Uber/ lyft Price Prediction

Brown University Data Science Institute
Winnie Zhang
12-06-2023

https://github.com/WWWW0203/data1030Project

Recap

Uber & Lyft - Ride Sharing company that bridge the gap between private transportation service and people needing a ride.

Predict Uber/Lyft ride price based on features such as distance travelled, weather, time of day, surge multiplier, etc. (Regression)

Why is it important?

- Understanding how the prices of Uber & Lyft changes under different circumstances will give us better insights & help determine our travel plans
    - For example, it can help us decide whether to take uber/lyft, rent a car, or take other public transportations when we travel to different places.

Kaggle link of the data: https://www.kaggle.com/datasets/ravi72munde/uber-lyft

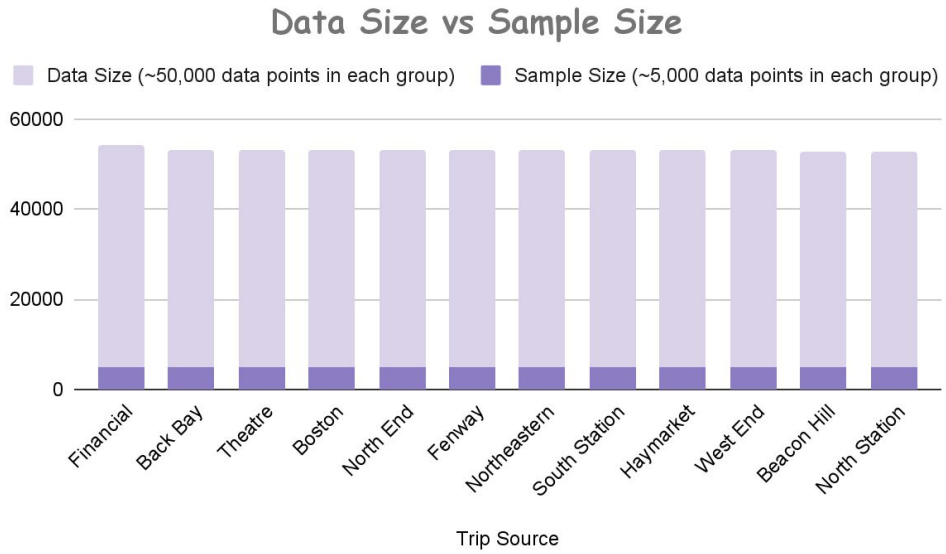
Data Dimensions & Missing Data

Before Preprocessing: 637, 976 rows x 14 features
After Preprocessing: 637, 976 rows x 37 features

Fraction of missing values in features (continuous):

Temperature: 4%
Clouds: 4%
Pressure: 4%
Humidity: 4%
Wind: 4%
Rain: 83%

Stratified Sampling



- Large dataset (~600,000 rows) is computationally expensive for model training.
- Used stratified sampling method to group the data points by trip source (equally weighted), then randomly choose 5,000 data points from each group.

Splitting

XGBoost
- Train Test Split (60% Train, 20% val, 20% Test)

Other Models
- Train Test Split (80% Other, 20% Test)
- KFold on Other Set with n_splits = 4

Preprocessing

XGBoost
- StandardScaler() on Continuous Features
- OneHotEncoder() on Categorical Features
- OrdinalEncoder() on Ordinal Features

Other Models
- Iterative Imputer with Linear Regression on Continuous Features with missing value
- StandardScaler() on Continuous Features
- OneHotEncoder() on Categorical Features
- OrdinalEncoder() on Ordinal Features

Training & Cross-Validation

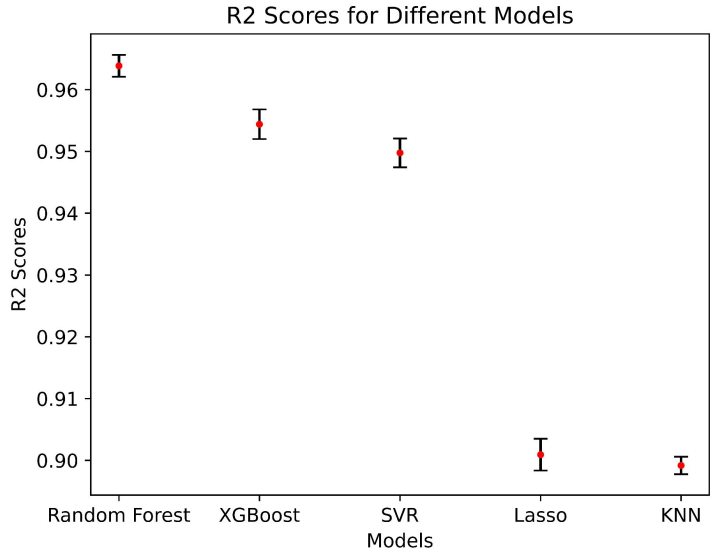
XGBoost
- Reduced feature approach
- Train model with all the combination of parameters
- Find the parameter set that gives the best validation score and apply to test set

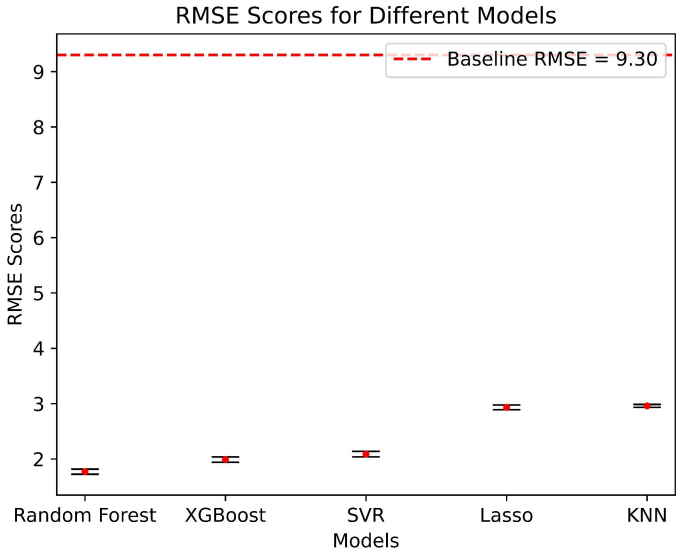
Other Models
- GridSearchCV with kFold and param_grid
- Find the parameter set that give the best validation score and apply to test set

ML Algorithms

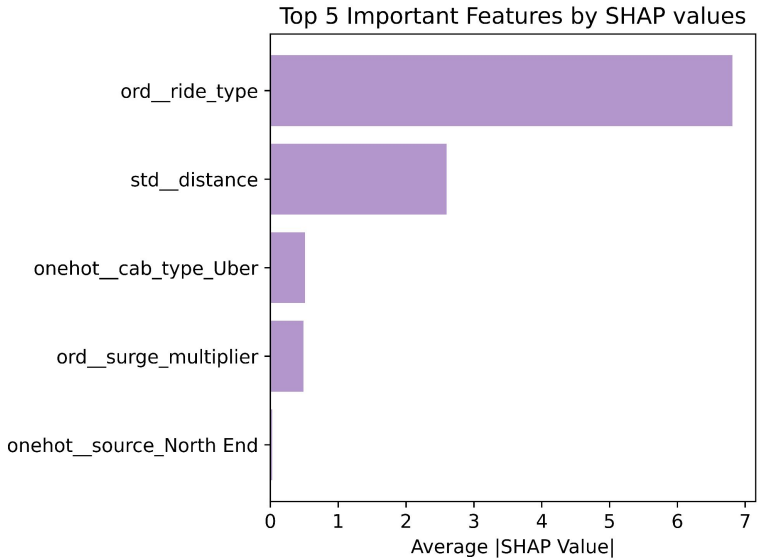
ML Algorithms	Parameters Tuned	Best Parameters
Lasso Regression	alpha: [0.001, 0.01, 0.1, 1, 10, 100]	alpha = 0.001
KNN	n_neighbors: [1, 3, 10, 30, 100] weights: [‘uniform’, ‘distance’]	n_neighbors = 10 weights = ‘distance’
SVR	gamma = [0.001, 0.1, 10, 1000, 100000] C = [0.1, 1, 10]	gamma = 0.1 C = 10
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

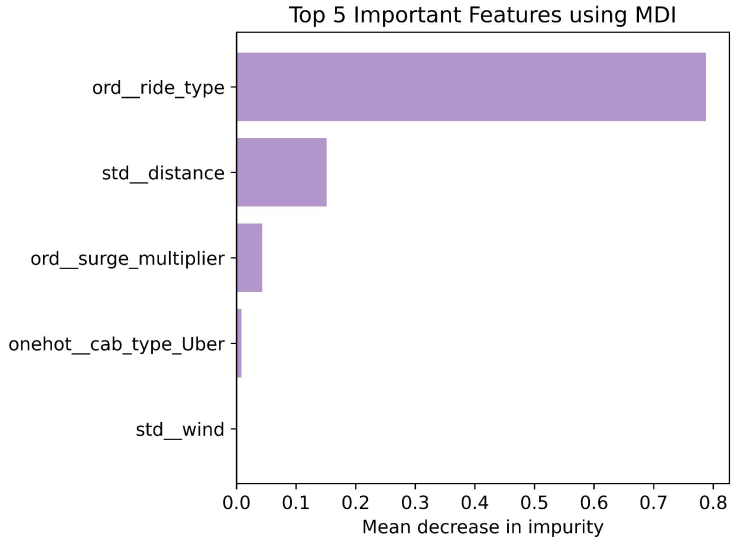
Results


R2 Scores for Different Models
R2 Scores
0.96
0.95
0.94
0.93
0.92
0.91
0.90
Random Forest
XGBoost
SVR
Lasso
KNN
Models

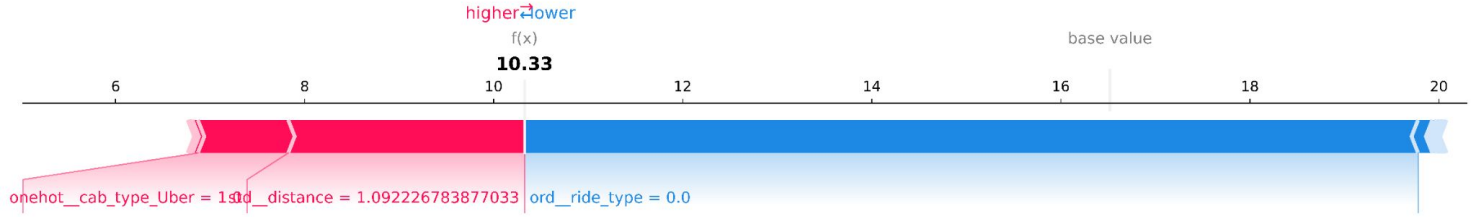

RMSE Scores for Different Models
Baseline RMSE = 9.30
RMSE Scores
9
8
7
6
5
4
3
2
Random Forest
XGBoost
SVR
Lasso
KNN
Models

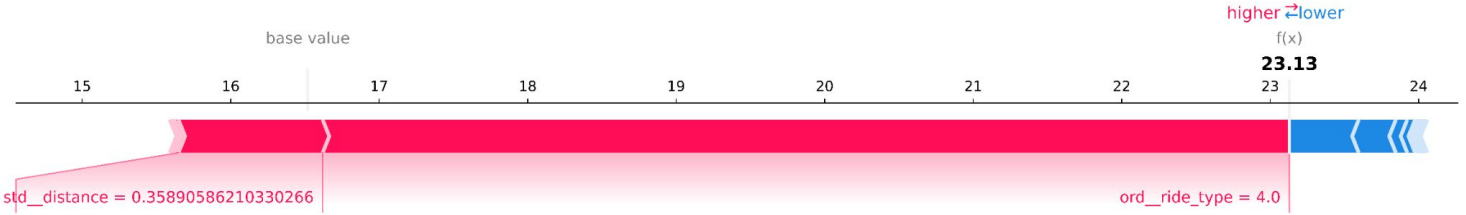
Results - Global





Results - Local





Outlook

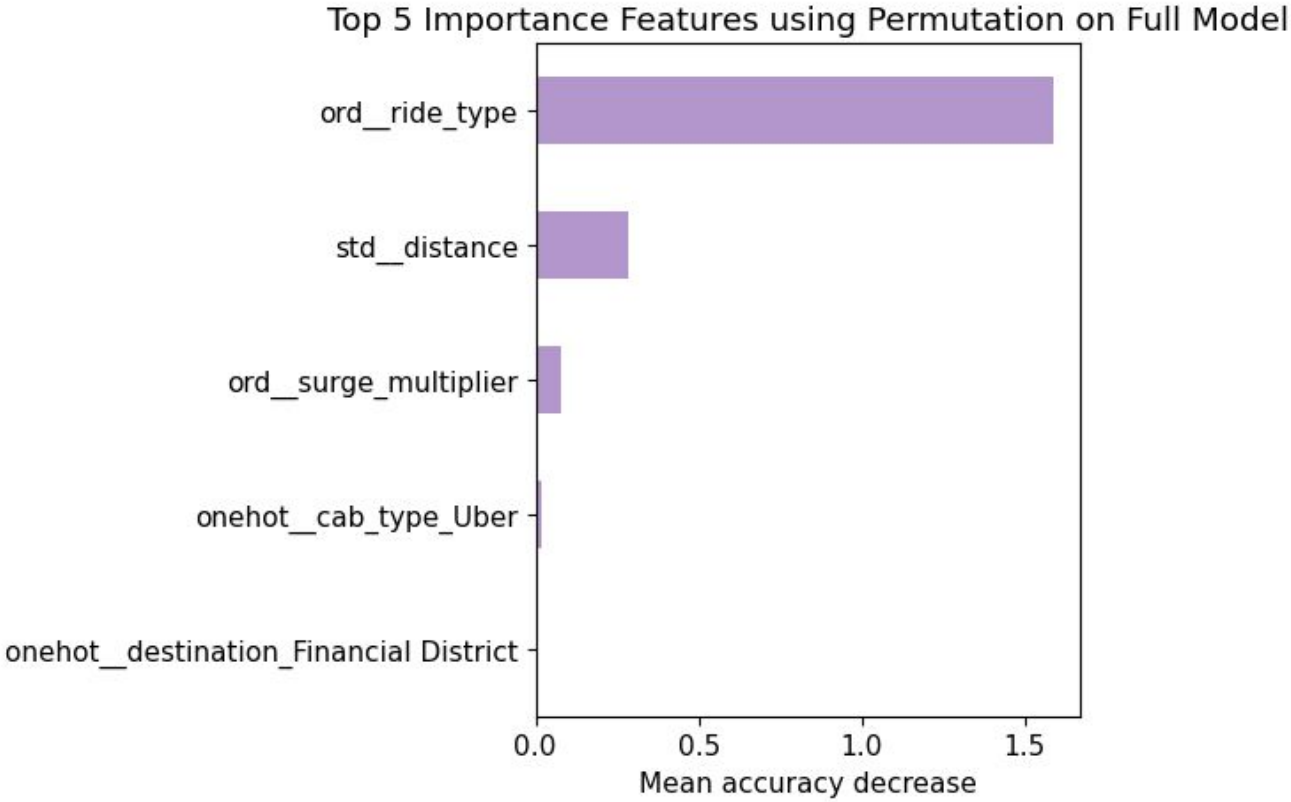
- Try XGBoost & reduced feature technique on other models
- Tune more parameters for each model
- Could spend time testing on the whole dataset to see how my model performs

THANKS

Questions?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

Other Figures



Other Figures