**FIT5196**

# Methodology

To analyze this data, following steps are followed –

Step 1: Building Python Environment

First is knowledge of Python's underlying data structures, particularly records and word references. For more data, look at Lists and Tuples in Python and Dictionaries in Python.

The second thing you'll require is a working Python climate. You can track within any terminal that has Python 3 introduced. On the off chance that you need to see more pleasant yield, particularly for the provided dataset to work with, at that point you should run the models in a Jupyter note pad.

Step 2: Use of Pythion Library

Using the library, I have looked upon the data provided by university. I utilize the Python worked in work len() to decide the number of columns. You additionally utilize the shape characteristic of the DataFrame to see its dimensionality. The outcome is a tuple containing the number of lines and segments.

For the missing data analysis of the provided dataset, initially the dataset includes,

order\_id 0

customer\_id 0

date 0

nearest\_warehouse 55

shopping\_cart 0

order\_price 15

delivery\_charges 40

customer\_lat 0

customer\_long 0

coupon\_discount 0

order\_total 15

season 0

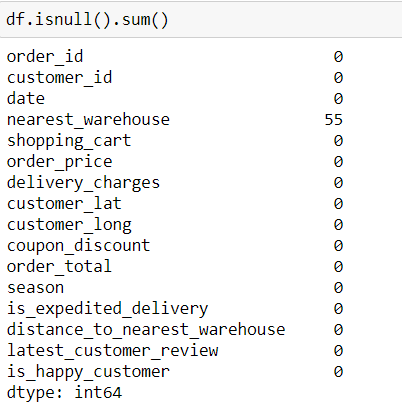
is\_expedited\_delivery 0

distance\_to\_nearest\_warehouse 31

latest\_customer\_review 0

is\_happy\_customer 40

For the dataset, we replaced the null/NaN values using the mean values. As the nearest datahouse is not numeric data, thus it was left as usual. Finally the resultant dataset had the following shape,



Step 3: Displaying Data Types

The initial phase in becoming more acquainted with your information is to find the diverse information types it contains. While I can place anything into a rundown, the sections of a DataFrame contain estimations of a particular information type.

Step 4: Exploring basics statistics

Since you've seen what information types are in your dataset, it's an ideal opportunity to get an outline of the qualities every section contains. You can do this with .desribe(): .describe() just breaks down numeric segments naturally. However you can give other information types on the off chance that you utilize the incorporate boundary.

For the outlier detection, the sns package is used and it depicted the outlier threshold values which were then used in order to remove from the dataframe as depicted below;

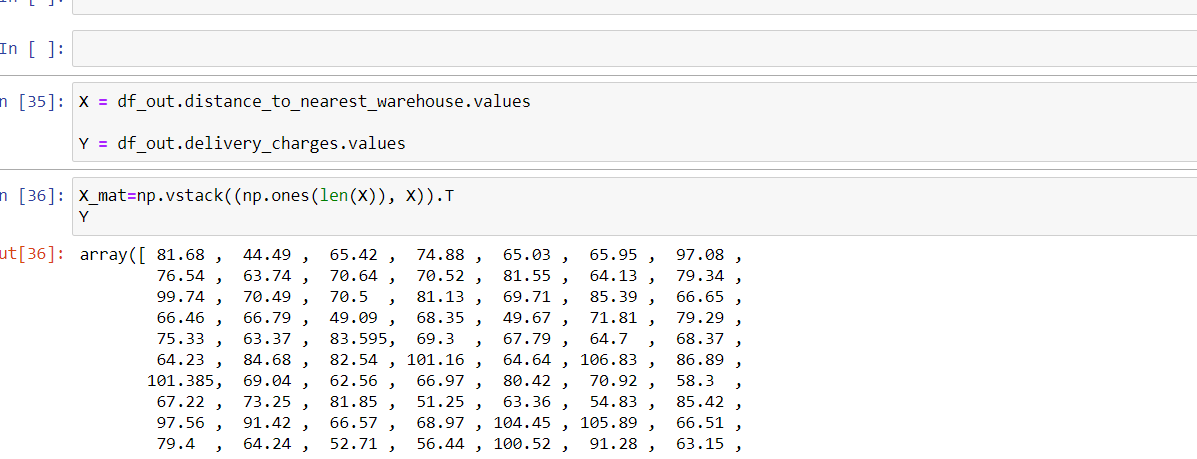


From the above image, it can be stated that , there are outliers in both sides of box plot, thus we considered min threshold as 40 and upper threshold value in order to remove the outliers from the delivery charges from the provided dataset.

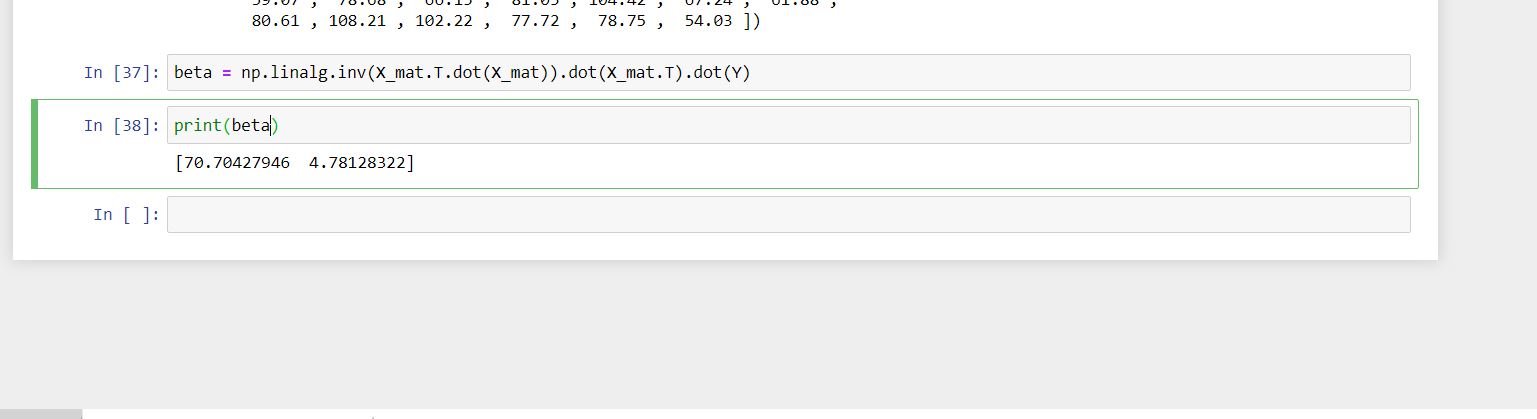
Step 5: Exploring data

EDA (Exploratory Data Analysis) can assist you in responding to inquiries concerning your dataset. Here, the record is a rundown of city names spoke to by strings. You may have seen that Python word references use string files also, and this is a helpful relationship to remember! You can utilize the code obstructs above to recognize two sorts of Series. While Pandas expands on NumPy, a critical distinction is in their ordering. Much the same as a NumPy exhibit, a Pandas Series likewise has a whole number record that is indeed characterized. This particular list demonstrates the component's situation in the Series.

From the dataset, the X,Y are selected in order develop a model using the numpy.linalg function;



Finally, the solution for the linear model generated values for X, Y as depicted in the given below image;



# Documentation

I spend a lot of time in cleaning datasets and getting them down to a structure with which they can work. A ton of information researchers contends that the underlying strides of acquiring and cleaning information comprise 80% of the work.

Subsequently, suppose you are merely venturing into this field or intending to venture into this field. In that case, it is critical to have the option to manage untidy information, regardless of whether that implies missing qualities, conflicting arranging, contorted records, or silly anomalies.

In this instructional exercise, we'll influence Python's Pandas and NumPy libraries to clean information.

I have covered the following:

* Dropping superfluous sections in a DataFrame
* Changing the list of a DataFrame
* Utilizing .str() techniques to clean segments
* I have utilized the DataFrame.applymap() capacity to clean the whole dataset, component astute.
* Renaming sections to a more unmistakable arrangement of marks
* Avoiding superfluous columns in a CSV record