**1. Introduction**

New York City taxi rides paint a vibrant picture of life in the city. The millions of rides taken each month can provide insight into traffic patterns, road blockage, or large-scale events that attract many New Yorkers. With ridesharing apps gaining popularity, it is increasingly important for taxi companies to provide visibility to their estimated fare and ride duration, since the competing apps provide these metrics upfront. Predicting duration of a ride can help passengers decide when is the optimal time to start their commute, or help drivers decide which of two potential rides will be more profitable, for example. In order to predict duration, only data which would be available at the beginning of a ride was used. This includes pickup and drop-off coordinates, trip distance, start time, number of passengers etc. Linear regression with model selection, ridge, lasso, decision tree, random forest and XGBooster models were used to predict duration and fare amount.

**2. Related Work**

One way to predict duration is by doing short term prediction with the help of real time data collection. In [1] the authors tackle the problem by using data from buses (GPS) and an algorithm based on Kalman filters. Using a similar approach, [2] uses real time data from smartphone placed

inside vehicles. Estimating travel time for highways yields better results than in the cities. This allows for more accurate predictions. In [3] the authors use a combination of traffic modelling, real

time data analysis and traffic history to predict travel time in congested freeways. They try to overcome the assumption that real time analysis communication is instantaneous. A lot of other papers also work on freeways. In [4] the prediction is done using Support Vector Regression (SVR) while in [4] Neural Networks (SSNN) are used. Predictive estimates of future transit times is a feature that was released in 2015 in the Google Maps API [4]. This shows the importance of being able to predict time travel without having real time data of traffic. We are trying to solve a similar problem: estimating ride duration without real time data, by analysing data collected from taxis. Being able to do such estimation would help making better future predictions.

**3. Data**

The data used in this study are all subsets of New York City Taxi and Limousine Commission’s trip data, which contains observations on around 14 lakhs taxi rides in New York City between Jan and June 2016**.** Since each month consists of about 2.3 lakhs observations, and there were computational limitations, subsets of the monthly data were used for model building, and other subsets were used for validation. To build the models, splitting the data, 65% used for training and 35% validation used for validation.

The original dataset contains features as pickup and drop-off locations, as longitude and latitude coordinates, time and date of pickup and drop-off passenger count. The data was processed to extract separate features for year, month, day, weekday, hour and minute from the date and time of each ride, as well as trip duration as the difference between drop-off and pickup time. Furthermore, with the objective to model and account for traffic in the predictions, two more features were calculated from the data; distance and speed. Distance represents the difference between pickup and drop-off location and the speed represents the speed of each ride.

Figure 1 show the distributions of trip duration. The objective of this study has been to predict trip duration, although as the results and models chosen are very similar, the illustrations and results have been focused on the prediction of trip duration.

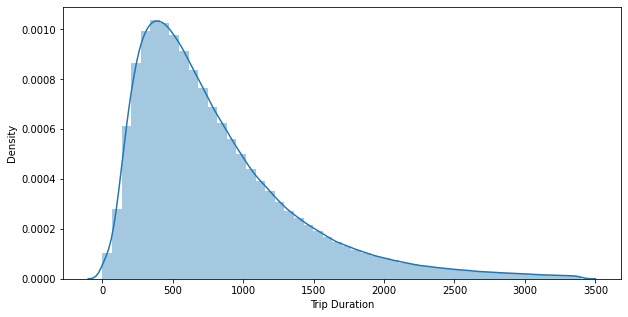


Figure 1: Duration Distribution

**4. Models and Methodology**

**4.1 Linear Regression**

As a baseline prediction, the mean duration and fare from the training set were used to predict a constant value for the validation set.

**4.2** **Ridge and Lasso Regression**

### Regularized linear regression models are very similar to least squares, except that the coefficients are estimated by minimizing a slightly different objective function. we **minimize the sum of RSS and a "penalty term"** that penalizes coefficient size.

### **Ridge regression** (or "L2 regularization") minimizes:

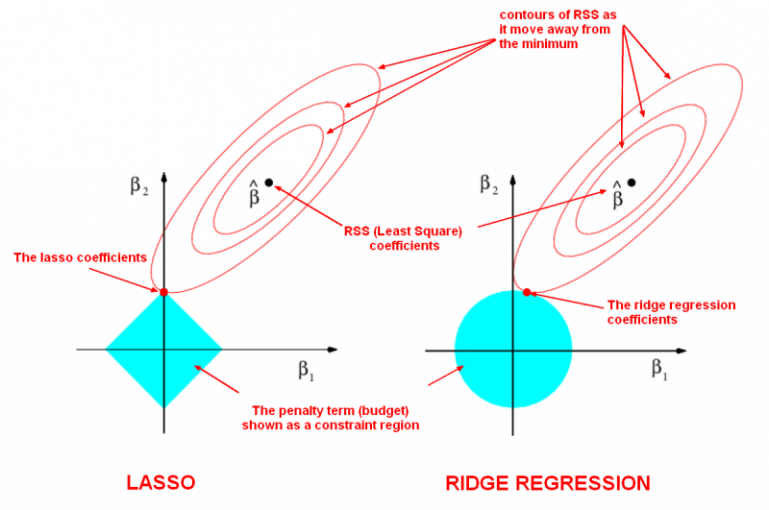
### **Lasso regression** (or "L1 regularization") minimizes:

Where λ is a **tuning parameter** that seeks to balance between the fit of the model to the data and the magnitude of the model's coefficients:

* A tiny λ imposes no penalty on the coefficient size, and is equivalent to a normal linear regression.
* Increasing λ penalizes the coefficients and thus shrinks them towards zero.
* Lasso stands for least absolute shrinkage and selection operator

### Thus, you can think of it as, we're balancing two things to measure the model's total quality. The RSS, measures how well the model is going to fit the data, and then the magnitude of the coefficients, which can be problematic if they become too big.

### **Visualizing Lasso & Ridge Regularization**

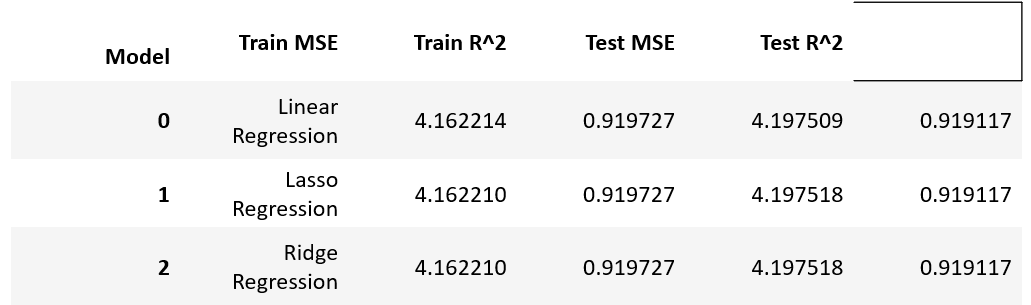


* **Lasso regression** shrinks coefficients all the way to zero, thus removing them from the model.
* **Ridge regression** shrinks coefficients toward zero, but they rarely reach zero.
* To get a sense of why this is happening, the visualization below depicts what happens when we apply the two different regularizations. The general idea is that we are restricting the allowed values of our coefficients to a certain "region" and within that region, we wish to find the coefficients that result in the best model.

In this diagram, we are fitting a linear regression model with two features, x1 and x2.

* β^ represents the set of two coefficients, β1 and β2, which minimize the RSS for the **unregularized model**.
* The ellipses that are centred around β^ represent **regions of constant RSS**. In other words, all of the points on a given ellipse **share** a common value of the RSS, despite the fact that they may have different values for β1 and β2. As the ellipses expand away from the least square’s coefficient estimates, the RSS increases.
* Regularization restricts the allowed positions of β^ to the **blue constraint region**. In this case, β^ is **not** within the blue constraint region. Thus, we need to move β^ until it intersects the blue region, while increasing the RSS as little as possible.
* For **ridge**, this region is a **circle** because it constrains the square of the coefficients. Thus the intersection will not generally occur on an axis, and so the coefficient estimates will be typically be non-zero.
* For **lasso**, this region is a **diamond** because it constrains the absolute value of the coefficients. Because the constraint has corners at each of the axes, and so the ellipse will often intersect the constraint region at an axis. When this occurs, one of the coefficients will equal zero. In higher dimensions, many of the coefficient estimates may equal zero simultaneously. In the figure above, the intersection occurs at β1=0, and so the resulting model will only include β2.
* The **size of the blue constraint region** is determined by λ, with a smaller λ resulting in a larger region:
* When λ is zero, the blue region is infinitely large, and thus the coefficient sizes are not constrained.
* When λ increases, the blue region gets smaller and smaller.

**5. Prediction Results**

****

**6. Conclusions**

* We can see that MSE and R2 which are the metrics used to evaluate the performance of regression model of Linear Regression, Lasso, Ridge
* The Linear models show good performance on our training and testing environment

From above table we can conclude that Linear Model gives us R2=91%. Which is the best model as compare to the other models to predict trip duration for a NYC taxi

**References**

1. Vanajakshi, L., S. C. Subramanian, and R. Sivanandan. "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses." IET intelligent transport systems 3.1 (2009): 1-9.

2. Biagioni, James, et al. "Easy tracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. “Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems. ACM, 2011.

3. Yildirimoglu, Mehmet, and Nikolas Geroliminis. "Experienced travel time prediction for congested freeways. Transportation Research Part B: Methodological 53 (2013): 45-63.