PREDICTIVE MODELLING WITH LINEAR REGRESSION

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
data = pd.read_csv(r'/content/House Price India.csv')
data
```

	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
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14615	6762830250	42734	2	1.50	1556	20000	1.0	0	0	4	...	1957	0	122066
14616	6762830339	42734	3	2.00	1680	7000	1.5	0	0	4	...	1968	0	122072
14617	6762830618	42734	2	1.00	1070	6120	1.0	0	0	3	...	1962	0	122056
14618	6762830709	42734	4	1.00	1030	6621	1.0	0	0	4	...	1955	0	122042
14619	6762831463	42734	3	1.00	900	4770	1.0	0	0	3	...	1969	2009	122018

14620 rows × 23 columns

```
data.head()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0

5 rows × 23 columns

```
data.tail()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	num vi
14615	6762830250	42734	2	1.5	1556	20000	1.0	0	
14616	6762830339	42734	3	2.0	1680	7000	1.5	0	
14617	6762830618	42734	2	1.0	1070	6120	1.0	0	
14618	6762830709	42734	4	1.0	1030	6621	1.0	0	
14619	6762831463	42734	3	1.0	900	4770	1.0	0	

5 rows × 23 columns

```
data.shape
```

```
(14620, 23)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   id                                     14620 non-null  int64
 1   Date                                   14620 non-null  int64
 2   number of bedrooms                     14620 non-null  int64
 3   number of bathrooms                    14620 non-null  float64
 4   living area                            14620 non-null  int64
 5   lot area                               14620 non-null  int64
 6   number of floors                       14620 non-null  float64
 7   waterfront present                     14620 non-null  int64
 8   number of views                        14620 non-null  int64
 9   condition of the house                 14620 non-null  int64
 10  grade of the house                     14620 non-null  int64
 11  Area of the house(excluding basement)  14620 non-null  int64
 12  Area of the basement                   14620 non-null  int64
 13  Built Year                             14620 non-null  int64
 14  Renovation Year                        14620 non-null  int64
 15  Postal Code                            14620 non-null  int64
 16  Lattitude                              14620 non-null  float64
 17  Longitude                              14620 non-null  float64
 18  living_area_renov                      14620 non-null  int64
 19  lot_area_renov                         14620 non-null  int64
 20  Number of schools nearby               14620 non-null  int64
 21  Distance from the airport              14620 non-null  int64
 22  Price                                  14620 non-null  int64
dtypes: float64(4), int64(19)
memory usage: 2.6 MB
```

```
data.describe()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot ar
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+

8 rows × 23 columns

```
data.columns
```

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

```
data.isnull()
```

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	cc
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
14615	False	False	False	False	False	False	False	False	False	
14616	False	False	False	False	False	False	False	False	False	
14617	False	False	False	False	False	False	False	False	False	
14618	False	False	False	False	False	False	False	False	False	
14619	False	False	False	False	False	False	False	False	False	

14620 rows × 23 columns

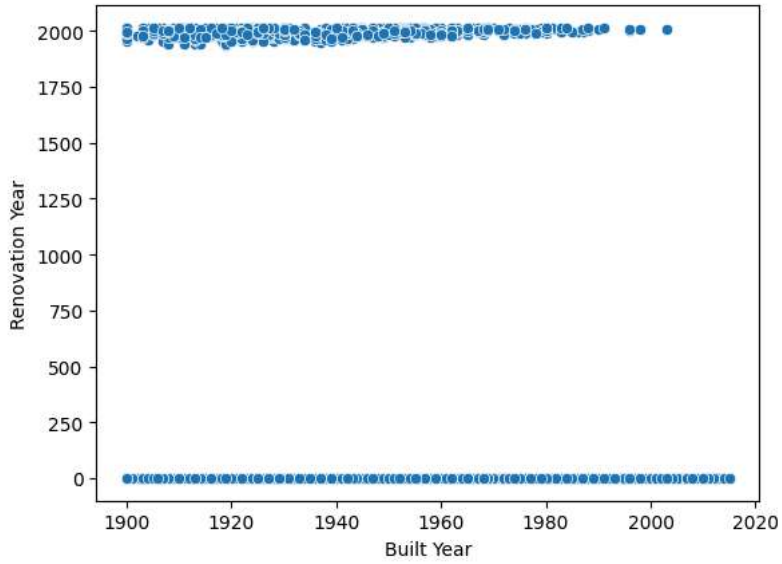
```
data.fillna(77)
```

front esent	number of views	condition of the house	...	Built Year	Renovation Year	Postal Code
0	4	5	...	1921	0	122003
0	0	5	...	1909	0	122004
0	0	3	...	1939	0	122004
0	0	3	...	2001	0	122005
0	0	4	...	1929	0	122006
...	...	...	...	...	...	...
0	0	4	...	1957	0	122066
0	0	4	...	1968	0	122072
0	0	3	...	1962	0	122056
0	0	4	...	1955	0	122042
0	0	3	...	1969	2009	122018

Double-click (or enter) to edit

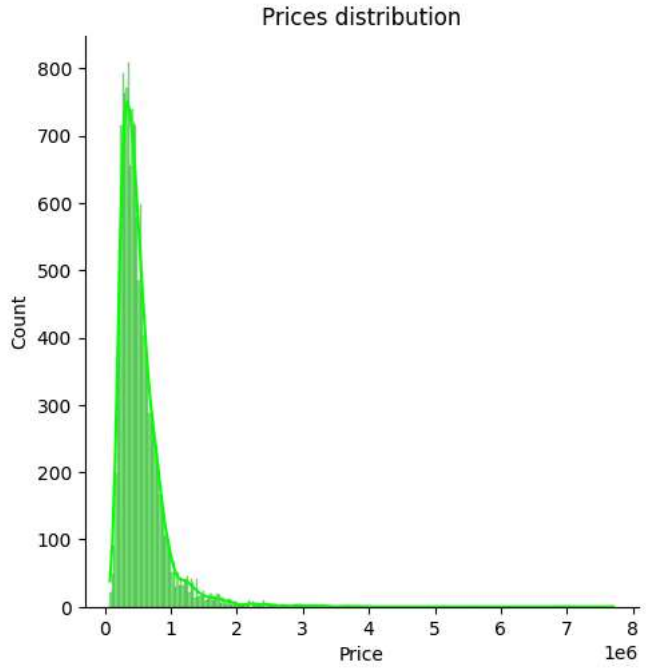
```
import seaborn as sns
sns.scatterplot(data,x='Built Year',y='Renovation Year')
```

```
<Axes: xlabel='Built Year', ylabel='Renovation Year'>
```



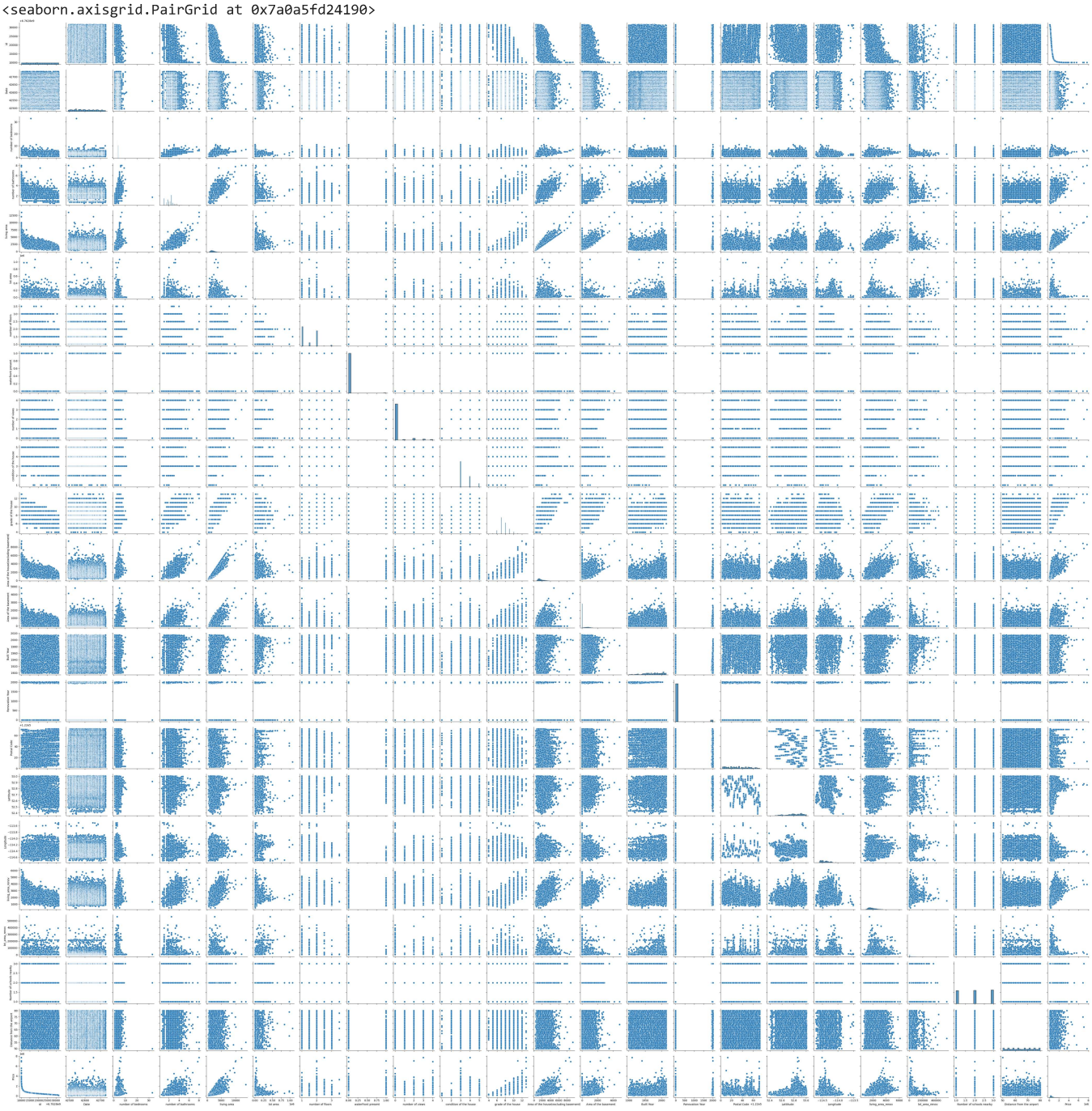

```
sns.displot(data.Price, kde = True, color='lime')
plt.title('Prices distribution')
```

```
Text(0.5, 1.0, 'Prices distribution')
```



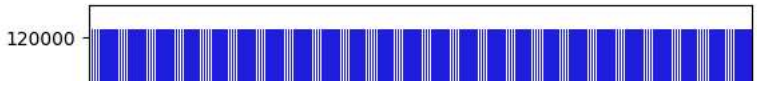

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7a0a5fd24190>
```



```
sns.barplot(data,x ='id', y='Postal Code', color ='blue')
```

```
<Axes: xlabel='id', ylabel='Postal Code'>
```




```
from sklearn.model_selection import train_test_split
```

```
train, test = train_test_split(data, test_size = 0.2)
```

```
x_train = train.iloc[:, :20].values
x_test = test.iloc[:, :20].values
```

```
y_train = train['Price'].values
y_test = test['Price'].values
```

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

```
model.fit(x_train, y_train)
```

```
LinearRegression
LinearRegression()
```

```
y_pred = model.predict(x_test)
y_pred
```

```
array([1090763.10681152,  897006.79806519,  636040.29742432, ...,
        256629.80529785,  581687.27770996,  106864.94985962])
```

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, mean_absolute_percentage_error, r2_score

print("MSE",round(mean_squared_error(y_test,y_pred), 3))
print("RMSE",round(np.sqrt(mean_squared_error(y_test,y_pred)), 3))
print("MAE",round(mean_absolute_error(y_test,y_pred), 3))
print("MAPE",round(mean_absolute_percentage_error(y_test,y_pred), 3))
print("R2 Score : ", round(r2_score(y_test,y_pred), 3))
```

```
MSE 33813459564.821
RMSE 183884.365
MAE 105833.586
MAPE 0.209
R2 Score :  0.735
```

```
model.score(x_test,y_test)
```