DATA 100:
Final Report
(Traffic Dataset)

Students: Sreeja Apparaju, James Jeong, Tanmay Vijaywargiya

Discussion Session 135 | Date: Dec 13, 2021

Introduction

We worked with Uber's Traffic Dataset for the DATA 100 Final Project. In the design document (part 1), we proposed our hypothesis by exploring the time traveled dataset. In this final report (part 2), we largely focus on the motivating factors for our baseline models and our improvements through visualizations and statistics methods. In the end, we added a section about future work.

Design Document Reprised

When we were exploring the dataset in our open EDA, we found that there was a drastic decrease in the time travel after the COVID 19 lockdown was imposed. Further, plotting the choropleth maps using geometry, we were able to see the regional variations within the time travel when taken as a proportion (pre-covid time travel / post-covid time travel). This motivated us to think if these differences could be due to differences in high-income and low-income areas. This was our motivating thought behind our hypothesis.

Null Hypothesis:
Median income by census tract is positively correlated with the proportion of pre-covid time traveled over post-covid time traveled.

Alternative Hypothesis:
Median income by census tract is not positively correlated (0 correlation or negative correlation) with the proportion of pre-covid time traveled over post-covid time traveled.

Building the Model

One of the toughest parts of the modeling was to build a data frame consisting of relevant features to run our model over. For our hypothesis, we wanted to understand how income level in a region affects the mean travel time. However, the dataset provided to us just contained the destination name and its associated geometry points and mean travel time from Hayes Valley. We worked cleverly around the data collection process:

1. We brainstormed on the granularity of the dataset that we need to merge our existing dataset with. Debating through several options like using plus_codes matrix for SF data through google plus_code or using clustering to divide our geometric points, we thought it would be best to work with the census tract because firstly, it is easier to extract census tract ID and the associated median income from government websites.
2. The next challenge was to find how to combine our existing table with destination name and geometry points with the new dataset containing census tract ID and its median income. In order to do this, we decided to leverage the MULTIPOLYGON geometry points and find a json dataset that contains geometry points for the census tract ID.
3. Now, we have three datasets:
    a. census_ready_to_merge: contains the census tract ID and the associated median income
    b. sf_geo : contains the census tract ID and geometry points (MULTIPOLYGON) dataset as geopandas
    c. pre_post: this is the cleaned dataset from our open EDA Part 1 where we organize the tract_to_times (given dataset) such that in our table, we merge the pre-covid (pre_named) dataframe next to the post-covid (post_named) side by side.

   After cleaning our column labels and values such that they are uniform across all dataframes, we perform merge and sjoin to combine our dataframe together.

Now, we have a finalized dataframe ‘match_census_pre_post’ that contains Destination’s Name, Movement ID, Geometry points, pre covid mean travel time, post covid mean travel time, census tract for the destination, and its associated household income.

In order to make a robust features table, we further use the speeds_to_tract dataset and merge it with match_census_per_post to get the latitude and longitude of the destination. With this, we finalize our ‘match_census_pre_post’ to have the following columns:
Destination’s name, movement ID, latitude, longitude, census tract, household income, pre-covid mean travel time, pre-covid lower bound for mean time travel, pre-covid upper bound for mean time travel, post-covid mean travel time, post-covid lower bound for mean time travel and post-covid upper bound for mean time travel.

Building the Baseline Model

In order to begin our modeling to see if the census tract median income was positively correlated with the proportional difference between pre-covid and post-covid travel times, we had to create a new column which displayed the proportional difference between pre-covid and post-covid times. To do this, we constructed a new column in our GeoDataFrame which took the pre-covid (left) times and post-covid (right) times, dividing pre-covid over the post-covid times.

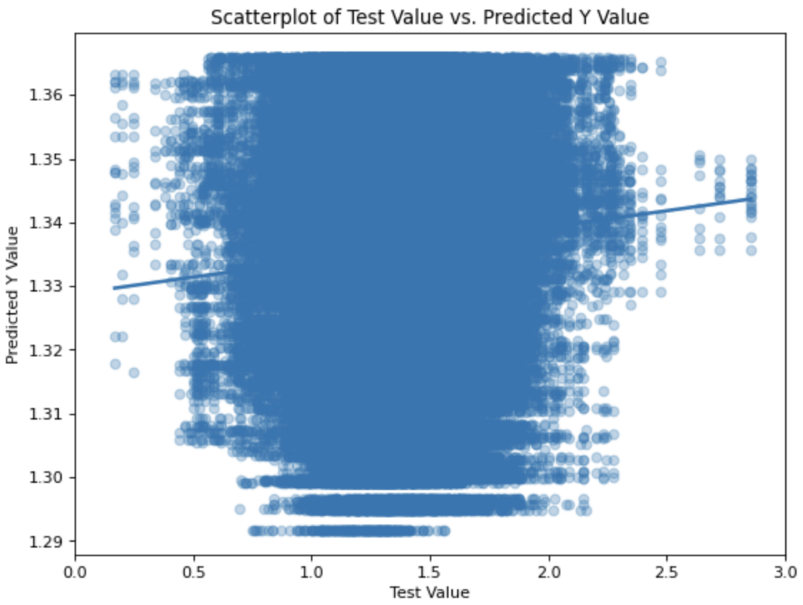
Taking the proportions of pre-covid and post -covid lockdown gave us a ratio expected above 1 for more or less all the rows in the table. This was because the pre-covid times were in general larger than the post-covid times. Our before/after stemmed from us thinking of using an after/before time proportion, as it would be more intuitive since the times had all decreased by a certain proportion. However, this would give us a negative correlation value with the incomes, since we were hypothesizing that as income increased, then the difference between the pre-covid and post-covid times increased. In order to show this accurately in comparison to each element in the dataset, we used pre-covid/post-covid travel times so we would expect a positive correlation with proportion and income.

To see if median income by census tract was a good predictor of the proportional difference between pre-covid and post-covid travel times, we began with a Simple Linear Regression model to estimate the sole effects of median income on the proportional difference. We started with our new table:

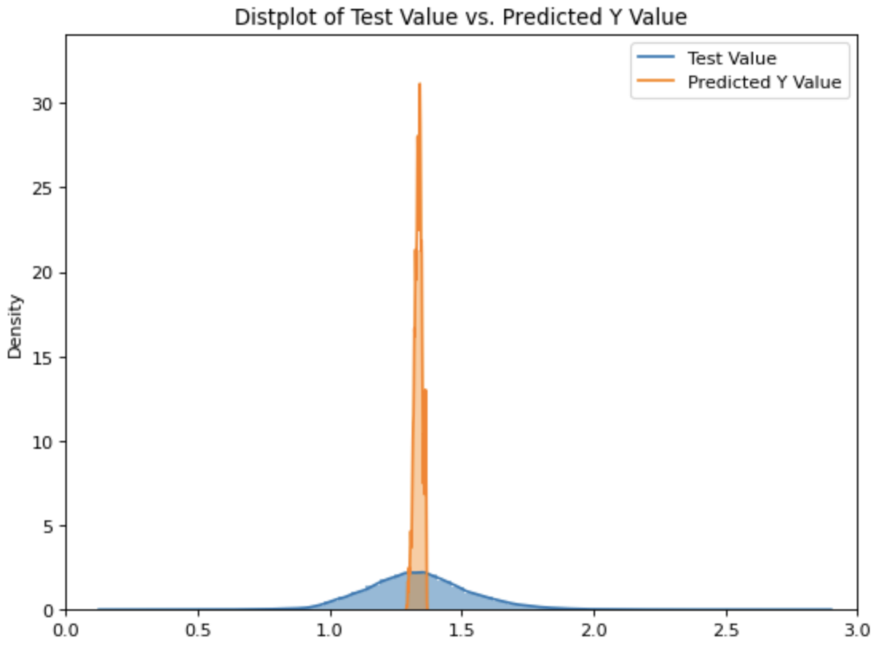
Selecting our 'Household Income' as our X and the 'proportion before/after' as our Y, we were able to develop a training and test split at a test size of 0.3 of the dataset. We further implemented the LinearRegression from sklearn to fit our data and evaluated the score on test and train sets. Our results were quite shocking and very interesting. Our scores on X test and y test were 0.00511, which was exceptionally low. This indicated that our model, through solely using median income, was extremely inaccurate in predicting the proportional difference.

Our hypothesis which was that median income by census tract is positively correlated with the proportion of pre-covid time traveled over post-covid time traveled, seemed to be moving towards the alternative that they are not well positively correlated. As our prediction score was very low, we decided instead, since this was a continuous variable being predicted, that a move in witnessing the mean squared error was more plausible. We then took the mean squared error and received a value of 0.04090.

In order to visualize what our linear regression was when comparing the test value with the predicted values, we plotted a scatter plot with our linear regression model running through it. Here we witnessed a very dispersed amount of data and a line that indicates a low level of correlation (not positively correlated as we had hypothesized).



In addition, we wanted to depict what the distribution of our predicted values was compared to the test values. We graphed each of the predicted and test value distributions and saw a massive difference. There is very little overlap between our Test Value and Predicted Y values as shown in the graph. From this distribution, it is evident that our baseline model consisting solely of median income households as a feature has a very low accuracy in predicting the mean travel time.



Due to our very low R-squared value, and a coefficient that is extremely close to 0, we have rejected our null hypothesis, that median income by census tract is positively correlated with the proportion of pre-covid time traveled over post-covid time traveled, in favor our our alternative hypothesis, that median income by census tract is not positively correlated (0 correlation or negative correlation) with the proportion of pre-covid time traveled over post-covid time traveled.

Improvements - Feature Engineering

Seeing the low R-squared prediction score for our model trained on just the median income of the destination, we reflected over our DATA 100 lectures and got inspired from our recent homework, Spam/Ham I where we attempted to build a model that gets us a 88% score. Leveraging our domain knowledge about time travelled, we decided to use feature engineering to pick relevant variable features from our dataset when creating our model. Looking at the our cleaned dataset, we brainstormed over several features and decided to add the following columns to our dataset:

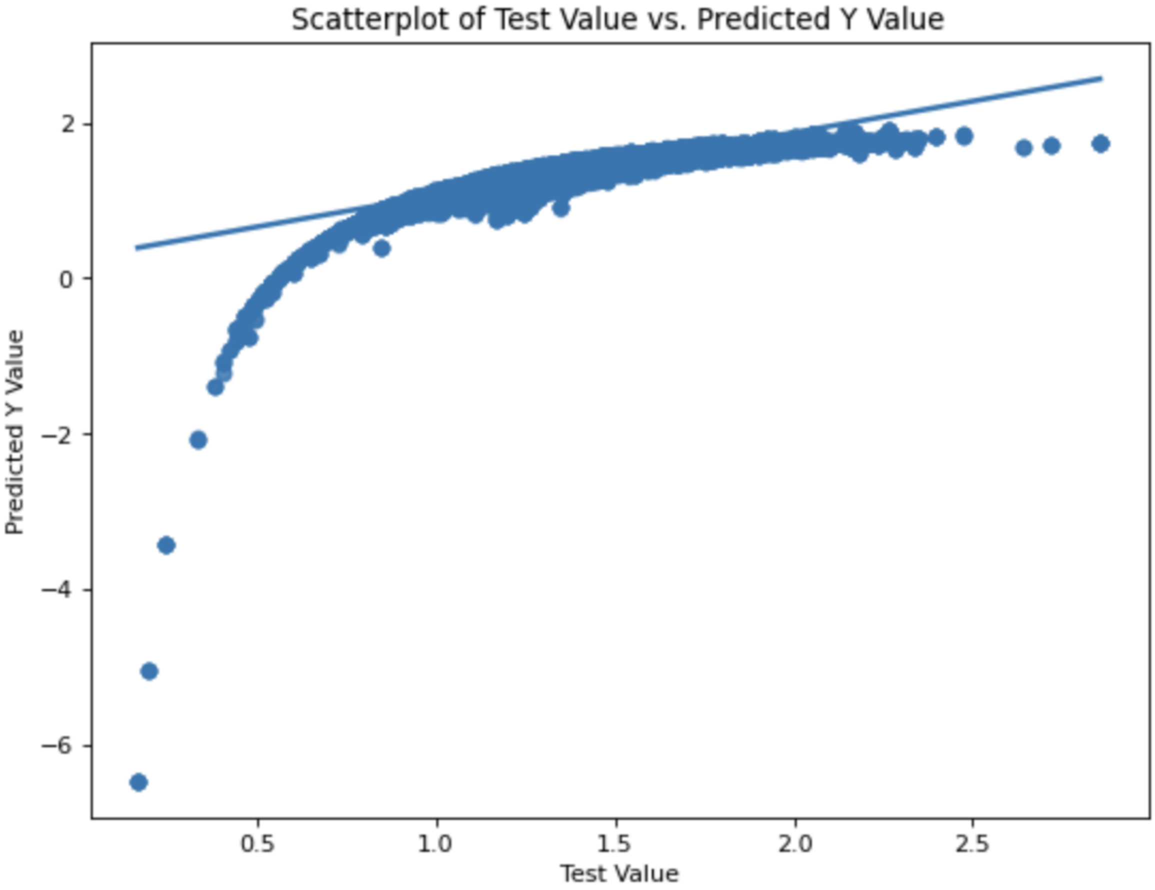
- Latitude and Longitude: In our original dataset, we are given the mean time travel to the specific destination. Roughly, looking over our proportions column, we observed that destinations that lie in the same census tract have more or less similar mean travel time. Therefore, we decided to include latitude and longitude as it makes it easier for the model to predict over these values. Additionally, latitude and longitude give us a sense of distance which is more intuitive to understand the travel time that is largely dependent.
- Range (Lower Bound Proportion): While the latitude and longitude were more intuitive features, we took inspiration from the google maps in order to understand what more features to use. While comparing time travel between different points, we noticed how google maps would first show the route which takes the least amount of time. Seeing how our given dataset comes from Uber, we decided to use the lower time range as the drivers would always prefer routes which have the shortest travel time. Also, having a lower range makes our model more susceptible to evaluating changes in situations like imposing COVID 19 lockdown where we see a drastic decrease in the time travel from our Open EDA Model in part 1. We create this column by dividing the pre-covid lower bound range with post-covid upper bound range.

In this effort to improve the model with feature engineering, we created a Multiple Linear Regression (MLR) model compared to Single Linear Regression (SLR) in our baseline model.

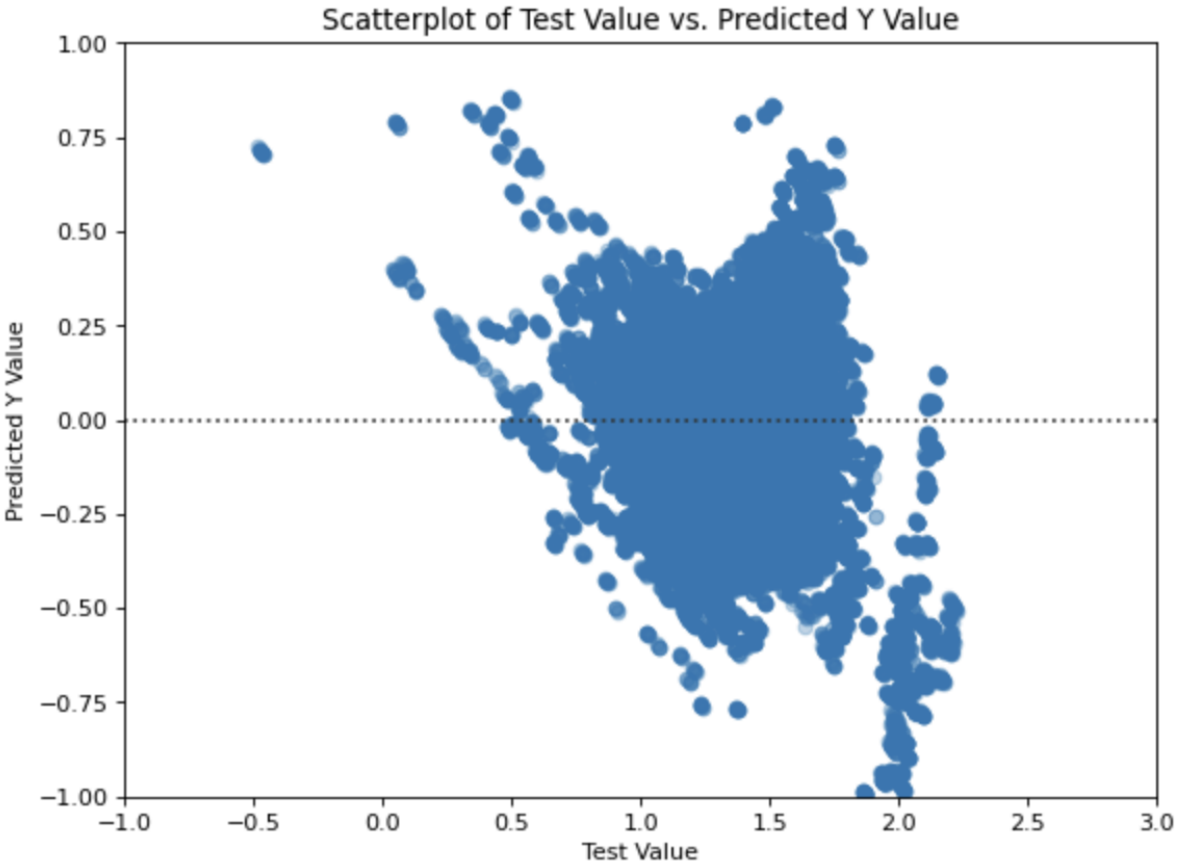
Our first attempt in feature engineering was selecting our features: 'Household Income' and 'Range (Lower Bound Proportion)' as our X and the 'proportion before/after' as our Y. We thought that these two features would be more relevant as it is the time travelled using the Uber dataset that we were trying to predict and from our understanding of uber algorithm, total time of the ride is extremely important in order to calculate several metric for the ride (like cost, driver availability etc.). After selecting the features, we were able to develop a training and test split at a test size of 0.3 of the dataset. We further implemented the LinearRegression from sklearn to fit our data and evaluated the score on test and train sets. Our results were consistent with what we expected as our scores on X test and y test were 0.23891, which was a massive jump from our previous value of 0.00511. This encouraged us to consider adding more features to our models and increasing our score.

This time we selected our features: 'Household Income', 'Latitude', 'Longitude' and 'Range (Lower Bound Proportion)' as our X and the 'proportion before/after' as our Y. Then, we were able to develop a training and test split at a test size of 0.3 of the dataset. We further implemented the LinearRegression from sklearn to fit our data and evaluated the score on test and train sets. Our results were quite shocking and very interesting. Our scores on X test and y test were 0.56418, which was a massive jump from our previous value 0.00511. This indicated that our model performed to give a better accuracy from applying feature engineering concept and was extremely useful in predicting the proportional difference. We then took the mean squared error and received a value of 0.01792.

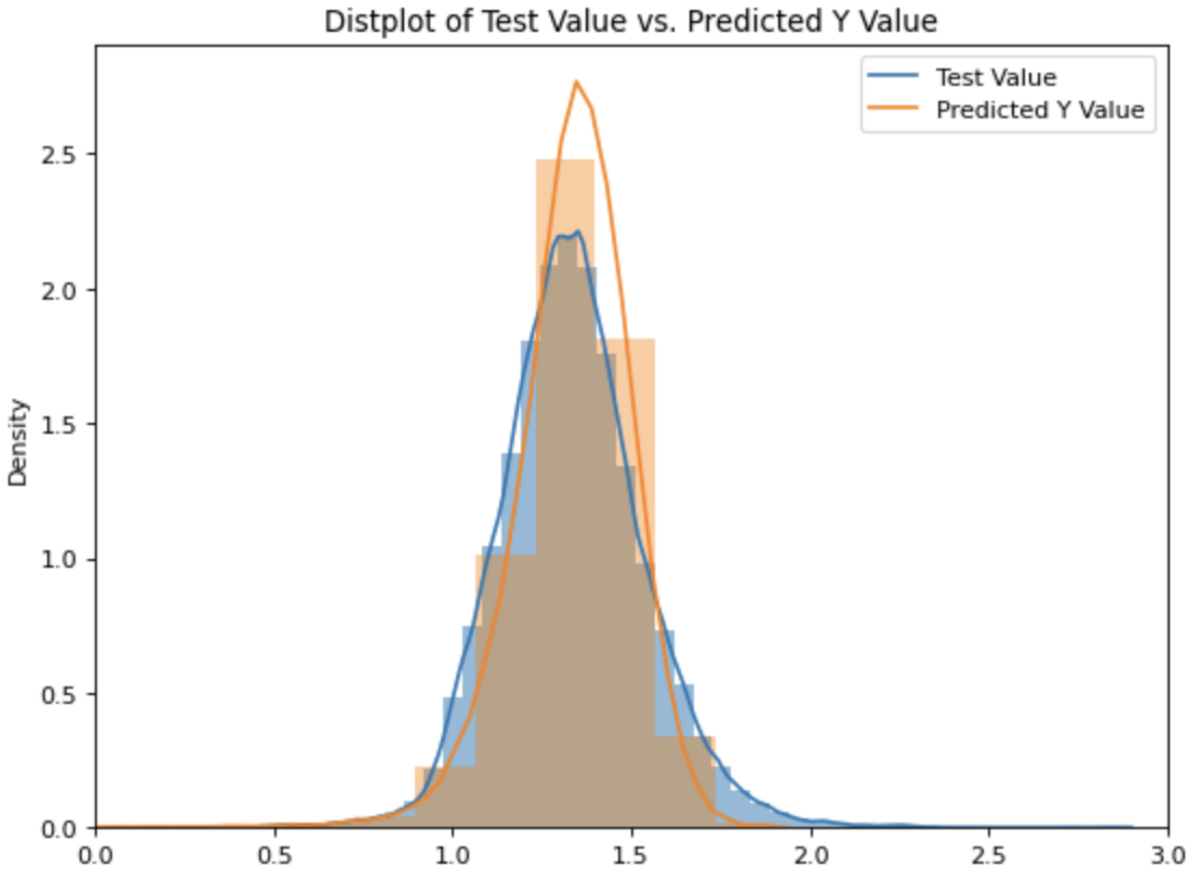
Similar to our visualization in the single regression model, we plotted a scatter plot for our new model comparing the test value with predicted values. From the graph below, we see that our visualization is representative of the score from the model. Hence, from this graph, we can see that there our test values are more consistent with our predicted y values.



To understand how far off each of the test values were from the predicted values in order to contextualize our mean squared error, we also created a residual plot of our data, which showed a large amount of scatter - indicating a better relationship. In this graph below, we can see that the points are extremely scattered and have no obvious pattern. The overplotting is due to the fact, we are plotting over 400000 data points on the graph.



In addition, we wanted to depict what the distribution of our predicted values was compared to the test values. We graphed each of the predicted and test value distributions and saw that there is a massive overlap between the Test Value and Predicted Y value. This shows us that our model has a good prediction accuracy.



Improvements & Additional Exploration

In our exploration for a better model, we realized that traffic on the weekend versus the weekdays has potential for differences. On top of this thought we wanted to see whether this was true in the pre-covid climate versus post-covid climate, since traffic dulled after the lockdown measures were announced. Therefore, could a model much more accurately predict the weekend or weekday in pre-covid or post-covid? Our expectation and hypothesis has pointed towards the pre-covid, since there was probably a much more clearer difference between weekday and weekend traffic. In our post-covid environment, people were more likely to work from home and that probably impacted weekday traffic times, making them closer to weekend traffic.

To begin our experiment, we would do a temporal split of our data into pre-covid and post-covid, which meant a split on the day 14. Therefore, we had our pre-covid dataset which was 1-13 and our post-covid dataset which ws from 14-31. We began first with our pre-covid dataset to manipulate how we will present our weekend versus weekday data. Therefore, we decided to one-hot encode our days. A 0 would indicate a weekday while a 1 would indicate a weekend. This would effectively convert our numbered days into the binary of weekend and weekday.

To predict if a certain day, based on it's mean time traveled, would be either a weekend or weekday, we decided on utilizing the Logistic Regression function. This function would help us learn which of the mean travel times would be a weekday or weekend. We would expect that our mean travel time would generally be greater on a weekday than on a weekend, due to increase in traffic. Therefore, we created a secondary hypothesis: pre-covid prediction score would be more accurate than our post-covid prediction score (the alternative being that it is the same accuracy or lower accuracy). With the current dataset we have, we can most definitely discover whether the pre-covid prediction score is more accurate than our post-cov id prediction score.

We began by first importing LogisticRegression from sklearn. Furthermore, we split the dataset into training and test sets. The training sets were fitted and we were able to score them on the test sets.

Our score was 0.76722.

We continued by creating a Logistic Regression model for post-covid. When we scored the model, we got 0.71147.

Therefore, this failed to reject our secondary hypothesis, that pre-covid prediction score would be more accurate than our post-covid prediction score (the alternative being that it is the same accuracy or lower accuracy).

We further used a the Lasso regression model, which takes into account any overfitting which occurred. Our Lasso regression model provided us with an R-squared of 0.45171.

Future Work

This project was very eye opening into the world of data science and machine learning. Segueing into the future, we would like to further explore our models and identify better ways to make them more accurate and reduce the mean squared error. One important aspect is including more features from additional datasets. We have hypothesized that using data such as demographic and racial makeup in different census tracts could potentially be a stronger indicator of a travel time or the proportion of travel time differences, as seen in reputable articles we have recently read.

Secondly, we would like to employ feature engineering in more ways than one. In our case, we would like to understand linearization more effectively, helping smoothen out the curves through squares and logs in a few of our scatterplots.

Lastly, we want to expand our modeling to other areas of interest. These include areas such as San Mateo, Contra Costa County, and Alameda County.

We are excited to take our project to new heights.