



Cab Fare Prediction

Vijay Gonsalves
15/9/2019

Table of Contents

1. INTRODUCTION	3
1.1 Problem statement.....	3
1.2 Data	3
2. Methodology	4
2.1 Pre Processing.....	4
2.1.1 Exploratory data analysis.....	4
2.1.2 Outlier Analysis.....	4
2.1.3 Missing Value Analysis.....	6
2.1.4 Feature Selection.....	7
2.1.5 Feature Scaling.....	8
2.1.6 Multicollinearity check.....	8
2.2 Modeling	9
2.2.1 Model Selection	9
2.2.2 C50 (Decision Tree).....	9
2.2.3 Random Forest.....	10
2.2.4 Linear Regression.....	12
3. Conclusion	133
3.1 Model Evaluation.....	13
3.1.1 Mean Absolute Percentage Error (MAPE).....	13
3.2 Model selection.....	14
Appendix A--R Code.....	15
Appendix B--Python code.....	15
Appendix C--Source data.....	15

1. INTRODUCTION

1.1 Problem statement

A cab is a taxi on hire. It picks up people from one point and drops them to another point typically within the same city but sometimes in adjoining cities also. People hire a cab and at the end of the journey fare is calculated based on distance covered. Time taken from picking up to dropping off is also used to charge the customers accordingly.

Typical customers of a cab are tourist wanting to explore the city on weekends and daily office workers who use it to travel to their office and home.

The objective of this Case is to Predication of Cab fare on daily based on historical pilot project data.

1.2 Data

Our task is predict the cab fare depending on date and pickup and drop-off locations.

Since we have to predict the fare i.e. a number this becomes a regression problem.

Given below is a sample of the data set that we are using to predict the fare of cab:

Table 1.1: Sample Data (Columns: 1-8)

fare amount	pickup_datetime	pickup longitude	pickup latitude	dropoff_longitude	dropoff_latitude	passenger count
4.5	2009-06-15 17:26:21 UTC	- 73.844311	40.721319	-73.84161	40.712278	1
16.9	2010-01-05 16:52:16 UTC	- 74.016048	40.711303	-73.979268	40.782004	1
5.7	2011-08-18 00:35:00 UTC	- 73.982738	40.76127	-73.991242	40.750562	2
7.7	2012-04-21 04:30:42 UTC	-73.98713	40.733143	-73.991567	40.758092	1
5.3	2010-03-09 07:51:00 UTC	- 73.968095	40.768008	-73.956655	40.783762	1
12.1	2011-01-06 09:50:45 UTC	- 74.000964	40.73163	-73.972892	40.758233	1
7.5	2012-11-20 20:35:00 UTC	- 73.980002	40.751662	-73.973802	40.764842	1
16.5	2012-01-04 17:22:00 UTC	-73.9513	40.774138	-73.990095	40.751048	1

Variables present in given dataset are Fare Amount, Pickup date time, Pickup longitude and latitude, drop off longitude and latitude and passenger count.

The details of variable present in the dataset are as follows:

- 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

2. Methodology

2.1 Pre Processing

Any predictive modeling requires that we look at the data before we start modeling. However, in data mining terms looking at data refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis.

2.1.1 Exploratory Data Analysis

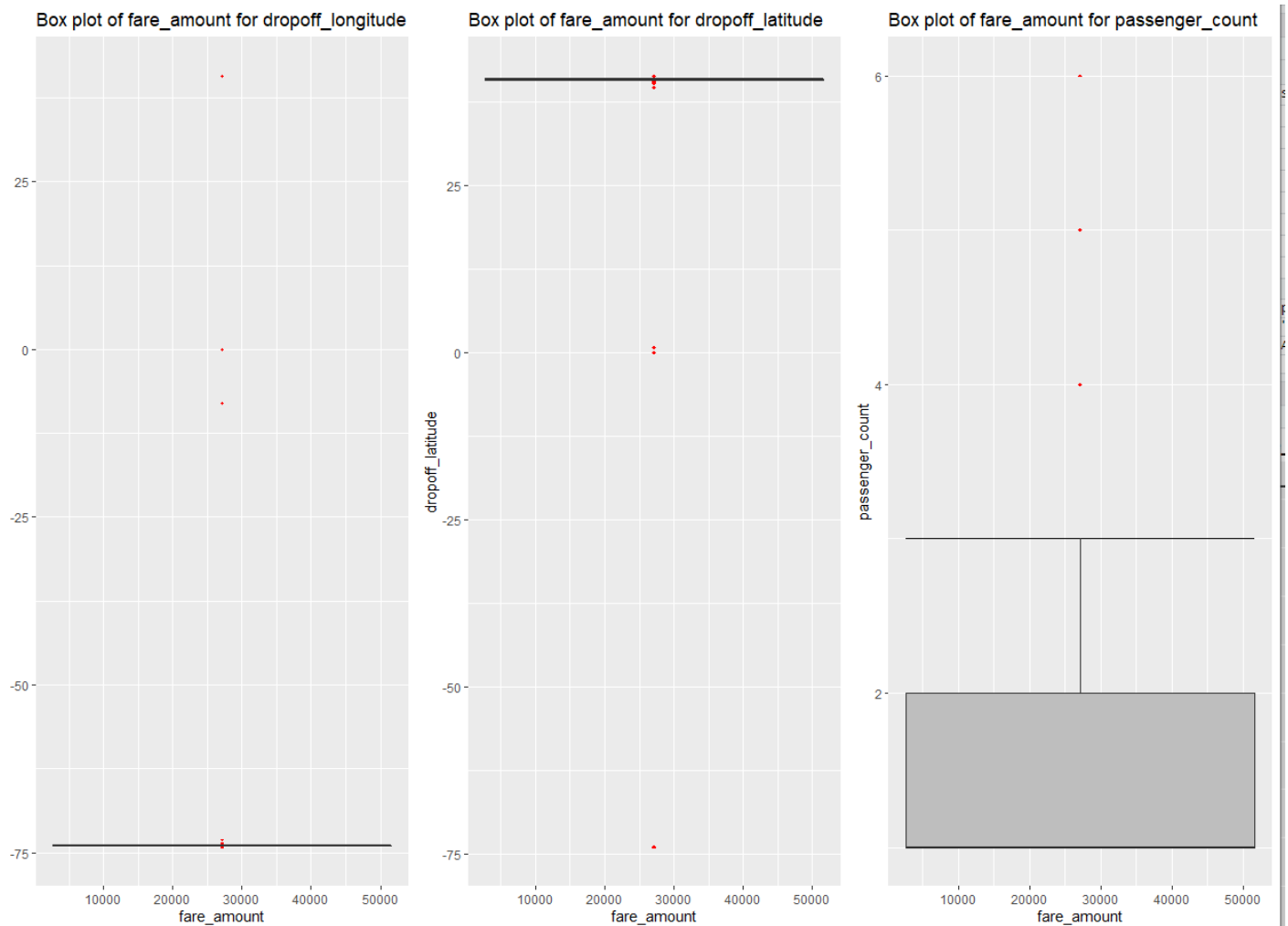
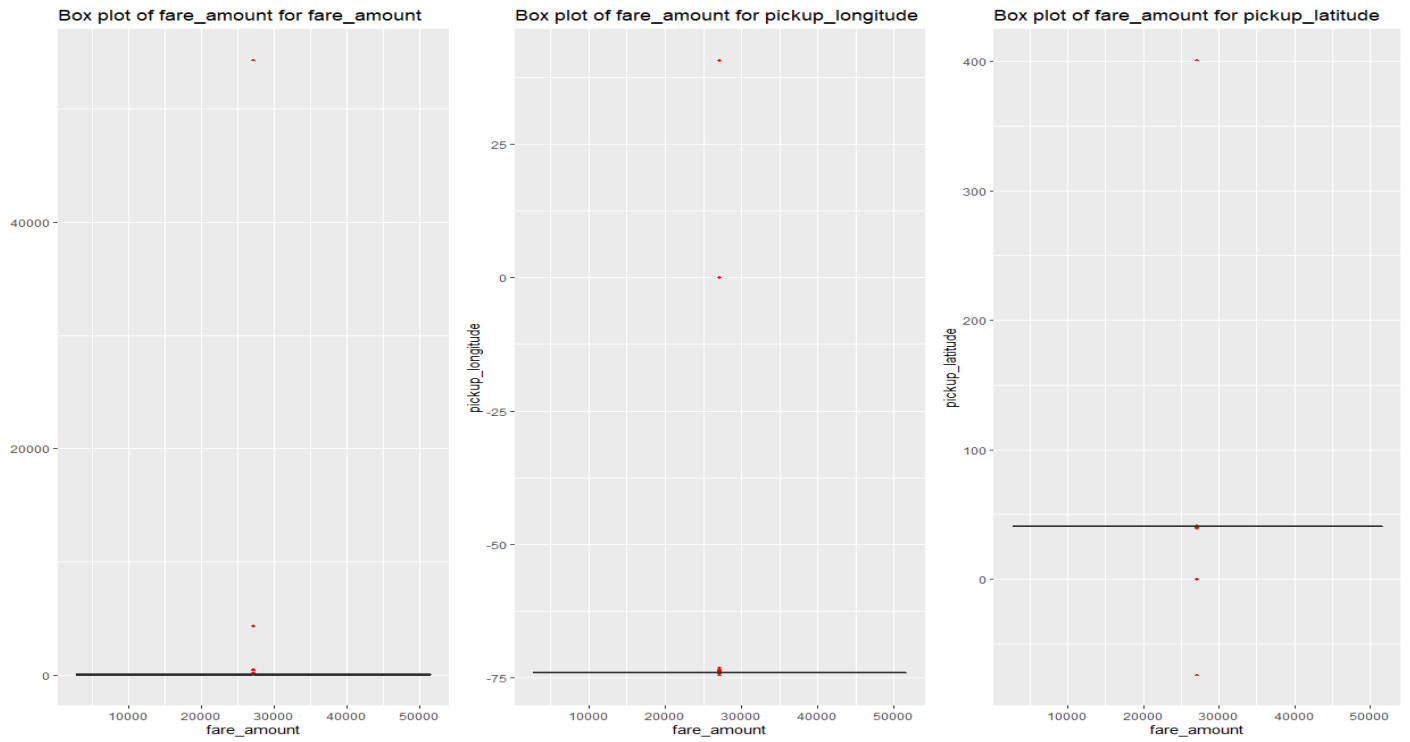
In exploring the data we have

- Date time variable has 1 value as 43 which is obviously wrong entry and hence deleting it and breaking pickup_datetime variable into month, year, day, dayofweek, hour and timeofday to better understand the data and then dropped pickup_datetime variable.
- Variable conversion to bring all variables in proper factor or continuous form based on their values like fare_amount should be numeric and cols derived from date time should be factor
- fare amount has -ve values which is not logically possible and hence making them as NA
- Pickup and drop-off location can't be the same so deleting such rows
- removing rows with passenger count <1 and >6 as logically not possible
- We have one value with passenger count as 1.3 which is obviously wrong and hence we round it to nearest value
- Check if Latitudes range from -90 to 90 and Longitudes range from -180 to 180. Found OK except for pickup latitude but it will be handled in outlier analysis

2.1.2 Outlier Analysis

Outlier analysis is done to handle all inconsistent observations i.e. rows having value very far away from the mean and which distorts the mean present in given dataset. As outlier analysis can only be done on continuous variable.

Below figure is visualization of numeric variable present in our dataset to detect outliers using boxplot. Outliers will be detected with red color



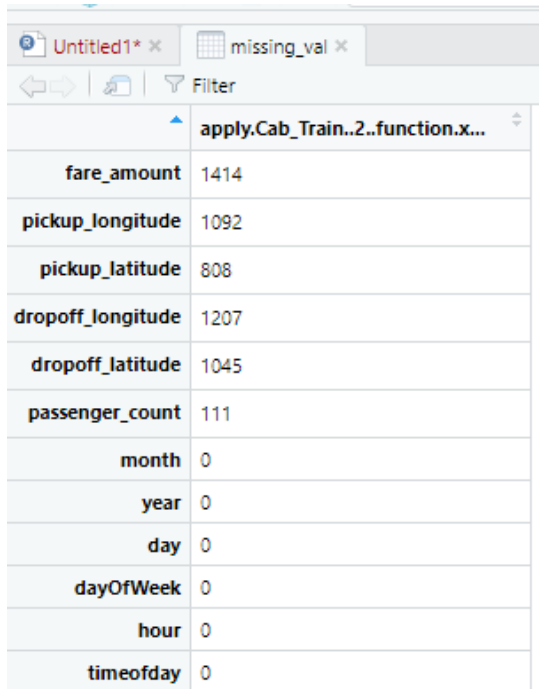
As can be seen we have some outliers in all columns but the count is too high to remove. So instead we will make them as NA. We don't mark as NA outlier passenger count as it will lead to loss of 4, 5, and 6 values which are valid values.

2.1.3 Missing Value Analysis

Missing value analysis is done to check if there are any missing values present in the given dataset. Missing values can be easily treated using various methods like mean, median method, knn method to impute missing values.

In R `function(x) {sum (is.na(x))}` is the function used to check the sum of missing values.

In python `Cab_Train.isnull ().sum ()` is used to detect any missing value



The screenshot shows a data table with the following variables and values:

Variable	Value
fare_amount	1414
pickup_longitude	1092
pickup_latitude	808
dropoff_longitude	1207
dropoff_latitude	1045
passenger_count	111
month	0
year	0
day	0
dayOfWeek	0
hour	0
timeofday	0

We find that all missing values are in numeric variables. We find the best of mean and median for every continuous variable by making a known value as NA and finding which of mean or median gives a value as close to the known value as possible and use that to calculate the rest of the missing values for that variable.

```

Cab_Train$pickup_longitude [is.na (Cab_Train$pickup_longitude)] = median
(Cab_Train$pickup_longitude, na.rm = TRUE)
Cab_Train$pickup_latitude [is.na (Cab_Train$pickup_latitude)] = mean
(Cab_Train$pickup_latitude, na.rm = TRUE)
Cab_Train$dropoff_longitude [is.na (Cab_Train$dropoff_longitude)] = median
(Cab_Train$dropoff_longitude, na.rm = TRUE)
Cab_Train$dropoff_latitude [is.na (Cab_Train$dropoff_latitude)] = median
(Cab_Train$dropoff_latitude, na.rm = TRUE)
Cab_Train$fare_amount [is.na (Cab_Train$fare_amount)] = mean (Cab_Train$fare_amount, na.rm
= TRUE)
Cab_Train$passenger_count [is.na (Cab_Train$passenger_count)] = median
(Cab_Train$passenger_count, na.rm = TRUE)

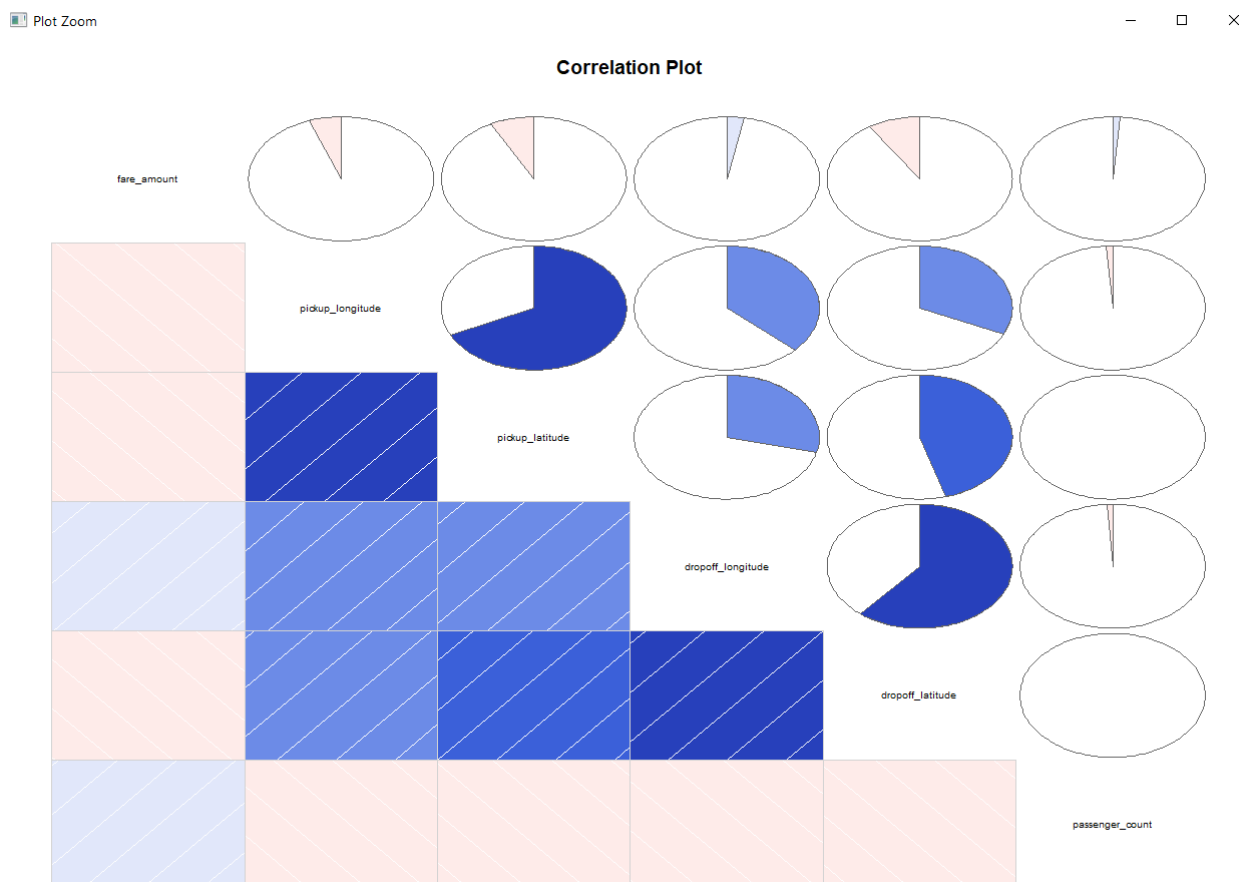
```

2.1.4 Feature Selection

Feature selection analysis is done to select subsets of relevant features (variables, predictors) to be in model construction.

We use correlation method for feature selection of continuous variables and anova test for feature selection of categorical variable.

Below Figure shows a correlation plot for all numeric variable present in dataset



In above visualization we can see that although pickup_longitude and pickup_latitude do so some correlation it's not strong enough and we need both cols and hence we don't delete any numeric variables. The above is supported by corr function which gives output as below:

```
C:/Users/vgonsalv/Desktop/DataScience/Edwisor/project/project 2--Cab Fare Prediction/Project Cab Fare Prediction--Vijay Gonsalves/
> cor(numeric_Cab_Train)
      fare_amount pickup_longitude pickup_latitude dropoff_longitude dropoff_latitude
fare_amount    1.000000000    -0.05380993    -0.07316962     0.02609629    -0.08119742
pickup_longitude -0.05380993     1.00000000     0.65869550     0.34831821     0.30538843
pickup_latitude  -0.07316962     0.65869550     1.00000000     0.27772204     0.43065757
dropoff_longitude 0.02609629     0.34831821     0.27772204     1.00000000     0.57093528
dropoff_latitude -0.08119742     0.30538843     0.43065757     0.57093528     1.00000000
> |
```

From anova test using F-value statistic we exclude day and dayofweek variables out of categorical variables.

2.1.5 Feature Scaling

Feature scaling includes two functions normalization and standardization. It is done reduce unwanted variation either within or between variables and to bring all of the variables into proportion with one another.

Both numerical variables are in same range and hence no need of feature scaling in this data set.

2.1.6 Multicollinearity Check

There is no Multicollinearity problem in the given data set

```
> vifcor(Cab_Train[,-c(1,6:10)], th = 0.9)
No variable from the 4 input variables has collinearity problem.

The linear correlation coefficients ranges between:
min correlation ( dropoff_longitude ~ pickup_latitude ): 0.2782098
max correlation ( pickup_latitude ~ pickup_longitude ): 0.6617794

----- VIFs of the remained variables -----
      Variables      VIF
1 pickup_longitude 1.870722
2 pickup_latitude 1.977738
3 dropoff_longitude 1.586554
4 dropoff_latitude 1.690659
> |
```


2.2 Modeling

2.2.1 Model Selection

In this case we have to predict the fare of cabs according to pick up and drop locations and day, month etc. i.e. time conditions. So the target variable here is a continuous variable. For Continuous we can use various Regression models. Model having less error rate and more accuracy will be our final model.

Models built are

c50 (Decision tree for regression target variable)

Linear regression

KNN regression

Random Forest

2.2.2 C50 Decision Tree

This model is also known a Decision tree for regression target variable.

For this model we have divided the dataset into train and validation part using stratified sampling on passenger count to avoid bias towards certain variables Where train contains 80% data of data set and validation contains 20% data and contains 10 variable where 10th variable is the target variable.

Creating Model in R

```
-->using decision tree algo
fit = rpart(c(fare_amount) ~ ., data = train_set, method = "anova")
predictions_DT = predict(fit, validation_set[, -c(1)])
MAPE = function(y, yhat) {
  mean(abs((y - yhat) / y)*100)
}
MAPE(validation_set[, c(1)], predictions_DT)
-->only fare_amount-->37.53% error
```

In python

```
In [64]: #####c50#####
fit_DT = DecisionTreeRegressor(max_depth=2).fit(train_set.iloc[:,1:10], train_set.iloc[:,0])
predictions_DT = fit_DT.predict(validation_set.iloc[:,1:10])
```

2.2.3 Random Forest

For this model we have divided the dataset into train and validation part using stratified sampling on passenger count to avoid bias towards certain variables. Where train contains 80% data of data set and validation contains 20% data and contains 10 variable where 10th variable is the target variable.

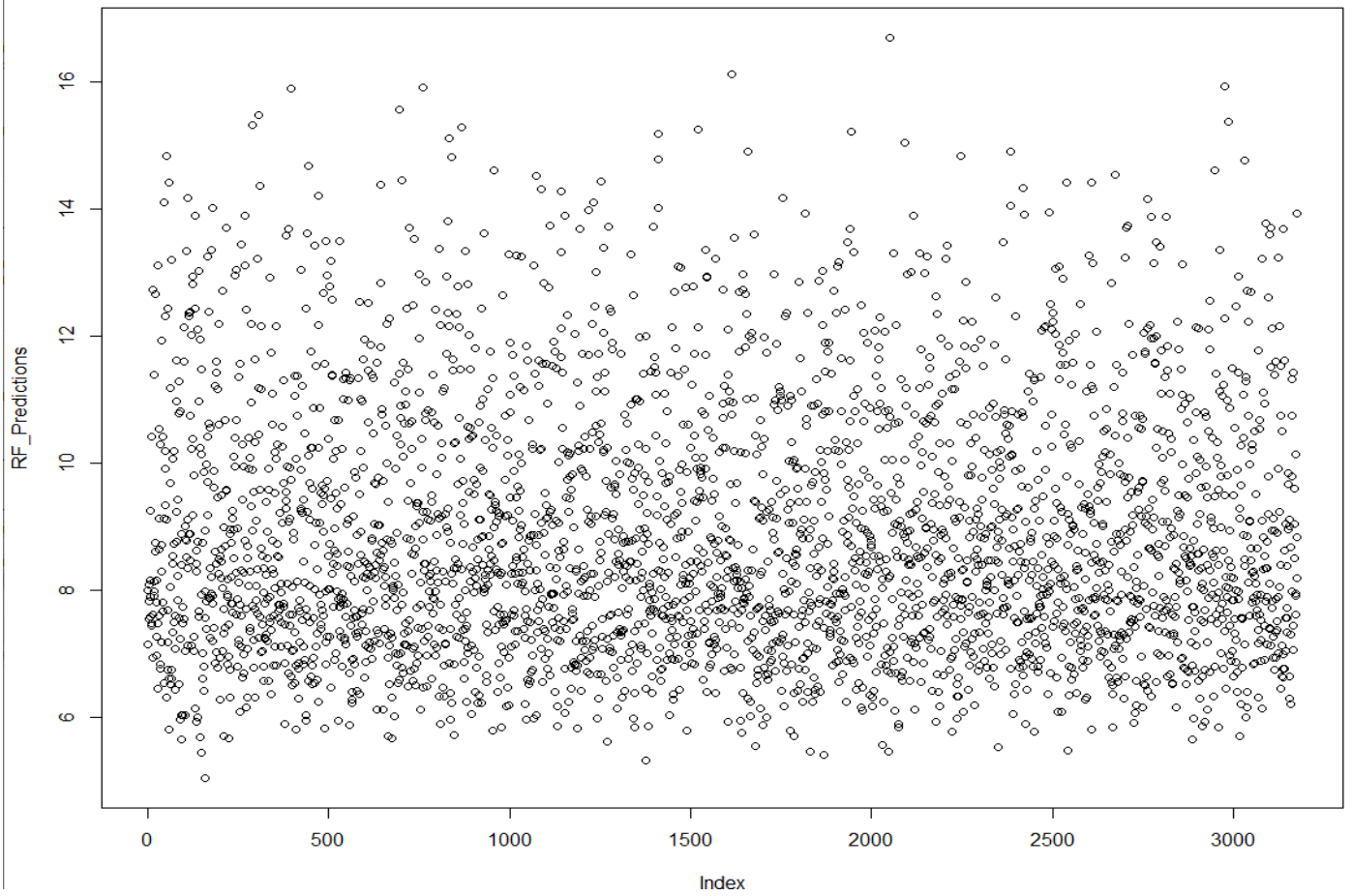
In this model we are using default trees to predict the target variable.

Creating Model in Python

```
In [67]: RF_model = RandomForestRegressor(n_estimators = 500).fit(train_set.iloc[:,1:10], train_set.iloc[:,0])
RF_Predictions = RF_model.predict(validation_set.iloc[:,1:10])
```

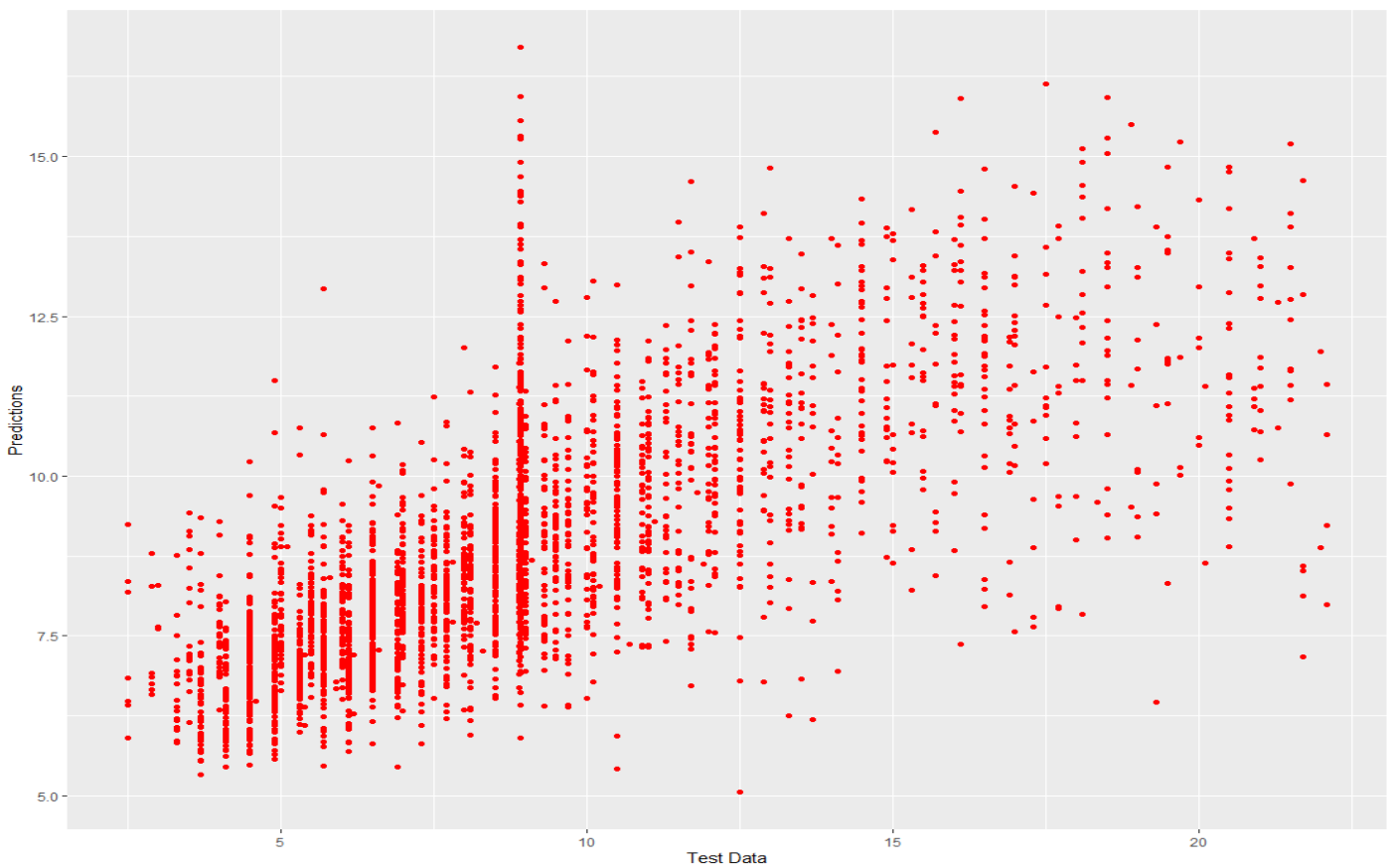
In R

```
#-----
#-->using random forest
RF_Model <- randomForest(fare_amount~.,data = train_set, importance = TRUE)
varImpPlot(RF_Model)
imp <- importance(RF_Model)
RF_Predictions <- predict(RF_Model,validation_set[, -c(1)])
MAPE(validation_set[, c(1)], RF_Predictions)
plot(RF_Predictions)
#-->only fare_amount-->26.95 %
#-----
```



Plot Zoom

— □ ×



2.2.4 Linear Regression

In Linear regression we first check collinearity of numeric variables and only if there is no collinearity we go for model creation.

For this model we have divided the dataset into train and validation part using stratified sampling on passenger count to avoid bias towards certain variables Where train contains 80% data of data set and validation contains 20% data and contains 10 variable where 10th variable is the target variable.

Creating Model in R

```
i8
i9 lm_model=lm(fare_amount~.,data = train_set)
i0 summary(lm_model)
i1 predictions_LR=predict(lm_model,validation_set[, -c(1)])
i2 MAPE(validation_set[, c(1)], predictions_LR)
i3
i4 #-->only fare_amount-->39.23% error
i5 #-----
```

In python

```
In [69]: ln_model=sm.OLS(train_set.iloc[:,0],train_set.iloc[:,1:10].astype(float)).fit()
         LN_Predictions = ln_model.predict(validation_set.iloc[:,1:10])
```

3. Conclusion

3.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. We can compare the models using any of the following criteria:

Predictive Performance

Interpretability

Computational Efficiency

In our case of Cab Fare Prediction, the latter two, Interpretability and Computation Efficiency, do not hold much significance. Therefore we will use Predictive performance as the criteria to compare and evaluate models.

Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

3.1.1 Mean Absolute Percentage Error (MAPE)

MAPE is one of the error measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous sections

```
MAPE = function(y, yhat) {
  mean(abs((y - yhat) / y)*100)
}
```

In above function y is the actual value and yhat is the predicted value. It will provide the error percentage of model.

MAPE value in Python are as follow

```
In [70]: MAPE(validation_set.iloc[:,0],LN_Predictions)
```

```
Out[70]: 38.80605662281014
```

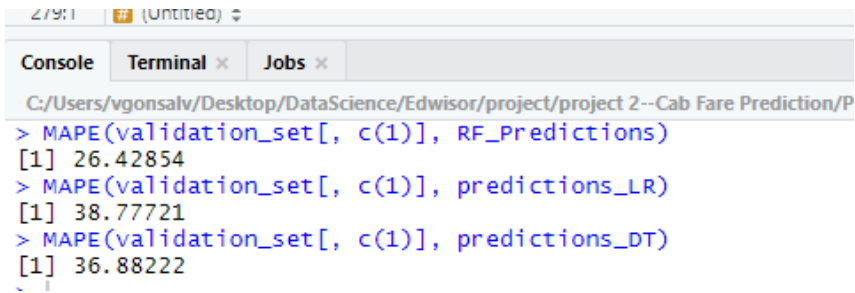
```
In [71]: MAPE(validation_set.iloc[:,0],RF_Predictions)
```

```
Out[71]: 22.991287919808258
```

```
In [72]: MAPE(validation_set.iloc[:,0],predictions_DT)
```

```
Out[72]: 37.78790177142182
```

MAPE values in R are as follows:



```
2/9/1  [Untitled]
Console Terminal x Jobs x
C:/Users/vgonsalv/Desktop/DataScience/Edwisor/project/project 2--Cab Fare Prediction/P
> MAPE(validation_set[, c(1)], RF_Predictions)
[1] 26.42854
> MAPE(validation_set[, c(1)], predictions_LR)
[1] 38.77721
> MAPE(validation_set[, c(1)], predictions_DT)
[1] 36.88222
```

Where predictions_DT are predicted values from C50 model. RF_predictions are predicted values from random forest model and predictions_LR are predicted values from linear regression model

3.2 Model Selection

We can see that from both R and Python Random forest model performs best out of C50 and linear regression. So random forest model is selected with 74% accuracy in R and with 77% accuracy in python.

We actually got better accuracy from KNN but we did not choose it as it is built only on continuous variables and not categorical variables and our data has influence of both continuous and categorical variables.

Extracted predicted value of random forest model are saved in "Random_Forest_output_R_on_validation_data.csv" file.

Model is run on test data and output is stored in "pred_output_on_test_data.csv" file.

Appendix A—R code



Cab_Fare_Prediction_
(R_Code)_Vijay_Gonsalves

Appendix B—Python code



Cab_Fare_Prediction_
(Python_Code)_Vijay_Gonsalves

Appendix C—Source data



train_cab.csv



test.csv