IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: **Mahindu Bandaranayake**
Student Reference Number: **10749841**

Module Code: **PUSL 3121**	Module Name: **Big Data Analytics**
Coursework Title:	
Deadline Date: **2nd May 2023**	Member of staff responsible for coursework: **Mr. Pramudya Thilakaratne**

Programme:
BSc. (Hons) Computer Science University of Plymouth

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

Mahindu Bandaranayake *- 10749841*
Madapathage Madushan *– 10749856*
Maharu Balapitiya *– 10749837*
Pamunuwe Jayalath *– 10749858*
Ramanayake Ramanayake *- 10749834*
Jathusnanan Nadarasa *- 10749857*

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group:

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed :

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software……………………………………………………………

Overall mark _____% **Assessors Initials ______** **Date__25/04/2023_______**

Table of Content

Introduction

Big Data concept is widely used in 21st century, creating pathways and solutions to huge amount of ***structured*** and ***unstructured*** sets of data. The umbrella term is containing 4 other dimensions giving the overall meaning to it, which are known by the names; ***Volume, Velocity, Veracity and Variety***. As mentioned in an article these stored processed datasets will be utilized to make business decisions in the present and for the future as well. (Aljanabi and Yahya Almayali, 2019).

The fundamental step is to identify a dataset relating to the topic field where the analysis is based on.

Based on several articles these datasets might be containing missing values and outliers concerning the effectiveness and efficiency degradation of the machine learning models (Seliem and Seliem, 2022). Presence of missing values compromises the statistical enhancement to evaluate the dataset, while outliers will affect the estimation process in certain attributes either overestimating or underestimating on some occasions (Kwak and Kim, 2017).

Scenario 1

01) What are Missing/Null Values

Missing values are occurred when certain variables are not containing relevant values particular to the definition (Null values). Such instances can be caused by reckless data insertion, equipment malfunctions or due to lost files and many other reasons.

According to the previously cited articles, there are three main types of missing values pertaining to errors encountered as discussed above.

- Missing Completely at Random (MCAR)
- Missing at Random (MAR)
- Not Missing at Random (NMAR)

- ❖ MCAR's data can be among the variables but might not be related to each other
- ❖ MAR data are not distributed among the other variables, but they certainly do hold accountability for the variation of other parameters in the dataset
- ❖ NMAR data will be systematically differ from the analyzed values

(Kwak and Kim, 2017)

Effects of Missing Values

- Containing missing values will cause problems based on the type and will result in unorthodox biasing resulting in de-generalizing the work field.
- Null or zero values will be limit the precision and will be disruptive when models are being trained.
- Under specific circumstances ignoring missing values will result in vague answers indicating less accuracies in the study.

Ways of handling Missing Values

Acceptance

Based on the type, values can be ignored. If the missing values are related to MCAR or MAR it certainly can be ignored.

Deletion

The complete row can be eliminated if the type of missing value is relating MNAR or either the relevant data point

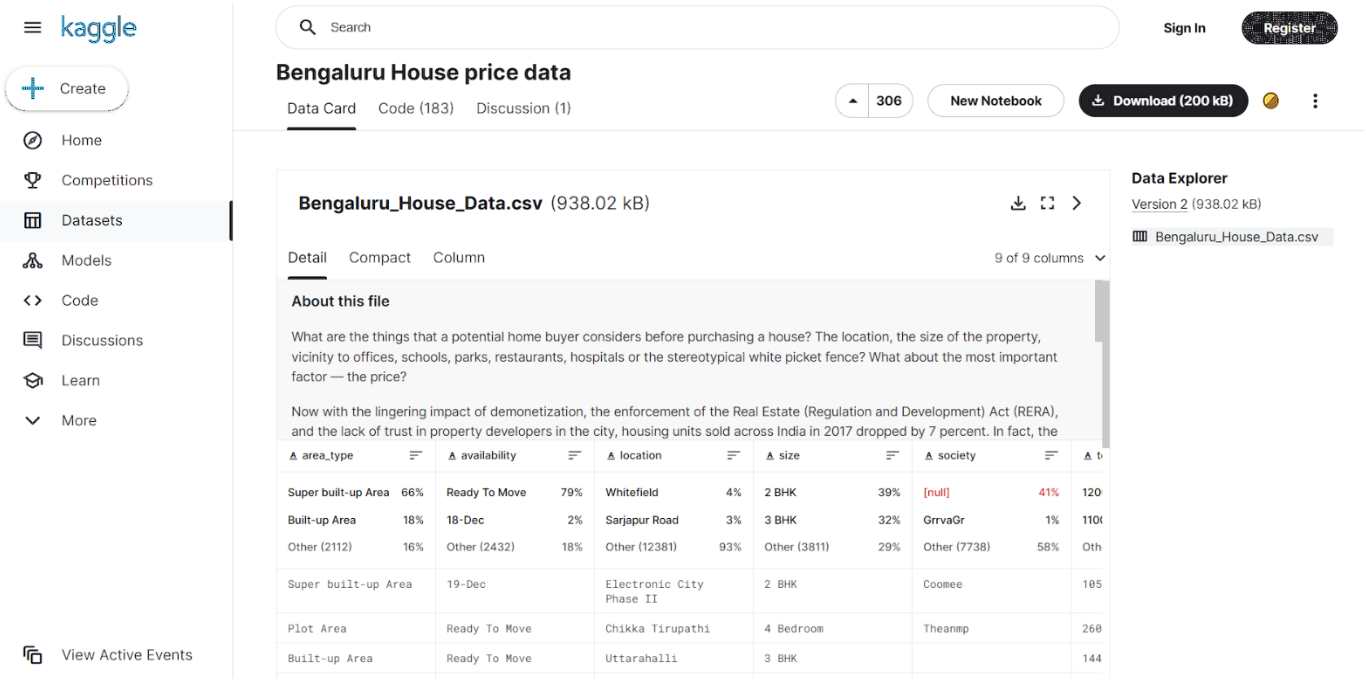
Imputation (Replacements)

Replacements can be done in several ways either by inserting a specific value from the dataset itself or inserting custom values relating to the attributes. In each instance values can be replaced row, column vice.

The above stated matters will be further discussed by analysing a dataset

Analyzes 1

A dataset containing more 1000 values (columns) were identified and downloaded from the Kaggle website in order to start up the proceedings.



Bengaluru House price data

Bengaluru_House_Data.csv (938.02 kB)

About this file

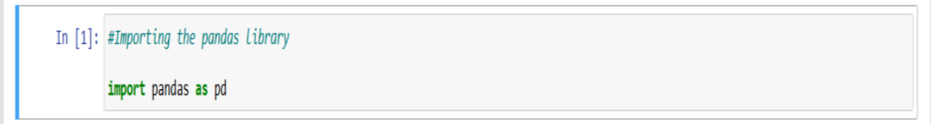
What are the things that a potential home buyer considers before purchasing a house? The location, the size of the property, vicinity to offices, schools, parks, restaurants, hospitals or the stereotypical white picket fence? What about the most important factor — the price?

Now with the lingering impact of demonetization, the enforcement of the Real Estate (Regulation and Development) Act (RERA), and the lack of trust in property developers in the city, housing units sold across India in 2017 dropped by 7 percent. In fact, the

area_type	availability	location	size	society	t
Super built-up Area 66%	Ready To Move 79%	Whitefield 4%	2 BHK 39%	[null] 41%	120
Built-up Area 18%	18-Dec 2%	Sarjapur Road 3%	3 BHK 32%	GrrvaGr 1%	110(
Other (2112) 16%	Other (2432) 18%	Other (12381) 93%	Other (3811) 29%	Other (7738) 58%	Oth
Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	105
Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	260
Built-up Area	Ready To Move	Uttarahalli	3 BHK		144

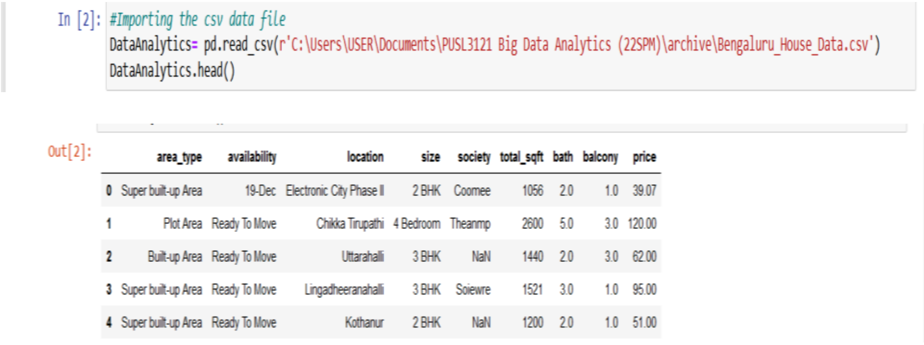
❖ The analyzes will be conducted in a python integrated environment (Jupyter Notebook) IDE.

Step 1: Importing necessary libraries relevant to data analysis



```
In [1]: #Importing the pandas library

        import pandas as pd
```

Step 2 : Loading the specific dataset using python keywords for verification (optional)

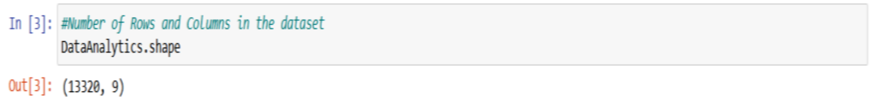


```
In [2]: #Importing the csv data file
        DataAnalytics= pd.read_csv(r'C:\Users\USER\Documents\PUSL3121 Big Data Analytics (22SPM)\archive\Bengaluru_House_Data.csv')
        DataAnalytics.head()
```

Out[2]:

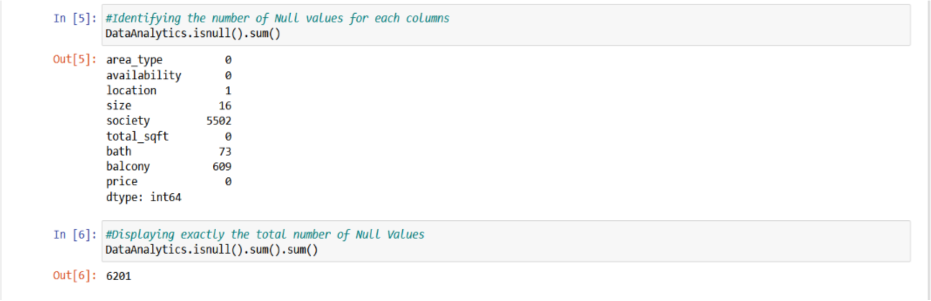
	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

Step 3 : Identifying the number of rows and columns in the dataset



```
In [3]: #Number of Rows and Columns in the dataset
        DataAnalytics.shape

Out[3]: (13320, 9)
```

Step 4 : Checking for Missing/Null values in the dataset using the sum function in python



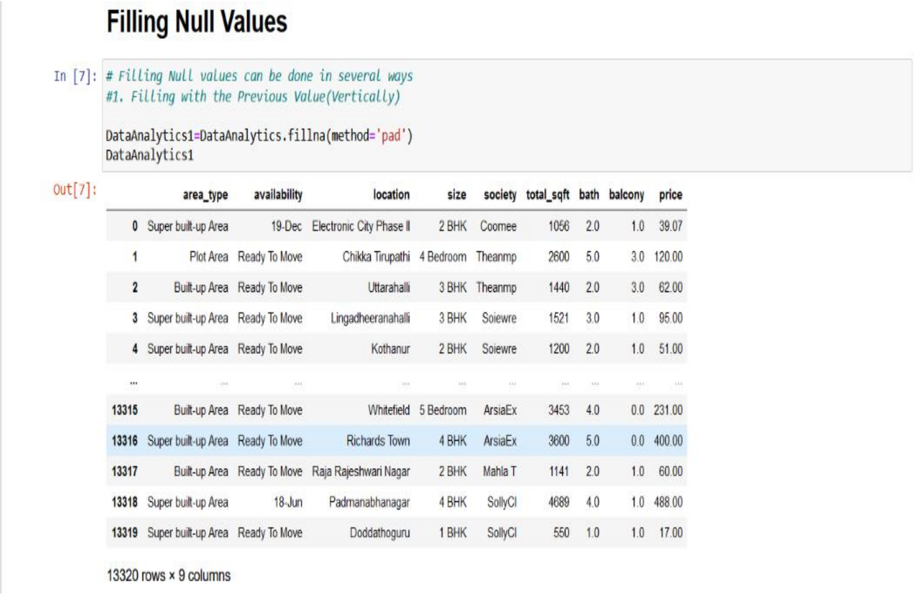
```
In [5]: #Identifying the number of Null values for each columns
        DataAnalytics.isnull().sum()

Out[5]: area_type          0
        availability       0
        location           1
        size              16
        society         5502
        total_sqft         0
        bath              73
        balcony          609
        price              0
        dtype: int64

In [6]: #Displaying exactly the total number of Null values
        DataAnalytics.isnull().sum().sum()

Out[6]: 6201
```

❖ Different ways were identified to solve the issue rising due to missing values, each cocept will be elaborated using python implementation.

Filling Missing/Null values

Previously it is mentioned that there can be several ways of filling null or missing values in a dataset. The following processes are conducted by filling a data point with the relevancy to the above and below values relative to the data point using the ***bfill*** and ***pad*** keywords, respectively.

Using pad keyword



Filling Null Values

```
In [7]: # Filling Null values can be done in several ways
        #1. Filling with the Previous Value(Vertically)

        DataAnalytics1=DataAnalytics.fillna(method='pad')
        DataAnalytics1
```

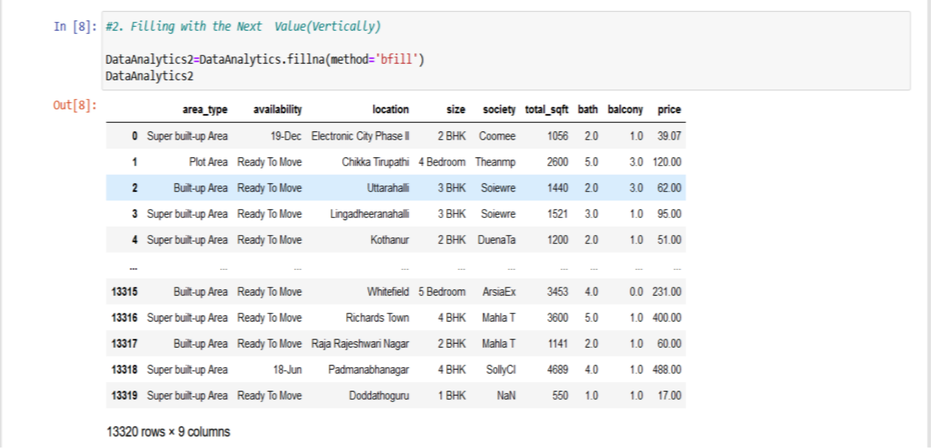
Out[7]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	Theanmp	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	Soiewre	1200	2.0	1.0	51.00
...	...	...	...	...	...	...	...	...	...
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.00
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	ArsiaEx	3600	5.0	0.0	400.00
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.00
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.00
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	SollyCl	550	1.0	1.0	17.00

13320 rows × 9 columns

Using this keyword, the missing/null values will be filled with the immediate next value below the missing/null data point per column. The filling will take place vertically moving downwards in the column.

Using bfill keyword

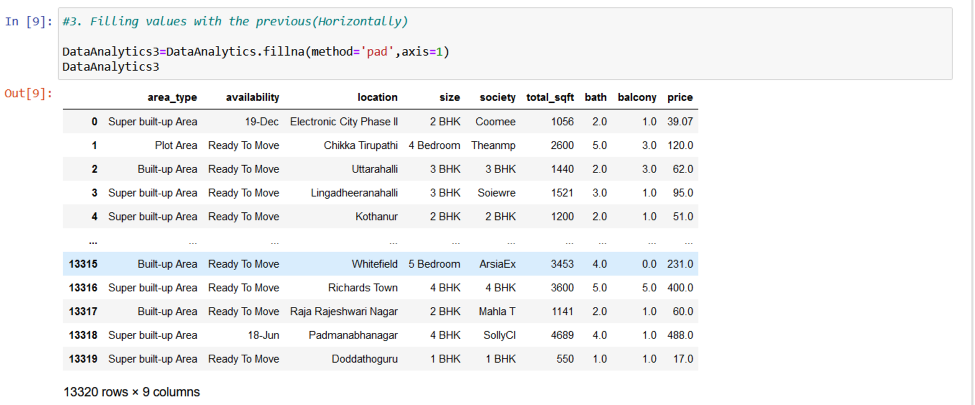


```
In [8]: #2. Filling with the Next  Value(Vertically)

        DataAnalytics2=DataAnalytics.fillna(method='bfill')
        DataAnalytics2
```

```
Out[8]:
                area_type  availability                location       size   society  total_sqft  bath  balcony   price
      0  Super built-up Area        19-Dec  Electronic City Phase II      2 BHK    Coomee        1056   2.0      1.0   39.07
      1            Plot Area  Ready To Move          Chikka Tirupathi  4 Bedroom   Theanmp        2600   5.0      3.0  120.00
      2        Built-up Area  Ready To Move                Uttarahalli      3 BHK   Soiewre        1440   2.0      3.0   62.00
      3  Super built-up Area  Ready To Move          Lingadheeranahalli      3 BHK   Soiewre        1521   3.0      1.0   95.00
      4  Super built-up Area  Ready To Move                  Kothanur      2 BHK   DuenaTa        1200   2.0      1.0   51.00
    ...                  ...            ...                       ...        ...       ...         ...   ...      ...     ...
  13315        Built-up Area  Ready To Move                Whitefield  5 Bedroom   ArsiaEx        3453   4.0      0.0  231.00
  13316  Super built-up Area  Ready To Move             Richards Town      4 BHK    Mahla T        3600   5.0      1.0  400.00
  13317        Built-up Area  Ready To Move     Raja Rajeshwari Nagar      2 BHK    Mahla T        1141   2.0      1.0   60.00
  13318  Super built-up Area        18-Jun            Padmanabhanagar      4 BHK    SollyCl        4689   4.0      1.0  488.00
  13319  Super built-up Area  Ready To Move               Doddathoguru      1 BHK       NaN         550   1.0      1.0   17.00

13320 rows × 9 columns
```

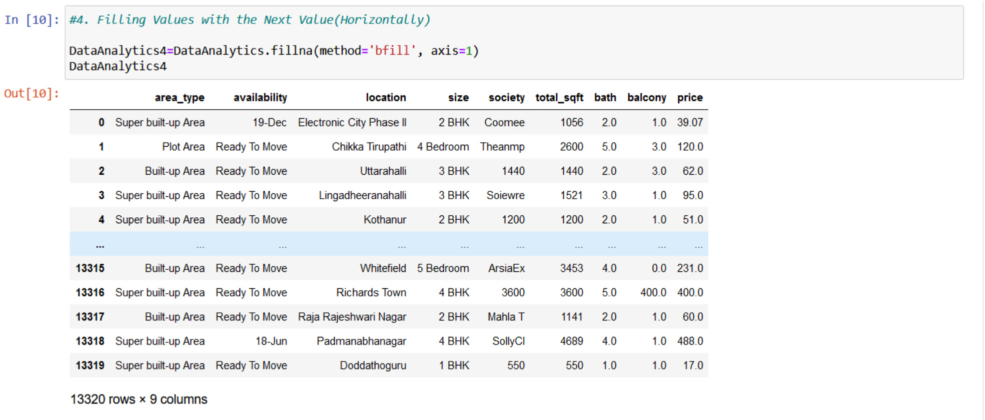
When this particular keyword is used, the missing/null values will be filled with the immediate next value above the missing/null data point per column. The filling will take place vertically moving upwards in the column.

❖ Same procedure can be done horizontally as well by changing the axis (defining a parameter using axis=1) by using the same keyword.



```
In [9]: #3. Filling values with the previous(Horizontally)

        DataAnalytics3=DataAnalytics.fillna(method='pad',axis=1)
        DataAnalytics3
```

```
Out[9]:
                area_type  availability                location       size   society  total_sqft  bath  balcony  price
      0  Super built-up Area        19-Dec  Electronic City Phase II      2 BHK    Coomee        1056   2.0      1.0  39.07
      1            Plot Area  Ready To Move          Chikka Tirupathi  4 Bedroom   Theanmp        2600   5.0      3.0  120.0
      2        Built-up Area  Ready To Move                Uttarahalli      3 BHK     3 BHK        1440   2.0      3.0   62.0
      3  Super built-up Area  Ready To Move          Lingadheeranahalli      3 BHK   Soiewre        1521   3.0      1.0   95.0
      4  Super built-up Area  Ready To Move                  Kothanur      2 BHK     2 BHK        1200   2.0      1.0   51.0
    ...                  ...            ...                       ...        ...       ...         ...   ...      ...    ...
  13315        Built-up Area  Ready To Move                Whitefield  5 Bedroom   ArsiaEx        3453   4.0      0.0  231.0
  13316  Super built-up Area  Ready To Move             Richards Town      4 BHK     4 BHK        3600   5.0      5.0  400.0
  13317        Built-up Area  Ready To Move     Raja Rajeshwari Nagar      2 BHK    Mahla T        1141   2.0      1.0   60.0
  13318  Super built-up Area        18-Jun            Padmanabhanagar      4 BHK    SollyCl        4689   4.0      1.0  488.0
  13319  Super built-up Area  Ready To Move               Doddathoguru      1 BHK     1 BHK         550   1.0      1.0   17.0

13320 rows × 9 columns
```

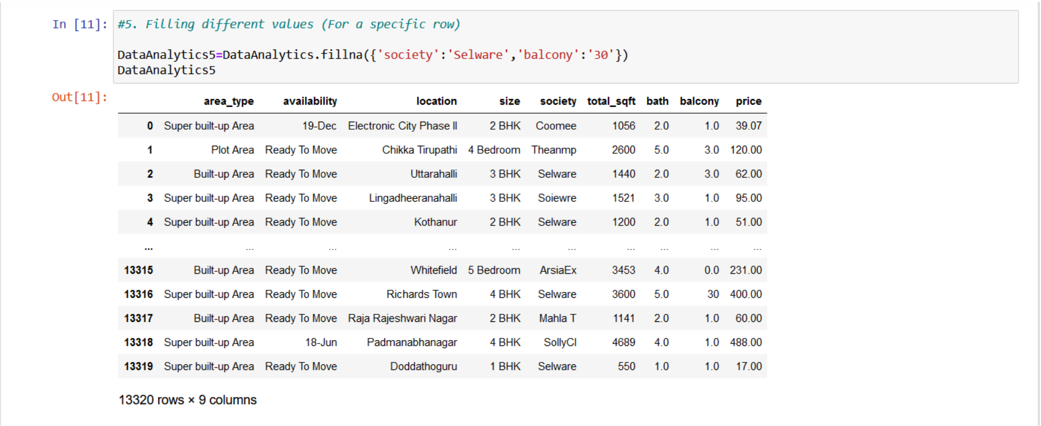


```
In [10]: #4. Filling Values with the Next Value(Horizontally)

         DataAnalytics4=DataAnalytics.fillna(method='bfill', axis=1)
         DataAnalytics4
```

```
Out[10]:
                area_type  availability                location       size   society  total_sqft  bath  balcony  price
      0  Super built-up Area        19-Dec  Electronic City Phase II      2 BHK    Coomee        1056   2.0      1.0  39.07
      1            Plot Area  Ready To Move          Chikka Tirupathi  4 Bedroom   Theanmp        2600   5.0      3.0  120.0
      2        Built-up Area  Ready To Move                Uttarahalli      3 BHK      1440        1440   2.0      3.0   62.0
      3  Super built-up Area  Ready To Move          Lingadheeranahalli      3 BHK   Soiewre        1521   3.0      1.0   95.0
      4  Super built-up Area  Ready To Move                  Kothanur      2 BHK      1200        1200   2.0      1.0   51.0
    ...                  ...            ...                       ...        ...       ...         ...   ...      ...    ...
  13315        Built-up Area  Ready To Move                Whitefield  5 Bedroom   ArsiaEx        3453   4.0      0.0  231.0
  13316  Super built-up Area  Ready To Move             Richards Town      4 BHK      3600        3600   5.0    400.0  400.0
  13317        Built-up Area  Ready To Move     Raja Rajeshwari Nagar      2 BHK    Mahla T        1141   2.0      1.0   60.0
  13318  Super built-up Area        18-Jun            Padmanabhanagar      4 BHK    SollyCl        4689   4.0      1.0  488.0
  13319  Super built-up Area  Ready To Move               Doddathoguru      1 BHK       550         550   1.0      1.0   17.0

13320 rows × 9 columns
```

Assigning custom values

Relevant values can be assigned by specifying the column name using the same ***fillna*** keyword as previously conducted in the above procedures.



```
In [11]: #5. Filling different values (For a specific row)

         DataAnalytics5=DataAnalytics.fillna({'society':'Selware','balcony':'30'})
         DataAnalytics5
```

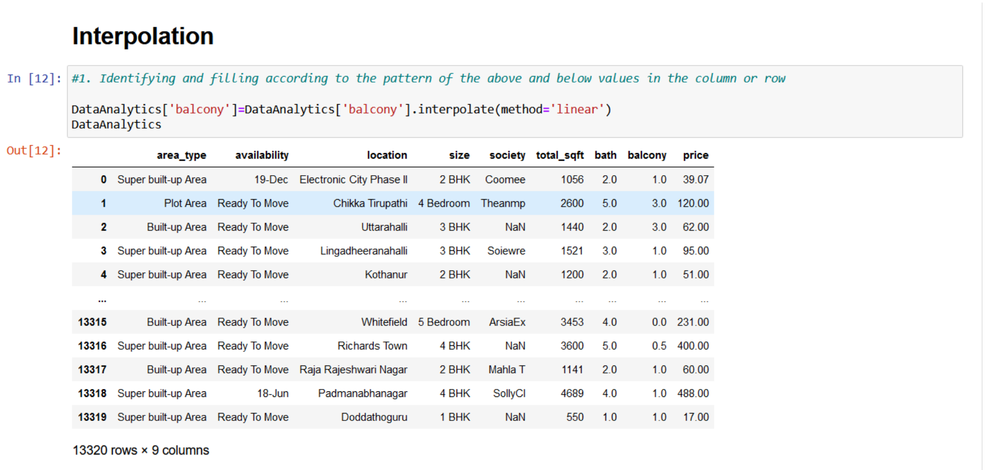
Out[11]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	Selware	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	Selware	1200	2.0	1.0	51.00
...	...	...	...	...	...	...	...	...	...
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.00
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	Selware	3600	5.0	30	400.00
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.00
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.00
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	Selware	550	1.0	1.0	17.00

13320 rows × 9 columns

Interpolation

The process which is conducted by this method is to identify the patterns between the known variables and define an estimated value for the unknown variable.



Interpolation

```
In [12]: #1. Identifying and filling according to the pattern of the above and below values in the column or row

         DataAnalytics['balcony']=DataAnalytics['balcony'].interpolate(method='linear')
         DataAnalytics
```

Out[12]:

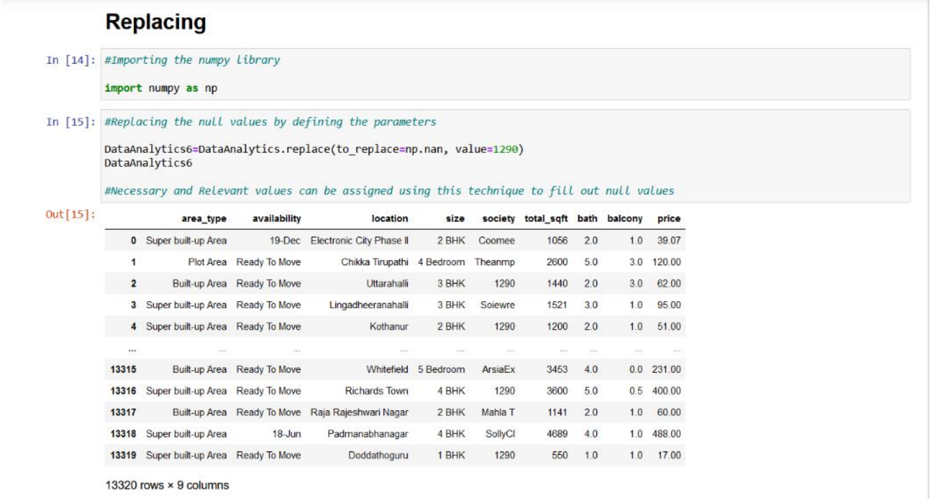
	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00
...	...	...	...	...	...	...	...	...	...
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.00
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	NaN	3600	5.0	0.5	400.00
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.00
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.00
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	NaN	550	1.0	1.0	17.00

13320 rows × 9 columns

Using the ***linear*** method, necessary column names can be defined, and the function would create an instance to identify the immediate values next to calculate the unknown variable by considering the known points.

Imputation (Replacements)

If an impure data point is found in the dataset another way of handling is to replace the incorrect detail values. The ***replace*** keyword and the identified impure data points are handled through the ***np*** keyword from the NumPy library.



Replacing

```
In [14]: #Importing the numpy library

         import numpy as np
```

```
In [15]: #Replacing the null values by defining the parameters

         DataAnalytics6=DataAnalytics.replace(to_replace=np.nan, value=1290)
         DataAnalytics6

         #Necessary and Relevant values can be assigned using this technique to fill out null values
```

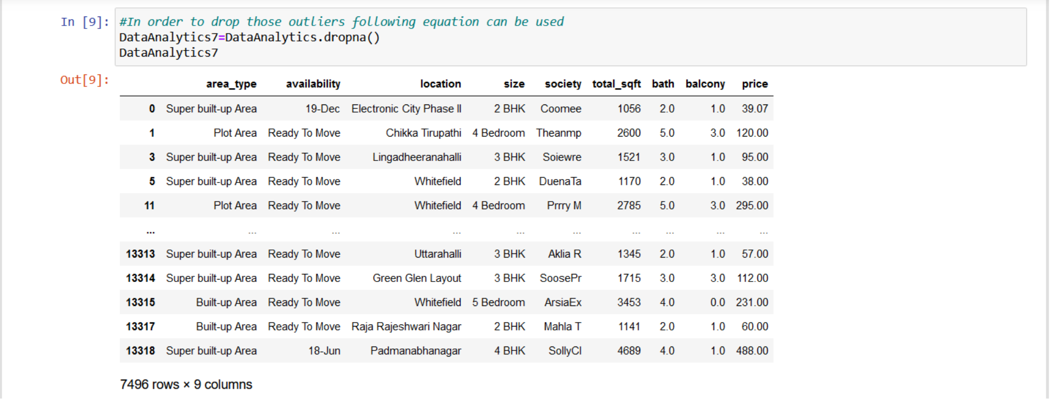
Out[15]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	1290	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	1290	1200	2.0	1.0	51.00
...	...	...	...	...	...	...	...	...	...
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.00
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	1290	3600	5.0	0.5	400.00
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.00
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.00
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	1290	550	1.0	1.0	17.00

13320 rows × 9 columns

Dropping

The fast and efficient way to getting rid of impure data is to drop the necessary data points from the dataset. Some disadvantages might be there such as causing the data values to drastically drop. But overall dropping missing/null value containing rows will increase the accuracy of the analysis.



```
In [9]: #In order to drop those outliers following equation can be used
        DataAnalytics7=DataAnalytics.dropna()
        DataAnalytics7
```

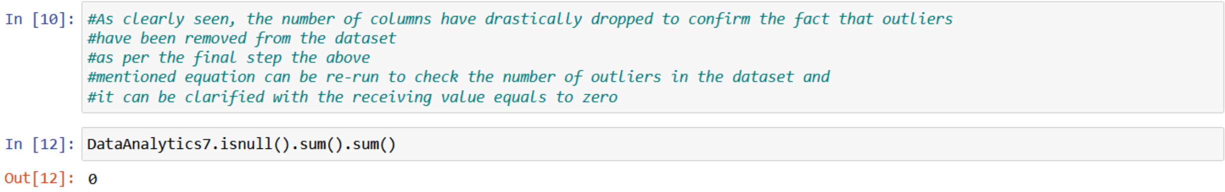
Out[9]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
5	Super built-up Area	Ready To Move	Whitefield	2 BHK	DuenaTa	1170	2.0	1.0	38.00
11	Plot Area	Ready To Move	Whitefield	4 Bedroom	Prrry M	2785	5.0	3.0	295.00
...	...	...	...	...	...	...	...	...	...
13313	Super built-up Area	Ready To Move	Uttarahalli	3 BHK	Aklia R	1345	2.0	1.0	57.00
13314	Super built-up Area	Ready To Move	Green Glen Layout	3 BHK	SoosePr	1715	3.0	3.0	112.00
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.00
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.00
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.00

7496 rows × 9 columns

As seen clearly when the ***dropna*** keyword is used and the code is run, the number of rows have drastically dropped proving the fact that all the null/missing values are removed from the dataset.

In order to confirm whether all the impure data is deleted from the dataset, previously mentioned null function can be used again.



```
In [10]: #As clearly seen, the number of columns have drastically dropped to confirm the fact that outliers
         #have been removed from the dataset
         #as per the final step the above
         #mentioned equation can be re-run to check the number of outliers in the dataset and
         #it can be clarified with the receiving value equals to zero

In [12]: DataAnalytics7.isnull().sum().sum()

Out[12]: 0
```

02) What are outliers

Outliers are considered as data points which exist outside the requirement boundaries in the dataset. Necessarily it does not mean that the data points are impure in fact data points are varied among the rest of the points.

Effects of Outliers

- Skewed data distribution will cause fluctuation in mean value calculations and vary the accuracy of prediction models.
- Presence of outliers will cause abnormal conditions when sampling process is calculated (Biased Sampling)

Ways of handling Outliers

There are four main procedures that needs to be conducted in order to identify outliers in a dataset and provide treatments for the cause.

- Sorting
- Data Visualization
- Z-score calculation
- Interquartile range value determination

Sorting Method

Basically, the process will be carried out until the data points are labelled as high and low meanwhile flagging any extreme data points. This simple procedure will be helpful to identify the unorthodox conditions in the dataset before calculations are done.

Data Visualizing Method

Using python in-built libraries, minimum/maximum range and median values of data points will be visualized for better understanding. Boxplot, Histplot and WhiskerPlot are mostly used to plot the data points in a diagrammatic view.

Z-score calculation

By calculating the Z-score values, relevant boundaries can be set in order to identify the common values for the data points and recognize the points containing more than -3 and +3 values, which can be concluded as outliers in the dataset.

Interquartile range value determination

Using this procedure, the middle range values of the dataset can be determined, and boundaries(fences) will be set to separate outliers from the other data points, where the outliers lie besides the boundaries, allowing the analyst to study the dataset in an easier manner.

[(Kwak and Kim, 2017),(Zhang *et al.*, 2019)].

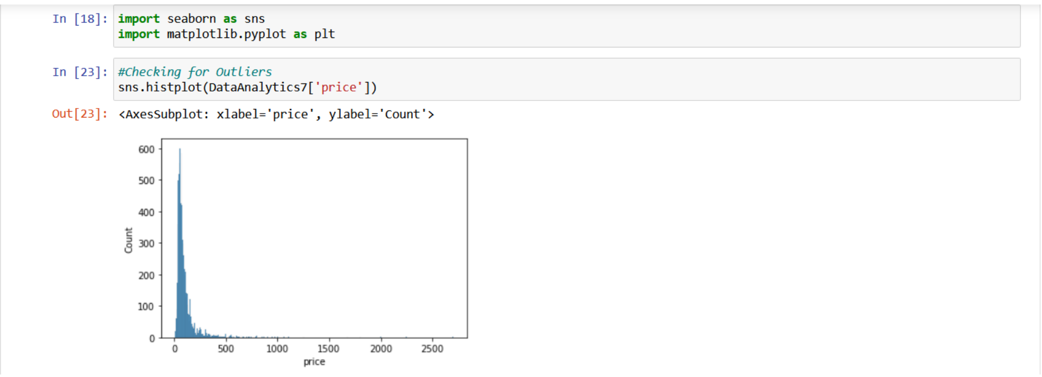
Analyzes 2

- The analysis will be based on the continuation of the above-mentioned dataset, using two of the main procedures mentioned above.
    - **Data Visualization**
    - **Z-score calculation**

The first step of the process is to determine whether outliers are present in the dataset after, part of cleansing was done during the previous steps (Removing all Null/Missing Values). Several methods can be used to identify the abnormal behavior of the random points present in the dataset, out of which plotting can be done, which will be the efficient and the fastest way of identifying whether outliers are present in the dataset.

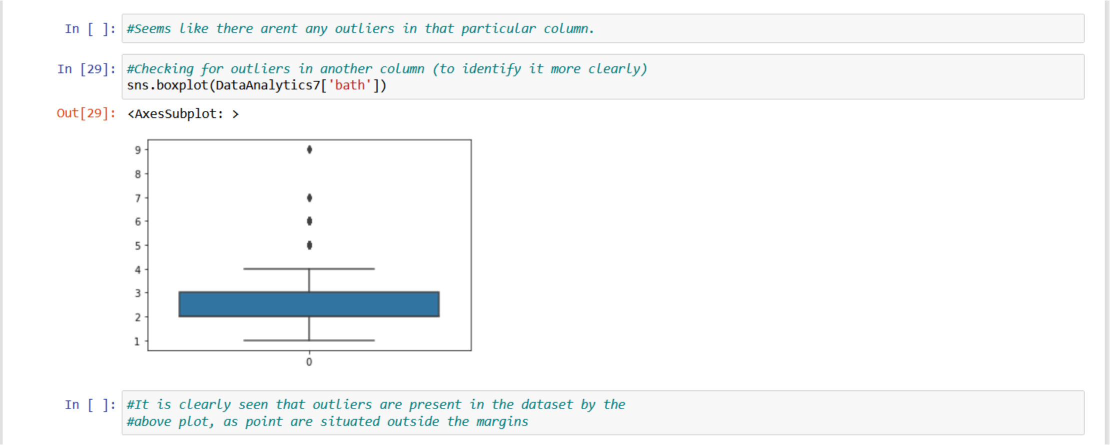
Since there are 9 columns in the dataset, each can be taken in to consideration by specifying the column name by defining it under the code implementation using the seaborn library.

Plotting using Histplot



```
In [18]: import seaborn as sns
         import matplotlib.pyplot as plt

In [23]: #Checking for Outliers
         sns.histplot(DataAnalytics7['price'])

Out[23]: <AxesSubplot: xlabel='price', ylabel='Count'>
```

Since there were no outliers identified in that particular column, another column will be analyzed and will continue to change the column name until specific requirements are met.



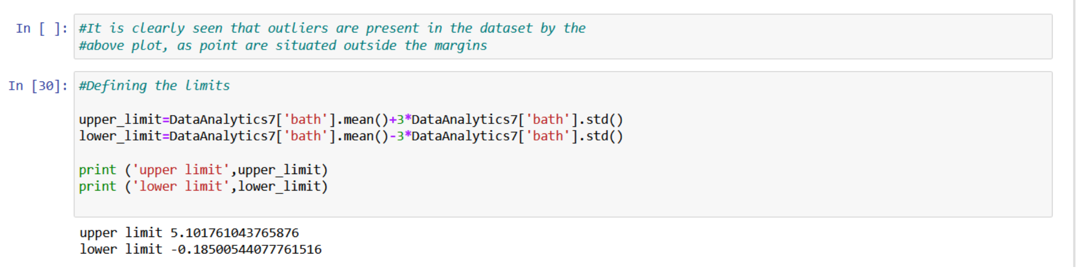
```
In [ ]: #Seems like there arent any outliers in that particular column.

In [29]: #Checking for outliers in another column (to identify it more clearly)
         sns.boxplot(DataAnalytics7['bath'])

Out[29]: <AxesSubplot: >
```

```
In [ ]: #It is clearly seen that outliers are present in the dataset by the
        #above plot, as point are situated outside the margins
```

Therefore, it can be clearly seen, outliers are present in this particular column of the dataset. Analysis will be conducted on this particular column, where the removal process will be begun from this stage.

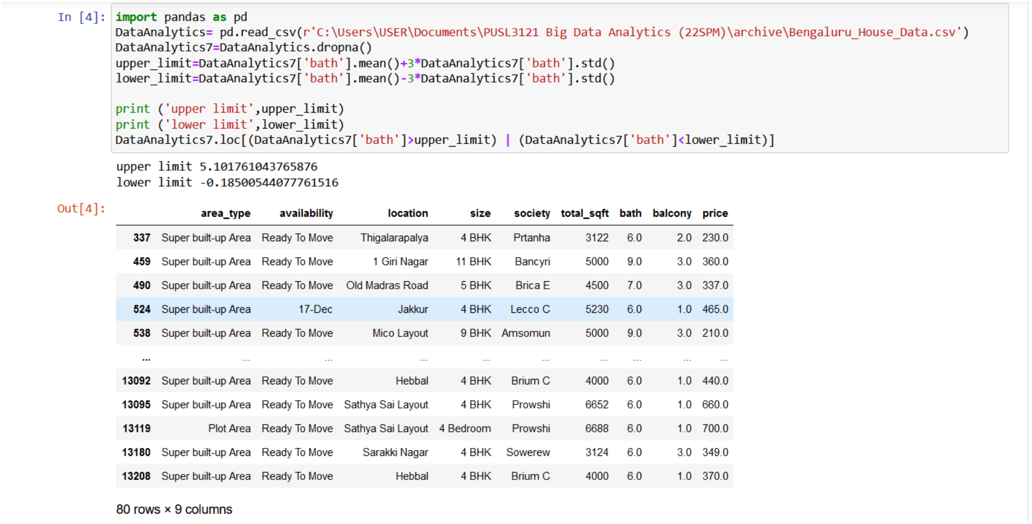
Z-score calculation

The limit values will be defined in order to identify the boundaries of the dataset



```
In [ ]: #It is clearly seen that outliers are present in the dataset by the
        #above plot, as point are situated outside the margins

In [30]: #Defining the limits

         upper_limit=DataAnalytics7['bath'].mean()+3*DataAnalytics7['bath'].std()
         lower_limit=DataAnalytics7['bath'].mean()-3*DataAnalytics7['bath'].std()

         print ('upper limit',upper_limit)
         print ('lower limit',lower_limit)

         upper limit 5.101761043765876
         lower limit -0.18500544077761516
```

Next step is to make the comparisons among the limits and the data points while eradicating it from the dataset.

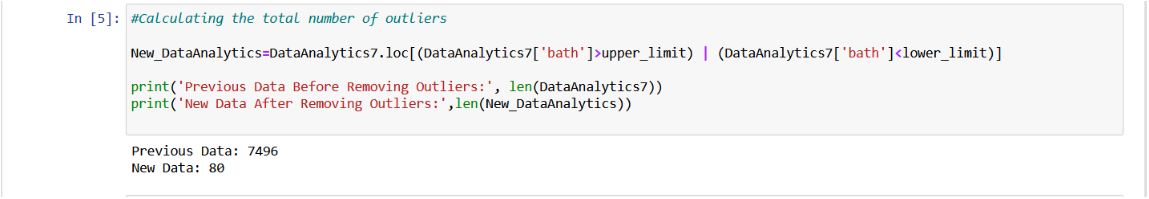


```
In [4]: import pandas as pd
        DataAnalytics= pd.read_csv(r'C:\Users\USER\Documents\PUSL3121 Big Data Analytics (22SPM)\archive\Bengaluru_House_Data.csv')
        DataAnalytics7=DataAnalytics.dropna()
        upper_limit=DataAnalytics7['bath'].mean()+3*DataAnalytics7['bath'].std()
        lower_limit=DataAnalytics7['bath'].mean()-3*DataAnalytics7['bath'].std()

        print ('upper limit',upper_limit)
        print ('lower limit',lower_limit)
        DataAnalytics7.loc[(DataAnalytics7['bath']>upper_limit) | (DataAnalytics7['bath']<lower_limit)]

        upper limit 5.101761043765876
        lower limit -0.18500544077761516

Out[4]:
```

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
337	Super built-up Area	Ready To Move	Thigalarapalya	4 BHK	Prtanha	3122	6.0	2.0	230.0
459	Super built-up Area	Ready To Move	1 Giri Nagar	11 BHK	Bancyri	5000	9.0	3.0	360.0
490	Super built-up Area	Ready To Move	Old Madras Road	5 BHK	Brica E	4500	7.0	3.0	337.0
524	Super built-up Area	17-Dec	Jakkur	4 BHK	Lecco C	5230	6.0	1.0	465.0
538	Super built-up Area	Ready To Move	Mico Layout	9 BHK	Amsomun	5000	9.0	3.0	210.0
...	...	...	...	...	...	...	...	...	...
13092	Super built-up Area	Ready To Move	Hebbal	4 BHK	Brium C	4000	6.0	1.0	440.0
13095	Super built-up Area	Ready To Move	Sathya Sai Layout	4 BHK	Prowshi	6652	6.0	1.0	660.0
13119	Plot Area	Ready To Move	Sathya Sai Layout	4 Bedroom	Prowshi	6688	6.0	1.0	700.0
13180	Super built-up Area	Ready To Move	Sarakki Nagar	4 BHK	Sowerew	3124	6.0	3.0	349.0
13208	Super built-up Area	Ready To Move	Hebbal	4 BHK	Brium C	4000	6.0	1.0	370.0

80 rows × 9 columns

Verification can be obtained as a summary by performing a simple calculation



```
In [5]: #Calculating the total number of outliers

        New_DataAnalytics=DataAnalytics7.loc[(DataAnalytics7['bath']>upper_limit) | (DataAnalytics7['bath']<lower_limit)]

        print('Previous Data Before Removing Outliers:', len(DataAnalytics7))
        print('New Data After Removing Outliers:',len(New_DataAnalytics))

        Previous Data: 7496
        New Data: 80
```

Scenario 2

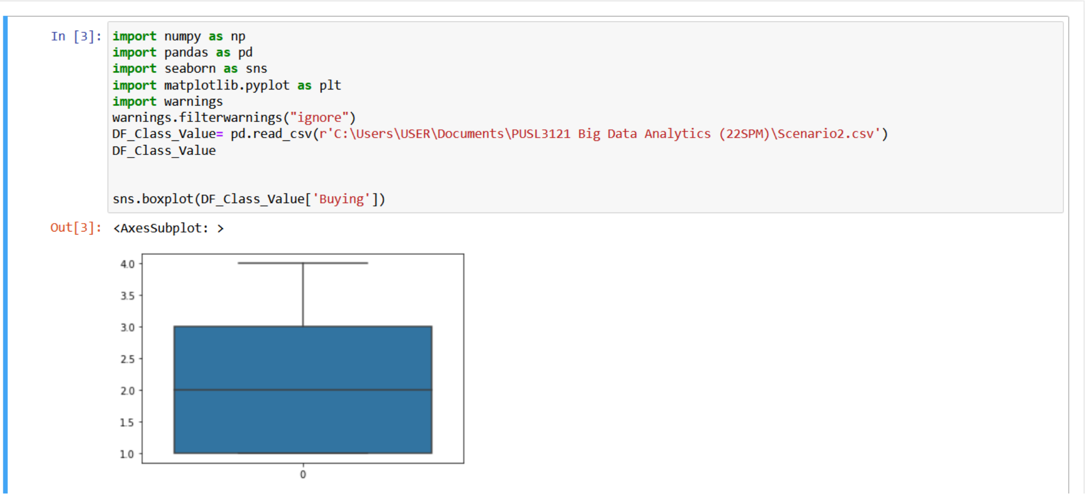
Dataset

The dataset, which was given using the xlsx file extension was converted to csv format and using the Jupyter notebook the dataset was loaded in to a pandas data frame enabling the view option.

The usual procedure is to check for any outliers or missing values in the dataset before any analysis is conducted using the python language.

The below screenshot will demonstrate how the outlier detection process was carried out.

Outlier Detection

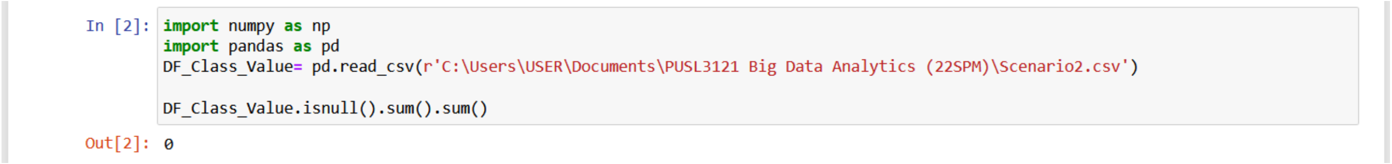


```
In [3]: import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt
        import warnings
        warnings.filterwarnings("ignore")
        DF_Class_Value= pd.read_csv(r'C:\Users\USER\Documents\PUSL3121 Big Data Analytics (22SPM)\Scenario2.csv')
        DF_Class_Value


        sns.boxplot(DF_Class_Value['Buying'])

Out[3]: <AxesSubplot: >
```

- ❖ Outliers can be identified using several methods, out of which the data visualization concept is used to graphically detect if any outliers are present in the dataset

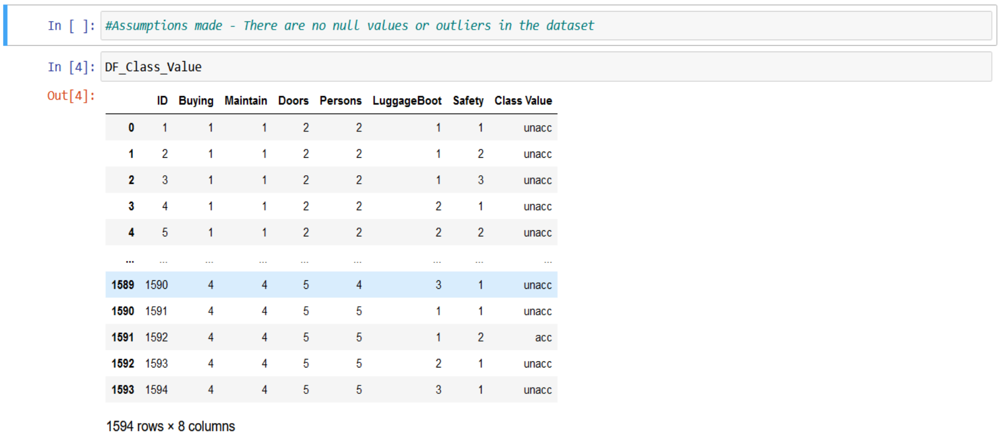
Null/Missing Values Detection

Another important aspect of data cleansing is to identify for any null or missing values in the dataset. The below shown screenshot will indicate the total number of impure data is present in the dataset



```
In [2]: import numpy as np
        import pandas as pd
        DF_Class_Value= pd.read_csv(r'C:\Users\USER\Documents\PUSL3121 Big Data Analytics (22SPM)\Scenario2.csv')

        DF_Class_Value.isnull().sum().sum()

Out[2]: 0
```

After the above processes are done the next step is load the dataset and check for the new number of rows and columns available after the cleansing is carried out. Since there were no outliers or null/missing values found, the total number of rows and columns were unchanged.



```
In [ ]: #Assumptions made - There are no null values or outliers in the dataset

In [4]: DF_Class_Value

Out[4]:
```

	ID	Buying	Maintain	Doors	Persons	LuggageBoot	Safety	Class Value
0	1	1	1	2	2	1	1	unacc
1	2	1	1	2	2	1	2	unacc
2	3	1	1	2	2	1	3	unacc
3	4	1	1	2	2	2	1	unacc
4	5	1	1	2	2	2	2	unacc
...	...	...	...	...	...	...	...	...
1589	1590	4	4	5	4	3	1	unacc
1590	1591	4	4	5	5	1	1	unacc
1591	1592	4	4	5	5	1	2	acc
1592	1593	4	4	5	5	2	1	unacc
1593	1594	4	4	5	5	3	1	unacc

1594 rows × 8 columns

Analyzes 3

Entropy

Entropy can be defined as the degree of randomness of data points in the system. Entropy can be calculated using the probabilities which are relative to the target group. Instead of manual calculations, in here python language will be used to derive a numerical value.

Information Gain

By determining the information gain value, a certain insight can be obtained on how much a particular variable or feature withholds information in the data point.

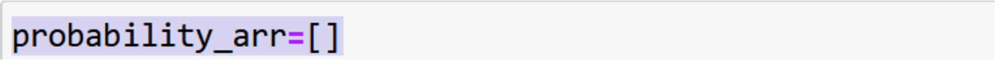
Decision Tree

The tree structure will be the graphical path way of identifying designated instructions for the calculated values for a certain scenario (flow chart).

Calculation

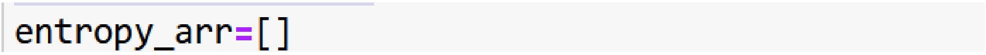
- In the initial step , empty arrays will be defined in order to containerize the calculations made before the decision tree is drawn

For Probabilities based on Column Names



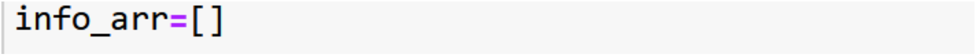
```
probability_arr=[]
```

Array for Entropy



```
entropy_arr=[]
```

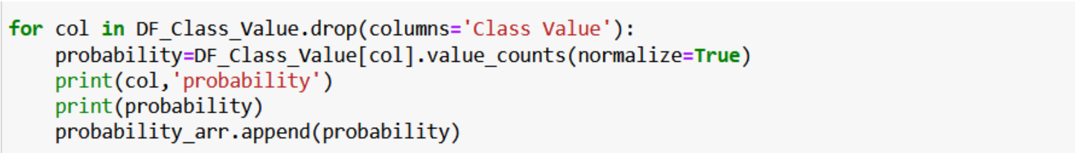
Array for Information Gain



```
info_arr=[]
```

- The next step would be to calculate the *probability* in order to use it in the entropy equation. A for loop will be used to calculate the unique column value using the ***value_counts()*** method. The values will be printed afterwards as outputs.

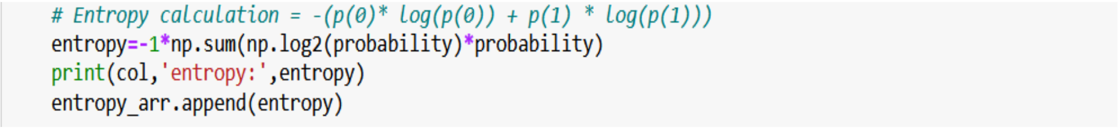
For loop



```
for col in DF_Class_Value.drop(columns='Class Value'):
    probability=DF_Class_Value[col].value_counts(normalize=True)
    print(col,'probability')
    print(probability)
    probability_arr.append(probability)
```

- Entropy calculation will followed after the above step, where logarithmic functions will be computed in order to obtain a specific entropy value for **each column** (Buying, Maintain, Doors, Persons, Luggage Boot, Safety) except for the **target class** (Class Value)

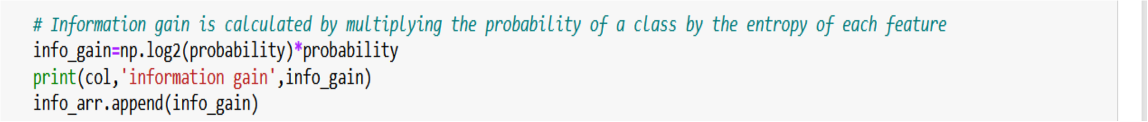
Entropy equation



```
# Entropy calculation = -(p(0)* log(p(0)) + p(1) * log(p(1)))
entropy=-1*np.sum(np.log2(probability)*probability)
print(col,'entropy:',entropy)
entropy_arr.append(entropy)
```

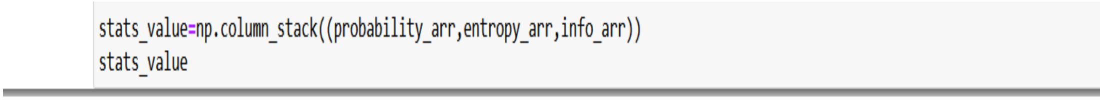
- ❖ Information gain is the summation of the multiplication between probability value and entropy of each column, where the highest value is selected on deciding the root node of the decision tree.

Theory : *The root node of the decision tree will be the attribute which contain the maximum information gain*



```
# Information gain is calculated by multiplying the probability of a class by the entropy of each feature
info_gain=np.log2(probability)*probability
print(col,'information gain',info_gain)
info_arr.append(info_gain)
```

- ❖ In the next step, all the values will be appended in to the empty arrays derived in one of the earlier steps.

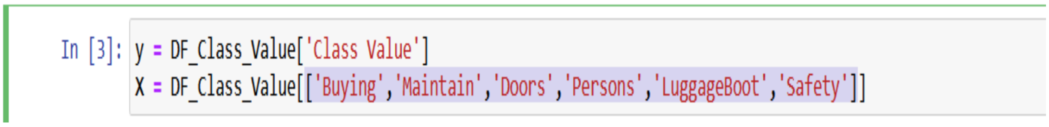


```
stats_value=np.column_stack((probability_arr,entropy_arr,info_arr))
stats_value
```

Plotting

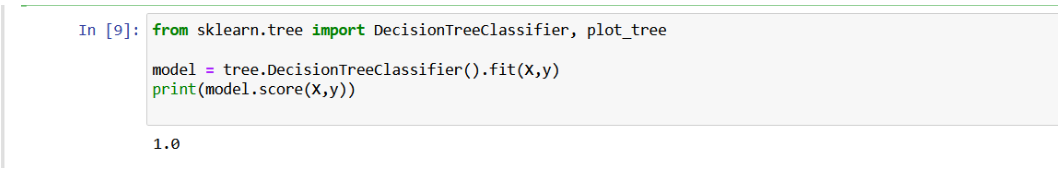
In order to plot the decision tree, the cleansed dataset needs to be trained.

The columns of the dataset are separately identified and assigned to X and Y variables.



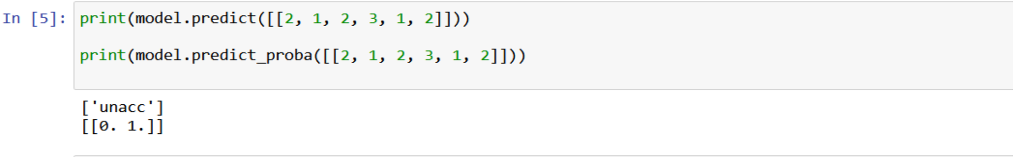
```
In [3]: y = DF_Class_Value['Class Value']
        X = DF_Class_Value[['Buying','Maintain','Doors','Persons','LuggageBoot','Safety']]
```

Next step taken was to calculate the accuracy of the model



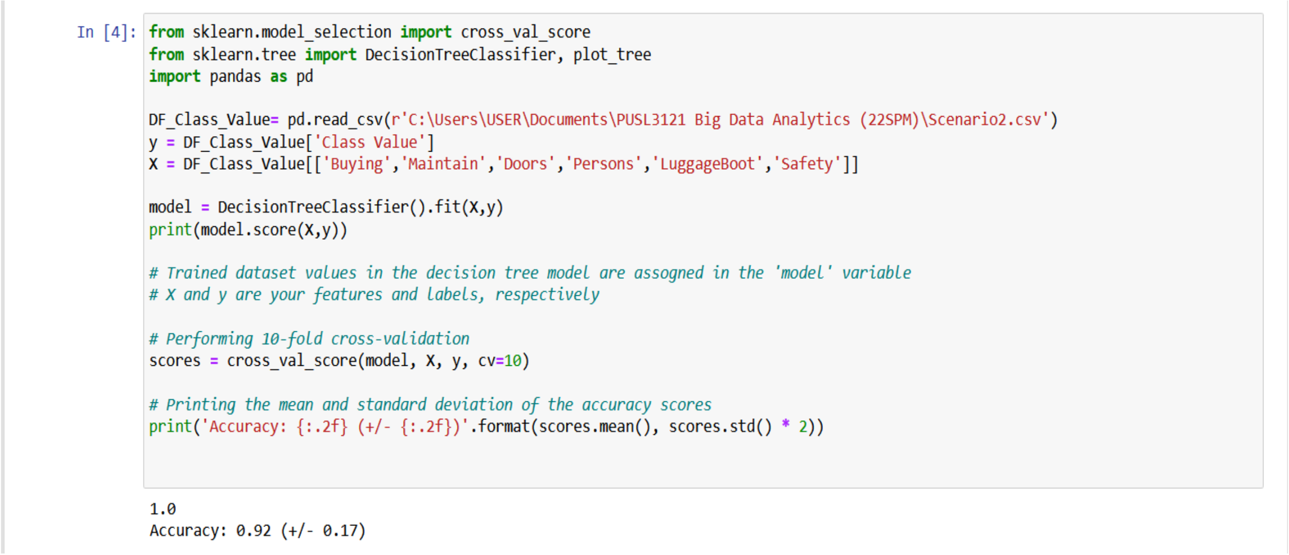
```
In [9]: from sklearn.tree import DecisionTreeClassifier, plot_tree

        model = tree.DecisionTreeClassifier().fit(X,y)
        print(model.score(X,y))

        1.0
```

Below mentioned lines will predict the target variable value for a new set of independent variables. In this case out of the two possible outcomes would be *‘unacc’ and ‘acc’*



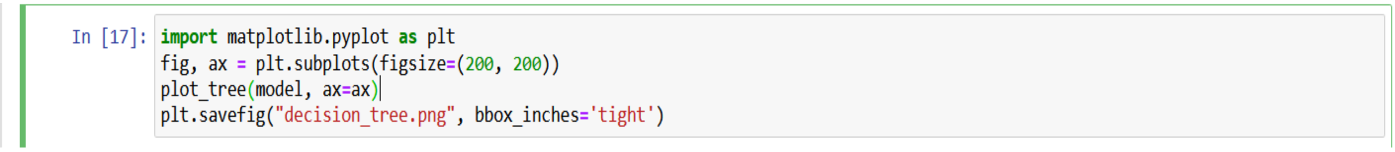
```
In [5]: print(model.predict([[2, 1, 2, 3, 1, 2]]))

        print(model.predict_proba([[2, 1, 2, 3, 1, 2]]))

        ['unacc']
        [[0. 1.]]
```

By running the below code the total accuracy level of model created can be found using 10 fold cross validation method.



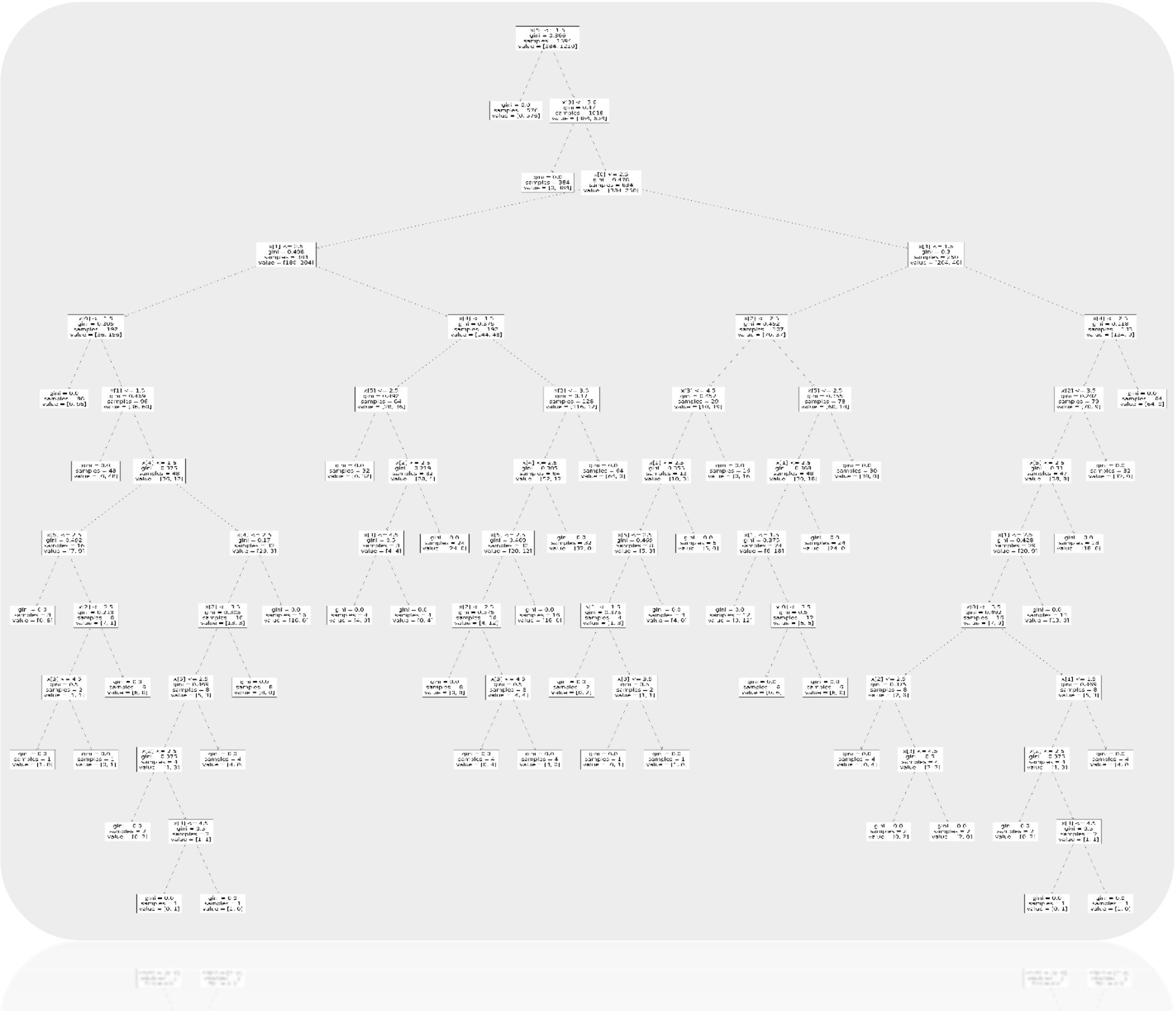
```
In [4]: from sklearn.model_selection import cross_val_score
        from sklearn.tree import DecisionTreeClassifier, plot_tree
        import pandas as pd

        DF_Class_Value= pd.read_csv(r'C:\Users\USER\Documents\PUSL3121 Big Data Analytics (22SPM)\Scenario2.csv')
        y = DF_Class_Value['Class Value']
        X = DF_Class_Value[['Buying','Maintain','Doors','Persons','LuggageBoot','Safety']]

        model = DecisionTreeClassifier().fit(X,y)
        print(model.score(X,y))

        # Trained dataset values in the decision tree model are assogned in the 'model' variable
        # X and y are your features and labels, respectively

        # Performing 10-fold cross-validation
        scores = cross_val_score(model, X, y, cv=10)

        # Printing the mean and standard deviation of the accuracy scores
        print('Accuracy: {:.2f} (+/- {:.2f})'.format(scores.mean(), scores.std() * 2))

        1.0
        Accuracy: 0.92 (+/- 0.17)
```

Finally, the decision tree can be plotted by mentioning the axes and specifying the relevant keywords from the matplotlib library



```
In [17]: import matplotlib.pyplot as plt
         fig, ax = plt.subplots(figsize=(200, 200))
         plot_tree(model, ax=ax)
         plt.savefig("decision_tree.png", bbox_inches='tight')
```

Decision Tree (PNG Image)



Conclusion

- Obtained datasets were cleansed by eradicating
  - Null/Missing Values
  - Outliers

- By using methods such as
  - Data Visualizing
  - Z-score calculations
  - Interpolation
  - Imputation
  - Deletion

- Then the New Dataset was transferred to the process of model training in order to develop the decision tree

- A major two steps were conducted in plotting the specific tree
  - Entropy Calculation
  - Information Gain Calculation

- Finally, the graphical illustration of the decision tree was plotted using the matplotlib library

Individual Workload Matrix

Name	Plymouth Id	E-mail	Contribution
Mahindu Bandaranayake	10749841	10749841@students.plymouth.ac.uk	- Report Structuring - Entropy Calculation for Scenario 2 - Data Cleansing for Scenario 1 and 2
Madapathage Madushan	*10749856*	10749856@students.plymouth.ac.uk	- Identifying relevant Dataset - Data Cleansing for Scenario 1
Maharu Balapitiya	*10749837*	10749837@students.plymouth.ac.uk	- Information Gain Calculation for Scenario 2 - Report Structuring
Pamunuwe Jayalath	*10749858*	10749858@students.plymouth.ac.uk	- Data Cleansing for Scenario 2 - Plotting
Charithma Ramanayake	*10749834*	10749834@students.plymouth.ac.uk	- Dataset Training for Scenario 2 - Information Gain Calculation
Jathusnanan Nadarasa	*10749857*	10749857@students.plymouth.ac.uk	- Report Structuring - Data Cleansing for Scenario 2

References

Aljanabi, K. and Yahya Almayali, Z. (2019) ‘Toward Good Quality Data: A Comparative Study for Solving Missing Values in Big Data Classification of Arabic texts using four classifiers View project’. Available at: https://doi.org/10.4206/aus.2019.n26.2.22.

Kwak, S.K. and Kim, J.H. (2017) ‘Statistical data preparation: Management of missing values and outliers’, *Korean Journal of Anesthesiology*. Korean Society of Anesthesiologists, pp. 407–411. Available at: https://doi.org/10.4097/kjae.2017.70.4.407.

Seliem, Muhammad M and Seliem, Muhammad Metwally (2022) *Handling Outlier Data as Missing Values by Imputation Methods: Application of Machine Learning Algorithms Non-parametric and Semi-parametric regression View project Non-parametric and Semi-parametric regression models View project Handling Outlier Data as Missing Values by Imputation Methods: Application of Machine Learning Algorithms*, *Turkish Journal of Computer and Mathematics Education*. Available at: https://www.researchgate.net/publication/358565692.

Zhang, S. *et al.* (2019) ‘An outlier detection scheme for dynamical sequential datasets’, *Communications in Statistics: Simulation and Computation*, 48(5), pp. 1450–1502. Available at: https://doi.org/10.1080/03610918.2017.1414249.