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.

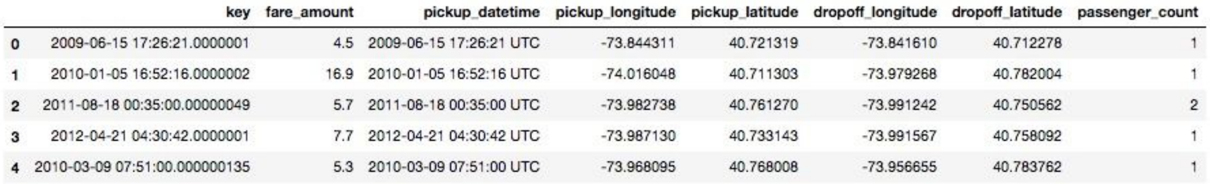
One such domain is predicting taxi fares. One might extrapolate this problem to consider pricing algorithms used by ride-sharing companies such as Uber and Lyft to price rides. Much of the existing scholarship in this domain considers both fare and travel duration prediction , and builds off of work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

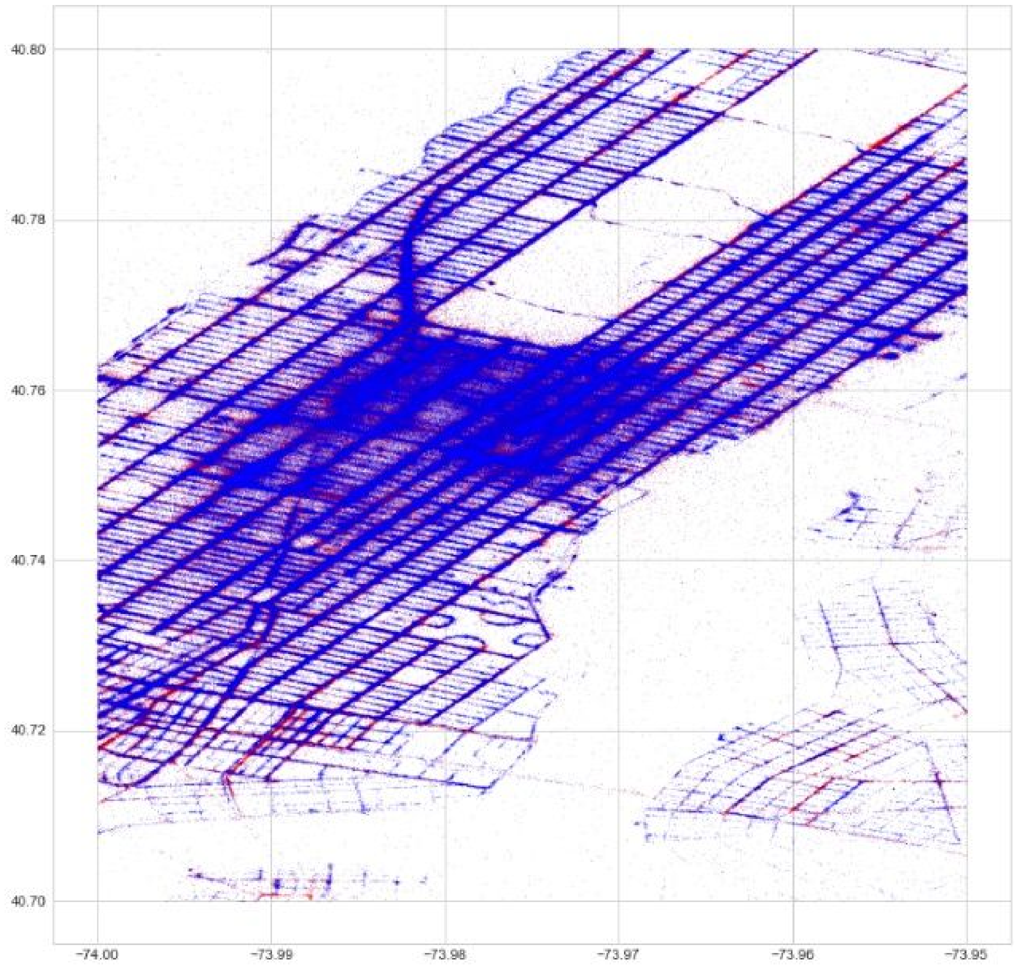
NOTES ON PROJECT

[0] Data exploration and cleaning

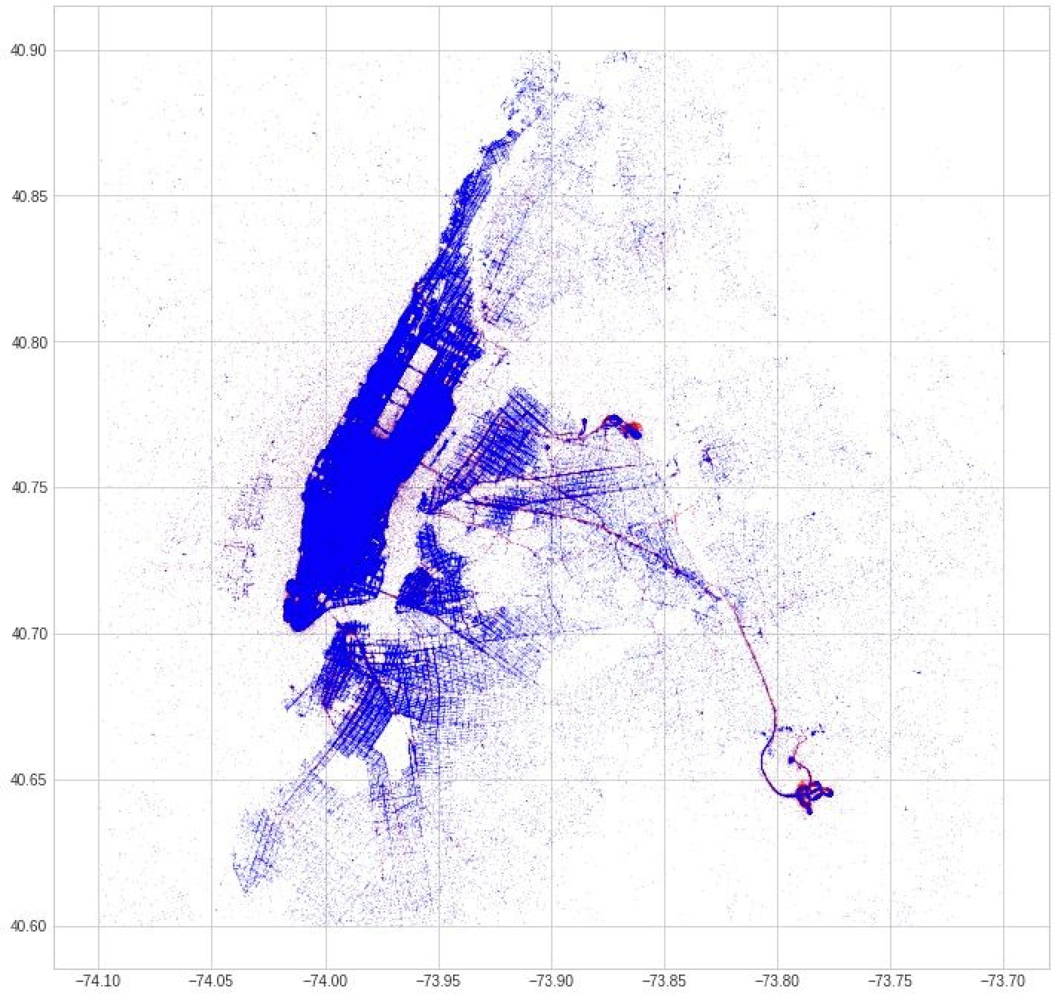
Began with 15M samples from the ~55M sample training dataset. Checked for null and zero values and removed them. Used the parameters of the testing dataset to crop the training dataset for outliers. Split the datetime feature into discrete time features for hour, day, month, year, etc. All told data cleaning resulted in a dataset with 14,628,013 samples of 13 features.

The data was fairly clean and easy to work with. I used a subset of the full training simply due to the training set's size. A note for further work on this project would be to open an EC2 instance or otherwise get ahold of more computing power to train with the full dataset (although results were fairly good using a subset).

[1] Visualizations



Fun to visualize the data.



[2] Feature Generation / Data Exploration

Notes on pricing:

From the NYC taxi and limousine commission:
http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml

The initial charge is $2.50. Plus 50 cents per 1/5 mile or 50 cents per 60 seconds in slow traffic or when the vehicle is stopped. In moving traffic on Manhattan streets, the meter should “click” approximately every four downtown blocks, or one block going cross-town (East-West). There is a 50-cent MTA State Surcharge for all trips that end in New York City or Nassau, Suffolk, Westchester, Rockland, Dutchess, Orange or

Putnam Counties. There is a 30-cent Improvement Surcharge. There is a daily 50-cent surcharge from 8pm to 6am.

Since the initial charge for every taxi ride in New York is 2.50 since 2012 (and 2.00 since earlier than our dataset starts recording), let's see how many samples have a fare at or below the $2.00 boundary.From the NYC taxi and limousine commission: http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml

The initial charge is $2.50. Plus 50 cents per 1/5 mile or 50 cents per 60 seconds in slow traffic or when the vehicle is stopped. In moving traffic on Manhattan streets, the meter should “click” approximately every four downtown blocks, or one block going cross-town (East-West). There is a 50-cent MTA State Surcharge for all trips that end in New York City or Nassau, Suffolk, Westchester, Rockland, Dutchess, Orange or Putnam Counties. There is a 30-cent Improvement Surcharge. There is a daily 50-cent surcharge from 8pm to 6am.

Since the initial charge for every taxi ride in New York is 2.50 since 2012 (and 2.00 since earlier than our dataset starts recording), let's see how many samples have a fare at or below the $2.00 boundary.

****

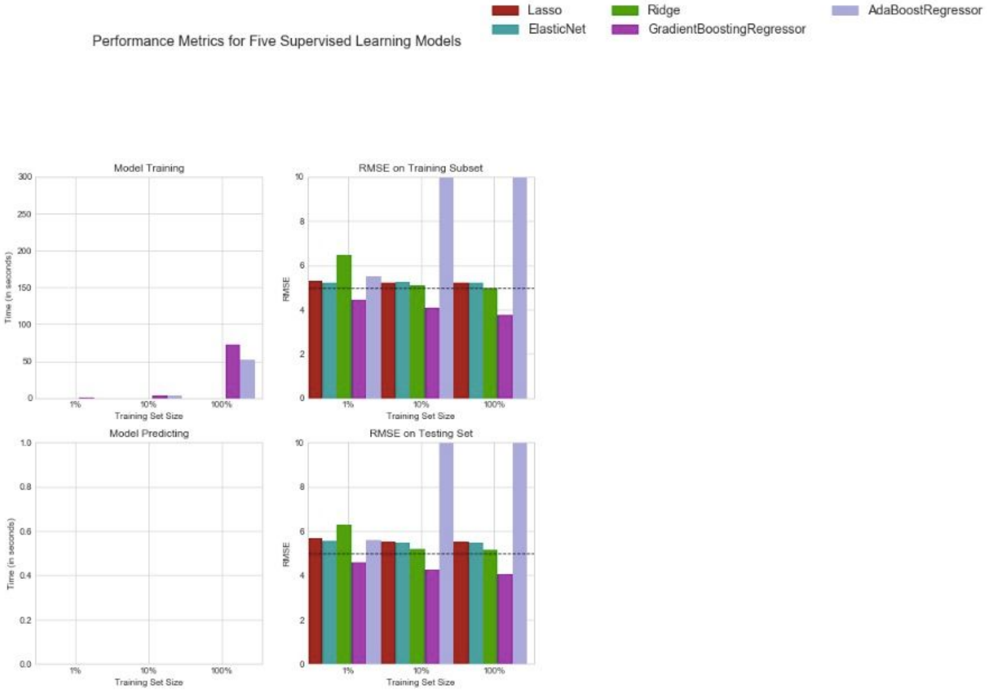
Added Distance feature using Haversine distance. Created dropoff and pickup features for each of the counties mentioned above that have unique pricing. Log-transformed distance. Checked correlation with fare amount and eliminated a few of the newly generated features.

[3] Model selection

I used a simple linear regression as a baseline model, and got a RMSE of 4.95 on the testing (validation) data.

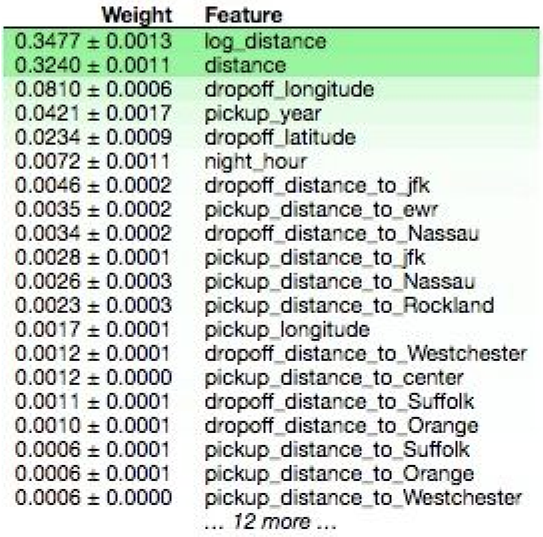
I borrowed visuals and some code from the Udacity Finding Donors project from earlier in the program to test several regressors on a subset of the data. Tested:

- Lasso
- ElasticNet
- Ridge
- GradientBoost
- AdaBoost



Of these GradientBoost, while the most computationally expensive, performed best.

I decided to go ahead with GradientBoost as my model. On that same subset of the data, I performed a Permutation Importance test to double check that my features were all useful. I tried making an arbitrary cutoff based on permutation importance test scores to eliminate some of my features, but the truncated model actually performed worse on the subset of training data. I decided to keep my original set of features. *Note: This could be another place for future improvements.*



Weight	Feature
0.3477 ± 0.0013	log_distance
0.3240 ± 0.0011	distance
0.0810 ± 0.0006	dropoff_longitude
0.0421 ± 0.0017	pickup_year
0.0234 ± 0.0009	dropoff_latitude
0.0072 ± 0.0011	night_hour
0.0046 ± 0.0002	dropoff_distance_to_jfk
0.0035 ± 0.0002	pickup_distance_to_ewr
0.0034 ± 0.0002	dropoff_distance_to_Nassau
0.0028 ± 0.0001	pickup_distance_to_jfk
0.0026 ± 0.0003	pickup_distance_to_Nassau
0.0023 ± 0.0003	pickup_distance_to_Rockland
0.0017 ± 0.0001	pickup_longitude
0.0012 ± 0.0001	dropoff_distance_to_Westchester
0.0012 ± 0.0000	pickup_distance_to_center
0.0011 ± 0.0001	dropoff_distance_to_Suffolk
0.0010 ± 0.0001	dropoff_distance_to_Orange
0.0006 ± 0.0001	pickup_distance_to_Suffolk
0.0006 ± 0.0001	pickup_distance_to_Orange
0.0006 ± 0.0000	pickup_distance_to_Westchester
	… 12 more …

Once I had determined my model (GradientBoostRegressor) and had decided to keep my original feature space after conducting a permutation importance test, I performed a Grid analysis to tune the hyperparameters of my model (using 1M samples).

Next I introduced a similar version of GradientBoost, Light Gradient Boost (LGBM). LGBM was delivering slightly lower RMSE scores than the original model, so I proceeded using LGBM. I trained first 3M and then 15M samples with the LGBM model (hyperparameter tuned according to my Grid Search). My best RMSE score once submitted to Kaggle was ~3.066, good for the top 25% of the competition.

Some thoughts on avenues to pursue for future work on this project / model improvements:

- Continue to tweak feature space. We wound up with a lot of features for each county drop off or pick up that had unique pricing. Perhaps principal component analysis could reduce the featurespace based on explained variance.
- Could train with more data. I trained with 15M samples, but the dataset was ~55M.
- Continue to tweak hyperparameters. Even though I performed several Grid searches to optimize certain hyperparameters, could do with more.
- Explore even more models. Dip into neural nets, perhaps.

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.