Workshop 9

Goal

Use python to perform logistic regression.

成功：1，失败：0

A trade is considered as success when it makes money. Success will be represented by 1 and failure by 0.

This strategy works with News and Sentiment. The column news intensity represents the number of news / minute and the column sentiment represents the news sentiment (1: good news…4: bad news).

Create a dataframe df by loading the data using read_csv

```
ts=
```

```
print (df.tail())
```

```
     success  news intensity  price  sentiment
395        0             620  60.00          2
396        0             560  45.60          3
397        0             460  39.45          2
398        0             700  54.75          2
399        0             600  58.35          3
```

Analyze the statistics of df

```
print df.XXX()
```

```
          success  news intensity       price  sentiment
count  400.000000      400.000000  400.000000  400.00000
mean     0.317500      587.700000   50.848500    2.48500
std      0.466087      115.516536    5.708502    0.94446
min      0.000000      220.000000   33.900000    1.00000
25%      0.000000      520.000000   46.950000    2.00000
50%      0.000000      580.000000   50.925000    2.00000
75%      1.000000      660.000000   55.050000    3.00000
max      1.000000      800.000000   60.000000    4.00000
```

Display the mean of each column separately

```
print df.XXX()
```

```
success          0.3175
news intensity   587.7000
price          50.8485
sentiment        2.4850
```

Since the news sentiment has only 4 levels, draw the following table using crosstab

```
print(df.crosstab(....))
```

```
          sentiment  1   2  3  4
success
0                   28  97  93  55
1                   33  54  28  12
```

Draw the histogram for each column

```
df.XXX()
```

Sentiment is a categorical variable. We are going to transform this variable into 4 dummy variables using the command get_dummies from pandas.

Example:

```
b=pd.DataFrame({'test' : pd.Series([1,2,3,1,2,3,1,1,1])})
 print(pd.get_dummies(b['test'],prefix='test'))
test_1 test_2 test_3
0     1     0     0
1     0     1     0
2     0     0     1
3     0     1     0
4     1     0     0
5     0     1     0
6     0     0     1
```

```
7    0    0    1
8    0    0    1
```

Following the previous example create dummy variables for Sentiment using the function get_dummies. Store the result into *data_dummy*

Create a joint to keep success, news intensity, price, sentiment_2, sentiment_3 and sentiment_4

```
data = df['XXXXXXXX'].join(.....)
```

```
      success  news intensity  price  sentiment_2  sentiment_3  sentiment_4
0           0             380  54.15            0            1            0
1           1             660  55.05            0            1            0
2           1             800  60.00            0            0            0
3           1             640  47.85            0            0            1
4           0             520  43.95            0            0            1
5           1             760  45.00            1            0            0
6           1             560  44.70            0            0            0
7           0             400  46.20            1            0            0
8           1             540  50.85            0            1            0
9           0             700  58.80            1            0            0
10          0             800  60.00            0            0            1
```

Add the intercept manually (a column named intersect will only 1)

Perform logistic regression.
Step 1: Remove the column name 'success'

```
colnames=
```

Step 2: Create the logistic model

```
Logit_model= sm.Logit(data['success'], data[colnames])
```

Step 3: Fi the model

```
result = logit.fit()
```

Interpret the following result:

```
print (result.summary())
                           Logit Regression Results                           
==============================================================================
Dep. Variable:                success   No. Observations:                  400
Model:                          Logit   Df Residuals:                      394
Method:                           MLE   Df Model:                            5
Date:                Sun, 20 Nov 2016   Pseudo R-squ.:                 0.08292
Time:                        02:54:02   Log-Likelihood:                -229.26
converged:                       True   LL-Null:                       -249.99
                                        LLR p-value:                 7.578e-08
==================================================================================
                     coef    std err          z      P>|z|      [95.0% Conf. Int.]
----------------------------------------------------------------------------------
news intensity     0.0023      0.001      2.070      0.038         0.000     0.004
price              0.0536      0.022      2.423      0.015         0.010     0.097
sentiment_2       -0.6754      0.316     -2.134      0.033        -1.296    -0.055
sentiment_3       -1.3402      0.345     -3.881      0.000        -2.017    -0.663
sentiment_4       -1.5515      0.418     -3.713      0.000        -2.370    -0.733
intersect         -3.9900      1.140     -3.500      0.000        -6.224    -1.756
==================================================================================
```

Calculate confidence interval with the function conf_int() associated to result

Display odds ration (just use np.exp in the params of result)