House Price Prediction

September 20, 2023

0.0.1 Mini Project - House Price Prediction

About A simple yet challenging project, to predict the housing price based on certain factors like house area, bedrooms, furnished, nearness to mainroad, etc. The dataset is small yet, it's complexity arises due to the fact that it has strong multicollinearity. Can you overcome these obstacles & build a decent predictive model?

Objective Understand the Dataset & cleanup (if required).

Build Regression models to predict the sales w.r.t a single & multiple feature.

Also evaluate the models & compare thier respective scores like R2, RMSE, etc.

0.0.2 Importing Necessary Packages

[1]:
```
# Data Manipulation
import pandas as pd
import numpy as np

# Data Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Train - Test Split
from sklearn.model_selection import train_test_split

# Encoding
from sklearn.preprocessing import LabelEncoder

# Feature Scalling
from sklearn.preprocessing import MinMaxScaler

import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Reading the Dataset

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

[3]:
```
df
```

[3]:
```
        price  area  bedrooms  bathrooms  stories mainroad guestroom basement  \
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
..        ...   ...       ...        ...      ...      ...       ...      ...
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]
```

[4]:
```
df.columns
```

[4]:
```
Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus'],
      dtype='object')
```

[5]:
```
df.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
```

Checking for Null Values

[6]:
```
df.isnull().sum()
```

[6]:
```
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
```

Checking for Duplicate values

[7]:
```
df.duplicated().any()
```

[7]:
```
False
```

Describing the data

[8]:
```
df.describe()
```

[8]:

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

```
          parking
count  545.000000
mean     0.693578
std      0.861586
min      0.000000
25%      0.000000
50%      0.000000
75%      1.000000
max      3.000000
```

Correlation Between the Variable

[9]:
```
df.corr()
```

[9]:
```
              price      area  bedrooms  bathrooms   stories   parking
price      1.000000  0.535997  0.366494   0.517545  0.420712  0.384394
area       0.535997  1.000000  0.151858   0.193820  0.083996  0.352980
bedrooms   0.366494  0.151858  1.000000   0.373930  0.408564  0.139270
bathrooms  0.517545  0.193820  0.373930   1.000000  0.326165  0.177496
stories    0.420712  0.083996  0.408564   0.326165  1.000000  0.045547
parking    0.384394  0.352980  0.139270   0.177496  0.045547  1.000000
```

0.0.3 Feature Engineering

Converting Categorical variable into numberical variable

[10]:
```
new = pd.get_dummies(df[['mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus']], drop_first=True)
```

[11]:
```
df1= pd.concat([df, new], axis=1)
```

[12]:
```
df1.drop(['mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus'], axis=1, inplace=True)
```

[13]:
```
df1.head()
```

[13]:
```
      price  area  bedrooms  bathrooms  stories  mainroad_yes  guestroom_yes  \
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

   basement_yes  hotwaterheating_yes  airconditioning_yes  prefarea_yes  \
0             0                    0                    1             1
1             0                    0                    1             0
```

```
2              1                    0                    0              1
3              1                    0                    1              1
4              1                    0                    1              0

   furnishingstatus_semi-furnished  furnishingstatus_unfurnished
0                                0                             0
1                                0                             0
2                                1                             0
3                                0                             0
4                                0                             0
```

Independent Feature and Dependent Feature

[14]:
```
X = df1.iloc[:, 1:] ###  Independent Feature
y = df1.iloc[:, :1] ### Target Feature
```

Train- Test split

[15]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=10)
```

[16]:
```
X_train.shape, X_test.shape
```

[16]:
```
((436, 12), (109, 12))
```

Feature Selection based on Correlation

[17]:
```
X_train.corr()
```

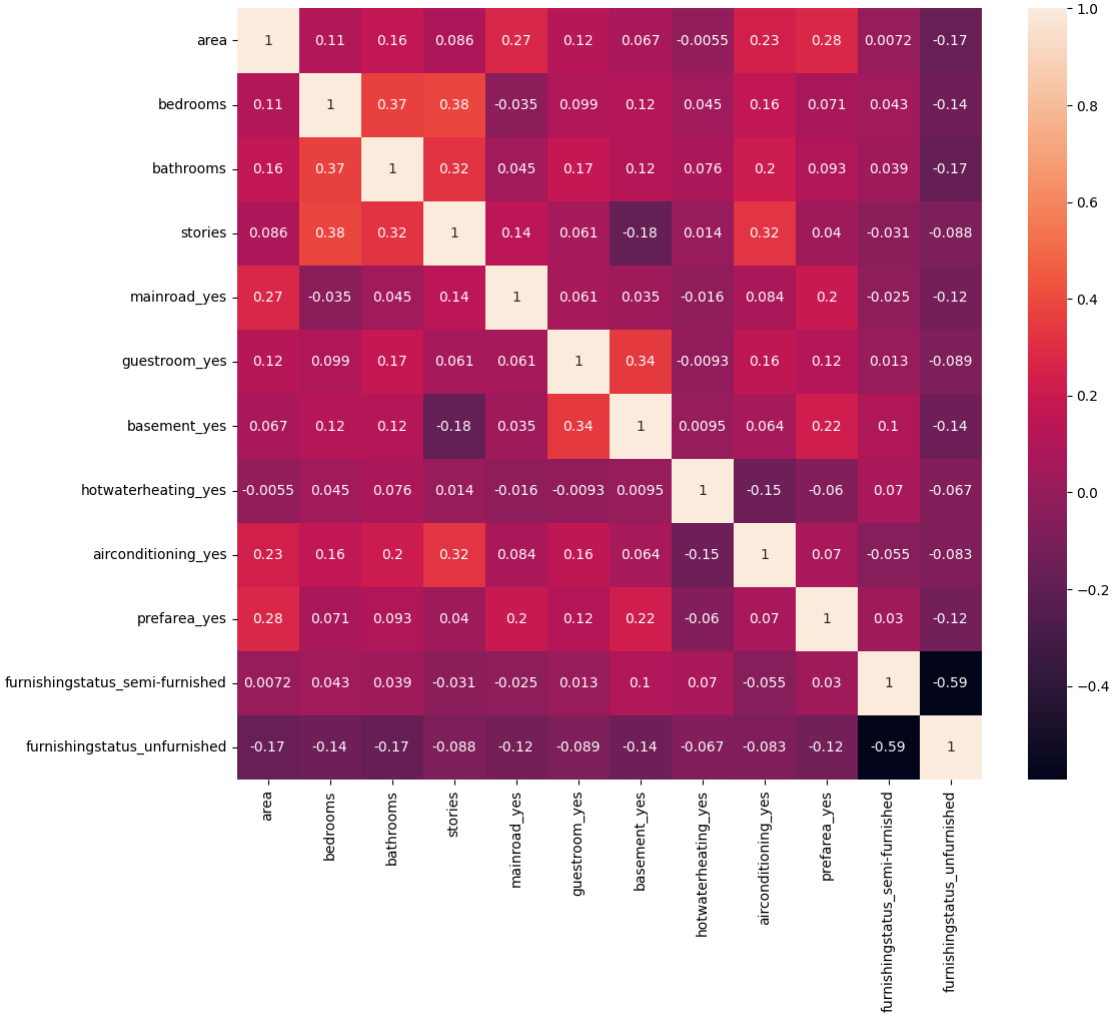
[17]:
```
                                    area  bedrooms  bathrooms   stories  \
area                            1.000000  0.109372   0.163872  0.085564
bedrooms                        0.109372  1.000000   0.368564  0.377708
bathrooms                       0.163872  0.368564   1.000000  0.315548
stories                         0.085564  0.377708   0.315548  1.000000
mainroad_yes                    0.268774 -0.035353   0.045180  0.142825
guestroom_yes                   0.122746  0.098983   0.172443  0.061337
basement_yes                    0.067032  0.119163   0.117333 -0.180341
hotwaterheating_yes            -0.005492  0.044958   0.075682  0.013982
airconditioning_yes             0.230790  0.160196   0.204903  0.322502
prefarea_yes                    0.275665  0.070757   0.092871  0.039715
furnishingstatus_semi-furnished 0.007233  0.043201   0.038859 -0.031475
furnishingstatus_unfurnished   -0.170114 -0.142063  -0.171999 -0.087608

                                mainroad_yes  guestroom_yes  basement_yes  \
area                                0.268774       0.122746      0.067032
bedrooms                           -0.035353       0.098983      0.119163
bathrooms                           0.045180       0.172443      0.117333
stories                             0.142825       0.061337     -0.180341
mainroad_yes                        1.000000       0.060698      0.035158
guestroom_yes                       0.060698       1.000000      0.344509
```

```
basement_yes                       0.035158       0.344509      1.000000
hotwaterheating_yes               -0.016026      -0.009304      0.009494
airconditioning_yes                0.083949       0.157837      0.064392
prefarea_yes                       0.196197       0.120601      0.223982
furnishingstatus_semi-furnished   -0.024506       0.013486      0.102504
furnishingstatus_unfurnished      -0.118004      -0.089196     -0.142858

                                  hotwaterheating_yes  airconditioning_yes  \
area                                        -0.005492             0.230790
bedrooms                                     0.044958             0.160196
bathrooms                                    0.075682             0.204903
stories                                      0.013982             0.322502
mainroad_yes                                -0.016026             0.083949
guestroom_yes                               -0.009304             0.157837
basement_yes                                 0.009494             0.064392
hotwaterheating_yes                          1.000000            -0.145693
airconditioning_yes                         -0.145693             1.000000
prefarea_yes                                -0.059629             0.069574
furnishingstatus_semi-furnished              0.070102            -0.055087
furnishingstatus_unfurnished                -0.067233            -0.083494

                                  prefarea_yes  \
area                                  0.275665
bedrooms                              0.070757
bathrooms                             0.092871
stories                               0.039715
mainroad_yes                          0.196197
guestroom_yes                         0.120601
basement_yes                          0.223982
hotwaterheating_yes                  -0.059629
airconditioning_yes                   0.069574
prefarea_yes                          1.000000
furnishingstatus_semi-furnished       0.030354
furnishingstatus_unfurnished         -0.123632

                                  furnishingstatus_semi-furnished  \
area                                                     0.007233
bedrooms                                                 0.043201
bathrooms                                                0.038859
stories                                                 -0.031475
mainroad_yes                                            -0.024506
guestroom_yes                                            0.013486
basement_yes                                             0.102504
hotwaterheating_yes                                      0.070102
airconditioning_yes                                     -0.055087
prefarea_yes                                             0.030354
furnishingstatus_semi-furnished                          1.000000
```

```
furnishingstatus_unfurnished                          -0.594155
```

	furnishingstatus_unfurnished
area	-0.170114
bedrooms	-0.142063
bathrooms	-0.171999
stories	-0.087608
mainroad_yes	-0.118004
guestroom_yes	-0.089196
basement_yes	-0.142858
hotwaterheating_yes	-0.067233
airconditioning_yes	-0.083494
prefarea_yes	-0.123632
furnishingstatus_semi-furnished	-0.594155
furnishingstatus_unfurnished	1.000000

[18]:
```
plt.figure(figsize=(12,10))

corr = X_train.corr()

sns.heatmap(corr, annot= True)

plt.show()
```



[19]:
```python
def correlation(dataset, threshold):
    col_corr = set()
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i,j]) > threshold:
                colname = corr_matrix.columns[i]
                col_corr.add(colname)
    return col_corr
```

[20]:
```python
corr_feature = correlation(X_train, 0.85)
```

[21]:
```python
corr_feature
```

[21]:
```
set()
```

Feature Scalling

[23]:
```
scaler = MinMaxScaler()
X_train_scalled = scaler.fit_transform(X_train)
X_test_scalled = scaler.transform(X_test)
```

[24]:
```
X_train_scalled
```

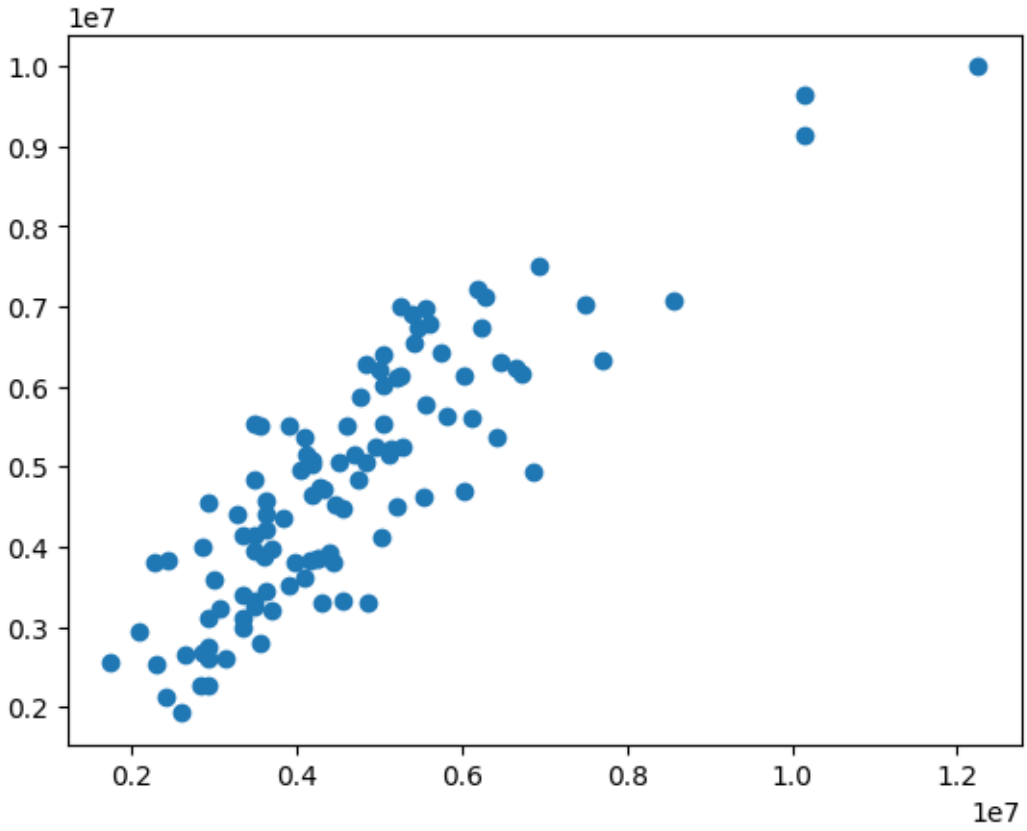
[24]:
```
array([[0.18848921, 0.4       , 0.        , …, 0.        , 0.        ,
        1.        ],
       [0.26618705, 0.4       , 0.        , …, 0.        , 1.        ,
        0.        ],
       [0.08035971, 0.6       , 0.5       , …, 0.        , 0.        ,
        0.        ],
       …,
       [0.00978417, 0.2       , 0.        , …, 0.        , 1.        ,
        0.        ],
       [1.        , 0.4       , 0.        , …, 0.        , 1.        ,
        0.        ],
       [0.08489209, 0.4       , 0.        , …, 1.        , 1.        ,
        0.        ]])
```

0.0.4 Linear Regression Model

[25]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score

regressor = LinearRegression()
regressor.fit(X_train_scalled, y_train)

y_pred = regressor.predict(X_test_scalled)

mae = mean_absolute_error(y_test, y_pred)

score = r2_score(y_test, y_pred)

print('Mean absolute error', mae)

print('Accuracy', score)

plt.scatter(y_test, y_pred)
```

```
Mean absolute error 735599.8421667002
Accuracy 0.71459095170096
```

[25]:
```
<matplotlib.collections.PathCollection at 0x27e23aef670>
```



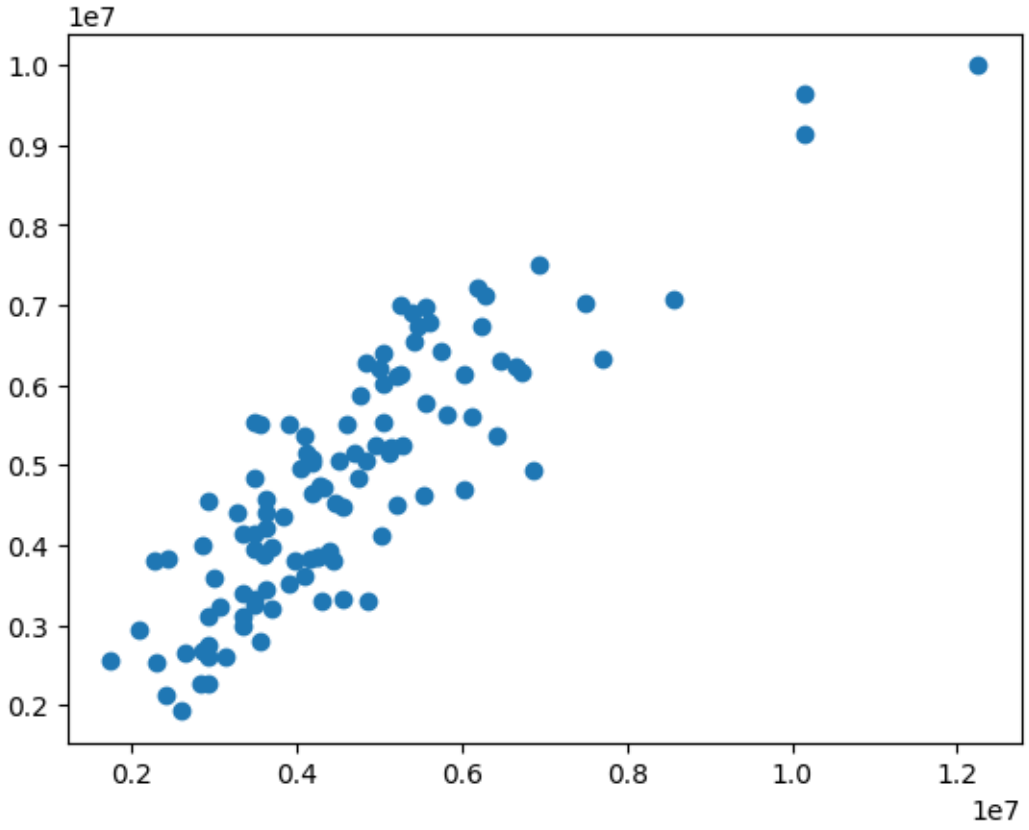
0.0.5 Lasso Regression

[26]:
```
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score

lasso = Lasso()

lasso.fit(X_train_scalled, y_train)

y_pred = lasso.predict(X_test_scalled)

mae = mean_absolute_error(y_test, y_pred)

score = r2_score(y_test, y_pred)

print('Mean absolute error', mae)

print('Accuracy', score)
```

```
plt.scatter(y_test, y_pred)
```

```
Mean absolute error 735597.8373825763
Accuracy 0.714592187822061
```

[26]: <matplotlib.collections.PathCollection at 0x27e23b4ae80>



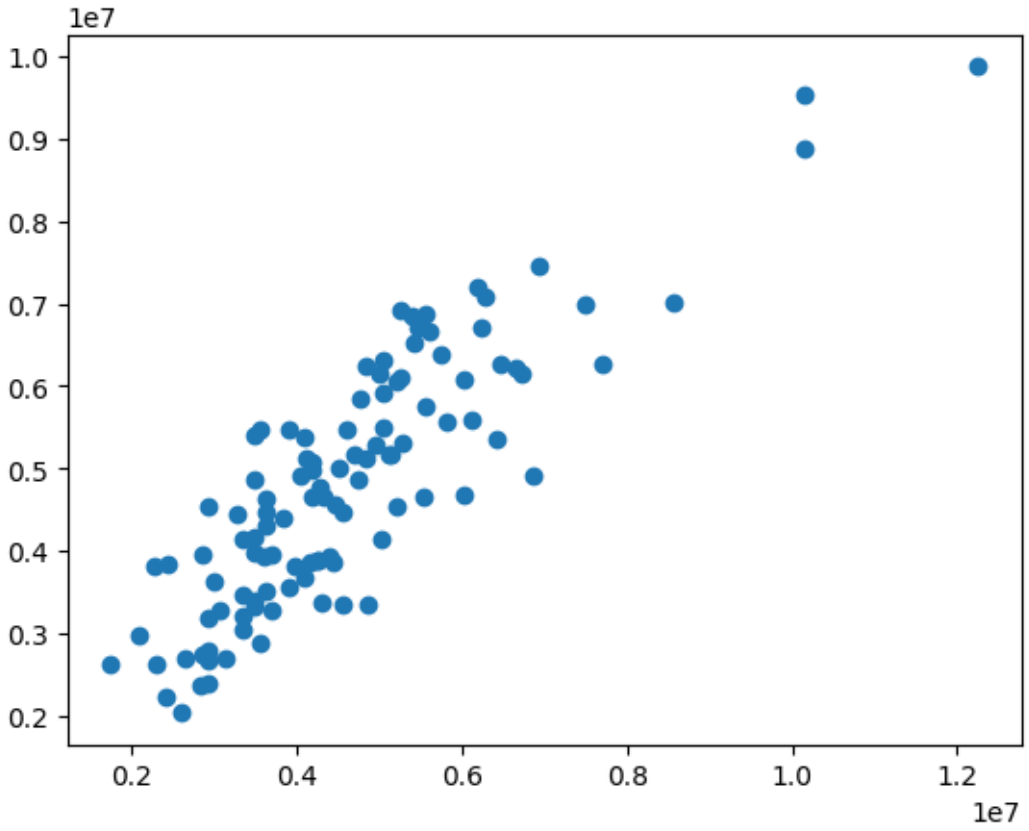
0.0.6 Cross Validation Lasso

[27]:
```
from sklearn.linear_model import LassoCV
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score

lassocv = LassoCV(cv=5)

lassocv.fit(X_train_scalled, y_train)

y_pred = lassocv.predict(X_test_scalled)

mae = mean_absolute_error(y_test, y_pred)
```

```
score = r2_score(y_test, y_pred)

print('Mean absolute error', mae)

print('Accuracy', score)

plt.scatter(y_test, y_pred)
```

```
Mean absolute error 722435.5202455592
Accuracy 0.7218516220558926
```

[27]: <matplotlib.collections.PathCollection at 0x27e23bd6e80>



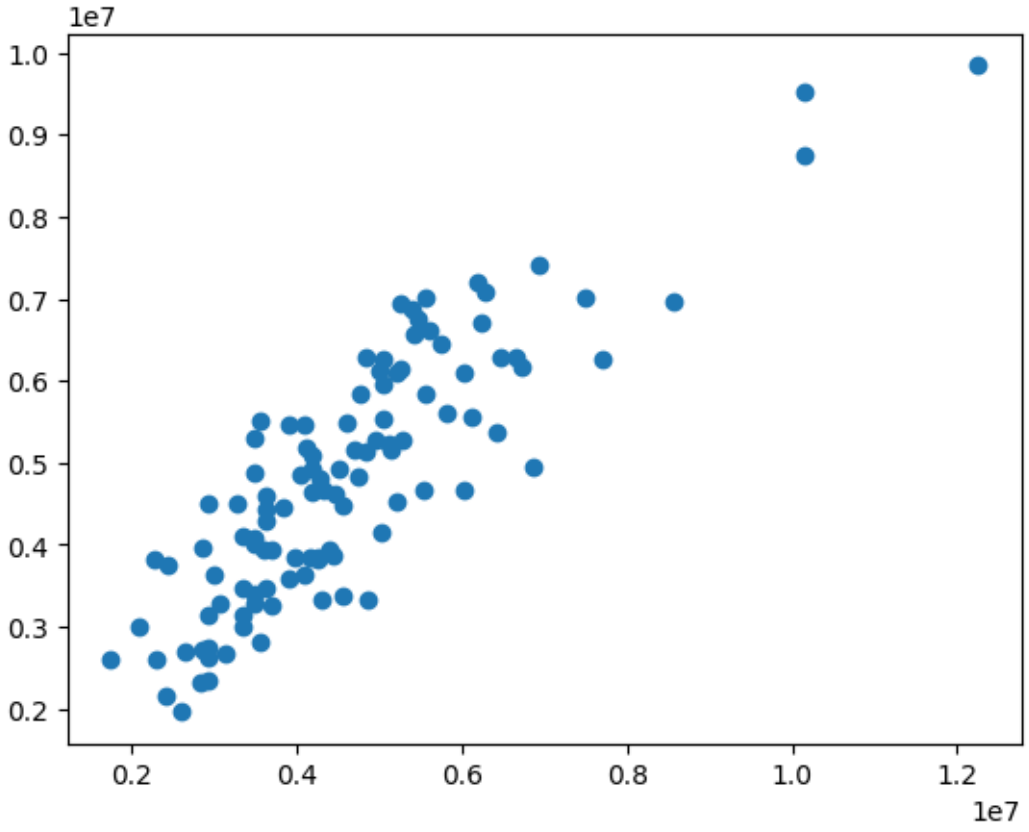
0.0.7 Ridge Regression

[28]:
```
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score

ridge = Ridge()
```

```
ridge.fit(X_train_scalled, y_train)

y_pred = ridge.predict(X_test_scalled)

mae = mean_absolute_error(y_test, y_pred)

score = r2_score(y_test, y_pred)

print('Mean absolute error', mae)

print('Accuracy', score)

plt.scatter(y_test, y_pred)
```

```
Mean absolute error 731976.7421036318
Accuracy 0.7167974495560834
```

[28]: <matplotlib.collections.PathCollection at 0x27e23c52820>



[ ]: