

# logistic\_regression\_breast\_cancer

December 8, 2018

```
In [65]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
%matplotlib inline

In [66]: from sklearn.preprocessing import StandardScaler

In [67]: from sklearn import preprocessing
from sklearn import neighbors
from sklearn.metrics import confusion_matrix, classification_report, precision_score
from sklearn.model_selection import train_test_split

In [68]: import statsmodels.api as sm
import statsmodels.formula.api as smf

In [69]: sns.set(style="whitegrid", color_codes=True, font_scale=1.3)

In [70]: from sklearn.datasets import load_breast_cancer

In [71]: cancer = load_breast_cancer()

In [72]: cancer.keys()

Out[72]: ['target_names', 'data', 'target', 'DESCR', 'feature_names']

In [73]: print(cancer['feature_names'])

['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
```

```

In [74]: cancer['target_names']

Out[74]: array(['malignant', 'benign'], dtype='|S9')

In [75]: cancer['data']

Out[75]: array([[1.799e+01, 1.038e+01, 1.228e+02, ..., 2.654e-01, 4.601e-01,
                  1.189e-01],
                 [2.057e+01, 1.777e+01, 1.329e+02, ..., 1.860e-01, 2.750e-01,
                  8.902e-02],
                 [1.969e+01, 2.125e+01, 1.300e+02, ..., 2.430e-01, 3.613e-01,
                  8.758e-02],
                 ...,
                 [1.660e+01, 2.808e+01, 1.083e+02, ..., 1.418e-01, 2.218e-01,
                  7.820e-02],
                 [2.060e+01, 2.933e+01, 1.401e+02, ..., 2.650e-01, 4.087e-01,
                  1.240e-01],
                 [7.760e+00, 2.454e+01, 4.792e+01, ..., 0.000e+00, 2.871e-01,
                  7.039e-02]])

In [76]: df_cancer = pd.DataFrame(cancer['data'], columns=cancer['feature_names'])

In [77]: df_cancer.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 30 columns):
mean radius           569 non-null float64
mean texture          569 non-null float64
mean perimeter        569 non-null float64
mean area             569 non-null float64
mean smoothness       569 non-null float64
mean compactness      569 non-null float64
mean concavity        569 non-null float64
mean concave points   569 non-null float64
mean symmetry         569 non-null float64
mean fractal dimension 569 non-null float64
radius error          569 non-null float64
texture error         569 non-null float64
perimeter error       569 non-null float64
area error            569 non-null float64
smoothness error      569 non-null float64
compactness error     569 non-null float64
concavity error       569 non-null float64
concave points error  569 non-null float64
symmetry error        569 non-null float64
fractal dimension error 569 non-null float64
worst radius          569 non-null float64
worst texture         569 non-null float64

```

```

worst perimeter      569 non-null float64
worst area           569 non-null float64
worst smoothness     569 non-null float64
worst compactness    569 non-null float64
worst concavity      569 non-null float64
worst concave points 569 non-null float64
worst symmetry       569 non-null float64
worst fractal dimension 569 non-null float64
dtypes: float64(30)
memory usage: 133.4 KB

```

```
In [125]: df_cancer.head()
```

```

Out[125]:   mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0         17.99         10.38         122.80       1001.0         0.11840
1         20.57         17.77         132.90       1326.0         0.08474
2         19.69         21.25         130.00       1203.0         0.10960
3         11.42         20.38          77.58        386.1         0.14250
4         20.29         14.34         135.10       1297.0         0.10030

   mean compactness  mean concavity  mean concave points  mean symmetry  \
0         0.27760         0.3001         0.14710         0.2419
1         0.07864         0.0869         0.07017         0.1812
2         0.15990         0.1974         0.12790         0.2069
3         0.28390         0.2414         0.10520         0.2597
4         0.13280         0.1980         0.10430         0.1809

   mean fractal dimension  ...  worst radius  \
0         0.07871         ...         25.38
1         0.05667         ...         24.99
2         0.05999         ...         23.57
3         0.09744         ...         14.91
4         0.05883         ...         22.54

   worst texture  worst perimeter  worst area  worst smoothness  \
0         17.33         184.60       2019.0         0.1622
1         23.41         158.80       1956.0         0.1238
2         25.53         152.50       1709.0         0.1444
3         26.50          98.87        567.7         0.2098
4         16.67         152.20       1575.0         0.1374

   worst compactness  worst concavity  worst concave points  worst symmetry  \
0         0.6656         0.7119         0.2654         0.4601
1         0.1866         0.2416         0.1860         0.2750
2         0.4245         0.4504         0.2430         0.3613
3         0.8663         0.6869         0.2575         0.6638
4         0.2050         0.4000         0.1625         0.2364

```

```

    worst fractal dimension
0          0.11890
1          0.08902
2          0.08758
3          0.17300
4          0.07678

```

```
[5 rows x 30 columns]
```

```
In [79]: cancer['target_names']
```

```
Out[79]: array(['malignant', 'benign'], dtype='<S9')

```

```
In [80]: cancer['target']
```

```

Out[80]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,
                1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,
                1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,
                1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,
                0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
                1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1,
                1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
                0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
                1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,
                1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,
                0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
                0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
                1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,
                1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1])

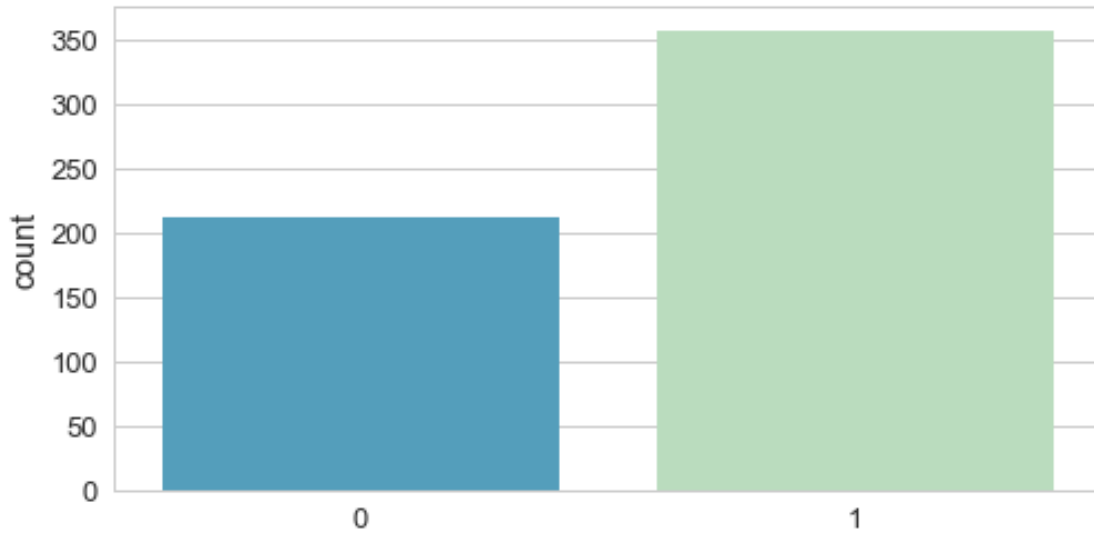
```

```

In [81]: plt.figure(figsize=(8, 4))
         sns.countplot(cancer['target'], palette='GnBu_r')

```

```
Out[81]: <matplotlib.axes._subplots.AxesSubplot at 0x1c1f973e10>
```



```
In [150]: X = cancer['data'][:, :2]
```

```
In [152]: print(X)
```

```
[[17.99 10.38]
 [20.57 17.77]
 [19.69 21.25]
 ...
 [16.6  28.08]
 [20.6  29.33]
 [ 7.76 24.54]]
```

```
In [195]: # y = (cancer['target'] != 0) * 1 == cancer['target']
```

```
In [214]: def sigmoid(z):
```

```
    return 1 / (1 + np.exp(-z))
    intercept = np.ones((X.shape[0], 1))
    X_new = np.concatenate((intercept, X), axis=1)
    theta = np.zeros(X_new.shape[1])
    max_iter = [10, 100, 1000, 10000]
    eta = [1e-1, 1e-2, 1e-3, 1e-4]
    for iter in max_iter:
        for eta_test in eta:
            for step in range(iter):
                theta = theta - eta_test * np.dot(X_new.T, (sigmoid(np.dot(X_new, theta)) - cancer['target']))
            plt.scatter(X[:,0], X[:,1], marker='o', c=y)
            y_test = -((theta[0]) + X_new.dot(theta[1])) / theta[2]
            plt.plot(X_new, y_test, 'green')
```

```
plt.title("eta{}, max_iter{}".format(eta_test,iter))
plt.xlim(left = 4)
plt.ylim(bottom = 0)
plt.show()
```

/anaconda2/lib/python2.7/site-packages/ipykernel\_launcher.py:2: RuntimeWarning: overflow encountered in ufunc

