|  |
| --- |
| Report Notebook |

### Edwisor

### Sai pavan

### Annamraju

Cab Fare Prediction

## Python & R

IN THIS SECTION:

* Problem Statement
* Shape of data
* Feature Definitions
* Missing values, Data Types

# 

Problem Statement:

# Data understanding

Report Notebook

Tab 1

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.

Characteristics of Data:

**Shape of Data:**

for **train\_cab.csv:**

|  |  |  |
| --- | --- | --- |
| S. No | Name of the variable | Count |
| 1 | fare\_amount | 16043 |
| 2 | pickup\_datetime | 16067 |
| 3 | pickup\_longitude | 16067 |
| 4 | pickup\_latitude | 16067 |
| 5 | dropoff\_longitude | 16067 |
| 6 | dropoff\_latitude | 16067 |
| 7 | passenger\_count | 16012 |

**for test.csv:**

|  |  |  |
| --- | --- | --- |
| S. No | Name of the variable | Count |
| 1 | pickup\_datetime | 9914 |
| 2 | pickup\_longitude | 9914 |
| 3 | pickup\_latitude | 9914 |
| 4 | dropoff\_longitude | 9914 |
| 5 | dropoff\_latitude | 9914 |
| 6 | passenger\_count | 9914 |

**Feature Definitions:**

|  |  |  |
| --- | --- | --- |
| S. No | Name of the variable | Description |
| 1 | fare\_amount | Final fare of the cab |
| 2 | pickup\_datetime | Date and time of the passenger pickup |
| 3 | pickup\_longitude | Longitude coordinate of the picking up the passenger |
| 4 | pickup\_latitude | Latitude coordinate of the picking up the passenger |
| 5 | dropoff\_longitude | Longitude coordinate of the droppint point of the passenger |
| 6 | dropoff\_latitude | Latitude coordinate of the dropping point of the passenger |
| 7 | passenger\_count | No. of passengers for the ride. |

**Missing values and Type conversions needed to be dealt:**

There are missing values in passenger count and also in fare amount.

Type conversions are needed to be done for both pickup\_datetime and also for fare amount.

These all need to be dealt with in both train and test datasets.

Missing Value Analysis:

IN THIS SECTION:

* Missing Value Analysis
* Datatype Conversions
* Feature Engineering
* Sanity Checks
* Outliers Analysis

# 

# Data Preprocesing

Report Notebook

Tab 2

Missing values are the values that are not either filled or lost in the process of collecting the data. Missing values can be of one of the types below:

1. Missing values completely at random

2. Missing values not at random

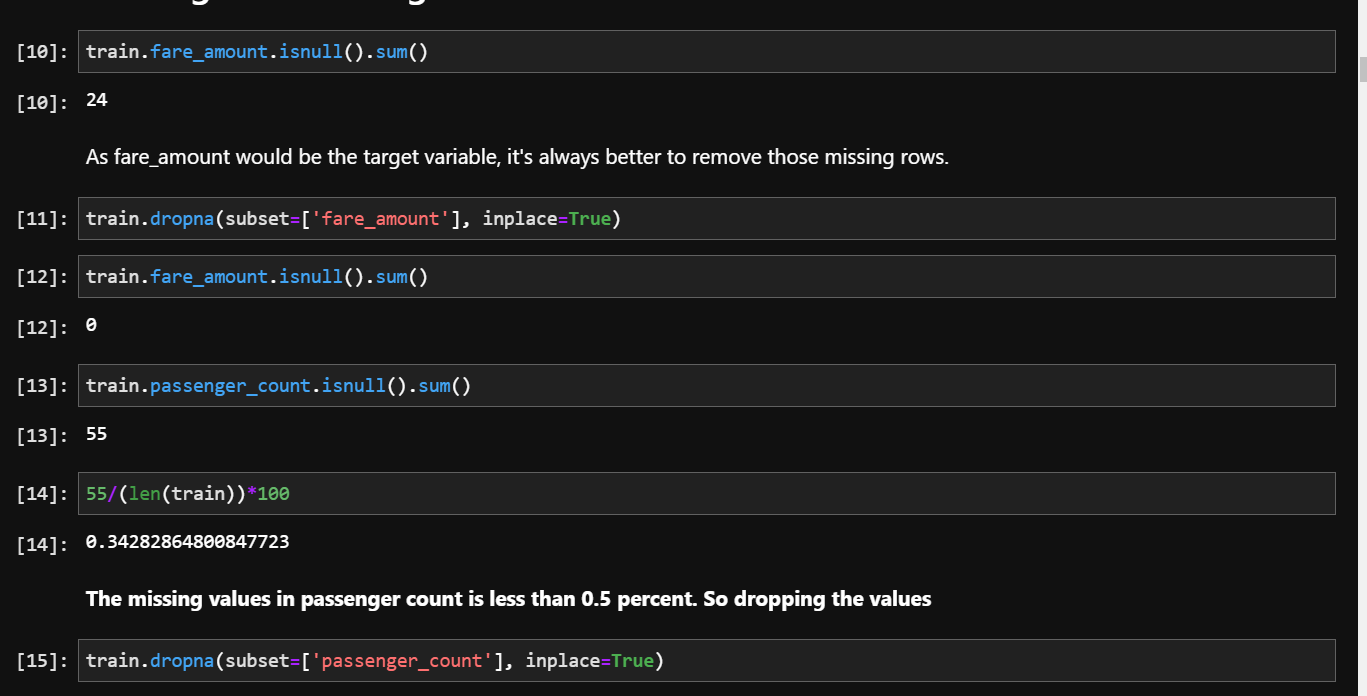
3. Missing values at random

**For train\_cab.csv:**

Missing values in train\_cab are:

1. Fare Amount: As this is the target variable, these values are being dropped

2. Passenger Count : As the no of values missing are less than 1 percent, these values are also being dropped.



**For test.csv:**

There are no missing values in test.csv

Datatype conversions:

The default types of the variables in the dataset are:

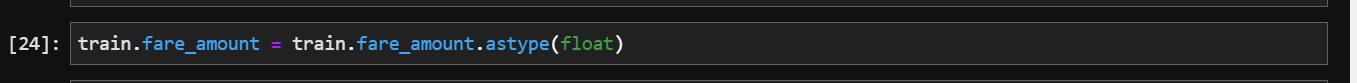
|  |  |  |
| --- | --- | --- |
| S. No | Name of the variable | Type |
| 1 | fare\_amount | Object |
| 2 | pickup\_datetime | Object |
| 3 | pickup\_longitude | Float |
| 4 | pickup\_latitude | Float |
| 5 | dropoff\_longitude | Float |
| 6 | dropoff\_latitude | Float |
| 7 | passenger\_count | Float |

So, the type conversions needed:

1. Fare Amount 🡪 Will be converted into float.

2. pickup\_datetime 🡪 needed tobe converted into datetime object but as features of date, time needed to be extracted, this will be dealt with in feature engineering section.

After removing invalid values in fare amount like ‘430-‘, the variable is converted into float.



After doing the feature engineering and sanity checks, will be converting passenger count

Feature Engineering

Feature Engineering is extracting useful features that will help the model from the given data.

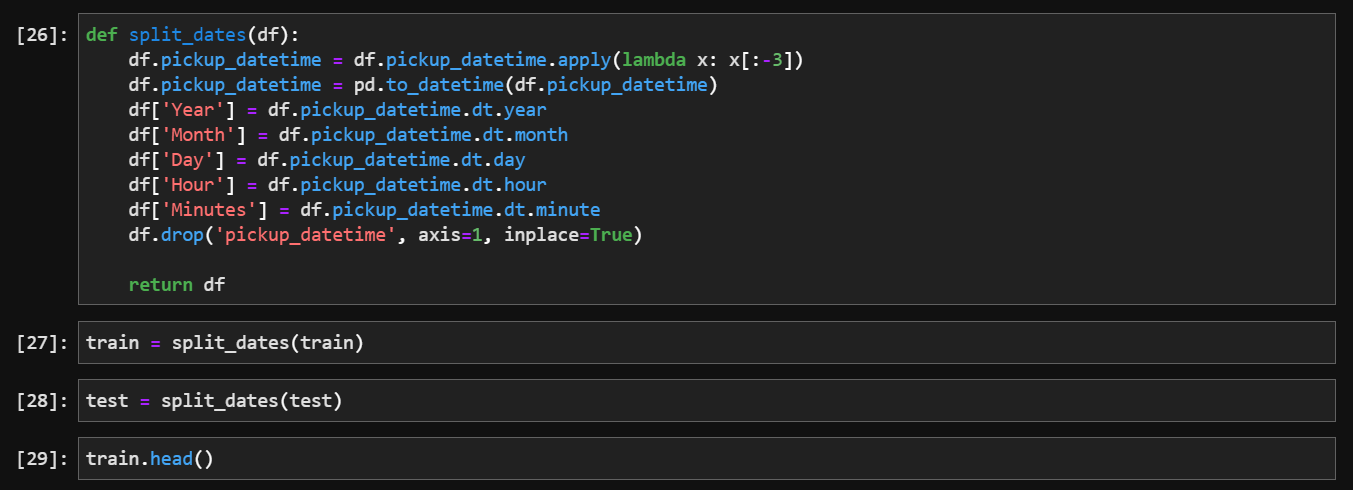
Feature Engineering here is needed to be performed on pickup\_datetime and from longitudes and latitudes of pickup and dropoff location to achieve distance.

For both test and train csv data,

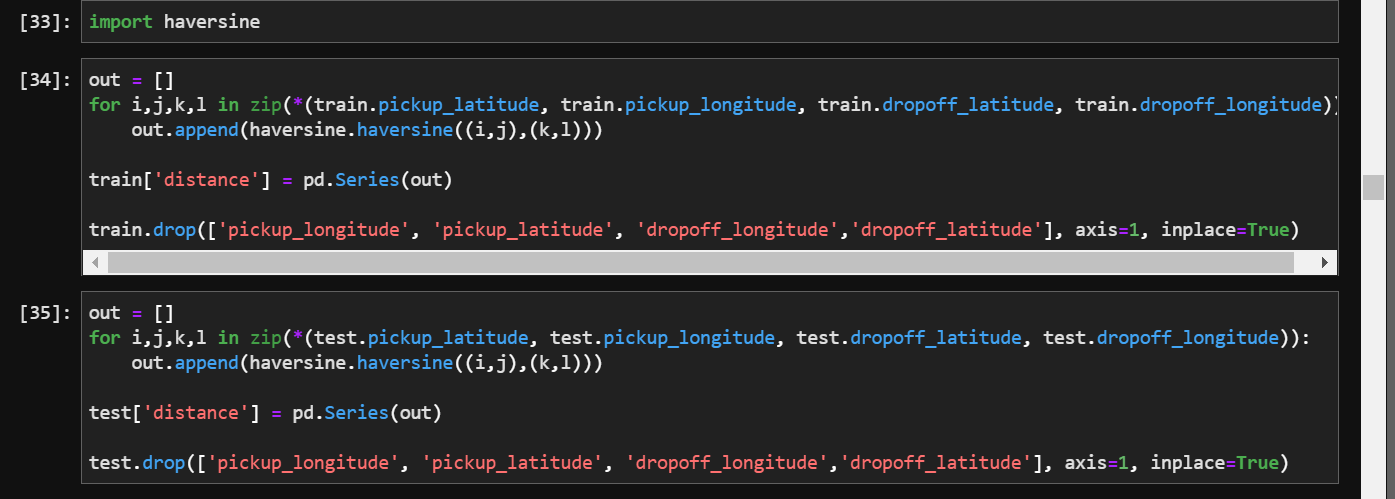
For pickup datetime has been converted into datetime using pandas pd.to\_numeric() method and then extracted new variables like **Year, Month, Day, Hours, Minutes**

For distance, I have installed haversine module which has the method haversine for converting coordinate data into distance.

This has been done for both training and testing data.



Using haversine module, will be calculating the distance from latitudes and longitudes from dropoff and pickup coordinates.



Sanity Checks

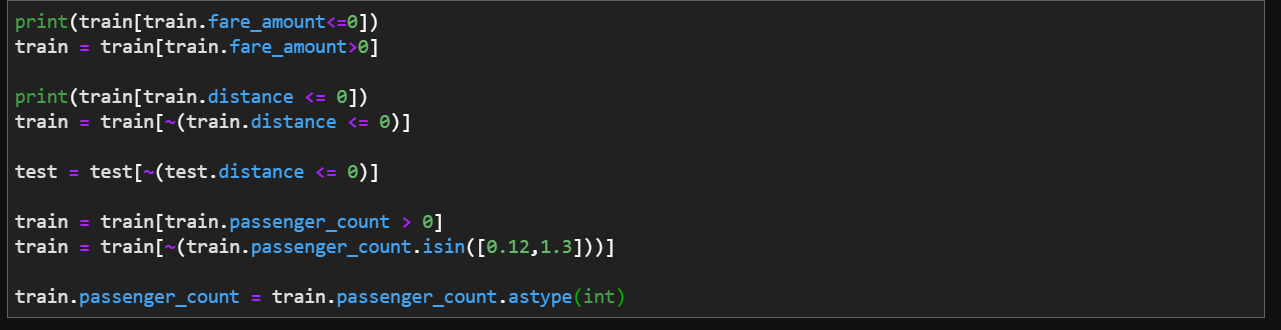
Sanity Checks are important for data validity. We cannot have variables like age or distance in a negative value or a zero value which will not contribute to the model.

So we need to perform sanity checks for making sure that the variables data is valid and accountable for model building.

Sanity checks to be performed:

1. fare amount should not be less than or equal to zero
2. We also have to convert the distance of the ride using pickup and dropoff coordinates
3. Then we have to check for the distances less than or equal to 0
4. Check Passenger count, which should be integer and less than or equal to 6 and greater than 0

This has been done both for test and training data.

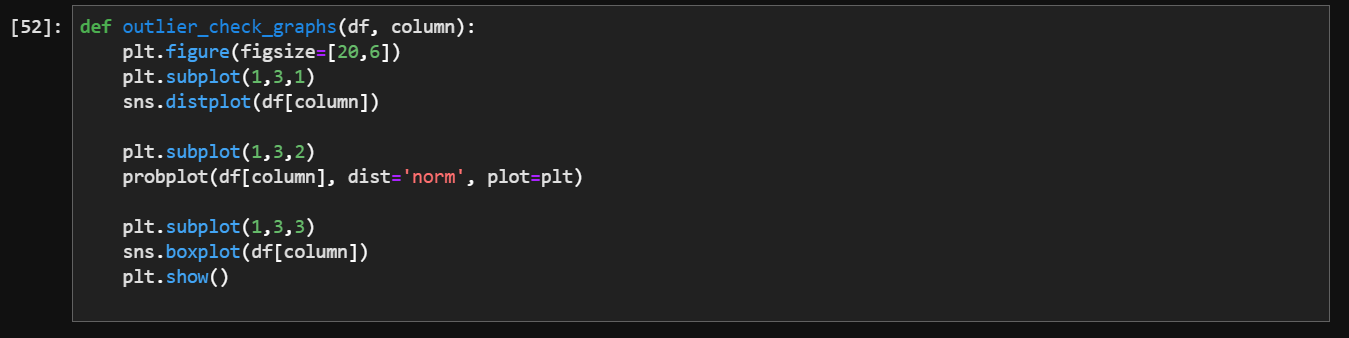


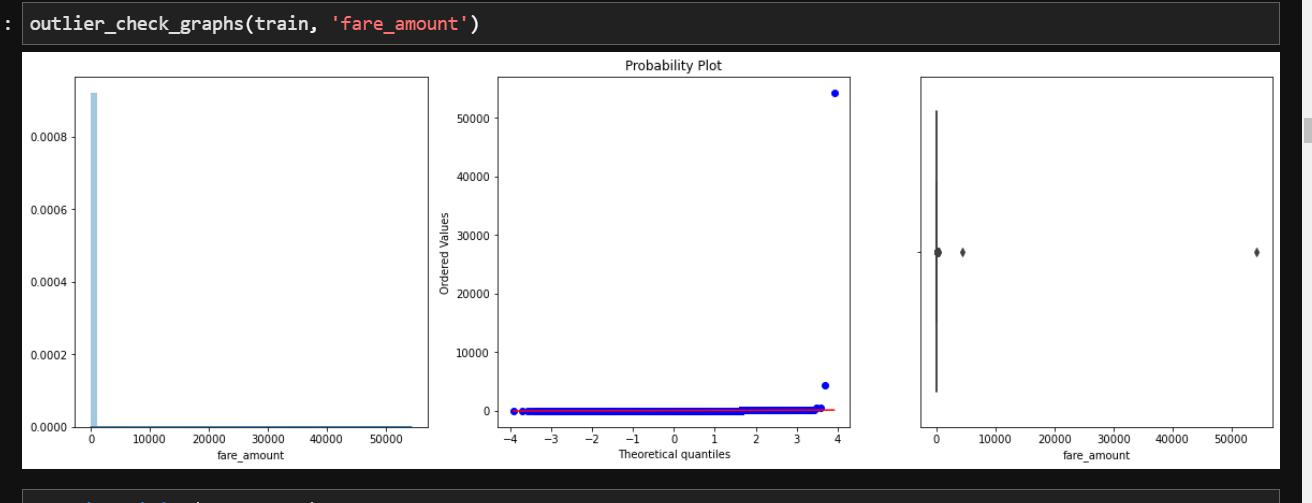
Outlier Analysis

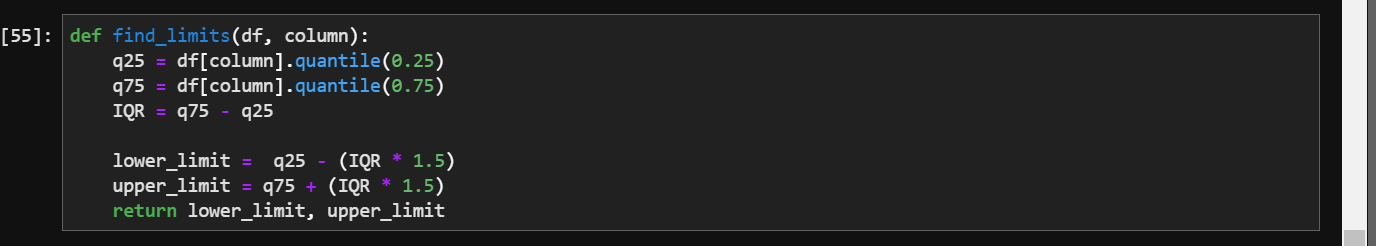
For checking outliers, I used a function where it displays qqplot graph, distribution plot and boxplot. Using that I analysed which ones had outliers and took action on them by capping the data with lower and upper limits of the boxplot using iqr and lower and upper values.

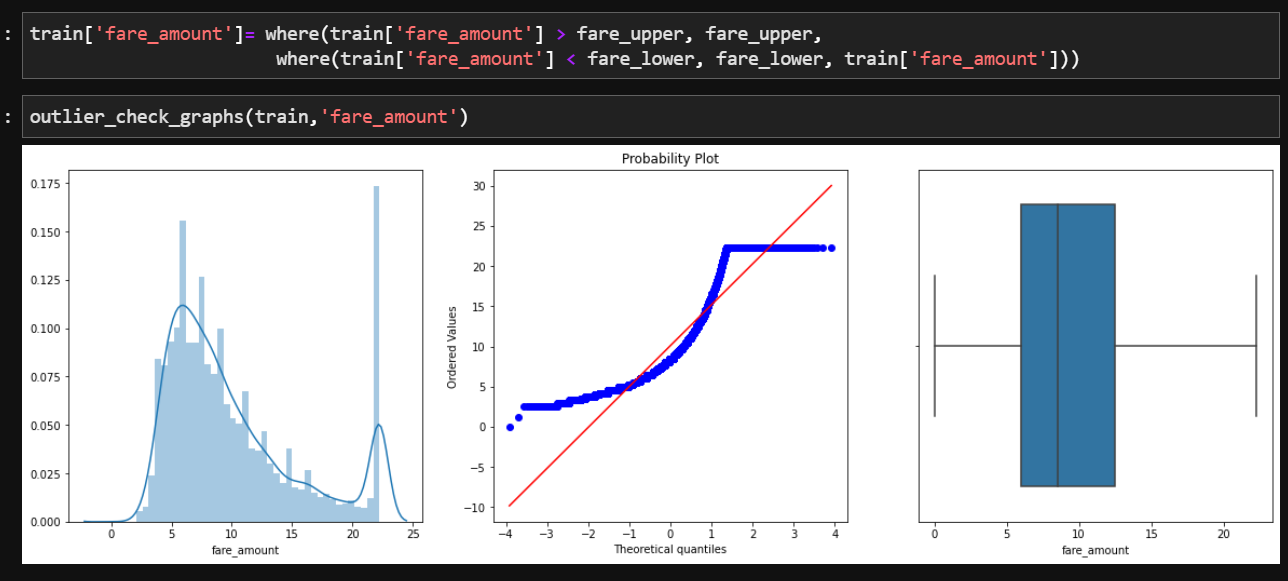
This is done for fare\_amount, distance, which are the only numerical data in the whole dataset.

This is done for both training and testing data.









# 

# Exploratory data analysis

IN THIS SECTION:

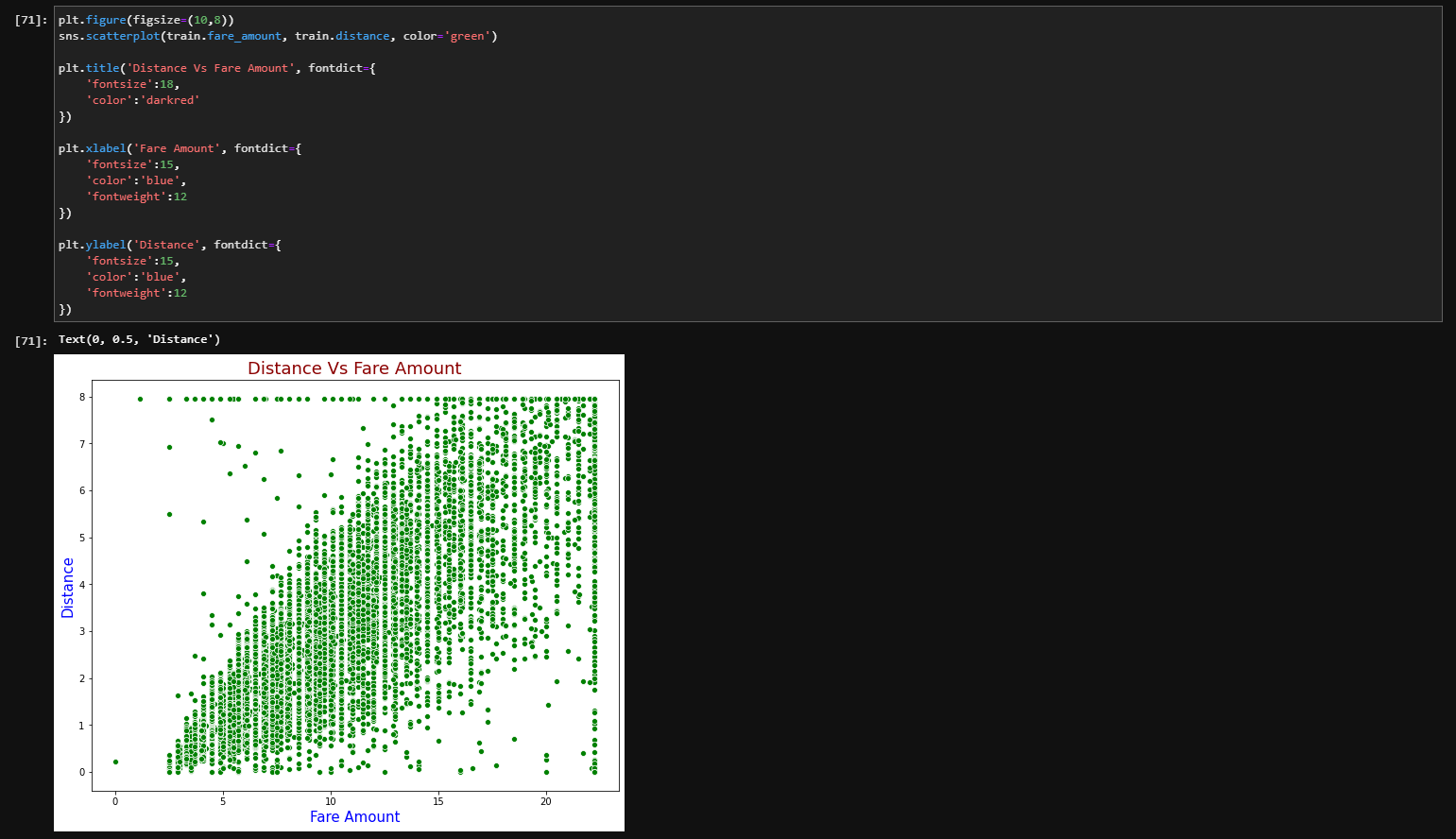
* Plotting Numerical VS Numerical Variables
* Plotting Categorical VS Numerical Variables
* Drawing insights from heatmap

Report Notebook

Tab 3

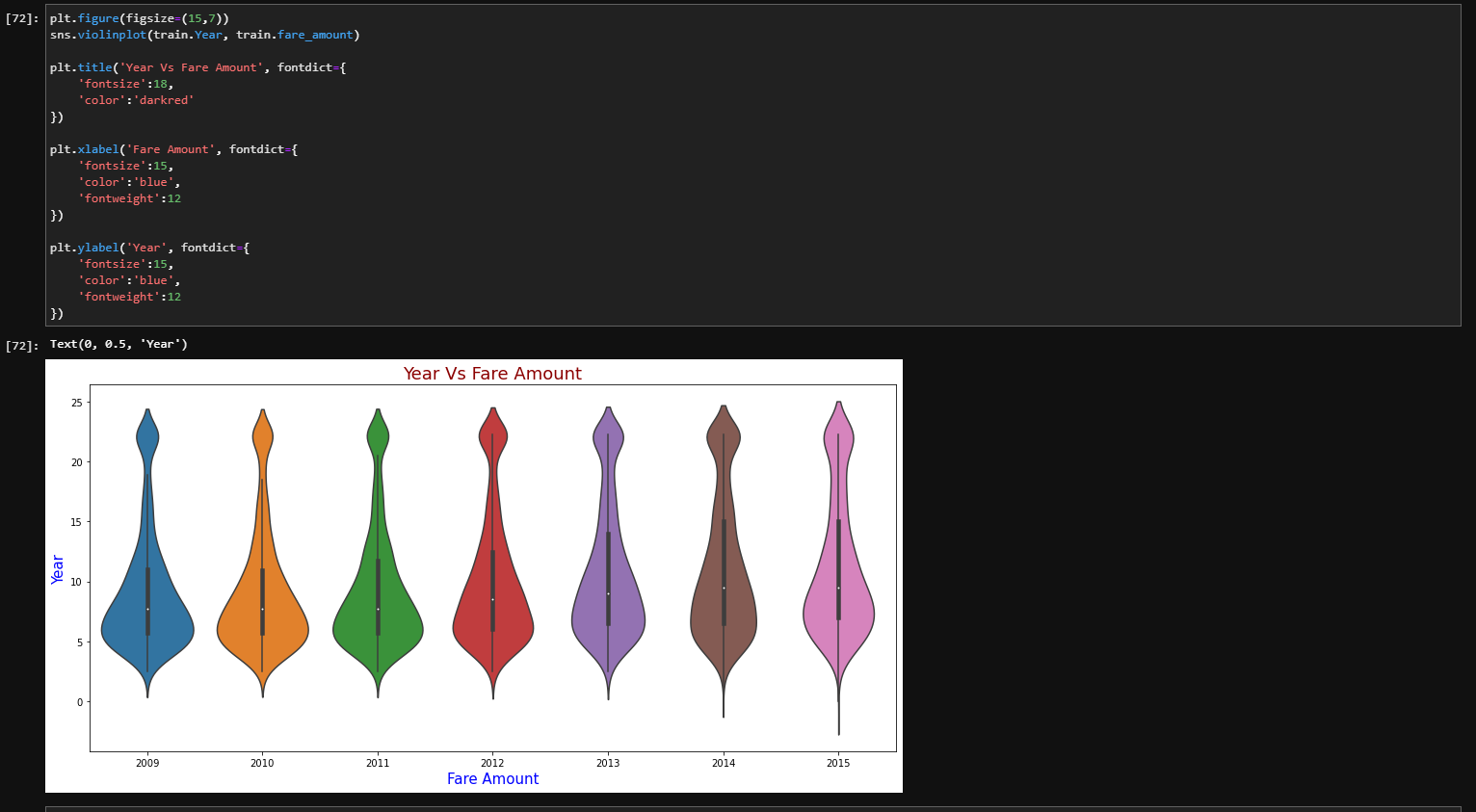
Plotting Numerical VS Numerical Data

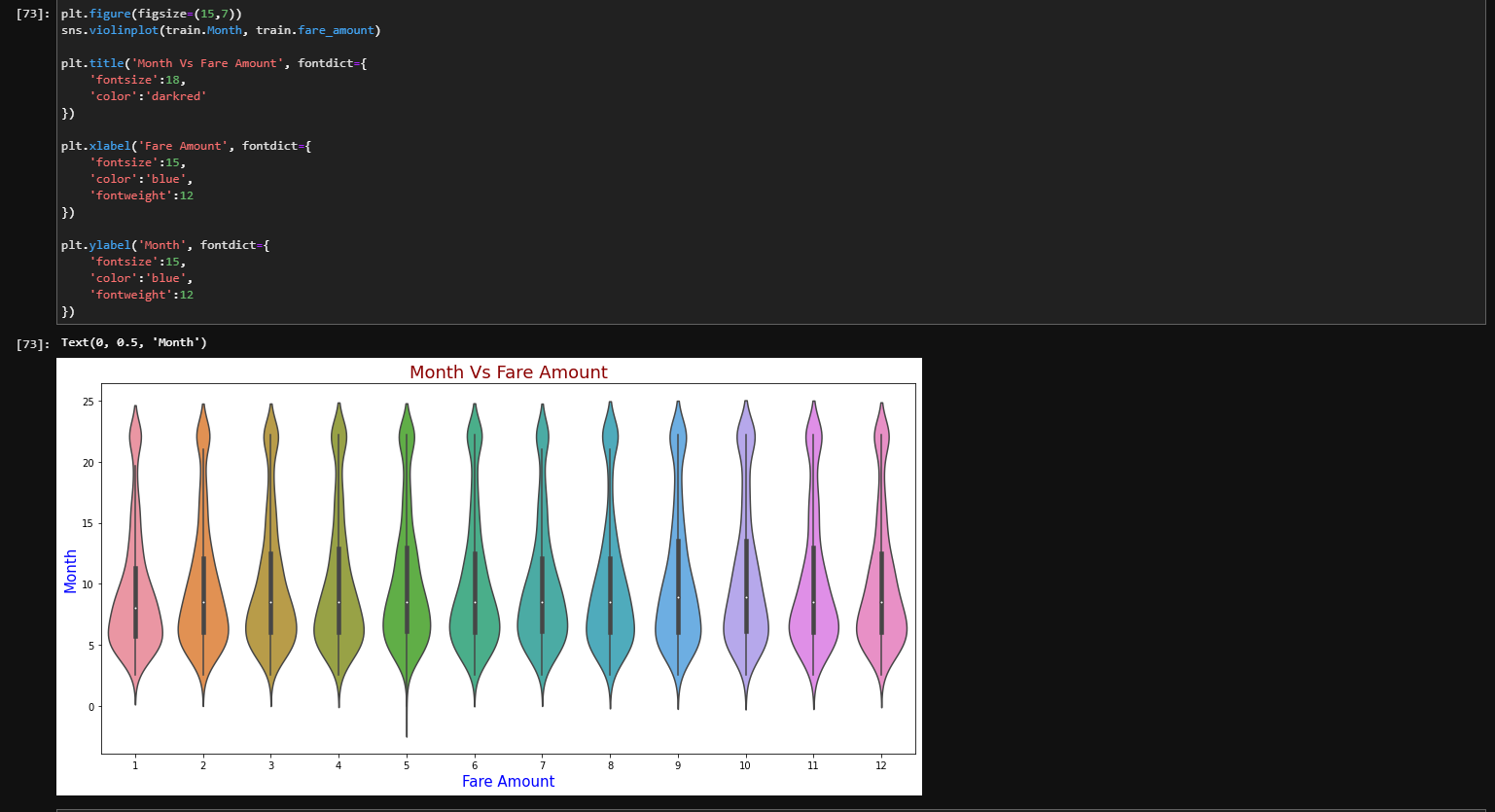
The target variable is here fare amount which is numerical data, also the distance. So plotting fare amount and distance.

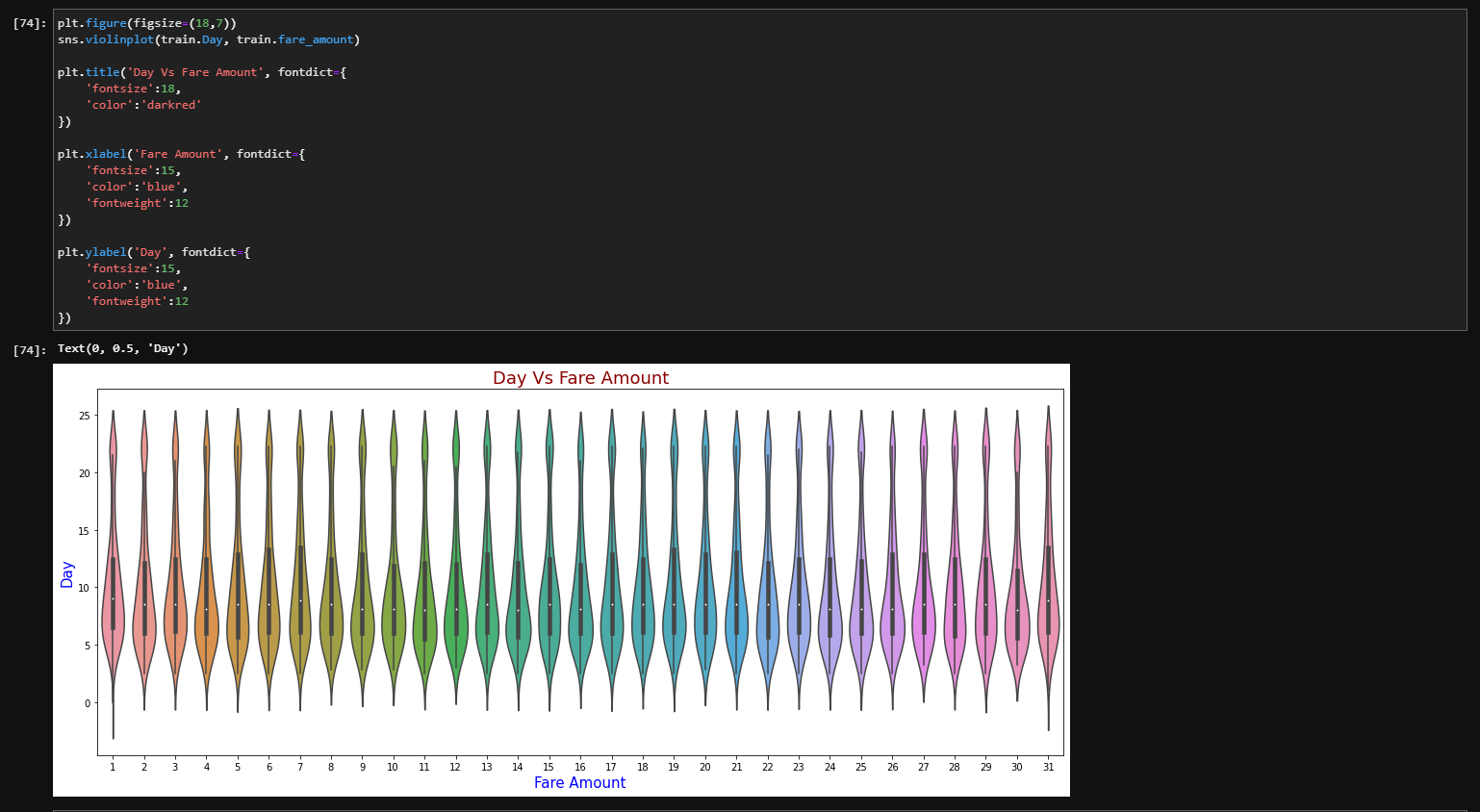


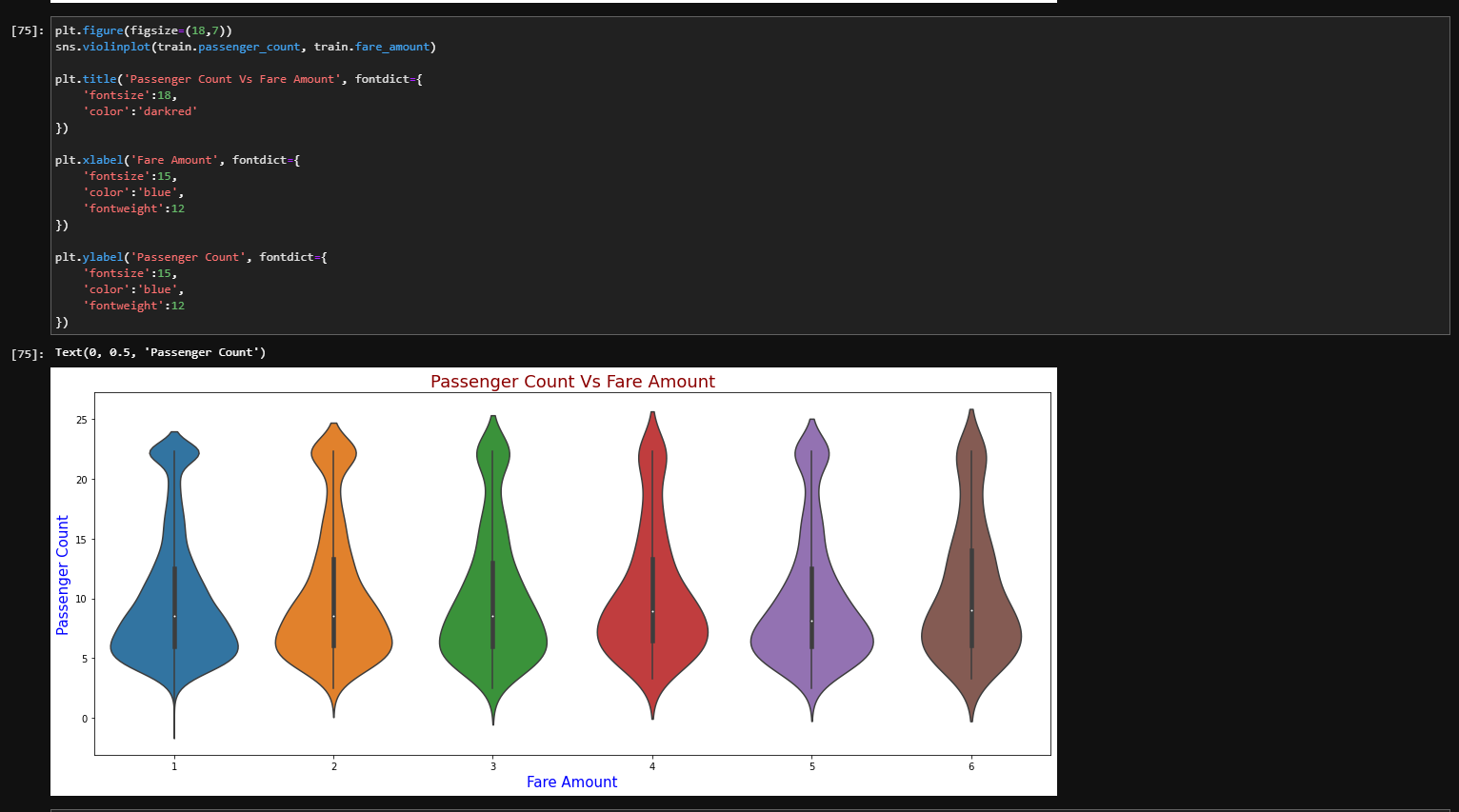
Plotting Categorical VS Numerical Variables

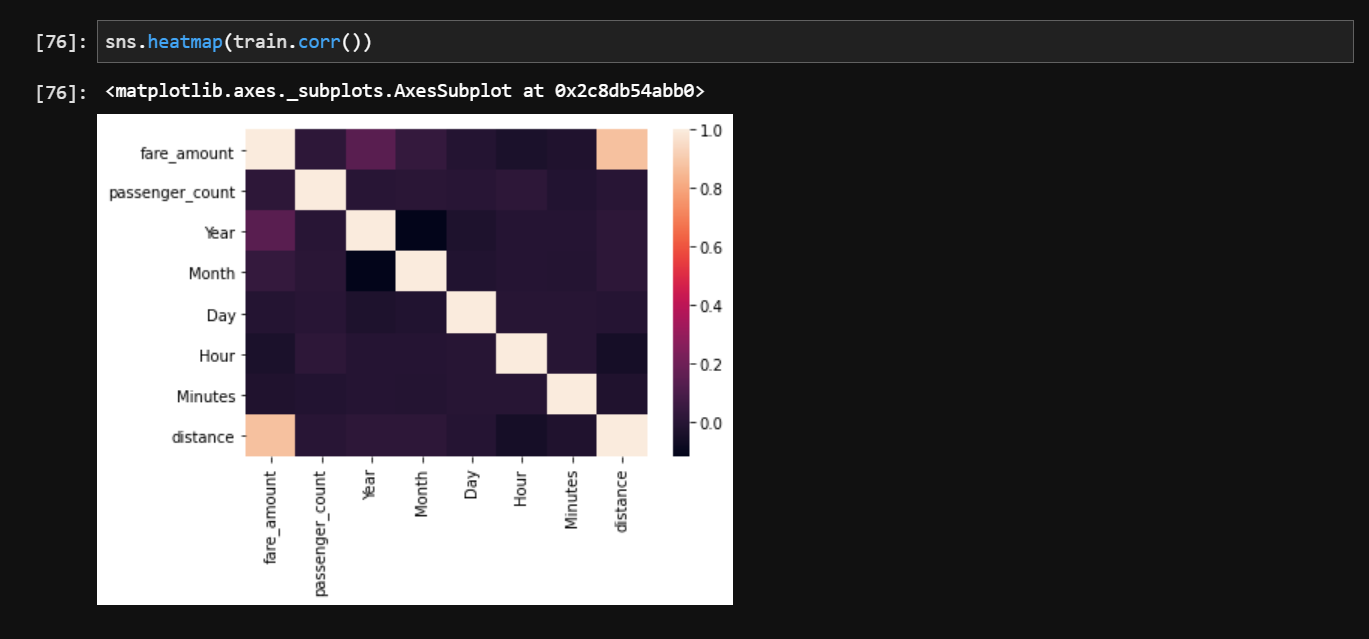
Drawing insights from heatmap











IN THIS SECTION:

* Splitting data into training and validation data
* Using and Analyzing Linear Regression
* Using Ridge, Lasso, Support Vector Regression
* Using Decision Tree and Random Forest Regressors
* Using Gradient Boosting Regression
* Analyzing the model results
* Deciding Metric

# 

# Modelling the data

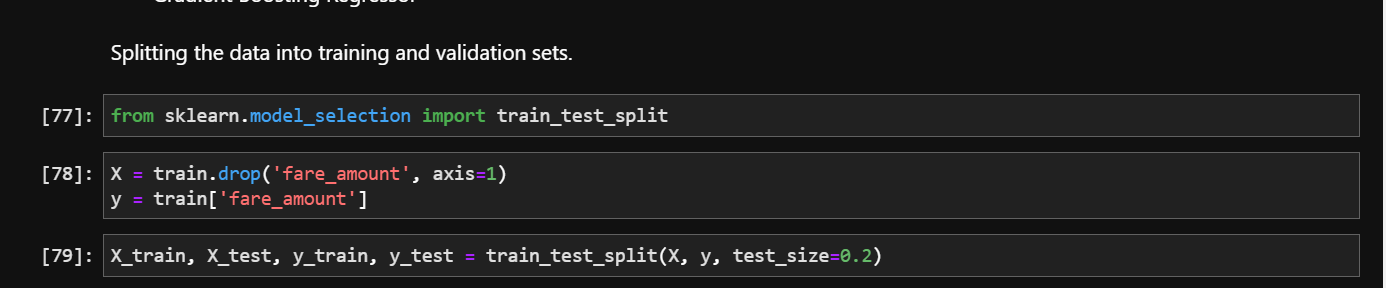
Report Notebook

Tab 4

Splitting data into training and testing data

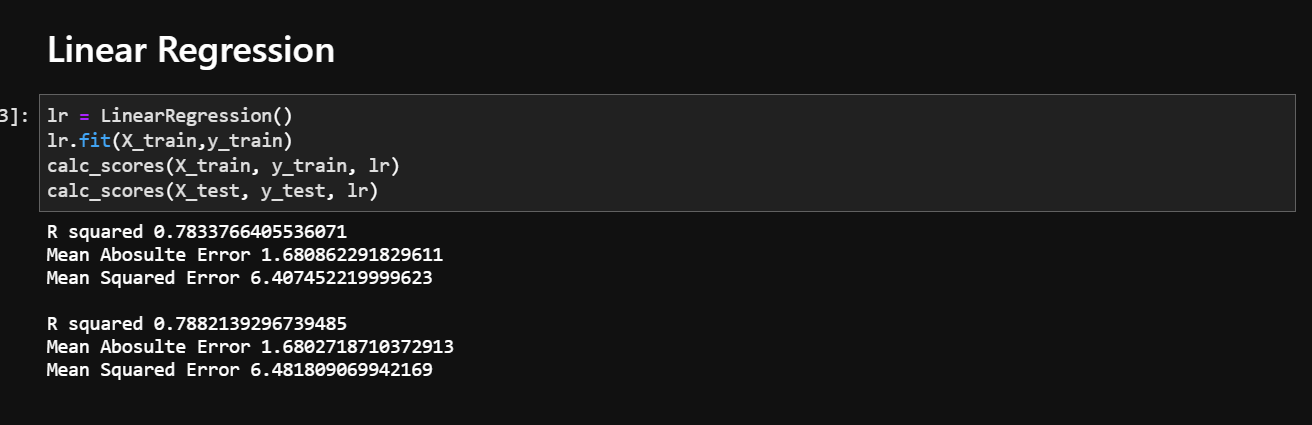
The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model.

It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure should not be used, such as when you have a small dataset and situations where additional configuration is required, such as when it is used for classification and the dataset is not balanced.



Using and Analyzing Linear Regression

Linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. The case of one explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression.



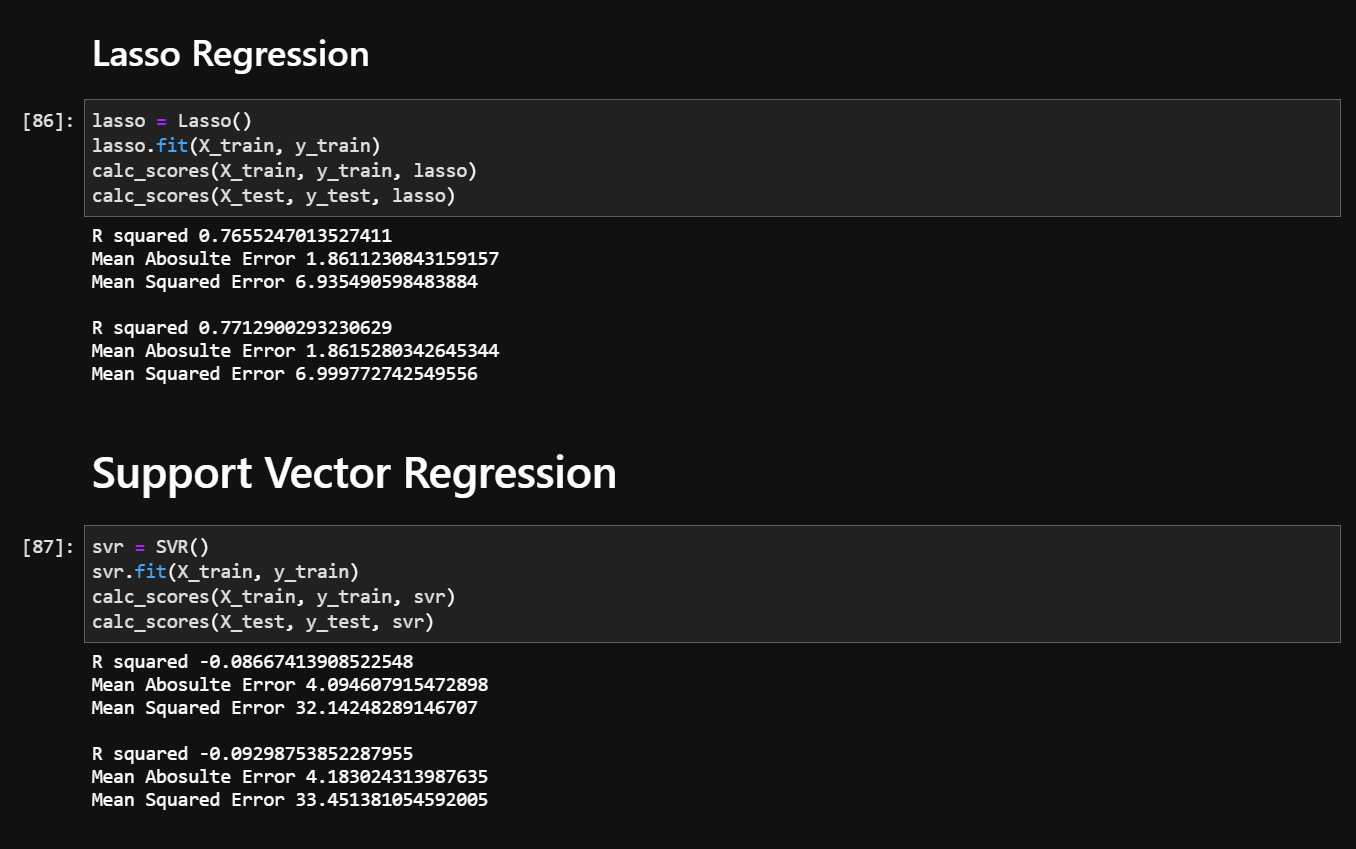
The reason why Linear Regression is not good enough for the data is that there is no linearity between most of the variables, being categorical data. This can be viewed in the heat map.

In these situations ensemble methods give the best results. But we will also view other regression algorithms like Lasso and SVR.

Using Lasso, Support Vector Regression

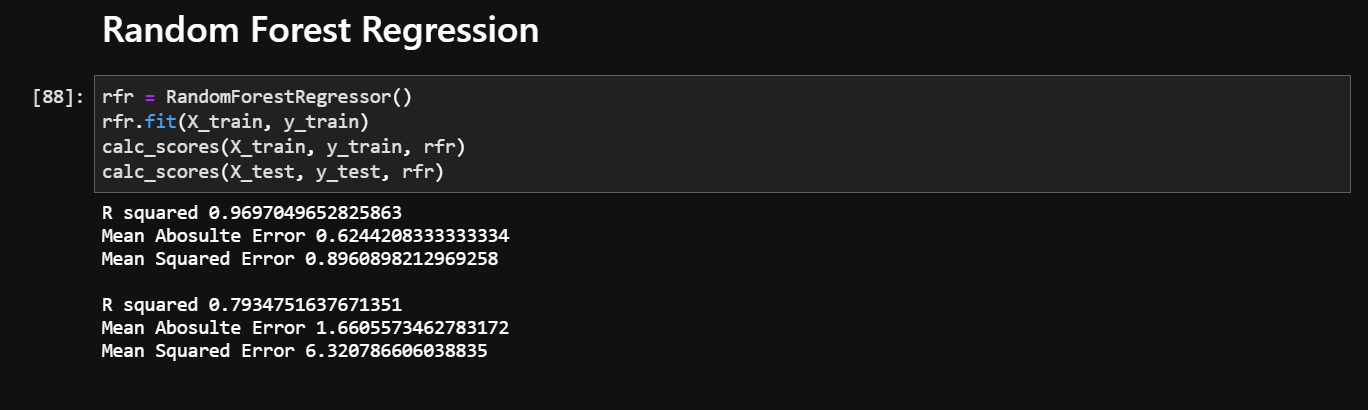
Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

Support vector regression (SVR) is characterized by the use of kernels, sparse solution, and VC control of the margin and the number of support vectors. Although less popular than SVM, SVR has been proven to be an effective tool in real-value function estimation.



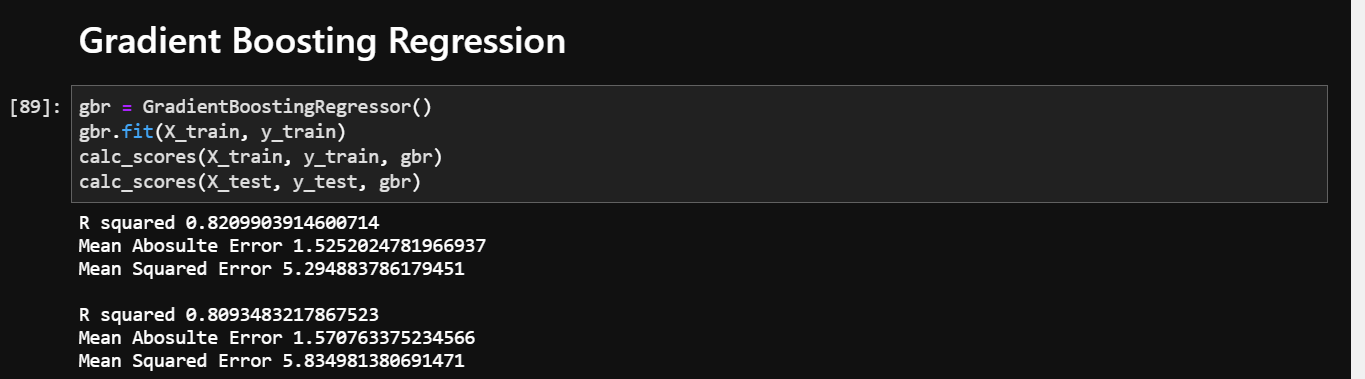
Using and Analyzing Random Forest Regression

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.



Using and Analyzing Gradient Regression

Gradient Descent is the process of minimizing a function by following the gradients of the cost function. This involves knowing the form of the cost as well as the derivative so that from a given point you know the gradient and can move in that direction, e.g. downhill towards the minimum value.



From all the predictions above, Random Forest and Gradient Boosting Algorithms work better on the data.

Deciding Metric

For evaluating the model, I have chosen r2 score, MAE, MSE. But I generally like to prefer r2 score itself as it gives us accuracy directly and we can just derive error by subtracting from 1. For MAE, MSE we can try to make it as minimum as possible but we also have to check the basic worst case error and compare which we can save time by using r2 score only.

# Final predictions

IN THIS SECTION:

* Predicting using Random Forest and Gradient Boosting
* Writing the predictions into a new file

# 

Report Notebook

‘

Tab 5

Predicting using Random Forest and Gradient Boosting

Predicting using Random Forest and Gradient boosting gave similar results so using Random Forest Algorithm

