Data 100 Final Project Report - Traffic

Catalina Villouta, Priya Jindal, Vandana Keshavamurthy

Feedback + Reconsiderations

Based on the feedback we received on our design document, we made 2 additions to our experiment. Firstly, we added the day of the week as one of the features for our model, this was already something we explored in the EDA but did not think to include it in the model. Secondly, we appended an external dataset from the Uber Movement website, which includes all the routes starting from UC Berkeley for each day in March 2020. We did this in order to have additional data from a different geographical location to train our model with, make coefficient estimation more robust, and overall have a model that generalizes better. We also showed that our route binary features, although simple, are very powerful. For the most part we kept the same hypothesis that we proposed earlier since the feedback for that part was positive, but made sure to integrate additional features for creativity and enhancing performance.

Exploratory Data Analysis Conducted

For the EDA, we looked into how different routes were affected by COVID-19.

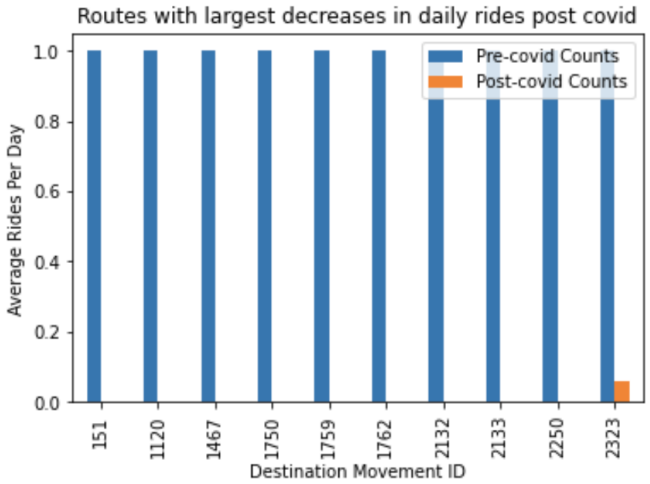

Figure 1. Routes with largest decreases in daily rides post covid.

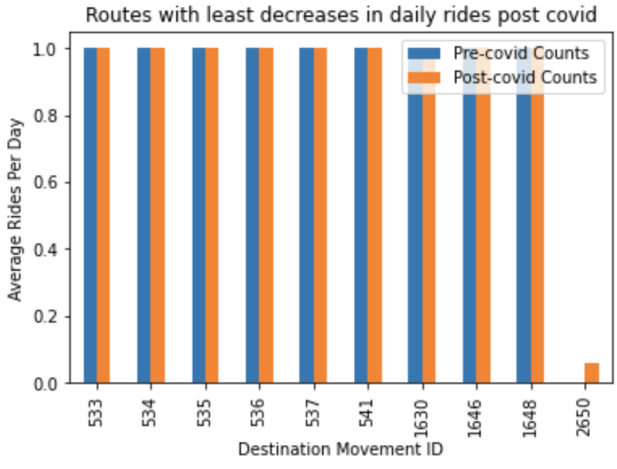

Figure 2. Routes with least decreases in daily rides post covid.

This bar plot shows the 10 routes (with their destination id's listed on the x axis), which had the largest differences in ridership frequencies pre and post covid lockdown. For 9 out of 10 of these routes, we can see that after covid there were no riders on that route.

This bar plot shows the 10 routes (with their destination id's listed on the x axis), which had the least differences in ridership frequencies pre and post covid lockdown. For 9 out of 10 of these we can see that there were no changes in average daily ride frequencies. However, for one

route to destination with movement ID 2650, the ridership actually went from no ridership to an increase after covid.

Overall, this helped us see how different routes were more or less impacted by the lockdown.

Open Ended Questions to Consider

1. What causes different routes to be impacted differently in terms of mean travel times pre and post covid? Maybe it is their locations near places of work and transportation hubs or areas of generally high traffic such as highways?

2. Are the changes in mean travel times pre and post covid due to changes in traffic only during rush hour but not the remaining hours in a day, or are the changes a reflection of every hour of the day?

Problem

Research Question: How is the range of travel times for any given day (difference between upper and lower bound travel times) impacted due to the covid lockdown?

Hypothesis: Lesser frequencies of rides on certain routes due to covid lockdown is likely to indicate a decrease in the difference between the lower bound and upper bound for travel times for any single day.

In our hypothesis, our 'X', is a computed feature which details whether the given day in March was during or before the lockdown. The 'Y' is the differences between upper and lower bounds of travel times for not only routes listed in the given dataset, but for routes in a different geographical area as well, which is obtained through another dataset and appended to our own 'Y' feature.

This hypothesis and its negation are both plausible. For example, it seems reasonable to anticipate a decrease in travel time ranges post covid as this would allude to traffic being constant throughout the day. This is explainable as sheltering due to the pandemic may have contributed to steep declines in rush hour to get to work, school, etc. However, the reverse may also be true: it is possible that the ranges actually stayed constant or even increased since people's priorities may no longer be to get to work on time, but rather get groceries as soon as a store opens to avoid large crowds, or visiting parks and play areas to get some fresh air. This is the main reason why we have our target listed as travel time range, rather than overall mean travel times; it is obvious that average travel times decreased during the lockdown, but it is not evident if the difference between the upper and lower bounds of travel times decreased as well, stayed the same, or increased. It is worth noticing at this point that the travel time range is directly related to the volatility of travel times. The higher the volatility of travel times, the higher the difference between the upper and lower bounds is expected to be. For this hypothesis, we will be considering features from the given dataset on San Francisco rides starting from Hayes

Street and an external data source with data regarding routes that begin in UC Berkeley to not only expand our current dataset, but to avoid overfitting our application to a specific geographical location.

Motivation for Hypothesis

For ridesharing companies such as Uber, the environment with traffic post covid is likely a situation which was neither seen nor anticipated prior to the pandemic. Gaining insight into the impacts of covid travel time is very relevant as it would allow us to make better forecasting models for these ridesharing companies, and offer a more perceptive view into their pricing strategies post covid.

When we were doing the EDA we noticed that there were steep declines in average travel time on March 14 (the day lockdown was announced) and March 17 (the day lockdown was issued). We also saw that many routes had much lower frequencies of ridership after the lockdown was in effect, indicating that there was less traffic, possibly because they were work commute routes. This previous statement is based on the decrease in days with data per route, since Uber does not release data regarding number of rides as far as we know. This led us to hypothesize that there is a correlation between the lockdown being issued and traffic amounts decreasing during rush hours, and therefore traffic throughout the day would be roughly equal, consequently leading the difference between upper bound of travel time and lower bound of travel time in a given day to be smaller as well. In other words, we were inclined to believe that rush hour (when everyone is commuting to work) is eliminated with the lockdown in effect, making the difference between the upper bound of travel time and lower bound of travel time to be smaller in that timeframe.

Testing the Hypothesis

We will test whether the lockdown measure had a statistically significant impact on the difference between the upper and lower bound of travel times using hypothesis testing as detailed below.

The hypothesis for this problem are as listed below:

- H_0: The differences between the upper and lower bounds of travel time do not change from pre to post covid.
- H_1: The differences between the upper and lower bounds of travel time *decrease* from pre to post covid.

For precisely stating our hypothesis, we will first show and explain our model. Our proposed model is the following:

$$log(y) = \sum_{i=0}^{\#\,routes} \omega_i \cdot route_i + \sum_{i=1}^{6} \lambda_i\, dayweek_i + \theta \cdot lockdown$$

Where:

- y is the difference between the upper and lower bound of travel time in a day
- $route_i$ is equal to 1 if trip (row) goes to routes i and 0 if not
- $dayweek_i$ is equal to 1 if trip (row) was on day of the week i and 0 if not. $i = 0$ represents Monday, $i = 6$ represents Sunday.
- $lockdown$ is equal to 1 if trip (row) was post-covid and 0 if not
- ω_i, θ and λ_i are the accompanying coefficients that will be found minimizing a loss function

We will also define the shrinking factor as follows:

$$shrinking\ factor\ =\ e^{\theta}$$

The shrinking factor is a factor correcting for lockdown, which is multiplied to the prediction when the trip (row) is post-lockdown only. If we are right, and the travel times range do decrease, then we would expect $\theta < 0$, and thus a $shrinking\ factor\ =\ e^{\theta} < 1$. The algebra behind this is as follows:

$$log(y)\ =\ \sum_{i=0}^{\#\ routes} \omega_i \cdot route_i + \sum_{i=1}^{6} \lambda_i\ dayweek_i\ +\ \theta\ \cdot lockdown$$

$$y\ = e^{\sum_{i=0}^{\#\ routes} \omega_i \cdot route_i + \sum_{i=1}^{6} \lambda_i\ dayweek_i + \theta\ \cdot lockdown}$$

$$y\ = y_{pre-lockdown}\, e^{\theta\ \cdot lockdown}$$

With

$$y_{pre-lockdown} = e^{\sum_{i=0}^{\#\ routes} \omega_i \cdot route_i} e^{\sum_{i=1}^{6} \lambda_i\ dayweek_i}$$

Notice that $y_{pre-lockdown}$ is the travel time range pre-lockdown assuming the model is correct. It should be easy to see now how the shrinking factor should be interpreted. For instance, a value of 0.8 is interpreted as a reduction of 20% in the travel time range. Note also that we can use the same shrinking factor definition for the day of the week variables. For instance, a Sunday shrinking factor would be equal to e^{λ_6}. If this value is equal to 0.9, it would mean that the travel time range tends to be 10% smaller on Sundays. Notice this variable is independent of the lockdown variable so the reduction in forecast on a Sunday would be proportionally the same for a pre or post-lockdown prediction, and would multiply further any lockdown reduction. We will come back to the interpretation of day of the week coefficients later.

Finally, having explained the proposed model in detail we can state our hypothesis more precisely.

The hypothesis for this problem are as listed below:

- $H_0: \theta = 0$
- $H_1: \theta < 0$

With a significance level set at 0.05 ($\alpha = 0.05$), the p-value linked to the corresponding t-test on θ can be computed and interpreted as follows:

- $p < \alpha = 0.05$
    - For this case, we reject the null hypothesis.
- $p >= \alpha = 0.05$
    - For this case, we fail to reject the null hypothesis.

Modeling

We aim to build a model to address our problem of predicting the difference in the upper and lower bounds of travel time in a day. With this model, we want to see if the difference of this range is truly smaller as we hypothesized due to rush hour (peak traffic hours when people are commuting to work) being eliminated due to the lockdown. In the section below, we will go into depth regarding the types of models we produced, and the methods used to produce these models.

Benchmark Model - Linear Regression Model on routes only

For our initial experimentation, we decided to use a linear regression model for our baseline model, since this will serve as a good benchmark and baseline model that we can improve on based on our results through feature engineering and other techniques. The benchmark linear regression model takes in an input feature of a one-hot encoding of the routes, i.e. each unique route as defined by its starting and ending node will have its own binary variable. Using these features, the model will output predictions for the difference between the upper and lower bound of average speed. The choice for the features was motivated by our EDA which demonstrated that different routes had various average travel times. This model basically computes the average target on the training data for each route. Any additional binary feature based on routes alone would be linearly dependent and thus superfluous. We also noticed that these binary features were very powerful in predicting average times as it will be shown later, thus we considered it would be a great baseline model to improve from.

Proposed Model - Linear Regression Model with all proposed features

Our final, proposed model is a linear regression based on the equation shown in the problem section. We stuck to a linear regression model for our final model, because we were able to identify improvements that helped us improve our baseline model. In addition to the route's binaries (one hot encoded route columns), it also includes a day of the week and a lockdown feature. Based on our EDA, we expected both features to be significant if we were forecasting average travel times. Since we are forecasting the difference between upper and lower travel times we will have to use data to test our model and our hypothesis, which we believe is not

obvious. The day of the week features were included since it allows us to differentiate between weekdays and weekends, which are important since traffic can vary between the two as observed in our EDA. One of the days of the week in the one hot encoded *day of the week* column was dropped to eliminate linear dependence (Monday). Since each day of the week starting Tuesday has its own feature, we can also capture other subtleties beyond just weekday and weekend patterns. It is important to highlight that even though we call this our *proposed model*, this model has already an improvement from the model we stated in our first delivery, and that is the inclusion of the *day of the week* feature. The feature *lockdown status* indicates whether the corresponding day of the observation is before (0) or after (1) sheltering in place orders due to the pandemic were put in place (March 17th). This feature helps substantiate our findings in our EDA of the disparities between pre and post lockdown traffic. We will go into further detail about how and why these features were chosen in the improvements section of this report. Using input features of one hot encoded route columns, day of the week, and lockdown status, the output for this model will be the same as the baseline, since we are still trying to predict the difference between the upper and lower bound of travel time.

Alternative Model - Random Forest

An alternative model that we also considered and trained was a Random Forest regression model. Random Forest classifiers and other tree based algorithms have been very popular and effective in Data Science efforts. We thought its regression counterpart, although less popular, would be interesting to explore. One important advantage of Random Forest regression is its ability to capture non-linear relationships between the variables. Additionally, Random Forest regression models can help in reducing the influence of outliers. We were interested in seeing whether this characteristic would help improve our overall predictions. The downside of this model is its added complexity and its propensity to overfit. Our decision to experiment with this model was motivated by the possibility that although our EDA presented generally linear trends for the average speed over time, perhaps this linear trend does not hold when it comes to the differences between the upper and lower bounds of speed. Additionally, since random forest regression models help in reducing the influence of outliers, we were interested in seeing whether this characteristic would help improve our overall predictions.

The input for this model consists of the same features included in our proposed model: one hot encoded *route*, one hot encoded *days of the week* and *lockdown status*. The one hot encoded *days of the week* feature provides information on what specific day of the week the observation is for. This column was included since it allows us to differentiate between weekdays and weekends, which is important since traffic can vary between the two as observed in our EDA and lockdown status was included due to our findings in the EDA, of disparities between pre and post lockdown traffic. The Random Forest regression model will use these features to output predictions for the difference between the upper and lower bound of travel time.

Model Analysis and Evaluation

Selected mechanisms of success

To evaluate our models we used 5-fold cross validation on R squared scores in our training set. We used R squared since it is a widely used and easy to understand metric. A value of 1 means the model is able to capture 100% of the variability in the data (perfect forecast) while a value of 0 means the model cannot improve using the average as the forecast. Keep in mind, that since we are using a cross-validated R squared then there is no guarantee that the metric will be above 0. We also calculated RMSE values for our models to use as a metric for success. RMSE is in the same units as y, which is log of travel time. We believe it is better to compute RMSE over the log of travel time difference, instead of travel time difference directly, because our target for different routes can be orders of magnitude apart. Thus, not using logs would end up biasing results into fitting correctly only the routes with the highest travel times difference overall. Using these two metrics we evaluated our models by looking for the one with the highest R squared and the least RMSE. These are appropriate evaluation metrics for our task and model since R Square provides a productive measure to determine how well the model fits the dependent variables, and RMSE allows us to better interpret how much the model's predictions differ from the actual differences of upper and lower bounds of travel times.

Benchmark model

For our benchmark model, we obtained an R squared score of 31.6% and a RMSE of 0.51 by using cross-validation.

There are two main forces determining the performance of this benchmark model. The first is that we are stacking routes of wildly different travel times together, and thus the difference between upper bound and lower bound of travel time should also be quite different between many routes. This helps in making the performance better. The second important factor is that we are not estimating the average of travel times, but the difference between the upper and the lower bound. Both these values are much more volatile than the average times, and therefore there is much more noise in this target as compared with using average time. This explains why we did not get a very high R squared value. If we were simply predicting the average time instead of the difference between travel time bounds, we would have obtained an R squared of 93.9%, which shows just how incredibly informative our routes binary features are. This is quite compelling evidence that our target variable is more noisy than average travel time, and thus makes our problem more challenging.

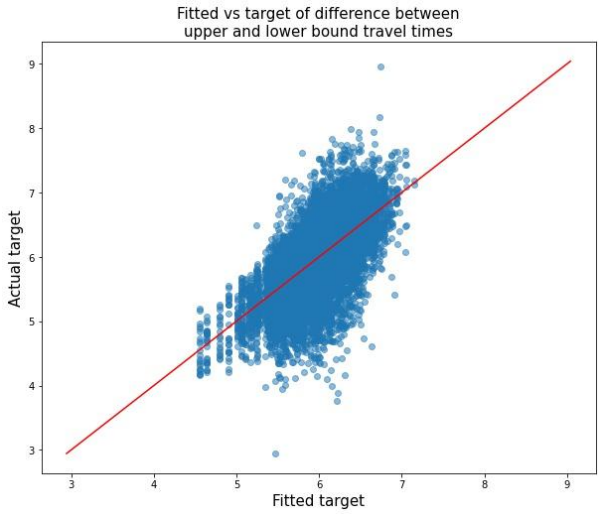

Figure 1. Fitted vs target of difference between upper and lower bound travel times using benchmark model

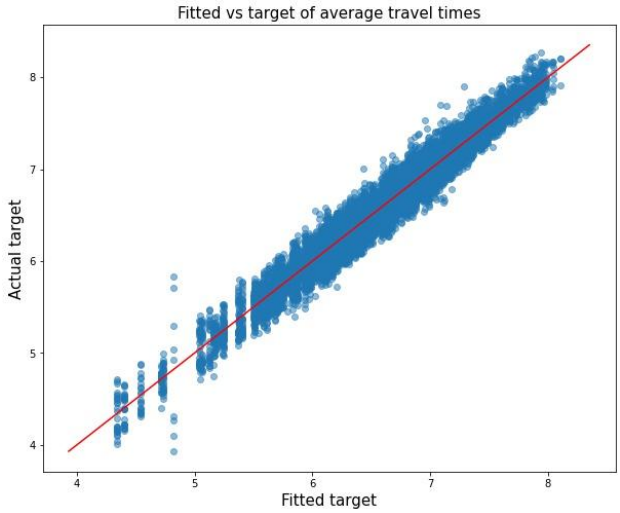

Figure 2. Fitted vs target of average travel times using benchmark model

The plots above are displayed to visually demonstrate the takeaway of how much easier it is for the model to better predict average travel times as opposed to the difference between upper and lower bounds of travel times. This provides a visual insight as to why the model may not be performing as well as hoped, and the difficulty of the problem that the model is being expected to solve for this project.

Although the results for this baseline model may look 'bad', the evaluations of this baseline model with just 2 features serves as our initial baseline R squared score and RMSE. We will add more features and improvements to our model in order to improve this baseline R squared and RMSE value.

Proposed Model

Our linear regression model using all features achieved an R squared of 51.1% and a RMSE of 0.43, which is a substantial improvement from the benchmark model performance of 31.6% and 0.51 respectively. Even though improvements on the Actual vs Fitted plots are subtle, the improvement achieved by this model over the benchmark model can be easily seen when comparing Figure 3 with Figure 1. Despite how much harder it is to forecast the travel time max-min diff, instead of say travel time averages, the improved performance of this model leads us to label the results of this model as 'good'. We will discuss the t-tests and p-values of the coefficients of this model in the Inference section.

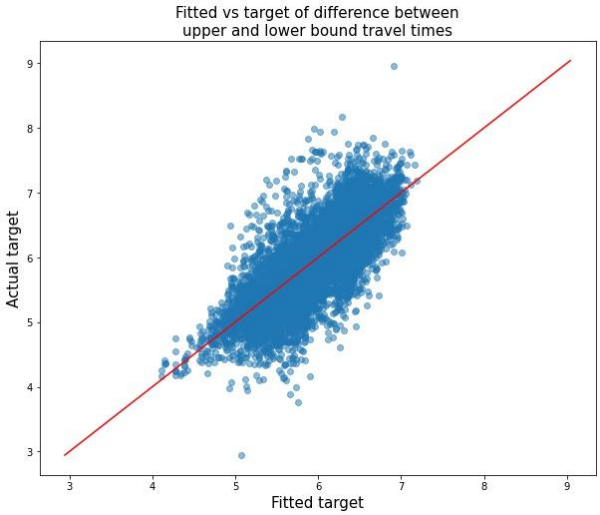

Figure 3. Fitted vs target of difference between upper and lower bound travel times, using proposed model

Alternative Model

For our Random Forest regression model, there are a number of hyperparameter values to choose. We decided to use a searching algorithm to determine an appropriate selection of values for two of the most critical hyperparameters in a Random Forest model: number of estimators (trees) and maximum depth (of each tree). We chose Bayes Search to determine the optimal hyperparameters instead of the more classical Grid Search which will try every single combination in the grid which can be extremely time consuming. Bayes Search is an efficient approach as it uses past hyperparameters and their performance to inform future searches, which avoids wasting time trying hyperparameters in areas of the grid where performance has been poor. Through cross validation, it was determined that the Random Forest regression model had an R squared value of 43.3% and a RMSE of 0.46. Even though the R squared score is lower and RMSE is higher than our proposed model, it is an improvement from our baseline model since the RMSE decreased and the R squared increased. This is clearly due to adding two additional features when training this model, as we can see that the linear regression model using the same information gets a significantly better performance in both metrics.

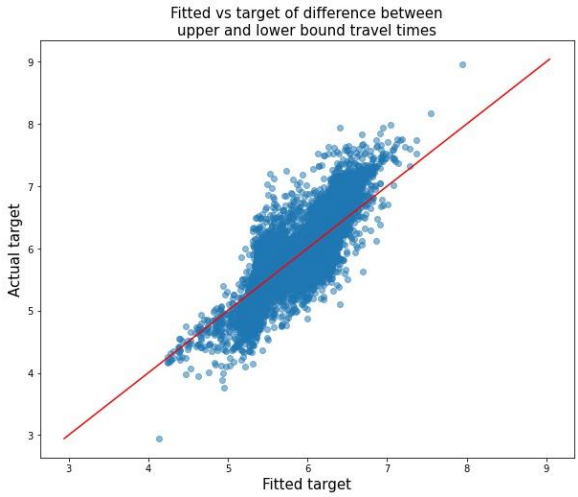

Figure 4. Fitted vs target of difference between upper and lower bound travel times using alternative model

An interesting output of tree based models like Random Forest is feature importance. Feature importance is defined as the decrease in node impurity achieved by a feature weighted by the probability of reaching that node. The higher the value the more important the feature. Even though this metric may sound abstract, it is a nice tool to find relative differences between feature importance in the variables used. In Figure 5 we can see that the lockdown feature is many times more important, as measured by the previous definition, than each individual day of the week variable.

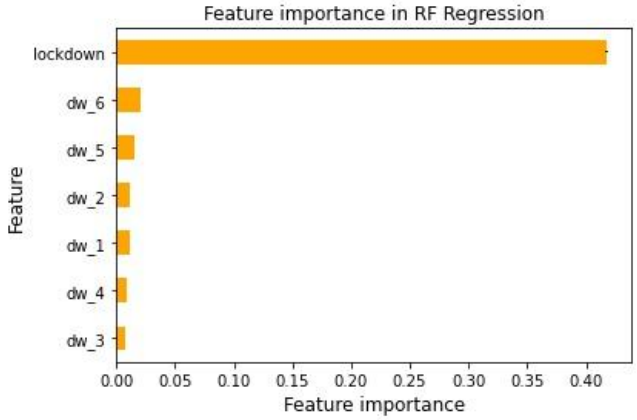

Figure 5. Feature importance in Random Forest regression model

Overall, this is not necessarily a “bad” result from this model because we are trying to predict the differences in upper and lower bounds of travel times and these differences vary greatly across our dataset since for all the different routes; this is only further pronounced with the external dataset we added which contains routes from another geographical area. Since what we are trying to predict is much more dispersed than just the average travel times, we expected

to see a lower R squared score and higher RMSE. Our later efforts will go into further improving the R squared score and RMSE, given the dispersed nature of the value we are trying to predict.

Model Improvements

Improvement 1 - Feature Engineering

Problem:
Initially, our benchmark model only included *one* hot encoded *routes*. Even though we have shown that these features are very informative and predictive, based on our exploration and our intuition, it is quite clear that there is important information we are leaving behind in this baseline model. This lack of sufficient information being fed into the model posed a problem of the model underfitting. This resulted in performance that can be considered as poor, as portrayed by our evaluation metrics with a very low r^2 score, and a very high RMSE, even when taking into account that our target is harder to predict than average travel time.

Solution:
To resolve this issue, we turned towards our findings from our EDA in that there are several other major factors that influence the differences between the upper and lower bounds of travel times in day other than just what route you are traveling to and from. Specifically, there were two features that were added which greatly improved the performance of the model: *day of the week* and *lockdown* status.

day of week:
From our EDA, it was observed how regardless of the day in march, there were significant differences in the traffic patterns and trends depending on the day of the week. This was most pronounced when looking at traffic trends on weekdays versus weekends. With this in mind, we computed a new column called *dw* consisting of numbers 0-6 corresponding to what day of the week it was based on one of the original features -- the *day* (1-31) it was in March 2020. 0 corresponds to Monday and 6 to Sunday. This column was subsequently one hot encoded into 7 columns and then included as a feature in our model, Monday was dropped in order to eliminate linear dependency. Thus, all coefficients should be interpreted as changes to Monday.

lockdown:
Through our EDA, we observed the prominent differences in traffic patterns before and after the lockdown was issued due to the pandemic. Specifically, there was a steep decline in average travel times on March 14 (the day lockdown was announced) and March 17 (the day lockdown was issued). With this insight in mind, we were motivated to compute a feature that would provide information regarding whether each observation in our data source was attributed to a day before or after the lockdown was issued. Thus, we made this feature equal to 0 from March 1 to March 16, and 1 from March 17 to March 31, the last date in our dataset. Thus, using our original features, we computed the *lockdown* feature which indicates whether or not the sheltering in place orders were issued for when the traffic datapoint was recorded.

Impact of Improvement 1:
Adding these two features to our model was a productive solution to our initial problem as it helped to greatly improve the model performance. Without this feature engineering, our original baseline model had a cross-validated R squared score of 31.6% and a RMSE of 0.51, but with the addition of these two features, the model improved with a higher R squared score of 51.1% and a lower RMSE of 0.43. The results of this experiment provide evidence that our intuition was correct: *day of week* and *lockdown* status are two key features that greatly influence the differences between the upper and lower bounds of average travel times for any given day. However, an unexpected side effect of this improvement was including a much higher number of features to train our model, which runs the risk of overfitting. This side effect was further investigated and addressed with regularization, which will be detailed in the next section.

Improvement 2 - Regularization

Problem:
As a side effect of improvement 1, our model so far has a fairly large number of features. We have 689 features corresponding to routes and 7 additional features corresponding to the lockdown status and Tuesday to Sunday binaries. Fortunately, we have 10,283 data points (rows), thus such a large number of features might not be an issue. Regardless, it is fair to state that we risk overfitting here.

Solution:
Thus, our second improvement was to add a regularization term to the loss function. The motivation is simple: regularization might help reduce the weight of unimportant features and consequently improve the out-of-sample performance of the model.

Impact of Improvement 2:
We used a Ridge regression model so that we can use L2 regularization. Since we only tuned one hyperparameter and Ridge regression runs very fast, this time we used Grid Search to find an optimal alpha hyperparameter among a reasonably granular set of choices. Then using cross validation, we determined that the cross-validated R squared score of this improved model is 53.4% and the RMSE is 0.42. This model had the highest R squared and the lowest RMSE value among all of our experiments. Our experimental results demonstrated that our intuition for using a regularization term to optimize our theta coefficients for prediction by minimizing the weight of less important features, was correct.

It is worth mentioning that in general it is considered best practice to standardize the features before using Ridge regression. The reason is that if features are orders of magnitude different between each other, then coefficients will tend to be so too, and then regularization would be applied differently between variables. However, it is also important to note that all our variables are binaries. Thus, we do not have the issue mentioned above. In fact, it is usually recommended *not to standardize* binary variables. Furthermore, using a min max scaler for instance would leave the variables *unchanged*.

Summary of Results with Training and Cross Validation Sets

In Figure 6 and Figure 7 we can see the *train* and cross-validated metrics. We call train results here to the metrics computed over the training set when fitted over the whole training set. By contrast, the cross-validated metrics are computed by holding out, five times, one subset of the training set for computing the metric and fitting the model over all other data points in the training set, and finally averaging the five computations. Cross-validated metrics should be much better reflections of the out-of-sample performance than the training metrics. In fact, we can see that the training R squared and RMSE are much higher than the cross-validated metrics. This is a sign of overfitting. As we have done up to this point already, cross-validated metrics are the ones we will consider for model selection, and training metrics will be shown but discarded for decision-making.

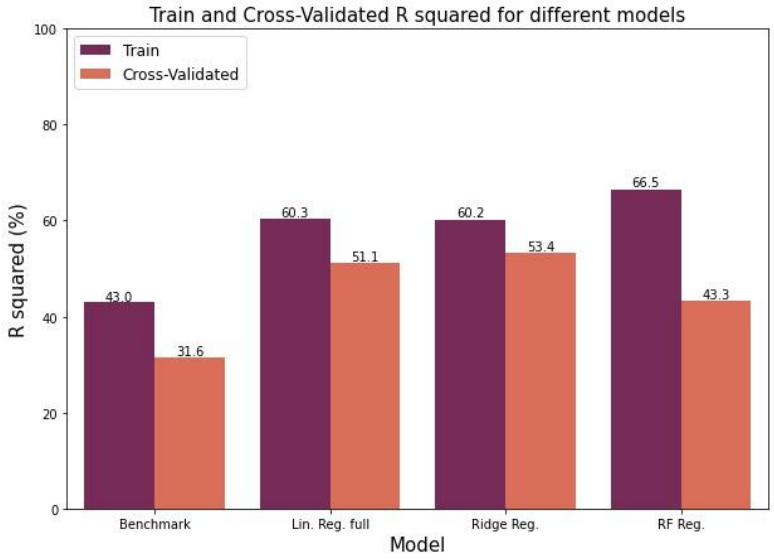

Figure 6. Train and Cross-Validated R squared for different models

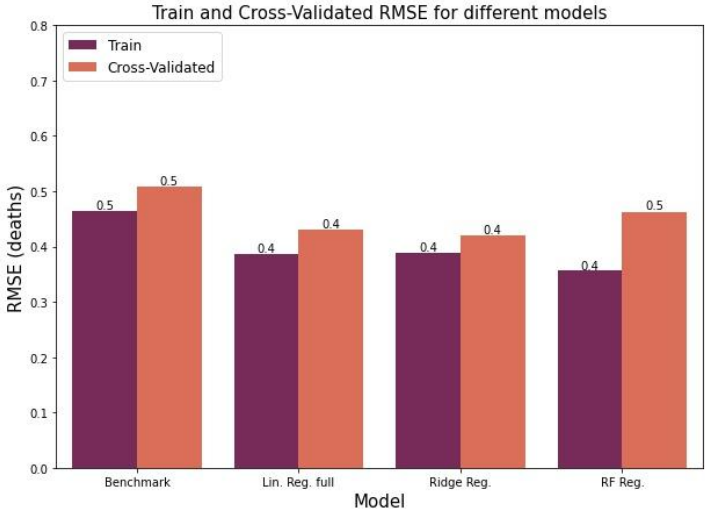

Figure 7. Train and Cross-Validated RMSE for different models

Overall Analysis After Improvements:

As seen in Figure 6 and Figure 7, Ridge regression model was our best model in terms of both maximizing the R squared and minimizing the RMSE. Since model selection should be based on training/cross-validated results alone, we can say at this point that our chosen model is Ridge regression using all considered features.

Ridge regression performance in the test data ended up being similar to the one obtained using cross-validation. We got a test R squared of 53.5% and a RMSE of 0.41 as compared with a cross-validated R squared of 53.4% and a RMSE of 0.42. This is evidence that our model has good out-of-sample forecasting capabilities. It also shows that cross-validation is a very useful methodology for getting a more realistic out-of-sample inspired metric.

We can also see that the Ridge model had a significantly better performance in the test set than the benchmark model (53.5% vs 33.4%), and a marginal improvement over the linear regression model using the same features (53.5% vs 52.9%). Similar to our cross-validation conclusions, our implementation of Random Forest regression was better than our baseline model, and worse than both the linear regression and Ridge regression.

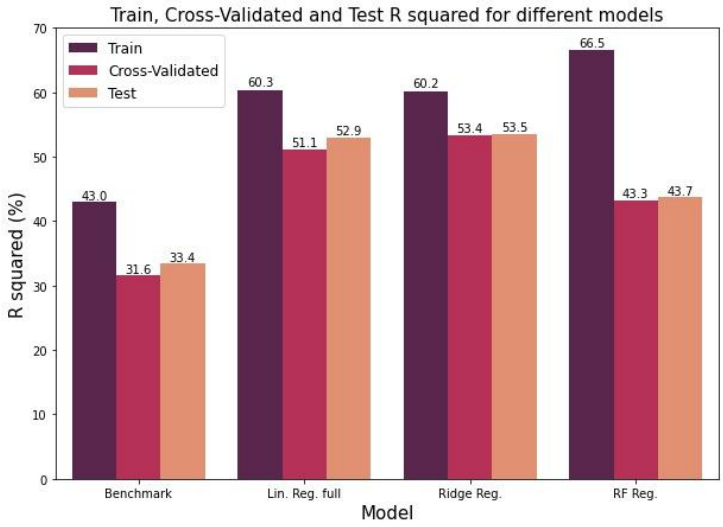

Figure 8. Train, Cross-Validated and Test R squared for different models

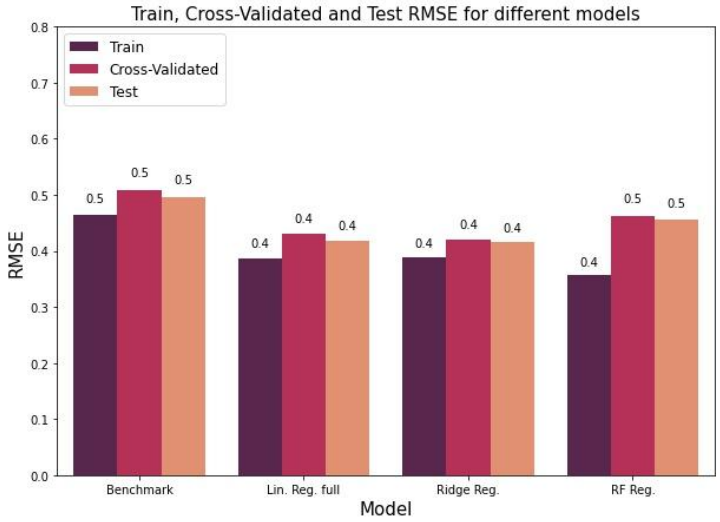

Figure 9. Train, Cross-Validated and Test RMSE for different models

We also performed a backtest which is inspired on the temporal nature of the problem. In this type of problem, obtaining a test set at random is not the best practice because we would be using in most cases information from the future and the past to forecast anything in between. What should be done is to use only the past data to forecast the days following. In this backtest, we see what would have been the result if we had used our Ridge model trained only on past data up to day 16 to forecast 17, and from there using all past information to forecast the next day, i.e. we then use information up to day 17 to forecast 18, then up to 18 to forecast 19, and so on.

Figure 10 was created from the R squared values of the predictions for days in March after the lockdown in March. We can see that it was very high despite how much harder it is to forecast the travel time max-min difference, instead of say travel time averages. R squared ranges from around 50% to around 70%, which we believe is quite remarkable.

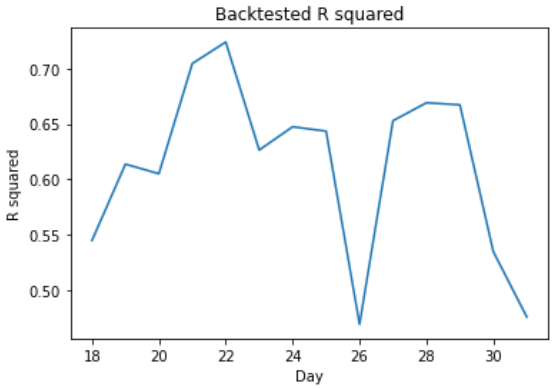

Figure 10. Backtested R squared

Finally, we show the shrinking factor computed, which is equal to $exp(\theta)$ and can be interpreted as the factor that multiplies the forecast pre-lockdown to get the forecast post-lockdown. Given that our training set in this exercise is an expanding window and that the first run had only one full day of lockdown, is that the values changed more in the first days post-lockdown and tended to stabilize after around 1 week of data. Nevertheless, we can see that the estimation of the shrinking factor with only one day of data (0.49) was not too far away from the shrinking factor computed using all the data (0.54). This means that with one day of lockdown data we would have computed a 51% decrease in regular forecasts, while with 2 weeks of lockdown data we would be calculating a 46% decrease. Thus it would have been a very reasonable adjustment to apply in the early days of the lockdown given the urgent situation the company was facing.

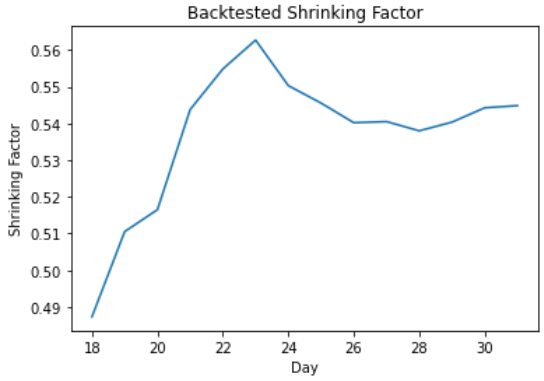

Figure 11. Backtested Shrinking Factor

Inference

Now we will check whether lockdown is significant, which was our research question and was the hypothesis to test. We can additionally test whether the days of the week features are significant as well. Using stats models we found out the t-statistic and p-value of the mentioned coefficients. The lockdown variable is immensely significant with a t-statistic of -62 and a p-value of virtually 0 in the OLS case. Given the high similarity between OLS and Ridge, and the outstanding significance, it is reasonable to conclude that the theta coefficient is also highly significant in the Ridge model. **Essentially, we can be sure that the lockdown had a significant impact on travel times range. We therefore *reject* the null hypothesis that the lockdown had no impact on the range of travel times, i.e. we reject $H_0: \theta = 0$ and accept the alternative hypothesis.**

A strong reduction of the difference between the maximum and the minimum is a clear indication that the dispersion of travel times was reduced greatly. There is a very simple cause for this reduction in the travel time range or dispersion: the streets had significantly less traffic. Besides just saying that the reduction in the max-min difference was significant, we can also quantify this impact which can be very useful for the real situation faced by Uber. We found an optimal value of theta (in the train data) of -0.61, which translates into a shrinking factor of 0.54.

This means that based on our best estimation, travel times max-min difference was reduced by 46% due to the lockdown measure.

Table 1. Linear regression coefficient results

	coef	std err	t	P>|t|	[0.025	0.975]
lockdown	-0.5954	0.010	-62.044	0.000	-0.614	-0.577
dw_1	0.1063	0.015	7.333	0.000	0.078	0.135
dw_2	0.0660	0.015	4.448	0.000	0.037	0.095
dw_3	0.0803	0.015	5.395	0.000	0.051	0.110
dw_4	-0.0017	0.015	-0.114	0.910	-0.031	0.027
dw_5	-0.1268	0.015	-8.375	0.000	-0.156	-0.097
dw_6	-0.1662	0.014	-11.801	0.000	-0.194	-0.139

The days of the week are also significant. We dropped the binary variable linked to Mondays to avoid linear dependency in the columns of the design matrix. We see that travel times range on Tuesdays to Thursdays tend to be higher than on Mondays, while weekends see a significant reduction when compared to Mondays. When using our now usual exponential transformation to coefficients to get the factor, we get an 11% increase for Tuesdays, a 7% increase for Wednesdays, an 8% increase for Tursdays, no significant change to Fridays, a 12% *decrease* for Saturdays, and a 15% *decrease* for Sundays; all when compared to Mondays. All weekday adjustments are highly significant as seen from both the t-stats and the p-values in Table 1, with the exception of Fridays which has a t-statistic of -0.1 and a p-value of 91%.

With these explanations and the results from our statistical analysis, we are able to confirm our initial hypothesis that the pandemic lockdown likely indicated a decrease in the difference between the lower bound and upper bound for travel times for any single day.

Future Work

In this project, our research question revolved around studying how the range of travel times would vary post lockdown. We have concluded that we can confirm our initial hypothesis that the travel time range was significantly reduced due to the lockdown and we built a model capable of accurately adjusting upper and lower differences in travel time, with a final estimation of a 46% reduction.

During our EDA we discovered that different routes were impacted differently post-lockdown — for example, some of the routes no longer had any rides post lockdown while some had an increasing number of average daily rides. This was taken into account in our project and in creating adequate models to robustly answer our hypothesis. We can point out however, that we had only one feature based on lockdown status, thus we have a universal estimation of travel time range reduction. This means that our estimation of travel time range reduction due to

lockdown status is not dependent on the day of the week, or the route, or any other geographical consideration. We have only one robust number, which in future work could be enriched with different interactions in order to answer other relevant questions. Was travel time volatility affected more on days of the week? In rush hour? On bridges? On highways? In each city or census tract?

For more future work, it would be interesting to see how grouping together routes geographically based on proximity, i.e. neighborhoods or districts impacts the lower and upper bounds of travel times for a given day, and potentially boost the R squared score for a predictive model. This exploration would also provide insight into a more focused perspective of the types of common characteristics that geographical groupings, with increases and decreases in ridership post covid, would have. For example, perhaps groups of routes near transportation hubs or areas of generally high traffic such as highways observed constant rates of ridership pre and post covid.

Based on our intuition and our work, we believe that travel time range was indeed reduced *more* on weekdays, in rush hour, on bridges, and in busy cities, but it would be interesting to analyze and quantify the differences. Just to be clear, we do know that day of the week features are important in forecasting travel time range, we just do not know how they *interact* with the lockdown status.