Ride Hailing Supply and Demand Forecasting using Didi-Tech Dataset Intermediate Report

Kimberly Williamson, Gopal Menon
Data Mining, Spring 2017, University of Utah

I. Background

Didi Chuxing is the leading ride hailing company in China and processes over 11 million trips, plans over 9 billion routes and collects over 50TB of data per day. They organized a worldwide algorithm challenge in the year 2016 [1] for forecasting ride supply and demand.

II. Project Scope

The project team will use the 2016 Didi algorithm competition dataset to forecast taxi trip supply, demand, and expected fare for any given date, time, and location using regression models explicated in the Data Mining 2017 Spring semester at the School of Computing of the University of Utah. The accuracy of the forecast will be evaluated using an average forecast error metric.

III. Current Progress

A. Data format conversion

In order to run the regression models, the categorical values in the Didi algorithm competition dataset needed to be converted into a regression friendly format. Depending on the type of categorical value, the new values are lists that consist of 0 values when the category is not present in an order and 1 or a count when the category is present in the order.

B. Data format details

We have converted the data into a format that can be used by the regression models. The input to the regression consists of:

1) An order key, comprised of the categorical values for start district, destination district and order time
2) An order value consisting of
    a) Traffic and points of interest for the start and destination districts
    b) Weather at the time of the order
    c) The value that needs to be predicted, which is the median order price or the number of orders

With the input data created, we have started experimenting with multiple regression models.

C. Data Size

We have 1.3 million rows in our training data consisting of ride hailing orders. When we took into consideration, only those orders where the start and end districts have data for traffic and places of interest, the number of training rows were reduced to $921,000$.

IV. Approaches Tried

We started using scikit-learn [2], the Python Machine Learning library, for running regressions, as it seemed to offer many regression models that can be fine tuned using hyper-parameters.

A. What did not work

The first scikit-learn regression model we tried was Ordinary Least Squares (OLS) regression as we wanted to see how the basic regression model would do in comparison with future models we planned to try. It was to be a sort of baseline for comparison. It did not return a result after running overnight and we realized that most scikit-learn regression models like OLS, Ridge and Lasso will only work [3] when the number of samples is less than $100,000$. For cases where the number of samples is more than this number, the only regression model available was the one using Stochastic Gradient Descent. We used this model and gave it the entire training set. The regression did finish running, but the results were wildly inaccurate with a mean squared error of $\approx 10^{35}$. We then discovered that districts in all orders did not have data for traffic and places of interest. We then removed the orders without this data, but the results were still not accurate.

B. What Worked

We used the scikit-learn Stochastic Gradient Descent (SGD) regression model after removing weather, traffic and points of interest from the regression input. With this data format, our mean squared error dropped to 223 from a value of $\approx 10^{35}$. This was achieved with default hyper-parameters. With 5-fold cross-validation, the error further dropped to 182.

The results of the initial regression can be found below. For predicting number of orders, approximately 13.76% of the values were determined to be significant, meaning the coefficient for the value was greater than 1 or the value was less than -1, in predicting the number of orders. The majority, $\approx 92.12\%$, of the significant values used to predict the number

TABLE I
INITIAL REGRESSION RESULTS

Predicted Variable	Mean Squared Error
Number of Orders	182.777
Order Price	518.663

of orders were part of the time categories. Meaning the time of the day had the most significant impact on the number of orders.

For predicting order price, approximately 23.90% of the values were determined to be significant in predicting the order price. The majority, $\approx 77.55\%$, of the significant values used to predict the number of orders were part of the district categories. Meaning the districts, both start and destination, had the most significant impact on the order price.

V. FUTURE WORK

We plan to experiment with other models of regression including Ridge, Lasso, and Orthogonal Matching Pursuit. Since scikit-learn cannot be used for this purpose, due to the limitation of the amount of data it can process, we will need to use other libraries or code our own solutions. During review of the regression model output the project team may adjust the input values by eliminating or adding variables that result in increased accuracy. We may also eliminate variables that have minimal impact.

REFERENCES

[1] "Algorithm Competition." *Algorithm Competition*. N.p., n.d. Web. 28 Jan. 2017.

[2] "Scikit-learn." *Scikit-learn: Machine Learning in Python - Scikit-learn 0.18.1 Documentation.* N.p., n.d. Web. 19 Mar. 2017.

[3] "Choosing the Right Estimator" *Choosing the Right Estimator - Scikit-learn 0.18.1 Documentation.* N.p., n.d. Web. 19 Mar. 2017.