

model_selection_homework_answer

August 29, 2021

0.1 Bài toán

- Dự đoán giá nhà.
- Dữ liệu gồm thông tin căn nhà và giá

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

from sklearn.model_selection import learning_curve
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.pipeline import Pipeline

from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.ensemble import RandomForestRegressor
%matplotlib inline
```

```
[30]: import sklearn
sklearn.metrics.SCORERS.keys()
```

```
[30]: dict_keys(['explained_variance', 'r2', 'max_error', 'neg_median_absolute_error',
'neg_mean_absolute_error', 'neg_mean_squared_error',
'neg_mean_squared_log_error', 'neg_root_mean_squared_error',
'neg_mean_poisson_deviance', 'neg_mean_gamma_deviance', 'accuracy', 'roc_auc',
'roc_auc_ovr', 'roc_auc_ovo', 'roc_auc_ovr_weighted', 'roc_auc_ovo_weighted',
'balanced_accuracy', 'average_precision', 'neg_log_loss', 'neg_brier_score',
'adjusted_rand_score', 'homogeneity_score', 'completeness_score',
'v_measure_score', 'mutual_info_score', 'adjusted_mutual_info_score',
'normalized_mutual_info_score', 'fowlkes_mallows_score', 'precision',
'precision_macro', 'precision_micro', 'precision_samples', 'precision_weighted',
'recall', 'recall_macro', 'recall_micro', 'recall_samples', 'recall_weighted',
'f1', 'f1_macro', 'f1_micro', 'f1_samples', 'f1_weighted', 'jaccard',
'jaccard_macro', 'jaccard_micro', 'jaccard_samples', 'jaccard_weighted'])
```

```
[2]: dataset = pd.read_csv('kc_house_data.csv')
dataset
```

```
[2]:
```

| | id | date | price | bedrooms | bathrooms | \ |
|-------|------------|-----------------|----------|----------|-----------|---|
| 0 | 7129300520 | 20141013T000000 | 221900.0 | 3 | 1.00 | |
| 1 | 6414100192 | 20141209T000000 | 538000.0 | 3 | 2.25 | |
| 2 | 5631500400 | 20150225T000000 | 180000.0 | 2 | 1.00 | |
| 3 | 2487200875 | 20141209T000000 | 604000.0 | 4 | 3.00 | |
| 4 | 1954400510 | 20150218T000000 | 510000.0 | 3 | 2.00 | |
| ... | ... | ... | ... | ... | | |
| 21608 | 263000018 | 20140521T000000 | 360000.0 | 3 | 2.50 | |
| 21609 | 6600060120 | 20150223T000000 | 400000.0 | 4 | 2.50 | |
| 21610 | 1523300141 | 20140623T000000 | 402101.0 | 2 | 0.75 | |
| 21611 | 291310100 | 20150116T000000 | 400000.0 | 3 | 2.50 | |
| 21612 | 1523300157 | 20141015T000000 | 325000.0 | 2 | 0.75 | |

| | sqft_living | sqft_lot | floors | waterfront | view | ... | grade | \ |
|-------|-------------|----------|--------|------------|------|-----|-------|---|
| 0 | 1180 | 5650 | 1.0 | 0 | 0 | ... | 7 | |
| 1 | 2570 | 7242 | 2.0 | 0 | 0 | ... | 7 | |
| 2 | 770 | 10000 | 1.0 | 0 | 0 | ... | 6 | |
| 3 | 1960 | 5000 | 1.0 | 0 | 0 | ... | 7 | |
| 4 | 1680 | 8080 | 1.0 | 0 | 0 | ... | 8 | |
| ... | ... | ... | ... | ... | ... | ... | | |
| 21608 | 1530 | 1131 | 3.0 | 0 | 0 | ... | 8 | |
| 21609 | 2310 | 5813 | 2.0 | 0 | 0 | ... | 8 | |
| 21610 | 1020 | 1350 | 2.0 | 0 | 0 | ... | 7 | |
| 21611 | 1600 | 2388 | 2.0 | 0 | 0 | ... | 8 | |
| 21612 | 1020 | 1076 | 2.0 | 0 | 0 | ... | 7 | |

| | sqft_above | sqft_basement | yr_built | yr_renovated | zipcode | lat | \ |
|-------|------------|---------------|----------|--------------|---------|---------|---|
| 0 | 1180 | 0 | 1955 | 0 | 98178 | 47.5112 | |
| 1 | 2170 | 400 | 1951 | 1991 | 98125 | 47.7210 | |
| 2 | 770 | 0 | 1933 | 0 | 98028 | 47.7379 | |
| 3 | 1050 | 910 | 1965 | 0 | 98136 | 47.5208 | |
| 4 | 1680 | 0 | 1987 | 0 | 98074 | 47.6168 | |
| ... | ... | ... | ... | ... | ... | | |
| 21608 | 1530 | 0 | 2009 | 0 | 98103 | 47.6993 | |
| 21609 | 2310 | 0 | 2014 | 0 | 98146 | 47.5107 | |
| 21610 | 1020 | 0 | 2009 | 0 | 98144 | 47.5944 | |
| 21611 | 1600 | 0 | 2004 | 0 | 98027 | 47.5345 | |
| 21612 | 1020 | 0 | 2008 | 0 | 98144 | 47.5941 | |

| | long | sqft_living15 | sqft_lot15 |
|---|----------|---------------|------------|
| 0 | -122.257 | 1340 | 5650 |
| 1 | -122.319 | 1690 | 7639 |
| 2 | -122.233 | 2720 | 8062 |
| 3 | -122.393 | 1360 | 5000 |

```

4      -122.045      1800      7503
...      ...      ...      ...
21608 -122.346      1530      1509
21609 -122.362      1830      7200
21610 -122.299      1020      2007
21611 -122.069      1410      1287
21612 -122.299      1020      1357

```

[21613 rows x 21 columns]

```
[62]: dataset.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

```

```
[65]: dataset['date'][0]
```

```
[65]: '20141013T000000'
```

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

```
[8]:
```

| | id | price | bedrooms | bathrooms | sqft_living | \ |
|-------|--------------|--------------|--------------|--------------|--------------|---|
| count | 2.161300e+04 | 2.161300e+04 | 21613.000000 | 21613.000000 | 21613.000000 | |
| mean | 4.580302e+09 | 5.400881e+05 | 3.370842 | 2.114757 | 2079.899736 | |
| std | 2.876566e+09 | 3.671272e+05 | 0.930062 | 0.770163 | 918.440897 | |
| min | 1.000102e+06 | 7.500000e+04 | 0.000000 | 0.000000 | 290.000000 | |
| 25% | 2.123049e+09 | 3.219500e+05 | 3.000000 | 1.750000 | 1427.000000 | |
| 50% | 3.904930e+09 | 4.500000e+05 | 3.000000 | 2.250000 | 1910.000000 | |
| 75% | 7.308900e+09 | 6.450000e+05 | 4.000000 | 2.500000 | 2550.000000 | |
| max | 9.900000e+09 | 7.700000e+06 | 33.000000 | 8.000000 | 13540.000000 | |

| | sqft_lot | floors | waterfront | view | condition | \ |
|-------|--------------|--------------|--------------|--------------|--------------|---|
| count | 2.161300e+04 | 21613.000000 | 21613.000000 | 21613.000000 | 21613.000000 | |
| mean | 1.510697e+04 | 1.494309 | 0.007542 | 0.234303 | 3.409430 | |
| std | 4.142051e+04 | 0.539989 | 0.086517 | 0.766318 | 0.650743 | |
| min | 5.200000e+02 | 1.000000 | 0.000000 | 0.000000 | 1.000000 | |
| 25% | 5.040000e+03 | 1.000000 | 0.000000 | 0.000000 | 3.000000 | |
| 50% | 7.618000e+03 | 1.500000 | 0.000000 | 0.000000 | 3.000000 | |
| 75% | 1.068800e+04 | 2.000000 | 0.000000 | 0.000000 | 4.000000 | |
| max | 1.651359e+06 | 3.500000 | 1.000000 | 4.000000 | 5.000000 | |

| | grade | sqft_above | sqft_basement | yr_built | yr_renovated | \ |
|-------|--------------|--------------|---------------|--------------|--------------|---|
| count | 21613.000000 | 21613.000000 | 21613.000000 | 21613.000000 | 21613.000000 | |
| mean | 7.656873 | 1788.390691 | 291.509045 | 1971.005136 | 84.402258 | |
| std | 1.175459 | 828.090978 | 442.575043 | 29.373411 | 401.679240 | |
| min | 1.000000 | 290.000000 | 0.000000 | 1900.000000 | 0.000000 | |
| 25% | 7.000000 | 1190.000000 | 0.000000 | 1951.000000 | 0.000000 | |
| 50% | 7.000000 | 1560.000000 | 0.000000 | 1975.000000 | 0.000000 | |
| 75% | 8.000000 | 2210.000000 | 560.000000 | 1997.000000 | 0.000000 | |
| max | 13.000000 | 9410.000000 | 4820.000000 | 2015.000000 | 2015.000000 | |

| | zipcode | lat | long | sqft_living15 | sqft_lot15 |
|-------|--------------|--------------|--------------|---------------|---------------|
| count | 21613.000000 | 21613.000000 | 21613.000000 | 21613.000000 | 21613.000000 |
| mean | 98077.939805 | 47.560053 | -122.213896 | 1986.552492 | 12768.455652 |
| std | 53.505026 | 0.138564 | 0.140828 | 685.391304 | 27304.179631 |
| min | 98001.000000 | 47.155900 | -122.519000 | 399.000000 | 651.000000 |
| 25% | 98033.000000 | 47.471000 | -122.328000 | 1490.000000 | 5100.000000 |
| 50% | 98065.000000 | 47.571800 | -122.230000 | 1840.000000 | 7620.000000 |
| 75% | 98118.000000 | 47.678000 | -122.125000 | 2360.000000 | 10083.000000 |
| max | 98199.000000 | 47.777600 | -121.315000 | 6210.000000 | 871200.000000 |

```
[3]: X_data = dataset.drop(['price', 'id', 'date'], axis=1)
      Y_data = dataset['price']

      X_train, X_test, Y_train, Y_test = train_test_split(X_data, Y_data, test_size=0.
      ↪2, random_state=5)

      print("Dữ liệu training = ", X_train.shape, Y_train.shape)
      print("Dữ liệu testing = ", X_test.shape, Y_test.shape)
```

```
Dữ liệu training = (17290, 18) (17290,)
Dữ liệu testing = (4323, 18) (4323,)
```

```
[5]: def cross_validation(estimator):
    _, train_scores, test_scores = learning_curve(estimator,
                                                    X_train, Y_train,
                                                    cv=10,
                                                    n_jobs=-1,
                                                    train_sizes=[1.0, ],
                                                    ↵
    ↪scoring='neg_mean_absolute_error')
    test_scores = test_scores[0]
    mean, std = test_scores.mean(), test_scores.std()
    return mean, std

def plot(title, xlabel, X, Y, error, ylabel = "mean_squared_error"):
    plt.xlabel(xlabel)
    plt.title(title)
    plt.grid()
    plt.ylabel(ylabel)

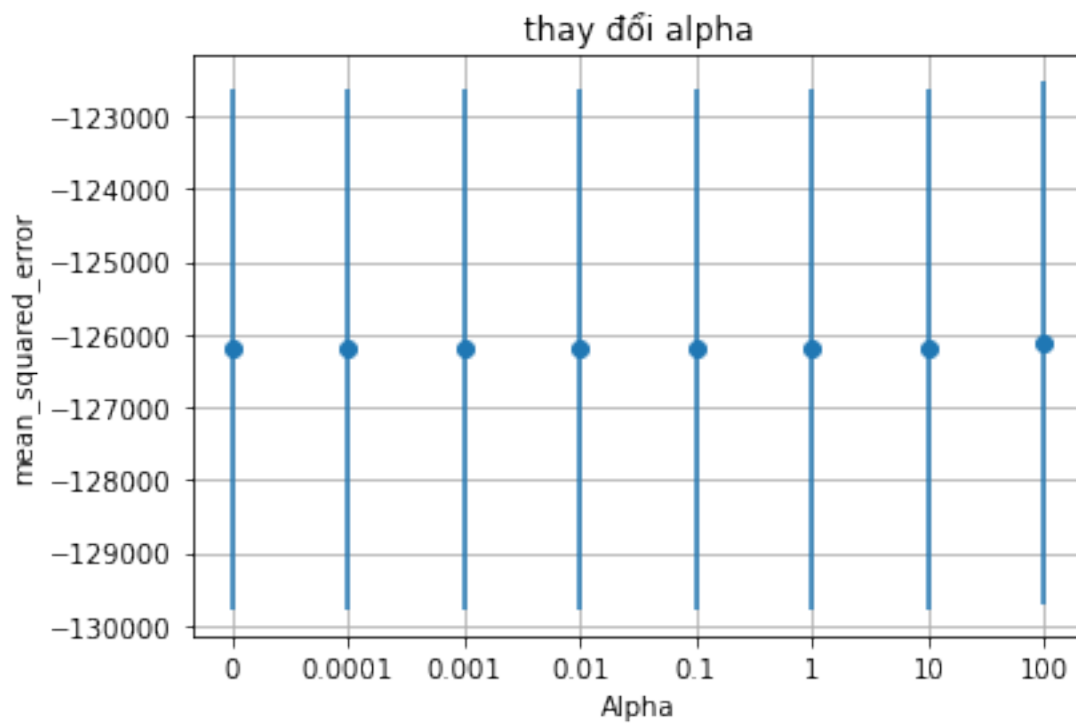
    plt.errorbar(X, Y, error, linestyle='None', marker='o')
```

```
[9]: title = "thay đổi alpha"
    xlabel = "Alpha"
    X = []
    Y = []
    error = []

    for aplha in tqdm([0, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100]):
        text_clf = Lasso(alpha=aplha)
        mean, std = cross_validation(text_clf)
        X.append(str(aplha))
        Y.append(mean)
        error.append(std)

    # lưu kết quả ra file ảnh
    plot(title, xlabel, X, Y, error)
    plt.show()
```

```
100%|
    | 8/8 [00:16<00:00, 2.08s/it]
```

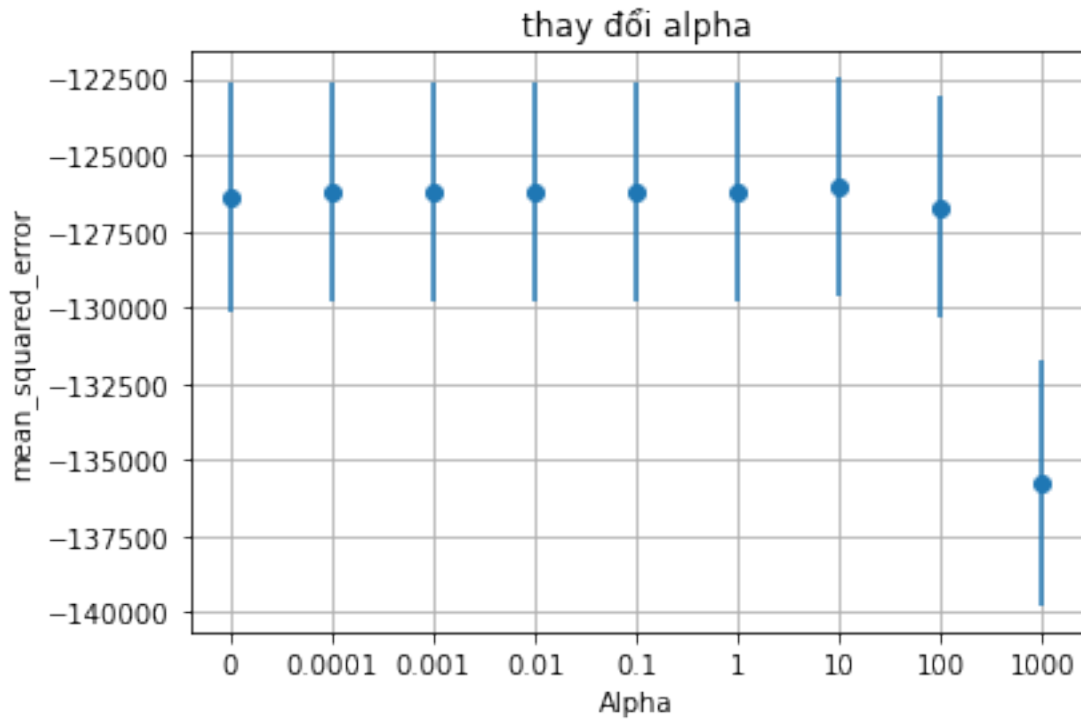


```
[12]: title = "thay đổi alpha"
      xlabel = "Alpha"
      X = []
      Y = []
      error = []

      for aplha in tqdm([0, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]):
          text_clf = Ridge(alpha=aplha)
          mean, std = cross_validation(text_clf)
          X.append(str(aplha))
          Y.append(mean)
          error.append(std)

      # lưu kết quả ra file ảnh
      plot(title, xlabel, X, Y, error)
      plt.show()
```

```
100%|
    | 9/9 [00:01<00:00, 5.09it/s]
```



```
[ ]: title = "thay đổi alpha, criterion = mae"
xlabel = "n_estimators"
X = []
Y = []
error = []

for n_estimators in tqdm([10, 50, 100, 300]):
    # Với từng giá trị n_estimators nhận được,
    # thực hiện xây dựng mô hình, huấn luyện và đánh giá theo cross-validation
    text_clf = RandomForestRegressor(criterion='mae', n_estimators=n_estimators)
    mean, std = cross_validation(text_clf)
    X.append(str(n_estimators))
    Y.append(mean)
    error.append(std)

# lưu kết quả ra file ảnh
plot(title, xlabel, X, Y, error)
# plt.savefig('images/RF_change_N.png', bbox_inches='tight')
plt.show()
```

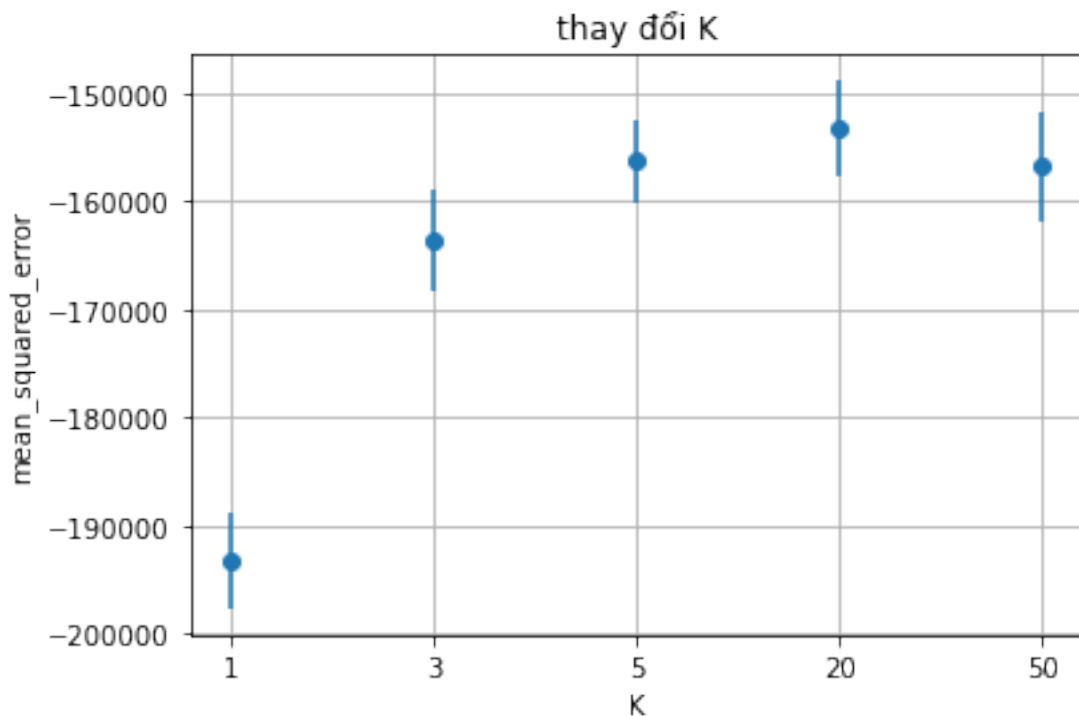
```
0%|
| 0/4 [00:00<?, ?it/s]
```

```
[18]: title = "thay đổi K"
      xlabel = "K"
      X = []
      Y = []
      error = []

      for k in tqdm([1, 3, 5, 20, 50]):
          # Với từng giá trị k nhận được,
          # thực hiện xây dựng mô hình, huấn luyện và đánh giá theo cross-validation
          text_clf = KNeighborsRegressor(n_neighbors=k)
          mean, std = cross_validation(text_clf)
          X.append(str(k))
          Y.append(mean)
          error.append(std)

      # lưu kết quả ra file ảnh
      plot(title, xlabel, X, Y, error)
      plt.savefig('images/KNN_change_K.png', bbox_inches='tight')
      plt.show()
```

100%|
| 5/5 [00:07<00:00, 1.46s/it]




```
[14]: # Kết quả dự đoán trên tập test
print(f'RF: {mean_absolute_error(Y_test, rf.predict(X_test))}')
print(f'KNN: {mean_absolute_error(Y_test, knn.predict(X_test))}')
print(f'Linear Regression: {mean_absolute_error(Y_test, lrg.predict(X_test))}')
print(f'Ridge: {mean_absolute_error(Y_test, ridge.predict(X_test))}')
print(f'Lasoo: {mean_absolute_error(Y_test, lasso.predict(X_test))}')
```

```
RF: 69948.35851615391
KNN: 158439.18634050427
Linear Regression: 125163.17629948513
Ridge: 125027.29664089366
Lasoo: 125090.93815064323
```

```
[13]: rf = RandomForestRegressor(criterion='mse', n_estimators=300)
knn = KNeighborsRegressor(n_neighbors=20)
lrg = LinearRegression()
ridge = Ridge(alpha=10)
lasso = Lasso(alpha=100)
# Huấn luyện các mô hình trên tập dữ liệu train đầy đủ
rf.fit(X_train, Y_train)
knn.fit(X_train, Y_train)
lrg.fit(X_train, Y_train)
ridge.fit(X_train, Y_train)
lasso.fit(X_train, Y_train)
```

```
C:\Users\Admin\anaconda3\lib\site-
packages\sklearn\linear_model\_coordinate_descent.py:531: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 319194110679868.75, tolerance: 232310058468.8226
positive)
```

```
[13]: Lasso(alpha=100)
```