

hsrhojwca

February 27, 2025

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: df=pd.read_csv("Housing.csv")
```

```
[3]: df
```

```
[3]:      price  area  bedrooms  bathrooms  stories  mainroad  guestroom  basement  \
0    1330000  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
..      ...  ...      ...      ...      ...      ...      ...      ...
540    1820000  3000         2          1         1         yes         no         yes
541    1767150  2400         3          1         1         no         no         no
542    1750000  3620         2          1         1         yes         no         no
543    1750000  2910         3          1         1         no         no         no
544    1750000  3850         3          1         2         yes         no         no

      hotwaterheating  airconditioning  parking  prefarea  furnishingstatus
0                no                yes         2        yes        furnished
1                no                yes         3         no        furnished
2                no                no         2        yes    semi-furnished
3                no                yes         3        yes        furnished
4                no                yes         2         no        furnished
..              ...              ...      ...      ...              ...
540                no                no         2         no        unfurnished
541                no                no         0         no    semi-furnished
542                no                no         0         no        unfurnished
543                no                no         0         no        furnished
544                no                no         0         no        unfurnished
```

[545 rows x 13 columns]

```
[5]: import pandas as pd

# Example: Loading from CSV
df = pd.read_csv("Housing.csv")

# Example: Loading from a Python object
class Housing:
    df = pd.read_csv("Housing.csv")

df = pd.DataFrame(Housing.df)
```

```
[8]: df.columns
```

```
[8]: Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
          'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
          'parking', 'prefarea', 'furnishingstatus'],
          dtype='object')
```

```
[15]: df.isnull().sum()
```

```
[15]: price          0
      area           0
      bedrooms       0
      bathrooms      0
      stories        0
      mainroad       0
      guestroom      0
      basement       0
      hotwaterheating 0
      airconditioning 0
      parking        0
      prefarea       0
      furnishingstatus 0
      dtype: int64
```

```
[16]: x = df.drop(['price'], axis=1)
      y = df['price']
```

```
[17]: from sklearn.model_selection import train_test_split
      xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size =0.
      ↪2, random_state = 0)
```

```
[19]: from sklearn.preprocessing import LabelEncoder
      le = LabelEncoder()
      df['mainroad'] = le.fit_transform(df['mainroad'])
      newdf=df
```

```
[20]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['guestroom'] = le.fit_transform(df['guestroom'])
newdf=df
```

```
[21]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['basement'] = le.fit_transform(df['basement'])
newdf=df
```

```
[22]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['hotwaterheating'] = le.fit_transform(df['hotwaterheating'])
newdf=df
```

```
[23]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['airconditioning'] = le.fit_transform(df['airconditioning'])
newdf=df
```

```
[24]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['prefarea'] = le.fit_transform(df['prefarea'])
newdf=df
```

```
[25]: df
```

```
[25]:
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	\
0	13300000	7420	4	2	3	1	0	
1	12250000	8960	4	4	4	1	0	
2	12250000	9960	3	2	2	1	0	
3	12215000	7500	4	2	2	1	0	
4	11410000	7420	4	1	2	1	1	
..		
540	1820000	3000	2	1	1	1	0	
541	1767150	2400	3	1	1	0	0	
542	1750000	3620	2	1	1	1	0	
543	1750000	2910	3	1	1	0	0	
544	1750000	3850	3	1	2	1	0	
	basement	hotwaterheating	airconditioning	parking	prefarea	\		
0	0	0		1	2	1		
1	0	0		1	3	0		
2	1	0		0	2	1		
3	1	0		1	3	1		
4	1	0		1	2	0		
..			

540	1	0	0	2	0
541	0	0	0	0	0
542	0	0	0	0	0
543	0	0	0	0	0
544	0	0	0	0	0

```

furnishingstatus
0      furnished
1      furnished
2    semi-furnished
3      furnished
4      furnished
..      ...
540    unfurnished
541  semi-furnished
542    unfurnished
543      furnished
544    unfurnished

```

[545 rows x 13 columns]

```
[26]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['furnishingstatus'] = le.fit_transform(df['furnishingstatus'])
newdf=df
```

```
[27]: df
```

```
[27]:
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	\
0	13300000	7420	4	2	3	1	0	
1	12250000	8960	4	4	4	1	0	
2	12250000	9960	3	2	2	1	0	
3	12215000	7500	4	2	2	1	0	
4	11410000	7420	4	1	2	1	1	
..		
540	1820000	3000	2	1	1	1	0	
541	1767150	2400	3	1	1	0	0	
542	1750000	3620	2	1	1	1	0	
543	1750000	2910	3	1	1	0	0	
544	1750000	3850	3	1	2	1	0	

	basement	hotwaterheating	airconditioning	parking	prefarea	\
0	0	0	1	2	1	
1	0	0	1	3	0	
2	1	0	0	2	1	
3	1	0	1	3	1	
4	1	0	1	2	0	

```

..      ...      ...      ...      ...
540      1      0      0      2      0
541      0      0      0      0      0
542      0      0      0      0      0
543      0      0      0      0      0
544      0      0      0      0      0

```

```

furnishingstatus
0      0
1      0
2      1
3      0
4      0
..      ...
540      2
541      1
542      2
543      0
544      2

```

[545 rows x 13 columns]

```
[28]: from sklearn.model_selection import train_test_split
```

```
[29]: from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(x, y, test_size=0.
↳4, random_state=10)
```

```
[30]: from sklearn.linear_model import LinearRegression
```

```
[32]: print(df.dtypes)
```

```

price      int64
area       int64
bedrooms   int64
bathrooms  int64
stories    int64
mainroad   int32
guestroom  int32
basement   int32
hotwaterheating int32
airconditioning int32
parking     int64
prefarea    int32
furnishingstatus int32
dtype: object

```

```
[33]: print(X_train.head()) # Check if any columns are still categorical
```

	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	\
109	6615	4	2	2	yes	yes	no	
155	6100	3	2	1	yes	no	yes	
389	4600	4	1	2	yes	no	no	
110	6600	3	1	1	yes	yes	yes	
38	6000	3	1	4	yes	yes	no	

	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
109	yes	no	1	no	semi-furnished
155	no	no	2	yes	furnished
389	no	no	0	no	semi-furnished
110	no	no	2	yes	semi-furnished
38	no	yes	2	no	unfurnished

```
[34]: X_train = pd.get_dummies(X_train, drop_first=True)
```

```
[36]: from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

X = df.drop('price', axis=1)
y = df['price']

X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42)

X_train = pd.get_dummies(X_train, drop_first=True)
X_test = pd.get_dummies(X_test, drop_first=True)

lm = LinearRegression()
lm.fit(X_train, Y_train)
```

```
[36]: LinearRegression()
```

```
[37]: Y_pred = lm.predict(X_test)
print("Predictions:", Y_pred)
```

```
Predictions: [5203691.70963178 7257004.02115476 3062828.59668171
4559591.65374424
3332932.30559782 3563080.67918997 5645466.31219972 6413979.66873635
2755831.54819 2668938.66075228 9570600.29915353 2827431.50860062
3195686.2583409 3352263.99438471 3713879.49996131 5301088.24435749
2987920.2666968 4810799.8212371 4383031.70489929 3525092.18938646
5796259.50068013 5840000.70299301 2760214.608641 4762590.14920607
5204755.73895206 7515542.71619025 3254681.68956382 5236164.45964444
8178523.16820284 3434166.15675649 6443921.58767582 3346004.77919184]
```

```

6742324.74004133 4154936.84088665 3589152.47491252 5788125.92515323
4768370.18154076 4391684.04193172 3217657.04549935 4638196.61928878
4522160.27786714 3541284.06127245 7238136.11941171 4021515.68926614
3701978.76822757 4298879.55563098 6705004.0206061 3993466.52296896
3798185.05328059 3451821.5624289 7293996.86867461 2832905.26972678
4378698.1476993 4468002.85418353 3714623.89765883 2718466.90282779
7524449.64983626 2950437.1495281 4194596.28592921 2795827.77318101
4801788.72577028 3618230.29875972 5091685.77019752 4248122.70647652
4729964.20281946 4620982.9363411 7215846.76983927 3485106.19132682
5933820.14349913 6236580.98095548 4809923.61015999 5130920.22938947
4527282.87392583 8006869.07208366 3543927.63713716 5463086.05281816
3902441.28505163 4661063.63495886 4825370.89331733 4276803.47099967
7754072.71344458 4008573.23235629 6500977.91345038 5386796.00909808
2785343.0092764 6836356.36771845 2638553.49378047 3634231.99215348
8004072.41154001 8048399.75418243 3227573.69687308 5935167.41983541
3605772.48317454 3817767.91420862 7631673.34074558 4775106.62670537
5093491.44442931 6447437.14797692 4943449.6813564 5943929.14295426
3810460.26405425 6457426.32302761 3701491.23218178 5960492.47343073
4933137.14530295 4377133.13624287 7009335.02696999 6351460.97128317
6409340.41089671]

```

```

[38]: import sklearn
      from sklearn.linear_model import LinearRegression
      lm = LinearRegression()
      model=lm.fit(X_train, Y_train)

```

```

[39]: Ytrain_pred = lm.predict(X_train)
      Ytest_pred = lm.predict(X_test)

```

```

[40]: df=pd.DataFrame(Ytrain_pred,Y_train)
      df=pd.DataFrame(Ytest_pred,Y_test)

```

```

[42]: from sklearn.metrics import mean_squared_error, r2_score
      mse = mean_squared_error(Y_test, Ytest_pred)
      print(mse)
      mse = mean_squared_error(Ytrain_pred,Y_train)
      print (mse)

```

```

1771751116594.0344
969902818698.3114

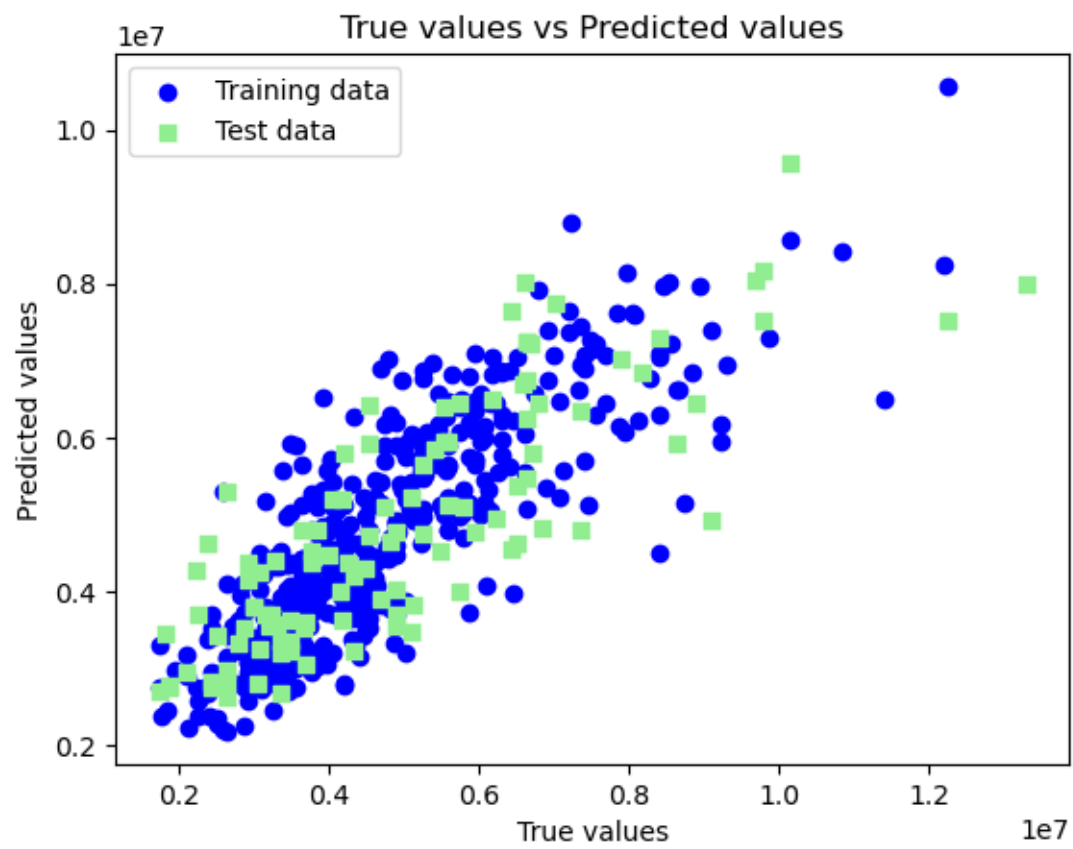
```

```

[43]: plt.scatter(Y_train, Ytrain_pred, c='blue', marker='o', label='Training data')
      plt.scatter(Y_test, Ytest_pred, c='lightgreen', marker='s', label='Test data')
      plt.xlabel('True values')
      plt.ylabel('Predicted values')
      plt.title("True values vs Predicted values")
      plt.legend(loc='upper left')

```

```
plt.show()
```



```
[ ]:
```