Uber/Lyft Ride Fare Analysis and Prediction

Wanying Zhang
Data Science Initiative, Brown University
GitHub

December 10, 2023

1 Introduction

Uber & Lyft are both ride-sharing companies that bridge the gap between private transportation services and people in need of a ride. Unlike traditional public transportation, such as buses or trains, the prices of Uber and Lyft are not constant; instead, they are greatly affected by various factors. With the popularity of Uber and Lyft services expanding across the United States, understanding the secrets behind the fluctuation of their prices becomes crucial. Recognizing how prices change under different circumstances provides valuable insights and helps individuals plan their travel ahead of time. For instance, when traveling to different places, this understanding aids in making informed decisions, such as choosing between Uber and Lyft, renting a car, or opting for public transportation.

The goal of this project is to develop regression models to predict ride prices for Uber and Lyft based on features such as distance traveled, weather, time of day, etc. Since neither Uber nor Lyft publicly releases their data, the dataset was obtained from Kaggle [1]. The dataset's author collected real-time data using Uber & Lyft API queries and corresponding weather conditions in a few hotspots [2] in Boston for over a week from Nov. 18, 2018.

The author of the dataset conducted a study and constructed a model for predicting the prices of Uber and Lyft [3], achieving an accuracy score of 92.79%. It is noteworthy that the dataset's author replaced the substantial missing values in the 'rain' feature with 0 but did not explicitly provide a rationale for this choice. In contrast, I retained the missing values in my work and addressed them using more advanced techniques, including iterative imputer and the reduced feature model.There is also other research that separately analyzed the price factors for Uber and Lyft [4], achieving a test R^2 score of 95% for Uber and 90% for Lyft using the Cubist model.

The datasets comprise two different tables. One contains information about each individual trip, such as trip distance, trip starting location, trip ending location, etc. The second table contains information about the weather at the places and

times when the trip information was retrieved, including features like location, time, temperature, pressure, and rain. These two tables were joined based on the trip source location and time, resulting in our final table of 63, 7976 rows and 15 features, where each row represents one trip.

2 Exploratory Data Analysis (EDA)

Table 1 provides a concise overview of feature names, types, and the respective percentages of missing values across all 15 features in the dataset. All missing values are observed in features associated with weather. Among these, the "rain" feature stands out with the highest percentage, reaching 83%. This implies that a substantial portion of the data lacks information on rainfall. Furthermore, it is noteworthy that a substantial 83% of all data points exhibit missing values, with most of them specifically missing the value in the feature "rain."

Feature Name	**Feature Type**	**Percent of Missing Value**
Price (Target)	Continuous	0
Distance	Continuous	0
Cab Type	Categorical	0
Source	Categorical	0
Destination	Categorical	0
Surge Multiplier	Ordinal	0
Ride Type	Ordinal	0
Hour of Day	Ordinal	0
Day of Week	Ordinal	0
Temperature	Continuous	4%
Clouds	Continuous	4%
Pressure	Continuous	4%
Wind	Continuous	4%
Humidity	Continuous	4%
Rain	Continuous	83%

Table 1: Features in the Data Set

The study delved into the correlation between the price of a trip and the distance traveled. Based on our common intuition, we would guess that a distinctly evident linear regression would characterize the relationship between distance traveled and price. However, Figure 1 surprisingly revealed that the anticipated clear linear connection between these two variables is not as pronounced, particularly in the case of Lyft. Interestingly, the price of Uber rides appears to be more noticeably influenced by the distance traveled compared to Lyft.

To investigate the connection between Price and Ride Type, we employed the box plot in Figure 2. Uber and Lyft ride types were categorized ordinally and displayed on the x-axis, ranging from regular to premium. As the ride type progresses from regular to premium, there is a corresponding increase in price along the y-axis. Additionally, it is noteworthy that the overall trend indicates Lyft rides tend to

Figure 1: Price vs. Distance

be slightly more economical than Uber rides for the regular ride types, while Uber demonstrates a cost advantage for the premium ride types.

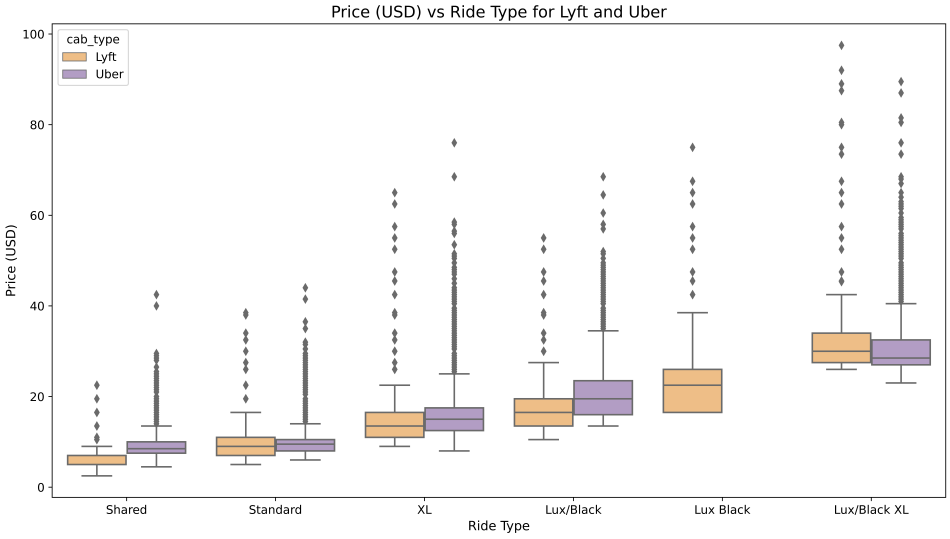

Figure 2: Price vs. Ride Type

Lastly, we examined the relationship between Price and Surge Multiplier. The Surge Multiplier is a feature that comes into play when the demand for rides exceeds the supply of cars, resulting in a price increase. The distribution of prices for all seven surge multipliers is is included in Figure 3, arranged from the lowest

to the highest surge multiplier. The top histogram corresponds to the lowest surge multiplier, which is 1. This plot exhibits a right-skewed distribution, with more trips stacked in the cheapest price range, and as we progress downwards, the distributions gradually approach to the right, indicated the overall price increase for the trips. This observation implies that as the surge multiplier increases in the dataset, there is a corresponding increase in the price of the ride.

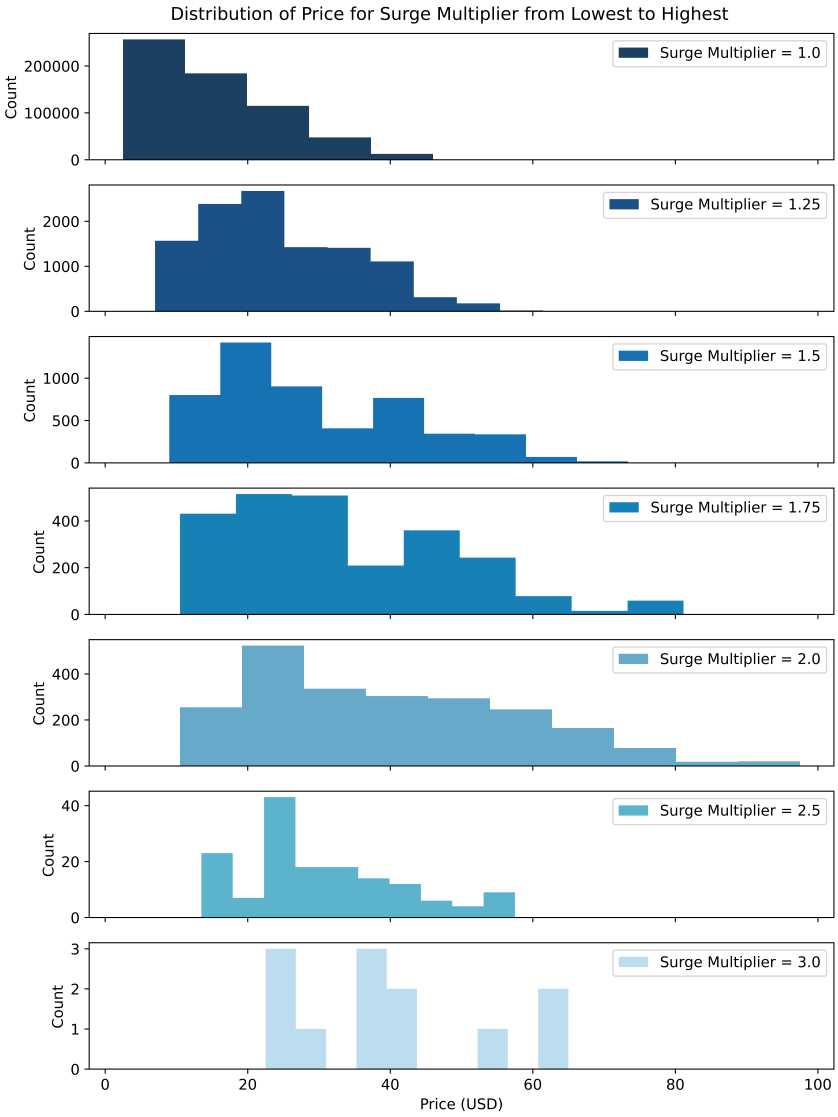

Figure 3: Distribution of Price for Surge Multipliers

3 Methods

Given our large dataset, utilizing the entire dataset for model training would be computationally expensive. Consequently, stratified sampling was employed to obtain a smaller, representative portion of the data for use in the subsequent steps.

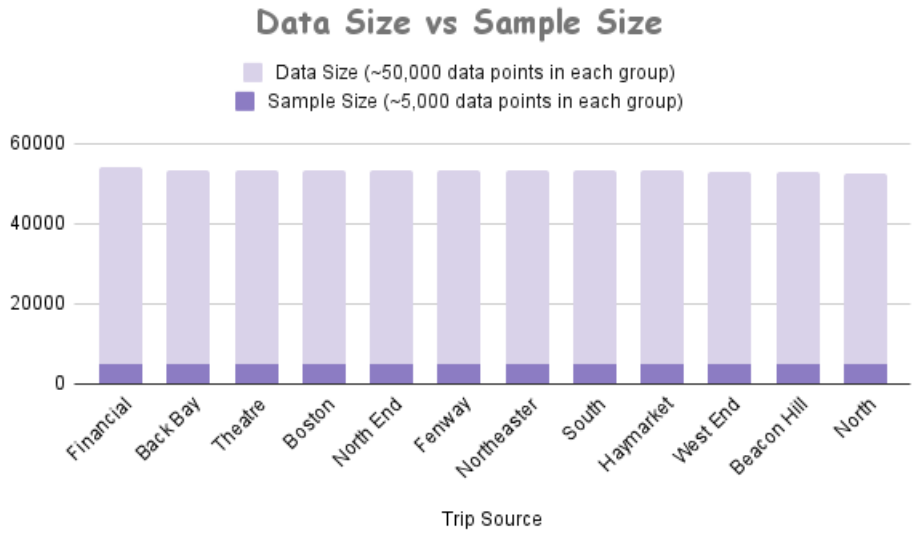

Figure 4: Original Data Size vs Sample Size to use to Develop Machine Learning Models

As shown in Figure 4, we initially group the data points based on the source of the trip. There are 12 trip sources, each carrying equal weight, and approximately $50,000$ data points in each group. Subsequently, we randomly select $5,000$ data points from each group, resulting in our final sample size of $60,000$ data points, which constitutes approximately $\frac{1}{10}$ of the entire dataset size.

Five models (Lasso Regression, Random Forest Regression, Support Vector Regression, K-Nearest Neighbors Regression, and XGBoost Regression) were implemented. Additionally, two different processes were developed: one for XGBoost and another for all other models. Moreover, each process was performed five times on five different random states to address uncertainty during splitting and undeterministic models.

For XGBoost, we employed a train-test split to divide the data into 60% training, 20% validation, and 20% testing sets. For all other models, we initially used a train-test split to allocate 80% to other and 20% to testing. Subsequently, we applied KFold with 4 splits to further divide the other set into 75% for training and 25% for cross-validation.

Figure 5 illustrates the preprocessing procedure for the models. In XGBoost, the reduced feature model was applied after initial preprocessing to handle missing values in the dataset. For all other models, iterative imputer with linear regres-

sion was used for continuous features to address missing values before applying StandardScaler.

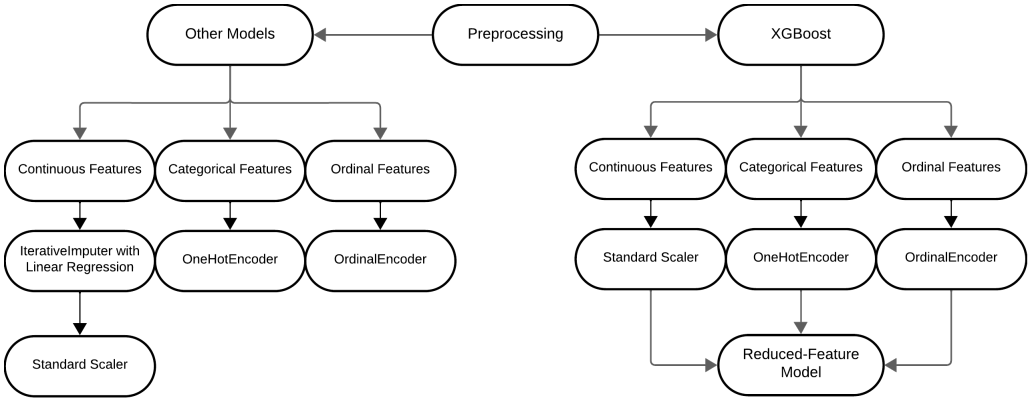

Figure 5: Preprocessing Process for Five Models

The pipeline for one random state is presented in Figure 6, and Root Mean Squared Error was used as the metric to evaluate model performance. XGBoost employed a different pipeline by training the model through all parameter combinations, identifying the parameters that yielded the best validation score, and applying them to the test set for calculating the final test score. Additionally, early stopping round = 50 was implemented in the XGBoost pipeline to enhance efficiency.

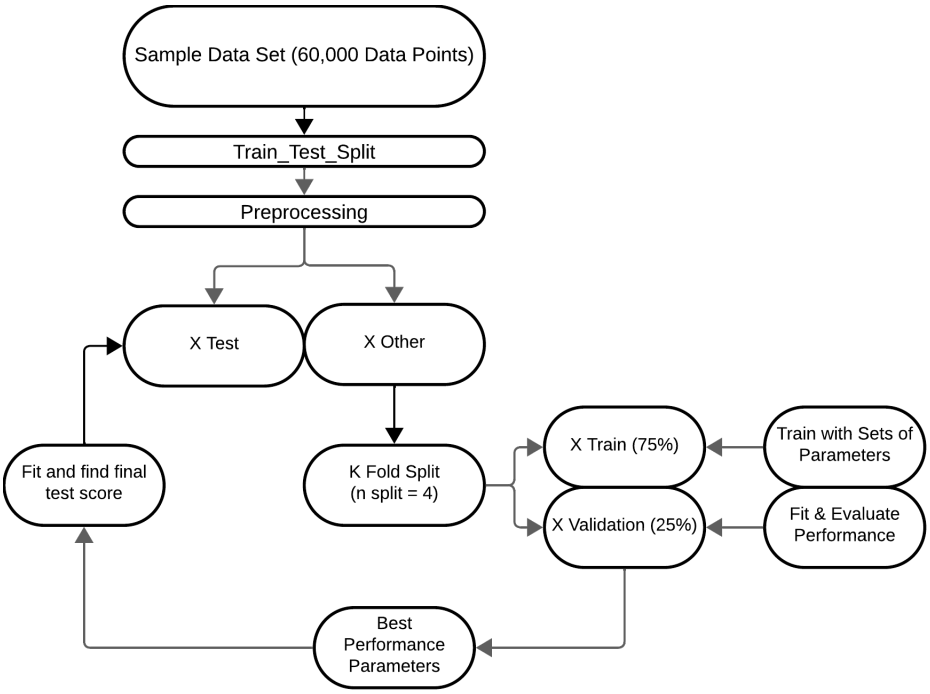

Figure 6: Cross Validation Pipeline for Models except XGBoost

Figure 7 presents a table summarizing the different models that were implemented, the tuned parameters, and the parameters that yielded the best score.

ML Algorithms	Parameters Tuned	Best Parameters
Lasso Regression	alpha: [0.001, 0.01, 0.1, 1, 10, 100]	alpha = 0.001
KNN	n_neighbors: [1, 3, 10, 30, 100] weights: ['uniform', 'distance']	n_neighbors = 10 weights = 'distance'
SVR	gamma = [0.001, 0.1, 10, 1000, 100000] C = [0.1, 1, 10]	gamma = 0.1 C = 10
Random Forest	n_estimators: [1, 3, 10, 30, 100, 300] max_depth: [1, 3, 10, 30, 100]	n_estimator = 300 max_depth = 10
XGBoost (Reduced Feature)	learning_rate: [0.01, 0.1, 0.2] max_depth: [3, 6, 10, 30, 100]	learning_rate = 0.2 max_depth = 30, 100

Figure 7: Summary Table of Models & Parameters

4 Results

4.1 Metrics Evaluation

Two different metrics were employed to evaluate the performance of the five models. Fig. 8 illustrates the R^2 score of the test set for each model, where Random Forest performed best among all five models, achieving a mean score of 96% and a standard deviation of 0.18%. This indicates that the model explains 96% of the variation in the y-variable (Price of the ride). On the other hand, Fig. 9 presents the Root Mean Square Error of the test set for each model in comparison to the baseline Root Mean Square Error. The baseline score is computed using the mean of the y variable. Random Forest attained a mean RMSE score of 1.77 with a standard deviation of 0.046, which is a lot better than the baseline RMSE score of 9.30

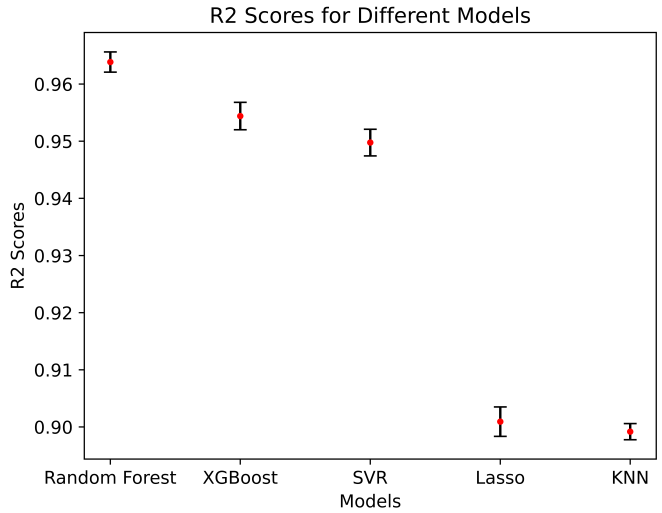

Figure 8: R^2 Score for 5 Models

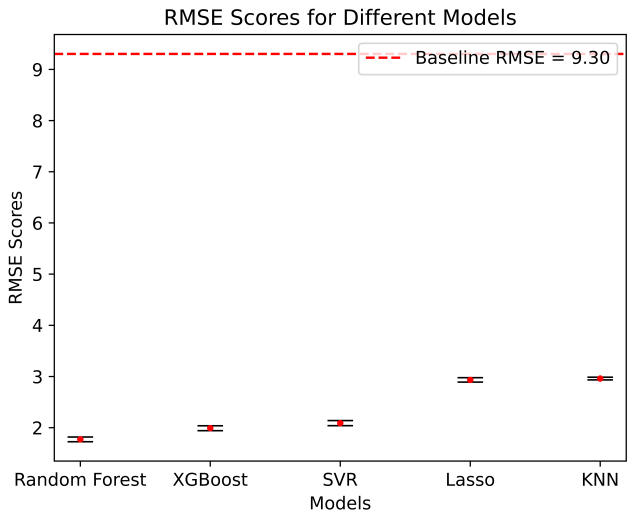

Figure 9: Root Mean Square Error for 5 Models & Baseline Score

4.2 Global Feature Importance

Global Feature Importance was studied to determine the most significantly contributed features to the model. Three different methods were implemented to study the importance of each feature based on our best model Random Forest Regression. The feature 'ord_cab_type_Lyft' were dropped when calculating the global importance for all three methods because of the strong correlation of -1 with the feature 'ord_cab_type_Uber'.

In Figure 10, permutation importance was utilized to determine the importance score of each feature. This involves randomly shuffling a single feature in the test set and recalculating the test score with the shuffled dataset. The greater the decline in score after shuffling, the more important the feature is considered. It indicated 'ord_ride_type', 'std_distance', 'ord_surge_multiplier' are top three important features.

Figure 11 computed feature importance based on the Random Forest's built-in feature importance parameter. This calculation relies on the mean accumulation of the decrease in impurity for each individual tree in the model. It indicated 'ord_ride_type', 'std_distance', 'ord_surge_multiplier' are top three important features.

Figure 12 calculated feature importance based on the average of the Shapley values derived from game theory. It indicated 'Ord_ride_type', 'std_distance', 'ord_cab_type_Uber' are top three important features.

All three figures indicate that ride type, distance, and surge multiplier are crucial features in the model, which aligns with our findings from the EDA stage. Additionally, features related to weather do not appear to be correlated with the variation in the price of the ride. In conclusion, the price of Uber and Lyft trips is strongly correlated with the chosen ride type, the distance of the trip, and the

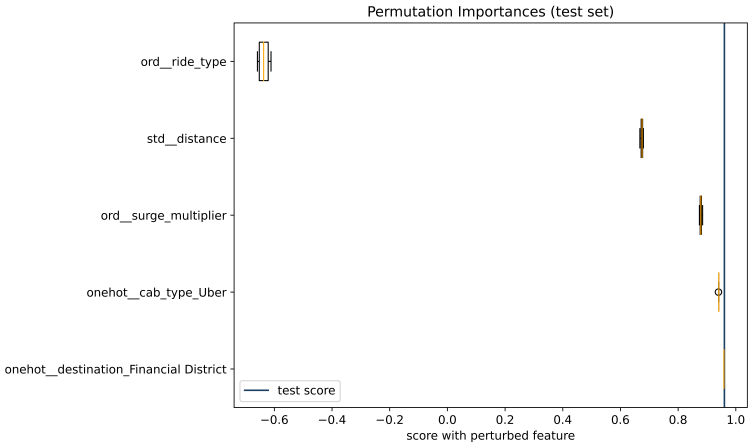

Figure 10: Global Feature Importance (Permutation)

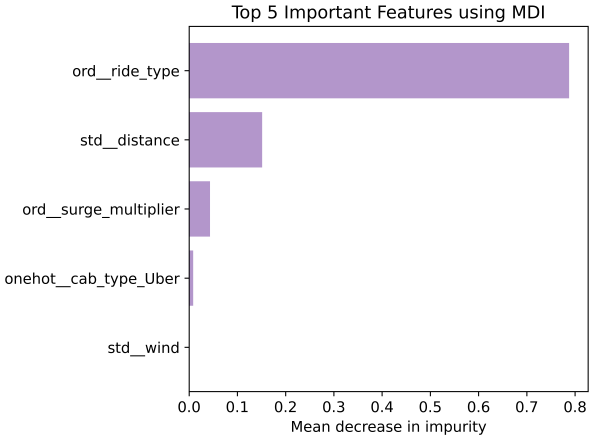

Figure 11: Global Feature Importance (MDI)

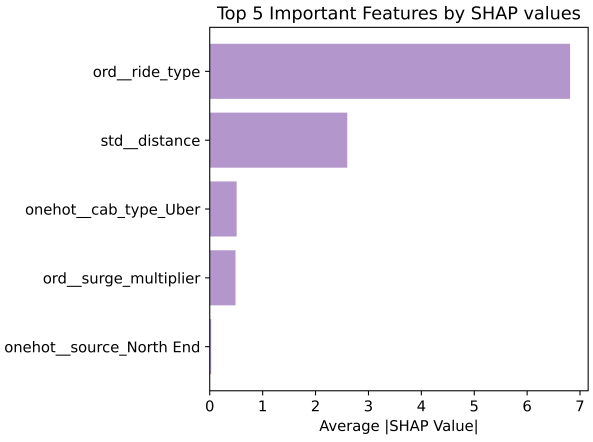

Figure 12: Global Feature Importance (Average SHAP)

current demand and supply of cars at the time the ride was requested.

4.3 Local Feature Importance

Local feature importance was also investigated to understand the contribution of a feature to the predictability of specific data points. Two randomly selected data points from the test set were examined to identify the features that predominantly influenced the predicted score. Larger blue arrows signify a substantial contribution to decreasing the y-value, while larger red arrows indicate a significant contribution to increasing the y-value.

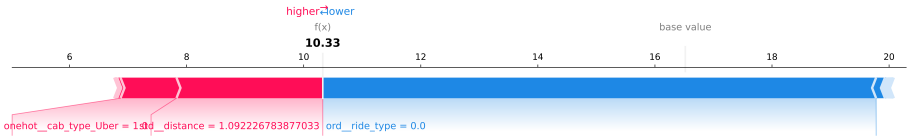

Figure 13: SHAP Local Feature Important Random Point 1

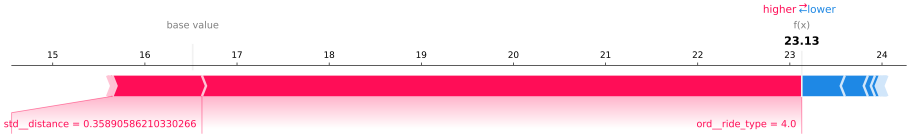

Figure 14: SHAP Local Feature Important Random Point 2

For the data point in Fig 13, the 'ord_ride_type = 0' corresponds to the most regular ride type, 'shared,' where the customer chooses to share a ride with others, playing a significant role in lowering the price. In comparison to the data point in Fig 14, although the second data point indicates a shorter travel distance, but the price is higher than the first one. This is because 'ord_ride_type = 4' signifies that the customer chose one of the premium ride types, contributing significantly to the increase in the price of the ride.

5 Outlook

If more time were given to enhance the model performance and interpretability of the project, several steps could be taken. First, we could implement additional models, such as XGBoost without the Reduced Feature Approach. Additionally, applying the Reduced Feature Approach to other models, such as Support Vector Regression and K-Nearest Neighbors Regression, would be beneficial. Secondly, expanding the set of parameters to tune could be explored to assess any potential

improvement in the test score. This involves a thorough examination of parameter configurations to optimize the model. Lastly, testing the model on the entire dataset could be considered to evaluate whether there is an increase in overall model performance. This step would provide insights into the generalization and robustness of the model across the entire dataset.

References

[1] Kaggle Data Set Uber & Lyft Cab Prices

[2] Locations in Boston where Trip Information was Collected

[3] scala-spark-cab-rides-predictions by Ravi Munde and Karan Barai

[4] Dent, Diana. "Machine Learning Uber vs. Lyft Price Prediction Modeling." Data Science Blog, NYC DATA SCIENCE ACADEMY, 11 Apr. 2023, nycdatascience.com/blog/meetup/uber-vs-lyft-price-prediction-machine-learning-model/

[5] GitHub Repository:
`https://github.com/WWWW0203/Uber-and-Lyft-Price-Prediction`