CAB FARE PREDICTION

Submitted by – Neeraj Mehra

JUNE 11, 2019

EDWISOR

Table of Contents

LIST OF FIGURES

INTRODUCTION

It is crucial for a start-up to have good insight of the business and any form of data regarding the business could serve the purpose. Here, the task is to start a cab-rental service across the country and a historical data has is collected for analysis. The aim of the project is to design a model that predicts the fare amount for a cab ride in the city. Such model could be of high significance since early prediction of amount for cab ride could be helpful in taking business decisions with more confidence.

DATA

The given dataset has fare_amount as dependent variable. The independent variables like pickup_dateTime contains useful information like hour and date of pickup. There is also variables containing pickup and dropoff longitude and latitude which tells the location of pickup and drop.

	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	4.5	2009-06-15 17:26:21 UTC	-73.844311	40.721319	-73.841610	40.712278	1.0
1	16.9	2010-01-05 16:52:16 UTC	-74.016048	40.711303	-73.979268	40.782004	1.0
2	5.7	2011-08-18 00:35:00 UTC	-73.982738	40.761270	-73.991242	40.750562	2.0
3	7.7	2012-04-21 04:30:42 UTC	-73.987130	40.733143	-73.991567	40.758092	1.0
4	5.3	2010-03-09 07:51:00 UTC	-73.968095	40.768008	-73.956655	40.783762	1.0

Figure 1 Train dataset snippet

Methodology

Pre-processing – Outlier Analysis

The raw data is usually incomplete, inconsistent and likely to contain many error therefore data pre-processing technique is used that involves transforming raw data into understandable format. The process involves data cleaning, checking for outliers, missing values and inconsistencies. In this data at fig 2.1 we have checked for outliers using box-plot

method. It is crucial to check for outliers since they can skew and mislead training process of machine learning algorithms. This may result in less accurate models and poor results.

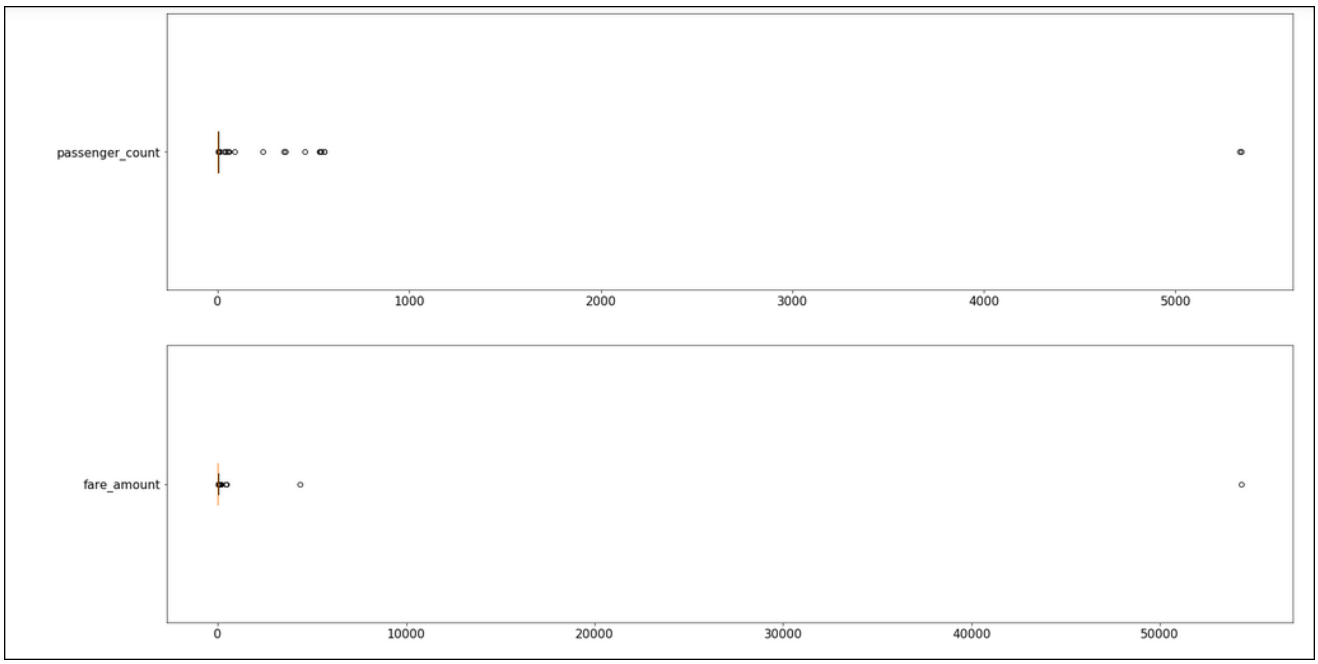

Figure 2 Box plot showing data distribution for passenger count and fare_amount

It was observed that variables like fare amount and passenger counts contains values that were abnormally high and could be a reason of wrong data input. Therefore such data is removed. Also, any value that is greater than is greater than longitude and latitude range (-180 to 180 and -90 to 90 respectively) is removed.

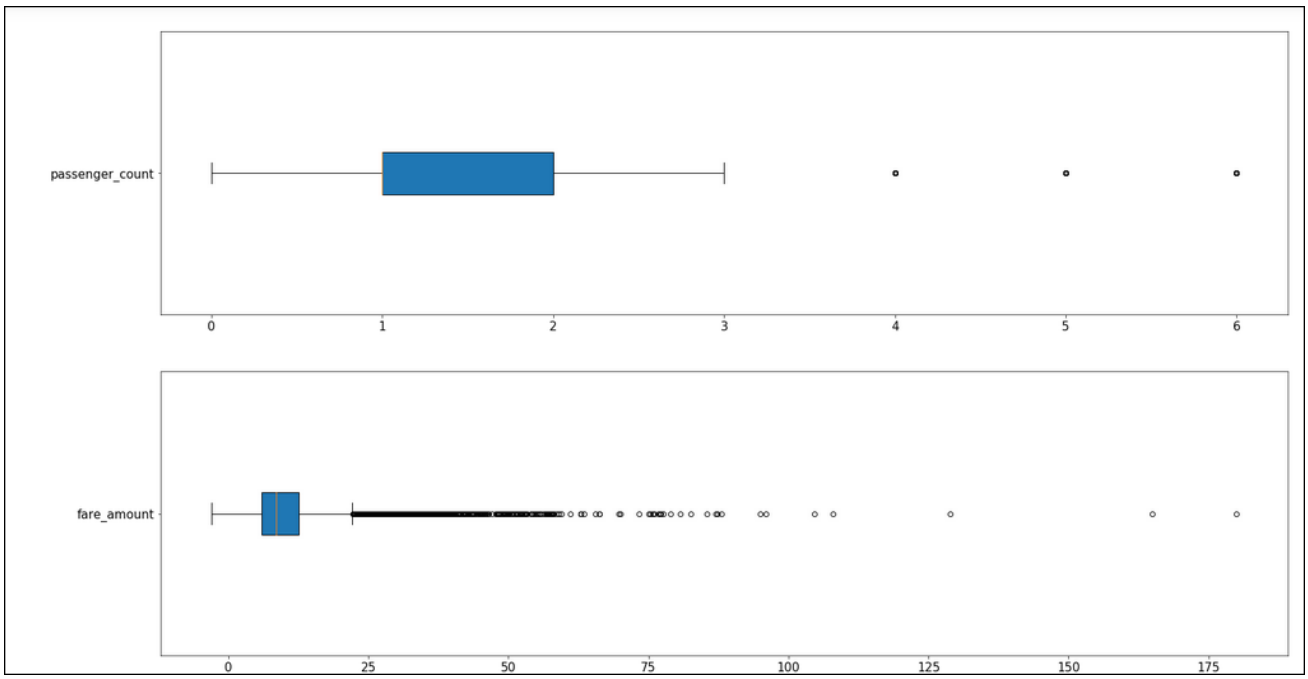

Figure 3 box-plots after removing outliers

Pre-processing – Probability Distribution

The probability distribution is plotted to check how continuous data is distributed over the range. It was observed that the distribution is not normal which was evident from outlier analysis. The above analysis indicates that imputing missing values with mean will not be a good measure.

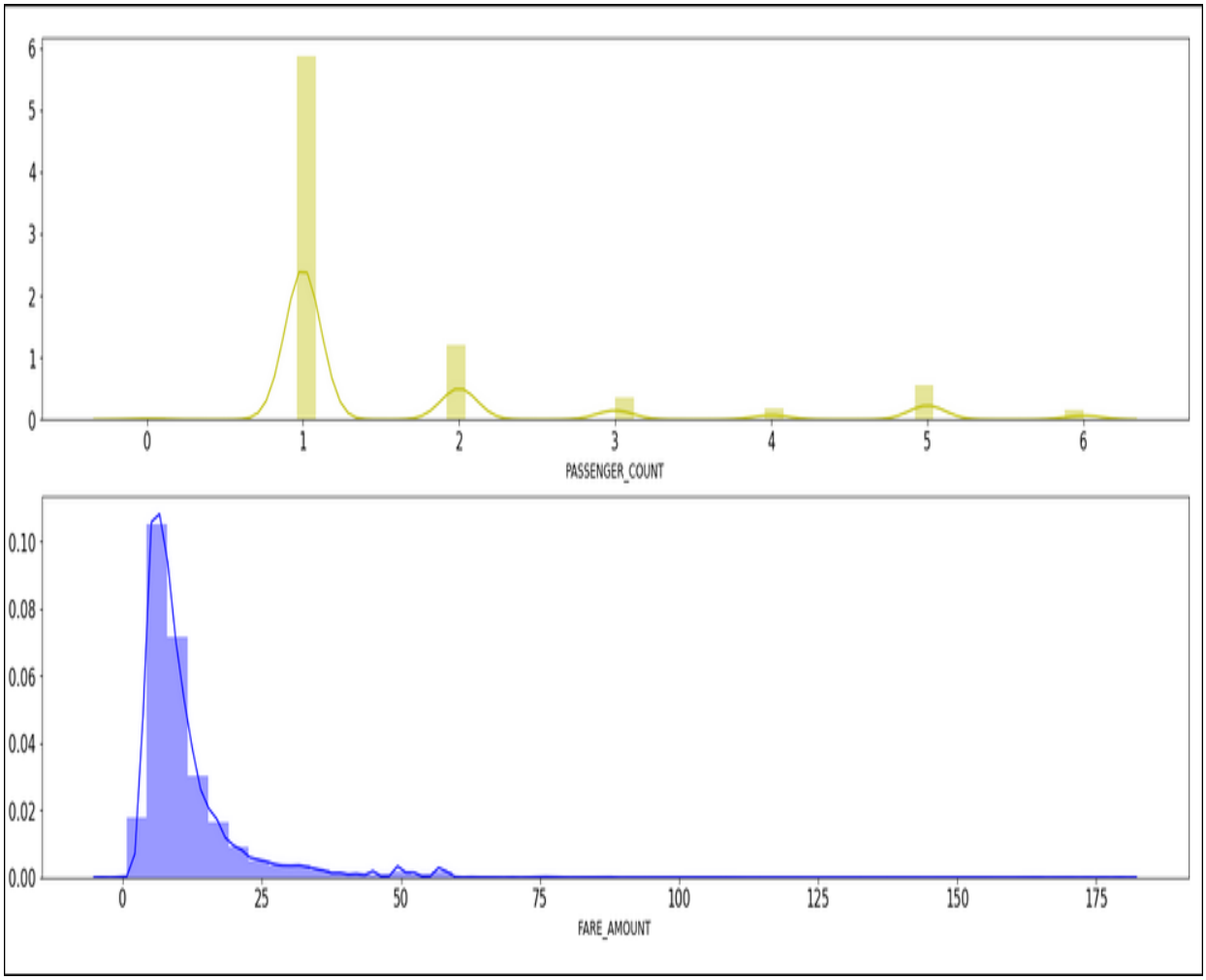

Figure 4 density-distribution of continuous variables

Pre-processing – Data Cleaning and Feature Engineering

The variable pickup_dateTime contained date in UTC format this word is stripped from the column. Later this column values were converted to datetime format and later important

information like hour, day of year, year, month and weekday are extracted. These extracted information were stored in separate columns.

pickup_weekday	pickup_hour	pickup_DayOfMonth	pickup_month	pickup_year
Monday	17.0	15.0	Jun	2009.0
Tuesday	16.0	5.0	Jan	2010.0
Thursday	0.0	18.0	Aug	2011.0
Saturday	4.0	21.0	Apr	2012.0
Tuesday	7.0	9.0	Mar	2010.0

Figure 5 Column extracted from pickup_dateTime column

The pickup and drop coordinate provided in the dataframe is further used to extract the distance covered by the cab between pickup and drop location. This distance data could be a crucial to predict the fare amount.

Haversine Formula is used to calculate the distance:

```
dlon = lon2 - lon1
dlat = lat2 - lat1
a = (sin(dlat/2))^2 + cos(lat1) * cos(lat2) * (sin(dlon/2))^2
c = 2 * atan2( sqrt(a), sqrt(1-a) )
d = R * c (where R is the radius of the Earth)
```

Figure 6 Haversine Formula refrence (https://andrew.hedges.name/experiments/haversine/)

The dataset had variables containing zeros as value were present which is not possible. This could be seen in the figure 7 below.

```
fare_amount              1
pickup_datetime          0
pickup_longitude       313
pickup_latitude        313
dropoff_longitude      312
dropoff_latitude       310
passenger_count         57
distance               457
dtype: int64
```

Figure 7 Table showing number of zero counts in each column of a dataframe

It is visible that passenger count, fare amount and location coordinates contain values zero which is not possible. Since counts of such values was large we could have lost lot of crucial data dropping them therefore it is concluded to replace them with NAN and later impute them with KNN.

Since most of the algorithms works on mathematical equations that uses Euclidean distances in its prediction. The classes in categorical variable if represented by values like (‘1’, ‘2’, ‘3’) then this might create problem since model will assume one class greater than other. Therefore, dummies are created for all categorical variables. However, some categorical feature include large number of classes for example week days and months .Therefore, top prevent from falling into dummy variable trap first dummy variable of each column is dropped.

	Morning	afternoon	evening	night	late_night	pickup_weekday_Friday	pickup_weekday_Saturday	pickup_weekday_Sunday	pickup_weekday_Thursday
0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0
3	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
4	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 8 dummy variable created

Also complete data is normalized to for quick convergence of models.

Pre-processing – Missing value Imputation

Since the distribution of data in each variable was not normal the KNN imputation method is used to impute all missing values.

Feature Selection

The correlation heat map is created for all numerical variable and highly correlated columns were dropped. This will prevent multi-collinearity since it reduces the precision estimate of coefficients and weakens the statistical power of regression model

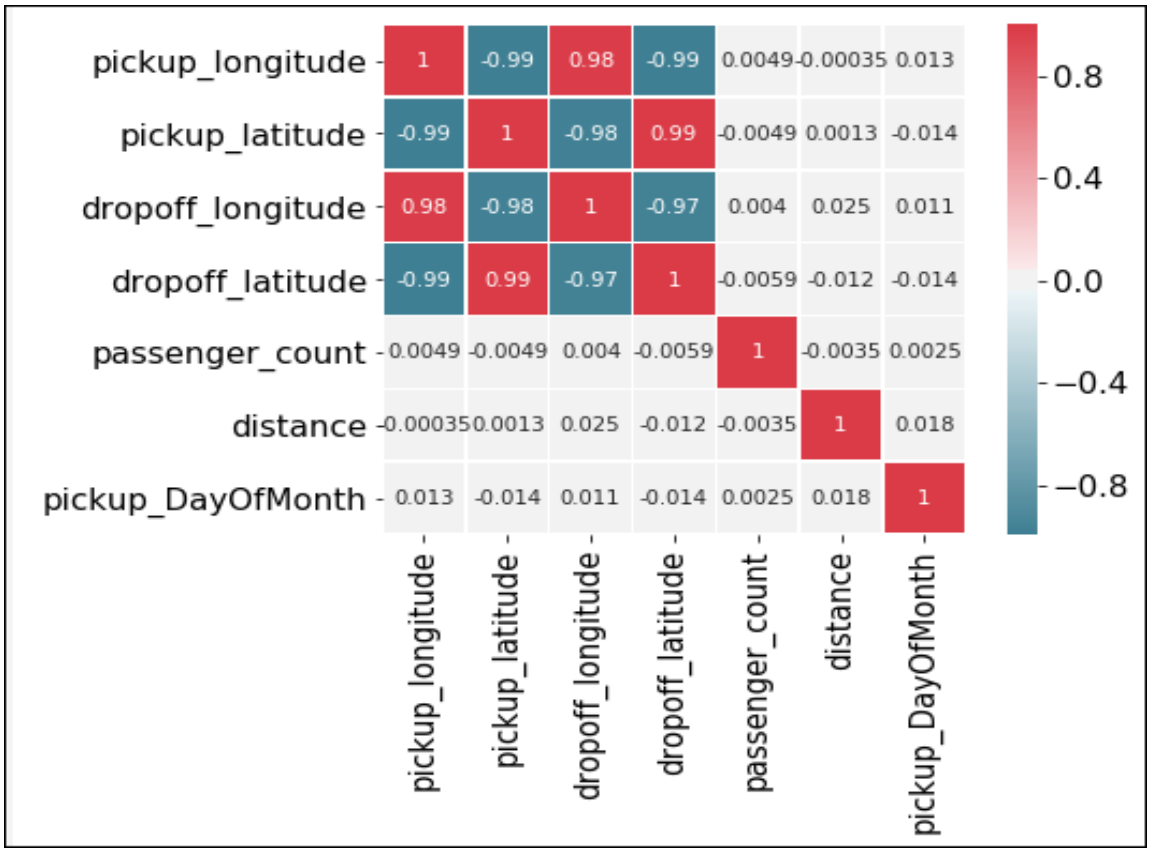

Figure 9 Correlation Heat Map

It can be observed from above figure that variables pickup_longitude, pickup_latitude, dropoff_latitude and dropoff_longitude are highly correlated and these variables are dropped since there collinearity is greater than 0.5.

For categorical variable we used chi-square test method and dropped all the variables that had chi-square test statistic greater than critical value (0.05).

Linear Relation between Dependent and Independent Variables

Before using the model to fit the dataframe it was decided to check linear relation between dependent and independent variable and also linear model regression line is used to see if it fits the model.

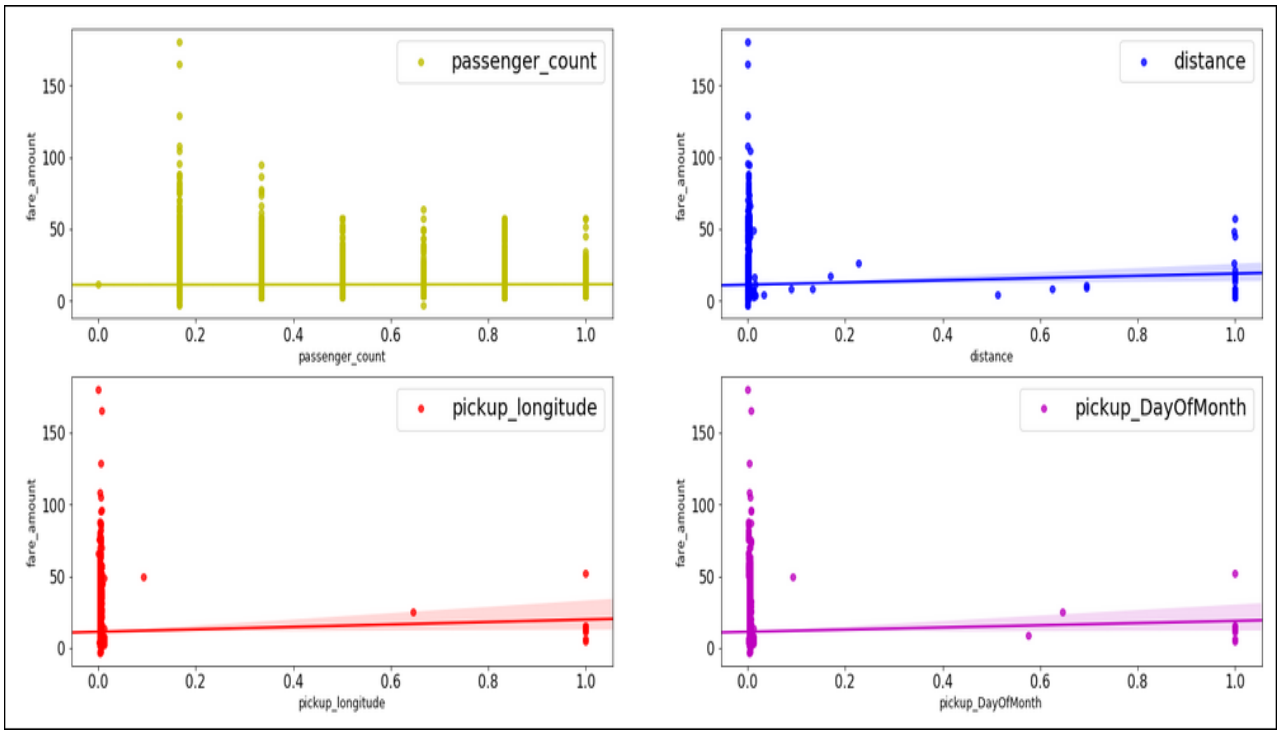

Figure 10 simple linear regression plot between dependent and independent variable

It was observed that variables are non-linearly distributed and regression line does not fits properly.

Modeling

Since our dependent variable is continuous it can be concluded that this is regression problem. Only regression models are used for prediction since reliable result can be expected. For evaluation error metrics like RMSE and R_squared error is used. RMSE is used basically measures average squared error of our predictions. For each point, it calculates square difference between the predictions and the target and then average those values and finally squared root is taken. The higher this value, the worse the model is. It is never negative, since we're squaring the individual prediction-wise errors before summing them, but would be zero for a perfect model.

The coefficient of determination, or R^2 (sometimes read as R-two), is another metric we may use to evaluate a model and it is closely related to MSE, but has the advantage of being scale-free — it doesn't matter if the output values are very large or very small, the R^2 is always going to be between $-\infty$ and 1.When R^2 is negative it means that the model is worse than predicting the mean. R^2 close to 1 means model is performing well. In conclusion, R^2 is the ratio between how good our model is vs how good is the naive mean model.

The first model is used was to multiple linear regression and backward elimination process used to get model with best fit. The second model used is Support vector Regression which can be used both for linear and non-linear regression. It makes sure that errors do not exceed the threshold. The third model we used Random Forest regression it uses combination of learning models to increase the accuracy. It uses bagging as a technique to average noisy and unbiased models to create a model with low variance.

Models	RMSE	R_squared_Error
Multiple linear Regression	9.5	0.02
Support Vector Regression	9.95	-0.074
Random Forest Regression	5.29	0.69

Since Random Forest Regression has the best result. Grid search is used to get best hyper-parameters for model to get best result. The below table shows hyper-parameter and R_square_error.

	Max_depth	Min_sample_leaf	n_estimators	R_squared_error
Random Forest Regression	20	10	70	0.74

Conclusion

Based on above result Random Forest Regression turns out be best model for this dataframe.

APPENDIX – 1 – Running Python File from dos prompt

1. Copy path to python file you want to run.
2. Open command prompt
3. Type cd and then with a space paste path to python file.
4. Press enter
5. Next type Python and with space type file name you want to run.

APPENDIX – 2 – Running

1. Copy path to R file you want to run.
2. Open command Prompt
3. Type cd and then with a space paste path to python file.
4. Press enter
5. Next type Rscript and then with space type file name you want to run.