

In [1]:

```
1 import os,sys,time
2 import numpy as np
3 from sklearn.metrics import accuracy_score
4 from sklearn.metrics import f1_score
5 from sklearn.metrics import recall_score
6 from sklearn.metrics import precision_score
7 from sklearn.metrics import confusion_matrix
8 from sklearn.metrics import f1_score
9 from sklearn.metrics import mean_absolute_error
10 from sklearn.metrics import mean_squared_error
11
12 from sklearn import datasets
13 from sklearn.model_selection import train_test_split
```

In [2]:

```

1  # KNN
2  from sklearn.neighbors import KNeighborsClassifier
3
4  # let's try to use pandas' fast csv reader
5  try:
6      import pandas
7      readcsv = lambda f, c, t=np.float64: pandas.read_csv(f, usecols=c, delimiter=','
8  except:
9      # fall back to numpy loadtxt
10     readcsv = lambda f, c, t=np.float64: np.loadtxt(f, usecols=c, delimiter=',',
11
12 start = time.clock() # 开始计时
13
14 # Input data set parameters
15 train_file = os.path.join('data', 'k_nearest_neighbors_train.csv')
16 predict_file = os.path.join('data', 'k_nearest_neighbors_test.csv')
17
18 # Read data. Let's use 5 features per observation
19 nFeatures = 5
20 nClasses = 5
21 train_data = readcsv(train_file, range(nFeatures))
22 train_labels = readcsv(train_file, range(nFeatures, nFeatures+1))
23 predict_data = readcsv(predict_file, range(nFeatures))
24 predict_labels = readcsv(predict_file, range(nFeatures, nFeatures+1))
25
26 knn = KNeighborsClassifier()#得到分类器
27
28 knn.fit(train_data, train_labels)#训练模型
29
30 predictedLabel = knn.predict(predict_data) # 进行预测
31 end=time.clock() #结束计时
32 print("time", end-start)
33
34 print("accuracy_score", accuracy_score(predict_labels, predictedLabel))
35 print("precision_score", precision_score(predict_labels, predictedLabel, average="micro"))
36 print("recall_score", recall_score(predict_labels, predictedLabel, average="micro"))
37 print("f1_score", f1_score(predict_labels, predictedLabel, average="micro"))
38 print("confusion_matrix\n", confusion_matrix(predict_labels, predictedLabel))
39 """
40 micro算法是指把所有的类放在一起算，具体到precision，就是把所有类的TP加和，再除以所有类的TP和
41 因此micro方法下的precision和recall都等于accuracy
42 """

```

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher
er.py:12: DeprecationWarning: time.clock has been deprecated in Python
3.3 and will be removed from Python 3.8: use time.perf_counter or time.process_time instead
if sys.path[0] == '':

```

```
time 0.36130699999999996  
accuracy_score 0.9655  
precision_score 0.9650210710710710
```

In [3]:

```

1  # 支持向量机
2  from sklearn.svm import SVC
3
4  # let's try to use pandas' fast csv reader
5  try:
6      import pandas
7      readcsv = lambda f, c, t=np.float64: pandas.read_csv(f, usecols=c, delimiter
8  except:
9      # fall back to numpy loadtxt
10     readcsv = lambda f, c, t=np.float64: np.loadtxt(f, usecols=c, delimiter=',',
11
12 start = time.clock() # 开始计时
13 # input data file
14 infile = "./data/svm_two_class_train_dense.csv"
15 testfile = "./data/svm_two_class_test_dense.csv"
16
17 data = readcsv(infile, range(20))
18 labels = readcsv(infile, range(20,21))
19 pdata = readcsv(testfile, range(20))
20 plabels = readcsv(testfile, range(20,21))
21
22 clf = SVC()
23 clf.fit(data, labels)
24 predictedLabel = clf.predict(pdata)
25
26 end=time.clock() #结束计时
27 print("time", end-start)
28
29 print("accuracy_score", accuracy_score(plabels, predictedLabel))
30 print("precision_score", precision_score(plabels, predictedLabel, average="macro")
31 print("recall_score", recall_score(plabels, predictedLabel, average="macro"))
32 print("f1_score", f1_score(plabels, predictedLabel, average="macro"))
33 print("confusion_matrix\n", confusion_matrix(plabels, predictedLabel))
34
35

```

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher
r.py:12: DeprecationWarning: time.clock has been deprecated in Python
3.3 and will be removed from Python 3.8: use time.perf_counter or tim
e.process_time instead

```

```

if sys.path[0] == '':

time 0.137589999999999988
accuracy_score 0.986
precision_score 0.9864077669902913
recall_score 0.9859719438877755
f1_score 0.985996415082261
confusion_matrix
[[1002    0]
 [ 28  970]]

```

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/sklearn/utils/vali
dation.py:73: DataConversionWarning: A column-vector y was passed w
hen a 1d array was expected. Please change the shape of y to (n_sampl
es, ), for example using ravel().

```

```

return f(**kwargs)
/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launch
er.py:26: DeprecationWarning: time.clock has been deprecated in Pytho

```

n 3.3 and will be removed from Python 3.8: use `time.perf_counter` or `time.process_time` instead

In [4]:

```

1  # 朴素贝叶斯
2  from sklearn.naive_bayes import GaussianNB
3
4  # let's try to use pandas' fast csv reader
5  try:
6      import pandas
7      read_csv = lambda f, c, t=np.float64: pandas.read_csv(f, usecols=c, delimiter=','
8  except:
9      # fall back to numpy loadtxt
10     read_csv = lambda f, c, t=np.float64: np.loadtxt(f, usecols=c, delimiter=',',
11
12 start = time.clock() # 开始计时
13
14 # input data file
15 infile = "./data/naivebayes_train_dense.csv"
16 testfile = "./data/naivebayes_test_dense.csv"
17 data = readcsv(infile, range(20))
18 labels = readcsv(infile, range(20,21))
19 pdata = readcsv(testfile, range(20))
20 plabels = readcsv(testfile, range(20,21))
21
22 gnb = GaussianNB()
23 predictedLabel = gnb.fit(data, labels).predict(pdata)
24
25 end=time.clock() #结束计时
26 print("time", end-start)
27
28 print("accuracy_score", accuracy_score(plabels, predictedLabel))
29 print("precision_score", precision_score(plabels, predictedLabel, average="macro"))
30 print("recall_score", recall_score(plabels, predictedLabel, average="macro"))
31 print("f1_score", f1_score(plabels, predictedLabel, average="macro"))
32 print("confusion_matrix\n", confusion_matrix(plabels, predictedLabel))
33
34

```

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher
r.py:12: DeprecationWarning: time.clock has been deprecated in Python
3.3 and will be removed from Python 3.8: use time.perf_counter or tim
e.process_time instead
    if sys.path[0] == '':

```

```

time 0.21341600000000005
accuracy_score 1.0
precision_score 1.0
recall_score 1.0
f1_score 1.0
confusion_matrix
[[ 83   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
0
  [ 0   0]
[ 0  90   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
0
  [ 0   0]
[ 0   0 104   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
0
  [ 0   0]
[ 0   0   0  91   0   0   0   0   0   0   0   0   0   0   0   0   0]
0
  [ 0   0]

```

```

[ 0 0 0 0 0 95 0 0 0 0 0 0 0 0 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 93 0 0 0 0 0 0 0 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 97 0 0 0 0 0 0 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 106 0 0 0 0 0 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 92 0 0 0 0 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 105 0 0 0 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 107 0 0 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 98 0 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 115 0 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 103 0 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 109 0 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 109 0
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 106
0
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
97
0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0
101 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0
0 99]]

```

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/sklearn/utils/validation.py:73: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().

```

```

    return f(**kwargs)

```

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher.py:25: DeprecationWarning: time.clock has been deprecated in Python 3.3 and will be removed from Python 3.8: use time.perf_counter or time.process_time instead

```

In [5]:

```

1  # adaboost
2  from sklearn.ensemble import AdaBoostClassifier
3  from sklearn.tree import DecisionTreeClassifier
4  from sklearn.datasets import make_gaussian_quantiles
5
6  # let's try to use pandas' fast csv reader
7  try:
8      import pandas
9      read_csv = lambda f, c, t=np.float64: pandas.read_csv(f, usecols=c, delimiter=','
10 except:
11     # fall back to numpy loadtxt
12     read_csv = lambda f, c, t=np.float64: np.loadtxt(f, usecols=c, delimiter=',',
13
14 start = time.clock() # 开始计时
15
16 infile = "./data/adaboost_train.csv"
17 testfile = "./data/adaboost_test.csv"
18
19 # Read data. Let's have 20 independent, and 1 dependent variable (for each obser
20 indep_data = readcsv(infile, range(20))
21 dep_data = readcsv(infile, range(20,21))
22 pdata = readcsv(testfile, range(20))
23 predict_labels = np.loadtxt(testfile, usecols=range(20,21), delimiter=',', ndmir
24
25 bdt = AdaBoostClassifier(DecisionTreeClassifier(max_depth=2, min_samples_split=2
26                        algorithm="SAMME",
27                        n_estimators=200, learning_rate=0.8)
28 bdt.fit(indep_data, dep_data)
29 predictedLabel = bdt.predict(pdata)
30
31 end=time.clock() #结束计时
32 print("time", end-start)
33
34 print("accuracy_score", accuracy_score(predict_labels, predictedLabel))
35 print("precision_score", precision_score(predict_labels, predictedLabel))
36 print("recall_score", recall_score(predict_labels, predictedLabel))
37 print("f1_score", f1_score(predict_labels, predictedLabel))
38 print("confusion_matrix\n", confusion_matrix(predict_labels, predictedLabel))
39
40
41

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher.py:14: DeprecationWarning: time.clock has been deprecated in Python 3.3 and will be removed from Python 3.8: use time.perf_counter or time.process_time instead

```

time 0.23176100000000001
accuracy_score 1.0
precision_score 1.0
recall_score 1.0
f1_score 1.0
confusion_matrix
[[1599    0]
 [   0  401]]

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/sklearn/utils/val


```
idation.py:73: DataConversionWarning: A column-vector y was passed whe  
n a 1d array was expected. Please change the shape of y to (n_samples,  
) , for example using ravel().  
    return f(**kwargs)  
/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launche  
r.py:31: DeprecationWarning: time.clock has been deprecated in Python  
3.3 and will be removed from Python 3.8: use time.perf_counter or tim  
e.process_time instead
```

In [6]:

```

1  # 随机森林
2  from sklearn.ensemble import RandomForestClassifier
3  from sklearn.datasets import make_classification
4  # let's try to use pandas' fast csv reader
5  try:
6      import pandas
7      read_csv = lambda f, c, t=np.float64: pandas.read_csv(f, usecols=c, delimiter=',')
8  except:
9      # fall back to numpy loadtxt
10     read_csv = lambda f, c, t=np.float64: np.loadtxt(f, usecols=c, delimiter=',')
11
12  # Get Intel(R) Data Analytics Acceleration Library (Intel(R) DAAL) version
13
14  start = time.clock() # 开始计时
15
16  # input data file
17  infile = "./data/df_classification_train.csv"
18  testfile = "./data/df_classification_test.csv"
19  # Read data. Let's use 3 features per observation
20  data = readcsv(infile, range(3), t=np.float32)
21  labels = readcsv(infile, range(3,4), t=np.float32)
22  pdata = readcsv(testfile, range(3), t=np.float32)
23  plabels = readcsv(testfile, range(3,4), t=np.float32)
24
25  clf = RandomForestClassifier(max_depth=3, random_state=0)
26  clf.fit(data, labels)
27  predictedLabel = clf.predict(pdata)
28
29  end=time.clock() #结束计时
30  print("time", end-start)
31
32  print("accuracy_score", accuracy_score(plabels, predictedLabel))
33  print("precision_score", precision_score(plabels, predictedLabel, average="macro"))
34  print("recall_score", recall_score(plabels, predictedLabel, average="macro"))
35  print("f1_score", f1_score(plabels, predictedLabel, average="macro"))
36  print("confusion_matrix\n", confusion_matrix(plabels, predictedLabel))
37
38

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher.py:14: DeprecationWarning: time.clock has been deprecated in Python 3.3 and will be removed from Python 3.8: use time.perf_counter or time.process_time instead

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher.py:26: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

```

time 3.9923739999999994
accuracy_score 0.363
precision_score 0.17200956937799045
recall_score 0.24420454545454545
f1_score 0.16618381618381614
confusion_matrix
[[ 0  0 163  0  0]
 [ 0  0 160  0  0]
 [ 0  0 318  2  0]

```

```
[ 0  0 153  45  0]
[ 0  0 118  41  0]]
```

```
/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher.py:29: DeprecationWarning: time.clock has been deprecated in Python 3.3 and will be removed from Python 3.8: use time.perf_counter or time.process_time instead
/opt/anaconda3/envs/daal/lib/python3.7/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

In [7]:

```
1 # EM算法
2 from sklearn.mixture import GaussianMixture
3 start = time.clock() # 开始计时
4
5 #鸢尾花数据集
6 X, y = datasets.load_iris(return_X_y=True)
7 #X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.3)
8
9 gmm = GaussianMixture(n_components=3)
10
11 predictedLabel = gmm.fit_predict(X)
12 end=time.clock() #结束计时
13 print("time", end-start)
14
15 print("accuracy_score", accuracy_score(y, predictedLabel))
16 print("precision_score", precision_score(y, predictedLabel, average="macro"))
17 print("recall_score", recall_score(y, predictedLabel, average="macro"))
18 print("f1_score", f1_score(y, predictedLabel, average="macro"))
19 print("confusion_matrix\n", confusion_matrix(y, predictedLabel))
20
21
22
```

```
/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher.py:3: DeprecationWarning: time.clock has been deprecated in Python 3.3 and will be removed from Python 3.8: use time.perf_counter or time.process_time instead
```

This is separate from the ipykernel package so we can avoid doing imports until

```
/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher.py:12: DeprecationWarning: time.clock has been deprecated in Python 3.3 and will be removed from Python 3.8: use time.perf_counter or time.process_time instead
```

```
if sys.path[0] == '':

time 0.024150000000000056
accuracy_score 0.36666666666666664
precision_score 0.3636363636363636
recall_score 0.36666666666666667
f1_score 0.3650793650793651
confusion_matrix
[[50  0  0]
 [ 0  5 45]
 [ 0 50  0]]
```

In [8]:

```

1  #随机森林 回归
2
3  from sklearn.ensemble import RandomForestRegressor
4  from sklearn.datasets import make_regression
5
6  # let's try to use pandas' fast csv reader
7  try:
8      import pandas
9      read_csv = lambda f, c, t=np.float64: pandas.read_csv(f, usecols=c, delimiter=',')
10 except:
11     # fall back to numpy loadtxt
12     read_csv = lambda f, c, t=np.float64: np.loadtxt(f, usecols=c, delimiter=',')
13
14
15 start = time.clock() # 开始计时
16 infile = "./data/df_regression_train.csv"
17 testfile = "./data/df_regression_test.csv"
18 # Read data. Let's have 13 independent, and 1 dependent variables (for each observation)
19 indep_data = readcsv(infile, range(13), t=np.float32)
20 dep_data = readcsv(infile, range(13,14), t=np.float32)
21 pdata = readcsv(testfile, range(13), t=np.float32)
22 ptdata = readcsv(testfile, range(13,14), t=np.float32)
23
24 regr = RandomForestRegressor(n_estimators=100, random_state=0)
25 regr.fit(indep_data, dep_data)
26 predict_result = regr.predict(pdata)
27
28 end=time.clock() #结束计时
29 print("time", end-start)
30
31 print("MAE", mean_absolute_error(ptdata, predict_result))
32 print("MSE", mean_squared_error(ptdata, predict_result))

```

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher
r.py:15: DeprecationWarning: time.clock has been deprecated in Python
3.3 and will be removed from Python 3.8: use time.perf_counter or tim
e.process_time instead

```

```

from ipykernel import kernelapp as app
/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher
r.py:25: DataConversionWarning: A column-vector y was passed when a 1d
array was expected. Please change the shape of y to (n_samples,), for
example using ravel().

```

```

time 0.31520400000000005
MAE 2.511535433070865
MSE 12.542506133858259

```

```

/opt/anaconda3/envs/daal/lib/python3.7/site-packages/ipykernel_launcher
r.py:28: DeprecationWarning: time.clock has been deprecated in Python
3.3 and will be removed from Python 3.8: use time.perf_counter or tim
e.process_time instead

```

In []:

1	
---	--

In []:

1	
---	--