Cab Fare Prediction

By, Sandeep A C

INTRODUCTION

Now a day's cab rental services are expanding with the multiplier rate. The use of Data Science can help the enterprise to provide their customers a better service than before.

PROBLEM STATEMENT

You are a cab rental start-up company. You have successfully run the pilot project and now want to launch your cab service across the country. You have collected the historical data from your pilot project and now have a requirement to apply analytics for fare prediction. You need to design a system that predicts the fare amount for a cab ride in the city.

DATA

Number of attributes: 6

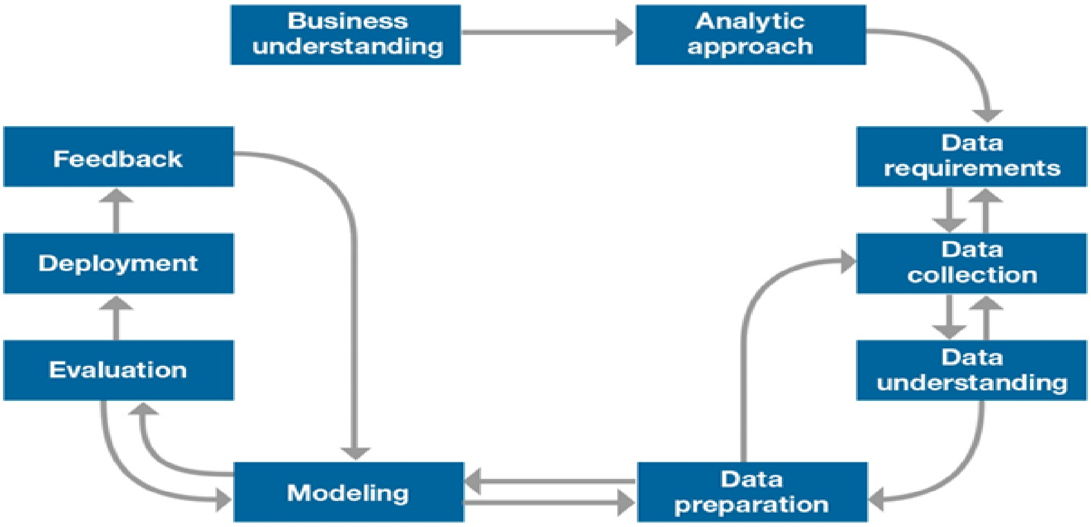
- pickup_datetime - timestamp value indicating when the cab ride started.
- pickup_longitude - float for longitude coordinate of where the cab ride started.
- pickup_latitude - float for latitude coordinate of where the cab ride started.
- dropoff_longitude - float for longitude coordinate of where the cab ride ended.
- dropoff_latitude - float for latitude coordinate of where the cab ride ended.
- passenger_count - an integer indicating the number of passengers in the cab ride.

Missing Values: Yes

SYSTEM REQUIREMENTS

Hardware Requirements: Micro Processor: Intel processor Main memory/RAM: 4GB Hard Disk: 40 GB HHD/SSD
Software Requirements: Operating system: windows, mac, Linux Framework: Anaconda, Jupyter Notebook Programming Language: Python 3

METHODOLOGY



DESIGN AND IMPLEMENTATION

1. Collect the Data

Typical information-gathering methods include:

- Face-to-face or telephone interviews
- Surveys
- General research using published information about market categories
- Focus groups

In our case the dataset is already provided, which has 16067 rows and 7 columns.

2. Data Pre-Processing

It includes many phases. Considering Python code as base for explanation,

i. Missing Value Analysis

Missing data can occur because of nonresponse, no information is provided for one or more items or for a whole unit. Sometimes missing values are caused by the researcher. For example, when the data collection is done improperly or mistakes are made in data entry, it is called Human Error.

There are different ways to deal with missing values in the data.

- **Imputation**

We can impute the missing values with its mean, median or mode.

There is also another method called KNN imputation, which is available in both R and Python.

- **Partial deletion**

When the missing values in the data are less, we can delete them since they do not affect the model accuracy as they are in less number.

In our case, we are doing partial deletion, since only 80 rows contain null, which is 0.49% of 16067 rows. By doing so, Number of rows reduced from 16067 to 15987.

ii. Outlier Analysis

An **Outlier** is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.

A simple way to way to detect outlier is to use boxplot.

For our data, the below images show the box plot, which are drawn in both R and Python.

If there is too much of data in outlier, then deleting them is not an efficient way, because we will lose lot of data.

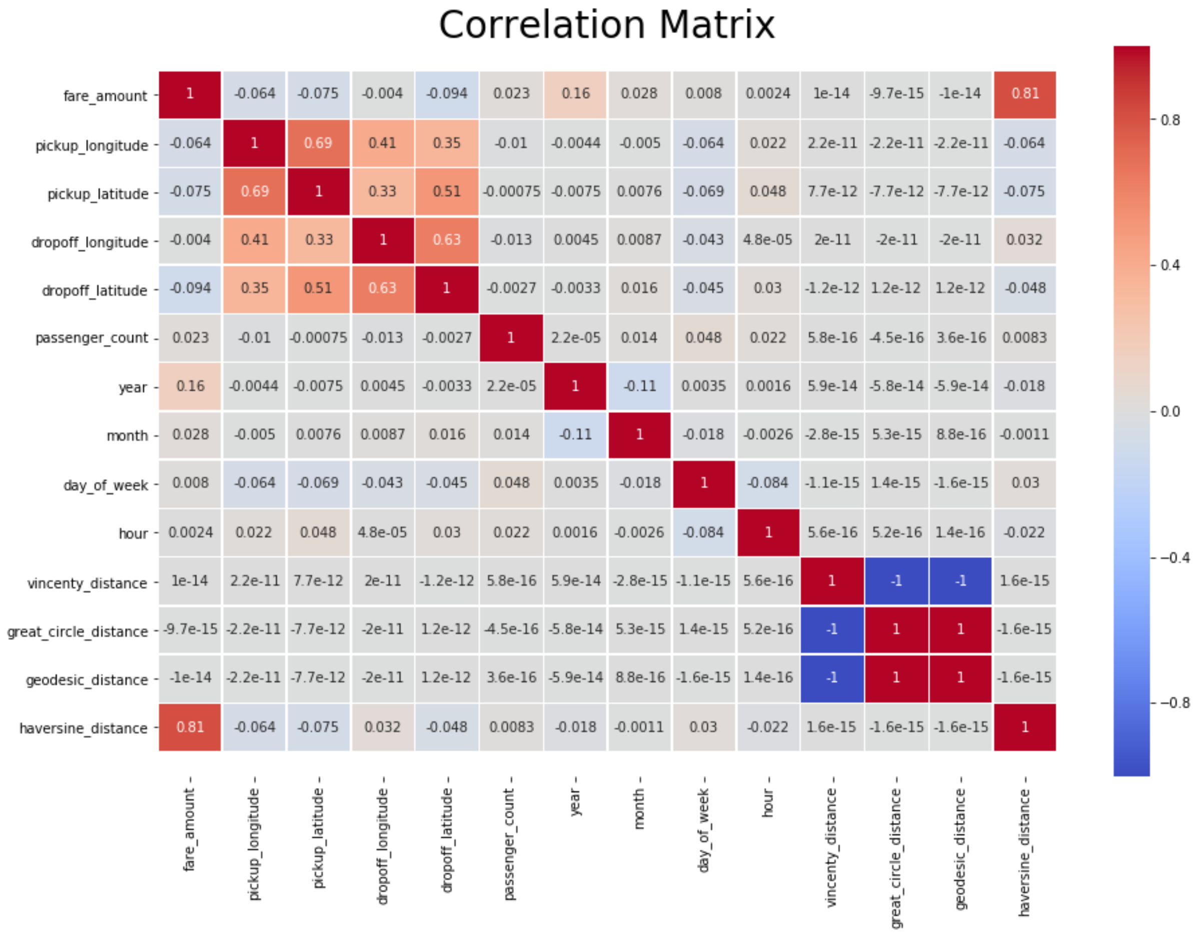
- Latitudes range from -90 to 90 and longitudes range from -180 to 180. In the latitude and longitudes columns, there might be outliers. Remove them by using the function *outOfRangeValues(<dataframe>)*
- If there are any values equal to zero, even they are considered as outliers. Remove them by using the function *dropRowsWithValuesEqualToZero(<dataframe>)*
- Remove all outliers from passenger_count variable using the function *cleanPassengerCount(<dataframe>)*

- Remove all outliers from fare_amount variable using the function *cleanFareAmount(<dataframe>)*
- Removing all other outliers from all other variables using the function *outliers_treatment(<dataframe>)*

By doing outlier analysis, we have reduced the number of rows to 13353.

iii. Feature Selection

For modelling, we have to select the columns or independent variables that will highly contribute to the dependent variable.
If there is no dependent variable, we have to find the collinearity between all the variables and if there is high collinearity between two variables, we can remove any one variable and keep the other.



	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count	year	month	day_of_week	hour	vincenty_distance	great_circle_distance	geodesic_distance	haversine_distance
fare_amount	1	-0.064	-0.075	-0.004	-0.094	0.023	0.16	0.028	0.008	0.0024	1e-14	-9.7e-15	-1e-14	0.81
pickup_longitude	-0.064	1	0.69	0.41	0.35	-0.01	-0.0044	-0.005	-0.064	0.022	2.2e-11	-2.2e-11	-2.2e-11	-0.064
pickup_latitude	-0.075	0.69	1	0.33	0.51	-0.00075	-0.0075	0.0076	-0.069	0.048	7.7e-12	-7.7e-12	-7.7e-12	-0.075
dropoff_longitude	-0.004	0.41	0.33	1	0.63	-0.013	0.0045	0.0087	-0.043	4.8e-05	2e-11	-2e-11	-2e-11	0.032
dropoff_latitude	-0.094	0.35	0.51	0.63	1	-0.0027	-0.0033	0.016	-0.045	0.03	-1.2e-12	1.2e-12	1.2e-12	-0.048
passenger_count	0.023	-0.01	-0.00075	-0.013	-0.0027	1	2.2e-05	0.014	0.048	0.022	5.8e-16	-4.5e-16	3.6e-16	0.0083
year	0.16	-0.0044	-0.0075	0.0045	-0.0033	2.2e-05	1	-0.11	0.0035	0.0016	5.9e-14	-5.8e-14	-5.9e-14	-0.018
month	0.028	-0.005	0.0076	0.0087	0.016	0.014	-0.11	1	-0.018	-0.0026	-2.8e-15	5.3e-15	8.8e-16	-0.0011
day_of_week	0.008	-0.064	-0.069	-0.043	-0.045	0.048	0.0035	-0.018	1	-0.084	-1.1e-15	1.4e-15	-1.6e-15	0.03
hour	0.0024	0.022	0.048	4.8e-05	0.03	0.022	0.0016	-0.0026	-0.084	1	5.6e-16	5.2e-16	1.4e-16	-0.022
vincenty_distance	1e-14	2.2e-11	7.7e-12	2e-11	-1.2e-12	5.8e-16	5.9e-14	-2.8e-15	-1.1e-15	5.6e-16	1	-1	-1	1.6e-15
great_circle_distance	-9.7e-15	-2.2e-11	-7.7e-12	-2e-11	1.2e-12	-4.5e-16	-5.8e-14	5.3e-15	1.4e-15	5.2e-16	-1	1	1	-1.6e-15
geodesic_distance	-1e-14	-2.2e-11	-7.7e-12	-2e-11	1.2e-12	3.6e-16	-5.9e-14	8.8e-16	-1.6e-15	1.4e-16	-1	1	1	-1.6e-15
haversine_distance	0.81	-0.064	-0.075	0.032	-0.048	0.0083	-0.018	-0.0011	0.03	-0.022	1.6e-15	-1.6e-15	-1.6e-15	1

Fig: Correlation Matrix for Variables

We are not removing any columns during feature selection.

iv. Feature Scaling

Feature Scaling is a technique to standardize the independent **features** present in the data in a fixed range. There are different methods used in feature scaling,

- Rescaling (min-max normalization)
- Mean Normalization
- Standardization (Z-score Normalization)
- Scaling to Unit etc.

Standardization has to be applied when the data is normally/uniformly distributed.

In our case, we are taking the log of variables.

3. Model Creation

Sometimes the data or the business objectives lend themselves to a specific algorithm or model. Other times the best approach is not so clear-cut. As you explore the data, run as many algorithms as you can; compare their outputs. Base your choice of the final model on the overall results. Sometimes you’re better off running an ensemble of models simultaneously on the data and choosing a final model by comparing their outputs.

Results

i. Linear Regression

```
Mean Absolute Percentage Error: 20.70335519304837
Which means model is (100 - 20.70) = 79.29664480695163% correct.
```

ii. Logistic Regression

Logistic regression can be applied only when there is binary output.

iii. Random Forest

```
Mean Absolute Percentage Error: 23.034737419577784
Which means model is (100 - 23.03) = 77.71% correct.
```

iv. Decision Tree

```
Mean Absolute Percentage Error: 28.03719176058775
Which means model is (100 - 28.03) = 71.96280823941225% correct.
```

v. SVR

```
Mean Absolute Percentage Error: 20.818938161417417
Which means model is (100 - 20.81) = 79.18106183858258% correct.
```

vi. K Neighbour Regressor

```
Mean Absolute Percentage Error: 26.352797895559853
Which means model is (100 - 26.35) = 73.64720210444014% correct.
```

vii. Gradient Boosting Regressor

```
Mean Absolute Percentage Error: 20.98421890922855
Which means model is (100 - 20.99) = 79.01578109077145% correct.
```