Fare Prediction for New York Taxi

Adnan Elahi
Jinhui Liu
Shilpa Konde Deshmukh
Siying Chen
Department of Information Systems, California State University
Los Angeles
e-mail :aelahi@calstatela.edu, jliu2@calstatela.edu, skonded@calstatela.edu, schen112@calstatela.edu
Mentor: Professor Jongwook Woo

Abstract:
Fare Prediction for Taxi can reveal the trends of the gain for taxi drivers from New York City, as one of the most popular travel destinations worldwide, which may help local authorities optimize and improve their transportation planning to reach the growing need in the city. In this project, we analyze the relative dataset by applying big data tools such as Hive and Azure Ml to conclude the trend and predict the outcome based on the model we build.

1. Introduction

Our group project is based on the datasets about new york taxi trips coming from Kaggle.com and Chriswhong.com, with specific time restrictions, which heavily rely on the distance of the trip to predict the potential outcome. Azure ML Studio, Databricks Community Edition, and Hadoop are the primary services we apply in this project. By comparing the outcome from those services, we can come out with the most accurate and reliable analysis tool in this project.

2. Related Work

From “NYC Taxi Fare Prediction”, reported by Allen Kong[1], training their dataset cost more than 8 hours by using CPU locally, and 17 minutes by using GPU locally. Compared to that, we use cloud services to train our datasets, and the cost time is less than 5 minutes. And according to Di Wu, reporting in “New York City Taxi Fare Prediction”[2]. They use LightGBM for their algorithm and return a RMSE with 2.96126. However, they did not provide the information of coefficient of determination(R2). And they used 55 million for the train set, but only 9914 for the test set. We split our datasets by 70% for the train set and 30% for the test set. As a result, our best RMSE returns around 3.4599 and R2 as 0.881874 for Boosted Decision Tree Regression. Based on “Predicting the Taxi Fare of Chicago Cabs”[3], reported by Thomas Synnott, they set up a project to predict the fare amount of taxi trips in chicago. But they only used linear regression in their project, and a small sample dataset size - 10,000. Their results do return a good RMSE and R2, but they also do a lot of work to modify their dataset. Compared to them, we used a larger sample dataset size and without modifying the datasets too much. I believe if we modify the datasets more, the result could be better.

3. General Instructions

The first dataset we are using is from Kaggle.com. We are only using the data from Nov 2019 to Feb 2020. And the second dataset is from Chriswhong.com. Both these two datasets are about new york taxi trips. By modifying the two datasets, we create three sample datasets as shown in table 1.

Table 1. Sample Datasets Details

	KTSample	fareSample	tripSample
File Size	10MB	9.99MB	9.92MB
Num. Row	113,400	92,530	62,700
Num. Col	18	11	14

And the details of original datasets are shown in table 2.

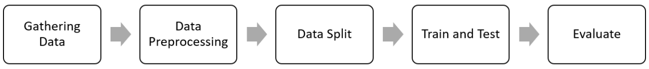
Table 2. Original Datasets Details

	KaggleTaxi	fare	trip
File Size	2.26 GB	1.56 GB	2.29 GB
Num. Row	26,478,791	14,776,616	14,776,616
Num. Col	18	11	14

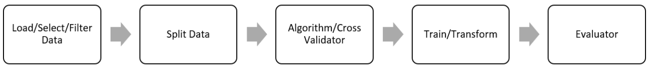
Services used in this project include Azure ML Studio(classic), Databricks Community Edition and Hadoop. The specification of Azure ML Studio is 10 GB storage with a single node. The specification of Databricks Community Edition is 15.3 GB memory size, 1 Driver, 2 cores, 1 DBU with Runtime: 8.1(Scala 2.12, Spark 3.1.1). And Hadoop using the spark with 2.3.2.3.1.4.0-315 version.

3.1 Workflow

To start, we load the datasets to Azure ML/Databricks/Hadoop first. We do the data cleaning by selecting columns we need and removing any rows with null value. Then we split the datasets to train and test datasets. For Azure ML, we train and test the data and give the evaluation. For Databricks and Hadoop, we build algorithms/cross validators before training/transforming. and give evaluation at the end.
Azure ML:



Databricks/Hadoop:



3.2 Azure ML

Here we have used three algorithms to build two models to compare the results and predict the outcome. One is Decision Forest Regression, Linear Regression and Boosted Decision Tree Regression. The data source input has been taken from three sources: fairsample.csv, tripsample.csv and kaggletaxisample.csv. Add Column is used to combine columns from two sources files. Next Select Columns in Dataset model is used to select the columns which will be used in prediction. Cleaning Missing Data is used to clean the data and algorithms are applied to get the score of RMSE and COD .

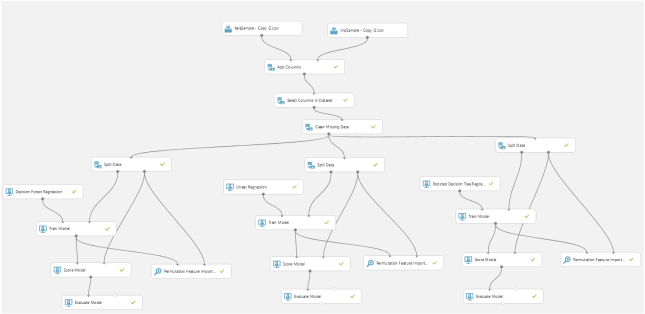
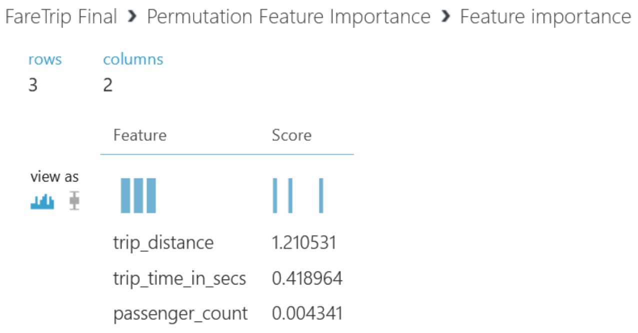

fareSample - Copy (2).csv
tripSample - Copy (2).csv
Add Columns
Select Columns in Dataset
Clean Missing Data
Split Data
Split Data
Split Data
Decision Forest Regression
Linear Regression
Boosted Decision Tree Regre...
Train Model
Train Model
Train Model
Score Model
Score Model
Score Model
Permutation Feature Import...
Permutation Feature Import...
Permutation Feature Import...
Evaluate Model
Evaluate Model
Evaluate Model

Figure 1. Microsoft Azure Model

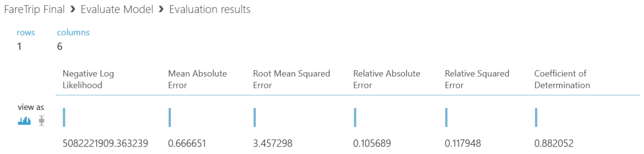
Trip distance is the most important feature in our table.


FareTrip Final › Permutation Feature Importance › Feature importance

rows	columns
3	2

Feature	Score
trip_distance	1.210531
trip_time_in_secs	0.418964
passenger_count	0.004341

Figure 2. Feature importance In azure ML

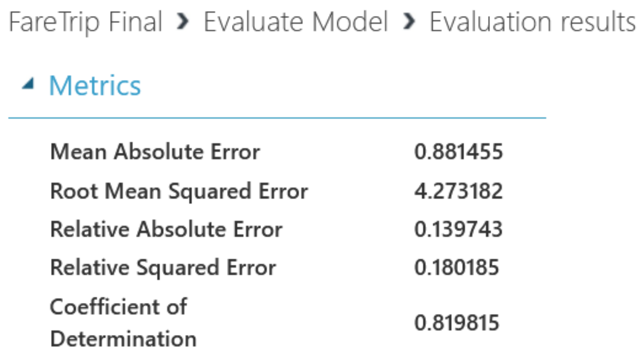

FareTrip Final › Evaluate Model › Evaluation results

rows	columns
1	6

Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
5082221909.363239	0.666651	3.457298	0.105689	0.117948	0.882052

Figure 3. Decision Forest Regression

- ALGORITHM USED: DECISION FOREST REGRESSION
- SPLIT DATA: 70 -30 is Randomized split
- RMSE: 3.457
- COD: 0.882

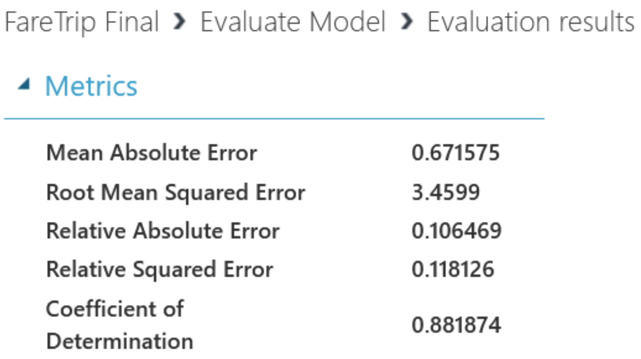

FareTrip Final › Evaluate Model › Evaluation results

Metrics

Mean Absolute Error	0.881455
Root Mean Squared Error	4.273182
Relative Absolute Error	0.139743
Relative Squared Error	0.180185
Coefficient of Determination	0.819815

Figure 4. Linear Regression

- ALGORITHM USED: LINEAR REGRESSION
- SPLIT DATA: 70 -30 is Randomized split
- RMSE: 4.273
- COD: 0.819


FareTrip Final › Evaluate Model › Evaluation results

Metrics

Mean Absolute Error	0.671575
Root Mean Squared Error	3.4599
Relative Absolute Error	0.106469
Relative Squared Error	0.118126
Coefficient of Determination	0.881874

Figure 5. Boosted Decision Tree Regression

- ALGORITHM USED: BOOSTED DECISION TREE REGRESSION
- SPLIT DATA: 70 -30 is Randomized split
- RMSE: 3.459
- COD: 0.881

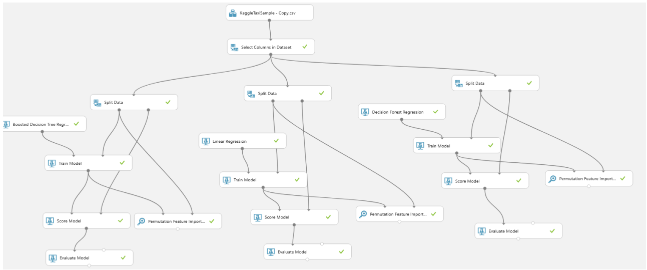
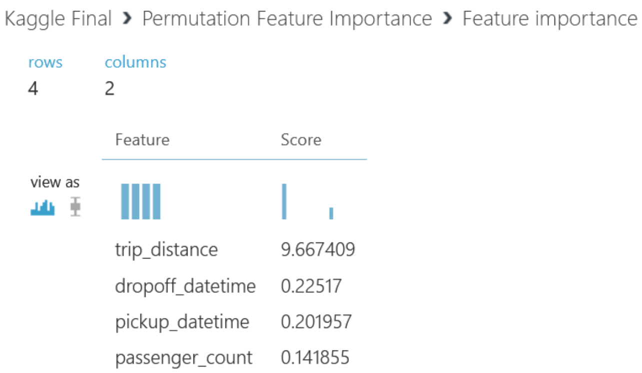

KaggleTaxiSample - Copy.csv
Select Columns in Dataset
Split Data
Split Data
Split Data
Boosted Decision Tree Regr...
Linear Regression
Decision Forest Regression
Train Model
Train Model
Train Model
Score Model
Score Model
Score Model
Permutation Feature Import...
Permutation Feature Import...
Permutation Feature Import...
Evaluate Model
Evaluate Model
Evaluate Model

Figure 6. Microsoft Azure Model

Trip distance is the most important feature in our table.

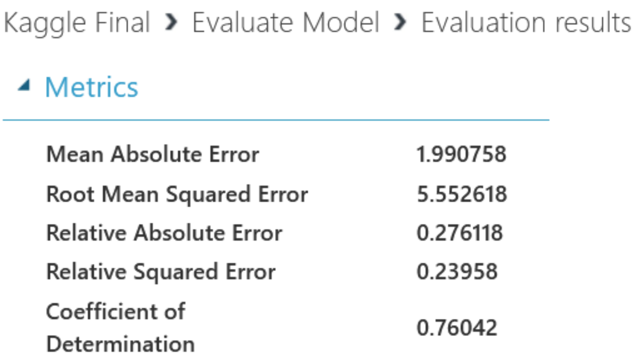


Kaggle Final › Permutation Feature Importance › Feature importance

rows	columns
4	2

Feature	Score
trip_distance	9.667409
dropoff_datetime	0.22517
pickup_datetime	0.201957
passenger_count	0.141855

Figure 7. Feature importance In azure ML



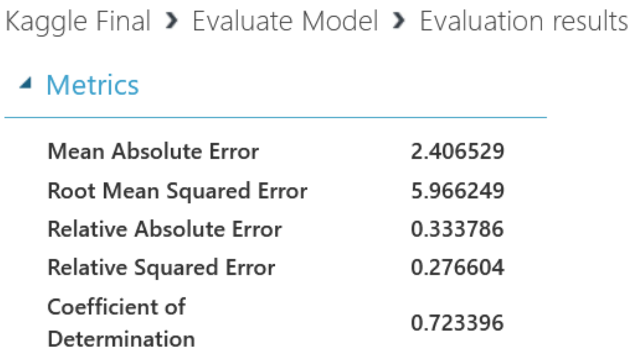
Kaggle Final › Evaluate Model › Evaluation results

Metrics

Mean Absolute Error	1.990758
Root Mean Squared Error	5.552618
Relative Absolute Error	0.276118
Relative Squared Error	0.23958
Coefficient of Determination	0.76042

Figure 8. Boosted Decision Tree Regression

- ALGORITHM USED: BOOSTED DECISION TREE REGRESSION
- SPLIT DATA: 70 -30 is Randomized split
- RMSE: 5.552
- COD: 0.760



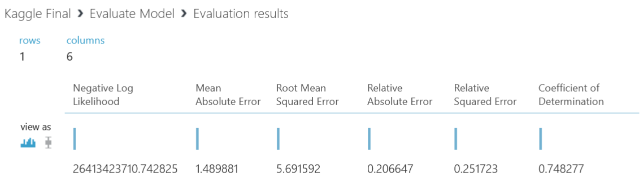
Kaggle Final › Evaluate Model › Evaluation results

Metrics

Mean Absolute Error	2.406529
Root Mean Squared Error	5.966249
Relative Absolute Error	0.333786
Relative Squared Error	0.276604
Coefficient of Determination	0.723396

Figure 9. Linear Regression

- ALGORITHM USED: LINEAR REGRESSION
- SPLIT DATA: 70 -30 is Randomized split
- RMSE: 5.966
- COD: 0.723



Kaggle Final › Evaluate Model › Evaluation results

rows	columns
1	6

Negative Log Likelihood	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error	Relative Squared Error	Coefficient of Determination
26413423710.742825	1.489881	5.691592	0.206647	0.251723	0.748277

Figure 10. Decision Forest Regression

- ALGORITHM USED: DECISION FOREST REGRESSION
- SPLIT DATA: 70 -30 is Randomized split
- RMSE: 5.691
- COD: 0.748

As we can see from figure 1 to figure 10, we have tested two datasets by using Azure ML. It will take approximately 1 minutes for each dataset. Boosted decision tree regression returns the best RMSE. And R2 for all the algorithms is between 0.7 to 0.9.

3.3 Databricks

Spark machine learning is used in both Databricks and hadoop. We have built the databricks project based on the Azure ML. According to the Azure ML, we built three algorithms, which include boosted decision tree regression, linear regression and decision forest regression. Both boosted decision tree regression and decision forest regression are built with cross validators. And linear regression is built with train validation split.

For the Kaggle dataset, we have calculated the trip time in seconds, before we split it to train and test datasets. For the fare and trip datasets, we have to insert a column named “id” to each table and inner join them for combination.

The purpose of using databricks is to calculate root mean square error and coefficient of determination for all three algorithms. And it will take approximately 4 minutes for each datasets.

Table 3. Kaggle Databricks Results

	RMSE	R2
GBT	5.341649852	0.778040532
Linear	5.505647477	0.764202257
Decision	6.021368513	0.717958343

Table 4. Fare and Trip Databricks Results

	RMSE	R2
GBT	9.504551261	0.155117871
Linear	9.535217876	0.149657019

Decision	9.569506788	0.143530308

As we can see above, compared to the other two algorithms, boosted decision tree regression has the smallest RMSE, and the highest R2. However, Fare and trip dataset does not return a good R2 as Kaggle or even fare-trip from Azure ML. One of the reasons can be that the fare-trip dataset are using the combination of two separate datasets.

3.4 Hadoop Spark

After we finished Databricks, we just exported the code as py files and used it in Hadoop Spark. We set up the same parameters as we have in databricks. The maxDepth is set from 2 to 3, and maxIter is from 10 to 20. For having more accuracy results, please set up the maxDepth more than 10, and maxIter more than 100.

Table 5. Kaggle Hadoop Results

	RMSE	R2
GBT	5.637465135	0.755592675
Linear	6.052575329	0.718274028
Decision	6.284134326	0.696305170

Table 6. Fare and Trip Hadoop Results

	RMSE	R2
GBT	7.841423948	0.392181010
Linear	7.905678795	0.382178916
Decision	8.046416658	0.359986065

As we can see above, Hadoop returns the same results as Databricks. Boosted decision tree regression returns the best results for all RMSE and R2 by using both Kaggle and Fare-Trip datasets. Hadoop using almost the same time to run the predictions, 3 minutes for Kaggle and 4 minutes for Fare-Trip. However, the Fare-Trip dataset have a better R2 in Hadoop compared to Databricks.

4. Background Work

We have used these two datasets to analyze and predict the fare amount based on some factors. Based on those factors, we have used three different algorithms for predictions. We built the model based on Kaggle dataset, and using Fare and Trip dataset as an additional for approvement.

5. Conclusion

Based on selected features predictive analytics was conducted to predict the fare of New York taxis.

Comparison between models in Azure ML, databricks and Spark ML is done and it is found that Gradient Boosted Tree Regression returns the best prediction. Compared to Databricks and Hadoop, Azure ML uses less time to run the prediction which is around 1 minutes. However, if increasing the dataset size. The storage space of Azure ML will be a limitation.

6. Limitations & Challenges Faced

- Feature prediction was much easier in AzureML than that of SparkML and Databricks.
- Boosted decision tree taking much time to run.

7. GitHub URL

https://github.com/liujh215/CIS5560.git

References

- [1] Kong, Allen. “NYC Taxi Fare Prediction,” *towards data science*, Dec 6, 2020, https://towardsdatascience.com/nyc-taxi-fare-prediction-605159aa9c24
- [2] Wu, Di. “New York City Taxi Fare Prediction,” *DIWUTECH*, https://diwu.tech/notes/2018/12/13/taxi
- [3] Synnott, Thomas. “Predicting The Taxi Fare of Chicago Cabs,” *Linkedin*, May 16, 2020, https://www.linkedin.com/pulse/predicting-taxi-fare-chicago-cabs-thomas-synnott