Rupesh Dharme
31124

DATE PAGE

Assignment 01

DOP : 03 Jan 2022 DOS: 10 Jan 2022

Title : Data Wrangling - I

Problem statement : Perform following operation using python on any open source database/ dataset.

1. Import all required libraries.
2. Locate an open source web dataset. Provide a clear description of the data and it's source.
3. Load the dataset into panda's dataframe.
4. Data preprocessing: check for missing values in data. Use pandas describe() function to get some initial statistics. provide variable discriptions, Types of variables etc. check dimensions of dataframe.
5. Data formatting and normalization: Summarize the types of variables by checking types of variables in dataset. If variables are not in correct datatypes apply proper type conversions.
6. Turn categorial values into quantitative variables.

Rupesh Dharme
31124

DATE PAGE

Learning objectives:

To learn and understanding data wrangling using python (pandas)

To perform data preprocessing, formatting and normalization.

To perform encoding on categorical data variables.

Learning outcomes: students will be able to:

perform basic data preprocessing, formatting and normalization.

perform encoding for conversion.

S/W and H/W requirements:

Win 10 64 bits, 8 GB RAM 512 GB SSD, Intel core i5, Pycharme (Jupyter notebook), python 3.9

Theory:

After the data is collected from different sources and before the data is used to prepare models, it has to be processed, this process makes the data more consistent and called as data wrangling.

Python is one of the most preferred languages for data science. The pandas library

Rupesh
31124

provides us with various pre implemented functions to preprocess our data some of the functions are.

df.shape → Gives us dimensions of df
df.isnull() → boolean df where each cell holds True if ~~not~~ null.
df.describe() → Gives us some statistical data around our df.
df.columns → returns list of columns.

pandas also provides us with many functions which help us fill the missing values or drop the rows. Also categorical data can be converted to quantative data variable.

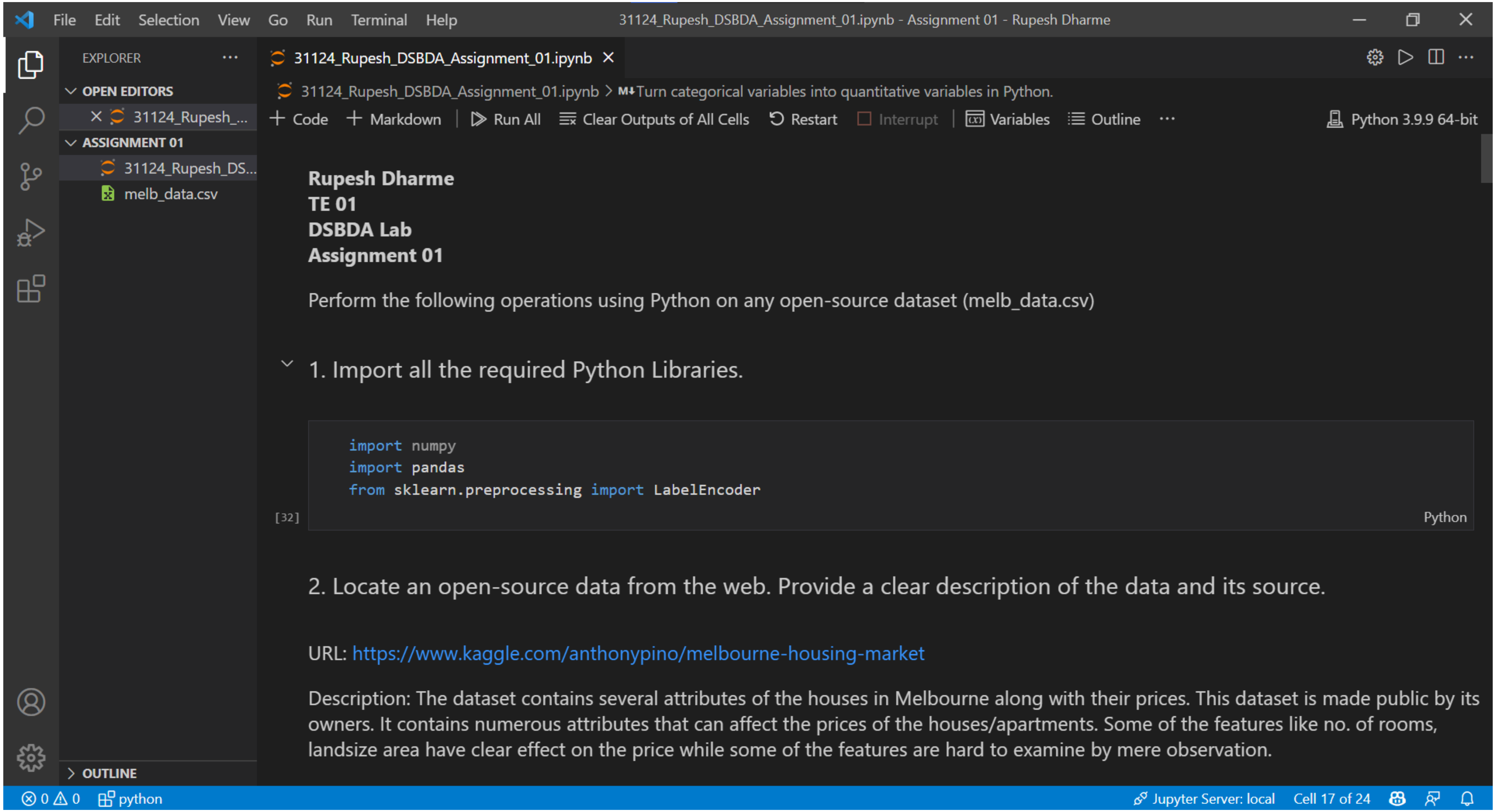
Analysis / methods:

The given dataset contained 13580 x 21 with missing values in some columns that we filled with mean for numeric columns. We converted 'object' datatypes into 'string' datatype. We also converted categorical data.

Rupesh
31124

DATE | PAGE

Conclusion:
successfully performed data wrangling tasks on the dataset.


File Edit Selection View Go Run Terminal Help
31124_Rupesh_DSBDA_Assignment_01.ipynb - Assignment 01 - Rupesh Dharme
EXPLORER
OPEN EDITORS
31124_Rupesh_...
ASSIGNMENT 01
31124_Rupesh_DS...
melb_data.csv
OUTLINE
31124_Rupesh_DSBDA_Assignment_01.ipynb
31124_Rupesh_DSBDA_Assignment_01.ipynb > M↓Turn categorical variables into quantitative variables in Python.
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.9.9 64-bit
Python
⊗ 0 ⚠ 0 python
Jupyter Server: local Cell 17 of 24

Rupesh Dharme
TE 01
DSBDA Lab
Assignment 01

Perform the following operations using Python on any open-source dataset (melb_data.csv)

1. Import all the required Python Libraries.

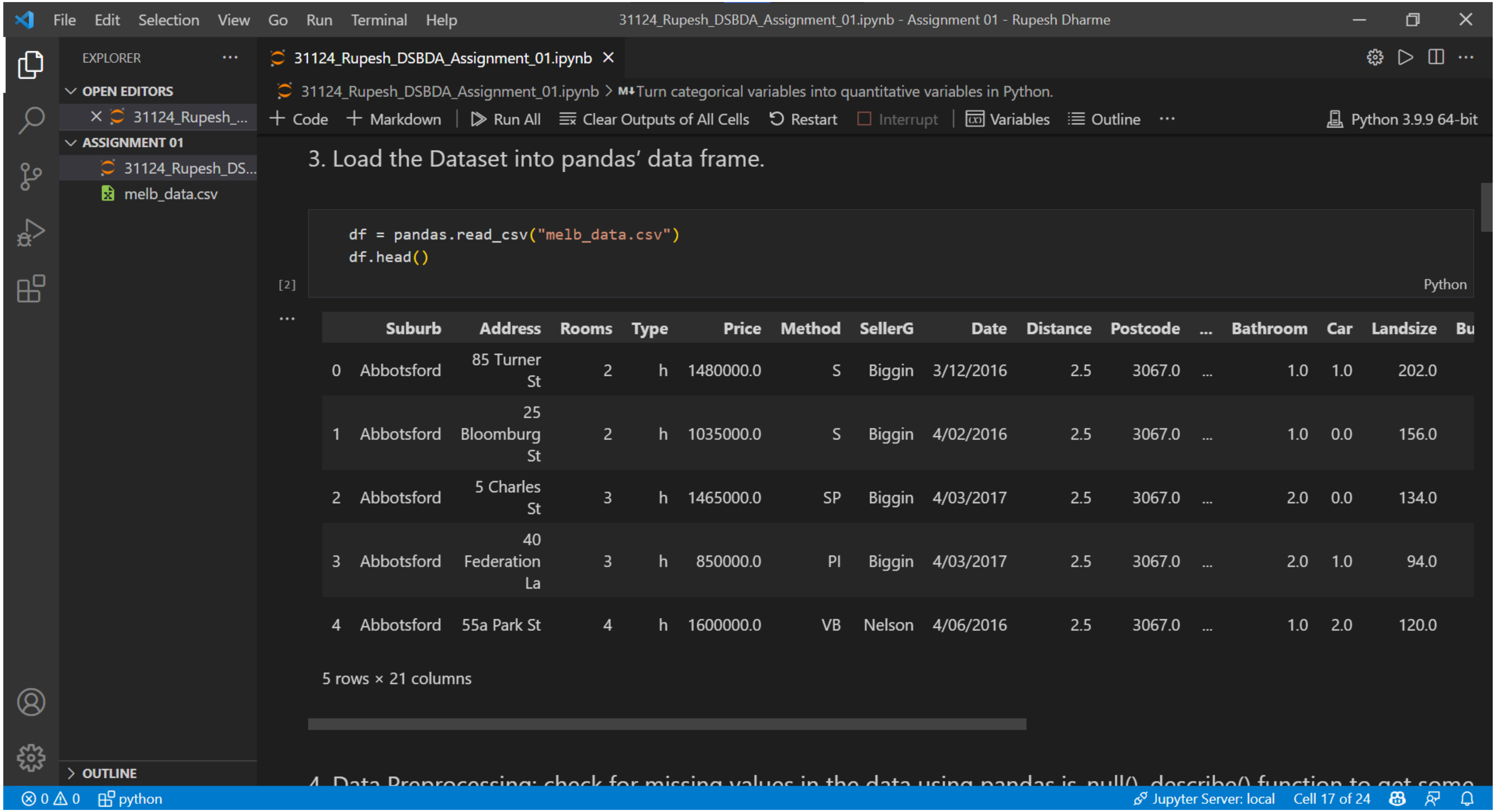
```python
import numpy
import pandas
from sklearn.preprocessing import LabelEncoder
```

[32]

2. Locate an open-source data from the web. Provide a clear description of the data and its source.

URL: https://www.kaggle.com/anthonypino/melbourne-housing-market

Description: The dataset contains several attributes of the houses in Melbourne along with their prices. This dataset is made public by its owners. It contains numerous attributes that can affect the prices of the houses/apartments. Some of the features like no. of rooms, landsize area have clear effect on the price while some of the features are hard to examine by mere observation.



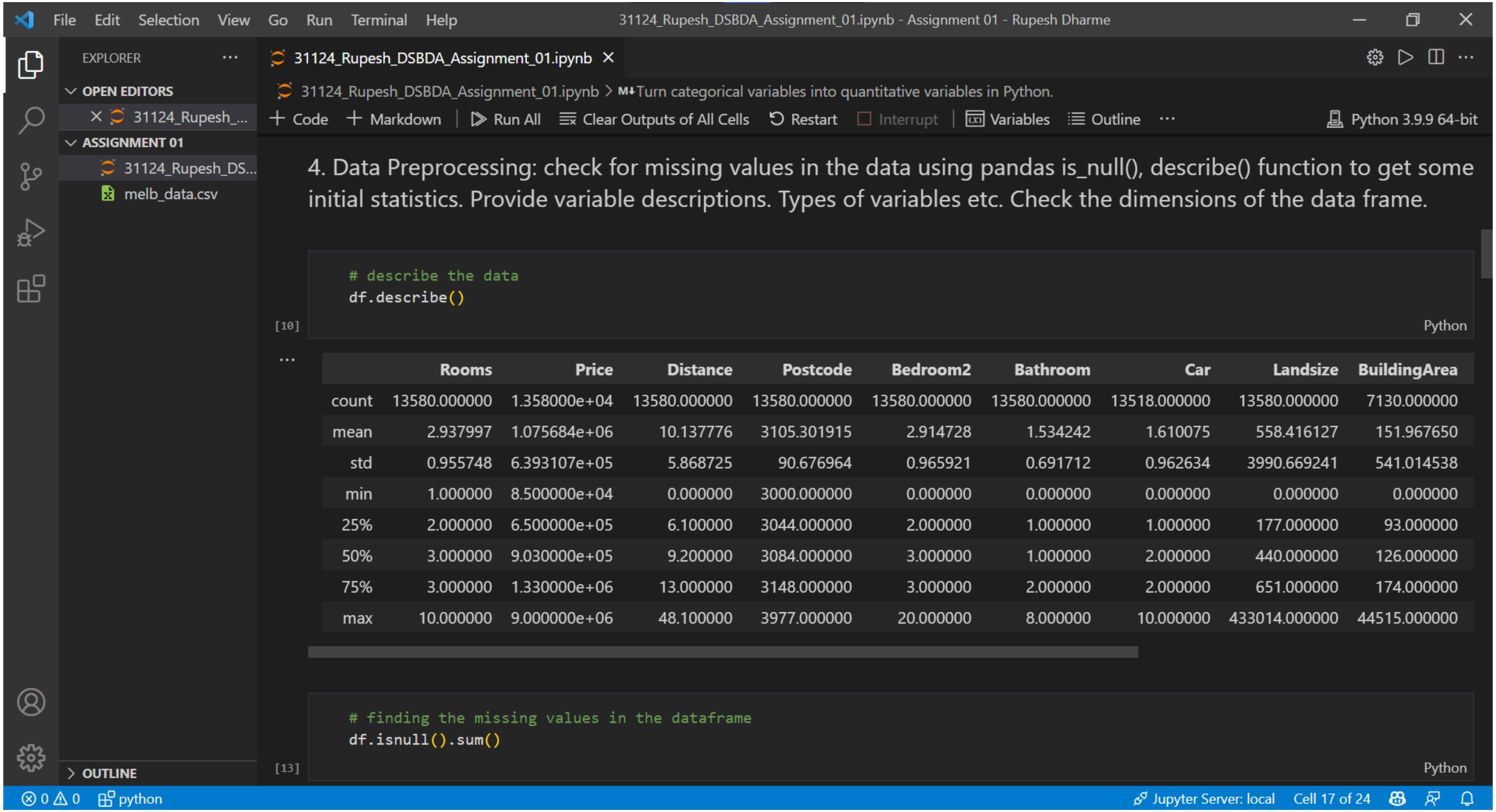
3. Load the Dataset into pandas' data frame.

```
df = pandas.read_csv("melb_data.csv")
df.head()
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...	Bathroom	Car	Landsize	Bu
0	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	3067.0	...	1.0	1.0	202.0	
1	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5	3067.0	...	1.0	0.0	156.0	
2	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5	3067.0	...	2.0	0.0	134.0	
3	Abbotsford	40 Federation La	3	h	850000.0	PI	Biggin	4/03/2017	2.5	3067.0	...	2.0	1.0	94.0	
4	Abbotsford	55a Park St	4	h	1600000.0	VB	Nelson	4/06/2016	2.5	3067.0	...	1.0	2.0	120.0	

5 rows × 21 columns

4. Data Preprocessing: check for missing values in the data using pandas is null() describe() function to get some

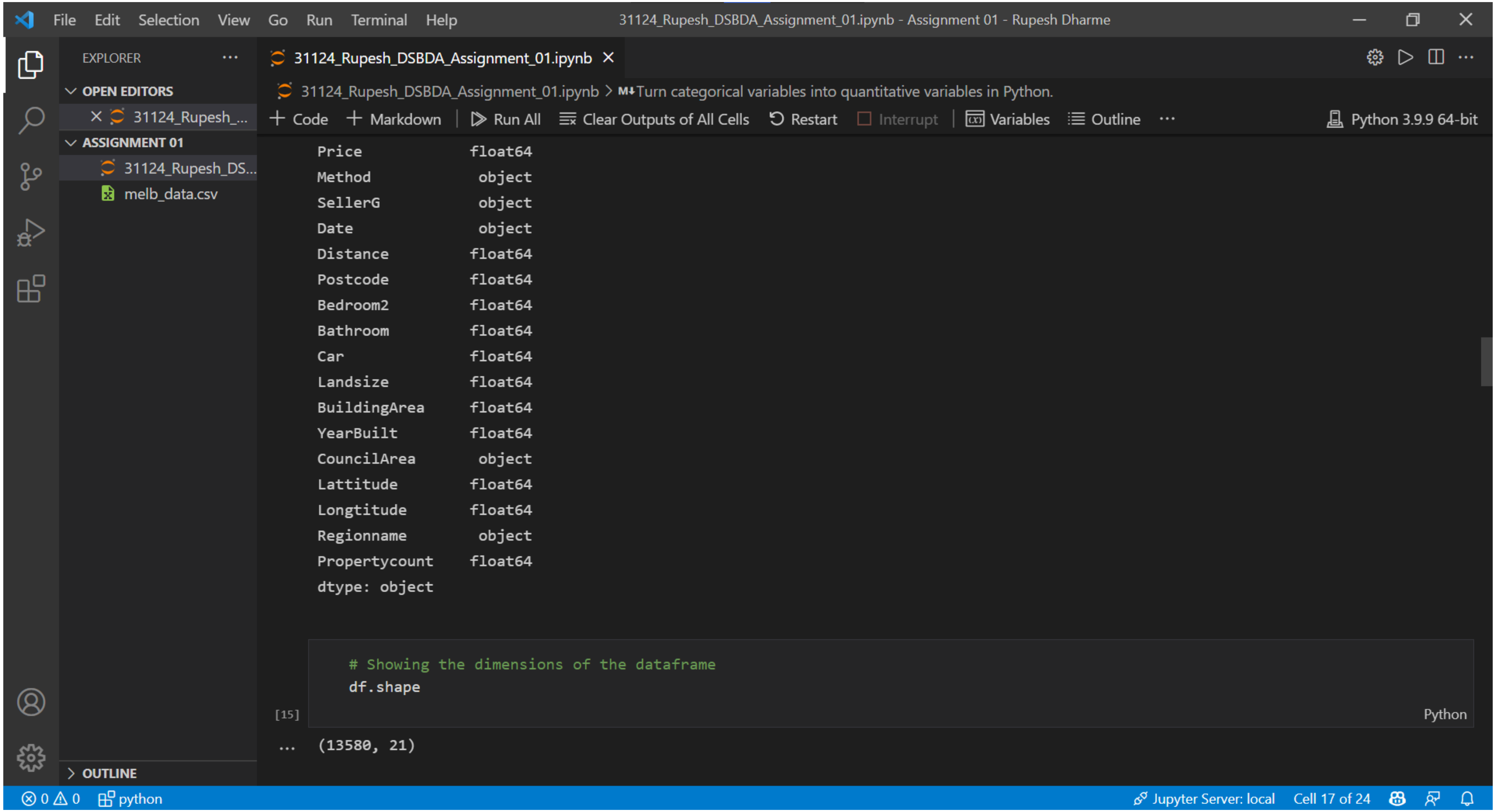


4. Data Preprocessing: check for missing values in the data using pandas is_null(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.

```
# describe the data
df.describe()
```

```
          Rooms         Price      Distance      Postcode      Bedroom2      Bathroom           Car       Landsize  BuildingArea
count  13580.000000  1.358000e+04  13580.000000  13580.000000  13580.000000  13580.000000  13518.000000   13580.000000   7130.000000
mean       2.937997  1.075684e+06     10.137776   3105.301915      2.914728      1.534242      1.610075     558.416127    151.967650
std        0.955748  6.393107e+05      5.868725     90.676964      0.965921      0.691712      0.962634    3990.669241    541.014538
min        1.000000  8.500000e+04      0.000000   3000.000000      0.000000      0.000000      0.000000       0.000000      0.000000
25%        2.000000  6.500000e+05      6.100000   3044.000000      2.000000      1.000000      1.000000     177.000000     93.000000
50%        3.000000  9.030000e+05      9.200000   3084.000000      3.000000      1.000000      2.000000     440.000000    126.000000
75%        3.000000  1.330000e+06     13.000000   3148.000000      3.000000      2.000000      2.000000     651.000000    174.000000
max       10.000000  9.000000e+06     48.100000   3977.000000     20.000000      8.000000     10.000000  433014.000000  44515.000000
```

```
# finding the missing values in the dataframe
df.isnull().sum()
```

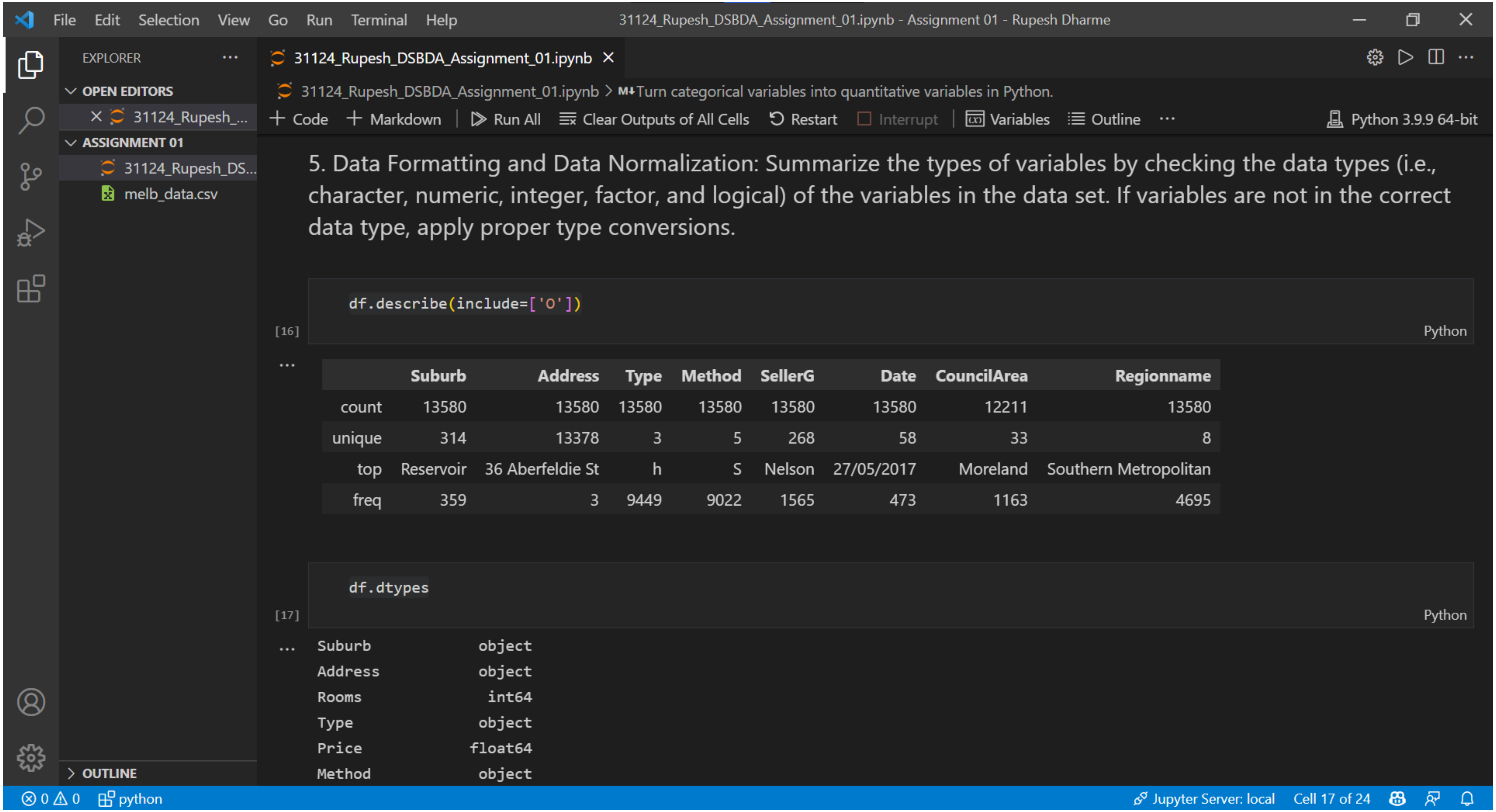

File Edit Selection View Go Run Terminal Help
31124_Rupesh_DSBDA_Assignment_01.ipynb - Assignment 01 - Rupesh Dharme
EXPLORER
OPEN EDITORS
31124_Rupesh_...
ASSIGNMENT 01
31124_Rupesh_DS...
melb_data.csv
OUTLINE
31124_Rupesh_DSBDA_Assignment_01.ipynb
31124_Rupesh_DSBDA_Assignment_01.ipynb > M↓Turn categorical variables into quantitative variables in Python.
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.9.9 64-bit
Python
⊗ 0 ⚠ 0 python
Jupyter Server: local Cell 17 of 24

```
Price            float64
Method            object
SellerG           object
Date              object
Distance         float64
Postcode         float64
Bedroom2         float64
Bathroom         float64
Car              float64
Landsize         float64
BuildingArea     float64
YearBuilt        float64
CouncilArea       object
Lattitude        float64
Longtitude       float64
Regionname        object
Propertycount    float64
dtype: object
```

```python
# Showing the dimensions of the dataframe
df.shape
```

[15]

```
(13580, 21)
```



5. Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.

[16]

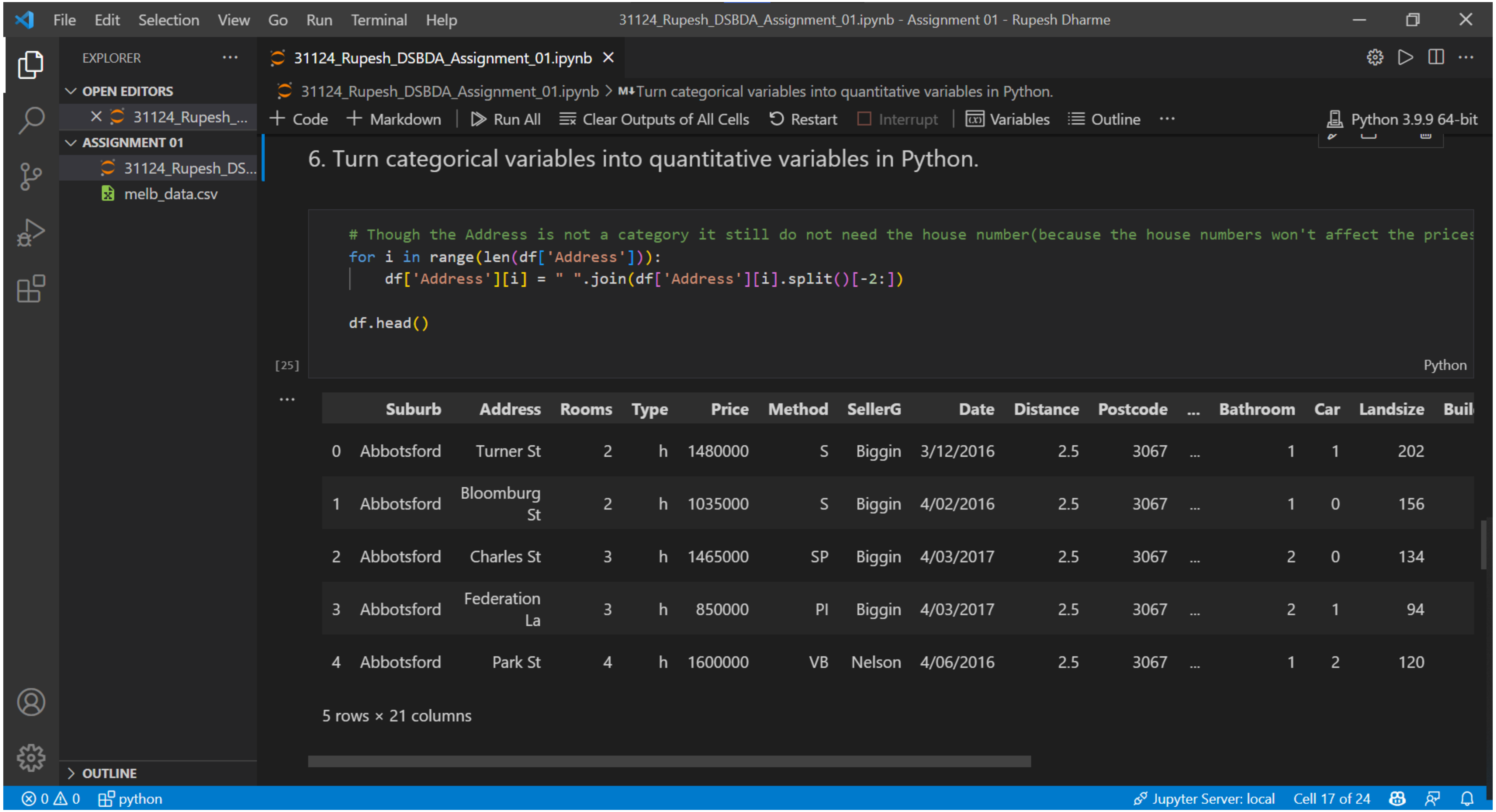
```
df.describe(include=['O'])
```

	Suburb	Address	Type	Method	SellerG	Date	CouncilArea	Regionname
count	13580	13580	13580	13580	13580	13580	12211	13580
unique	314	13378	3	5	268	58	33	8
top	Reservoir	36 Aberfeldie St	h	S	Nelson	27/05/2017	Moreland	Southern Metropolitan
freq	359	3	9449	9022	1565	473	1163	4695

[17]

```
df.dtypes
```

```
Suburb            object
Address           object
Rooms              int64
Type              object
Price            float64
Method            object
```



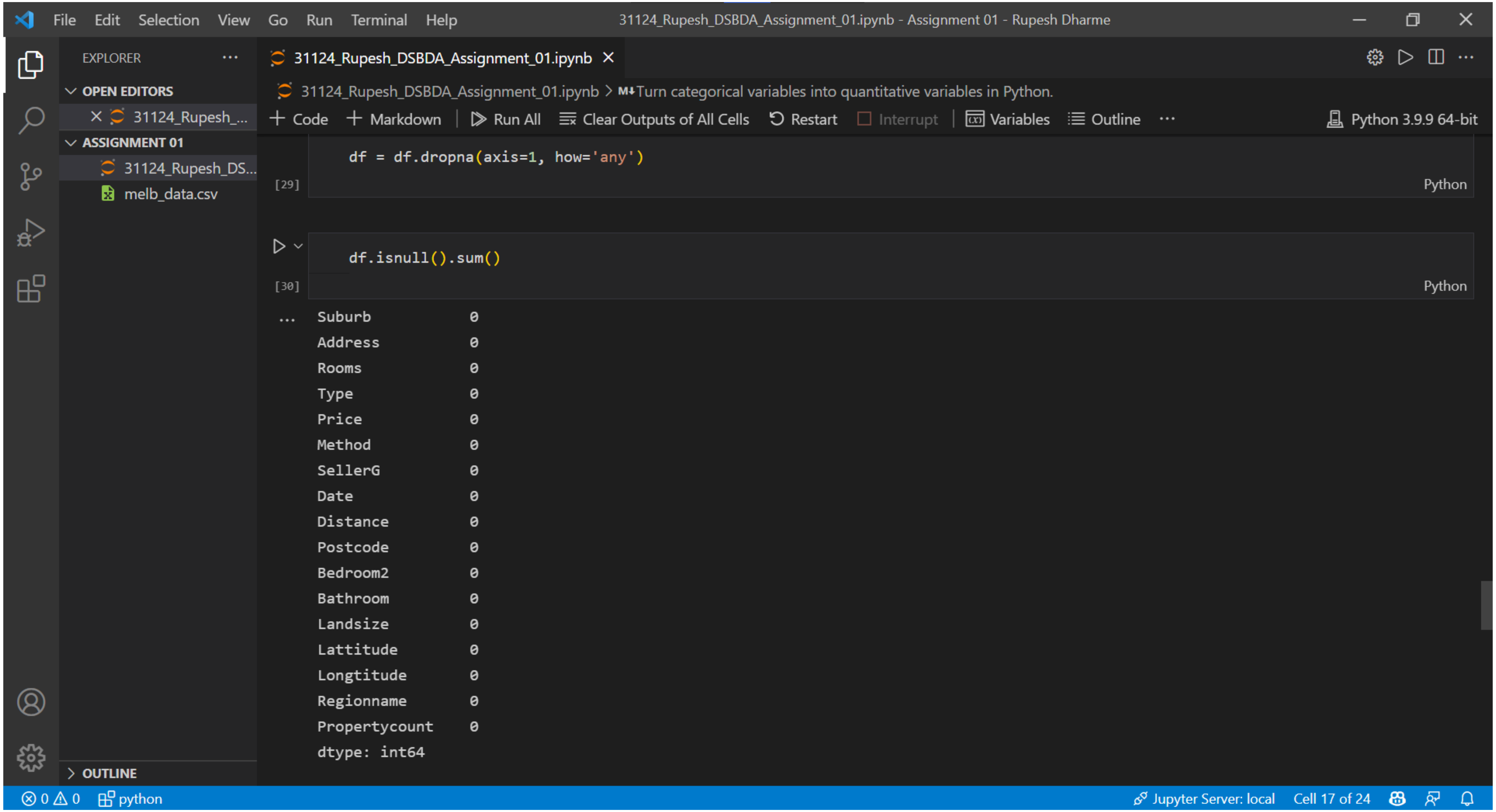
6. Turn categorical variables into quantitative variables in Python.

```
# Though the Address is not a category it still do not need the house number(because the house numbers won't affect the prices
for i in range(len(df['Address'])):
    df['Address'][i] = " ".join(df['Address'][i].split()[-2:])

df.head()
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...	Bathroom	Car	Landsize	Buil
0	Abbotsford	Turner St	2	h	1480000	S	Biggin	3/12/2016	2.5	3067	...	1	1	202	
1	Abbotsford	Bloomburg St	2	h	1035000	S	Biggin	4/02/2016	2.5	3067	...	1	0	156	
2	Abbotsford	Charles St	3	h	1465000	SP	Biggin	4/03/2017	2.5	3067	...	2	0	134	
3	Abbotsford	Federation La	3	h	850000	PI	Biggin	4/03/2017	2.5	3067	...	2	1	94	
4	Abbotsford	Park St	4	h	1600000	VB	Nelson	4/06/2016	2.5	3067	...	1	2	120	

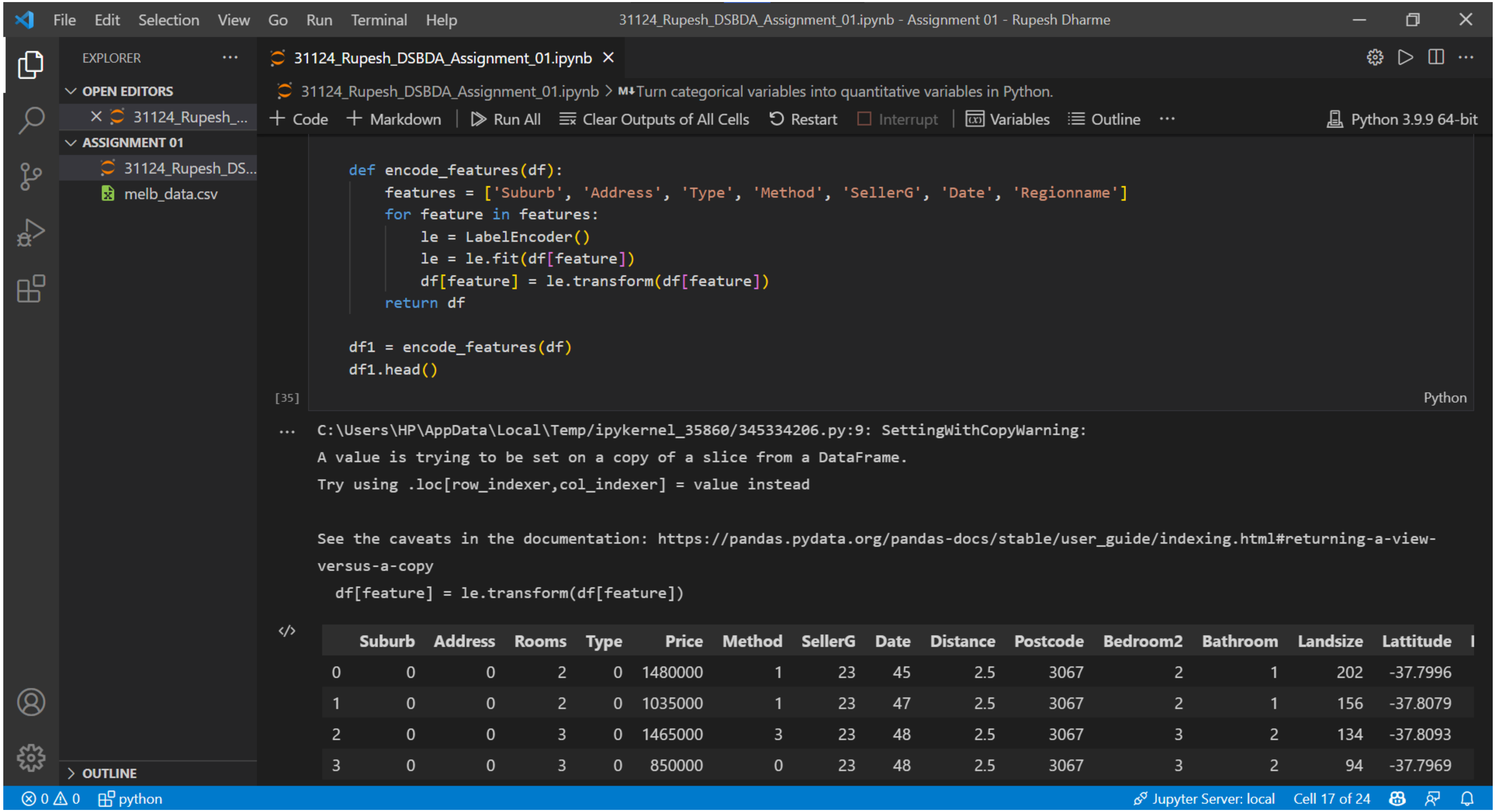
5 rows × 21 columns


File Edit Selection View Go Run Terminal Help
31124_Rupesh_DSBDA_Assignment_01.ipynb - Assignment 01 - Rupesh Dharme
EXPLORER
OPEN EDITORS
31124_Rupesh_...
ASSIGNMENT 01
31124_Rupesh_DS...
melb_data.csv
OUTLINE
31124_Rupesh_DSBDA_Assignment_01.ipynb
31124_Rupesh_DSBDA_Assignment_01.ipynb > M↓Turn categorical variables into quantitative variables in Python.
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.9.9 64-bit
Python
Python
0 0 python
Jupyter Server: local Cell 17 of 24

```
[29]
df = df.dropna(axis=1, how='any')
```

```
[30]
df.isnull().sum()
```

```
Suburb           0
Address          0
Rooms            0
Type             0
Price            0
Method           0
SellerG          0
Date             0
Distance         0
Postcode         0
Bedroom2         0
Bathroom         0
Landsize         0
Lattitude        0
Longtitude       0
Regionname       0
Propertycount    0
dtype: int64
```


File Edit Selection View Go Run Terminal Help
31124_Rupesh_DSBDA_Assignment_01.ipynb - Assignment 01 - Rupesh Dharme
EXPLORER
OPEN EDITORS
31124_Rupesh_...
ASSIGNMENT 01
31124_Rupesh_DS...
melb_data.csv
OUTLINE
31124_Rupesh_DSBDA_Assignment_01.ipynb
31124_Rupesh_DSBDA_Assignment_01.ipynb > M↓Turn categorical variables into quantitative variables in Python.
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.9.9 64-bit
[35]
Python
0 0 python
Jupyter Server: local Cell 17 of 24

```python
def encode_features(df):
    features = ['Suburb', 'Address', 'Type', 'Method', 'SellerG', 'Date', 'Regionname']
    for feature in features:
        le = LabelEncoder()
        le = le.fit(df[feature])
        df[feature] = le.transform(df[feature])
    return df

df1 = encode_features(df)
df1.head()
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_35860/345334206.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df[feature] = le.transform(df[feature])
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	Bedroom2	Bathroom	Landsize	Lattitude
0	0	0	2	0	1480000	1	23	45	2.5	3067	2	1	202	-37.7996
1	0	0	2	0	1035000	1	23	47	2.5	3067	2	1	156	-37.8079
2	0	0	3	0	1465000	3	23	48	2.5	3067	3	2	134	-37.8093
3	0	0	3	0	850000	0	23	48	2.5	3067	3	2	94	-37.7969