Praticle 1 : Data Preprocessing

In [ ]:
```
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder, StandardScaler
```

In [17]:
```
# Load the dataset
data = pd.read_csv('data_house.csv')
```

In [19]:
```
# Handling missing data
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
imputer.fit(data[['bedrooms', 'bathrooms']])
data[['bedrooms', 'bathrooms']] = imputer.transform(data[['bedrooms', 'ba
```

In [20]:
```
# Encoding the independent variable
floors_encoder = LabelEncoder()
data['floors'] = floors_encoder.fit_transform(data['floors'])

# Encoding the dependent variable
price_encoder = LabelEncoder()
data['price'] = price_encoder.fit_transform(data['price'])
```

In [22]:
```
# Feature Scaling
scaler = StandardScaler()
num_cols = ['bedrooms', 'bathrooms', 'floors']
data[num_cols] = scaler.fit_transform(data[num_cols])
data['price'] = scaler.fit_transform(data[['price']])
```

In [23]:
```
# Data Splitting
X = data.drop('price', axis=1)
y = data['price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
```

In [26]:
```
print(X)
```

```
       bedrooms  bathrooms    floors
0     -0.398737  -1.447464 -0.915427
1     -0.398737   0.175607  0.936506
2     -1.473959  -1.447464 -0.915427
3      0.676485   1.149449 -0.915427
4     -0.398737  -0.149007 -0.915427
...         ...        ...       ...
21608 -0.398737   0.500221  2.788439
21609  0.676485   0.500221  0.936506
21610 -1.473959  -1.772078  0.936506
21611 -0.398737   0.500221  0.936506
21612 -1.473959  -1.772078  0.936506

[21613 rows x 3 columns]
```

In [27]:
```
print(y)
```

```
0       -1.411276
1        0.321547
2       -1.579385
3        0.602062
4        0.198201
           ...
21608   -0.592611
21609   -0.366807
21610   -0.357854
21611   -0.366807
21612   -0.805484
Name: price, Length: 21613, dtype: float64
```

In [28]:
```
print(X_train)
```

```
       bedrooms  bathrooms    floors
6325  -0.398737  -0.473621 -0.915427
13473 -1.473959  -1.447464 -0.915427
17614 -0.398737  -1.447464  0.010539
16970 -0.398737   0.175607 -0.915427
20868 -1.473959   0.500221  0.936506
...         ...        ...       ...
11964 -0.398737  -0.798235 -0.915427
21575 -0.398737   0.500221  0.936506
5390  -0.398737   0.500221  0.936506
860   -2.549182  -1.772078 -0.915427
15795  0.676485   0.500221  0.936506

[17290 rows x 3 columns]
```

In [29]:
```
print(X_test)
```

```
       bedrooms  bathrooms    floors
735    0.676485   0.175607  0.936506
2830   1.751707   1.149449 -0.915427
4106   0.676485   0.500221  0.936506
16218 -0.398737   1.798677  0.936506
19964 -0.398737   0.500221  0.936506
...         ...        ...       ...
13674 -0.398737  -0.473621 -0.915427
20377 -0.398737   0.824835  0.936506
8805   0.676485   0.500221  0.936506
10168  0.676485  -0.473621 -0.915427
2522   0.676485   0.500221  0.936506

[4323 rows x 3 columns]
```

In [30]:
```
print(y_train)
```

```
6325    -0.805484
13473   -1.228245
17614   -1.377455
16970   -1.035267
20868    0.054959
           ...
11964   -0.502091
21575   -0.370786
5390     0.484683
860     -1.295887
15795   -0.858205
Name: price, Length: 17290, dtype: float64
```

In [31]:
```
print(y_test)
```

```
735     -0.565753
2830     1.402821
4106     1.686320
16218    1.970813
19964    0.973097
           ...
13674   -0.720932
20377    0.859697
8805    -1.058146
10168    0.605046
2522    -0.614495
Name: price, Length: 4323, dtype: float64
```

Steps Followed :

- Handling missing data by imputing the missing values in the 'bedrooms' and 'bathrooms' columns with their respective means.
- Encoding the categorical variable 'floors' using LabelEncoder to assign unique numeric labels to each category.
- Encoding the dependent variable 'price' using LabelEncoder to convert it into numeric labels.
- Performing feature scaling on the numerical columns 'bedrooms', 'bathrooms', 'floors', and the dependent variable 'price' using StandardScaler to standardize the values.
- Splitting the preprocessed data into training and test sets using train_test_split for further analysis or model training.

In [ ]: