Artificial Intelligence and Machine Learning

Assignment -2

Jeyavvanth.R

21BCE2472

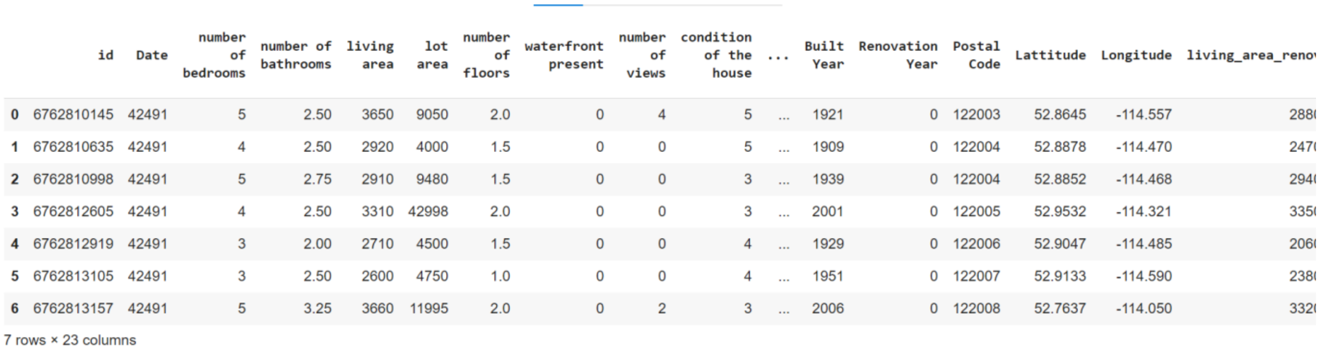
jeyavvanth.2021@vitstudent.ac.in

Task-1,2

Code

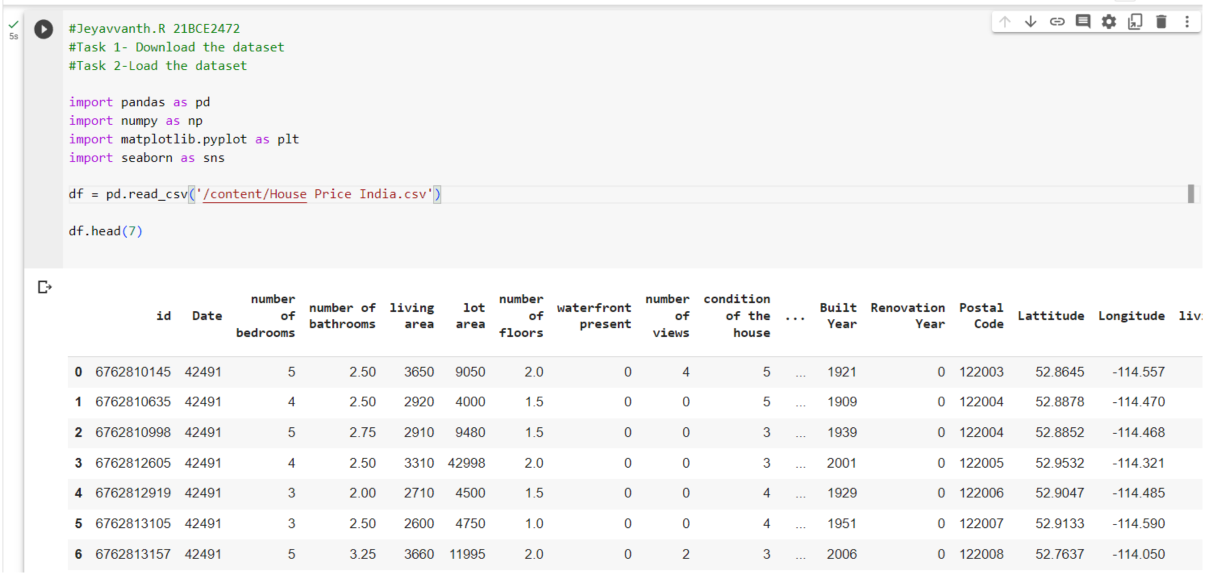
```
#Jeyavvanth.R 21BCE2472
#Task 1- Download the dataset
#Task 2-Load the dataset

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('/content/House Price India.csv')

df.head(7)
```

Output



	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renovation Year	Postal Code	Lattitude	Longitude	living_area_reno
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52.8645	-114.557	288
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52.8878	-114.470	247
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52.8852	-114.468	294
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52.9532	-114.321	335
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52.9047	-114.485	206
5	6762813105	42491	3	2.50	2600	4750	1.0	0	0	4	...	1951	0	122007	52.9133	-114.590	238
6	6762813157	42491	5	3.25	3660	11995	2.0	0	2	3	...	2006	0	122008	52.7637	-114.050	332

7 rows × 23 columns



```
#Jeyavvanth.R 21BCE2472
#Task 1- Download the dataset
#Task 2-Load the dataset

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('/content/House Price India.csv')

df.head(7)
```

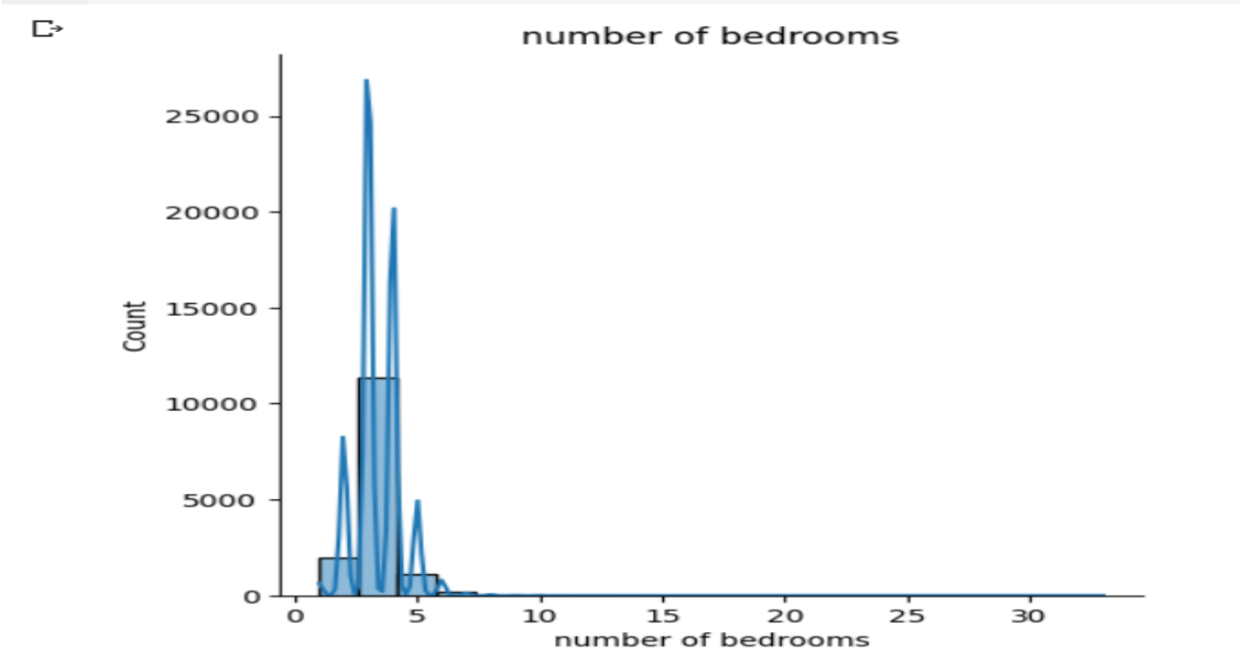
	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renovation Year	Postal Code	Lattitude	Longitude	liv
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52.8645	-114.557	
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52.8878	-114.470	
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52.8852	-114.468	
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52.9532	-114.321	
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52.9047	-114.485	
5	6762813105	42491	3	2.50	2600	4750	1.0	0	0	4	...	1951	0	122007	52.9133	-114.590	
6	6762813157	42491	5	3.25	3660	11995	2.0	0	2	3	...	2006	0	122008	52.7637	-114.050	

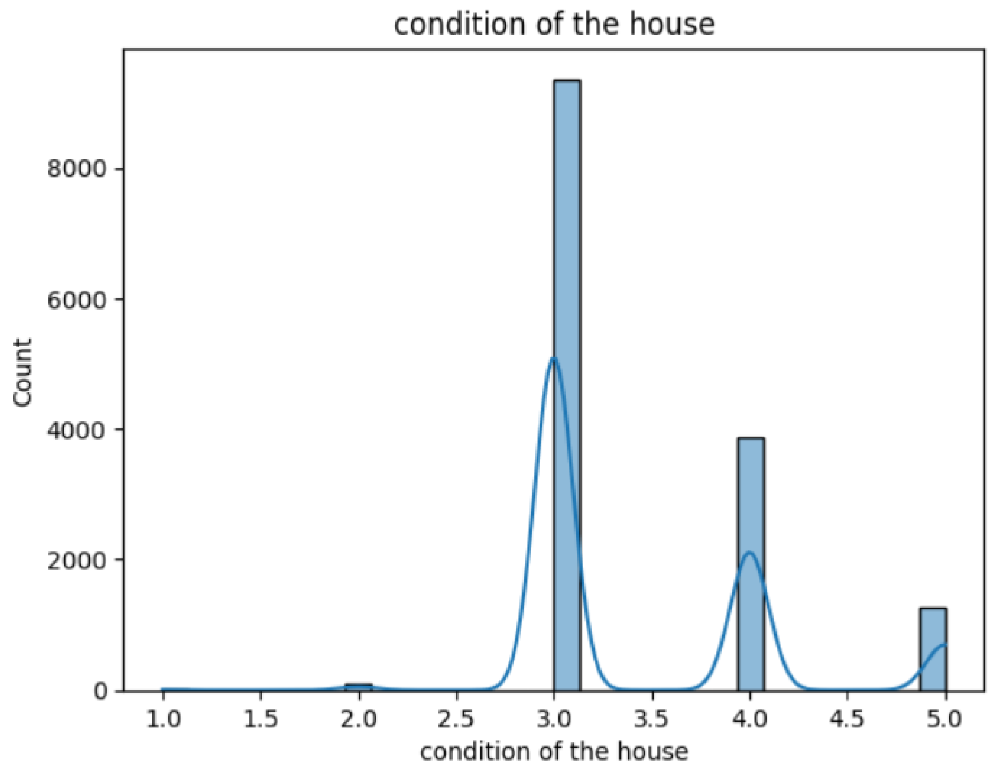
Task-3.1

Code

```
#Univariate analysis
#Jeyavvanth.R 21bce2472
#Task 3-Perform the below visualisation
#Task 3.1) Univariate Analysis

# Univariate Analysis for numerical column 'number of bedrooms'
sns.displot(df['number of bedrooms'], bins=20, kde=True)
plt.title('number of bedrooms')
plt.show()

# Univariate Analysis for column 'id'
sns.histplot(df['condition of the house'],bins=30, kde=True)
plt.title('condition of the house')
plt.show()
```

Output



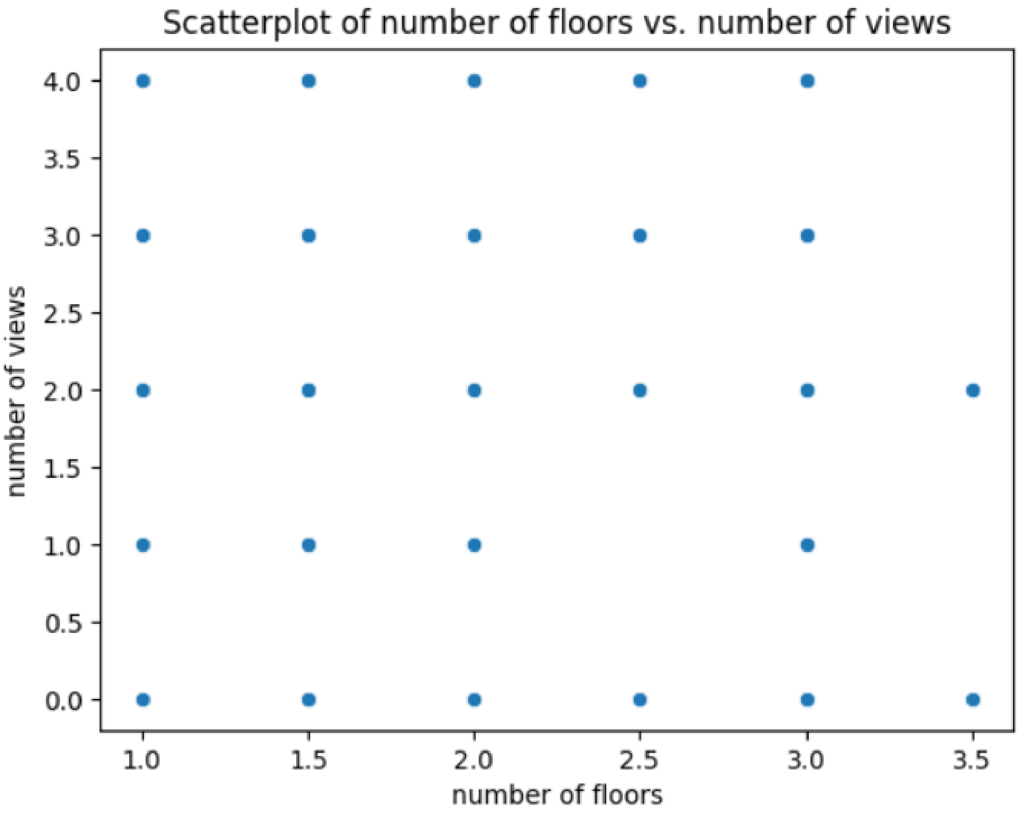


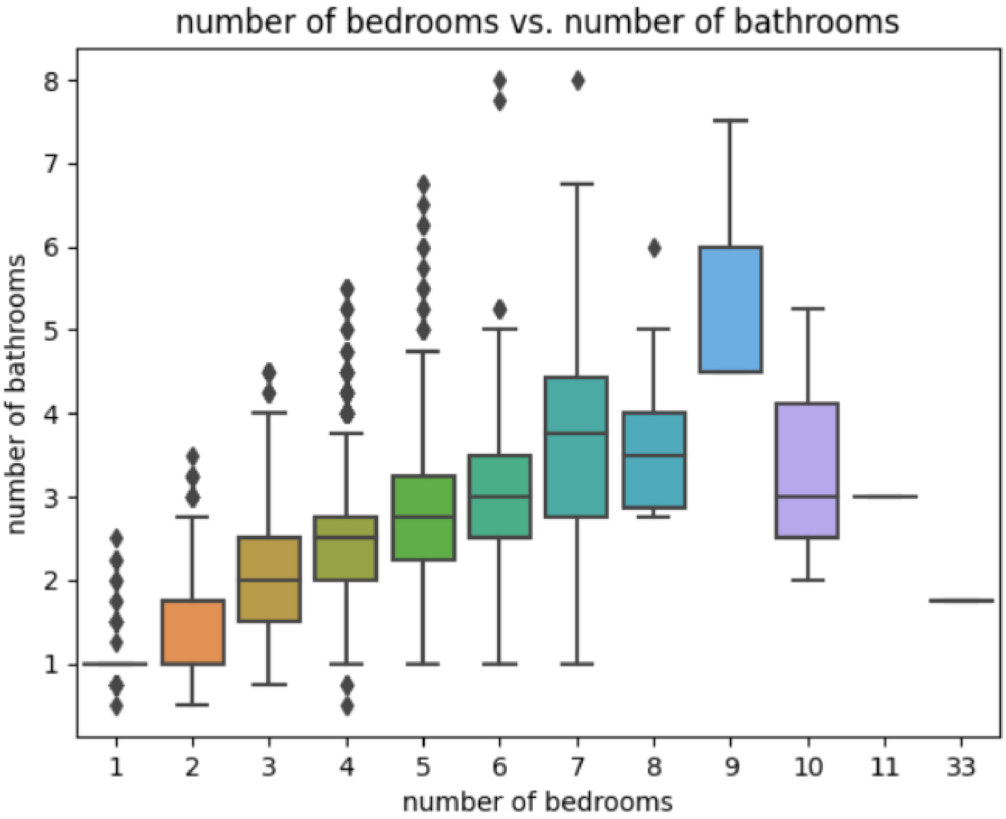
Task-3.2

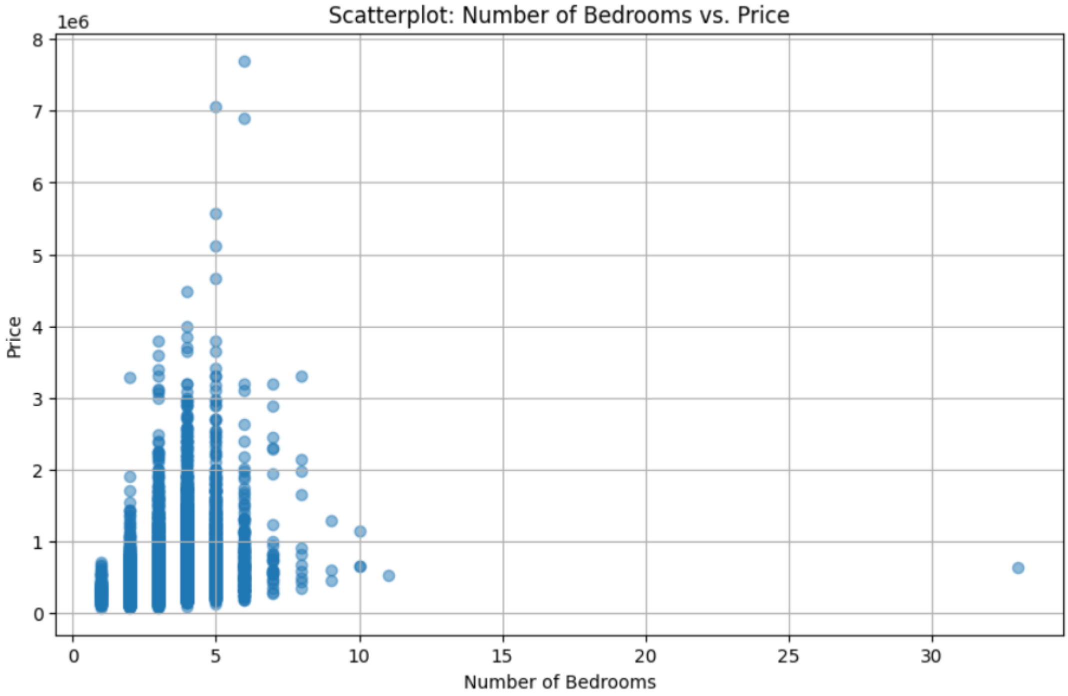
Code

```
#Jeyavvanth.R 21bce2472
#Task 3.2)Bivariate analysis

# Bivariate Analysis: number of floors vs. number of views
sns.scatterplot(x='number of floors', y='number of views', data=df)
plt.title('Scatterplot of number of floors vs. number of views')
plt.show()

# Bivariate Analysis number of bedrooms vs. number of bathrooms
sns.boxplot(x='number of bedrooms', y='number of bathrooms', data=df)
plt.title('number of bedrooms vs. number of bathrooms')
plt.show()

# Bivariate Analysis number of bedrooms vs. Price
bedrooms = df['number of bedrooms']
price = df['Price']
plt.figure(figsize=(10, 6))
plt.scatter(bedrooms, price, alpha=0.5)
plt.title('Scatterplot: Number of Bedrooms vs. Price')
plt.xlabel('Number of Bedrooms')
plt.ylabel('Price')
plt.grid(True)
plt.show()
```

Output




number of bedrooms vs. number of bathrooms
number of bathrooms
number of bedrooms
1
2
3
4
5
6
7
8
9
10
11
33

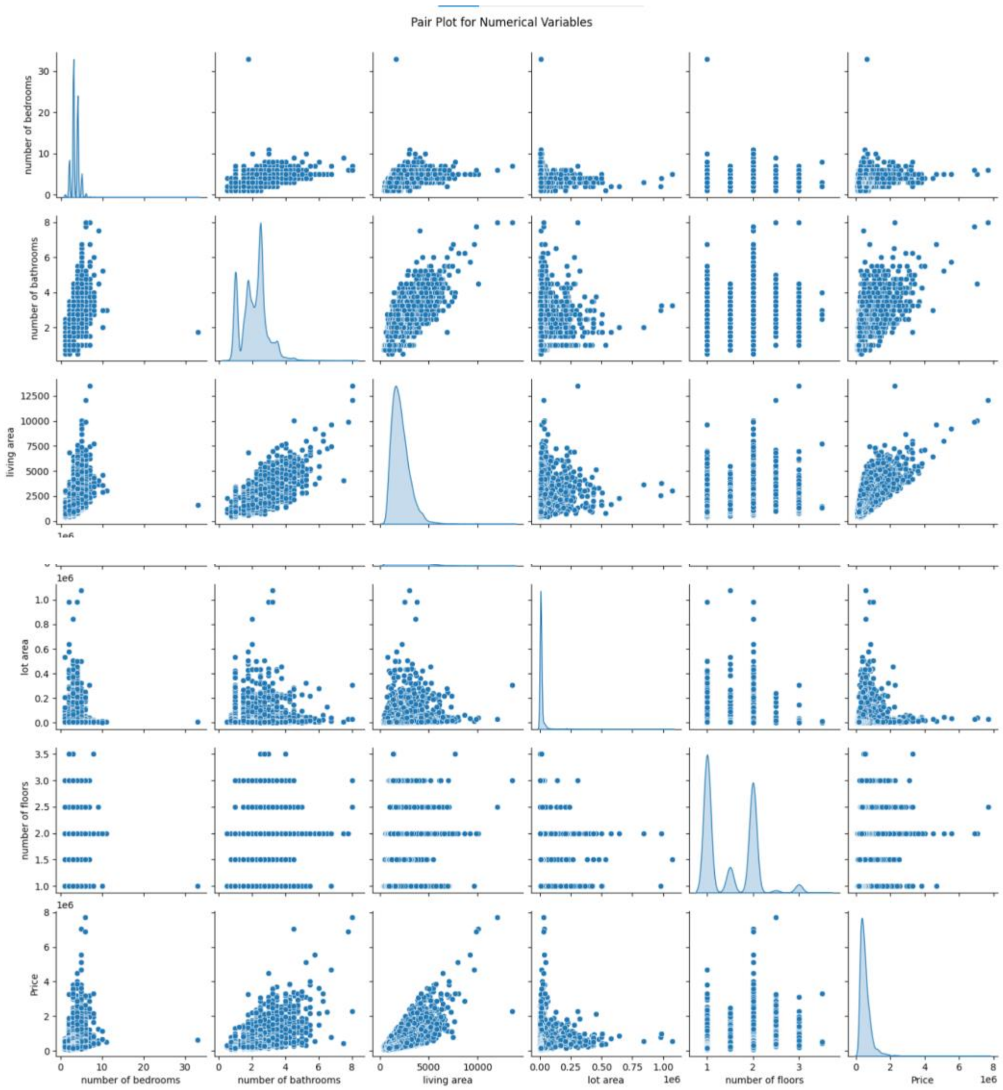

Scatterplot: Number of Bedrooms vs. Price
1e6
Price
Number of Bedrooms
0
5
10
15
20
25
30
1
2
3
4
5
6
7
8

Task-3.3

Code

```
#Jeyavvanth.R 21bce2472
#Task 3.3)Multivariate Analysis

numerical_vars = ['number of bedrooms', 'number of bathrooms', 'living
area', 'lot area', 'number of floors', 'Price']
sns.pairplot(df[numerical_vars], diag_kind='kde')
plt.suptitle("Pair Plot for Numerical Variables", y=1.02)
plt.show()
```

Output

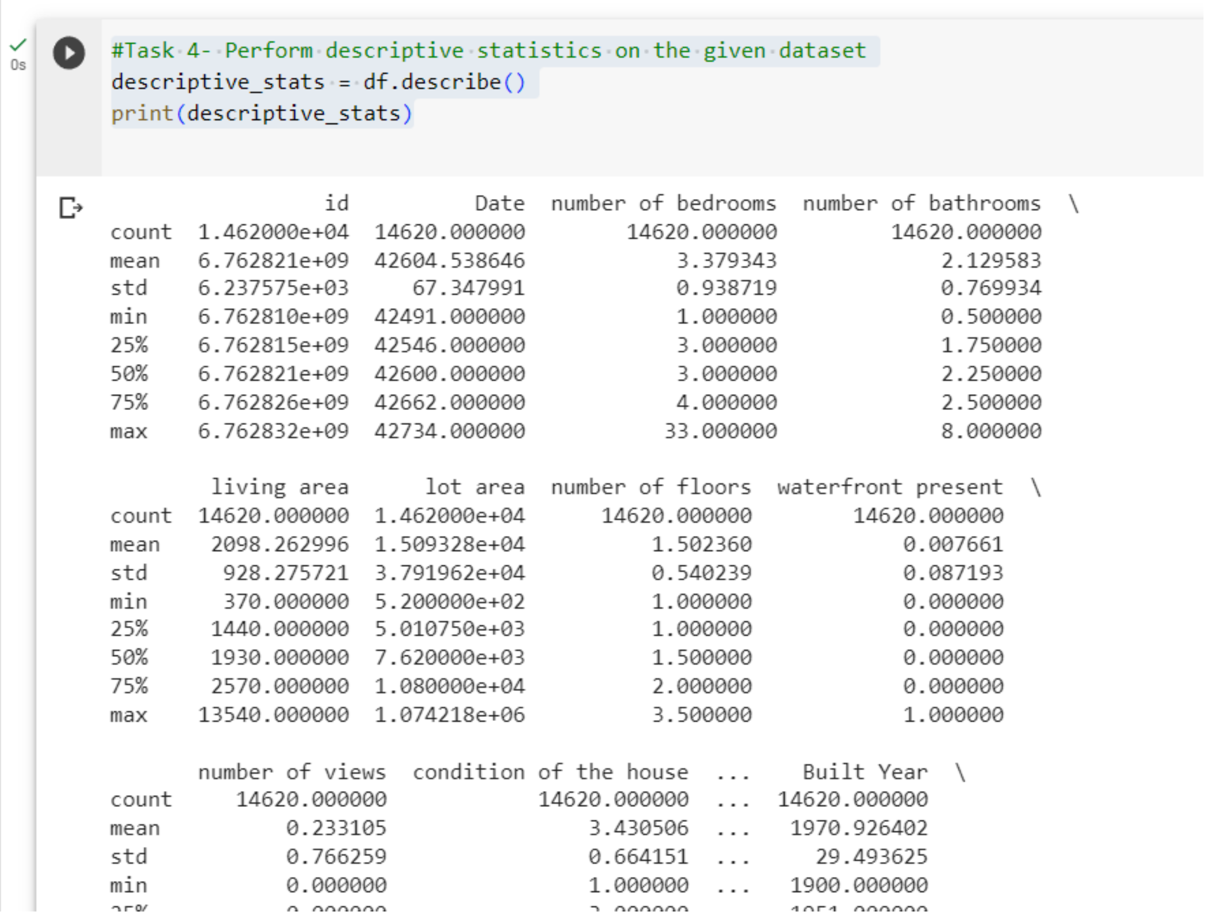


Task- 4

Code

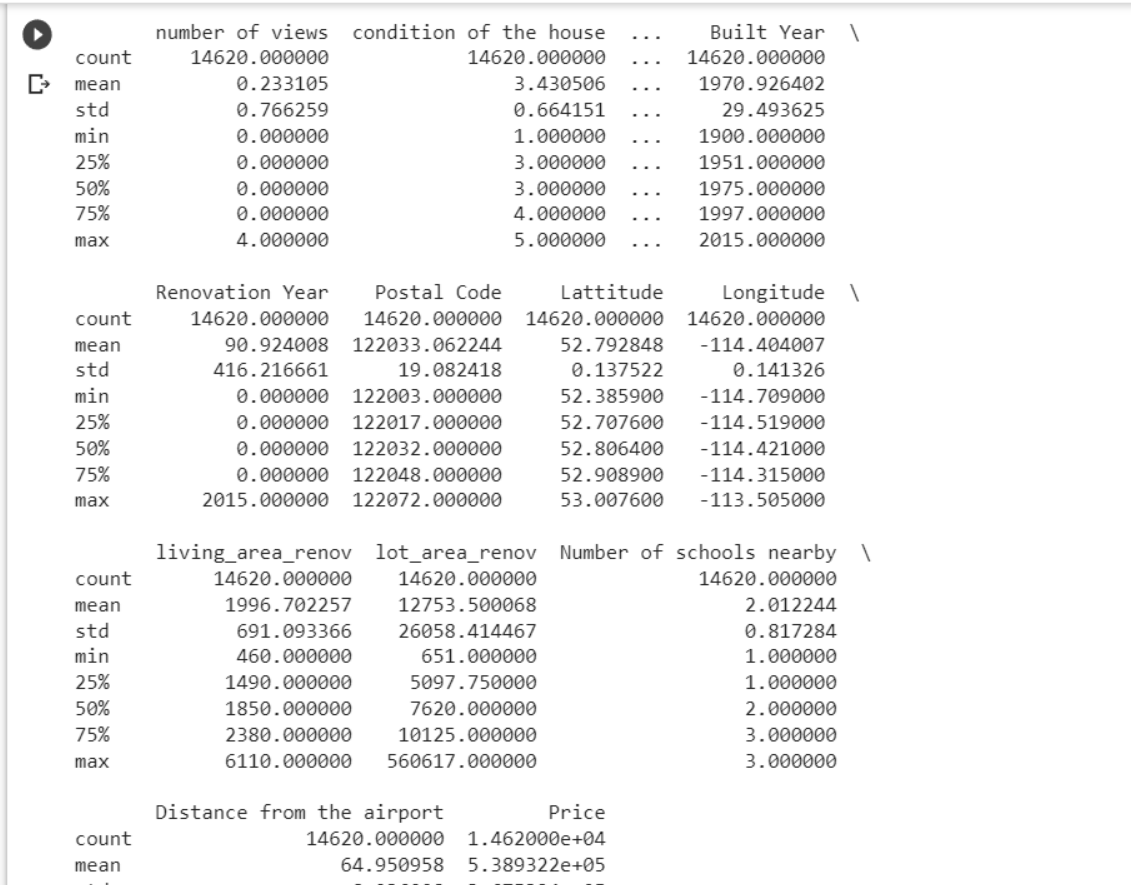
```
#Jeyavvanth.R 21BCE2472
#Task 4- Perform descriptive statistics on the given dataset
descriptive_stats = df.describe()
print(descriptive_stats)
```

Output

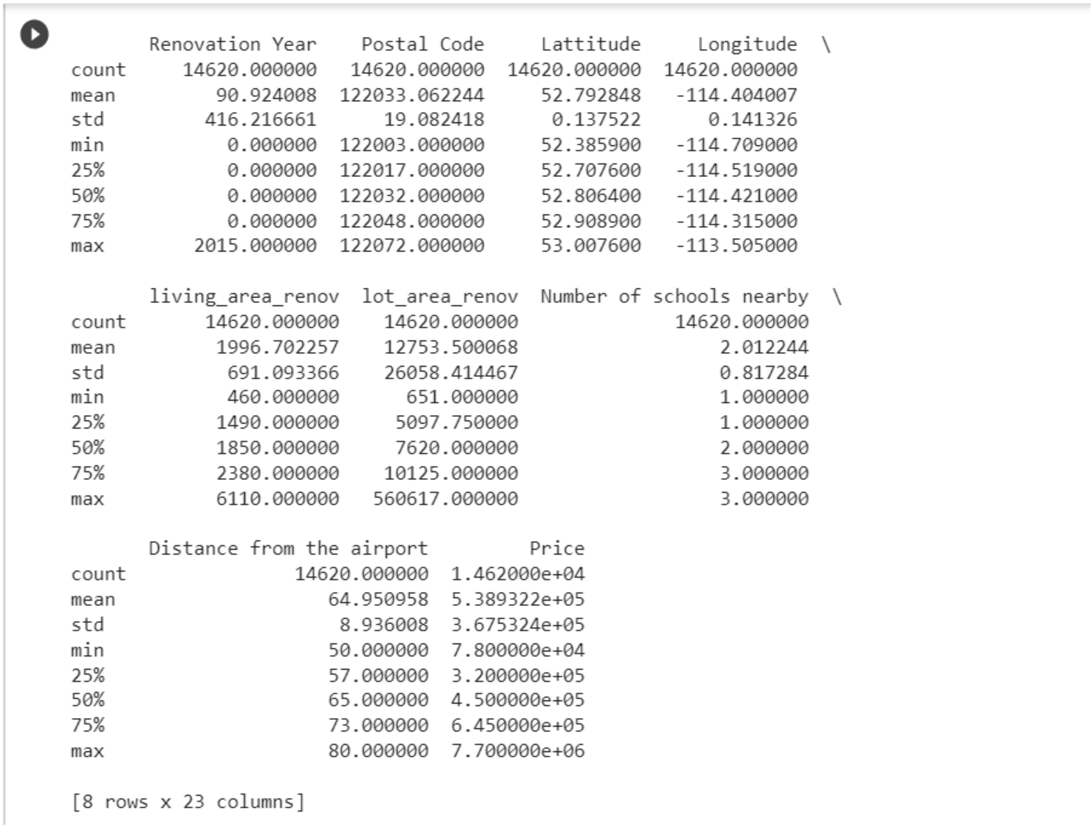


```
#Task 4- Perform descriptive statistics on the given dataset
descriptive_stats = df.describe()
print(descriptive_stats)
```

```
                 id          Date  number of bedrooms  number of bathrooms  \
count  1.462000e+04  14620.000000        14620.000000         14620.000000   
mean   6.762821e+09  42604.538646            3.379343             2.129583   
std    6.237575e+03     67.347991            0.938719             0.769934   
min    6.762810e+09  42491.000000            1.000000             0.500000   
25%    6.762815e+09  42546.000000            3.000000             1.750000   
50%    6.762821e+09  42600.000000            3.000000             2.250000   
75%    6.762826e+09  42662.000000            4.000000             2.500000   
max    6.762832e+09  42734.000000           33.000000             8.000000   

        living area     lot area  number of floors  waterfront present  \
count  14620.000000  1.462000e+04      14620.000000        14620.000000   
mean    2098.262996  1.509328e+04          1.502360            0.007661   
std      928.275721  3.791962e+04          0.540239            0.087193   
min      370.000000  5.200000e+02          1.000000            0.000000   
25%     1440.000000  5.010750e+03          1.000000            0.000000   
50%     1930.000000  7.620000e+03          1.500000            0.000000   
75%     2570.000000  1.080000e+04          2.000000            0.000000   
max    13540.000000  1.074218e+06          3.500000            1.000000   

       number of views  condition of the house  ...    Built Year  \
count     14620.000000            14620.000000  ...  14620.000000   
mean          0.233105                3.430506  ...   1970.926402   
std           0.766259                0.664151  ...     29.493625   
min           0.000000                1.000000  ...   1900.000000   
25%           0.000000                3.000000  ...   1951.000000   
```



```
       number of views  condition of the house  ...    Built Year  \
count     14620.000000            14620.000000  ...  14620.000000
mean          0.233105                3.430506  ...   1970.926402
std           0.766259                0.664151  ...     29.493625
min           0.000000                1.000000  ...   1900.000000
25%           0.000000                3.000000  ...   1951.000000
50%           0.000000                3.000000  ...   1975.000000
75%           0.000000                4.000000  ...   1997.000000
max           4.000000                5.000000  ...   2015.000000

       Renovation Year   Postal Code     Lattitude     Longitude  \
count     14620.000000  14620.000000  14620.000000  14620.000000
mean         90.924008 122033.062244     52.792848   -114.404007
std         416.216661     19.082418      0.137522      0.141326
min           0.000000 122003.000000     52.385900   -114.709000
25%           0.000000 122017.000000     52.707600   -114.519000
50%           0.000000 122032.000000     52.806400   -114.421000
75%           0.000000 122048.000000     52.908900   -114.315000
max        2015.000000 122072.000000     53.007600   -113.505000

       living_area_renov  lot_area_renov  Number of schools nearby  \
count       14620.000000    14620.000000              14620.000000
mean         1996.702257    12753.500068                  2.012244
std           691.093366    26058.414467                  0.817284
min           460.000000      651.000000                  1.000000
25%          1490.000000     5097.750000                  1.000000
50%          1850.000000     7620.000000                  2.000000
75%          2380.000000    10125.000000                  3.000000
max          6110.000000   560617.000000                  3.000000

       Distance from the airport         Price
count               14620.000000  1.462000e+04
mean                   64.950958  5.389322e+05
```



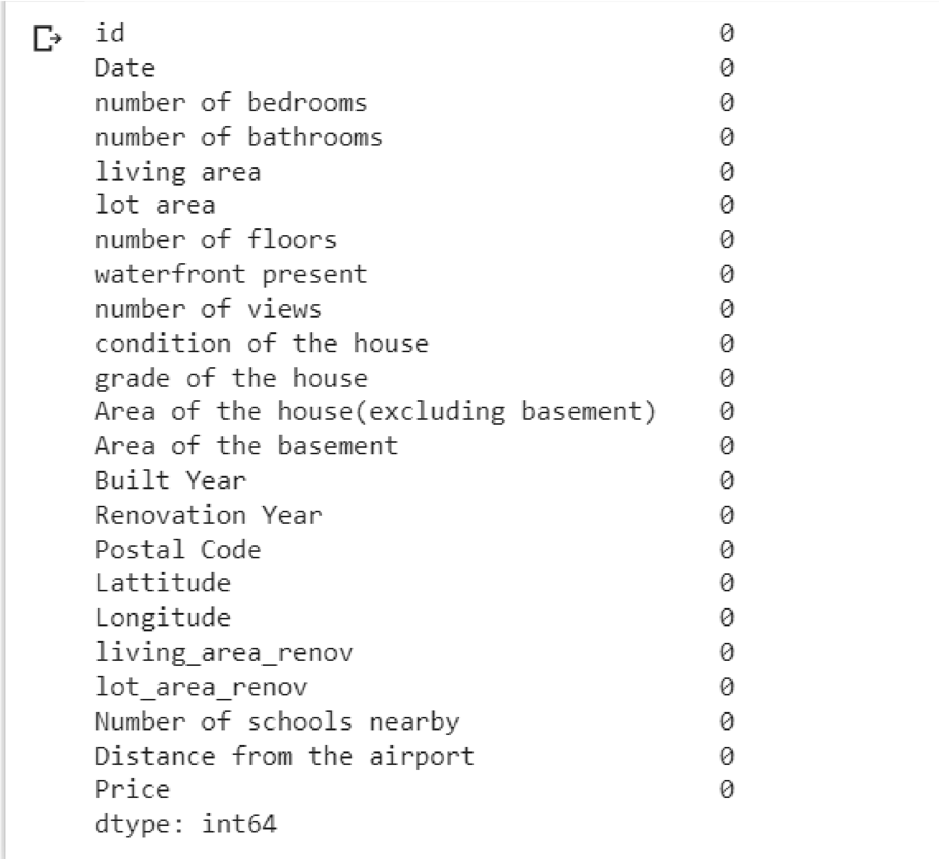
```
       Renovation Year   Postal Code     Lattitude     Longitude  \
count     14620.000000  14620.000000  14620.000000  14620.000000
mean         90.924008 122033.062244     52.792848   -114.404007
std         416.216661     19.082418      0.137522      0.141326
min           0.000000 122003.000000     52.385900   -114.709000
25%           0.000000 122017.000000     52.707600   -114.519000
50%           0.000000 122032.000000     52.806400   -114.421000
75%           0.000000 122048.000000     52.908900   -114.315000
max        2015.000000 122072.000000     53.007600   -113.505000

       living_area_renov  lot_area_renov  Number of schools nearby  \
count       14620.000000    14620.000000              14620.000000
mean         1996.702257    12753.500068                  2.012244
std           691.093366    26058.414467                  0.817284
min           460.000000      651.000000                  1.000000
25%          1490.000000     5097.750000                  1.000000
50%          1850.000000     7620.000000                  2.000000
75%          2380.000000    10125.000000                  3.000000
max          6110.000000   560617.000000                  3.000000

       Distance from the airport         Price
count               14620.000000  1.462000e+04
mean                   64.950958  5.389322e+05
std                     8.936008  3.675324e+05
min                    50.000000  7.800000e+04
25%                    57.000000  3.200000e+05
50%                    65.000000  4.500000e+05
75%                    73.000000  6.450000e+05
max                    80.000000  7.700000e+06

[8 rows x 23 columns]
```

Task- 5

Code

```
#Jeyavvanth.R 21BCE2472
#Task 5- Handle missing values
print(df.isna().sum())

#Simulation of how to replace missing values

#data['age'].fillna(data['age'].mean(), inplace=True)
#data['gender'].fillna(data['gender'].mode()[0], inplace=True)
```

Output



```
id                                      0
Date                                    0
number of bedrooms                      0
number of bathrooms                     0
living area                             0
lot area                                0
number of floors                        0
waterfront present                      0
number of views                         0
condition of the house                  0
grade of the house                      0
Area of the house(excluding basement)   0
Area of the basement                    0
Built Year                              0
Renovation Year                         0
Postal Code                             0
Lattitude                               0
Longitude                               0
living_area_renov                       0
lot_area_renov                          0
Number of schools nearby                0
Distance from the airport               0
Price                                   0
dtype: int64
```

There are no missing values so cannot replace missing values using mode or median or mean.If missing values present we can use code like:

data['age'].fillna(data['age'].mean(), inplace=True)

data['gender'].fillna(data['gender'].mode()[0], inplace=True)