Capstone Project Proposal
Niko Laskaris
September 24, 2018

DOMAIN BACKGROUND

Price prediction is a category of regression problem with applications in a broad array of industries and academic disciplines. Airlines use complex pricing algorithms to price airline tickets, investors increasingly deploy machine learning-based algorithms to predict stock and asset prices and inform investment decisions. The problem is, at root, a regression problem, but with complex domains comes the need for complex approaches to building predictive models therein.

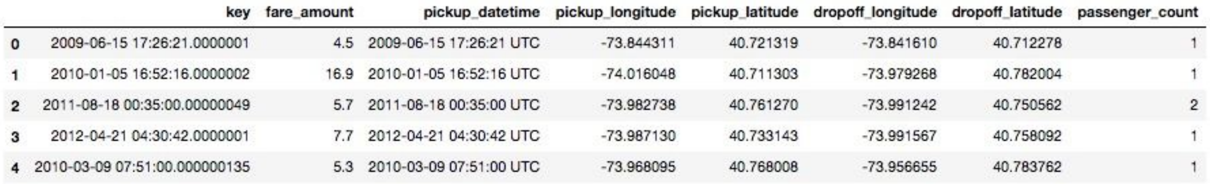
One such domain is predicting taxi fares. One might extrapolate this problem to consider pricing algorithms used by ride-sharing companies such as Uber and Lyft to price rides. Much of the existing scholarship in this domain considers both fare and travel duration prediction , and builds off of work done in the broader category of price prediction problems [1][2][3].

PROBLEM STATEMENT

The primary goal of this project is to predict the fare price of taxi rides in New York City. This is a task taken from the list of Kaggle open competitions. The loss function for measuring model accuracy will be RMSE (root mean squared error). As the problem of price prediction is a regression problem, many regression techniques can be applied: linear regression, Lasso, Ridge, SVRs, Boosting and other ensemble methods, and more.

DATASETS AND INPUTS

The data for this problem comes from Kaggle, and includes samples of data from 2007 to 2012.



	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	2009-06-15 17:26:21.0000001	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.841610	40.712278	1
1	2010-01-05 16:52:16.0000002	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1
2	2011-08-18 00:35:00.00000049	5.7	2011-08-18 00:35:00 UTC	-73.982738	40.761270	-73.991242	40.750562	2
3	2012-04-21 04:30:42.0000001	7.7	2012-04-21 04:30:42 UTC	-73.987130	40.733143	-73.991567	40.758092	1
4	2010-03-09 07:51:00.000000135	5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008	-73.956655	40.783762	1

Each sample contains pickup_datetime, pickup_longitude and latitude, dropoff_longitude and latitude, passenger_count and fare_amount features.

The training dataset contains ~55 million samples, and the testing dataset about 10 thousand. Validation data can and will be stripped from the training dataset. While the feature space is relatively sparse, some clear feature engineering opportunities are to create distance and, perhaps, neighborhood features from the latitude/longitude features, as well as to split the datetime feature into discrete date time fields for time of day, week, month, year, etc.

Background research into the pricing for the NYC metropolitan taxi system will additionally aid in engineering our features, and will likely generate features to/from counties, airports, and other special pricing situations.

SOLUTION STATEMENT

Once the feature space has been established, various regression algorithms will be tested on a subset of the training data to determine which are the best candidates for model selection and tuning. Some such models include Linear Regressors, ElasticNet, Ridge Regressors, Gradient Boosting Regressors, and AdaBoost Regressors. The selected model will undergo hyperparameter tuning and regularization to improve performance.

BENCHMARK MODEL

The official documentation for the Kaggle competition says that a baseline model using just the distance between pickup and dropoff locations will get a RMSE of ~$5-8. Here I use a simple Linear Regression on the cleaned and feature engineered data and achieved a RMSE of ~$5 ($4.98).

EVALUATION METRICS

As our data is labeled we can use a straightforward metric to measure the accuracy of our model. In this case the recommended loss function, and the one I will use, is RMSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where y_i is the target and $\hat{y}_i$ is the prediction. A successful model will produce a RMSE lower than our benchmark model (lower than ~$5).

PROJECT DESIGN

Project Pipeline:

1. Data exploration and cleaning
2. Data and feature engineering
3. Model selection
4. Model tuning
5. Training and testing

[1] *Data exploration and cleaning*. Training and testing datasets will be explored and cleaned of outliers, missing data, etc.

[2] *Data and feature engineering*. Here a variety of features will be created and explored for relevance to predicting fare. Among these will be: distance between pickup and dropoff, various datetime features, neighborhood and location features relative to specific New York taxi pricing codes, fixed rate taxi rides to airports, and more. These features will be tested and either kept or removed from our data. Once we feel satisfied with our feature exploration, the newly cleaned and engineered training and testing datasets will be saved.

[3] *Model selection*. Various regressors will be trained on a subset of our training data to note which are the best candidates for model selection and tuning. Among these are linear regressors, adaboost regressors, gradient boosting regressors, Ridge, Lasso and ElasticNet regressors. Training time will be considered but is not crucial to this task.

[4] *Model tuning*. Once a model is selected from step [3], that model will undergo hyperparameter tuning to optimize its performance, measured by RMSE. Various performance optimizers will be applied to the selected model to optimize performance on the test data.

[5] *Training and testing*. Tuned model will be trained and then used to predict fare on the test data.

References

[1] Antoniades, Christophoros, Fadavi Delara, Foba Amon Jr., Antoine. Fare and Duration Prediction: A Study of New York City Taxi Rides (2016).

[2] Hegazy, Osman & Soliman, Omar S. & Abdul Salam, Mustafa. (2013). A Machine Learning Model for Stock Market Prediction. International Journal of Computer Science and Telecommunications. 4. 17-23.

[3] Vanajakshi, L., S. C. Subramanian, and R. Sivanandan. "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses." IET intelligent transport systems 3.1 (2009): 1-9.