CIS 520 FINAL PROJECT: Chicago Taxi Price Prediction

Ye Dong, Xufei Huang, Yujie Sun

ABSTRACT

The taxi business in Chicago is dynamic, and it has become a more interesting field of study with the advent of ridesharing companies. This project aims to predict taxi fare in Chicago city. We explore and visualize our dataset to select features. Then, we use various regression techniques, and they were able to generate fairly accurate results.

1. Motivation

Our project Chicago Taxi Trips predicts taxi fare in Chicago city using the publicly available Chicago taxi data (from Oct.10, 2019 to Nov.10, 2019). The transportation industry in Chicago has always been an interesting topic of study, and it has become more dynamic with the advent of new ridesharing companies like Uber and Lyft. The results of our project can help taxi firms in Chicago maximize their revenue and remain competitive in the industry, and help Chicago Transit Authority better allocate their resources and make transportation in the city more efficient.

We think machine learning is particularly suitable to solve this problem. First, the question we are interested in is a predictive problem. Our dataset includes observable outcomes (fare amount) and relevant features including starting and ending times of the trips, length of trips in both time and distance. ML models are helpful for determining how data features are related to the outcomes of interest. Second, our dataset is sufficiently self-contained. The features are pretty comprehensive and include many influential variables characterizing a trip, which means there would be enough for the learning algorithms to understand the problem.

2. Related work

We have looked through two sources to get a basic idea of how people have been approaching similar problems in the past.

The first project [1] we looked at predicted taxi fare in Chicago using Chicago Taxi Data from the year 2013-2017(July). They first performed Exploratory Data Analysis, which includes both visual and numerical analysis. Then, they used three algorithms

for prediction: Linear Regression, Random Forest and LightGBM, and two metrics for evaluation: RMSE and R-squared. To reduce overfitting, they used cross-validation.
The exploratory data analysis project 1 performed helped them better understand the data and prepared them for their future analysis. It was particularly useful for data cleaning (removing outliers like zero trip seconds and zero trip fare values) and feature selection (identifying that time has strong predictive power). The three models they used each had its own advantages and disadvantages. Linear Regression performed well on smaller dataset (weekly data), but failed to fully capture the non-linear relationship in the model. Random Forest and LightGBM, in comparison, performed better on larger dataset (hourly and daily data) but seemed to be overfitting the data even after cross-validation.

The second project [2] analyzed a random sample of 49999 NYC taxi trips (specifically in Manhattan) in 2013 using regression trees and random forests to predict the taxi fare based primarily on locations and time measurements. Then they used density map to visualize the actual fare/predicted fare in the cities. The random forest model had better performance and was able to capture some of the patterns of the data. One regression tree was not enough because only the most predictive variable latitude is splitted on, and random forest, where many trees are fitted, successfully included other variables.

3. Dataset

The dataset we are using is the publicly available Chicago data. The dataset can be found at:
https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew/data

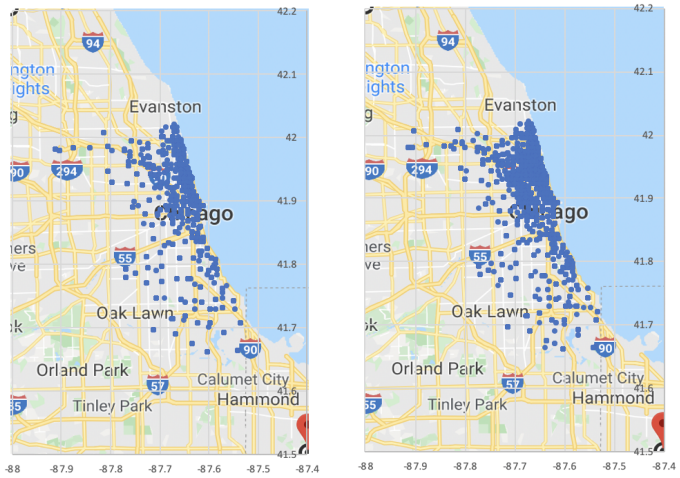
The original dataset has 23 features with 188,451,469 entries.

Data Pre-processing:
Due to the huge volume of data, we only analyzed taxi-ride within the past month. We first tried to deleted faulty data with qualities like trips with 0 miles yet 0 seconds, and dropped features that are irrelevant to price predicting(i.e. Trip ID, Taxi ID, pickup/-dropoff census tract, Payment types). Since we were trying to estimate the total fare, we dropped fare (i.e. Pre tips/extra) since this feature is redundant with the result.

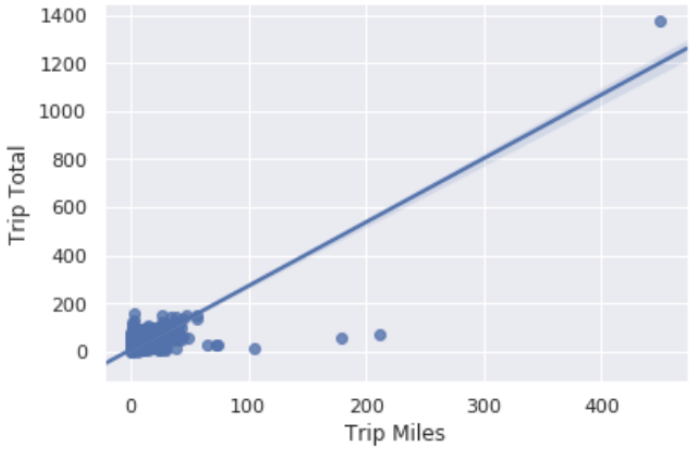
All the data containing ‘NaN’ were ignored because the data missing longitude and latitude could not be used. 103731 out of 115057 set were remained. The feature “Tolls” were dropped because only 7 out of 103731 set of data have a non-zero “Tolls”. Then transfer all the numbers into float type.

Now to we want to determine if the features we have left are really relevant to the

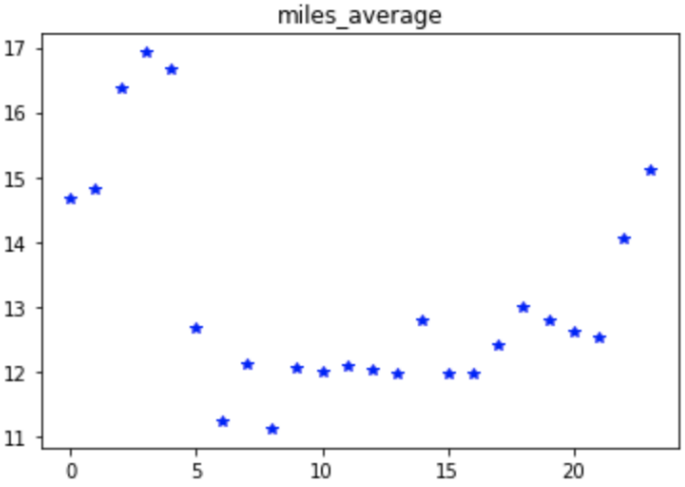
total fare as we expected. We first plotted the pick up(left)/drop off(right) centroid locations on the map.



Note the points here are very concentrated since we are only considering rides within the city and if we consider the bounds of latitude and longitude, this feature will be much more insignificant than the ride length features. In addition, since we are dealing with taxi instead of rider app services like uber or lyft, the price within an area should be the same. However note that the distribution for fare v. distance graph. Therefore, we discarded the pick up/drop off locations from the set of features. However, relying on the common conception that taxi keep track of prices by distance, we will be using total distance as a feature. But just as a double check, this is what total distance v. time plot looks like:



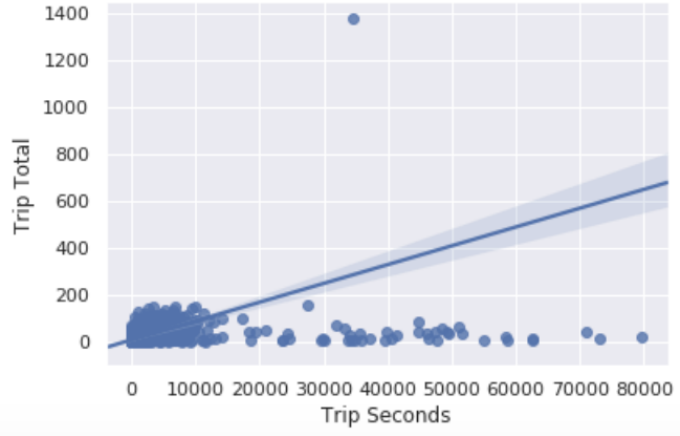
Note the correlation coefficient here is only about 0.9 which already shows a pretty strong linear correlation. This confirms our choice of keeping trip time.



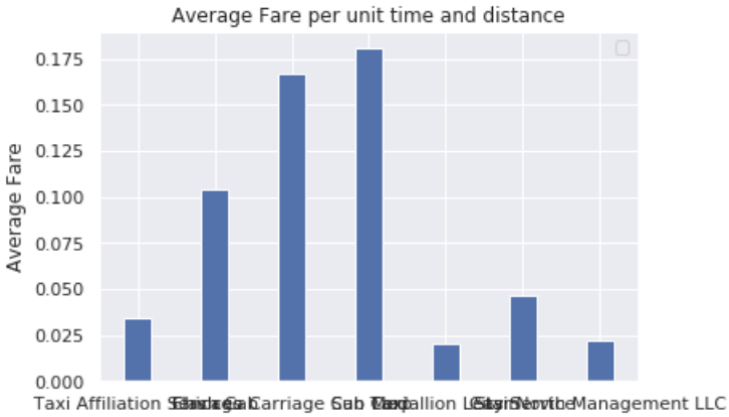
Similarly, we examined the importance of time stamp of the feature. Since we already know that the relationship between mile and fare is roughly linear, we can do that by looking at the average fare per mile show right above. Note the fare between 10-4am is much higher than other times. So we added a binary feature called midnight, which is 1 if the trip took place between 10-4am and 0 otherwise. Note since time stamp only goes back 2 months, we won't

have enough data to examine effect of taxi over time.(due to global economy reasons) So we won't need the time stamps otherwise, and decided to drop those two features.

In addition, since the fare does go up at time of a congestion or other special scenarios, the time of the trip should be taken into consideration when predicting taxi price. Note the correlation coefficient here is 0.6, which means that this is not as correlated as trip distance, which is expected. A graph of the fare v. time plot is shown below. Here not only the best line of fit does not fit the patterns as well but also there is a larger variance towards when both trip time and fare are larger.



Since each company might be evaluating fares differently, trip fare variation could be biased among companies.(in fact looking at the graph we plotted below for avg fare v. the 7 companies with most data, it is!) we evaluated the number of data from each company and realized that the population size varies from 159k to 11k. To avoid unbalanced data, we originally decided to delete data from the companies with less data but then realized from our sources that we could combine the smallest pools to make the data more balanced. Therefore, we treated all companies with less than 25k data as one company ("others") and each of the other ones as their own. The top 7 companies have 159024, 130524, 100582, 73094, 68076, 65283, 47949 records respectively, and the remaining 12 companies only have 155020 records in total.



To make the companies into numerical data, we used one-hot encoding, where each company will have a column and each data set will have a 1 for the company that the taxi belongs to and 0 otherwise.

We graphed the average fare based on miles (since that is how taxi fare is designed to be measured) and found that the average fare per mile during 10pm - 4am is significantly higher than that during the other period, as shown in the figure below. Thus, we decided to add a dummy variable to differentiate trips started during 10pm-4pm from other trips.

Data Summarization After processing, we are left with 15 features and 799552 entries. The remaining features are:

- Trip Seconds: length of the trip in time
- Trip Miles: length of the trip in distance
- Tips: the amount of tips of the trip
- Extras: other components of the trip cost
- mid_night: a dummy variable which is 1 if the start time of the trip is between 10pm and 4am, and 0 otherwise
- Taxi Affiliation Services, Flash Cab, Chicago Carriage Cab Corp, Sun Taxi, Medallion Leasin, City Service, Star North Management LLC, and Others: company dummies indicating the taxi company of the trip

4. Problem Formulation

We will predict the price of the taxi fare for Chicago Taxi based on features like miles traveled, time of the day, and taxi company (specific features will be determined via feature selection) via different regression models.

We chose L2 loss as our loss function. We decided to use L2 loss because there aren't a lot of outliers in our data. Out of 100,000 data points, there are 66 outliers. Additionally, mean squared error is essentially the best error function for regression models.

Our data set consists of 100,000 data on the taxi trips people have taken during the past month (10/10-11/10). From the above data observations, we make the hypothesis that the fare should be roughly linear with distance and time. However, running a linear regression on the cleaned data (by the above procedure) showed that Tips, and extra has a very strong linear relationship with the trip total while the linear relationship between trip total and trip hours/miles with trip total was very weak. This either means that time and distance might not have as strong a linear relationship with fare as we expected. Therefore, we will be using one linear method - regularized linear regression- to check if we overfitted the data, which made the linear relationship between fare and time/distance to appear so weak, and two nonlinear methods- random forest regression and kernel regression - in case the relationship between total fare and time/distance is really non-linear.

We have selected features based on the process and reasoning described above in the data processing section.

5. Methods

Note we split the data into 5 portions and used 1 for testing and rest for training. Also all the cross validation were five fold cross validations.

Since here we are predicting the taxi fare, which is a numerical value, we want to use regression. Therefore we choose linear regression because it is the most straight-

forward way of evaluating if the features have any relationships with the variable we are exploring, even if some of the features might not have linear relationship with the model. In addition, since it is the most basic model of fit and doesn't require additional parameter input like random forests or batch regression, it is much less likely to overfit compared to the more advanced models like ridge regression and nonlinear models. In addition, the linear regression model also doesn't require additional parameter input like random forests or batch regression. The less "choice of design" we have for the base model, the more we will be able to leverage our prediction using the other models. We built this model using sklearn's linear regression function and used sklearn's cross valdiation score function to assess how much our model overfitted.

The other methods we chose are ridge regression, random forest regression, and kernel ridge regression. Considering that the features may or may not be linearly related to the y variable, we used an linear (linear and L2 regressions) and nonlinear models (random forest and kernel ridge regressions).
We built L2 regression using sklearn's ridge regression model and cross validated it using sklearn's RidgeCV function.
We decided to use random forest regression because the random forest model is good at handling data with numerical features. It also helps capture non-linear relationship between features and the y-variable.
We chose ridge regression because it is useful for analyzing data that suffer from multicollinearity and we believed near-linear relationships might exist among some of our features. It reduces the model complexity and multicollinearity by adding a penalty equivalent to square of the magnitude of the coefficients. We built random forest regression using sklearn's RandomForestRegressor and cross validated it using sklearn's RandomizedSearchCV function.
Kernel ridge regression is a non-parametric technique. We chose kernel regression because it is able to capture non-linear interaction between features and the y-variable. Due to our massive dataset, we weren't able to feed in the whole dataset directly into sklearn's kernel regression. So we broke the data down into 4 per set and feed them into the regression one by one, and predict results after feeding in the data.

6. Experiments and Results

The hyperparameters for the random forest regression were initialized using sklearn's default parameters.
To choose hyperparameters for the kernel regression, we need to decide an appropriate alpha, gamma, and kernel function. Since actually running them through all data sets takes forever, we only run about 2000 of them each time and omit the ones that gives awful error. So our candidates for kernels

are linear kernel, rbf, and polynomial kernel. Note polynomial kernels gives error up to millions already so we will omit it and focus on rbf and linear kernels. To determine alpha and gamma, I did a lot of trying out random numbers and swapping my inner and outer loops of the function so I can better compare my results. At the end, I got the best result using $\alpha = 1, \gamma = 0.01$ with linear kernel and eventually got 214.5 as the MSE. We wanted to perform a 5-fold cross validation to check on over fitting, but running this once already took a long time and running it 5-fold made my program crush easily. So we just decided to not include this.

The performance metrics we used were cross-validation accuracy and MSE Error. We could not use the common methods such as ROC curve and F-1 score because the predicted value y is continuous. The performance results of across all our models are summarized in the table below:

	CV Accuracy	MSE Error
Linear Regression	47.3%	235.149
Ridge Regression	47.3%	235.149
Random Forest Regression	88.42%	26.0
Kernel Ridge Regression	-	214.5

7. Conclusion and Discussion

In conclusion, we were able to build fairly accurate models that predict taxi fare in Chicago city using the publicly available taxi data. All three methods, Ridge Regression, Random Forest Regression, and Kernel Ridge Regression performed better than our baseline method in terms of MSE error.(although not very significant for ridge regression) Specifically, the random forest regression performed the best, kernel ridge regression the second, and ridge regression the third. Both random forest and kernel ridge regressions improved performance by capturing non-linearity in the dataset. They were able to reduce the error significantly without over-fit the data. The improvement of ridge regression, on the other hand, was minimal. Its CV accuracy was also pretty low (only around 50%), which suggests that it over-fit the data even after cross-validation.

An interesting fact we observed is that the linear and ridge regression produced such similar result and KKR produced the (est.)best result when it had a linear kernel and still had a MSE similar to that of linear regression. This might be a sign that the relationship between taxi fare and the parameters are almost linear. Note in the data analysis section, we saw that there existed a strong linear relationship between fare and distance and not so strong relationship between fare and travel time. The other aspects we included in the model likely mattered yet

won't be too significant. Therefore we can make the deduction that fare is based mostly on trip distance with a little variability coming from trip time.

One thing we learned from the overall project process was the importance of data pre-processing and exploration. Pre-processing allowed us to drop outliers and faulty data points, and irrelavant features, which increased model accuracy. Initial exploration gave us a better understanding of the dataset, which helped us select features and consequently build algorithms that yield better results.

We also learned that running complex models on a large dataset like this one could take a long time, so it is important to find a balance between model efficiency and accuracy. For example, initially we planned to do cross-validation on all models, but we realized this was not a good idea given the time constraint. Similarly at the case of doing grid search, once you find the range of parameters, it might take you a long time to get the absolute global optimal(if possible) yet only improve the mse when running the testing data on the model by <10.

In addition, we also learned the importance of "reading" in the cases of doing this type of project. We think looking at other people's approaches to similar problems really help because they might have thought of approaches or assumptions that we would not have otherwise noticed. We thought it was really interesting on how doing data-science is very similar to doing research in this aspect.

In the future, we are interested in predicting taxi price using the entire dataset, which will allow us to get more accuracy. In addition, we also hope to get a GPU or some type of parallelization service that can speed up the proccess of gride search so we can increase our search space and get the best searching parameters as possible.

References

[1] Mustafa Panbiharwala. Prediction of Pickup Density for Chicago Cabs, 2018. *https://github.com/mustafashabbir10/Prediction-of-Pickup-Density-for-Chicago-Cabs*.

[2] Noahweber1. Prediction of Pickup Density for Chicago Cabs. *https://github.com/noahweber1/datacamp-project---PREDICT-TAXI-FARES-WITH-RANDOM-FORESTS/blob/master/notebook.ipynb*.