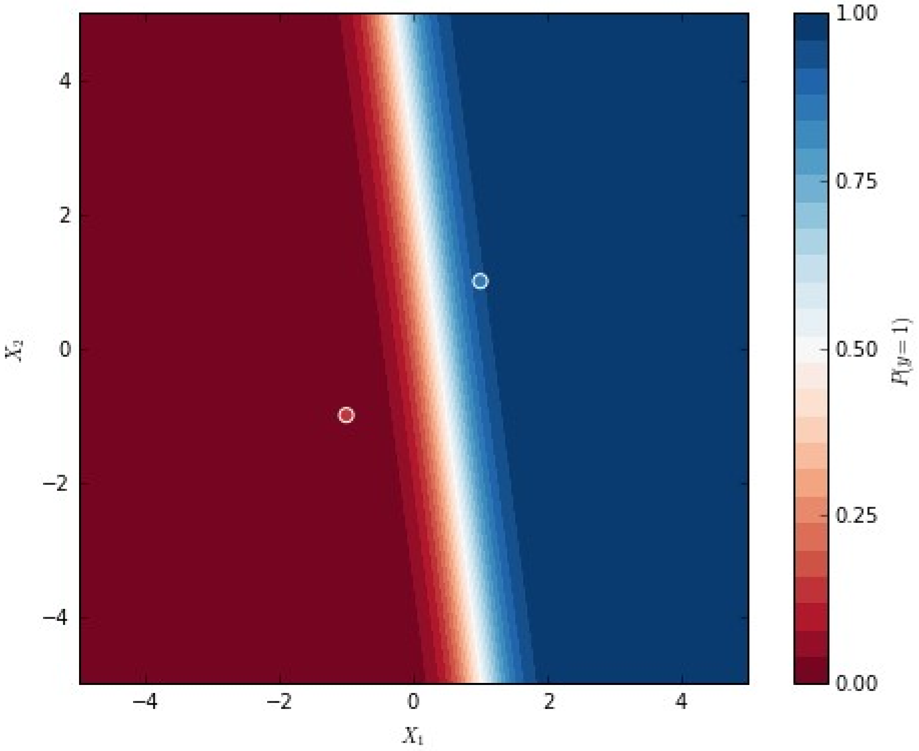
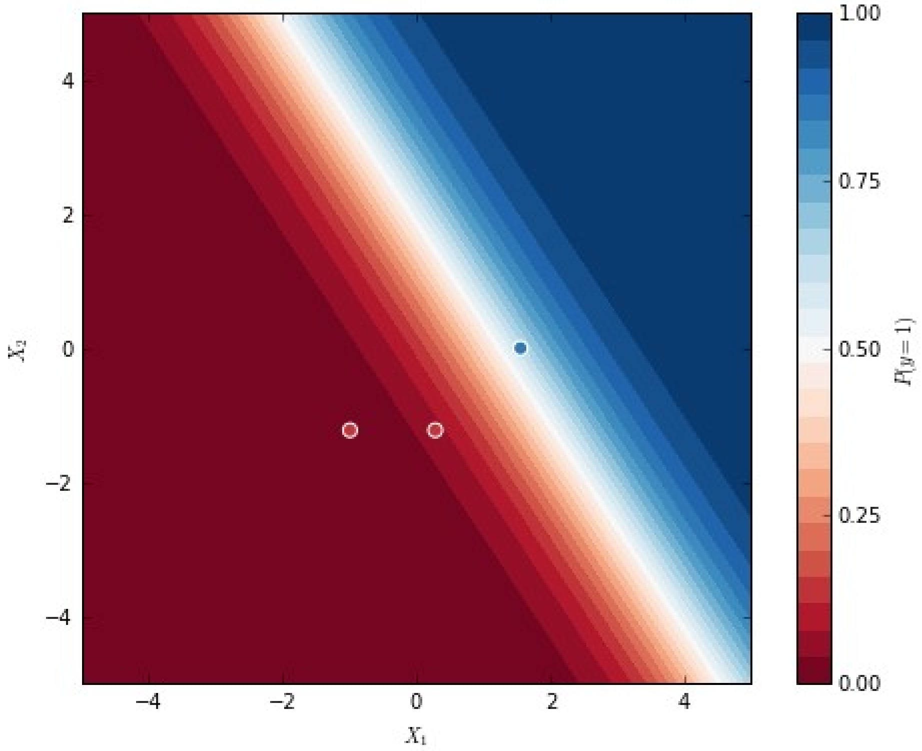
```
Python 3.5.6 |Anaconda 4.2.0 (64-bit)| (default, Aug 26 2018, 16:05:27) [MSC v.1900 64
bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: 

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

In [2]: phishingData = pd.read_csv('phishing.txt')
   ...: X = phishingData.iloc[:,:-1].values
   ...: y = phishingData.iloc[:,30].values

In [3]: from sklearn.cross_validation import train_test_split
   ...: X_train,X_test,y_train,y_test = train_test_split
(X,y,test_size=0.3,random_state=4)
   ...: 

In [4]: from sklearn.preprocessing import StandardScaler
   ...: scalar = StandardScaler()
   ...: X_train = scalar.fit_transform (X_train)
   ...: X_test = scalar.fit_transform (X_test)
   ...: 
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)

In [5]: from sklearn.linear_model import LogisticRegression
   ...: LRclassifier = LogisticRegression(C=100,random_state=0)
   ...: LRclassifier.fit(X_train,y_train)
   ...: 
   ...: LRpredict = LRclassifier.predict(X_test)

In [6]: LRclassifier.score(X_train,y_train)
   ...: 
Out[6]: 0.92981775882124851

In [7]: LRclassifier.score(X_test,y_test)
Out[7]: 0.92734398552909258

In [8]: from sklearn.metrics import confusion_matrix
   ...: confusionMatrix = confusion_matrix(y_test,LRpredict)
   ...:

In [9]: confusionMatrix
Out[9]:
array([[1357,  153],
       [  88, 1719]])

In [10]: X = phishingData.iloc[0:5,[6,14]].values
    ...: y = phishingData.iloc[0:5,30].values
    ...:

In [11]: from sklearn.cross_validation import train_test_split
    ...: X_train,X_test,y_train,y_test = train_test_split
(X,y,test_size=0.3,random_state=4)
    ...:

In [12]: from sklearn.preprocessing import StandardScaler
    ...: scalar = StandardScaler()
    ...: X_train = scalar.fit_transform (X_train)
    ...: X_test = scalar.fit_transform (X_test)
    ...:
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)

In [13]: from sklearn.linear_model import LogisticRegression
    ...: LRclassifier1 = LogisticRegression(C=100,random_state=0)
    ...: LRclassifier1.fit(X_train,y_train)
    ...:
Out[13]:
LogisticRegression(C=100, 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 [14]: LRpredict1 = LRclassifier1.predict(X_test)
    ...:

In [15]: LRclassifier1.score(X_train,y_train)
    ...:
Out[15]: 1.0

In [16]: LRclassifier1.score(X_test,y_test)
    ...:
Out[16]: 1.0

In [17]: from sklearn.metrics import confusion_matrix
    ...: LRconfusionMatrix1 = confusion_matrix(y_test,LRpredict1)
```

```
In [18]: LRconfusionMatrix1
Out[18]:
array([[1, 0],
       [0, 1]])

In [19]: xx, yy = np.mgrid[-5:5:.01, -5:5:.01]
    ...: grid = np.c_[xx.ravel(), yy.ravel()]
    ...: probs = LRclassifier1.predict_proba(grid)[:, 1].reshape(xx.shape)
    ...:
    ...: print(probs)
    ...:
    ...: f, ax = plt.subplots(figsize=(8, 6))
    ...: contour = ax.contourf(xx, yy, probs, 25, cmap="RdBu",
    ...:                       vmin=0, vmax=1)
    ...: ax_c = f.colorbar(contour)
    ...: ax_c.set_label("$P(y = 1)$")
    ...: ax_c.set_ticks([0, .25, .5, .75, 1])
    ...:
    ...: ax.scatter(X_test[:, 0], X_test[:, 1],c = (y_test == 1 ), s=50,
    ...:            cmap="RdBu", vmin=-.2, vmax=1.2,
    ...:            edgecolor="white", linewidth=1)
    ...:
    ...: ax.set(aspect="equal",
    ...:        xlim=(-5, 5), ylim=(-5, 5),
    ...:        xlabel="$X_1$", ylabel="$X_2$")
    ...:
    ...: plt.show()
    ...:
[[  8.12989106e-11   8.17055455e-11   8.21142142e-11 ...,   1.17595783e-08
    1.18183965e-08   1.18775089e-08]
 [  8.44905622e-11   8.49131608e-11   8.53378731e-11 ...,   1.22212386e-08
    1.22823659e-08   1.23437989e-08]
 [  8.78075123e-11   8.82467013e-11   8.86880871e-11 ...,   1.27010228e-08
    1.27645498e-08   1.28283946e-08]
 ...,
 [  9.99999739e-01   9.99999740e-01   9.99999742e-01 ...,   9.99999998e-01
    9.99999998e-01   9.99999998e-01]
 [  9.99999749e-01   9.99999750e-01   9.99999751e-01 ...,   9.99999998e-01
    9.99999998e-01   9.99999998e-01]
 [  9.99999758e-01   9.99999760e-01   9.99999761e-01 ...,   9.99999998e-01
    9.99999998e-01   9.99999998e-01]]
```



```
In [20]: X = phishingData.iloc[0:13,[6,14]].values
    ...: y = phishingData.iloc[0:13,30].values
    ...:

In [21]: from sklearn.cross_validation import train_test_split
    ...: X_train,X_test,y_train,y_test = train_test_split
(X,y,test_size=0.3,random_state=4)
    ...:
    ...: #perform feature scaling

In [22]: from sklearn.preprocessing import StandardScaler
    ...: scalar = StandardScaler()
    ...: X_train = scalar.fit_transform (X_train)
    ...: X_test = scalar.transform (X_test)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)
C:\Users\SRIKANT\Anaconda3\lib\site-packages\sklearn\utils\validation.py:420:
DataConversionWarning: Data with input dtype int64 was converted to float64 by
StandardScaler.
  warnings.warn(msg, DataConversionWarning)

In [23]: from sklearn.linear_model import LogisticRegression
    ...: LRclassifier11 = LogisticRegression(C=100,random_state=0)
    ...: LRclassifier11.fit(X_train,y_train)
Out[23]:
LogisticRegression(C=100, 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 [24]: LRpredict11 = LRclassifier11.predict(X_test)

In [25]: LRclassifier11.score(X_train,y_train)
    ...:
Out[25]: 0.8888888888888884

In [26]: LRclassifier11.score(X_test,y_test)
    ...:
Out[26]: 1.0

In [27]: from sklearn.metrics import confusion_matrix
    ...: LRconfusionMatrix11 = confusion_matrix(y_test,LRpredict11)
    ...:

In [28]: LRconfusionMatrix11
Out[28]:
array([[2, 0],
       [0, 2]])

In [29]: xx, yy = np.mgrid[-5:5:.01, -5:5:.01]
    ...: grid = np.c_[xx.ravel(), yy.ravel()]
    ...: probs = LRclassifier11.predict_proba(grid)[:, 1].reshape(xx.shape)
    ...:
    ...: print(probs)
    ...:
    ...: f, ax = plt.subplots(figsize=(8, 6))
    ...: contour = ax.contourf(xx, yy, probs, 25, cmap="RdBu",
    ...:                       vmin=0, vmax=1)
    ...: ax_c = f.colorbar(contour)
    ...: ax_c.set_label("$P(y = 1)$")
    ...: ax_c.set_ticks([0, .25, .5, .75, 1])
    ...:
    ...: ax.scatter(X_test[:, 0], X_test[:, 1],c = (y_test == 1 ), s=50,
    ...:            cmap="RdBu", vmin=-.2, vmax=1.2,
    ...:            edgecolor="white", linewidth=1)
    ...:
    ...: ax.set(aspect="equal",
    ...:        xlim=(-5, 5), ylim=(-5, 5),
    ...:        xlabel="$X_1$", ylabel="$X_2$")
    ...:
    ...: plt.show()
    ...:
[[  5.10217931e-07   5.15342853e-07   5.20519253e-07 ...,   1.07298055e-02
    1.08364140e-02   1.09440700e-02]
 [  5.18016238e-07   5.23219491e-07   5.28475009e-07 ...,   1.08920165e-02
    1.10002188e-02   1.11094838e-02]
 [  5.25933737e-07   5.31216518e-07   5.36552362e-07 ...,   1.10566524e-02
    1.11664717e-02   1.12773693e-02]
 ...,
 [  6.53557676e-01   6.55817138e-01   6.58069573e-01 ...,   9.99975065e-01
    9.99975313e-01   9.99975558e-01]
 [  6.56984116e-01   6.59232882e-01   6.61474502e-01 ...,   9.99975440e-01
    9.99975684e-01   9.99975926e-01]
 [  6.60394275e-01   6.62632166e-01   6.64862792e-01 ...,   9.99975810e-01
    9.99976050e-01   9.99976288e-01]]
```



```
In [30]:
```