

```

import warnings
warnings.filterwarnings('ignore')

# Import the numpy and pandas package

import numpy as np
import pandas as pd


# Data Visualisation

import matplotlib.pyplot as plt
import seaborn as sns


housing = pd.DataFrame(pd.read_csv("/content/House price prediction.csv"))

# Check the head of the dataset
housing.head()

```



	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furn
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	



```

housing.shape

(545, 13)

```

```

housing.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
#   Column              Non-Null Count  Dtype
---  ---              -
0   price               545 non-null   int64
1   area                545 non-null   int64
2   bedrooms            545 non-null   int64
3   bathrooms            545 non-null   int64
4   stories              545 non-null   int64
5   mainroad             545 non-null   object
6   guestroom            545 non-null   object
7   basement             545 non-null   object
8   hotwaterheating      545 non-null   object
9   airconditioning      545 non-null   object
10  parking              545 non-null   int64
11  prefarea             545 non-null   object
12  furnishingstatus     545 non-null   object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB

```

```

housing.describe()

```



	price	area	bedrooms	bathrooms	stories	parking
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	3.000000

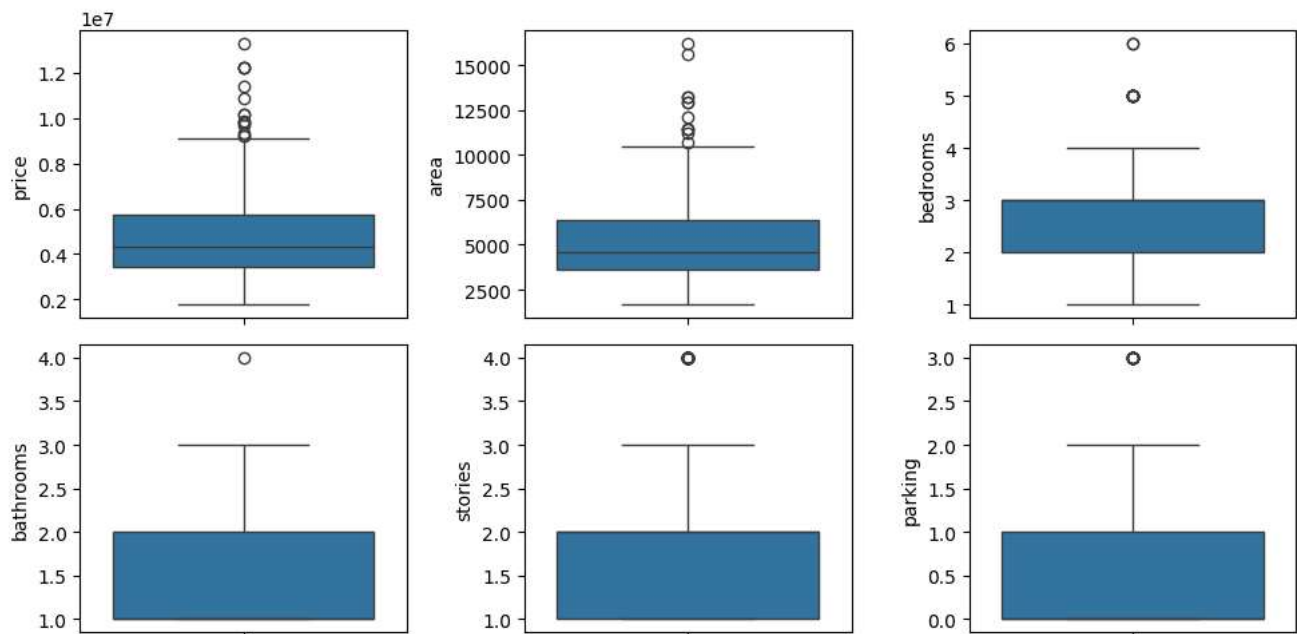
```
# Checking Null values
housing.isnull().sum()*100/housing.shape[0]
# There are no NULL values in the dataset, hence it is clean.
```



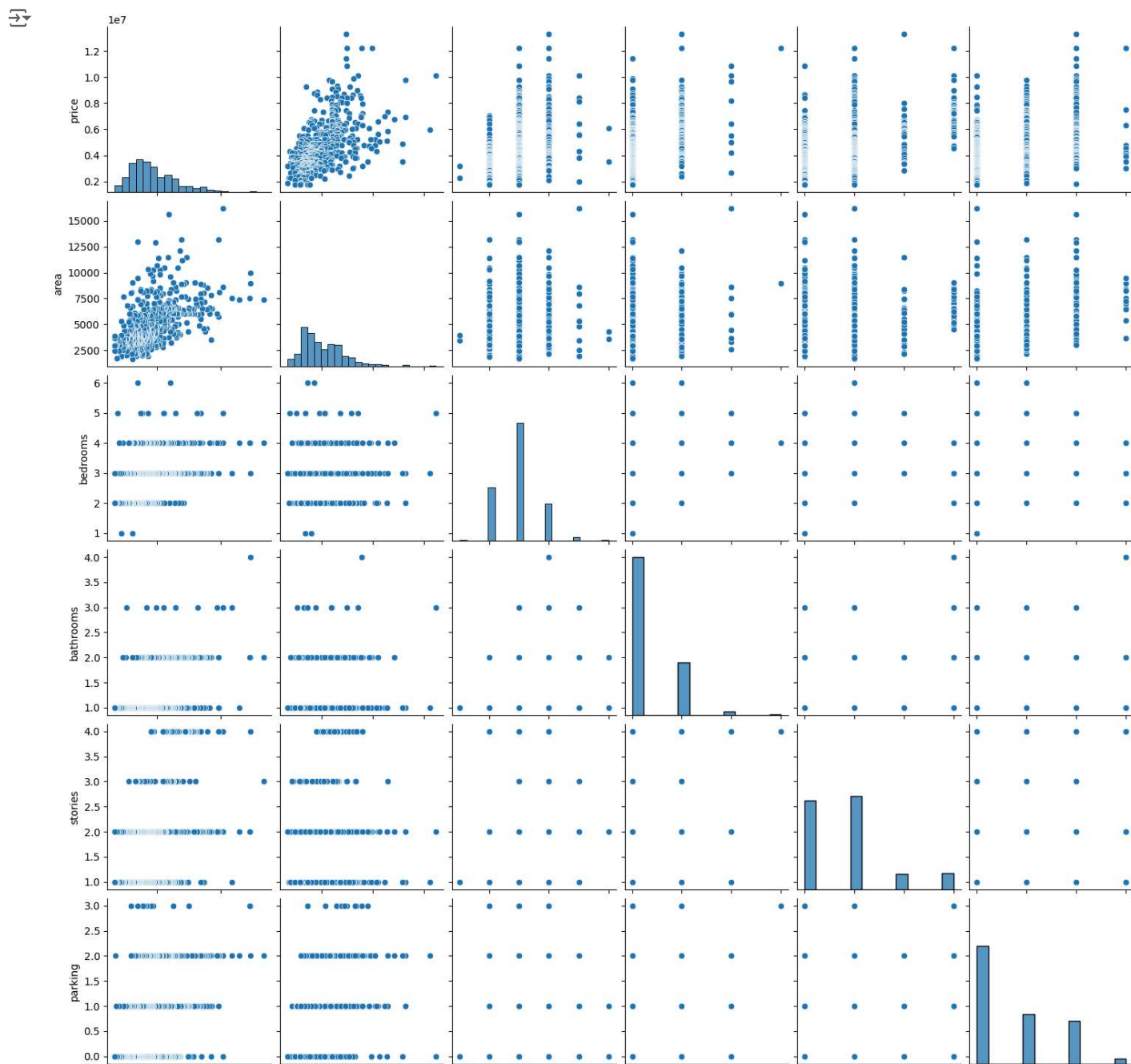
```
price      0.0
area       0.0
bedrooms   0.0
bathrooms  0.0
stories    0.0
mainroad   0.0
guestroom  0.0
basement   0.0
hotwaterheating 0.0
airconditioning 0.0
parking    0.0
prefarea   0.0
furnishingstatus 0.0
dtype: float64
```

```
# Outlier Analysis
fig, axs = plt.subplots(2,3, figsize = (10,5))
plt1 = sns.boxplot(housing['price'], ax = axs[0,0])
plt2 = sns.boxplot(housing['area'], ax = axs[0,1])
plt3 = sns.boxplot(housing['bedrooms'], ax = axs[0,2])
plt1 = sns.boxplot(housing['bathrooms'], ax = axs[1,0])
plt2 = sns.boxplot(housing['stories'], ax = axs[1,1])
plt3 = sns.boxplot(housing['parking'], ax = axs[1,2])
```

```
plt.tight_layout()
```



```
sns.pairplot(housing)
plt.show()
```



```
# Get the dummy variables for the feature 'furnishingstatus' and store it in a new variable - 'status'
status = pd.get_dummies(housing['furnishingstatus'])
# Check what the dataset 'status' looks like
status.head()
```

	furnished	semi-furnished	unfurnished
0	True	False	False
1	True	False	False
2	False	True	False
3	True	False	False
4	True	False	False

```
# Let's drop the first column from status df using 'drop_first = True'

status = pd.get_dummies(housing['furnishingstatus'], drop_first = True)
# Add the results to the original housing dataframe

housing = pd.concat([housing, status], axis = 1)
# Now let's see the head of our dataframe.

housing.head()
```



	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwater
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	