

Logistic_reg_cancer

December 13, 2017

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

In [32]: dataset = pd.read_csv('cancer_small.csv')
X = dataset.iloc[2:, [2,3]].values
y = dataset.iloc[2:, 1].values

In [33]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state=0)

In [34]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

In [35]: from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

Out[35]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
penalty='l2', random_state=0, solver='liblinear', tol=0.0001,
verbose=0, warm_start=False)

In [36]: y_pred = classifier.predict(X_test)

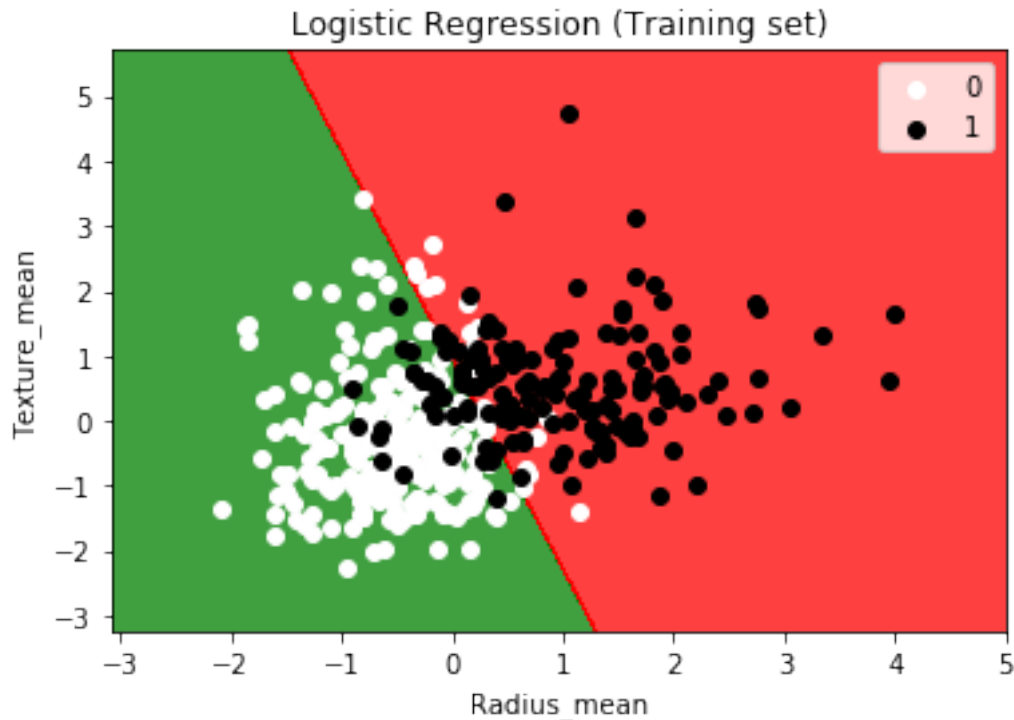
In [37]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

In [49]: from matplotlib.colors import ListedColormap
X_set, y_set = X_train, y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1,
                             step = 0.5),
                     np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1,
                             step = 0.5))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),
             alpha = 0.75, cmap = ListedColormap(('green', 'red')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
```

```

for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('white', 'black'))(i), label = j)
plt.title('Logistic Regression (Training set)')
plt.xlabel('Radius_mean')
plt.ylabel('Texture_mean')
plt.legend()
plt.show()

```



```

In [50]: from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max(
                                np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max(
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(
                                alpha = 0.75, cmap = ListedColormap(('green', 'red'))))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('white', 'black'))(i), label = j)
plt.title('Logistic Regression (Test set)')
plt.xlabel('Radius_mean')
plt.ylabel('Texture_mean')

```

```
plt.legend()  
plt.show()
```

