1. Model Objective

For models in this project, we are trying to achieve two goals: generating models to predict total price accurately and finding the important features for the following recommendation in Module 3. With the confident pricing model for both Uber and Lyft, we can compare price differences between two companies' prices and make recommendations for the operational team, this will be deeply discussed in the next module.

2. Data Preparation

Due to the large original dataset and computer limitations, we choose zone 61 in the pick-up location for our training dataset, reducing from 11 million rows to 150k rows. Based on the analysis from module 1, we find zone 61 in the pick-up location has the highest number in the dataset.

We also redefine the total price variable, it can be expressed as the following equation:

$$\begin{aligned} Total\ Price &= Base\ Passenger\ Fare + Tolls + bcf(mandatory\ fund\ charge) \\ &+ Sales\ Tax + Congestion\ Surcharge \end{aligned}$$

Additionally, in order to investigate the time influence for the dependent variable, we split the *'request DateTime'* variable into *hour, day, day of week* variables.

3. Pricing Models

We would like to utilize these models to reasonably predict the total price that the passengers should pay based on trip miles, weather conditions, datetime, etc. Figure 1 illustrates the independent variables that we use for the pricing models.

Since we want to predict the total price, which is a continuous variable, we choose *regression* models rather than classification models. In the following sections, we will discuss the model that we choose, the principle of the model, the tuning process, and model performance.

3.1. Linear Regression

Linear regression is a model to find a linear equation that best describes the correlation of the independent variables with the dependent variable by fitting a line to the data using least squares. We choose this model because it is the simplest model for regression, and it is also very quick and efficient.

In this model, we have not done any hyperparameters tuning since it does not have any hyperparameters. Besides that, the limitation of using this model is that we assume the linearity between dependent and independent variables.

After training the model, we get 21.75 for MSE (mean square error) and 0.89 for OSR (out of sampleR^2).

3.2. CART

A CART model is easier to interpret than a linear regression. It will select the split that decreases the total impurity the most. For Regression, the best split points are chosen by reducing the squared or absolute errors. However, decision trees may not be used well with continuous numerical variables, and *overfitting* is one of the practical difficulties for decision tree models. It happens when the learning algorithm continues developing hypotheses that reduce the training set error but at the cost of increasing test set error. But this issue can be resolved by *tree pruning* and setting constraints on the *complexity parameter (cp)*.

From the result metrics, we can see that the Out-of-sample R-Square 0.87, and the Out-of-sample MSE is 25. The result decision tree is shown in Figure 2.

3.3. Random Forest

The principle of random forest is to build decision trees on different samples and take their average in regression. Because of the high efficiency in dealing with a large dataset, we choose this model. However, one of the limitations of the model is the difficulty in interpreting the parameters.

We utilize the ***RandomizedSearchCV*** for hyperparameter tuning in this model, and the hyperparameters that we search can be found in Figure 3. After the tuning, decided by the best OSR scoring, we find the best hyperparameters are {'n_estimators': 400, 'min_samples_split': 10, 'min_samples_leaf': 2, 'max_depth': 30, 'bootstrap': True}. It gets 23.75 in MSE and 0.88 in OSR.

3.4. Support Vector Machines (SVM) Regression

SVM Regression works by finding the best fitting hyperplane to the data points such that the margin between the hyperplane and the closest data points is maximized. Comparatively, SVM is based on the geometrical properties of the data while logistic regression is based on statistical approaches.

A significant limitation of SVM Regression is that it is computationally expensive, especially when the number of input variables or the size of the dataset is large. This is because SVM Regression involves solving a quadratic optimization problem that can be time-consuming for large datasets.

Support Vector Machines Regression can be tuned by using “GridSearchCV.” However, as the limitation mentioned above, SVM Regression is computationally expensive while our dataset has various features, it is hard to run the tuning process in this case with the given technique conditions. Even if we cannot process the whole tuning process, the model is tried with different Kernel types: Linear and Radial Basis Function(RBF), which are the two most commonly used hyperparameters for SVM Regression. Linear gives us a better result showing below:

MSE: 24.12

R-Square: 0.8662

3.5. Bayesian Regression

Bayesian Regression combines prior information about the model parameters with observed data to update and obtain a posterior distribution of the model parameters, which is used to make predictions while accounting for the uncertainty in the estimated parameters.

It is a good choice for this dataset because it can handle a large number of features and works well with high-dimensional data. Also, it avoids overfitting by using a prior distribution on the model parameters.

We utilize "RandomizedSearchCV" to perform a random search of the hyperparameter space defined in "param_dist" for the Bayesian Regression model. It gives the given distributions to randomly sample hyperparameters to try during the search. The best hyperparameter is shown below:

Best hyperparameters: {'alpha_1': 5.7158e-09, 'alpha_2': 5.375e-08, 'lambda_1': 1.290e-09, 'lambda_2': 9.991e-07, 'n_iter': 100, 'tol': 4.796e-08}

Using the best hyperparameters, we get the result:

MSE: 22.99

R-Square: 0.8738

3.6. LASSO

We build LASSO Regression to perform variable selection. Our dataset has a large number of features, so it is often the case that many of the features have little or no effect on Y, so we select the most relevant features for predicting Y to produce a more predictable model with little or no sacrifice in predictive performance. For selecting the regularization parameter lambda, we use ***GridSearchCV*** and set the value to be the one that yielded the lowest *Mean-Square Error* value.

From Figure 4, we can see that LASSO coefficients are shrinking towards zero. And Figure 5 tells us where the optimal choice of lambda is. If the parameter is too small, which is the points to the very left, it may overfit due to too much variance and if lambda is too big, which is the points to the very right may cause under-fitting since the models are too simple and coefficient estimates are very biased.

We used *One Standard Error Rule* to compare models with different numbers of parameters in order to select the most parsimonious model with low error. Our selected parameter lambda is 0.00037. Using this parameter, we can get the first 28 selected variables. And using those variables to predict the price and get a result of R-Square is 0.89, and MSE is 20.412

4. Feature Importance

To determine the importance of features in the initial datasets, it is useful to start with a correlation matrix to measure the correlation between each variable. However, the Pearson correlation is only valid under the assumption that both variables follow a normal distribution. To ensure the validity of our results, I first tested the normality of the distribution of the price using a QQ plot and various statistical tests before using the correlation matrix (Figure 6). The tests indicated that the assumption of normal distribution may not be reliable, and the shape of the distribution appears to be right-skewed.

As the plot and statistical test show, it is not appropriate to rely solely on the Pearson correlation to determine the feature importance for this dataset. Instead, we can leverage machine learning models to identify the most significant features. Some of these models do not depend strictly on the distribution of the target data. For example, the feature importance of tree models is calculated based on the decrease in node impurity, weighted by the probability of reaching that node. The higher the value, the more important the feature is considered. Using this approach, we applied the LGBM model to visualize the initial dataset's feature importance. The result showed that factors such as trip miles, trip time, and location will have a significant impact on the price (Figure 7).

To delve into our dataset, we performed additional data processing We then compared the feature importance results of various well-tuned models after the data processing, which gave us further insights (see Figure 8). The red bar line in the figure represents the feature with the highest importance among others. The random forest model revealed that trip miles have the greatest impact on determining the price, while the Adaboost model yielded similar results and also showed some interest in temperature, wind speed, and humidity. Conversely, the linear regression models exhibited more interest in different locations. In module 3, we will delve into the business implications of these findings.

5. Clustering Models

5.1. Model Objective and Data Variable Choosing

Based on the study of feature importance, we derived some variables that have an impact on trip classification. They are "*base_passenger_fare*", "*trip_miles*", "*trip_time*", "*DOLocationID*", "*sealevelpressure*", "*temp*", "*windspeed*", "*humidity*", "*winddir*", and "*request_datetime*".

We attempted to build clustering models designed to classify each trip according to its fare as a measure of the value for each company. Also, we sought to explore the potential relationship between weather and trip prices. To standardize the trip price information, we created two new unit cost variables, "*fare/mile*" and "*fare/time*", which are calculated by dividing "*base_passenger_fare*" by "*trip_miles*" and "*trip_time*", respectively.

Consistent with the other models in this project, we chose the Uber61 dataset to build the model. We tried K-Means, DBSCAN, Gaussian Mixture, Bisecting K-Means, and AgglomerativeClustering, and finally only kept K- Means because the other models did not fit our dataset size or the algorithm didn't fit our expectations.

K-Means aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It has the advantages of simple principles, high efficiency, and high interpretability. However, there will be limitations on the amount of data as well as the type of data. Also, the value of k is difficult to determine, and different choices of k values can greatly affect the performance of the model. Even so, it is still the best clustering model for our project and dataset.

5.2. K-Means with Dataset Uber61

First, we built the clustering model without scaling. Based on the suggestion of Elbow Method results as in Figure 9, we chose the K-Means model with K=4 as the optimal number of clusters. There are 52783 trips in cluster 0, 44,813 in cluster 1, 26239 in cluster 2, and 27,761 in cluster 3. As shown in Figure 9, the number of the four clusters is relatively evenly distributed.

According to the violin plots of the seven variables in Figure 10, it is clear that the four clusters differ significantly in the weather variables: Wind Speed, Wind Direction, Humidity Value, Celsius Temperature, and Sea Level Pressure. The differences in wind direction are particularly pronounced. Whereas the four clusters show very similar distributions for the unit price-related variables: "*fare/mile*" and "*fare/time*". The unscaled clustering model demonstrates that for the Uber61 dataset, the price factor is a weak indicator for classifying trips, while the weather conditions can be used as a potential indicator for classification.

After scaling the model, the model shows different results from the unscaled model. As shown in Figure 11, the model classifies trips in Uber61 into four extremely uneven clusters based on the same selection of K=4. There are 13,132 in cluster 0, 275 in cluster 1, 112,530 in cluster 2, and 25,659 in cluster 3. The differences in the distribution of weather-related variables across the four clusters are much attenuated compared to the unscaled model. As shown in the violin plot in Figure 12, the trips in each cluster behave more dispersed across the different weather variables. The price-related variables, however, show differences in the distribution in this model, and although the distribution trends remain similar across the four clusters, significant differences can be seen in the numerical values. As a result, we can see that in the scaling model, unit price can be used as an indicator to classify trips while the effect of weather variables is not significant.

5.3. Result and Problem of Uber61

Based on the results of the two models above, we were unable to test our hypothetical goals. There is no highly significant difference in the performance of prices across clusters. The evidence that we can take price as an indicator to distinguish the value of different trips is lacking. Also, the Uber61 dataset shows a tendency for price and weather to be independent of each other. Weather is also difficult to use as a basis for classifying trips. In this regard, we speculate that this result may be due to the limitations of the dataset itself. We chose zone 61 with the highest number of Uber orders for code efficiency. However, a single zone rarely has different weather conditions. In addition, there may be peculiarities in the area that make riders' ride preferences unaffected by weather, etc.

5.4. K-Means with Dataset Lyft

We eventually decided to use the Lyft dataset for our clustering model while keeping the same variables as in our previous clustering model to better compare the result. However, what differed from the model using dataset Uber61 is the cluster number. Using the elbow method shown in Figure 13, we can see the optimal cluster number K = 3.

Similar to the model using dataset Uber61, we also did two models with one unscaled and one scaled. The unscaled K-Means model clustered the dataset into a relatively equal number of 1525809 in cluster 1, 896386 in cluster 2, and 648419 in cluster 3 as shown in Figure 13. It also demonstrated a similar result as the previous model. The price-related variables: *Base Passenger Fare/Trip Mile* and *Base Passenger Fare/Trip Time* showed a similar distribution. The weather-related variables: Wind Speed, Humidity, Wind Direction, Temperature, and Sea Level Pressure showed a larger distinction. However, the distinction of the price related variables is numerically larger compared to the previous model. It indicated a tendency that as the dataset size increases which cover more weather information, the relation between weather and price is more significant (Figure 14).

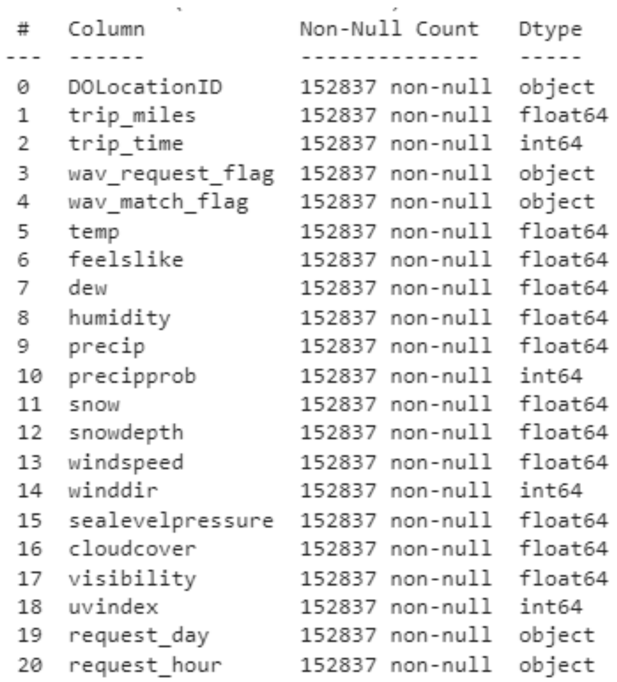
The scaled K-Means model showed a different cluster size distribution. With 17764 in cluster 1, 1835643 in cluster 2, and 1217206 in cluster 3, cluster 1 is small compared to the other two clusters (Figure 15). In this model, we can see a clearer distinction of the price-related variables while keeping the distinction of the weather-related variables. For better analysis, we only did comparison between cluster 2 and cluster 3. Cluster 2 has a higher *fare/mile* but a lower *fare/time* compared to cluster 3. The humidity level is much higher for cluster 3 while the wind speed and the wind direction is higher for cluster 2. Temperature and sea level pressure still remains insignificant in this model (Figure 16).

6. Conclusion

From Figure 17, we conclude that linear regression and LASSO have the lowest MSE and highest OSR. Ignoring the data limitation, in which we only use a portion of the original data, we believe that linear regression model is the best model for price prediction considering its simplicity and efficiency.

The conclusion for the clustering model using the Lyft dataset is that the relationship between the price-related variables and the weather-related variables are still weak and insignificant. However, as we increased the size of our dataset, the significant level started to rise. Larger weather data variation may help the model to better determine the distinction between clusters. In our scaled K-Means model, the distinction is much more obvious compared to our previous clustering model. We expect to find more results using the whole Uber dataset.

Appendix



```
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   DOLocationID      152837 non-null  object
 1   trip_miles        152837 non-null  float64
 2   trip_time         152837 non-null  int64
 3   wav_request_flag  152837 non-null  object
 4   wav_match_flag    152837 non-null  object
 5   temp              152837 non-null  float64
 6   feelslike         152837 non-null  float64
 7   dew               152837 non-null  float64
 8   humidity          152837 non-null  float64
 9   precip            152837 non-null  float64
 10  precipprob        152837 non-null  int64
 11  snow              152837 non-null  float64
 12  snowdepth         152837 non-null  float64
 13  windspeed         152837 non-null  float64
 14  winddir           152837 non-null  int64
 15  sealevelpressure  152837 non-null  float64
 16  cloudcover        152837 non-null  float64
 17  visibility        152837 non-null  float64
 18  uvindex           152837 non-null  int64
 19  request_day       152837 non-null  object
 20  request_hour      152837 non-null  object
```

Figure 1: Independent Variables

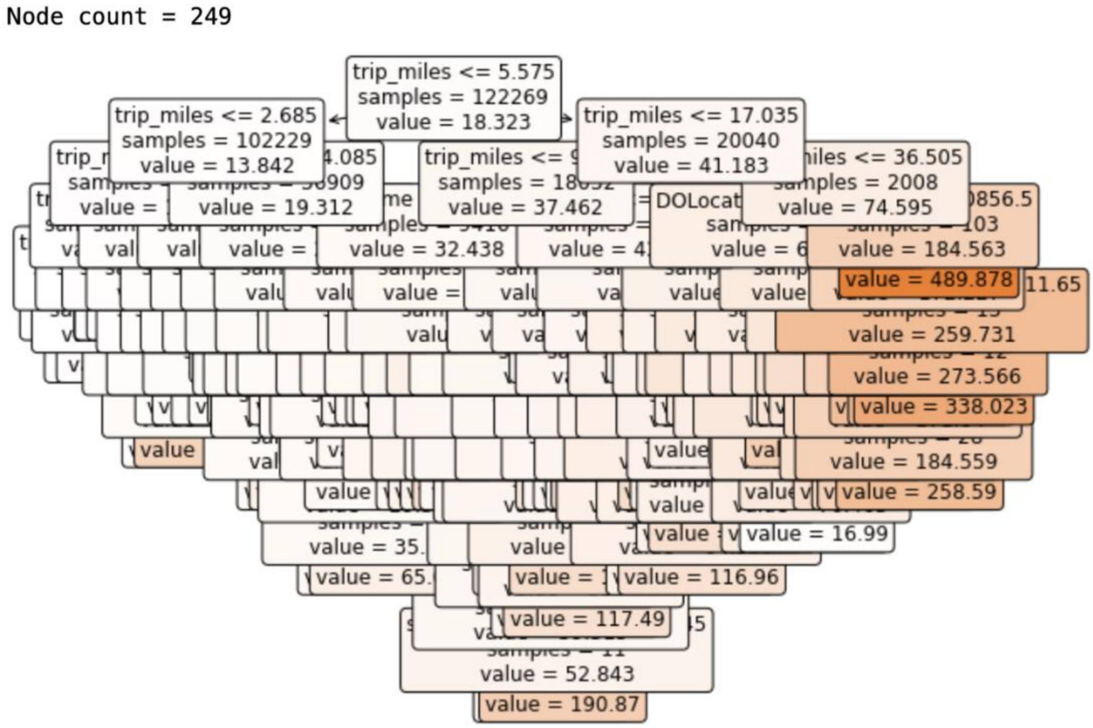
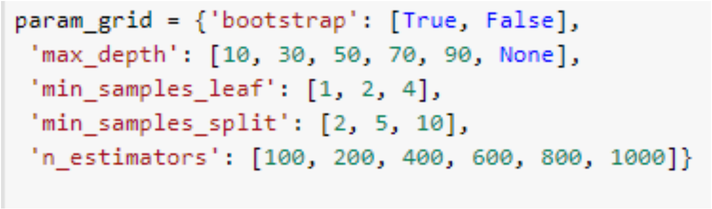


Figure 2: The Result of CART



```
param_grid = {'bootstrap': [True, False],
 'max_depth': [10, 30, 50, 70, 90, None],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [100, 200, 400, 600, 800, 1000]}
```

Figure 3: Hyperparameter Result

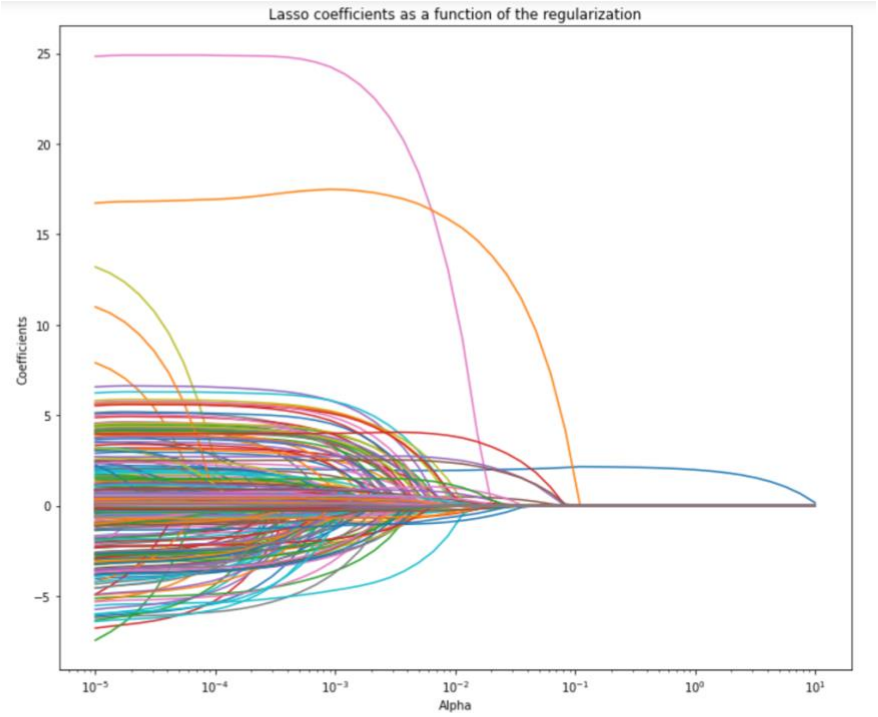


Figure 4: LASSO Coefficients Shrinkage

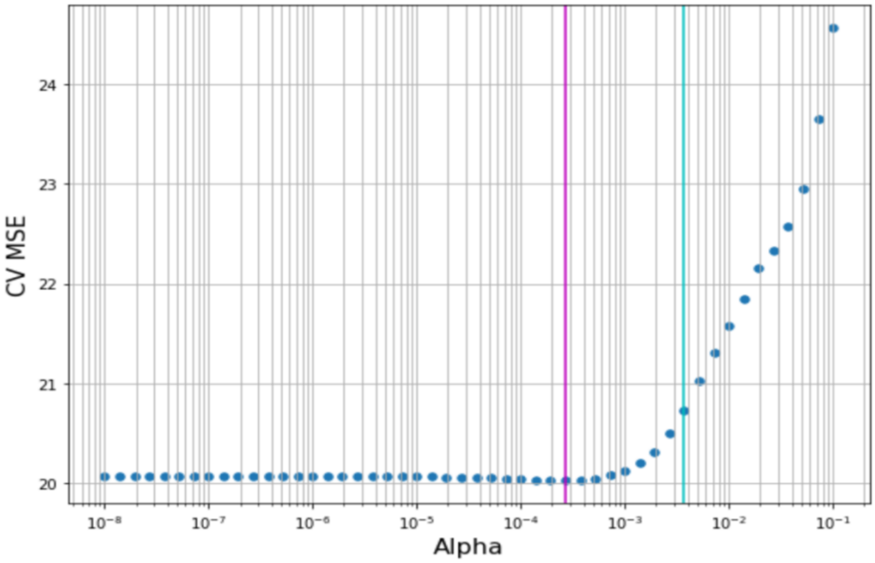


Figure 5: Selecting Lambda via Cross-Validation

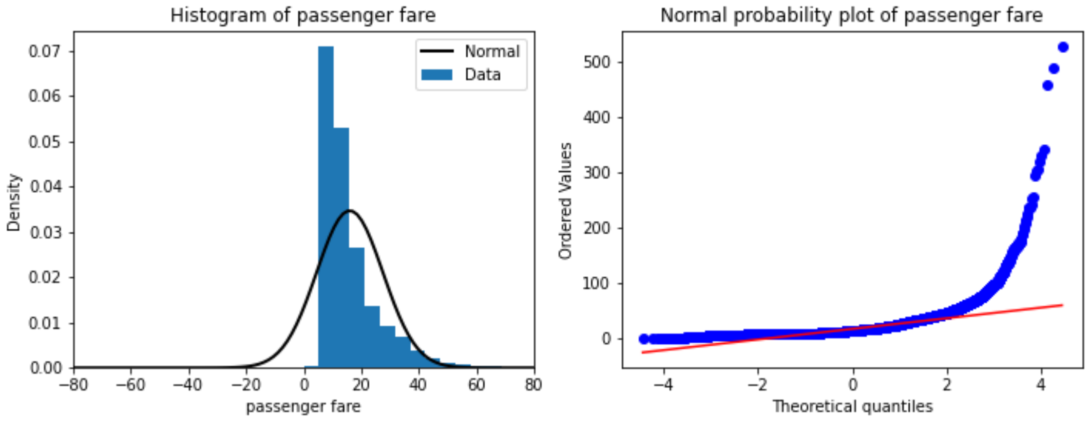


Figure 6: The Distribution of Passenger Fare

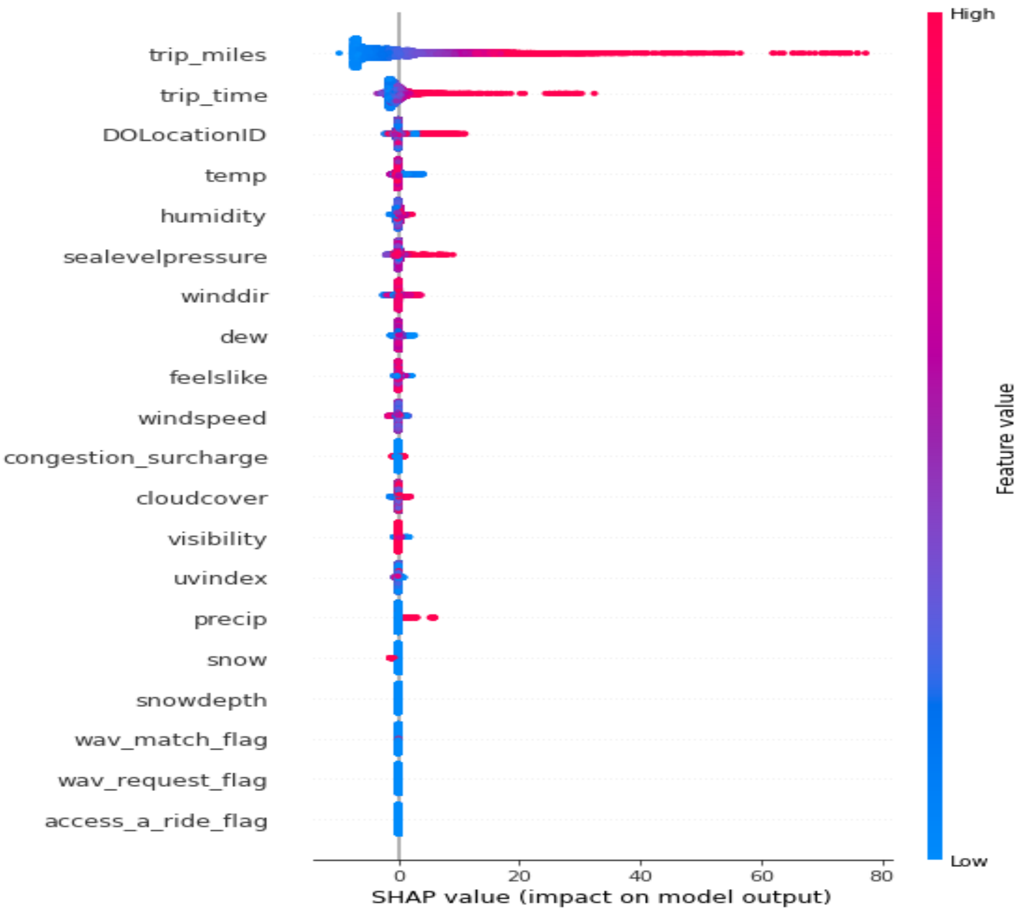


Figure 7: Visualization of LGBM Feature Importance

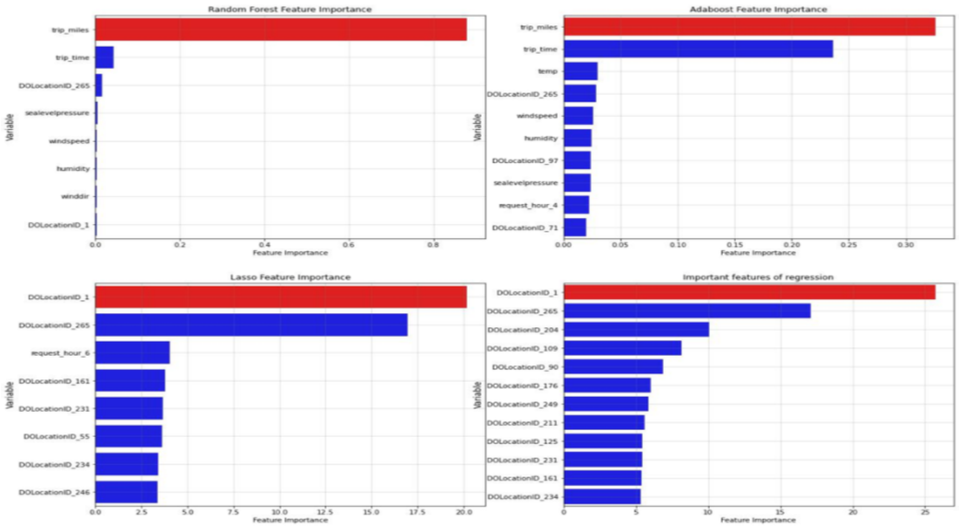


Figure 8: Visualization of Different Models' Feature Importance

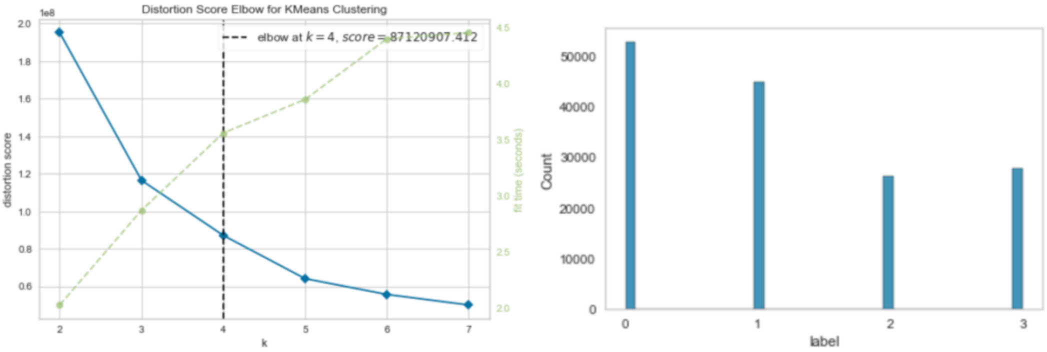


Figure 9: Unscaled Clustering Model for Uber61 Dataset

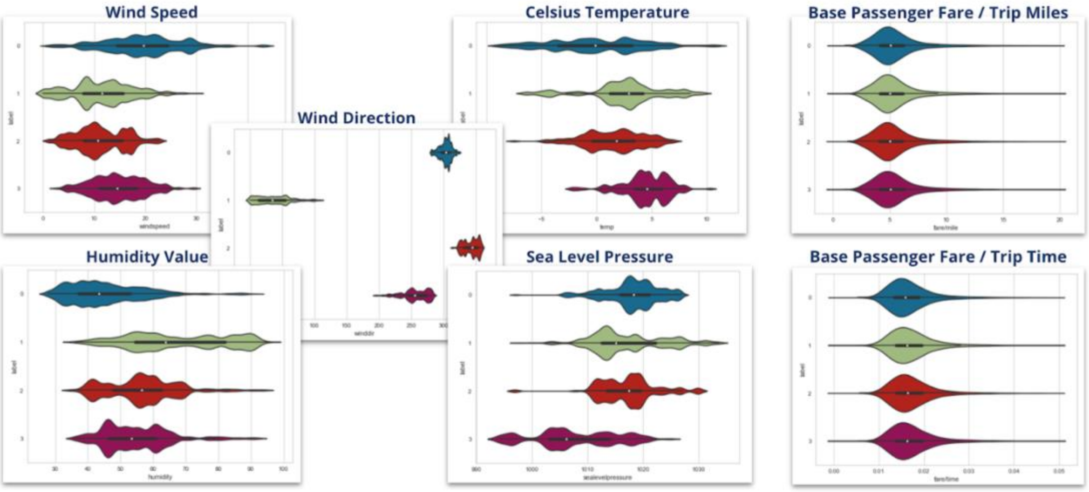


Figure 10: Violin Plot of Variables for the Unscaled Clustering Model of Uber61 Dataset

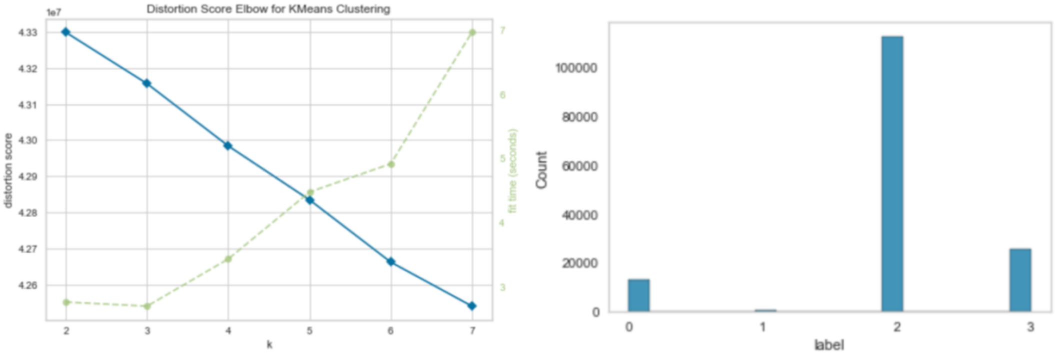


Figure 11: Scaled Clustering Model for Uber61 Dataset

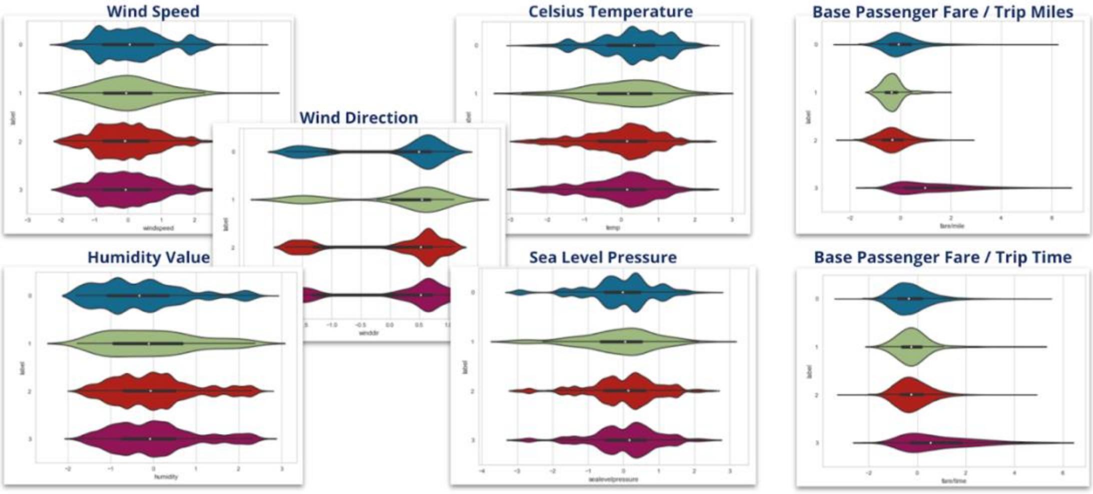


Figure 12: Violin Plot of Variables for the Scaled Clustering Model of Uber61 Dataset

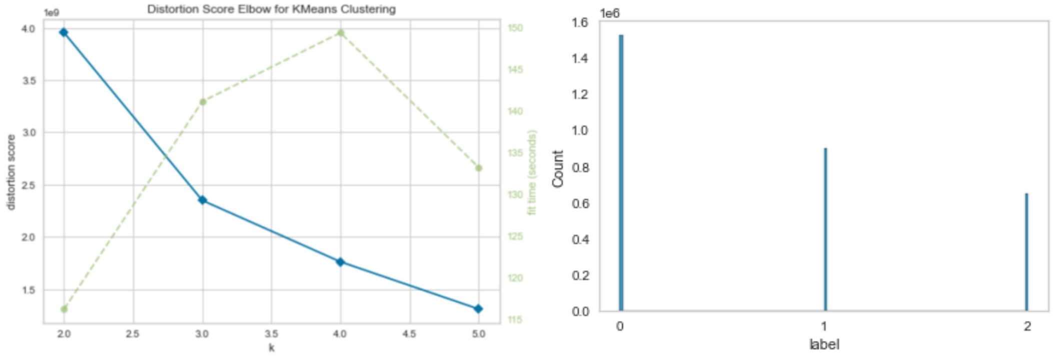


Figure 13: Unscaled Clustering Model for Lyft Dataset

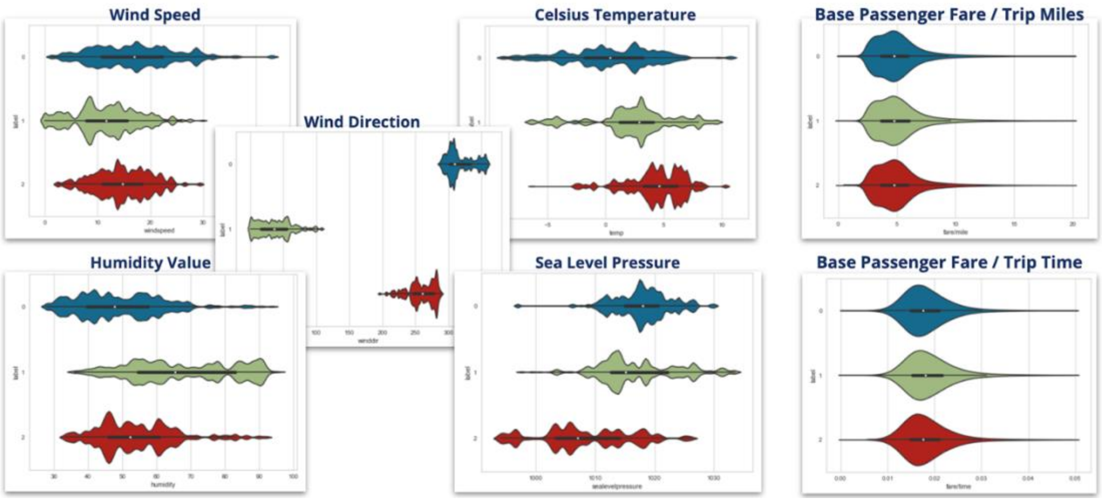


Figure 14: Violin Plot of Variables for the Unscaled Clustering Model of Lyft Dataset

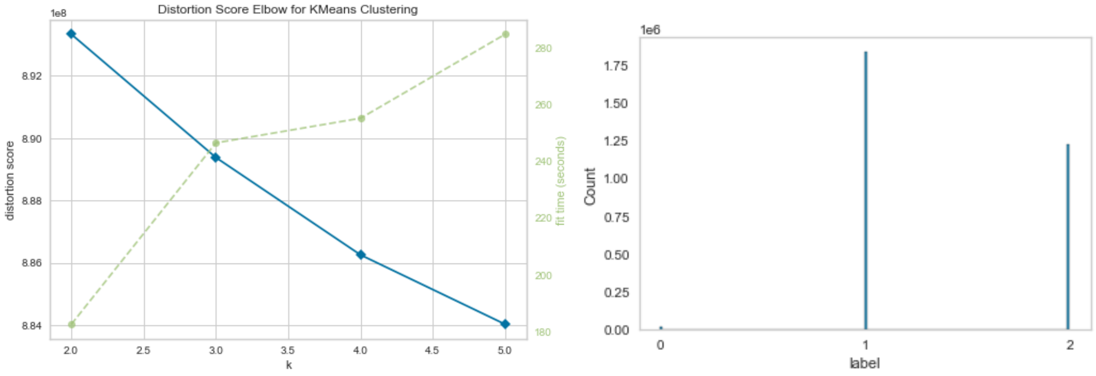


Figure 15: Scaled Clustering Model for Lyft Dataset

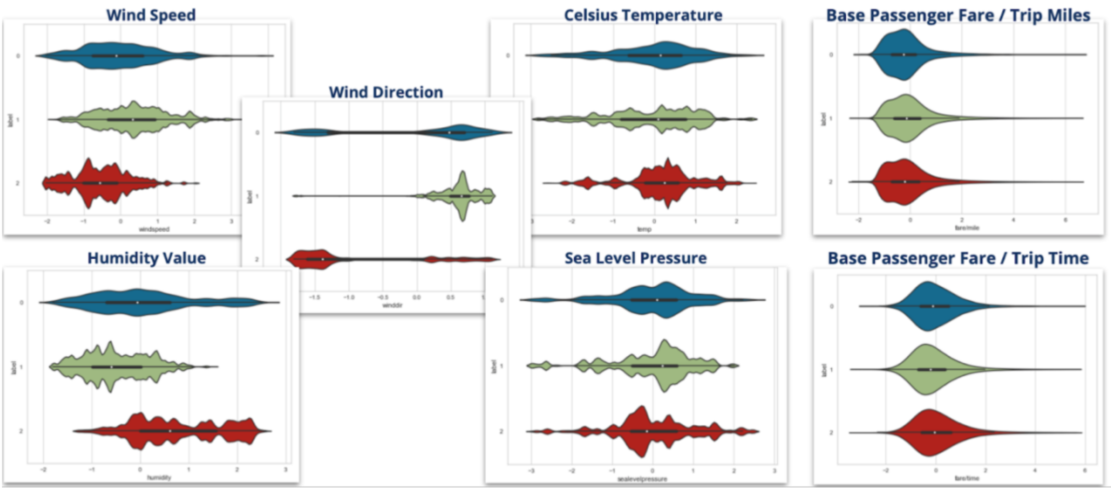


Figure 16: Violin Plot of Variables for the Scaled Clustering Model of Lyft Dataset

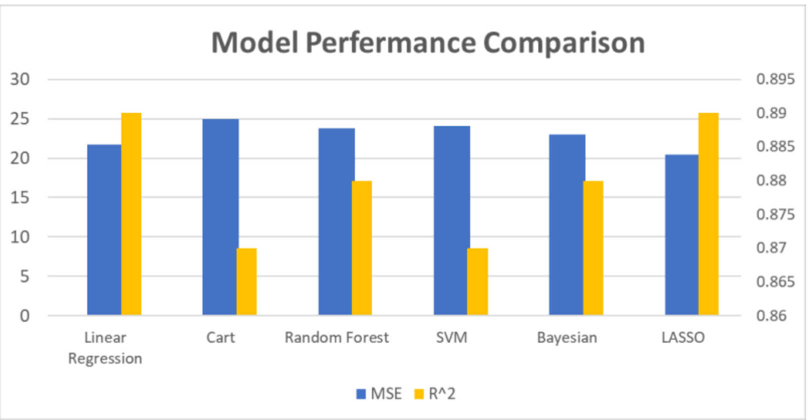


Figure 17: Pricing Model Performance Comparison