房價預測

線性回歸

```
In [53]: import graphlab
         import matplotlib.pyplot as plt
         %matplotlib inline
```

```
In [54]: sales = graphlab.SFrame('home_data.gl/')
```

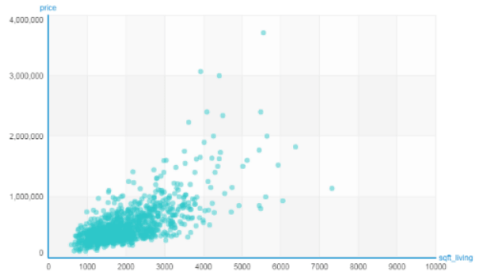
```
In [55]: sales
```

Out[55]:

id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
7129300520	2014-10-13 00:00:00+00:00	221900	3	1	1180	5650
6414100192	2014-12-09 00:00:00+00:00	538000	3	2.25	2570	7242
5631500400	2015-02-25 00:00:00+00:00	180000	2	1	770	10000
2487200875	2014-12-09 00:00:00+00:00	604000	4	3	1960	5000
1954400510	2015-02-18 00:00:00+00:00	510000	3	2	1680	8080
7237550310	2014-05-12 00:00:00+00:00	1225000	4	4.5	5420	101930
1321400060	2014-06-27 00:00:00+00:00	257500	3	2.25	1715	6819
2008000270	2015-01-15 00:00:00+00:00	291850	3	1.5	1060	9711
2414600126	2015-04-15	229500	3	1	1780	7470

資料視覺化 居住面積與價格關係

```
In [56]: graphlab.canvas.set_target('ipynb')
         sales.show(view="Scatter Plot",x="sqft_living",y="price")
```



房價分佈圖 x=price y=sqft_living

```
In [58]: sqft_model = graphlab.linear_regression.create(train_data,
                target='price',features=['sqft_living'])
```

```
PROGRESS: Creating a validation set from 5 percent of training data.
This may take a while.
          You can set ``validation_set=None`` to disable validation
 tracking.

Linear regression:
--------------------------------------------------------
Number of examples          : 16507
Number of features          : 1
Number of unpacked features : 1
Number of coefficients    : 2
Starting Newton Method
--------------------------------------------------------
+-----------+----------+--------------+--------------------+-----
----------------+---------------+----------------+
| Iteration | Passes   | Elapsed Time | Training-max_error | Vali
dation-max error | Training-rmse | Validation-rmse |
+-----------+----------+--------------+--------------------+-----
----------------+---------------+----------------+
| 1         | 2        | 0.003000     | 4362849.543679     | 1716
069.806193       | 262251.776072 | 275695.937406   |
+-----------+----------+--------------+--------------------+-----
----------------+---------------+----------------+
SUCCESS: Optimal solution found.
```

```
In [59]: print test_data['price'].mean()
```

```
543054.042563
```

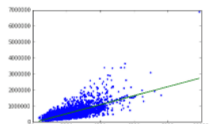
```
In [60]: print sqft_model.evaluate(test_data)
```

```
{'max_error': 4154021.349286474, 'rmse': 255166.16211062897}
```

以上看出房子平均價格54萬,而最大誤差值413萬,RMSE均方根誤差255萬

```
In [61]: plt.plot(test_data['sqft_living'],test_data['price'],'.',
                  test_data['sqft_living'],sqft_model.predict(test_data),'-')
```

```
Out[61]: [<matplotlib.lines.Line2D at 0x213f5400>,
          <matplotlib.lines.Line2D at 0x213f54a8>]
```



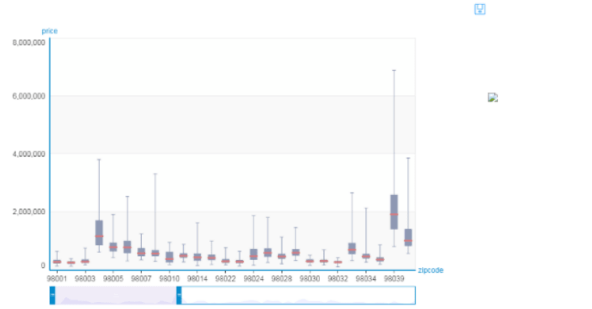
加入其他特徵修改模組，使其更為精準

```
In [75]: new_features = ['bedrooms','bathrooms','sqft_living','sqft_lot','floors','zipcode']
```

```
In [76]: sales[new_features].show()
```

bedrooms		bathrooms		sqft_living		sqft_lot		floo
dtype:	str	dtype:	str	dtype:	int	dtype:	int	dty
num_unique (est.):	13	num_unique (est.):	30	num_unique (est.):	1,036	num_unique (est.):	9,747	nun (est
num_undefined:	0	num_undefined:	0	num_undefined:	0	num_undefined:	0	nun
frequent items:		frequent items:		min:	290	min:	520	freq
3		2.5		max:	13,540	max:	1,651,359	1
4		1		median:	1,910	median:	7,617	2
2		1.75		mean:	2,079.9	mean:	15,106.968	1.5
5		2.25		std:	918.42	std:	41,419.553	3
6		2		distribution of values:		distribution of values:		2.5
1		1.5						3.5
7		2.75						
0		3						
8		3.5						
9		3.25						
10		3.75						
11		4						

```
In [77]: sales.show(view='BoxWhisker Plot',x='zipcode',y='price')
```



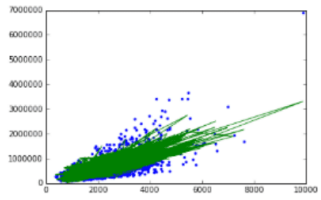
不同地區有不同的價格

加入其他特徵,使其更為精準,畢竟房價的關係不只有居住面積

```
Linear regression:
----------------------------------------------------------
Number of examples          : 16520
Number of features          : 6
Number of unpacked features : 6
Number of coefficients    : 115
Starting Newton Method
----------------------------------------------------------
+-----------+----------+--------------+--------------------+----------------------+-------------
--+-----------------+
| Iteration | Passes   | Elapsed Time | Training-max_error | Validation-max_error | Training-rms
e | Validation-rmse |
+-----------+----------+--------------+--------------------+----------------------+-------------
--+-----------------+
| 1         | 2        | 0.013001     | 2575327.231143     | 3843314.067030       | 178544.22117
5 | 250182.474302   |
+-----------+----------+--------------+--------------------+----------------------+-------------
--+-----------------+
SUCCESS: Optimal solution found.
```

```
[85]: plt.plot(test_data['sqft_living'],test_data['price'],'.',
              test_data['sqft_living'],new_features_model.predict(test_data),'-')
```

```
:[85]: [<matplotlib.lines.Line2D at 0x23540c18>,
        <matplotlib.lines.Line2D at 0x23540cc0>]
```



比較兩模組的線性,加入越多模組預測越準(基本上,選擇正確的特徵也很重要以及特徵正規化)

```
hourse1 = sales[sales['id']=='1321400060']
```

```
hourse1
```

id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floor
1321400060	2014-06-27 00:00:00+00:00	257500	3	2.25	1715	6819	2

view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	
0	3	7	1715	0	1995	0	98003	

long	sqft_living15	sqft_lot15
-122.32704857	2238.0	6819.0

[? rows x 21 columns]
Note: Only the head of the SFrame is printed. This SFrame is lazily evaluated.
You can use sf.materialize() to force materialization.

```
print hourse1['price']
```

```
[257500L, ... ]
```

```
print sqft_model.predict(hourse1)
```

```
[436786.74754503416]
```

```
print new_features_model.predict(hourse1)
```

```
[253807.60005562165]
```

同一間房子使用簡易模組誤差極大,導入越多的模組預測能力越接近真實價格

線性回歸雖然能預測房價,但實用度還是比不上多線性回歸

而過多的特徵在預測上反而表現低,極有可能造成overfitting