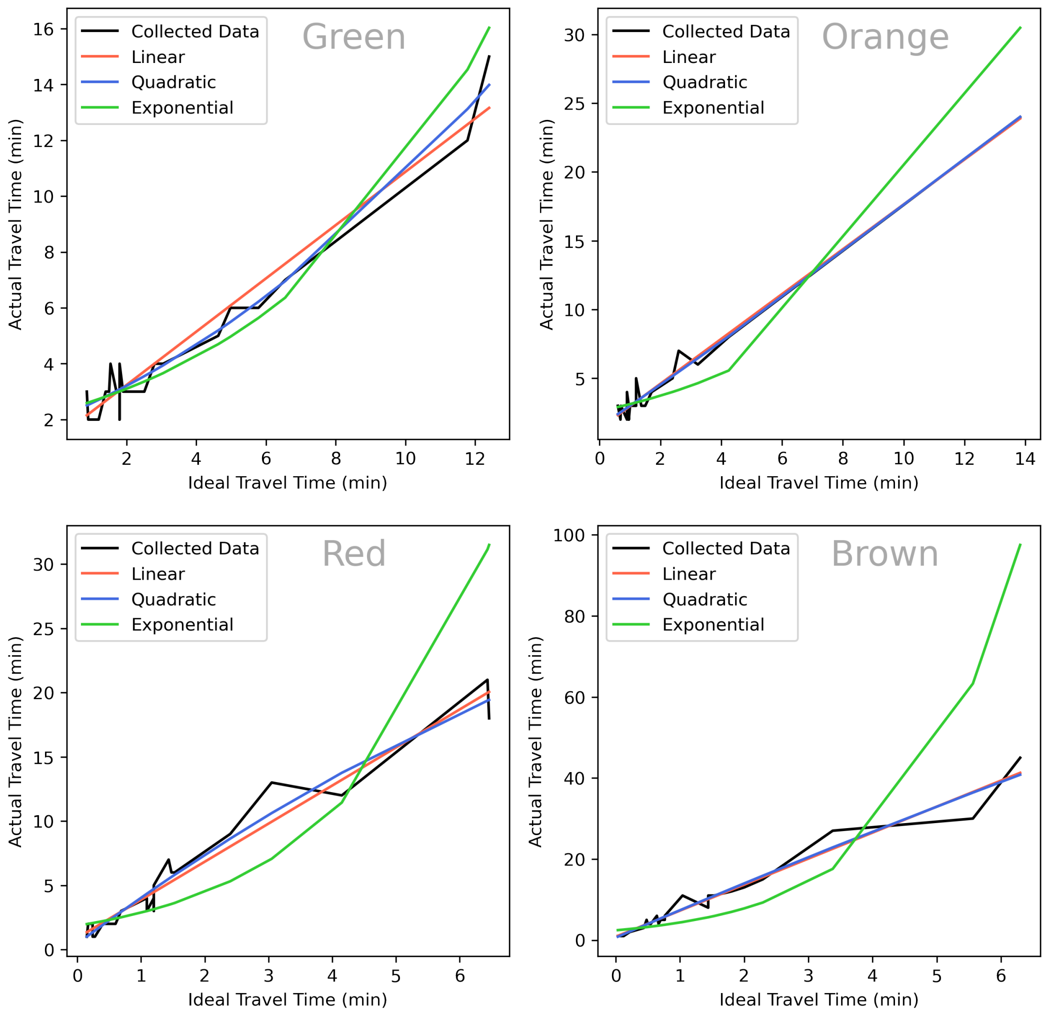


Figure 8: Calculations of RSS and AIC for each model

Green	**RSS**	**AIC**
Quadratic	26.7341	0.2465
Linear	27.8466	-.4488
Exponential	43.0648	13.5030
Green- Removing outliers		
Quadratic	8.7252	-24.4998
Linear	12.0470	-17.7895
Exponential	15.5798	-10.8462
Orange	**RSS**	**AIC**
Quadratic	42.6591	16.5613
Linear	42.9438	14.7608
Exponential	134.7919	49.0760
Orange- Removing outliers		
Quadratic	15.1792	-4.9951
Linear	15.3095	-6.7899
Exponential	71.3681	30.1551
Red	**RSS**	**AIC**
Quadratic	45.7485	18.6589
Linear	49.8393	19.2282
Exponential	514.0603	89.2343
Red- Removing Outliers		
Quadratic	25.0076	3.9302
Linear	28.3898	5.3552
Exponential	367.7838	74.5148
Brown	**RSS**	**AIC**
Quadratic	245.7527	69.0939
Linear	271.1134	70.0402
Exponential	8239.4403	172.4647
Brown- Removing Outliers		
Quadratic	101.5087	41.7563
Linear	102.3210	39.9715
Exponential	4181.9920	140.1531

2.3 Algorithms

In our project, we use Dijkstra's Algorithm to determine the shortest path from one location to all other locations. We also use a Jupyter Notebook that contains the user interface for our project. Figure 9 is a flowchart that illustrates how the algorithms are connected.

Algorithm. *Dijkstra's Algorithm (G, s)*

Given a graph G and a starting node s, Dijkstra's Algorithm finds the shortest path from the starting node s to every possible node in the graph.

In the Graph G, **for** each node n, set distance(n,s) $= -1$ and previous node b to NULL.

If $n = s$, set distance(n,s) $= 0$, and place n into the priority queue M ascending by distance.

While M is **not** empty, let the first node be m. Assign m into a temporary variable f. Then, remove the node from M.

Loop through f's list of edges E, and calculate the new potential distance p of the new node m by applying this formula for each edge e: $p = \text{distance}(f, s) + \text{weight}[e]$.

Let d = distance(m, s). **If** $d = -1$ or $p < d$, check if m is already in M. If it is, remove it, and set distance(m, s) $= p$ and its previous node b to f. Add m back to M.

Algorithm. *Depth First Search (G, s)*

Given a graph G and a starting node s, Depth First Search determines if there is a possible path from the starting node s to the ending node e.

In the Graph G, **for** each node n, set its visited field to **false**.

If node s's visited field is **true**, **return**.

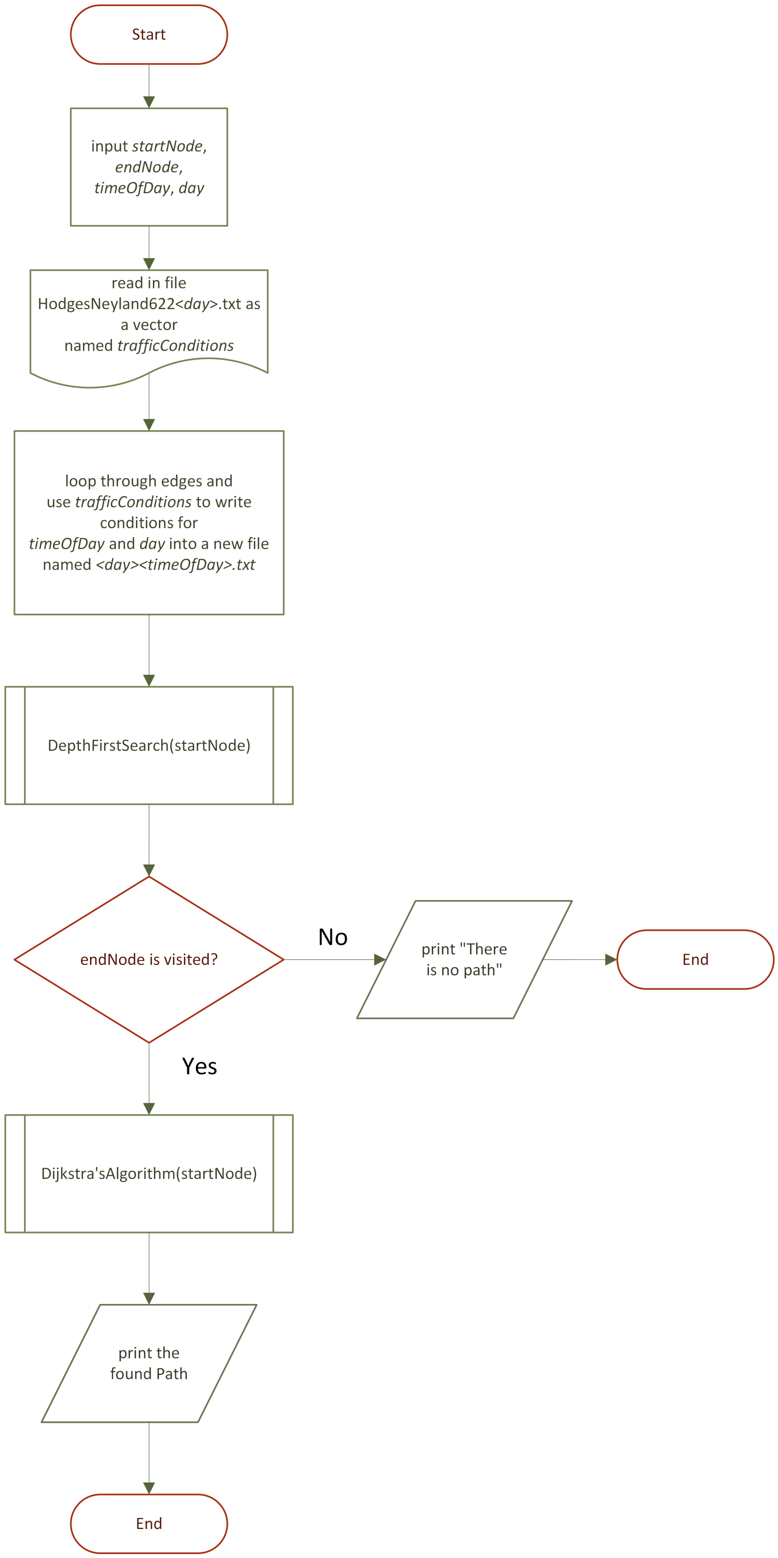
If not, set s's visited field to **true**.

Loop through s's edge list E. **For** each edge e, call Depth First Search on the node m it leads to: *Depth First Search* (G, m).

After the algorithm ends, check e's visited field.

If it is **true**, **return true**
If not, **return false**

Figure 9: Flowchart of Program



3 Results

We have concluded that the quadratic model has the lowest AIC overall, so it has the best fit of the models evaluated. We developed a jupyter notebook program that uses this model to determine the shortest path between 2 points on our map from figure 3.

The program asks if the user wants to use the current date and time or use a different date and time.

```
shortestdistance()

The current time and day rounded to the closest 15 minute mark is Sunday at 1515

If you want a different time and day type yes. If not, type no: maybe

Inalid option. Type yes or no: |
```

If the user selects yes, the user will be prompted to enter the time and day they would like to analyze.

```
shortestdistance()

The current time and day rounded to the closest 15 minute mark is Sunday at 1515

If you want a different time and day type yes. If not, type no: yes
Enter the time of day in military time (between 600 and 2200): 1300
Enter the day: mon
Enter the starting location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: canes
Enter the ending location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: volunteer
Path:
17 & Cumberland->Cumberland & Volunteer->16 & White->James & White

Directions:
From 17 & Cumberland take Cumberland Avenue to get to Cumberland & Volunteer
From Cumberland & Volunteer take 16th Street to get to 16 & White
From 16 & White take White Avenue to get to James & White

Total Time: 2:46
```

If the user selects no, then the current time and day will be rounded to the nearest 15 minute interval.

```
shortestdistance()

The current time and day rounded to the closest 15 minute mark is Sunday at 1515

If you want a different time and day type yes. If not, type no: no
Enter the starting location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: hodges
Enter the ending location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: neyland
Path:
Hodges->Melrose & Volunteer->Peyton & Volunteer->James & Peyton->Neyland

Directions:
From Hodges take Melrose Avenue to get to Melrose & Volunteer
From Melrose & Volunteer take Volunteer to get to Peyton & Volunteer
From Peyton & Volunteer take Peyton Manning Pass to get to James & Peyton
From James & Peyton take James Agee to get to Neyland

Total Time: 4:15
```

Then, the user will be prompted to enter the starting and ending locations. The program accepts landmarks or intersections as valid input.

```
shortestdistance()

The current time and day rounded to the closest 15 minute mark is Sunday at 1515

If you want a different time and day type yes. If not, type no: no
Enter the starting location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: hodges
Enter the ending location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: neyland
Path:
Hodges->Melrose & Volunteer->Peyton & Volunteer->James & Peyton->Neyland

Directions:
From Hodges take Melrose Avenue to get to Melrose & Volunteer
From Melrose & Volunteer take Volunteer to get to Peyton & Volunteer
From Peyton & Volunteer take Peyton Manning Pass to get to James & Peyton
From James & Peyton take James Agee to get to Neyland

Total Time: 4:15
```

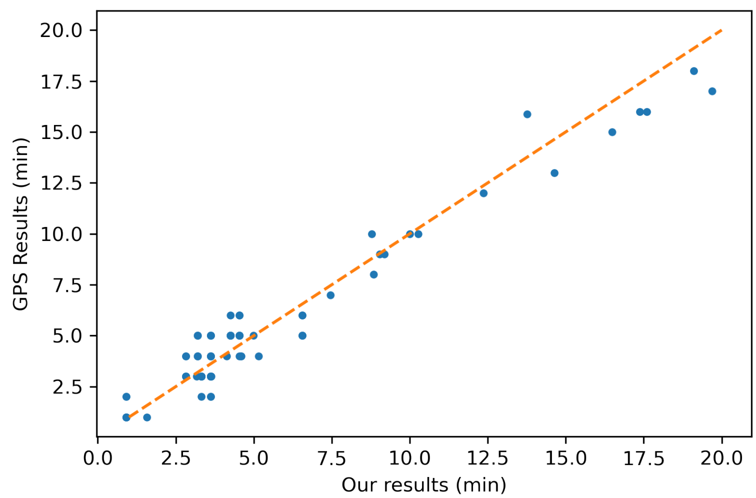
```
shortestdistance()

The current time and day rounded to the closest 15 minute mark is Sunday at 1530

If you want a different time and day type yes. If not, type no: no
Enter the starting location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: melrose volunteer
Enter the ending location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: cumberland phillip
Path:
Melrose & Volunteer->Cumberland & Volunteer->Cumberland & James

Directions:
From Melrose & Volunteer take Volunteer to get to Cumberland & Volunteer
From Cumberland & Volunteer take Cumberland Avenue to get to Cumberland & James

Total Time: 1:54
```

Then, the shortest path and the travel time will be printed.

```
shortestdistance()

The current time and day rounded to the closest 15 minute mark is Sunday at 1515

If you want a different time and day type yes. If not, type no: yes
Enter the time of day in military time (between 600 and 2200): 1300
Enter the day: mon
Enter the starting location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: canes
Enter the ending location.
 If you enter a street intersection, please enter only the first word of the street names.
 If you enter a landmark, please enter only the first word of the landmark: volunteer
Path:
17 & Cumberland->Cumberland & Volunteer->16 & White->James & White

Directions:
From 17 & Cumberland take Cumberland Avenue to get to Cumberland & Volunteer
From Cumberland & Volunteer take 16th Street to get to 16 & White
From 16 & White take White Avenue to get to James & White

Total Time: 2:46
```

To test the accuracy of our program, we randomly selected 2 nodes on our map and used our program to determine the shortest path. We compared this travel time to Google Maps' result for the same 2 nodes. We performed 60 comparisons, and graphed them in Figure 10. We note there are slight differences between our program's results and Google Maps' results. This could be due to error in the model. However, these differences are slight. The orange line in the figure denotes the 1:1 ratio between the GPS results and our program

Figure 10: Comparison of Our Data and GPS Data



results. The comparison between GPS and program results stay close to the orange line, so we conclude that our results coincide with the GPS travel time results.

4 Conclusion

Our data and technique can be reused to further and extend the analysis of cities. In terms of further analysis, more data can be collected on our current map to verify or adapt our current equations for the traffic conditions. While the modeling done in this project was specific to Knoxville, Tennessee, the modeling techniques and shortest path algorithm can be applied to many different scenarios. Some examples of using the same technique we used in our project include using a GPS to model emissions or using Dijkstra's algorithm to find the best route for firemen.

Our project could be used in places where Google Maps might not have data yet. Town-planners and trip-planners could adapt the modeling function we developed to create their own GPS-like system. Our work can be adapted in town planning, specifically to optimize routes and paths in towns. Our code is open source and can be customized for other applications. Additionally, Google Maps can occasionally have wrong information. There is not much a user can do to correct this. In our program, the user has full control of the data that is given. Traffic analysts, town-planners, and the casual trip-planner can make use of and build on our work.

5 Acknowledgements

ND, KW, and OP acknowledge the support of NSF award number 2045843.

Competing interests

The author(s) has/have no competing interests to declare.

6 Supplementary Materials

All supplementary files are located in the repository at https://github.com/noahd15/ModelingShortestPath. This includes our collected traffic data, calculations, jupyter notebook with user interface, pathfinder program used in the notebook, and other relevant images and screenshots.

References

[1] K. P. Burnham and D. R. Anderson, *Multimodel Inference: Understanding AIC and BIC in Model Selection*.

[2] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference A Practical Information- Theoretic Approach*, 2002.

[3] Google, *Google maps traffic color legend*. https://www.google.com/maps/. Accessed August 4 2022.

[4] ——, *Google maps traffic simulation: Thursday 7:05 pm*. https://www.google.com/maps/@35.9579519,-83.9316209,17z/data=!5m1!1e1. Accessed August 5 2022.

[5] R. Khatri, *Modeling route choice of utilitarian bikeshare users from gps data*, TRACE, (2015).

[6] S. C. S. Melki Friaswanto, Erick Alfons Lisangan, *The simulation of traffic signal preemption using gps and dijkstra algorithm for emergency fire handling at makassar city fire service*, IJASST, 3 (2021).

[7] J. E. Rito, N. S. Lopez, and J. B. M. Biona, *Modeling traffic flow, energy use, and emissions using google maps and google street view: The case of edsa, philippines*, Sustainability, 13 (2021).

[8] M. Snipes and D. C. Taylor, *Model selection and akaike information criteria: An example from wine ratings and prices*, Wine Economics and Policy, (2014).