Kaggle_Scikit

November 19, 2016

```
In [1]: import pandas as pd

In [2]: train = pd.read_csv('http://bit.ly/kaggletrain')

In [4]: train.head(2)

Out[4]:    PassengerId  Survived  Pclass  \
        0            1         0       3
        1            2         1       1

                                                  Name     Sex   Age  SibSp
        0                        Braund, Mr. Owen Harris    male  22.0      1
        1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1

           Parch     Ticket     Fare Cabin Embarked
        0      0  A/5 21171   7.2500   NaN        S
        1      0   PC 17599  71.2833   C85        C

In [5]: feature_cols = ['Pclass', 'Parch']

In [7]: X = train.loc[:, feature_cols]

In [8]: X.shape

Out[8]: (891, 2)

In [9]: y = train.Survived

In [10]: y.shape

Out[10]: (891,)

In [11]: from sklearn.linear_model import LogisticRegression
         logreg = LogisticRegression()
         logreg.fit(X,y)

Out[11]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=Tru
                   intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                   penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                   verbose=0, warm_start=False)
```

```
In [12]: test = pd.read_csv('http://bit.ly/kaggletest')

In [13]: test.head()

Out[13]:    PassengerId  Pclass                                          Name     S
         0          892       3                              Kelly, Mr. James    ma
         1          893       3              Wilkes, Mrs. James (Ellen Needs)  fema
         2          894       2                      Myles, Mr. Thomas Francis    ma
         3          895       3                              Wirz, Mr. Albert    ma
         4          896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  fema

            Age  SibSp  Parch   Ticket     Fare Cabin Embarked
         0  34.5      0      0   330911   7.8292   NaN        Q
         1  47.0      1      0   363272   7.0000   NaN        S
         2  62.0      0      0   240276   9.6875   NaN        Q
         3  27.0      0      0   315154   8.6625   NaN        S
         4  22.0      1      1  3101298  12.2875   NaN        S

In [14]: X_new = test.loc[:, feature_cols]

In [15]: X_new.shape

Out[15]: (418, 2)

In [16]: new_pred_class = logreg.predict(X_new)

In [17]: test.PassengerId

Out[17]: 0         892
         1         893
         2         894
         3         895
         4         896
         5         897
         6         898
         7         899
         8         900
         9         901
         10        902
         11        903
         12        904
         13        905
         14        906
         15        907
         16        908
         17        909
         18        910
         19        911
         20        912
```

```
         21      913
         22      914
         23      915
         24      916
         25      917
         26      918
         27      919
         28      920
         29      921
               ...
         388    1280
         389    1281
         390    1282
         391    1283
         392    1284
         393    1285
         394    1286
         395    1287
         396    1288
         397    1289
         398    1290
         399    1291
         400    1292
         401    1293
         402    1294
         403    1295
         404    1296
         405    1297
         406    1298
         407    1299
         408    1300
         409    1301
         410    1302
         411    1303
         412    1304
         413    1305
         414    1306
         415    1307
         416    1308
         417    1309
         Name: PassengerId, dtype: int64

In [18]: new_pred_class

Out[18]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1
                1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0
                1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1
                1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

```
       1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0
       0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0
       0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1
       0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0
       0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0
       0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0
       1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1
       1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0
       1, 0, 0, 0])
```

In [21]: pd.DataFrame({'PassengerId' : test.PassengerId, 'Survived' : new_pred_clas

In [ ]:
```