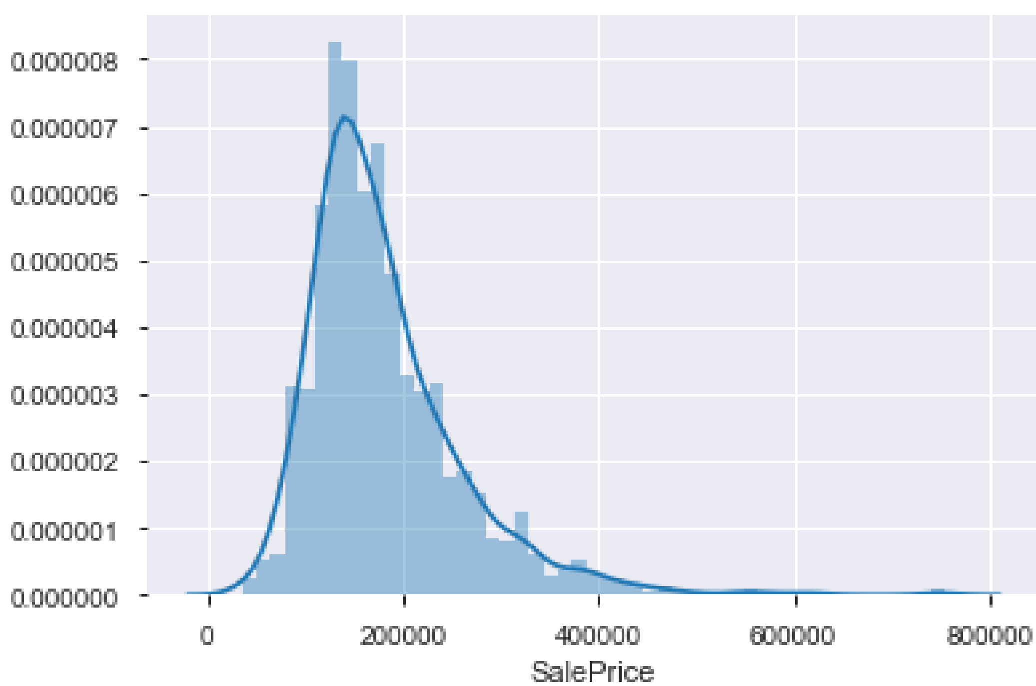
1 進階房價預測

[程式]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.stats import skew
from scipy.special import boxcox1p
from sklearn.linear_model import Lasso
from sklearn.model_selection import cross_val_score


train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

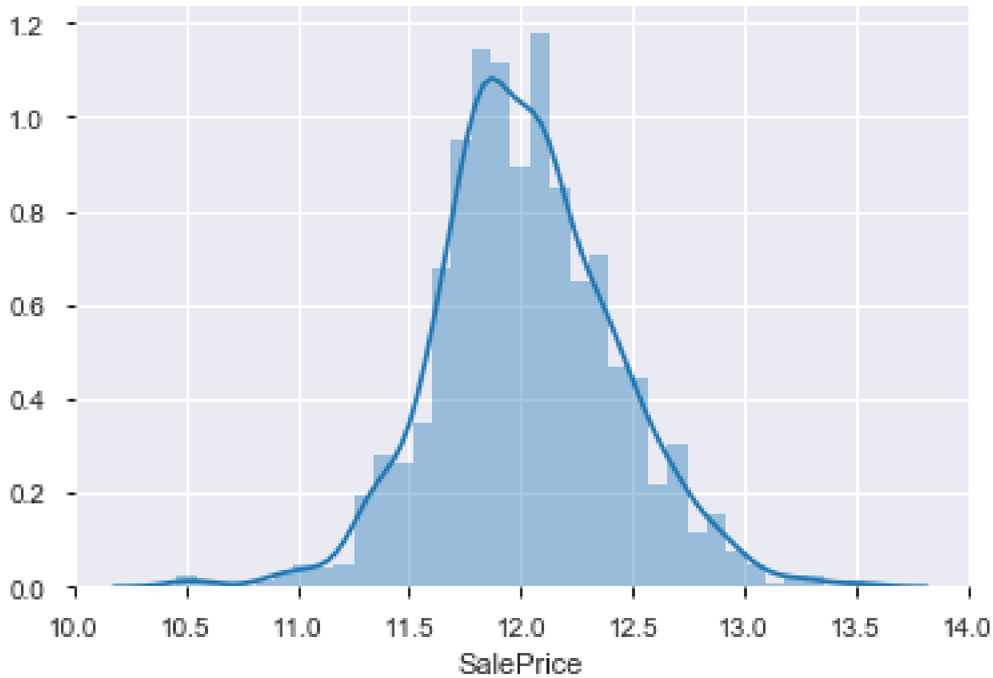
[程式]:
```
# 原本的答案分佈
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.distplot(train['SalePrice'])
```

[輸出]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x119407588>
```



[程式]:
```
# 取 log 後的結果:
sns.distplot(np.log1p(train["SalePrice"]))
```

[輸出]: <matplotlib.axes._subplots.AxesSubplot at 0x11a1b3908>



[程式]:
```python
all_data = pd.concat((train.loc[:,'MSSubClass':'SaleCondition'],
                      test.loc[:,'MSSubClass':'SaleCondition']))
train["SalePrice"] = np.log1p(train["SalePrice"])
```

[程式]:
```python
# 得到所有數字形態的特徵
numeric_feats = all_data.dtypes[all_data.dtypes != "object"].drop(["MSSubClass"]).index
```

[程式]:
```python
# 觀察所有特徵左偏/右偏
pd.DataFrame(train[numeric_feats].apply(lambda x: skew(x.dropna())).sort_values(ascending=Fals
```

[輸出]:

	0
MiscVal	24.451640
PoolArea	14.813135
LotArea	12.195142
3SsnPorch	10.293752
LowQualFinSF	9.002080
KitchenAbvGr	4.483784
BsmtFinSF2	4.250888
ScreenPorch	4.117977
BsmtHalfBath	4.099186
EnclosedPorch	3.086696

```
MasVnrArea       2.666326
OpenPorchSF      2.361912
LotFrontage      2.160866
BsmtFinSF1       1.683771
WoodDeckSF       1.539792
TotalBsmtSF      1.522688
1stFlrSF         1.375342
GrLivArea        1.365156
BsmtUnfSF        0.919323
2ndFlrSF         0.812194
OverallCond      0.692355
TotRmsAbvGrd     0.675646
HalfBath         0.675203
Fireplaces       0.648898
BsmtFullBath     0.595454
OverallQual      0.216721
MoSold           0.211835
BedroomAbvGr     0.211572
GarageArea       0.179796
YrSold           0.096170
FullBath         0.036524
GarageCars      -0.342197
YearRemodAdd    -0.503044
YearBuilt       -0.612831
GarageYrBlt     -0.648708
```

[程式]:
```python
# 利用 boxcox1p 拯救 skewness
skewed_feats = train[numeric_feats].apply(lambda x: skew(x.dropna()))
skewed_feats = skewed_feats[skewed_feats > 0.65]
skewed_feats = skewed_feats.index
all_data[skewed_feats] = boxcox1p(all_data[skewed_feats], 0.15)
pd.DataFrame(all_data[skewed_feats].apply(lambda x: skew(x.dropna())).sort_values(ascending=Fa
```

[輸出]:
```
                        0
PoolArea        15.119426
3SsnPorch        8.924822
LowQualFinSF     8.744143
MiscVal          5.597060
BsmtHalfBath     3.786685
KitchenAbvGr     3.698825
ScreenPorch      2.978396
BsmtFinSF2       2.563858
EnclosedPorch    2.025461
MasVnrArea       0.623664
HalfBath         0.590565
2ndFlrSF         0.327362
```

```
1stFlrSF        0.223905
WoodDeckSF      0.222631
LotArea         0.210453
GrLivArea       0.175316
TotRmsAbvGrd    0.142600
OpenPorchSF     0.100164
OverallCond    -0.470559
BsmtFinSF1     -0.487832
LotFrontage    -0.587532
BsmtUnfSF      -1.539637
TotalBsmtSF    -3.972167
```

[程式]:
```
# One-Hot Encoding
all_data = pd.get_dummies(all_data)
all_data = pd.get_dummies(all_data,  columns=["MSSubClass"])
all_data
```

[輸出]:
```
      LotFrontage    LotArea  OverallQual  OverallCond  YearBuilt  \
0        5.831328  19.212182            7     2.055642       2003
1        6.221214  19.712205            6     2.602594       1976
2        5.914940  20.347241            7     2.055642       2001
3        5.684507  19.691553            7     2.055642       1915
4        6.314735  21.325160            8     2.055642       2000
5        6.337529  21.282283            5     2.055642       1993
6        6.098626  19.907529            8     2.055642       2004
7             NaN  20.023862            7     2.259674       1973
8        5.392276  17.989871            7     2.055642       1931
9        5.357203  18.712544            5     2.259674       1939
10       5.968981  20.329199            5     2.055642       1965
11       6.337529  20.584023            9     2.055642       2005
12            NaN  20.929243            5     2.259674       1962
13       6.469750  20.126838            7     2.055642       2006
14            NaN  20.226881            6     2.055642       1960
15       5.392276  17.989871            7     2.602594       1929
16            NaN  20.343998            6     2.440268       1970
17       6.021742  20.178990            4     2.055642       1967
18       5.859551  21.155939            5     2.055642       2004
19       5.968981  18.783793            5     2.259674       1958
20       6.674652  21.311893            8     2.055642       2005
21       5.591427  18.727396            7     2.440268       1930
22       6.098626  19.770362            8     2.055642       2002
23       5.133567  16.656244            5     2.440268       1976
24            NaN  19.117502            5     2.602594       1968
25       6.844946  21.316319            8     2.055642       2007
26       5.684507  18.598238            5     2.440268       1951
27       6.615044  20.428657            8     2.055642       2007
```

```
28       5.248357  21.897710          5     2.259674       1957
29       5.684507  18.111423          4     2.259674       1927
…           …         …        …           …       …
1429     5.357203  18.507860          4     2.259674       1925
1430     6.098626  19.484144          6     2.055642       1957
1431     5.942124  20.733197          3     2.055642       1945
1432     5.357203  19.458096          5     2.259674       1951
1433     5.684507  19.244223          3     2.055642       1916
1434     5.012077  17.759065          8     2.055642       2005
1435     5.133567  16.327057          8     2.055642       2004
1436     5.942124  23.518465          6     2.259674       1979
1437     5.831328  19.182228          6     2.055642       1978
1438     5.968981  19.508324          8     2.055642       2001
1439     7.338607  20.285618          6     2.259674       1975
1440          NaN  27.131123          6     2.055642       1958
1441          NaN  19.047559          6     2.055642       2000
1442     6.553880  21.132420          8     2.055642       2005
1443     6.404587  20.463582          9     2.055642       2005
1444     7.104297  24.820844          1     1.540963       1951
1445     6.172972  18.502486          7     2.055642       1997
1446     5.012077  15.099791          5     2.259674       1977
1447     5.622899  19.942183          5     2.440268       1968
1448          NaN  20.553764          5     2.055642       1970
1449     3.932510  13.242388          4     2.259674       1970
1450     3.932510  13.270696          4     1.820334       1972
1451     6.221214  21.060245          5     2.055642       1969
1452     3.932510  13.368020          4     2.055642       1970
1453     3.932510  13.354279          4     2.055642       1970
1454     3.932510  14.081426          4     2.440268       1970
1455     3.932510  14.013314          4     2.055642       1970
1456     7.620056  22.782058          5     2.440268       1960
1457     5.744420  20.046557          5     2.055642       1992
1458     6.073289  19.723319          7     2.055642       1993

      YearRemodAdd  MasVnrArea  BsmtFinSF1  BsmtFinSF2  BsmtUnfSF  \
0             2003    8.059126   11.170327    0.000000   7.483296
1             1976    0.000000   12.062832    0.000000   8.897844
2             2002    7.646538   10.200343    0.000000   9.917060
3             1970    0.000000    8.274266    0.000000  10.468500
4             2000    9.391827   10.971129    0.000000  10.221051
5             1995    0.000000   11.267217    0.000000   5.802739
6             2005    7.944503   13.031093    0.000000   9.155750
7             1973    8.511220   11.702249    4.597157   8.274266
8             1950    0.000000    0.000000    0.000000  11.987364
9             1950    0.000000   11.676516    0.000000   7.338607
```

10	1965	0.000000	11.849447	0.000000	7.247551
11	2006	8.914179	12.119733	0.000000	7.836797
12	1962	0.000000	11.285513	0.000000	7.812236
13	2007	9.072419	0.000000	0.000000	13.290777
14	1960	8.232627	11.270884	0.000000	10.371953
15	2001	0.000000	0.000000	0.000000	11.614567
16	1970	7.873203	10.643870	0.000000	9.870950
17	1967	0.000000	0.000000	0.000000	0.000000
18	2004	0.000000	10.934619	0.000000	10.105326
19	1965	0.000000	10.292420	0.000000	10.396381
20	2006	9.590598	0.000000	0.000000	12.543063
21	1950	0.000000	0.000000	0.000000	10.897674
22	2002	8.873158	0.000000	0.000000	13.816566
23	1976	0.000000	11.640795	0.000000	8.103594
24	2001	0.000000	7.967837	11.023123	8.147316
25	2007	10.910038	0.000000	0.000000	13.432085
26	2000	0.000000	8.453930	10.200343	7.873203
27	2008	8.103594	12.689052	0.000000	10.200343
28	1997	0.000000	12.826768	0.000000	8.179634
29	1950	0.000000	0.000000	0.000000	10.371953
…	…	…	…	…	…
1429	1950	0.000000	0.000000	0.000000	10.914148
1430	1957	9.229405	0.000000	0.000000	12.031113
1431	1950	0.000000	0.000000	0.000000	0.000000
1432	1951	0.000000	0.000000	0.000000	10.991230
1433	1950	0.000000	0.000000	0.000000	8.274266
1434	2006	10.132026	13.445527	0.000000	0.000000
1435	2005	7.944503	13.428235	0.000000	4.492018
1436	1979	0.000000	11.424722	0.000000	11.670050
1437	1978	0.000000	0.000000	0.000000	13.615805
1438	2001	0.000000	0.000000	0.000000	13.284765
1439	1975	8.667201	10.634887	0.000000	10.524981
1440	1958	0.000000	11.858621	0.000000	11.234013
1441	2000	0.000000	12.487921	6.952064	7.169005
1442	2006	8.081455	12.989872	0.000000	9.577769
1443	2006	9.603371	13.211803	0.000000	9.609736
1444	1951	0.000000	0.000000	0.000000	0.000000
1445	1997	8.103594	12.748083	0.000000	5.172535
1446	1977	0.000000	0.000000	0.000000	8.728898
1447	2003	0.000000	9.956819	0.000000	9.853469
1448	1970	0.000000	7.469200	0.000000	13.308753
1449	1970	0.000000	10.381748	0.000000	6.808145
1450	1972	0.000000	8.622254	0.000000	8.978567
1451	1979	8.036603	7.003881	9.350349	10.914148
1452	1970	0.000000	9.764456	0.000000	7.308628

```
1453          1970   0.000000    0.000000    0.000000  10.496872
1454          1970   0.000000    0.000000    0.000000  10.496872
1455          1970   0.000000    8.622254    0.000000   8.978567
1456          1996   0.000000   12.703313    0.000000   0.000000
1457          1992   0.000000    9.301176    0.000000  10.630386
1458          1994   6.533131   11.361228    0.000000   8.492259

          …        MSSubClass_70  MSSubClass_75  MSSubClass_80  \
0         …                    0              0              0
1         …                    0              0              0
2         …                    0              0              0
3         …                    1              0              0
4         …                    0              0              0
5         …                    0              0              0
6         …                    0              0              0
7         …                    0              0              0
8         …                    0              0              0
9         …                    0              0              0
10        …                    0              0              0
11        …                    0              0              0
12        …                    0              0              0
13        …                    0              0              0
14        …                    0              0              0
15        …                    0              0              0
16        …                    0              0              0
17        …                    0              0              0
18        …                    0              0              0
19        …                    0              0              0
20        …                    0              0              0
21        …                    0              0              0
22        …                    0              0              0
23        …                    0              0              0
24        …                    0              0              0
25        …                    0              0              0
26        …                    0              0              0
27        …                    0              0              0
28        …                    0              0              0
29        …                    0              0              0
…       …                …            …            …
1429      …                    0              0              0
1430      …                    0              0              0
1431      …                    0              0              0
1432      …                    0              0              0
1433      …                    0              0              0
1434      …                    0              0              0
```

```
1435  ...  0  0  0
1436  ...  0  0  0
1437  ...  0  0  0
1438  ...  0  0  0
1439  ...  0  0  1
1440  ...  0  0  0
1441  ...  0  0  0
1442  ...  0  0  0
1443  ...  0  0  0
1444  ...  0  0  0
1445  ...  0  0  0
1446  ...  0  0  0
1447  ...  0  0  0
1448  ...  0  0  0
1449  ...  0  0  0
1450  ...  0  0  0
1451  ...  0  0  0
1452  ...  0  0  0
1453  ...  0  0  0
1454  ...  0  0  0
1455  ...  0  0  0
1456  ...  0  0  0
1457  ...  0  0  0
1458  ...  0  0  0

      MSSubClass_85  MSSubClass_90  MSSubClass_120  MSSubClass_150  \
0                 0              0               0               0
1                 0              0               0               0
2                 0              0               0               0
3                 0              0               0               0
4                 0              0               0               0
5                 0              0               0               0
6                 0              0               0               0
7                 0              0               0               0
8                 0              0               0               0
9                 0              0               0               0
10                0              0               0               0
11                0              0               0               0
12                0              0               0               0
13                0              0               0               0
14                0              0               0               0
15                0              0               0               0
16                0              0               0               0
17                0              1               0               0
18                0              0               0               0
```

19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	1	0
24	0	0	0	0
25	0	0	0	0
26	0	0	0	0
27	0	0	0	0
28	0	0	0	0
29	0	0	0	0
…	…	…	…	…
1429	0	0	0	0
1430	0	0	0	0
1431	0	0	0	0
1432	0	0	0	0
1433	0	0	0	0
1434	0	0	1	0
1435	0	0	1	0
1436	0	0	0	0
1437	0	1	0	0
1438	0	0	0	0
1439	0	0	0	0
1440	0	0	0	0
1441	0	0	0	0
1442	0	0	0	0
1443	0	0	0	0
1444	0	0	0	0
1445	0	1	0	0
1446	0	0	0	0
1447	0	0	0	0
1448	0	1	0	0
1449	0	0	0	0
1450	0	0	0	0
1451	0	0	0	0
1452	0	0	0	0
1453	0	0	0	0
1454	0	0	0	0
1455	0	0	0	0
1456	0	0	0	0
1457	1	0	0	0
1458	0	0	0	0

	MSSubClass_160	MSSubClass_180	MSSubClass_190
0	0	0	0

1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	1
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
17	0	0	0
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0
24	0	0	0
25	0	0	0
26	0	0	0
27	0	0	0
28	0	0	0
29	0	0	0
…	…	…	…
1429	0	0	0
1430	0	0	0
1431	0	0	0
1432	0	0	1
1433	0	0	0
1434	0	0	0
1435	0	0	0
1436	0	0	0
1437	0	0	0
1438	0	0	0
1439	0	0	0
1440	0	0	0
1441	0	0	0
1442	0	0	0
1443	0	0	0

```
1444          0          0          0
1445          0          0          0
1446          1          0          0
1447          0          0          0
1448          0          0          0
1449          0          1          0
1450          1          0          0
1451          0          0          0
1452          1          0          0
1453          1          0          0
1454          1          0          0
1455          1          0          0
1456          0          0          0
1457          0          0          0
1458          0          0          0

[2919 rows x 303 columns]
```

[程式]:
```
# 針對所有缺失值填入
all_data = all_data.fillna(all_data.mean())
#from sklearn.experimental import enable_iterative_imputer
#from sklearn.impute import IterativeImputer
#imp = IterativeImputer()
#all_data = imp.fit_transform(all_data)
```

[程式]:
```
# 拿回我們訓練/測試資料
X_train = all_data[:train.shape[0]]
X_test = all_data[train.shape[0]:]
y = train["SalePrice"]
```

[程式]:
```
# 使用 Lasso 當作我們的模型選擇
model = Lasso(alpha=0.0004)
model.fit(X_train, y)

# 因為有對 Sale Price 做 log1p, 所以要反向做 expm1
preds = np.expm1(model.predict(X_test))
solution = pd.DataFrame({"id":test.Id, "SalePrice":preds})

solution.to_csv("lasso.csv", index = False)
```