exercise-module-51

October 30, 2024

1 Module 51: Linear Regression and Time Series Analysis

Author: Juliho Castillo Colmenares

[26]:
```python
import pandas as pd
import numpy as np
from sklearn.metrics import r2_score, mean_squared_error
```

[27]:
```python
# Load data
data = pd.read_csv("kc_house_data.csv")
data.head()
```

[27]:
```
           id             date     price  bedrooms  bathrooms  sqft_living  \
0  7129300520  20141013T000000  221900.0         3       1.00         1180
1  6414100192  20141209T000000  538000.0         3       2.25         2570
2  5631500400  20150225T000000  180000.0         2       1.00          770
3  2487200875  20141209T000000  604000.0         4       3.00         1960
4  1954400510  20150218T000000  510000.0         3       2.00         1680

   sqft_lot  floors  waterfront  view  …  grade  sqft_above  sqft_basement  \
0      5650     1.0           0     0  …      7        1180              0
1      7242     2.0           0     0  …      7        2170            400
2     10000     1.0           0     0  …      6         770              0
3      5000     1.0           0     0  …      7        1050            910
4      8080     1.0           0     0  …      8        1680              0

   yr_built  yr_renovated  zipcode      lat     long  sqft_living15  \
0      1955             0    98178  47.5112 -122.257           1340
1      1951          1991    98125  47.7210 -122.319           1690
2      1933             0    98028  47.7379 -122.233           2720
3      1965             0    98136  47.5208 -122.393           1360
4      1987             0    98074  47.6168 -122.045           1800

   sqft_lot15
0        5650
1        7639
2        8062
3        5000
```

```
4        7503

[5 rows x 21 columns]
```

[28]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   id             21613 non-null  int64
 1   date           21613 non-null  object
 2   price          21613 non-null  float64
 3   bedrooms       21613 non-null  int64
 4   bathrooms      21613 non-null  float64
 5   sqft_living    21613 non-null  int64
 6   sqft_lot       21613 non-null  int64
 7   floors         21613 non-null  float64
 8   waterfront     21613 non-null  int64
 9   view           21613 non-null  int64
 10  condition      21613 non-null  int64
 11  grade          21613 non-null  int64
 12  sqft_above     21613 non-null  int64
 13  sqft_basement  21613 non-null  int64
 14  yr_built       21613 non-null  int64
 15  yr_renovated   21613 non-null  int64
 16  zipcode        21613 non-null  int64
 17  lat            21613 non-null  float64
 18  long           21613 non-null  float64
 19  sqft_living15  21613 non-null  int64
 20  sqft_lot15     21613 non-null  int64
dtypes: float64(5), int64(15), object(1)
memory usage: 3.5+ MB
```

[29]:
```python
data["date"] = pd.to_datetime(data["date"])
data.head()
```

[29]:
```
           id       date     price  bedrooms  bathrooms  sqft_living  \
0  7129300520 2014-10-13  221900.0         3       1.00         1180
1  6414100192 2014-12-09  538000.0         3       2.25         2570
2  5631500400 2015-02-25  180000.0         2       1.00          770
3  2487200875 2014-12-09  604000.0         4       3.00         1960
4  1954400510 2015-02-18  510000.0         3       2.00         1680

   sqft_lot  floors  waterfront  view  …  grade  sqft_above  sqft_basement  \
0      5650     1.0           0     0  …      7        1180              0
1      7242     2.0           0     0  …      7        2170            400
```

```
2      10000     1.0            0      0  …      6         770               0
3       5000     1.0            0      0  …      7        1050             910
4       8080     1.0            0      0  …      8        1680               0

   yr_built  yr_renovated  zipcode      lat     long  sqft_living15  \
0      1955             0    98178  47.5112 -122.257           1340
1      1951          1991    98125  47.7210 -122.319           1690
2      1933             0    98028  47.7379 -122.233           2720
3      1965             0    98136  47.5208 -122.393           1360
4      1987             0    98074  47.6168 -122.045           1800

   sqft_lot15
0        5650
1        7639
2        8062
3        5000
4        7503

[5 rows x 21 columns]
```

[30]:

```python
# Calculate the correlation matrix
correlation_matrix = data.corr()
correlation_matrix
```

[30]:

```
                     id      date     price  bedrooms  bathrooms  sqft_living  \
id             1.000000  0.005577 -0.016762  0.001286   0.005160    -0.012258
date           0.005577  1.000000 -0.004357 -0.016800  -0.034410    -0.034559
price         -0.016762 -0.004357  1.000000  0.308350   0.525138     0.702035
bedrooms       0.001286 -0.016800  0.308350  1.000000   0.515884     0.576671
bathrooms      0.005160 -0.034410  0.525138  0.515884   1.000000     0.754665
sqft_living   -0.012258 -0.034559  0.702035  0.576671   0.754665     1.000000
sqft_lot      -0.132109  0.006313  0.089661  0.031703   0.087740     0.172826
floors         0.018525 -0.022491  0.256794  0.175429   0.500653     0.353949
waterfront    -0.002721  0.001356  0.266369 -0.006582   0.063744     0.103818
view           0.011592 -0.001800  0.397293  0.079532   0.187737     0.284611
condition     -0.023783 -0.050769  0.036362  0.028472  -0.124982    -0.058753
grade          0.008130 -0.039912  0.667434  0.356967   0.664983     0.762704
sqft_above    -0.010842 -0.027924  0.605567  0.477600   0.685342     0.876597
sqft_basement -0.005151 -0.019469  0.323816  0.303093   0.283770     0.435043
yr_built       0.021380 -0.000355  0.054012  0.154178   0.506019     0.318049
yr_renovated  -0.016907 -0.024509  0.126434  0.018841   0.050739     0.055363
zipcode       -0.008224  0.001404 -0.053203 -0.152668  -0.203866    -0.199430
lat           -0.001891 -0.032856  0.307003 -0.008931   0.024573     0.052529
long           0.020799 -0.007020  0.021626  0.129473   0.223042     0.240223
sqft_living15 -0.002901 -0.031515  0.585379  0.391638   0.568634     0.756420
sqft_lot15    -0.138798  0.002566  0.082447  0.029244   0.087175     0.183286
```

	sqft_lot	floors	waterfront	view	…	grade \
id	-0.132109	0.018525	-0.002721	0.011592	…	0.008130
date	0.006313	-0.022491	0.001356	-0.001800	…	-0.039912
price	0.089661	0.256794	0.266369	0.397293	…	0.667434
bedrooms	0.031703	0.175429	-0.006582	0.079532	…	0.356967
bathrooms	0.087740	0.500653	0.063744	0.187737	…	0.664983
sqft_living	0.172826	0.353949	0.103818	0.284611	…	0.762704
sqft_lot	1.000000	-0.005201	0.021604	0.074710	…	0.113621
floors	-0.005201	1.000000	0.023698	0.029444	…	0.458183
waterfront	0.021604	0.023698	1.000000	0.401857	…	0.082775
view	0.074710	0.029444	0.401857	1.000000	…	0.251321
condition	-0.008958	-0.263768	0.016653	0.045990	…	-0.144674
grade	0.113621	0.458183	0.082775	0.251321	…	1.000000
sqft_above	0.183512	0.523885	0.072075	0.167649	…	0.755923
sqft_basement	0.015286	-0.245705	0.080588	0.276947	…	0.168392
yr_built	0.053080	0.489319	-0.026161	-0.053440	…	0.446963
yr_renovated	0.007644	0.006338	0.092885	0.103917	…	0.014414
zipcode	-0.129574	-0.059121	0.030285	0.084827	…	-0.184862
lat	-0.085683	0.049614	-0.014274	0.006157	…	0.114084
long	0.229521	0.125419	-0.041910	-0.078400	…	0.198372
sqft_living15	0.144608	0.279885	0.086463	0.280439	…	0.713202
sqft_lot15	0.718557	-0.011269	0.030703	0.072575	…	0.119248

	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode \
id	-0.010842	-0.005151	0.021380	-0.016907	-0.008224
date	-0.027924	-0.019469	-0.000355	-0.024509	0.001404
price	0.605567	0.323816	0.054012	0.126434	-0.053203
bedrooms	0.477600	0.303093	0.154178	0.018841	-0.152668
bathrooms	0.685342	0.283770	0.506019	0.050739	-0.203866
sqft_living	0.876597	0.435043	0.318049	0.055363	-0.199430
sqft_lot	0.183512	0.015286	0.053080	0.007644	-0.129574
floors	0.523885	-0.245705	0.489319	0.006338	-0.059121
waterfront	0.072075	0.080588	-0.026161	0.092885	0.030285
view	0.167649	0.276947	-0.053440	0.103917	0.084827
condition	-0.158214	0.174105	-0.361417	-0.060618	0.003026
grade	0.755923	0.168392	0.446963	0.014414	-0.184862
sqft_above	1.000000	-0.051943	0.423898	0.023285	-0.261190
sqft_basement	-0.051943	1.000000	-0.133124	0.071323	0.074845
yr_built	0.423898	-0.133124	1.000000	-0.224874	-0.346869
yr_renovated	0.023285	0.071323	-0.224874	1.000000	0.064357
zipcode	-0.261190	0.074845	-0.346869	0.064357	1.000000
lat	-0.000816	0.110538	-0.148122	0.029398	0.267048
long	0.343803	-0.144765	0.409356	-0.068372	-0.564072
sqft_living15	0.731870	0.200355	0.326229	-0.002673	-0.279033
sqft_lot15	0.194050	0.017276	0.070958	0.007854	-0.147221

	lat	long	sqft_living15	sqft_lot15

```
id            -0.001891  0.020799      -0.002901   -0.138798
date          -0.032856 -0.007020      -0.031515    0.002566
price          0.307003  0.021626       0.585379    0.082447
bedrooms      -0.008931  0.129473       0.391638    0.029244
bathrooms      0.024573  0.223042       0.568634    0.087175
sqft_living    0.052529  0.240223       0.756420    0.183286
sqft_lot      -0.085683  0.229521       0.144608    0.718557
floors         0.049614  0.125419       0.279885   -0.011269
waterfront    -0.014274 -0.041910       0.086463    0.030703
view           0.006157 -0.078400       0.280439    0.072575
condition     -0.014941 -0.106500      -0.092824   -0.003406
grade          0.114084  0.198372       0.713202    0.119248
sqft_above    -0.000816  0.343803       0.731870    0.194050
sqft_basement  0.110538 -0.144765       0.200355    0.017276
yr_built      -0.148122  0.409356       0.326229    0.070958
yr_renovated   0.029398 -0.068372      -0.002673    0.007854
zipcode        0.267048 -0.564072      -0.279033   -0.147221
lat            1.000000 -0.135512       0.048858   -0.086419
long          -0.135512  1.000000       0.334605    0.254451
sqft_living15  0.048858  0.334605       1.000000    0.183192
sqft_lot15    -0.086419  0.254451       0.183192    1.000000

[21 rows x 21 columns]
```

[31]:
```python
# Filter features with correlation > 0.1 with `price`
target_correlation = correlation_matrix["price"].abs()
selected_features = (
    target_correlation[target_correlation > 0.1].index.drop("price").tolist()
)

selected_features
```

[31]:
```
['bedrooms',
 'bathrooms',
 'sqft_living',
 'floors',
 'waterfront',
 'view',
 'grade',
 'sqft_above',
 'sqft_basement',
 'yr_renovated',
 'lat',
 'sqft_living15']
```

[32]:
```python
X = data[selected_features]
y = data["price"]
```

[33]:
```
# Add a column of ones to X for the intercept
X = np.c_[np.ones(X.shape[0]), X]  # X with intercept
y = y.values  # Convert y to a numpy array
```

[34]:
```
# Calculate coefficients using the Normal Equation
X_transpose = X.T
beta = np.linalg.inv(X_transpose @ X) @ X_transpose @ y
```

[35]:
```
# Predictions
y_pred = X @ beta
```

[36]:
```
# Model evaluation
r2 = r2_score(y, y_pred)
mse = mean_squared_error(y, y_pred)
mae = np.mean(np.abs(y - y_pred))

print("Coefficients:", beta)
print("R-squared:", r2)
print("Mean Squared Error:", mse)
print("Mean Absolute Error:", mae)
```

```
Coefficients: [-3.48577489e+07  1.85911382e+06 -1.61179143e+06 -1.10598216e+04
  9.53393703e+04 -3.23780072e+06  3.97419987e+05 -6.34352356e+04
  1.14403791e+04  1.13043386e+04  5.84567892e+01  6.67581646e+05
  7.68441511e+00]
R-squared: -18.72207459869437
Mean Squared Error: 2658065131096.9844
Mean Absolute Error: 1241765.8926068344
```