CS3244 Project: Taxi Fare Prediction

Team 14
Sidhant Bansal (A0178042H)
Aadit Kamat (A0164761B)
Aadyaa Maddi (A0161468Y)
Mukesh Gadupudi (A0161426L)
Rahul Baid (A0176876H)
Ronak Lakhotia (A0161401Y)

Abstract

This project aims to analyze previous taxi trips to predict the taxi fare for trips in the future. To get the most accurate results, we tailored the dataset to our needs and experimented with different models. We found that we got the best results, i.e. an RMSE of 2.87, using the Random Forest Regression model. By further analysis of the dataset, we also gained some insights into the transport patterns in cities.

Introduction

Singapore has an affordable yet world-class public transport network of trains, buses and taxis. Although trains and buses are cheaper alternatives, taxis are faster and widely available. Despite being one of the most expensive countries to live in, Singapore has the twentieth cheapest taxi fares in the world (2017). Additionally, high taxes and fuel prices on private cars make owning a car quite expensive for a majority of the population. Due to these reasons, taxis are a popular mode of public transport in Singapore.

By analyzing previous taxi trips to predict the taxi fare for trips in the future, our application aims to improve the quality of life of Singaporeans by helping them to plan their time and budget accordingly. This will help passengers decide the optimal time to start their commute. It will also help taxi drivers by allowing them to pick the most profitable option for their next trip.

Project Goals

1. What is the taxi fare for a trip based on the locations of the pickup and drop-off points, passenger count, time of pickup, and so on?
2. Out of the given factors, which ones influence the taxi fare the most?

Stretch Goals

1. Which type of machine learning model is the most suitable one for our problem and why?
2. What are the insights gained from analyzing the results and the features involved?

In this report, we first highlight the works we referenced to get some background information on the topic. We then describe the dataset and input features, along with the methods we used to process the dataset before it could be used by our model. Next, we explain the techniques used to train our model, our testing setup and our results. This is followed by a discussion on some insights gained into the taxi fare trends, as well as future improvements that can be made to our model. Finally, we conclude the report by describing the potential impact of our application.

Related Work

Similar work has been done to predict taxi fares for rides in New York using the New York taxi fare prediction dataset (2018). Due to a lack of sufficient data on taxi fares in Singapore, we decided to use the New York dataset to train our model. Even though New York and Singapore are situated in geographically distant regions, the cost of living and observed consumer patterns are quite similar in both the cities. Hence, our project makes an assumption that the variation in the geographical locations will not introduce significant differences while using our model to predict taxi fares in Singapore.

Ramachandran (2018) approached this problem with a hypothesis specific to New York and added input features based on prior knowledge of the city. For example, boolean features that indicated whether trips were originating from certain localities or airports were added to the dataset. However, we decided to eliminate input features specific to New York since we need our model to be able to generalize to Singapore in the future. We analyzed the remaining input features and made improvements to the dataset. For instance, we made use of the one-hot encoding mechanism to make our model more accurate. We analyzed the dataset and obtained results with greater rigour to find useful insights about the taxi fare trends.

Dataset

As mentioned in the previous section, we used the New York City taxi fare prediction dataset to train our model. This dataset has input instances corresponding to 55 million taxi trips in New York between 2009 and 2015. Each input instance has the following features:

Feature Name	Feature Description
pickup_datetime	Timestamp of when the taxi ride began
pickup_longitude	The longitude coordinate of where the taxi ride began
pickup_latitude	The latitude coordinate of where the taxi ride began
dropoff_longitude	The longitude coordinate of where the taxi ride ended
dropoff_latitude	The latitude coordinate of where the taxi ride ended
passenger_count	The number of passengers in the taxi ride

Each of the input instances has a corresponding *fare_amount*, which is the value we want our model to predict.

Data Cleaning and Feature Generation

We used **one million rows** from the dataset to train our model.

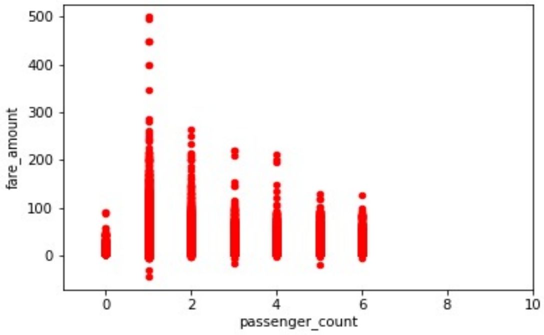


Figure 1: *Distribution of passenger_count in the dataset.*

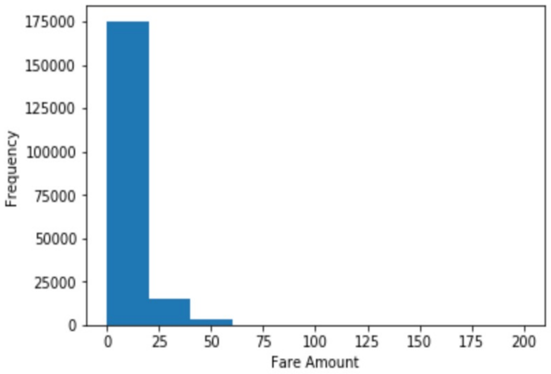


Figure 2: *Distribution of fare_amount in the dataset.*

After analyzing the distributions of the input features, we removed the following input instances to clean the dataset:

1. Input instances whose fare amount was out of the range (0, 50].
2. Input instances whose latitudes or longitudes were out of range (outside New York).
3. Input instances whose passenger count was 0 or negative.

Next, we split the *pickup_datetime* feature into *pickup_hour*, *pickup_day*, *pickup_day_of_week*, *pickup_month*, and *pickup_year*. The intuition behind this was that taxi fares will vary over the day (e.g. peak hours vs midnight), and over the week (e.g. weekdays vs weekends).

One-Hot Encoding Consider the *pickup_day_of_week* feature, which previously assigned a numerical value between 0 to 6 to each day of the week. To ensure that the algorithm does not rank these numerical values, we used **one-hot encoding** to split the feature into seven columns. For example, if the day of the week is a Sunday, its column will have a value of 1 while all other columns will have a value of 0.

Another possible one-hot encoding that we could have done is to divide the entire city into grids. Each input instance would have been assigned two grids, one for the pickup location and the other one for the drop-off location. This would have allowed our model to detect trends specific to localities. For example, some localities might have important landmarks or facilities like airports, or have a more congested road network. However, this would have increased the number of input features drastically.

This would have resulted in a more complex model increasing the likelihood of it overfitting the data. Keeping this in mind, we decided not to apply one-hot encoding to the other features that take a large range of discrete values (such as *pickup_hour*, *pickup_day*, dividing the city into grids, etc).

Methodology

The following sections describe our experimental setup and the machine learning techniques we used to train our model on the dataset.

Experimental Setup

- We split the dataset into 80% training set and 20% testing set. We found that this split was optimal after manually experimenting with multiple values.
- We used two types of error analysis methods to evaluate the performance of our model. Our first metric was **Root Mean Square Error (RMSE)**. RMSE represents the sample standard deviation of the differences between the predicted target values and the actual target values. It can be represented using the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(A_t - F_t)^2} \qquad (1)$$

Our second metric was **Mean Absolute Percent Error (MAPE)**. It can be calculated as follows:

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \qquad (2)$$

where A_t is the actual target value, F_t is the predicted target value and n is the total number of input instances. MAPE is used to assess the accuracy of our predicting model. In real world applications, MAPE is only used when F_t (in our case, taxi fare) is known to remain *greater than zero*. If the target values are very small (usually less than one), MAPE yields extremely large percentage errors (outliers).

- We used `pandas` to process our dataset, and `numpy` and `scikit-learn` to train our model. We ran our code in a Jupyter notebook.
- We used the SoC Compute Cluster (Tembusu Cluster) to train our model. Performing the computationally intensive job on the cluster decreased the training time drastically.

Baseline Model

To establish a baseline, we used the mean of all fare amounts in the training set as the predicted target value. This gave us an RMSE of 7.69 and a MAPE of 60.47%.

Linear Regression

Since *fare_amount* is continuous in nature, we selected a standard linear regression model to test against the baseline model. Without additional feature engineering, this model lowered the RMSE to 6.64 and MAPE to 54.64%.

Feature Engineering We calculated the trip distance between the pickup and drop-off points using the **Haversine Formula**. After plotting the relationship between the trip distance and the corresponding *fare_amount*, we observed that there is a linear correlation between the two. This is an expected result since taxis generally charge passengers by distance travelled.

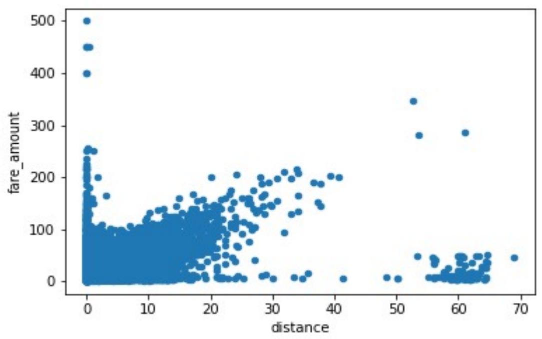


Figure 3: *Linear relationship between trip distance and fare amount.*

We considered trip distance to be an important feature to add to our dataset since it is not a linear combination of the latitude and longitude coordinates. After adding this feature, we observed an improvement in both RMSE (6.64 to 4.24) and MAPE (54.64% to 25.14%) as expected.

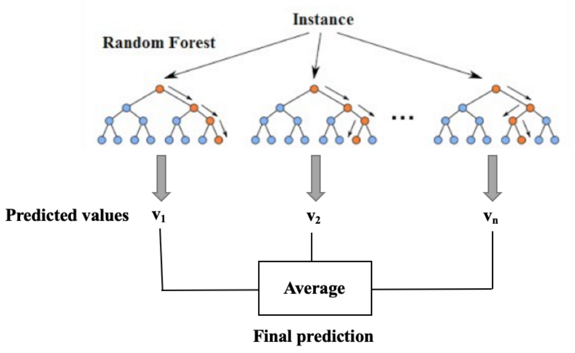


Figure 4: *How the Random Forest Regression technique works.*

Random Forest Regression

We then used the random forest regression technique to train our model. Random forest regression is an ensemble learning technique that uses **bagging**, where each decision tree in the forest is trained on the dataset using a randomly selected (with replacement) subset of features. Each tree uses **Mean Squared Error (MSE)**, or variance, for selecting the next feature for a split. Predictions are then made by applying the averaging principle to the results obtained from the individual decision trees. This is an application of the probabilistic concept that the differences in the results from uncorrelated decision trees (trained on different parts of the dataset) cancel out to zero when combined.

We experimented with the number of decision trees to reduce the variance observed in the individual trees. Figure 5 shows that there is a decrease in the variance with respect to the number of trees. However, this decrease in variance tapers off after a certain threshold (20 in this case). Adding any more trees would just increase the complexity of the model.

The random forest regression model gave us an RMSE of 2.87 and a MAPE of 18.10%.

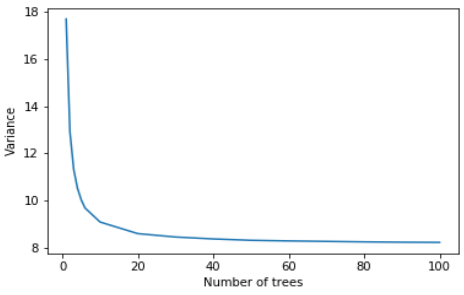


Figure 5: *Relationship between variance and number of trees used in the Random Forest Regression model.*

Results

In addition to using RMSE and MAPE to compare the performance of our models, we used best fit lines to compare the predicted and actual outputs for each model. The result is shown in Figure 6 below.

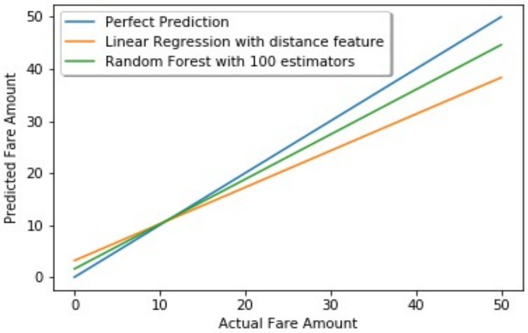


Figure 6: *Performance of our models against a perfect prediction i.e. a best fit line over all the input instances.*

We can see that the random forest regression model most closely follows the blue line and hence, is the most accurate. Additionally, we can see that as the fare amounts increase, the predictions made by our model become inaccurate.

Intersection Point of Models

The intersection point of the lines indicates that our model is accurate when predicting fare amounts of relatively shorter trips. This can be because our dataset does not take traffic as a variable. It is intuitive that longer trips will have larger fare amounts. This also shows that our application would perform better if we had a higher dimension of input variables.

Discussion

Choosing Random Forest Regression

Random forest regression makes use of decision trees, which are robust to noise. Given that our dataset is susceptible to noise, the use of this model is justified (Breiman 2001). Apart from cleaning the data, there is minimal preprocessing of data required for using this technique. In addition, Random forest regression overcomes the performance issues exhibited by the individual trees. It reduces the variance using techniques such as bagging and averaging the results from all of the trees. The model is also able to rank features by importance. This enabled us to make predictions about the target outputs and helped us see which features were the most correlated with our data. Our application is able to leverage upon all these properties of Random Forest Regression to effectively analyze the dataset.

Feature Importance

The importance of a feature is computed by measuring how effective the feature is in reducing the variance when creating decision trees within the random forest.

Figure 7 shows that important features like trip distance and pickup and drop-off points have the most effect on the prediction model. It is also important to note that due to one-hot encoding, it is hard to get a sense of how much categorical variables (e.g. *pickup_day_of_week*) affect the prediction model.

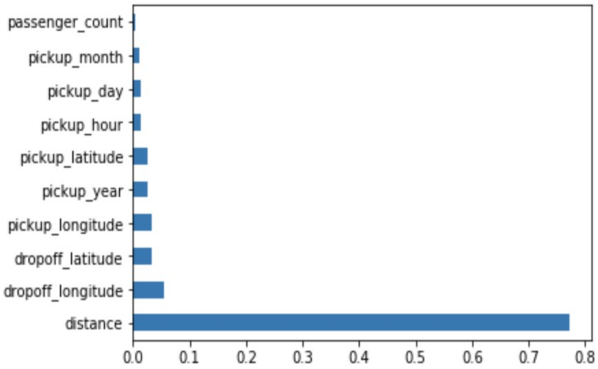


Figure 7: *Graph showing the importance of features in our model (top 10 shown).*

Feature Engineering

Encoding Cyclical Continuous Features Ideally, we want our model to see that 23:55 and 00:05 are 10 minutes apart. But if time is considered as a numerical value, then the time instances 23:55 and 00:05 will appear to be 23 hours and 50 minutes apart. Therefore, we created two new features (London 2016) by applying a sine and cosine transformation on the hour and minute features as follows:

$$sin\_hm = \sin\left(\frac{2\pi(hour * 60 + minutes)}{1440}\right) \tag{3}$$

$$cos\_hm = \cos\left(\frac{2\pi(hour * 60 + minutes)}{1440}\right) \tag{4}$$

From Figure 8, we can observe that the distance between any two points corresponds to the difference in time as we expect from a 24-hour cycle.

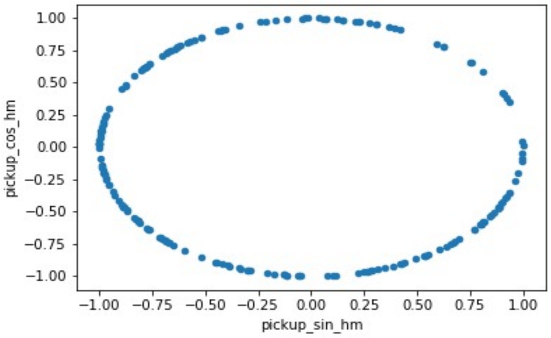


Figure 8: *Uniform distribution of (sine, cosine) coordinates taken from a sample of 200 input instances from our dataset.*

However, we did not observe a significant improvement in our model on using this feature transformation.

Insights

One of our stretch goals for this project was to gain some insights into the transport patterns of the city.

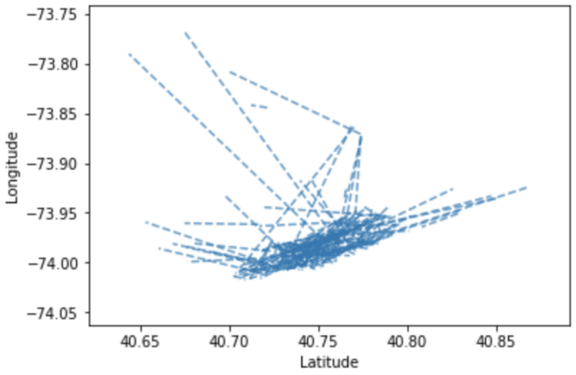


Figure 9: *Taxi trips in New York tend to be latitudinal, or horizontal in nature (sampled across 200 input instances).*

The line segments in Figure 9 denote the start and end points of 200 trips from the dataset. It can be inferred that there was a higher frequency of taxi rides in the latitudinal (horizontal) direction in the dataset. This could be because of either one of the following hypotheses:

1. The city is wider in terms of longitude as compared to latitude i.e. its width is larger than its height.
2. Other forms of public transport in the city give passengers a cheaper option to travel in the vertical direction than the horizontal one. This means a lesser number of passengers opt for taxis in the longitudinal, or vertical direction.

As shown in Figure 10, the structure of the subway system in New York is longitudinal in nature. We can hence conclude that our second hypothesis was more likely to be the reason why there were more taxi rides in the latitudinal, or horizontal direction.

Improvements

1. A possible improvement to our model could be using *exponential smoothing* (Brownlee 2018) to weight recent input instances higher than the older ones. Giving more importance to recent data would make our model perform correctly even in the presence of constantly changing factors that affect the taxi fare e.g. fuel prices or toll charges tend to change over time.
2. Another possible improvement that we could be made to our model is to train it on data specific to Singapore. We could have done this by collecting data in Singapore ourselves.
3. We could have also added more input features to the dataset based on the weather and availability of taxis at the pickup location. For example, it is common for taxi fares to increase during rainy conditions. Having a knowledge of the weather will help our model make predictions more accurately. Another factor that affects taxi fares is the supply-demand proportion at a given location. Taking this ratio into account will help the model account for the effects of surge pricing as time progresses.
4. Additionally, we could have explored deep learning methods for structured data (Ruizendaal 2018) to train our model.

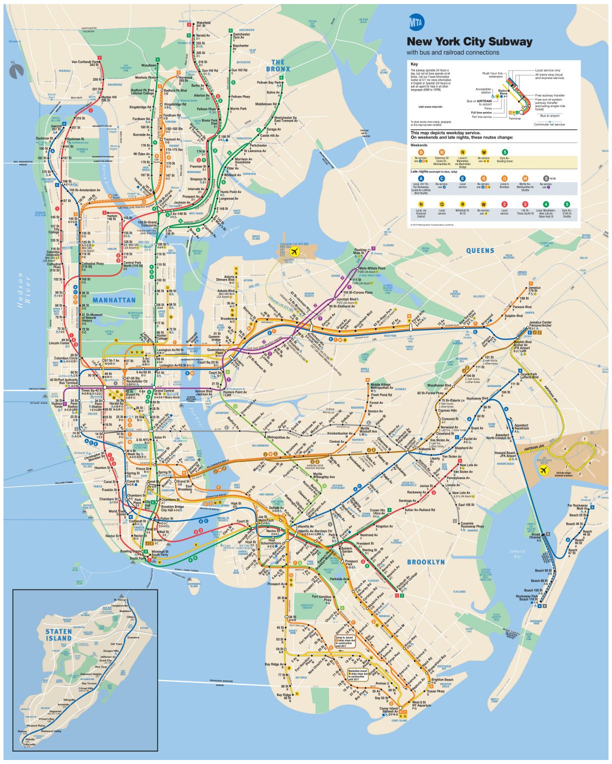


Figure 10: *Map showing New York Subway lines.*

Impact

Although our model is trained on a dataset of taxi rides in New York City, we have removed the region specific features so as to generalize the model. We do not expect any significant deviation in taxi fare patterns between the two urban centres. Hence our findings can be extrapolated to Singapore, after performing additional verification tests of our hypothesis.

How does our proposed model help Singapore?

1. Our model can be used by transport authorities such as LTA and ComfortDelGro to develop an application for commuters that can give them locale and time specific taxi fare information. Navigation apps such as Google Maps do provide a rough estimate of prices. However, this information is obtained using APIs provided by the individual taxi operators and is subject to the vagaries of their pricing strategies. Our model, however, does not rely on any domain-specific knowledge. This makes our results largely independent of such price fluctuations.
2. Our research concluded that taxi rides in New York City are more frequent in the horizontal direction. This was

confirmed by observing the NYC subway lines which provides commuters a cheaper option in the vertical direction. Such a result can be helpful for governments to better plan the transport facilities. In the case of Singapore it could mean deploying more MRT lines or starting bus services to regions that may not be well connected. It will also help identify regions in Singapore where the taxi fleet size is low.

Appendix

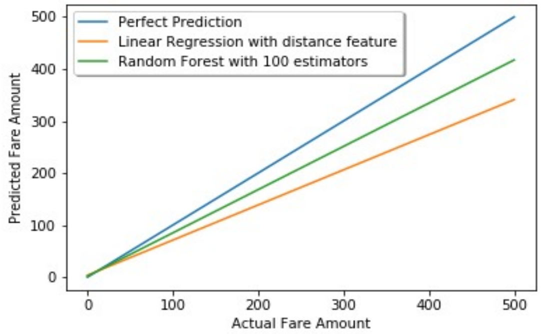
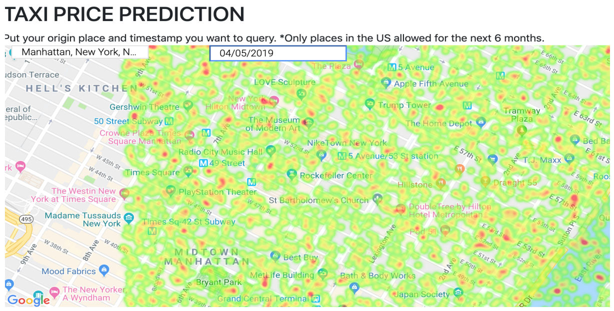


Figure 11: *Performance of our model without filtering fare_amount. Improvements after filtering can be seen in Figure 6.*



TAXI PRICE PREDICTION

Put your origin place and timestamp you want to query. *Only places in the US allowed for the next 6 months.

Manhattan, New York, N...

04/05/2019

Figure 12: *A prototype application that uses our model to populate a heatmap. Areas that are red correspond to higher predicted taxi fares.*

GitHub

https://github.com/cs3244-group-14/ml-singapore

Glossary

Haversine Formula This formula uses latitude and longitude coordinates of two points to calculate the shortest distance between the two points along the surface of a sphere.

Reflections

We learnt how to build an end-to-end machine learning application, which includes finding and processing the dataset, experimenting with and implementing different machine learning techniques to train our model, using standard measures of evaluating accuracy, and lastly, presenting our findings in an accessible manner.

Contributions

Sidhant Bansal (A0178042H): Implemented all the ML models using the described experimental setup. Plotted all the graphs and charts required in `pyplot`. Worked on the prototype application front-end. Project Report: Methodology section, Feature Engineering, One-Hot encoding.

Aadit Kamat (A0164761B): Wrote sections of the Project Report such as one-hot encoding, random forest regression, choice of model and impact of application. Hosted the front-end of the prototype application on `Heroku` using the `Flask` microservice.

Rahul Baid (A0176876H): Referred to research papers relating to regressional analysis to learn about different types of models used such as linear regression, random forest, XGBoost and Light GBM. Project report: introduction, goals and random forest model. Suggested the alternative error measure MAPE for a better analysis of results.

Ronak Lakhotia (A0161401Y): Extensively analyzed the working of the three learning models. Project Report: Results section, possible improvements to the proposed model, researched on potential Impact of model to Singapore. Consolidated the results of the three models used and inferred the outcomes from the stats shown and Feature Engineering.

Aadyaa Maddi (A0161468Y): Researched on problem statement. Cleaned and initially processed dataset using `pandas`. Wrote the Introduction, Related Work, Dataset, Insights, and Improvements sections in the project report, revised and formatted the final report.

Mukesh Gadupudi (A0161426L): Implemented various machine learning models using the described experimental setup.Analysed the working of the learning models. Contributed to the introduction, and improvements section in the project report.

References

Breiman, L. 2001. Random forests. *Machine Learning* 45(1):5–32.

Brownlee, J. 2018. Exponential smoothing for time series forecasting in python.

2017. Taxi fares in singapore are 20th cheapest in the world.

Kaggle. 2018. New york city taxi fare prediction.

London, I. 2016. Encoding cyclical continuous features - 24-hour time.

Ramachandran, A. 2018. Machine learning to predict taxi fare - part one : Exploratory analysis.

Ruizendaal, R. 2018. Using deep learning for structured data with entity embeddings.