2017 Scala programming in Spark Homework 7

2018/01/04

范真瑋

Part I: Spark Statistic & Machine Learning Practice

1. BinaryClassificationMetricsExample.scala

```
Threshold: 4.016860619821217E-20, F-score: 0.8518518518518519, Beta = 1
Threshold: 2.4431947801289467E-21, F-score: 0.8363636363636363, Beta = 1
Threshold: 1.3562225203357031E-21, F-score: 0.8214285714285715, Beta = 1
Threshold: 2.0245539016166966E-23, F-score: 0.8070175438596492, Beta = 1
Threshold: 4.1289766496320484E-26, F-score: 0.7931034482758621, Beta = 1
Threshold: 1.7379382366748526E-27, F-score: 0.7796610169491525, Beta = 1
Threshold: 3.5358917301048076E-31, F-score: 0.7666666666666667, Beta = 1
Threshold: 3.064241439023516E-32, F-score: 0.7540983606557378, Beta = 1
Threshold: 2.4931035213678524E-33, F-score: 0.7419354838709677, Beta = 1
Threshold: 1.1366707910957578E-33, F-score: 0.7301587301587301, Beta = 1
Threshold: 1.2295559395714E-34, F-score: 0.71875, Beta = 1
Threshold: 8.485702779479716E-36, F-score: 0.7076923076923077, Beta = 1
Threshold: 2.7175943470211644E-36, F-score: 0.6969696969696969, Beta = 1
Threshold: 1.8277277022436243E-8, F-score: 0.9787234042553191, Beta = 0.5
Threshold: 1.0, F-score: 0.29629629629629634, Beta = 0.5
Threshold: 0.9999999999999998, F-score: 0.3571428571428571, Beta = 0.5
Threshold: 0.9999999999999996, F-score: 0.41379310344827586, Beta = 0.5
Threshold: 0.9999999999999984, F-score: 0.4666666666666667, Beta = 0.5
Threshold: 0.9999999999999964, F-score: 0.5161290322580645, Beta = 0.5
Threshold: 3.763845367411271E-11, F-score: 0.9583333333333334, Beta = 0.5
Threshold: 0.9999999999999252, F-score: 0.5625, Beta = 0.5
Threshold: 0.9999999999998983, F-score: 0.6060606060606061, Beta = 0.5
Threshold: 0.9999999999997895, F-score: 0.6470588235294118, Beta = 0.5
Threshold: 0.9999999999997462, F-score: 0.6857142857142856, Beta = 0.5
Threshold: 0.9999999999996714, F-score: 0.7222222222222222, Beta = 0.5
Threshold: 0.9999999999994154, F-score: 0.7567567567567568, Beta = 0.5
Threshold: 0.9999999999992724, F-score: 0.7894736842105263, Beta = 0.5
Threshold: 0.9999999999992462, F-score: 0.8205128205128205, Beta = 0.5
Threshold: 0.9999999998299527, F-score: 0.85, Beta = 0.5
Threshold: 0.9999999995588322, F-score: 0.878048780487805, Beta = 0.5
Threshold: 0.9999999984316541, F-score: 0.9047619047619047, Beta = 0.5
Threshold: 0.9999999159471471, F-score: 0.9302325581395349, Beta = 0.5
Threshold: 0.9999994847753312, F-score: 0.9545454545454545, Beta = 0.5
Threshold: 0.008307776528273265, F-score: 0.9777777777777777, Beta = 0.5
Threshold: 1.294409115662929E-14, F-score: 0.9387755102040816, Beta = 0.5
Threshold: 2.266520744760814E-5, F-score: 1.0, Beta = 0.5
Threshold: 9.280128947907467E-16, F-score: 0.92, Beta = 0.5
Threshold: 2.4754640595100176E-17, F-score: 0.9019607843137255, Beta = 0.5
Threshold: 1.9972204406612271E-19, F-score: 0.8846153846153846, Beta = 0.5
Threshold: 7.424317994690846E-20, F-score: 0.8679245283018869, Beta = 0.5
Threshold: 4.016860619821217E-20, F-score: 0.8518518518518519, Beta = 0.5
Threshold: 2.4431947801289467E-21, F-score: 0.8363636363636363, Beta = 0.5
Threshold: 1.3562225203357031E-21, F-score: 0.8214285714285715, Beta = 0.5
Threshold: 2.0245539016166966E-23, F-score: 0.8070175438596492, Beta = 0.5
Threshold: 4.1289766496320484E-26, F-score: 0.7931034482758621, Beta = 0.5
Threshold: 1.7379382366748526E-27, F-score: 0.7796610169491525, Beta = 0.5
Threshold: 3.5358917301048076E-31, F-score: 0.7666666666666667, Beta = 0.5
Threshold: 3.064241439023516E-32, F-score: 0.7540983606557378, Beta = 0.5
Threshold: 2.4931035213678524E-33, F-score: 0.7419354838709677, Beta = 0.5
Threshold: 1.1366707910957578E-33, F-score: 0.7301587301587301, Beta = 0.5
Threshold: 1.2295559395714E-34, F-score: 0.71875, Beta = 0.5
Threshold: 8.485702779479716E-36, F-score: 0.7076923076923077, Beta = 0.5
Threshold: 2.7175943470211644E-36, F-score: 0.6969696969696969, Beta = 0.5
Area under precision-recall curve = 1.0
Area under ROC = 1.0
```

訓練二元分類模型，使用不同的評估指標進行評估。

2. ChiSqSelectorExample.scala

根據獨立的卡方測試對特徵進行排序，然後選擇排序最高的特徵。

3. DecisionTreeRegressionExample.scala

```
scala> :load HW7_3.scala
Loading HW7_3.scala...
Test Mean Squared Error = 0.0
Learned regression tree model:
DecisionTreeModel regressor of depth 2 with 5 nodes
  If (feature 434 <= 0.0)
   If (feature 100 <= 165.0)
    Predict: 0.0
   Else (feature 100 > 165.0)
    Predict: 1.0
  Else (feature 434 > 0.0)
   Predict: 1.0
```

決策樹，通過不斷設置新的條件標準對當前的數據進行劃分。

4. GradientBoostingClassificationExample.scala

```
scala> :load HW7_4.scala
Loading HW7_4.scala...
Test Error = 0.0
Learned classification GBT model:
TreeEnsembleModel classifier with 3 trees

  Tree 0:
    If (feature 434 <= 0.0)
     If (feature 100 <= 165.0)
      Predict: -1.0
     Else (feature 100 > 165.0)
      Predict: 1.0
    Else (feature 434 > 0.0)
     Predict: 1.0
  Tree 1:
    If (feature 540 <= 0.0)
     If (feature 179 <= 32.0)
      If (feature 323 <= 252.0)
       Predict: 0.47681168808847024
      Else (feature 323 > 252.0)
       Predict: 0.47681168808847024
     Else (feature 179 > 32.0)
      Predict: 0.4768116880884712
    Else (feature 540 > 0.0)
     If (feature 491 <= 40.0)
      If (feature 157 <= 0.0)
       If (feature 123 <= 20.0)
        Predict: -0.4768116880884702
       Else (feature 123 > 20.0)
        Predict: -0.4768116880884703
      Else (feature 157 > 0.0)
       Predict: -0.4768116880884703
     Else (feature 491 > 40.0)
      Predict: 0.4768116880884694
  Tree 2:
    If (feature 434 <= 0.0)
     If (feature 293 <= 253.0)
      If (feature 289 <= 8.0)
       Predict: -0.4381935810427206
      Else (feature 289 > 8.0)
       Predict: -0.4381935810427206
     Else (feature 293 > 253.0)
      Predict: 0.43819358104271977
    Else (feature 434 > 0.0)
     If (feature 234 <= 0.0)
      If (feature 319 <= 92.0)
       Predict: 0.4381935810427206
      Else (feature 319 > 92.0)
       Predict: 0.43819358104272155
     Else (feature 234 > 0.0)
      Predict: 0.43819358104272155
```

Gradient Boosting，每一次建立模型是在之前建立模型損失函數的梯度下降方向。(Gradient Boosting 分類)

5. **GradientBoostingRegressionExample.scala**

```
scala> :load HW7_5.scala
Loading HW7_5.scala...
Test Mean Squared Error = 0.06896551724137932
Learned regression GBT model:
TreeEnsembleModel regressor with 3 trees

  Tree 0:
    If (feature 406 <= 72.0)
     Predict: 0.0
    Else (feature 406 > 72.0)
     Predict: 1.0
  Tree 1:
    Predict: 0.0
  Tree 2:
    Predict: 0.0
```

(Gradient Boosting 回歸)

6. **IsotonicRegressionExample.scala**

```
scala> :load HW7_6.scala
Loading HW7_6.scala...
Mean Squared Error = 0.008860256490591361
```

保序迴歸，具有分段迴歸的效果，採用平方誤差。

7. **KernelDensityEstimationExample.scala**

```
scala> :load HW7_7.scala
Loading HW7_7.scala...
0.04145944023341912
0.07902016933085627
0.08962920127312339
```

根據已知的樣本估計未知的密度，屬於非參數檢驗方法之一。原理是觀察某一事物的已知分佈，如果某一個數在觀察中出現了，可認為這個數的機率密度很大，和這個數比較近的數的機率密度也會較大，而離這個數遠的數的機率密度會較小。

8. PCAExample.scala

```
scala> :load HW7_8.scala
Loading HW7_8.scala...
warning: there was one deprecation warning; re-run with -deprecation for details
warning: there was one deprecation warning; re-run with -deprecation for details
18/01/04 22:33:20 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeSystemLAPACK
18/01/04 22:33:20 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeRefLAPACK
warning: there was one deprecation warning; re-run with -deprecation for details
warning: there was one deprecation warning; re-run with -deprecation for details
18/01/04 22:33:23 WARN LinearRegressionWithSGD: The input data is not directly cached, which may hurt performance if its parent RDDs are also uncached.
18/01/04 22:33:23 WARN LinearRegressionWithSGD: The input data was not directly cached, which may hurt performance if its parent RDDs are also uncached.
Mean Squared Error = 8.817683208094909
PCA Mean Squared Error = 7.33878951845746
```

主成分分析，可以將特徵向量投影到低維空間，實現對特徵向量的降維。

9. RandomForestClassificationExample.scala

```
scala> :load HW7_9.scala
Loading HW7_9.scala...
Test Error = 0.02857142857142857
Learned classification forest model:
TreeEnsembleModel classifier with 3 trees

  Tree 0:
    If (feature 433 <= 0.0)
     Predict: 0.0
    Else (feature 433 > 0.0)
     Predict: 1.0
  Tree 1:
    If (feature 434 <= 0.0)
     Predict: 0.0
    Else (feature 434 > 0.0)
     Predict: 1.0
  Tree 2:
    If (feature 468 <= 12.0)
     If (feature 481 <= 57.0)
      Predict: 1.0
     Else (feature 481 > 57.0)
      Predict: 0.0
    Else (feature 468 > 12.0)
     Predict: 0.0
```

隨機森林，由多個決策樹構成的森林，分類結果由這些決策樹投票得到。(隨機森林分類)

10. RandomForestRegressionExample.scala

```
scala> :load HW7_10.scala
Loading HW7_10.scala...
Test Mean Squared Error = 0.03535353535353536
Learned regression forest model:
TreeEnsembleModel regressor with 3 trees

  Tree 0:
    If (feature 489 <= 11.0)
     If (feature 296 <= 253.0)
      Predict: 0.0
     Else (feature 296 > 253.0)
      Predict: 1.0
    Else (feature 489 > 11.0)
     Predict: 1.0
  Tree 1:
    If (feature 399 <= 0.0)
     If (feature 466 <= 221.0)
      Predict: 1.0
     Else (feature 466 > 221.0)
      Predict: 0.0
    Else (feature 399 > 0.0)
     Predict: 0.0
  Tree 2:
    If (feature 462 <= 0.0)
     Predict: 0.0
    Else (feature 462 > 0.0)
     Predict: 1.0
```

(隨機森林回歸)