credit_fraud_LinearSVC

April 24, 2018

```
In [7]: %matplotlib inline
        import matplotlib.pyplot as plt
        import pandas as pd
        import numpy as np
        from sklearn.multiclass import OneVsRestClassifier
        from sklearn import svm
        from sklearn.model_selection import train_test_split
```

```
In [8]: df = pd.read_csv('notebooks/My/creditcard.csv')
```

```
In [9]: df.head()
```

```
Out[9]:    Time        V1        V2        V3        V4        V5        V6        V7  \
        0   0.0 -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388  0.239599
        1   0.0  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361 -0.078803
        2   1.0 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  0.791461
        3   1.0 -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203  0.237609
        4   2.0 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921  0.592941

                 V8        V9  ...         V21       V22       V23       V24  \
        0  0.098698  0.363787  ...   -0.018307  0.277838 -0.110474  0.066928
        1  0.085102 -0.255425  ...   -0.225775 -0.638672  0.101288 -0.339846
        2  0.247676 -1.514654  ...    0.247998  0.771679  0.909412 -0.689281
        3  0.377436 -1.387024  ...   -0.108300  0.005274 -0.190321 -1.175575
        4 -0.270533  0.817739  ...   -0.009431  0.798278 -0.137458  0.141267

                V25       V26       V27       V28  Amount  Class
        0  0.128539 -0.189115  0.133558 -0.021053  149.62      0
        1  0.167170  0.125895 -0.008983  0.014724    2.69      0
        2 -0.327642 -0.139097 -0.055353 -0.059752  378.66      0
        3  0.647376 -0.221929  0.062723  0.061458  123.50      0
        4 -0.206010  0.502292  0.219422  0.215153   69.99      0

        [5 rows x 31 columns]
```

```
In [10]: y=df['Class']
         x=df
         x = x.drop('Class',axis=1)
         x.head()
         y.head()

Out[10]: 0    0
         1    0
         2    0
         3    0
         4    0
         Name: Class, dtype: int64

In [11]: x_train,x_test,y_train,t_test=train_test_split(x,y)

In [12]: # clf = svm.SVC()
         clf = OneVsRestClassifier(svm.LinearSVC(), n_jobs=-1)

In [13]: clf.fit(x_train,y_train)

Out[13]: OneVsRestClassifier(estimator=LinearSVC(C=1.0, class_weight=None, dual=True, fit_interc
              intercept_scaling=1, loss='squared_hinge', max_iter=1000,
              multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
              verbose=0),
                   n_jobs=-1)

In [14]: clf.score(x_test,t_test)

Out[14]: 0.9982865649841297

In [15]: predicted = clf.predict(x_test)

In [16]: np.mean((predicted-t_test)**2)

Out[16]: 0.0017134350158703408

In [17]: # x = x.values
         # y = y.values
         #plt.figure(1, figsize=(6, 4))
         # plt.clf()

         # plt.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1], s=80,facecolors='
         # plt.scatter(x[:, 0], x[:, 1], c=y, zorder=10, cmap=plt.cm.Paired,edgecolors='k')

         # plt.axis('tight')
         # x_min = -3
         # x_max = 3
         # y_min = -3
         # y_max = 3
         # import numpy as np
         # XX, YY = np.mgrid[x_min:x_max:200j, y_min:y_max:200j]
         # Z = clf.decision_function(np.c_[x.ravel(), y.ravel()])
#     # Put the result into a color plot
# Z = Z.reshape(XX.shape)
# plt.figure(fignum, figsize=(4, 3))
# plt.pcolormesh(XX, YY, Z > 0, cmap=plt.cm.Paired)
# plt.contour(XX, YY, Z, colors=['y', 'g', 'p'], linestyles=['--', '-', '--'], levels=[

# plt.xlim(x_min, x_max)
# plt.ylim(y_min, y_max)

# plt.xticks(())
# plt.yticks(())
# plt.show()
```