The Chicago Gridlock: An Analysis of Chicago Traffic and Implications

Team Number: 11689
Modeling the Future Challenge 2023

March 3, 2023

Contents

1 Executive Summary **3**

2 Background Information **4**

3 Data Methodology **5**

4 Mathematics Methodology **6**
4.1 Overview 6
4.2 Assumptions 6
4.3 Variables 6
4.4 Random Forest Model 9
4.5 Street 1: 1034 (Wacker) 10
4.6 Street 2: 907 (Michigan) 11
4.7 Street 3: 1300 (Jackson) 12
4.8 Street 4: 860 (Higgins) 13
4.9 Street 5: 518 (Archer) 14
4.10 Overall Results 14
4.11 Strengths 15
4.12 Weaknesses 15

5 Risk Analysis **16**
5.1 Variables 16
5.2 Risk Overview 16
5.3 Expected Time Lost to Traffic 16
5.4 Expected Cost of Time Lost to Traffic 18
5.5 Expected Gallons of Gas Burnt due to Traffic 18
5.6 Expected Cost of Gas Burnt due to Traffic 18

6 Recommendations **19**
6.1 Congestion Pricing 19
6.2 Mass Transit and Alternative Transportation Options 19
6.3 Infrastructure Technologies 20

7 Conclusion **20**

8 Acknowledgments **20**

References **21**

1 Executive Summary

In 2022, $9.5 billion was lost due to traffic snarls on Chicago streets, making Chicago the most congested city in the US. As the economy recovers from the COVID slump and more and more drivers begin to hit the road again, the socioeconomic issues of gridlock are threatening to ramp up. In this study, we predict the risks associated with traffic congestion, as well as make recommendations for reducing congestion.

First, we identified the 5 most congested street segments in Chicago. In order to do so, we created a congestion index for the 1032 street segments in Chicago by dividing the average traffic speed during rush hour of the segment by the segment's maximum speed. Our data came from the Chicago Traffic Tracker, which updates every 10 minutes. From our function, we were able to identify the five most congested segments in Chicago (from most congested to least): Segment ID 1034 (Wacker), Segment ID 907 (Michigan), Segment ID 1300 (Jackson), Segment ID 860 (Higgins), Segment ID 518 (Archer).

With the 5 street segments identified, we wanted to dive deeper into each specific segment. We used three parameters, the hour of the day, the day of the week, and the month of the year while using the average speed on the segment as the dependent variable. We then used a Random Forest Regression model in order to forecast the traffic speed, with the input of the three parameters. We reached the highest accuracy of 86.22% from Segment ID 907 (Michigan).

In our risk analysis section, we calculated the expected cost of lost wages per year due to traffic to be $9,940,375,877. We also calculated the expected cost of gas burnt due to traffic to be $617,374,065. Based on our analysis, we provided city-level recommendations that would reduce the impact and risks associated with Chicagoan congestion. Specifically, we focused on opportunity costs stemming from lost time and resulting in lost wages and fuel costs. Using our model, we predicted that implementing congestion pricing would have the largest positive effect on time losses composed mostly of lost wages. In addition to congestion pricing, we suggest investing in alternative and public transport in order to reduce the second component, fuel costs. Finally, we recommend using new infrastructure technologies such as ITS, which would complement the two previous strategies in reducing time losses.

As Chicago traffic increases in the coming years, we believe our recommendations will help relieve the associated financial burden.

2 Background Information

Every day, in cities all around the world, a familiar scene unfolds. As the sun sets behind the towering skyscrapers of the city, the rush hour traffic comes to a standstill. Cars, buses, and trucks are bumper to bumper, the sound of honking horns and the hum of engines filling the air. Drivers sit impatiently behind the wheel, some tapping their fingers on the steering wheel, others drumming their thumbs on the dashboard. The road ahead is a sea of red taillights, stretching out as far as the eye can see.

Traffic congestion is a complex problem that has numerous negative impacts on both individuals and society as a whole. In addition to causing frustration and delays, traffic congestion can lead to increased fuel consumption, air pollution, and carbon emissions, which can harm the environment and public health. It can also contribute to traffic accidents, safety hazards, and negatively impact local businesses and the economy by reducing the efficiency of goods and service transportation. Furthermore, traffic congestion puts a strain on a city's infrastructure and resources, requiring more investments and maintenance.

Chicago is one of the cities that has been hit hardest by traffic congestion. According to INRIX, commuters in the Chicago area lost 155 hours to traffic congestion in 2022, making it the worst city for delays in the U.S. and the second-worst in the world after London. This represents a 49 percent increase in a single year from 104 hours lost in 2021.[9] The economic loss due to traffic congestion is estimated to be $9.5 billion dollars for the entire city or $2618 per driver. [6]

The reasons for the recent growth in traffic congestion in Chicago are varied. Economic growth and the city's geography, including high density next to a lake, limit how and where traffic congestion can be diffused across the surrounding area. The rapid rise of ride-hailing services like Uber and Lyft has also contributed to the increase in traffic congestion. According to separate data released by the city, Uber and Lyft trips in Chicago increased by 271 percent between 2015 and 2018. [7]

The concentration of destinations within Chicago's dense downtown contributes to the higher percentage of shorter rides in the city. The 2019 INRIX report found that 51 percent of all car trips in Chicago were less than three miles, and 22 percent were less than one mile. Seasonal factors like winter weather also influence drivers to take more short-distance trips.[12]

3 Data Methodology

The aim of this study is to develop a mathematical model to predict traffic congestion on Chicago streets using real-time traffic data. To achieve this, we gathered our data from the Chicago Traffic Tracker, a publicly available dataset that provides information on traffic congestion in the city. This dataset estimates traffic congestion on arterial streets in real-time by continuously monitoring and analyzing GPS traces from Chicago Transit Authority buses. [3]

The Chicago Traffic Tracker dataset contains a series of segment IDs that correspond to specific lengths of a street. These segment IDs are used to track the movement of buses along the streets in the city and are an essential component of the dataset. For this study, we utilized the traffic speed data from the dataset, which is a variable we used to measure the level of congestion at particular times.

To preprocess the data, we removed any missing or invalid data points from the dataset. Furthermore, we filtered out any outliers or extreme values in the data to avoid skewing our analysis. We then performed exploratory data analysis on the cleaned dataset to gain insights into the patterns and trends of traffic congestion in Chicago. This involved visualizing the data using plots and charts to identify trends in the data.

4 Mathematics Methodology

4.1 Overview

In this project, we first used the imported data and implemented this into a scaling formula to identify the five most congested segments in Chicago. Then we used a Random-Forest model to determine the weight of each of three factors - hour, day of the week, and month - in contributing to Chicago traffic levels.

4.2 Assumptions

1. *Data from 2022 is the most representative of Chicago's traffic.* Due to the sheer amount of data (244 million rows) that was available in the data portal, it was most logical and efficient to choose only one year; 2022 was the most recent year with a full year's worth of data and is after the brunt of the COVID pandemic's effects with the years of 2020 and 2021 having significantly different trends due to the lack of commuting.

2. *The 15th of the month is representative of the average of the rest of the days in the month.* Again because of the tremendous amount of data, we used the 15th as the day for the first eleven months, from January to November, and used the 19th of December because December did not contain any data for the 15th. By also checking that there weren't any significant events or holidays on each of the 12 days, it ensured that none of the values for these days were outliers. **This assumption is only used for finding the most congested segments.**

3. *Morning rush hour occurs from 6:00 AM to 8:00 AM and afternoon rush hour occurs from 4:00 PM to 6:00 PM.* [17] Our main goal was to estimate traffic levels in just these hours because this is when congestion is at its peak and thus is representative of the entire day's worth of traffic data.

4. *Data rows with no estimate available are insignificant to the general trend of the data.* All traffic data rows with a speed of -1 were deleted, which meant that no estimate was available.

4.3 Variables

S_{avg}	the Average Speed Recorded on a Segment (Miles per Hour)
S_{max}	the Maximum Speed Recorded on a Segment (Miles per Hour)
C	Congestion Index
H	Hour of the Day
D	Day of the Week
M	Month of Year

Table 1: Variables

Due to the vast amount of segments within our dataset, we first wanted to identify the segments that were the most congested and address those areas specifically. To do this, we developed a formula that would output a relative congestion rating or a congestion index of a segment on a scale from 0 to 1. [16] [20] The formula is as follows:

$$C = \frac{S_{avg}}{S_{max}}$$

where S_{avg} is the average speed of traffic during rush hours, and S_{max} is the maximum speed identified in hours outside of rush hour. We assumed that the value of S_{max} was representative of the actual speed limit of the segment, and would serve as the baseline. By dividing the average speed during rush hours by S_{max}, we could find how relatively congested the segment was, with values closer to 0 indicative of greater congestion and values closer to 1 indicative of less congestion.

For each segment, we found the average speed during rush hour times across the twelve months in 2022 (using the 15th/19th of each month) and ran it through our formula to produce a traffic congestion index score for each segment. The top five most congested segments are as follows:

Segment ID	Congestion Index
1034	0.501240
907	0.503151
1300	0.509485
860	0.510299
518	0.519819

Table 2: Top Five Most Congested Segment IDs

Given that each of the scaled speeds was approximately 0.5, this indicates that each of the segment's average speeds during rush hour was approximately half of the speed limit.

These Segment IDS of the previous table correspond to the following streets:

Segment ID	Name, Beginning, and Endpoint
1034	Wacker, from Lake to Madison
907	Michigan, from Congress to Randolph
1300	Jackson, from Halstead to Wacker
860	Higgins, from Milwaukee to Central
518	Archer, from Damen to Western

Table 3: Corresponding Top Five Most Congested Segments

We can verify our results because Wacker Drive appears twice in our top 5 (ID 1034 and ID 1300) and Wacker Drive is known to experience frequent and chaotic gridlocks. [1]

Below is a map highlighting the locations of four out of the five street segments where red corresponds to 1034, blue to 907, purple to 1300, and pink to 518. 860 is not included due to its distance from the rest of the segments.

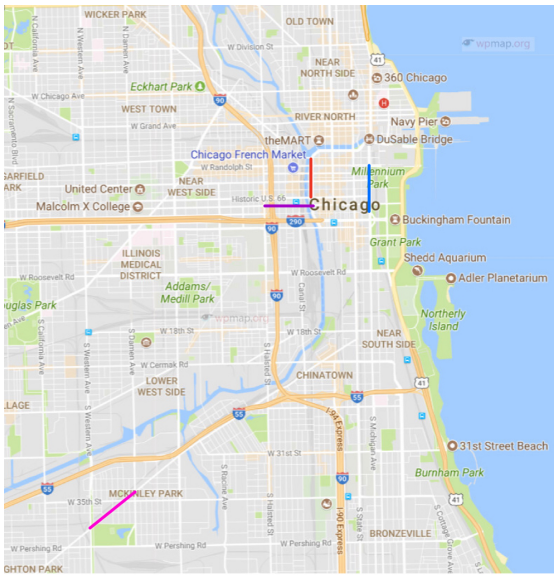


Figure 1: Map Of Congested Streets

The following histogram shows the distribution of the traffic congestion index among the streets in Chicago:

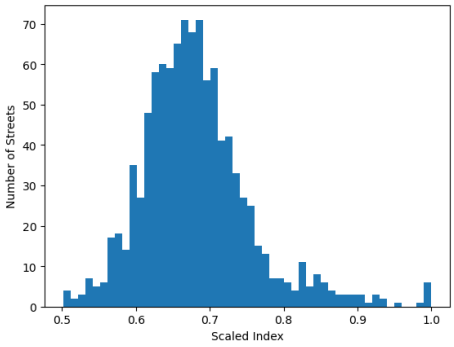


Figure 2: Distribution of Congestion of Street Segments

Considering that our top five segments are close to an index of 0.5 and that most streets are near the 0.65 to 0.70 range, we confirm that the traffic in these five segments is significantly worse compared to other segments.

4.4 Random Forest Model

Upon deciding which model to utilize to predict traffic congestion in Chicago we landed on the idea of using an ARIMA model and a Random Forest Regression model. [10] ARIMA stands for auto-regressive integrated moving average model, where the model references prior data points in order to forecast future values. The model is commonly used for stock prices and weather forecasts. However, due to the fluctuating nature of our data and the size of our data, we concluded that the ARIMA model was inappropriate for short-term and long-term prediction. [2] [13] Thus, we resorted to using a Random Forest Regression, which is a supervised learning algorithm that uses an ensemble of decision trees to make predictions.

Using a Random Forest Regression model utilizing the parameters of the hour of the day, the day in the week, and the month of the year as our parameters, we derived a model for each Segment ID. We used the pandas and NumPy packages on a Jupyter Notebook to read the CSV files. Then, because the Mean Absolute Percentage Error (MAPE) function was used to show the accuracy of our model and 0 is not allowed in the MAPE model, we had to change all values of 0 into 1. Then, using the train-test-split package from sklearn.model-selection, we split our data into training and testing data, in a 75/25 ratio. [8] Before running the Random Forest model, we calculated the average baseline error by taking the average difference between the mean speed of the segments and the test data. This step sets a baseline for our model to perform better because the average baseline error is if we used the mean speed to guess every single test data. We instantiated the Random Forest model with 1000 decision trees and with a random state of 42 for all segments.

After setting up our Random Forest model, we were able to visualize the importance of our three variables (hour, day of the week, and month of the year). We also visualized in a graph the predicted value vs the actual value of the first 100 test data points. Because the data was so large, we couldn't fit the entirety in a graph, which was why we graphed the first 100 points for the purpose of approximately visualizing how our prediction compares to the actual values. [19]

4.5 Street 1: 1034 (Wacker)

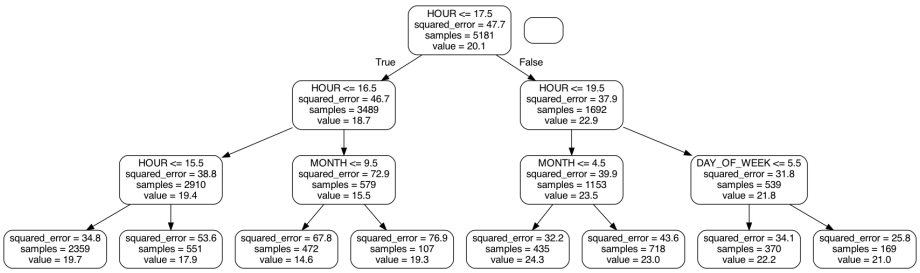
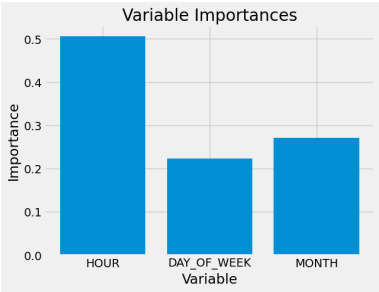


Figure 3: 1034 Random Forest

The figure above represents the overall decision tree of the random forest model. This model had an average baseline of 5.31, a mean absolute error of 4.81, and an accuracy of 52.71%. The accuracy seems to be very low, yet this can be attributed to the fact that the street often lacked data with a lot of -1 as its value. Thus, the street only had 10,000 data points as opposed to the others which contained approximately 40,000.

From this model we also determined the degree of importance of each of the parameters and plotted the predicted values vs the actual values of the first 100 data points in our testing set:



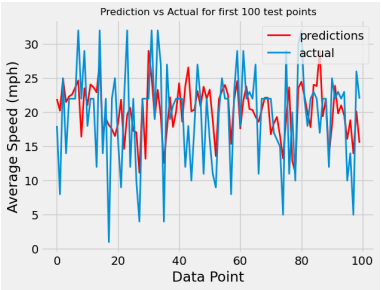


Figure 4: 1034 Variable Importance —— Actual vs Predicted 100 Data Points

From the prediction vs actual values graph, we can see that the actual values fluctuate greatly. The value can jump from close to 0 mph all the way to 30 mph, which made the accuracy value very low. Yet, we can still see that our prediction captured the general pattern of the graph.

4.6 Street 2: 907 (Michigan)

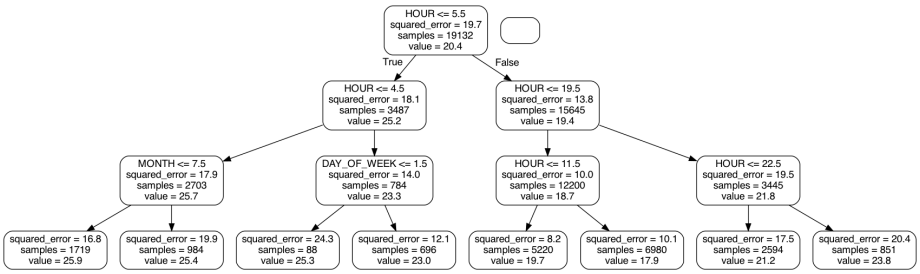
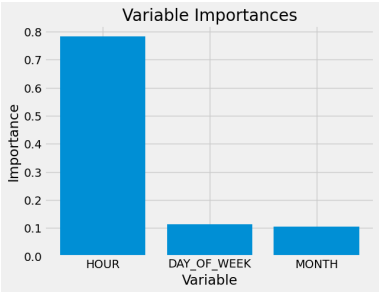


Figure 5: 907 Random Forest

The figure above represents the overall decision tree of the random forest model. This model had an average baseline error of 3.46, and a mean absolute error of 2.42, meaning that our model was able to predict 1 mph better than without the model. The model also had an accuracy of 86.22%, which was by far the highest score of all our models. From this model we also determined the degree of importance of each of the parameters and plotted the predicted values vs the actual values of the first 100 data points in our testing set:



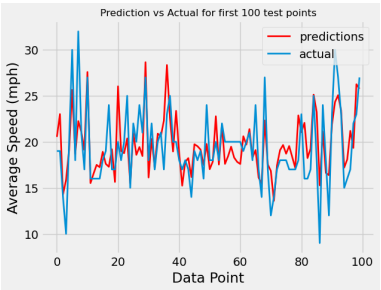


Figure 6: 907 Variable Importance —— Actual vs Predicted 100 Data Points

Looking at the prediction vs actual values graph, we can see how closely our prediction values resemble the actual values. Although the graph only shows the first 100 test points, we can verify by the 86.22% accuracy score that the model will accurately predict the average speed.

4.7 Street 3: 1300 (Jackson)

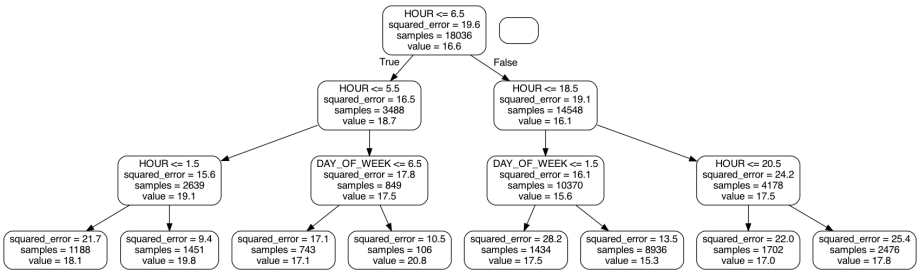
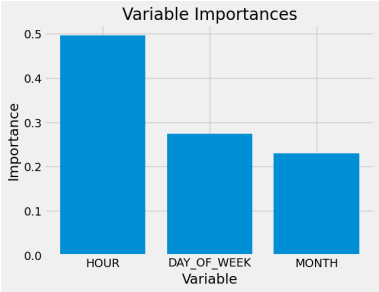


Figure 7: 1300 Random Forest

The figure above represents the overall decision tree of the random forest model. This model had an average baseline error of 3.48, a mean absolute error of 3.0, and an accuracy of 68.83%. From this model we also determined the degree of importance of each of the parameters and plotted the predicted values vs the actual values of the first 100 data points in our testing set:



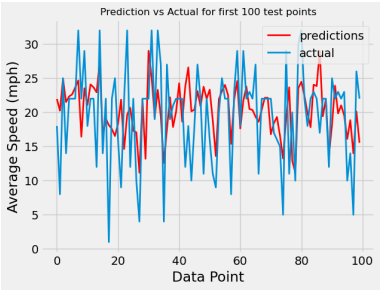


Figure 8: 1300 Variable Importance —— Actual vs Predicted 100 Data Points

This street also had a high fluctuation between each data point, which is shown by the graph above. The accuracy is between Street 1 and Street 2, but our model was able to predict 0.48 mph better than the baseline.

4.8 Street 4: 860 (Higgins)

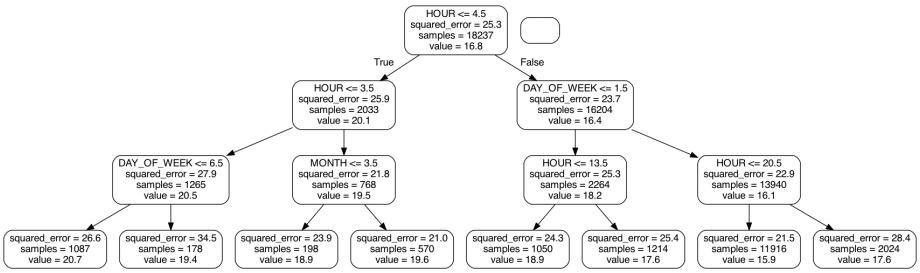
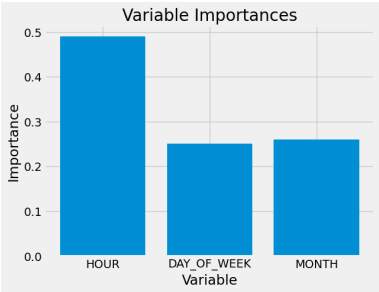


Figure 9: 860 Random Forest

The figure above represents the overall decision tree of the random forest model. This model had an average baseline error of 4.21, a mean absolute error of 4.04, and an accuracy of 72.66%. From this model we also determined the degree of importance of each of the parameters and plotted the predicted values vs the actual values of the first 100 data points in our testing set:



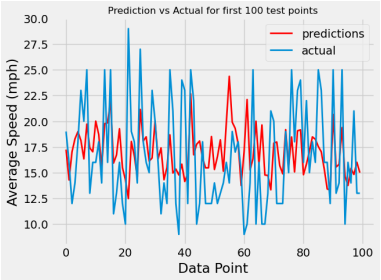


Figure 10: 860 Variable Importance ——— Actual vs Predicted 100 Data Points

Looking at the prediction vs actual value graph above, we can see the level of fluctuation on Street 860 (Higgins) as well. Because our Random Forest model took in over 30,000 data points, the prediction essentially averages the fluctuations and creates a relatively stable output.

4.9 Street 5: 518 (Archer)

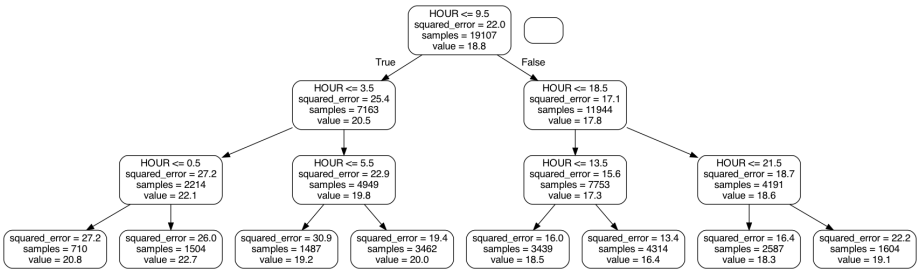
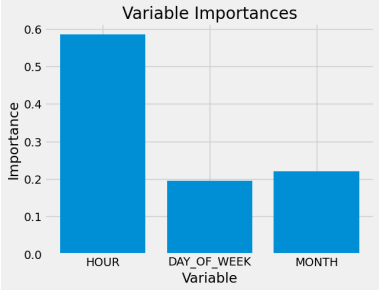


Figure 11: 518 Random Forest

The figure above represents the overall decision tree of the random forest model. This model had an average baseline error of 3.89, a mean absolute error of 3.51, and an accuracy of 78.46%. From this model we also determined the degree of importance of each of the parameters and plotted the predicted values vs the actual values of the first 100 data points in our testing set:



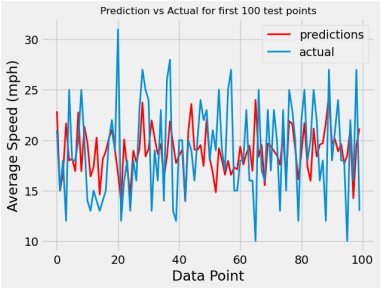


Figure 12: 518 Variable Importance ——— Actual vs Predicted 100 Data Points

As seen above, Street 518 (Archer) also captures a similar trend as the streets before, where the actual values fluctuate greatly and our prediction values are far more stable.

4.10 Overall Results

From the five graphs of "Variable Importance," it can be observed that the variable "Hour" has significantly greater importance than the two other variables, "Day Of Week" and "Month", confirming that the hour of the day is the most important factor when considering traffic. This suggests that driving during rush hours is the biggest factor contributing to Chicago traffic. In addition, although the accuracy for each street varied greatly, our Random Forest model accurately predicted the overall trend.

4.11 Strengths

To measure the accuracy of our model, we used the Mean Absolute Percentage Error (MAPE), which is a way to measure the statistical accuracy of a forecast system. The formula for MAPE is as follows:

$$M = \frac{1}{n}\sum_{t=1}^{n} |\frac{A_t - F_t}{A_t}|$$

where M is the MAPE, n is the number of data points, A_t is the actual value for the time segment, and F_t is the forecasted time segment. MAPE returns the percentage error of the forecast, so we subtracted each of our MAPE values from 100 to find the accuracy of our Random Forest model for each of our selected streets. Based on the five actual versus prediction graphs for the five congested streets, it can be observed that Street 907's prediction was quite accurate with an accuracy of 86.22%.

The results of our model are strong due to many of the advantages of using a Random Forest regression. Random Forests are non-parametric models, meaning they don't rely on input data being normally distributed in order to give accurate predictions. This lowers the necessity of cleaning data for future uses of this model, resulting in further ease of use when expanding the model to more streets. This also ensures our model is not heavily affected by outliers. Random Forests are also not greatly harmed by missing data, meaning our model will not be hampered by equipment malfunctions when expanding our model to new streets. Lastly, Random Forests are efficient in their processing, which is much needed when considering the vastness of the city of Chicago.

4.12 Weaknesses

One weakness of our mathematical model is its inability to identify individual factors impacting traffic such as weather conditions and infrastructure level. This is due to the lack of sufficient data to be able to incorporate these factors. While these factors do not need to be taken into account when planning for future congestion due to their largely unpredictable nature, they are certainly useful in predicting current-day congestion.

Furthermore, the parameters used in our model were limited only to the hour of the day, the day of the week, and which month, which fails to capture the minute fluctuations of traffic within a single hour. Although we only had three parameters, we believed they were the most important features. Adding more unnecessary features to our Random Forest model can actually decrease the accuracy of our model, which is why we limited our selection to those three.

5 Risk Analysis

5.1 Variables

Variable Name	Variable Description
S_{avg}	The Average Speed Cars Travel at in Chicago
L	The Assumed Length of a street in Chicago
S_{obs}	Observed Speeds per hour
G_p	Average Price of Gas for 2022
G_L	Average Number of Gallons of Gas Lost to Idling Cars
$G_{L_{total}}$	The Total Gallons Lost to Traffic Congestion each Year
D	Total Distance (miles) Driven by Chicago Drivers Each Day
W	Average Hourly Wage of a Chicago Resident
$T_{L_{weekly}}$	The Expected Time Lost Each Week When Driving on a Street
C_G	The Expected Cost of Gas Wasted Each Year from Idling in Traffic
C_T	The Expected Cost of Time Lost Each Year to Traffic Congestion
$T_{L_{total}}$	The Total Time Lost Each Year to Traffic Congestion

Table 4: Risk Analysis Variables

5.2 Risk Overview

Traffic is typically considered nothing more than an annoyance, yet the true costs of traffic are enormous. These costs are typically concealed due to their nature. Wasted time spent in traffic is excessive, depriving many people of personal time they could instead spend with loved ones. More concretely, this time wasted also reduces the amount of time available in a day for a person's work, depriving them of hours of wages. Gas burnt while idling in traffic also costs commuters an enormous amount. Overall, traffic is a problem that has grown beyond an annoyance into a truly costly issue.

5.3 Expected Time Lost to Traffic

Using the data from Section 4.6, we generated a graph of the speed over time on Street 907, using the weekdays of a randomly selected week. The average speed over the course of each weekday that week is shown in Figure 13 and will be used in the following calculations.

Time lost due to traffic takes up a large part of commuters' days. To predict the total time lost to traffic, we generalize Segment 907 (the Segment that our model was most accurate at regressing) to the rest of the city. We find the average speed per day for Segment 907 - 23.147 mph - and also identify the hours of the day for which the average speed of the street falls below the daily average. This is done so that our calculations of time lost only take into account daytime hours and ignore nighttime hours when streets are relatively clear. For each hour that the average hourly speed is below the average daily speed, we

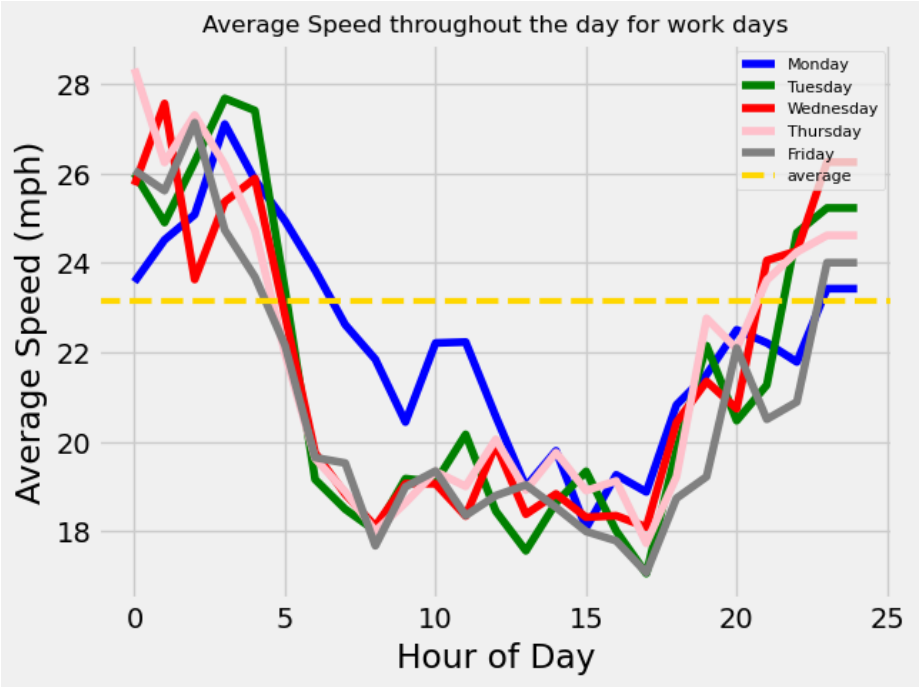


Figure 13: Average Vehicle Speed on Weekdays

calculate the time lost during a trip on that street during that specific hour using the equation:

$$T_{L_{weekly}} = \sum_{Monday}^{Friday} \frac{L}{S_{obs}} - \frac{L}{S_{avg}}$$

where the length of the street is 0.6 miles. Adding these values across the 5 weekdays gets us a $T_{L_{weekly}}$ of 0.02384387.

$$T_{L_{total}} = \left(\frac{T_{L_{weekly}}}{5}\right) \cdot \left(\frac{D}{L} \cdot 5 \cdot 52\right)$$

We then use the above equation to get the total time, in hours, lost each year to traffic. We divide the T_W by 5 to get the average daily time lost from driving through Segment 907. Our T_D - the estimated total miles driven by Chicago drivers each day - is 136 million [4], which we divide by L because we are generalizing Segment 907 to all of Chicago. We multiply this quotient by 5 and then again by 52 to get expand our daily miles traveled to all the weekdays in a year. Substituting the appropriate values into this equation yields a $T_{L_{total}}$ of 281,039,748 hours lost.

5.4 Expected Cost of Time Lost to Traffic

Beyond the obvious loss of personal time caused by hours lost to traffic congestion, spending time in traffic prevents drivers from working. This deprives drivers of wages that could have been gained from hours working instead of driving.

The average hourly wage of a Chicago worker (W) is estimated to be \$35.37. [21] Multiplying this wage by our derived $T_{L_{weekly}}$ gives us C_T, the total amount lost to traffic congestion due to lost wages:

$$C_T = T_{L_{total}} \cdot W$$

The resulting approximation gives us \$9,940,375,877 worth of lost wages per year due to traffic congestion. This value is supported by the INRIX estimation that traffic delays cost the Chicago area \$9.5 billion in the year 2022. [6]

5.5 Expected Gallons of Gas Burnt due to Traffic

Missing out on possible wages is not the only egregious cost of traffic congestion. Time spent idling in traffic causes Chicago drivers to burn and waste an enormous amount of gas each year. The estimated gas burnt by leaving an idling engine on for an hour (G_L) is approximately half a gallon. [18] Assuming this value is constant for cars in gridlock, we can obtain the total gallons of gas lost to traffic congestion each year through the formula:

$$G_{L_{total}} = G_L \cdot T_{L_{total}}$$

Using this formula we find that approximately 140,519,873 unnecessary gallons of gas are burnt each year due to traffic congestion. This excess burning of gas has a disastrous effect on the environment, contributing massively to global warming as greenhouse gasses are added to the atmosphere.

5.6 Expected Cost of Gas Burnt due to Traffic

This massive amount of excess gasoline burning also has a massive economic cost. The equation:

$$C_G = G_{Ltotal} \cdot G_P$$

where we substitute in \$4.3935 as the cost of gas (found by averaging monthly gas prices from 2022) [22], we find that \$617,374,065 worth of excess gas is burned each year due to traffic congestion.

6 Recommendations

Through our mathematical modeling and identification of the most important factors contributing to traffic, we now offer recommendations to reduce traffic congestion and help mitigate its impacts.

6.1 Congestion Pricing

One of the most promising methods of reducing traffic is through the use of congestion pricing. Congestion pricing is a system where drivers are charged a fee to over congested areas or use congested roads during peak hours. The use of congestion pricing discourages driving during times of high congestion and encourages the use of alternative transportation options. [14] The pricing level should reflect the traffic level of the street and the time of day. Doing so will encourage Chicago drivers to refrain from driving during rush hours; if they decide to drive despite the fee, they will be incentivized to utilize the road system to its maximum efficiency. The system will especially decrease traffic during the most congested times or days of the year as predicted by our model, decreasing traffic and increasing the use of public transportation during periods of high congestion. To promote behaviors that would reduce congestion, these tolls could be reduced or even eliminated for carpoolers and families, thus increasing average vehicle occupancy (AVO) and reducing congestion. Overall, implementing congestion pricing would significantly lower the lost wages that result from time lost in traffic. [11]

6.2 Mass Transit and Alternative Transportation Options

If drivers are willing to drive despite the charge, the extra revenue earned by the city can be used to augment the city's public transportation such as buses, trains, and subways, which can provide commuters with alternative transportation options, reducing the number of cars on the road. Improvements to mass transit can also increase its appeal, such as making it more reliable, affordable, and efficient. One specific area of improvement is the increased use of Bus Rapid Transit (BRT) systems: high-capacity buses that efficiently shuttle passengers around the city. While Chicago already has its own bus system, expanding upon that system would reduce traffic and congestion. [15]

The city can also channel funds raised from congestion pricing to the CTA (Chicago Transit Authority) to improve the efficiency and reliability of Chicago public transportation. Furthermore, the extra revenue can be dedicated to encouraging the use of alternative transportation options, such as biking, walking, or carpooling, which can help reduce the number of cars on the road. The city can invest in bike lanes and pedestrian infrastructure to make these options more accessible and safer.

Using this strategy, the city of Chicago would benefit from the reduction of greenhouse gases resulting from decreased congestion and increased alternative

transportation. Overall, drivers will experience fewer time losses in traffic if this strategy is implemented, which would improve the fuel cost component of financial time losses.

6.3 Infrastructure Technologies

Another suggestion is an investment in infrastructure technologies to reduce overall congestion. Advanced traffic systems (including smart traffic lights) can take into account different weather and traffic conditions to suggest more feasible travel routes. These advanced systems could also be adapted to account for current congestion and limit the number of cars allowed onto trafficked streets through the use of stoplights. Other technologies include ITS (Intelligence Transportation Systems) which can inform drivers of safer routes to travel, encouraging the use of less trafficked roads. [5]

While there are multiple applications such as ITS or Google Maps that already warn drivers of traffic during rides, these applications lack the ability to accurately predict traffic far in advance. Our model could be used to forecast congestion days or even weeks into the future, allowing drivers to plan ahead for congestion and instead opt for public transport or other options. Our model will overall aid optimization and ease of limiting congestion, as cities and citizens will have an abundance of predictive information to plan for upcoming congestion. This proactive approach limits the frustration that can sometimes be caused by the reactivity of Advanced Traffic Systems and other applications that only limit congestion during drives rather than before the user even gets in their car.

7 Conclusion

Overall, our model predicted the congestion levels of Chicago streets and analyzed the risk brought on due to traffic delays. We analyzed the congestion levels of the five most trafficked segments in Chicago, which we determined using public Chicago traffic databases. Based on our results, we recommend the strategy of congestion pricing in order to reduce the lost costs due to traffic and the associated financial risks. Furthermore, we suggest using the funds raised from our congestion pricing model to invest in public and alternative transport alongside ITS development, lowering the financial risks of traffic. [23]

8 Acknowledgments

We would like to thank our mentor, Jason Witcraft, for his expert advice and insights.

We would also like to thank our coach, Paul Kim, for giving us the invaluable opportunity to participate in this learning experience.

Finally, we would like to thank the Actuarial Foundation for making this wonderful experience possible.

References

[1] Kelly Bauer. Police shutdown of downtown traffic saturday was 'unorganized chaos,' frustrated residents say. *Block Club Chicago*, Sep 2022. URL https://blockclubchicago.org/2022/09/19/police-shutdown-of-downtown-traffic-over-the-weekend-was-unorganized-chaos-frustrated-residents-say/.

[2] Neha Bora. Understanding arima models for machine learning. *Capital One*, Nov 2021. URL https://www.capitalone.com/tech/machine-learning/understanding-arima-models/.

[3] City of Chicago. Chicago traffic tracker - historical congestion estimates by segment - 2018-current: City of chicago: Data portal, Mar 2023. URL https://data.cityofchicago.org/Transportation/Chicago-Traffic-Tracker-Historical-Congestion-Esti/sxs8-h27x.

[4] Chicago Metropolitan Agency for Planning. Navigation, 2022. URL https://www.cmap.illinois.gov/onto2050/snapshot-reports/transportation-network/travel-trends.

[5] Department of Transportation. Intelligent transportation systems - its in use today, 2022. URL https://www.its.dot.gov/resources/fastfacts.htm.

[6] Sarah Freishtat. Chicago commuters lost more hours to congestion in 2022 than drivers in any other major u.s. city, report shows. *Chicago Tribune*, Jan 2023. URL https://www.chicagotribune.com/business/ct-biz-chicago-worst-traffic-nation-study-20230110-7c5ydouxq5dw3ndugqo2gkgf3e-story.html.

[7] Sara Freund. Chicago's new ride-hailing tax is here, and it's the country's highest fee. *Curbed Chicago*, Jan 2020. URL https://chicago.curbed.com/2019/12/2/20992472/chicago-transportation-tax-uber-lyft-taxi.

[8] Michael Galarnyk. Understanding train test split. *Built In*, Jul 2022. URL https://builtin.com/data-science/train-test-split.

[9] INRIX. Inrix 2022 global traffic scorecard. retrieved february 15, 2023., 2022. URL https://inrix.com/scorecard/.

[10] Will Koehrsen. Random forest in python. *Medium*, Jan 2018. URL https://towardsdatascience.com/random-forest-in-python-24d0893d51c0.

[11] Jay Koziarz. Is congestion pricing the key to solving chicago's traffic woes? *Curbed Chicago*, Oct 2017. URL https://chicago.curbed.com/2017/10/26/16549842/transportation-traffic-congestion-pricing.

[12] Jay Koziarz. Chicago's traffic was the second worst in the nation in 2019, says report. *Curbed Chicago*, Mar 2020. URL https://chicago.curbed.com/2019/2/14/18224967/chicago-traffic-report-worst-nation-transportation.

[13] S. Vasantha Kumar and Lelitha Vanajakshi. Short-term traffic flow prediction using seasonal arima model with limited input data. *European Transport Research Review*, 7(3), 2015. doi: 10.1007/s12544-015-0170-8.

[14] 2022 May 25. What is congestion pricing? *NRDC*, May 2022. URL https://www.nrdc.org/stories/what-is-congestion-pricing.

[15] Team Moovaz. Transport in chicago: 8 different ways to get around the city, Jun 2022. URL https://www.moovaz.com/blog/transport-in-chicago/.

[16] Florian Neukart, Gabriele Compostella, Christian Seidel, David von Dollen, Sheir Yarkoni, and Bob Parney. Traffic flow optimization using a quantum annealer. *Frontiers*, Dec 2017. URL https://www.frontiersin.org/articles/10.3389/fict.2017.00029/full.

[17] Sam Rakestraw. Traffic patterns in chicago. *Insurance Navy*, 2022. URL https://www.insurancenavy.com/traffic-patterns-chicago/.

[18] S.C. Department of Health and Environmental Control. Idling: Why it's a problem and what you can do, 2009. URL https://scdhec.gov/sites/default/files/Library/CR-010109.pdf.

[19] Mark R Segal. Machine learning benchmarks and random forest regression. *UCSF: Center for Bioinformatics and Molecular Biostatistics*, Apr 2003.

[20] PhD Serafeim Loukas. Everything you need to know about min-max normalization in python. *Medium*, Mar 2023. URL https://towardsdatascience.com/everything-you-need-to-know-about-min-max-normalization-in-python-b79592732b79.

[21] U.S. Bureau of Labor Statistics. Occupational employment and wages in chicago-naperville-elgin - may 2021 : Midwest information office, Aug 2022. URL https://www.bls.gov/regions/midwest/news-release/occupationalemploymentandwages_chicago.htm.

[22] U.S. Energy Information Administration. Chicago regular all formulations retail gasoline prices, Feb 2023. URL https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=pet&s=emm_epmr_pte_yord_dpg&f=m.

[23] USAFacts. Transportation, Apr 2020. URL https://usafacts.org/data/topics/people-society/transportation/.

Code: Random Forest Regression

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

features = pd.read_csv('907.csv')
features = features.sort_values(['TIME'])
features = features.reset_index()
features = features[['SPEED', 'HOUR', 'DAY_OF_WEEK', 'MONTH']]
labels = np.array(features['SPEED'])
features = features.drop('SPEED', axis = 1)
feature_list = list(features.columns)
features = np.array(features)

train_features, test_features, train_labels, test_labels =
    train_test_split(features, labels, test_size = 0.25,
    random_state = 42)
for i in range(len(test_labels)):
    if (test_labels[i] == 0):
        test_labels[i] = 1

print('Training Features Shape:', train_features.shape)
print('Training Labels Shape:', train_labels.shape)
print('Testing Features Shape:', test_features.shape)
print('Testing Labels Shape:', test_labels.shape)

baseline_errors = abs(20.43 - test_labels)
print('Average baseline error: ', round(np.mean(baseline_errors),
    2))

# Instantiate model with 1000 decision trees
rf = RandomForestRegressor(n_estimators = 1000, random_state = 42)
# Train the model on training data
rf.fit(train_features, train_labels);

predictions = rf.predict(test_features)
# Calculate the absolute errors
errors = abs(predictions - test_labels)
# Print out the mean absolute error (mae)
print('Mean Absolute Error:', round(np.mean(errors), 2), 'degrees.'
    )

mape = 100 * (errors / test_labels)
print(mape)
# Calculate and display accuracy
accuracy = 100 - np.mean(mape)
print('Accuracy:', round(accuracy, 2), '%.')

# Import tools needed for visualization
from sklearn.tree import export_graphviz
import pydot
# Pull out one tree from the forest
tree = rf.estimators_[5]
# Import tools needed for visualization
from sklearn.tree import export_graphviz
```

```
import pydot
# Pull out one tree from the forest
tree = rf.estimators_[5]
# Export the image to a dot file
export_graphviz(tree, out_file = 'tree907.dot', feature_names =
    feature_list, rounded = True, precision = 1)
# Use dot file to create a graph
(graph, ) = pydot.graph_from_dot_file('tree907.dot')
# Write graph to a png file
graph.write_png('tree907.png')

# Limit depth of tree to 3 levels
rf_small = RandomForestRegressor(n_estimators=10, max_depth = 3)
rf_small.fit(train_features, train_labels)
# Extract the small tree
tree_small = rf_small.estimators_[5]
# Save the tree as a png image
export_graphviz(tree_small, out_file = 'small_tree907.dot',
    feature_names = feature_list, rounded = True, precision = 1)
(graph, ) = pydot.graph_from_dot_file('small_tree907.dot')
graph.write_png('small_tree907.png');

# Get numerical feature importances
importances = list(rf.feature_importances_)
# List of tuples with variable and importance
feature_importances = [(feature, round(importance, 2)) for feature,
     importance in zip(feature_list, importances)]
# Sort the feature importances by most important first
feature_importances = sorted(feature_importances, key = lambda x: x
    [1], reverse = True)
# Print out the feature and importances
[print('Variable: {:20} Importance: {}'.format(*pair)) for pair in
    feature_importances];

# Import matplotlib for plotting and use magic command for Jupyter
    Notebooks
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
# Set the style
plt.style.use('fivethirtyeight')
# list of x locations for plotting
x_values = list(range(len(importances)))
# Make a bar chart
plt.bar(x_values, importances, orientation = 'vertical')
# Tick labels for x axis
plt.xticks(x_values, feature_list, rotation='horizontal')
# Axis labels and title
plt.ylabel('Importance'); plt.xlabel('Variable'); plt.title('
    Variable Importances');

new_labels = test_labels[0:100]
new_predictions = predictions[0:100]
x = np.arange(200)
plt.plot(new_predictions, c = 'red', linewidth = 2, label = '
    predictions')
plt.plot(new_labels, linewidth = 2, label = 'actual')
plt.legend(loc = "upper right")
```

```
plt.xlabel("Data Point")
plt.ylabel("Average Speed (mph)")
plt.title("Prediction vs Actual for first 100 test points", size = "12")

list_day_M = []
list_day_T = []
list_day_W = []
list_day_Th = []
list_day_F = []
list_day_Sa = []
list_day_Su = []
for i in range(25):
    predicted = rf.predict([[i,1,1]])
    list_day_M.append(predicted)
for i in range(25):
    predicted = rf.predict([[i,2,1]])
    list_day_T.append(predicted)
for i in range(25):
    predicted = rf.predict([[i,3,1]])
    list_day_W.append(predicted)
for i in range(25):
    predicted = rf.predict([[i,4,1]])
    list_day_Th.append(predicted)
for i in range(25):
    predicted = rf.predict([[i,5,1]])
    list_day_F.append(predicted)
for i in range(25):
    predicted = rf.predict([[i,6,1]])
    list_day_Sa.append(predicted)
for i in range(25):
    predicted = rf.predict([[i,7,1]])
    list_day_Su.append(predicted)

plt.plot(list_day_M, color = "blue", label = "Monday")
plt.plot(list_day_T, color = "green", label = "Tuesday")
plt.plot(list_day_W, color = "red", label = "Wednesday")
plt.plot(list_day_Th, color = "pink", label = "Thursday")
plt.plot(list_day_F, color = "gray", label = "Friday")
plt.axhline(y = 23.15, color = "gold", linestyle = "--", linewidth = "3", label = "average")
plt.legend(loc = "upper right", prop={'size': 8})
plt.xlabel("Hour of Day")
plt.ylabel("Average Speed (mph)")
plt.title("Average Speed throughout the day for work days", size = "12")

def time_lost_func(list1, avg, length):
    time_lost = []
    for i in range (25):
        if (list1[i] < avg):
            time_lost.append(list1[i])
    time_lost_2 = 0
    for i in range (len(time_lost)):
        x = (length)/(time_lost[i]) - (length)/(avg)
        time_lost_2 += x
    return(time_lost_2/len(time_lost))
```

```

time_lost_M = time_lost_func(list_day_M, 23.14716671, 0.6)
time_lost_T = time_lost_func(list_day_T, 23.14716671, 0.6)
time_lost_W = time_lost_func(list_day_W, 23.14716671, 0.6)
time_lost_Th = time_lost_func(list_day_Th, 23.14716671, 0.6)
time_lost_F = time_lost_func(list_day_F, 23.14716671, 0.6)

count = 0
for i in range (25):
    count += list_day_M[i]
print(count/24)
```

Code: Identifying 5 Most Congested Streets

```
import pandas as pd
import numpy as np

traffic_df = pd.read_csv('traffictotal.csv')
other_df = pd.read_csv('othertimes.csv')

traffic = traffic_df.groupby(['SEGMENT_ID']).mean().sort_values('
    SEGMENT_ID', ascending = True)

max_speed = other_df.groupby(['SEGMENT_ID']).max().sort_values('
    SEGMENT_ID', ascending = True)

max_speed = max_speed.reset_index()
traffic = traffic.reset_index()

max_speed_2 = max_speed[max_speed['SEGMENT_ID'].isin(traffic['
    SEGMENT_ID'])]
#min_speed_2 = min_speed[min_speed['SEGMENT_ID'].isin(evening['
    SEGMENT_ID'])]

traffic = traffic.reset_index()
max_speed_2 = max_speed_2.reset_index()

minmaxnorm = pd.DataFrame(columns = ['SEGMENT_ID', 'Scaled'])
for x in range(len(traffic)):
    scaled = (traffic.loc[x]['SPEED'] - 0)/(max_speed_2.loc[x]['
    SPEED'])
    minmaxnorm = minmaxnorm.append({'SEGMENT_ID' : traffic.loc[x]['
    SEGMENT_ID'], 'Scaled' : scaled}, ignore_index = True)
    print(x, scaled)

minmaxnorm.sort_values('Scaled', ascending = True).head(5)

traffic_df.loc[traffic_df['SEGMENT_ID'] == 1034].sort_values('SPEED
    ', ascending = False)

import matplotlib.pyplot as plt

plt.hist(minmaxnorm['Scaled'], bins = 50)
plt.xlabel('Scaled Index')
plt.ylabel('Number of Streets')
```