**Cab Fare Prediction**

**In**

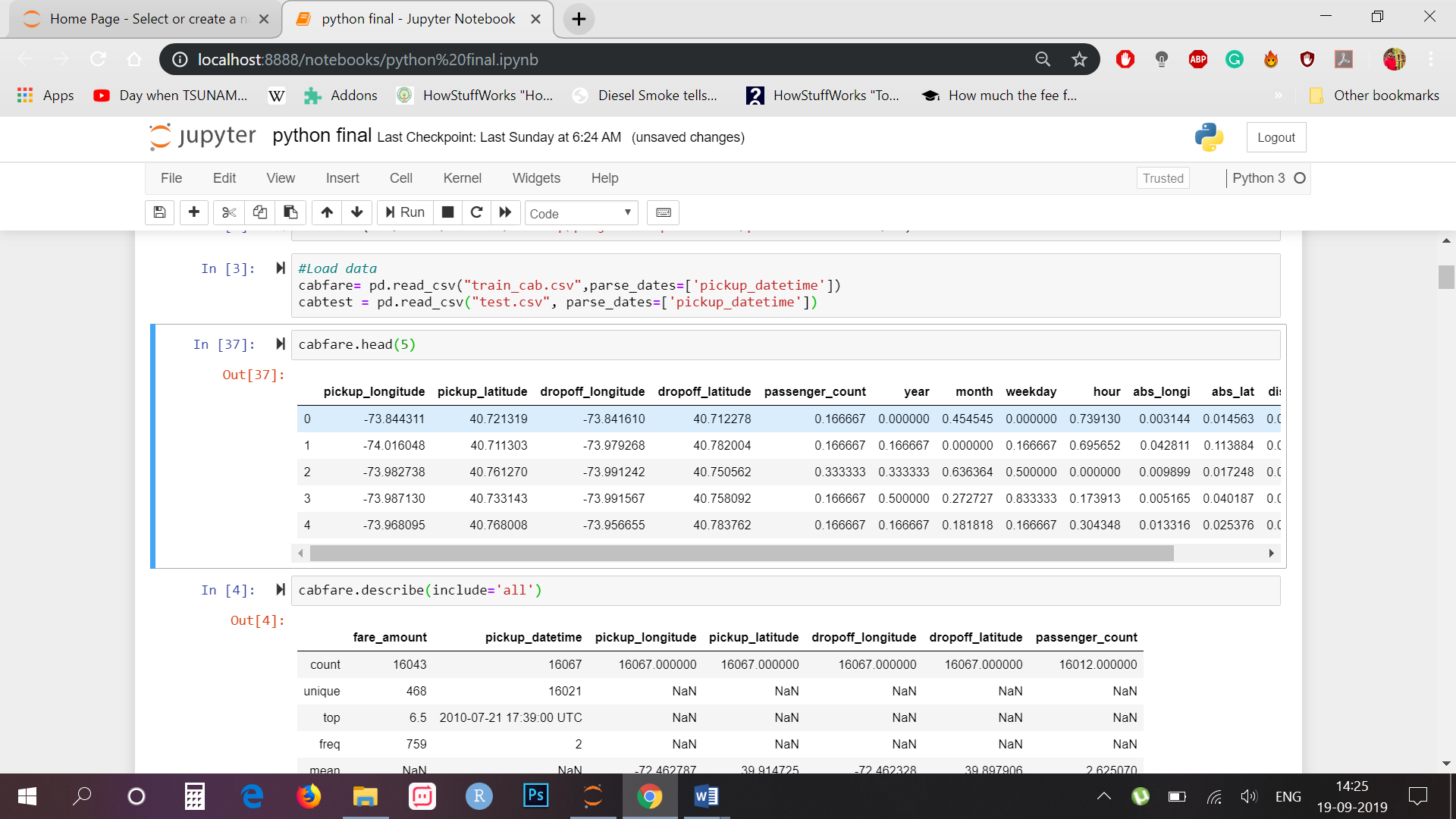
**Python**

A sridhar

## *Introduction*

### *Problem Statement:* The objective of this Project is to predict the fare of the Cab rental in the city. This Fare prediction takes distance, date/time and other factors in account from historical data which was gathered from the pilot project for the same. We would be building a model that can successfully predict the fare of rentals on relevant factors.

* Given with train\_cab data set and test dataset.
* *Data:* As the dataset given has dependent and independent values, it will come under supervise Machine learning. Our task is to build Regression models which will help us predicting the fare for our cab which depends on the factors provided. Given below is a sample of the data set that we are using for our prediction.
* This dataset contains 07 variables in which 6 are independent variables and 1 (Fare\_amount) is dependent variable.



|  |  |
| --- | --- |
| **Variable** | **Explanation** |
| fare\_amount | Float amount of the ride. |
| 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. |

### 

### *Exploring the data :*-

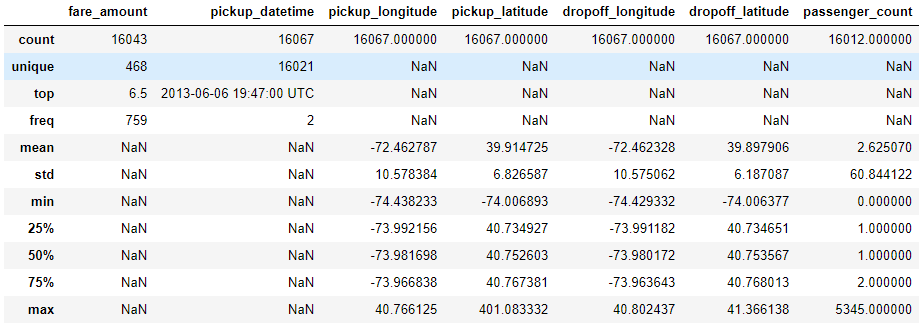
### Before we proceeding to create our model on top of the provided data. It is necessary to do Exploratory Data Analysis. EDA is very first and necessary step to take before proceeding further. As the result depends on the data, EDA makes sure the quality of input data is high which will lead to high quality results. We can perform EDA as follows:

#### **Variable Identification:** In Order to understand the data, we need to first, Identifying Predictor (Input) and Target (output) variables. Then, Identifying the data type and category of the variables

#### Types of Variable: Our Target Variable is ‘fare\_amount’, and Predictor variables are (pickup\_datetime,pickup\_longitude,pickup\_latitude,dropoff\_longitude,dropoff\_latitude,passenger\_count) .

#### Data Types: Character(passenger\_count), Numeric(fare\_amount) ,factor(\pickup\_longitude,pickup\_latitude,dropoff\_longitude,dropoff\_latitude), datetime(pickup\_datetime )

Before I made passanger\_count to category , our data looked like below



After I made passanger\_count to category our data looks like

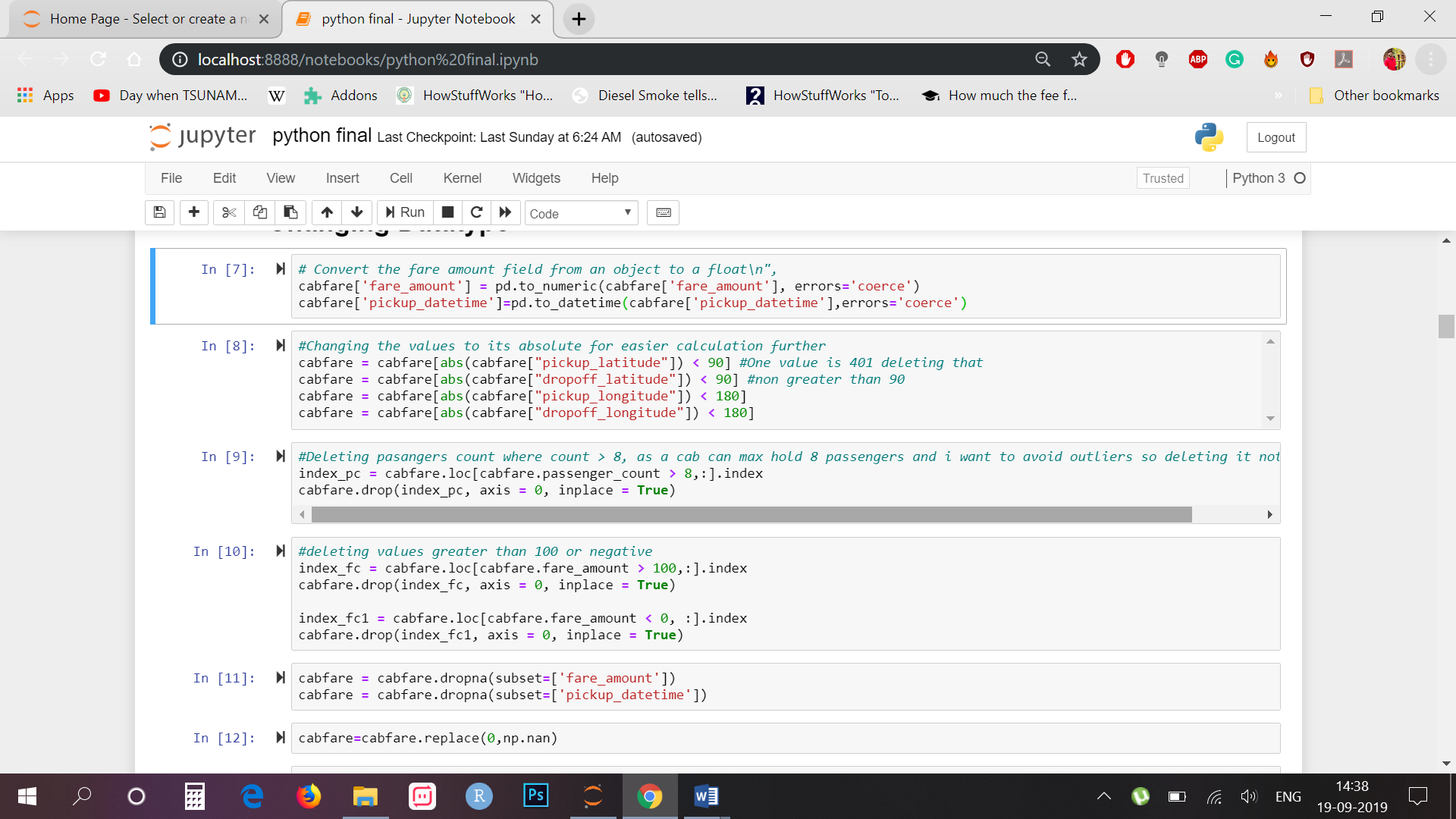
#### 

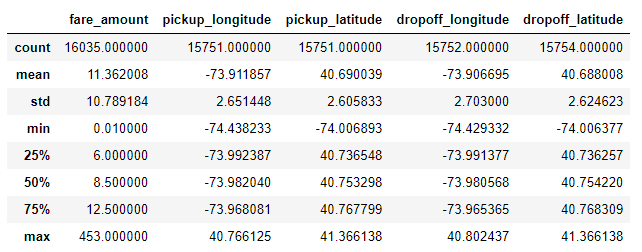
#### We have converted the data as per our requirement

#### NOTE: I made passenger\_count to category after making it to Int and cleaning it.

Variable Categories: Categorical (passenger\_count), Continuous (pickup\_datetime,pickup\_longitude,pickup\_latitude,dropoff\_longitude,dropoff\_latitude).

##### **Data Cleaning:** We can clearly observe from Summary in R and Describe Function in Python that Passenger counts of maximum values is very high and Pickup\drop off longitude and latitude is not under 90 and 180 which is not possible as per geographical information. Passenger count is too high as we know cab can accommodate max 7 passenger if consider its SUV. Distance is also extremely high as per regular cab which roams within the city.



* Keeping fare\_amount (as during visualization I realized the distribution of data is under 60 and after that just tail is stretched), Pickup\drop off longitude and latitude under under 90 and 180, passenger\_count under 8. I have not dropped observation for not but just imputed it with NA. After cleaning our data looks like
* 

#### **Feature Engineering:**

#### 1) I have split pick\_up datetime into hours, day, month and year and dropped the main variable pickup\_datetime. This will help us understanding our data more efficiently.

#### 2) I have calculated distance based on our Pickup\drop off longitude and latitude using great\_circle\_distance function.

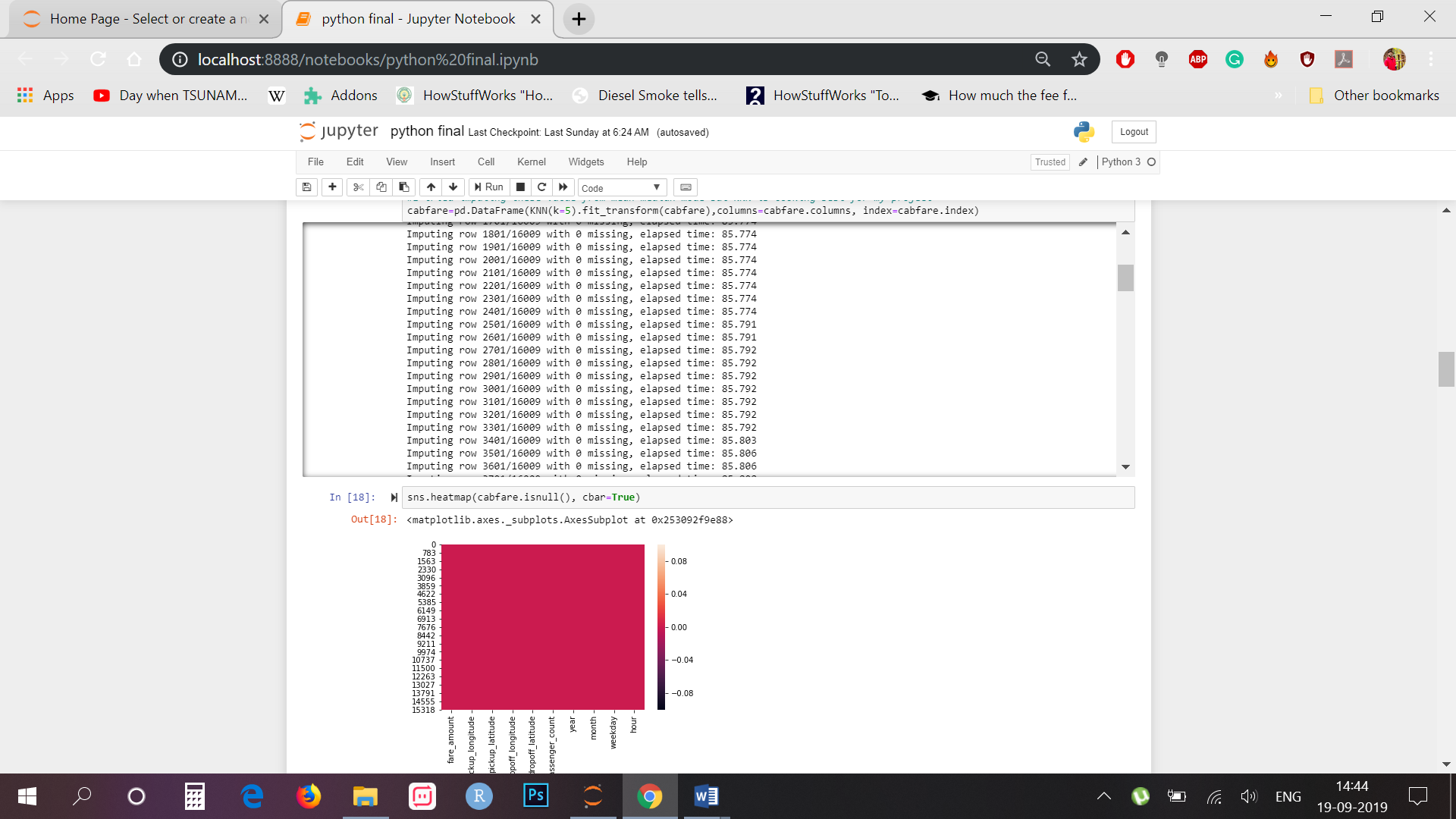
#### After Feature engineering our data looks like this

#### 

#### **Missing values treatment:** Missing values occur when no data value is stored for the variable in an observation. Missing values are a common occurrence, and you need to have a strategy for treating them. A missing value can signify a number of different things in your data. Perhaps the data was not available or not applicable or the event did not happen. It could be that the person who entered the data did not know the right value, or missed filling in. Typically, ignore the missing values, or exclude any records containing missing values, or replace missing values with the mean, or infer missing values from existing values. We check for missing values in our data and came to know we have missing data in almost every variable.

#### 

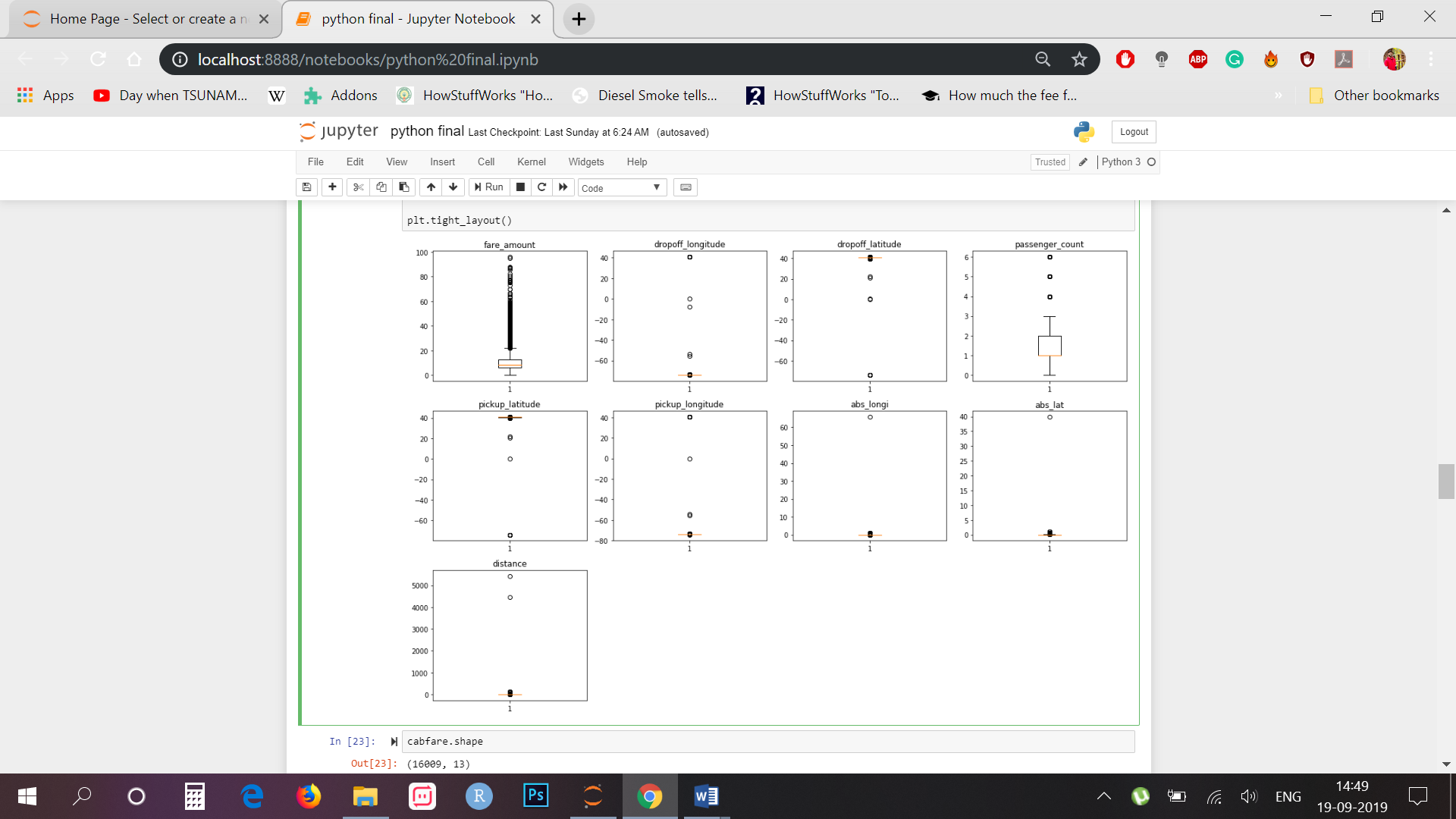
I have used median method with Missing Values in R coding since distribution of missing values are same across the different variable and as a another try imputed using, KNN and other fitting formulas in Python.



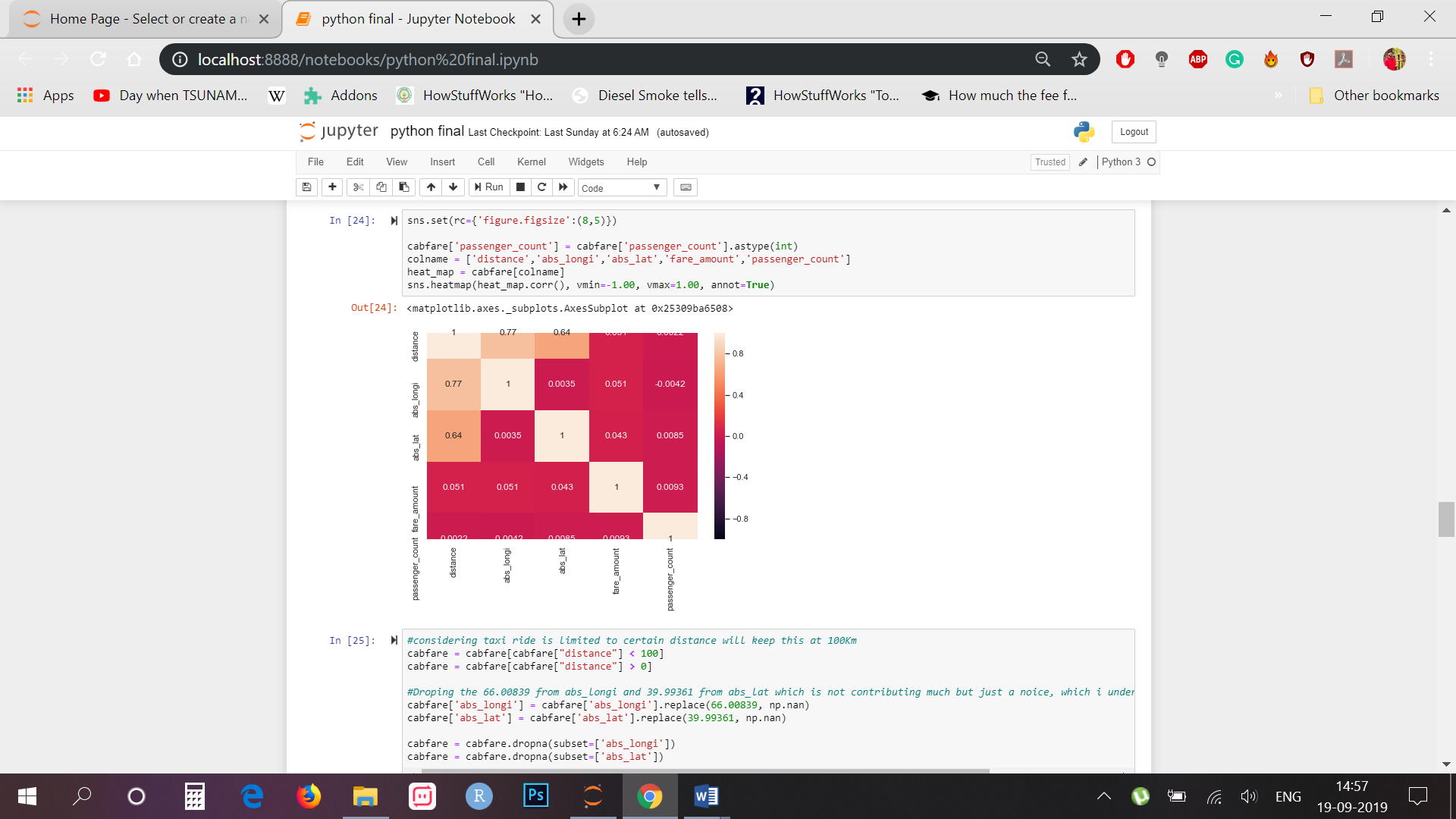
#### **Visualization:**

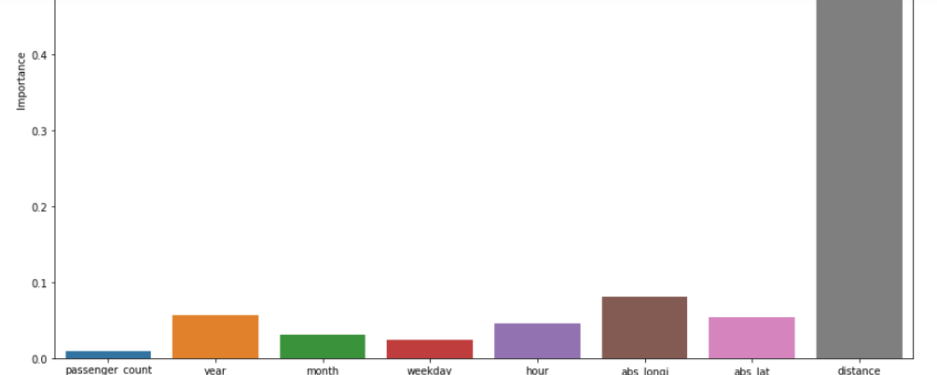
#### **E**xploring Variables one by one to understand central tendency, spread of the variable, distribution of each category, association and disassociation between variables at a predefined significance level.

* **Outlier treatment:** An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. Outliers can drastically change the results of the data analysis and statistical modelling. There are numerous unfavourable impacts of outliers in the data set. It increases the error variance and reduces the power of statistical tests. If the outliers are non-randomly distributed, they can decrease normality. They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions.
* By using Bloxplot methods we can easily find outliers in our data.



* In our data, we can clearly observe from Summary in R and Describe Function in Python that Passenger counts of maximum values is very high and Pickup\drop off longitude and latitude is not under 90 and 180, and also fare\_amount. I have also minimized Distance to 100km.
* **Feature Selection:** We have converted Pickup \ drop off latitude and longitude as absolute location points and from these variables we have extracted the total distance travelled. From Pick date and Time extracted Year, Month, day, Hours. Here is some graphical representation of the same **Correlation Analysis :** We make heat map to understand the co relation of continuous variable. A heatmap is a graphical representation of data where the individual values are represented as colours. Here each numerical variable’s correlation is map.



* Here , we can see the the importance of distance is extremely high. So, instead of deleting all other variables , I am going to create out model with two inputs one with distance only and one with all the variable including distance.
* **Feature Importance:** The concept is really straightforward: We measure the importance of a feature by calculating the increase in the model’s prediction error after permuting the feature. A feature is “important” if shuffling its values increases the model error, because in this case the model relied on the feature for the prediction. A feature is “unimportant” if shuffling its values leaves the model error unchanged, because in this case the model ignored the feature for the prediction. 

#### **Feature Scaling:** Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization.

#### Normalization also called Min-Max scaling. It is the process of reducing unwanted variation either within or between variables. Normalization brings all of the variables into proportion with one another. It transforms data into a range between 0 and 1.

#### All our continuous variables are already normalized except the target and the distance which we took out from logi/lati variable which we prefer not to scale because its variation is spread quite widely and after scaling, the difference between the number is diminishing.

#### 

#### 

#### **Checking for Skewness:**

#### Skewness is usually described as a measure of a dataset’s symmetry – or lack of symmetry.

#### A perfectly symmetrical data set will have a skewness of 0.

#### If the skewness is between -0.5 and 0.5, the data are fairly symmetrical.

#### If the skewness is between -1 and – 0.5 or between 0.5 and 1, the data are moderately skewed.

#### If the skewness is less than -1 or greater than 1, the data are highly skeweness.

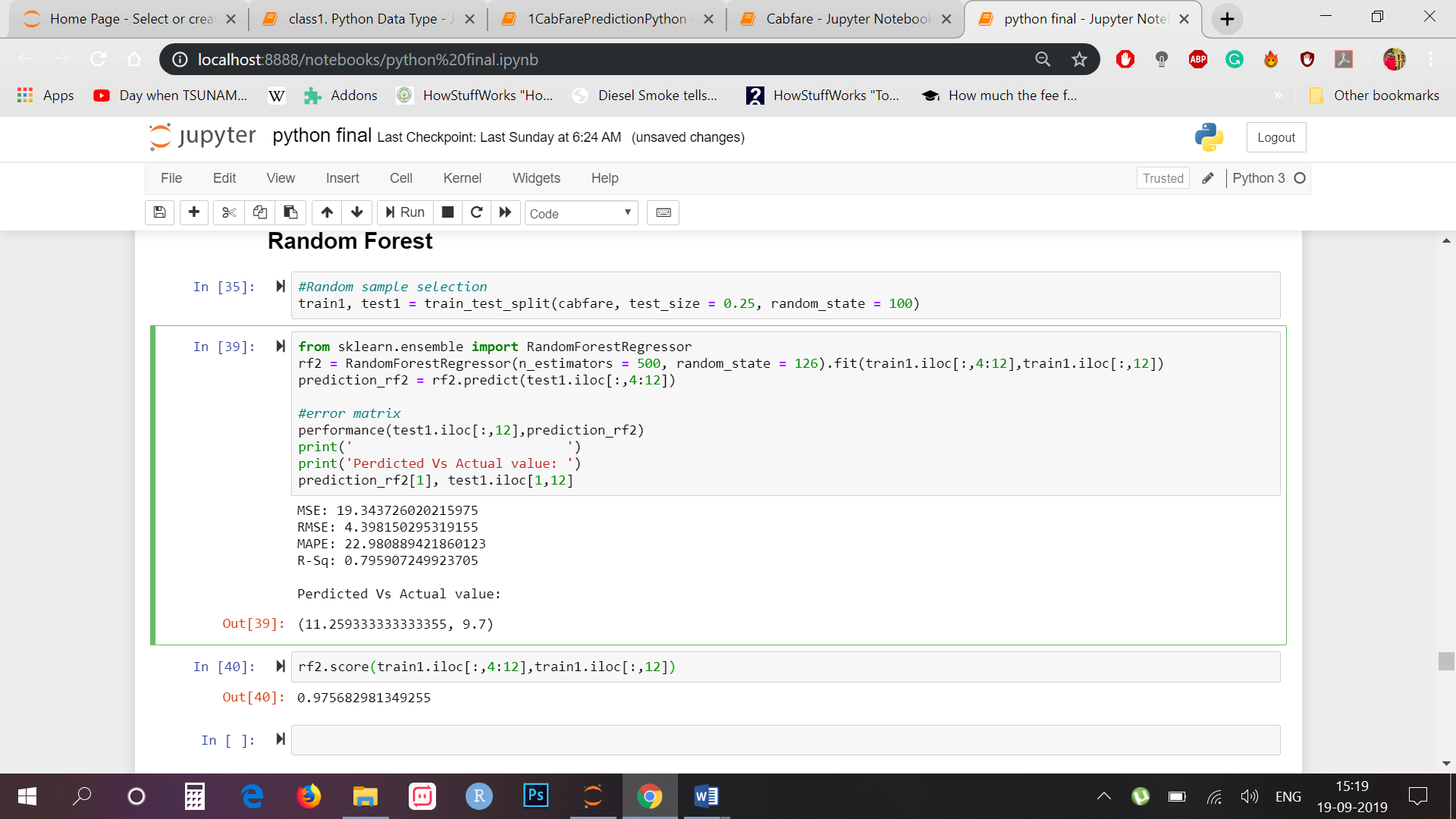
### Model Selection: For modelling, we are going to use some famous models to our data-set and will conclude the result according to it.

##### ***Decision Tree:*** Decision tree is a rule. Each branch connects nodes with “and” and multiple branches are connected by “or”. It can be used for classification and regression. It is a supervised machine learning algorithm. Accept continuous and categorical variables as independent variables. Extremely easy to understand by the business users. Split of decision tree is seen in the below tree.

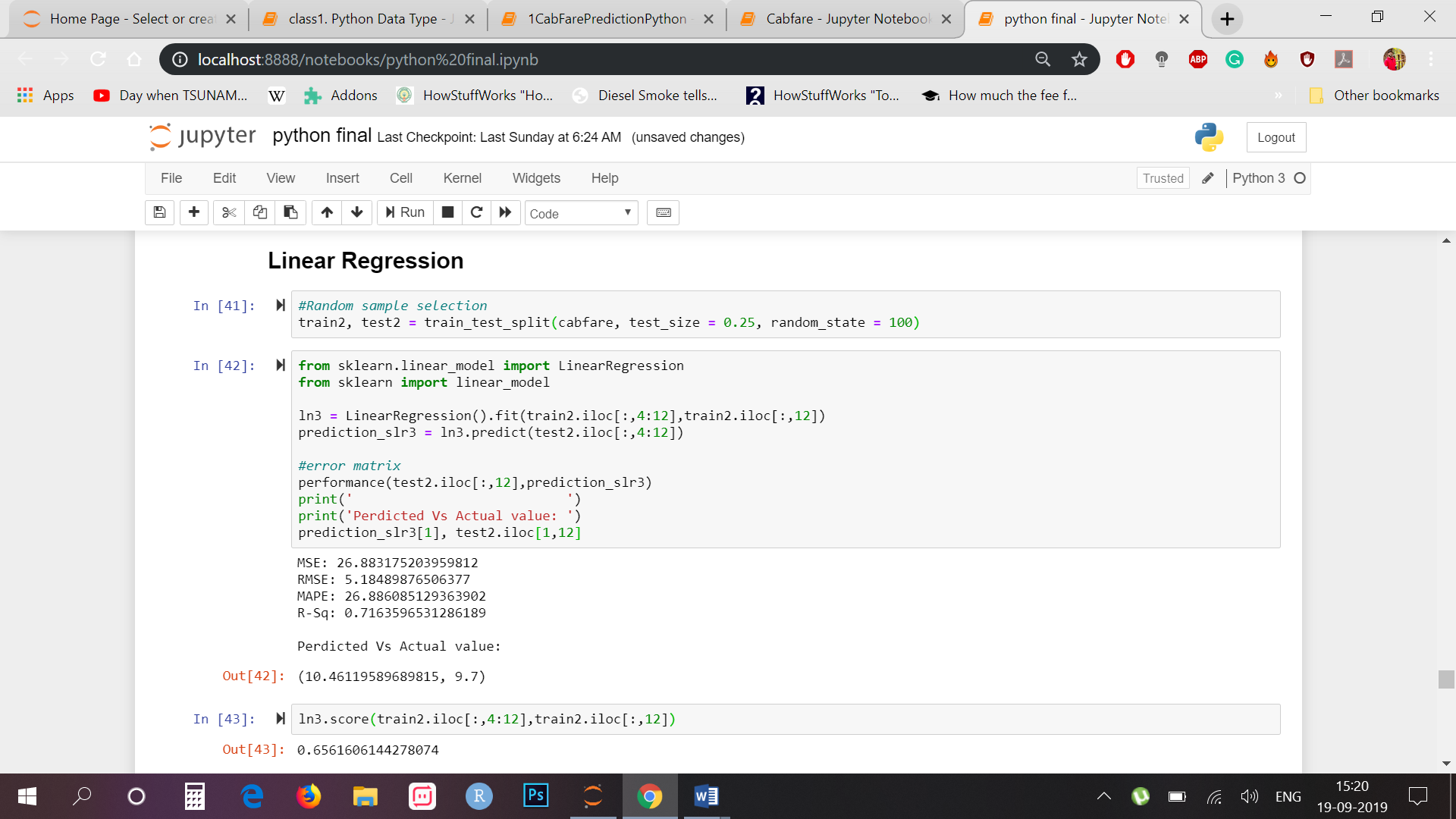
##### 

##### 

##### ***Random Forest:*** Random Forest or decision tree forests are an ensemble learning method for classification, regression and other tasks. It consists of an arbitrary number of simple trees, which are used to determine the final outcome. In the regression problem, their responses are averaged to obtain an estimate of the dependent variable. Using tree ensembles can lead to significant improvement in prediction accuracy (i.e., better ability to predict new data cases). The goal of using a large number of trees is to train enough that each feature has a chance to appear in several model--> As we increase the number of trees the error count decrease until a point (500 trees) and then becomes constant.



##### ***Linear Regression: li***near regression is the most basic type of regression and commonly used predictive analysis. Linear regression is an approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.



## *Conclusion*

* ***Model Evaluation:*** Model evaluation is done on basis of evaluation metrics or error metrics, Evaluation metrics explain the performance of a model.
* An important aspect of evaluation metrics is their capability to discriminate among model results. Simply, building a predictive model is not our motive. But, creating and selecting a model which gives high accuracy on out of sample data.
* Hence, it is crucial to check accuracy or other metric of the model prior to computing predicted values. In our data as we applied regression models we have error metrics like Mean square error (MSE), MAPE, Root mean square error (RMSE), Mean absolute error (MAE).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Language/Model |  | Python |  |  |
| MODELS | MSE | RMSE | MAPE | R-SQ |
| Decision Tree | 22.59 | 4.71 | 24.4 | 0.761 |
| Random Forest | 19.34 | 4.39 | 22.9 | 0.795 |
| Linear regression | 26.6 | 5.15 | 26.88 | 0.714 |

### We can see that all models perform comparatively on average and therefore we select random forest classifier models for better prediction.

### From the above plots of Actual Vs Predicted values, we can infer that values of Random forest falls on straight line indicating random forest fits better than the other three models.

### Also amongst the three models, Random forest has best R-sq. (Coef. of determination). Hence we’ll fix Random Forest as our model.

### Applying the prediction model on test data, we get below prediction distribution.

### Applying the prediction model on test data, we get below prediction distribution.

### 