Predictive Analysis on High-dimensional Rideshare Price Data

Authors: Yunzhi Kong; Ella Wang; Shengdan Jin; Yiyang Sheng

Data Science And Analytics Program Of Georgetown University

INTRODUCTION

While taxi has been on the streets since 1800s, it seems to be steadily losing passengers due to the rise of other ride-sharing services. Launched in 2010, Uber quickly took over the rideshare market, followed by its competitor, Lyft in 2012. Unlike the pricing of traditional transportation, the rates of Uber and Lyft are generated for the dynamic pricing models. The models match fares to lots of variables such as weather, current demand, time.

In this project, a dataset includes pricing information of Uber and Lyft in Boston will be analyzed. The method of supervised learning will be used for rideshare pricing model fitting. And the most influential features for the rideshare fares will be selected from the model and be used for price prediction.

EXPLORING DATA

The dataset "rideshare_kaggle.csv" is credited to BM from Kaggle. It is about the price of a single ride with the respect of different factors, for instance, hour, distance, temperature, destination and cab type. The cleaned dataset contains "price" as the target variable, 6 categorical predictors, 22 predictors and their two-way interaction term. There are 258 features in total.

Major Predictors

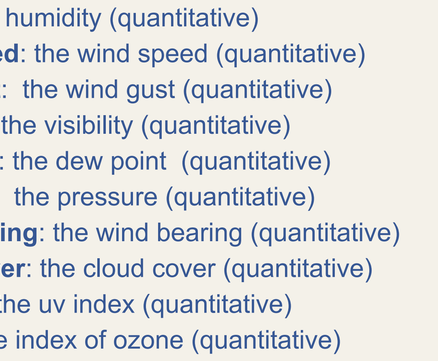
- **hour**: hour of the day (quantitative)
- **day**: day of the week (quantitative)
- **month**: month of the year (quantitative)
- **distance**: total distance of the ride (quantitative)
- **surge_multiplier**: a demand multiplier(quantitative)
- **temperature**: outdoor/apparent temperature (quantitative)
- **precipIntensity**: the intensity of precip (quantitative)
- **precipProbability**: the probability of precip (quantitative)
- **humidity**: humidity (quantitative)
- **windSpeed**: the wind speed (quantitative)
- **windGust**: the wind gust (quantitative)
- **visibility**: the visibility (quantitative)
- **dewPoint**: the dew point (quantitative)
- **pressure**: the pressure (quantitative)
- **windBearing**: the wind bearing (quantitative)
- **cloudCover**: the cloud cover (quantitative)
- **uvindex**: the uv index (quantitative)
- **ozone**: the index of ozone (quantitative)
- **moonPhase**: the moon phase (quantitative)
- **precipIntensityMax**: the maximum of intensity of precip
- **cab_type:** the type of the cab (qualitative)
- **icon:** summary of the weather (qualitative)
- **destination:** destination of the ride (qualitative)
- **source:** initial source of the ride (qualitative)
- **name:** the type of the ride (qualitative)

Response Variable

- **Price**: price of individual ride (quantitative)

UMAP

UMAP reduce the dimension to 2D and preserves as much of the structure of the data as possible. The color represents cab from two different brand Uber and Lyft.



REGRESSION METHODS AND RMSE

1. Linear Regression (SGDRegressor)
2. Ridge Regression
3. Forward Stepwise Regression
4. Random Forest Regression
5. Lasso Regression
6. Graphical Lasso

The criteria to compare each regression method is RMSE. It measures the standard deviation of the residuals (prediction error).

LINEAR REGRESSION

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression.

To see how well all variables serve the model building process, 100 runs (each run has a new train-test split) on building a linear model is applied through linear_model function in sklearn package. And the average RMSE from these 100 runs is **10^19.**

RIDGE REGRESSION

Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

To see how well all variables serve the model building process, 100 runs (each run has a new train-test split) on building a linear model is applied through linear_model function in sklearn package. And the average RMSE from these 100 runs is **54.8**.

FORWARD STEPWISE REGRESSION

fs and *fsInf* functions from *selectiveinference* package are used for implementing forward stepwise regression and computing p-values and confidence intervals to find out significant variables. After 100 runs, the 11 most frequent variables are found from the selection results.

To see how well these 11 variables serve the model building process, 100 runs (each run has a new train-test split) on building a linear model is applied through linear_model function in sklearn package. And the average RMSE from these 100 runs is **6.62**.

RANDOM FOREST REGRESSION

RandomForestClassifier function from sklearn.ensemble package is used for building random forests, and accuracy_score function from sklearn.metric package is used in the process of determining how many trees to build in a single forest.

Aftering trying 1 to 100 trees, it turns out that having 20 trees, which corresponds to the parameter 'n_estimators' = 20 in the function RandomForestClassifier, gives the highest accuracy core.

With 20 trees in each forest, the RandomForestClassifier is implemented 100 times (each time there is a new train-test split). After which, importance score is given for each variable, and the average RMSE for these 100 runs is **0.40**.

LASSO REGRESSION

Lasso Regression, just like Ridge Regression, penalizes the Linear Regression with a tuning parameter λ. Lasso Regression will force the coefficient of insignificant predictors to be 0, so Lasso Regression is an efficient way to do variable selection. When λ is close to 0, the penalty term is weak and the model selects less predictors when λ increases.

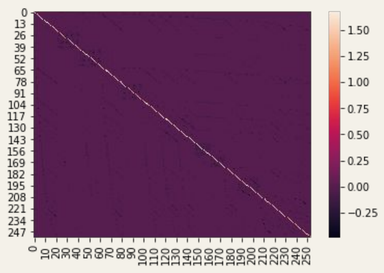
The advantage of Lasso Regression is to avoid overfitting. However, in reality, it might cause underfitting if λ is too large.

In this project, the 10-fold cross-validation is used to obtain the optimal λ with the function LassoCV in sklearn package. And the model is built by plugging the optimal λ in.

By repeating the procedure 100 times to built model and make predictions, the average RMSE of test dataset is **8.16** and there are 11 significant predictors that appears over 30 times among 100 repetitions.

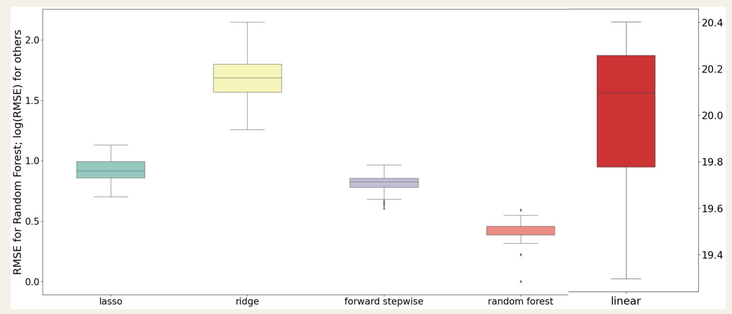
GRAPHICAL LASSO

The graphical lasso is a sparse penalized maximum likelihood estimator for the precision matrix (inverse of covariance matrix) of a multivariate distribution. In this project, the method of graphical lasso is adopted to the matrix of numerical predictors and target variable after removing the empirical means from the columns of the matrix. The figure below shows the results. The precision matrix show that the response variables seems to show no significant relationship with any predictors.



PREDICTION

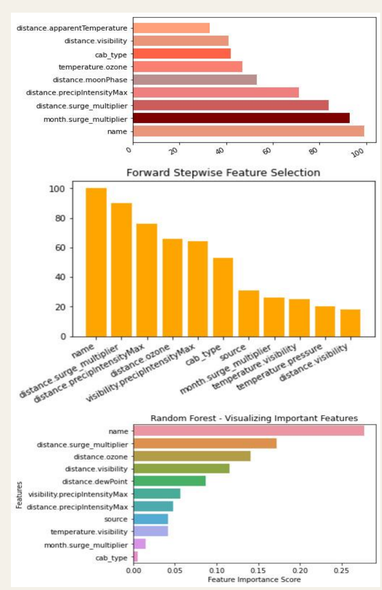
The prediction test error is plotted into four boxplots, which shows the best model is obtained from Random Forest.



CONCLUSION

Among three common dimension reducing regression methods, *Forward Stepwise* is efficient in both model selection and prediction with average RMSE = **6.62**. *LASSO*, with average RMSE = **8.16**, is also a useful tool method to do variable selection but its prediction performance is worse than that of *Forward Stepwise Regression*. And with *Graphical LASSO*, we find no numerical variables that are conditionally correlated with the response variable, so no model is built.

A further proof on the effectiveness of *Forward Stepwise* method providing good variables to build model is that from the prediction of Random Forest model, it presents that although the price difference between two brand, Uber and Lyft is not very significant, when the ride is a standard ride instead of Lyft Lux or UberXL, and if the distance of the ride is short, the price of the ride tends to be low which is consistent with reality.



REFERENCES

Yang: Machine Learning in Action in Finance: Using Graphical Lasso to Identify Trading Pairs in International Stock ETFs. https://towardsdatascience.com/machine-learning-in-action-in-finance-using-graphical-lasso-to-identify-trading-pairs-in-fa00d29c71a7. 2020 Jul 15.

UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2018, Leland McInnes Revision

Choueiry George. c (2021). Understand Forward and Backward Stepwise Regression [Internet]: Quantifyinghealth.com. [cited 2021 Apr 28]. Available from: https://quantifyinghealth.com/stepwise-selection/

Taylor J, Tibshirani RJ. Statistical learning and selective inference. Proceedings of the National Academy of Sciences of the United States of America. 2015;112(25):7629–7634.

Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software. 2010 [accessed 2021 Apr 29];33(1):1–22.

Contact Information

Emails:

Ella Wang (yw701@georgetown.edu)
Shengdan Jin(sj820@georgetown.edu)
Yunzhi Kong (yk659@georgetown.edu)
Yiyang Sheng(ys811@georgetown.edu)

Code Appendix

https://drive.google.com/file/d/1p1vOsBCBddtUbg9C13u48qvNLpLKMEj0/view?usp=sharing