Zeppelin Notebook

Search your Notebooks

anonymous

spark-ml-lr-wisconsin-breastcancer

default

FINISHED

```
import org.apache.spark._
import org.apache.spark.rdd.RDD

import org.apache.spark.sql.SQLContext

import org.apache.spark.sql._
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._

import org.apache.spark._
import org.apache.spark.rdd.RDD
import org.apache.spark.sql.SQLContext
import org.apache.spark.sql._
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
```

Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:04 PM.

FINISHED

```
/*
 * https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)
 * Array(1000025,5,1,1,1,2,1,3,1,1,2)
   0. Sample code number            id number
   1. Clump Thickness               1 - 10
   2. Uniformity of Cell Size       1 - 10
   3. Uniformity of Cell Shape      1 - 10
   4. Marginal Adhesion             1 - 10
   5. Single Epithelial Cell Size   1 - 10
   6. Bare Nuclei                   1 - 10
   7. Bland Chromatin               1 - 10
   8. Normal Nucleoli               1 - 10
   9. Mitoses                       1 - 10
  10. Class:                        (2 for benign, 4 for malignant)
 */
case class Obs(clazz: Double, thickness: Double, size: Double, shape: Double, madh: Double, epsize: Double, bnuc: Double, bchrom: Double, nNuc: Double, mit: Double)
 def parseObs(line: Array[Double]): Obs = {
    Obs(
      if (line(9) == 4.0) 1 else 0, line(0), line(1), line(2), line(3), line(4), line(5), line(6), line(7), line(8)
    )
  }


defined class Obs
parseObs: (line: Array[Double])Obs
```

Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:05 PM.

FINISHED

```
def parseRDD(rdd: RDD[String]): RDD[Array[Double]] = {
    rdd.map(line => line.split(",").filter(fields => fields(6) != "?").map(fields => fields.drop(1)).map(fields => fields.map(field => field.toDouble))
}


parseRDD: (rdd: org.apache.spark.rdd.RDD[String])org.apache.spark.rdd.RDD[Array[Double]]
```

Took 1 sec. Last updated by anonymous at January 25 2017, 11:11:06 PM.

FINISHED

```
val obsText: RDD[String] = sc.textFile("/home/anindya/zeppelin-spark-projects/spark-ml-lr-wisconsin-breastcancer/data/wbcd.csv")
val obsRDD: RDD[Obs] = parseRDD(obsText).map(parseObs)
val obsDF: DataFrame = obsRDD.toDF().cache()

obsText: org.apache.spark.rdd.RDD[String] = /home/anindya/zeppelin-spark-projects/spark-ml-lr-wisconsin-breastcancer/data/wbcd.csv MapPartitionsRDD[27185] at textFile at <console>:185
obsRDD: org.apache.spark.rdd.RDD[Obs] = MapPartitionsRDD[27190] at map at <console>:193
obsDF: org.apache.spark.sql.DataFrame = [clazz: double, thickness: double ... 8 more fields]
```

Took 3 sec. Last updated by anonymous at January 25 2017, 11:11:08 PM.

FINISHED

```
obsDF.show(10)

+-----+-----------+----+----+-----+------+----+-------+-----+----+
|clazz|thickness|size|shape|madh|epsize|bnuc|bchrom|nNuc|mit|
+-----+-----------+----+----+-----+------+----+-------+-----+----+
|  0.0|     5.0| 1.0|  1.0|  1.0|   2.0| 1.0|   3.0| 1.0|1.0|
|  0.0|     5.0| 4.0|  4.0|  5.0|   7.0|10.0|   3.0| 2.0|1.0|
|  0.0|     3.0| 1.0|  1.0|  1.0|   2.0| 2.0|   3.0| 1.0|1.0|
|  0.0|     6.0| 8.0|  8.0|  1.0|   3.0| 4.0|   3.0| 7.0|1.0|
|  0.0|     4.0| 1.0|  1.0|  3.0|   2.0| 1.0|   3.0| 1.0|1.0|
|  1.0|     8.0|10.0| 10.0|  8.0|   7.0|10.0|   9.0| 7.0|1.0|
|  0.0|     1.0| 1.0|  1.0|  1.0|   2.0|10.0|   3.0| 1.0|1.0|
|  0.0|     2.0| 1.0|  2.0|  1.0|   2.0| 1.0|   3.0| 1.0|1.0|
|  0.0|     2.0| 1.0|  1.0|  1.0|   2.0| 1.0|   1.0| 1.0|5.0|
|  0.0|     4.0| 2.0|  1.0|  1.0|   2.0| 1.0|   2.0| 1.0|1.0|
+-----+-----------+----+----+-----+------+----+-------+-----+----+
only showing top 10 rows
```

Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:08 PM.

FINISHED

```
obsDF.printSchema

root
 |-- clazz: double (nullable = false)
 |-- thickness: double (nullable = false)
 |-- size: double (nullable = false)
 |-- shape: double (nullable = false)
 |-- madh: double (nullable = false)
 |-- epsize: double (nullable = false)
 |-- bnuc: double (nullable = false)
 |-- bchrom: double (nullable = false)
 |-- nNuc: double (nullable = false)
 |-- mit: double (nullable = false)
```

Took 1 sec. Last updated by anonymous at January 25 2017, 11:11:09 PM.

FINISHED

```
import org.apache.spark.ml.feature.VectorAssembler
val featureCols: Array[String] = Array("thickness", "size", "shape", "madh", "epsize", "bnuc", "bchrom", "nNuc", "mit")
val assembler: VectorAssembler = new VectorAssembler().setInputCols(featureCols).setOutputCol("features")
val assembledDF: DataFrame = assembler.transform(obsDF)
assembledDF.sample(withReplacement=false,fraction=0.01).show()


featureCols: Array[String] = Array(thickness, size, shape, madh, epsize, bnuc, bchrom, nNuc, mit)
assembler: org.apache.spark.ml.feature.VectorAssembler = vecAssembler_cfc66ca548c0
assembledDF: org.apache.spark.sql.DataFrame = [clazz: double, thickness: double ... 9 more fields]
+-----+-----------+----+----+-----+------+----+-------+-----+----+--------------------+
|clazz|thickness|size|shape|madh|epsize|bnuc|bchrom|nNuc|mit|            features|
+-----+-----------+----+----+-----+------+----+-------+-----+----+--------------------+
|  0.0|     3.0| 1.0|  1.0|  1.0|   2.0| 3.0|   3.0| 1.0|1.0|[3.0,1.0,1.0,1.0,...|
|  1.0|     7.0| 3.0|  4.0|  4.0|   3.0| 3.0|   3.0| 2.0|7.0|[7.0,3.0,4.0,4.0,...|
|  0.0|     3.0| 1.0|  1.0|  1.0|   3.0| 1.0|   2.0| 1.0|1.0|[3.0,1.0,1.0,1.0,...|
|  0.0|     1.0| 1.0|  1.0|  1.0|   1.0| 1.0|   2.0| 1.0|1.0|[1.0,1.0,1.0,1.0,...|
|  0.0|     1.0| 1.0|  3.0|  1.0|   2.0| 1.0|   1.0| 1.0|1.0|[1.0,1.0,3.0,1.0,...|
|  0.0|     5.0| 1.0|  1.0|  1.0|   2.0| 1.0|   2.0| 1.0|1.0|[5.0,1.0,1.0,1.0,...|
+-----+-----------+----+----+-----+------+----+-------+-----+----+--------------------+
```

Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:11 PM. (outdated)

FINISHED

```
import org.apache.spark.ml.feature.StringIndexer
val labelIndexer: StringIndexer = new StringIndexer().setInputCol("clazz").setOutputCol("label")
val labelIndexedDF: DataFrame = labelIndexer.fit(assembledDF).transform(assembledDF)
labelIndexedDF.show()

|  0.0|     4.0| 1.0|  1.0|  3.0|   2.0| 1.0|   3.0| 1.0|1.0|[4.0,1.0,1.0,3.0,...|  0.0|
|  1.0|     8.0|10.0| 10.0|  8.0|   7.0|10.0|   9.0| 7.0|1.0|[8.0,10.0,10.0,8.0,...|  1.0|
|  0.0|     1.0| 1.0|  1.0|  1.0|   2.0|10.0|   3.0| 1.0|1.0|[1.0,1.0,1.0,1.0,...|  0.0|
|  0.0|     2.0| 1.0|  2.0|  1.0|   2.0| 1.0|   3.0| 1.0|1.0|[2.0,1.0,2.0,1.0,...|  0.0|
|  0.0|     2.0| 1.0|  1.0|  1.0|   2.0| 1.0|   1.0| 1.0|5.0|[2.0,1.0,1.0,1.0,...|  0.0|
|  0.0|     4.0| 2.0|  1.0|  1.0|   2.0| 1.0|   2.0| 1.0|1.0|[4.0,2.0,1.0,1.0,...|  0.0|
|  0.0|     1.0| 1.0|  1.0|  1.0|   1.0| 1.0|   3.0| 1.0|1.0|[1.0,1.0,1.0,1.0,...|  0.0|
|  0.0|     2.0| 1.0|  1.0|  1.0|   2.0| 1.0|   2.0| 1.0|1.0|[2.0,1.0,1.0,1.0,...|  0.0|
|  1.0|     5.0| 3.0|  3.0|  3.0|   2.0| 3.0|   4.0| 4.0|1.0|[5.0,3.0,3.0,3.0,...|  1.0|
|  0.0|     1.0| 1.0|  1.0|  1.0|   2.0| 3.0|   3.0| 1.0|1.0|[1.0,1.0,1.0,1.0,...|  0.0|
|  1.0|     8.0| 7.0|  5.0|10.0|   7.0| 9.0|   5.0| 5.0|4.0|[8.0,7.0,5.0,10.0,...|  1.0|
|  1.0|     7.0| 4.0|  6.0| 4.0|   6.0| 1.0|   4.0| 3.0|1.0|[7.0,4.0,6.0,4.0,...|  1.0|
|  0.0|     4.0| 1.0|  1.0|  1.0|   2.0| 1.0|   2.0| 1.0|1.0|[4.0,1.0,1.0,1.0,...|  0.0|
|  0.0|     4.0| 1.0|  1.0|  1.0|   2.0| 1.0|   3.0| 1.0|1.0|[4.0,1.0,1.0,1.0,...|  0.0|
|  1.0|    10.0| 7.0|  7.0| 6.0|   4.0|10.0|   4.0| 1.0|2.0|[10.0,7.0,7.0,6.0...|  1.0|
|  0.0|     6.0| 1.0|  1.0|  1.0|   2.0| 1.0|   3.0| 1.0|1.0|[6.0,1.0,1.0,1.0,...|  0.0|
+-----+-----------+----+----+-----+------+----+-------+-----+----+--------------------+-----+
only showing top 20 rows
```

Took 3 sec. Last updated by anonymous at January 25 2017, 11:11:13 PM.

FINISHED

```
val splitSeed: Int = 5043
val Array(trainingData, testData) = labelIndexedDF.randomSplit(Array(0.7, 0.3), seed=splitSeed)

splitSeed: Int = 5043
trainingData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [clazz: double, thickness: double ... 10 more fields]
testData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [clazz: double, thickness: double ... 10 more fields]
```

Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:14 PM.

FINISHED

```
import org.apache.spark.ml.classification.LogisticRegression
import org.apache.spark.ml.classification.LogisticRegressionModel
val lr: LogisticRegression = new LogisticRegression().setMaxIter(10).setRegParam(0.3).setElasticNetParam(0.8)
val model: LogisticRegressionModel = lr.fit(trainingData)

import org.apache.spark.ml.classification.LogisticRegression
import org.apache.spark.ml.classification.LogisticRegressionModel
lr: org.apache.spark.ml.classification.LogisticRegression = logreg_b8f10eeab02c
model: org.apache.spark.ml.classification.LogisticRegressionModel = logreg_b8f10eeab02c
```

Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:16 PM.

FINISHED

```
val predictions: DataFrame = model.transform(testData)
predictions.sample(withReplacement=false,fraction=0.1).select("clazz", "label", "prediction").show(10)

predictions: org.apache.spark.sql.DataFrame = [clazz: double, thickness: double ... 13 more fields]
+-----+-----+----------+
|clazz|label|prediction|
+-----+-----+----------+
|  0.0|  0.0|       0.0|
|  0.0|  0.0|       0.0|
|  0.0|  0.0|       0.0|
|  0.0|  0.0|       0.0|
|  0.0|  0.0|       0.0|
|  1.0|  1.0|       0.0|
|  1.0|  1.0|       1.0|
|  0.0|  0.0|       0.0|
|  0.0|  0.0|       0.0|
|  0.0|  0.0|       0.0|
+-----+-----+----------+
only showing top 10 rows
```

Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:17 PM.

FINISHED

```
import org.apache.spark.ml.classification.LogisticRegressionTrainingSummary
val trainingSummary: LogisticRegressionTrainingSummary = model.summary

import org.apache.spark.ml.classification.LogisticRegressionTrainingSummary
trainingSummary: org.apache.spark.ml.classification.LogisticRegressionTrainingSummary = org.apache.spark.ml.classification.BinaryLogisticRegressionTrainingSummary@62c85d6f
```

Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:18 PM.

FINISHED

```
val objectiveHistory: Array[Double] = trainingSummary.objectiveHistory
objectiveHistory.foreach(loss => println(loss))

0.6469335347082146
0.6417044783862828
0.6279056511527211
0.6222099171715268
0.6107931957424084
0.6043687436643167
0.5992662865597
0.5976641305656915
0.5942531448167665
0.594141456502649 2
0.5941236192228847
objectiveHistory: Array[Double] = Array(0.6469335347082146, 0.6417044783862828, 0.6279056511527211, 0.6222099171715268, 0.6107931957424084, 0.6043687436643167, 0.5992662865597, 0.5976641305656915, 0.5942531448167665, 0.594141456502649 2, 0.5941236192228847)
```

Took 1 sec. Last updated by anonymous at January 25 2017, 11:11:18 PM.

FINISHED

```
import org.apache.spark.ml.classification.BinaryLogisticRegressionSummary
val binarySummary: BinaryLogisticRegressionSummary = trainingSummary.asInstanceOf[BinaryLogisticRegressionSummary]

import org.apache.spark.ml.classification.BinaryLogisticRegressionSummary
binarySummary: org.apache.spark.ml.classification.BinaryLogisticRegressionSummary = org.apache.spark.ml.classification.BinaryLogisticRegressionTrainingSummary@62c85d6f
```

Took 1 sec. Last updated by anonymous at January 25 2017, 11:11:19 PM.

FINISHED

```
// Obtain the receiver-operating characteristic as a dataframe and areaUnderROC.
val roc: DataFrame = binarySummary.roc
roc.show()
println(binarySummary.areaUnderROC)

roc: org.apache.spark.sql.DataFrame = [FPR: double, TPR: double]
+--------------------+---+
|FPR|                 TPR|
+--------------------+---+
|0.0|                 0.0|
|0.0|0.047317278106508875|
|0.0| 0.8591715976331360|
|0.0| 0.0828402366863905 4|
|0.0|  0.12426035502958 58|
|0.0|  0.1301775147928994|
|0.0| 0.1369046745562130 2|
|0.0| 0.1420118343195266 3|
|0.0| 0.1479289940828402 4|
|0.0| 0.1538461538461538 5|
|0.0| 0.1597633136094674 7|
|0.0| 0.1656804733727810 8|
|0.0| 0.1715976331360946 6|
|0.0| 0.1775147928994082 7|
+--------------------+---+
Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:21 PM.
```

FINISHED

```
// Set the model threshold to maximize F-Measure
val fMeasure: DataFrame = binarySummary.fMeasureByThreshold
val maxFMeasure: Double = fMeasure.select(max("F-Measure")).head().getDouble(0)


fMeasure: org.apache.spark.sql.DataFrame = [threshold: double, F-Measure: double]
maxFMeasure: Double = 0.9623188405797102
Took 3 sec. Last updated by anonymous at January 25 2017, 11:11:22 PM.
```

FINISHED

```
val bestThreshold: Double = fMeasure.where($"F-Measure" === maxFMeasure).select("threshold").head().getDouble(0)
model.setThreshold(bestThreshold)

bestThreshold: Double = 0.3173119651739381 7
res31: model.type = logreg_b8f10eeab02c
Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:23 PM.
```

FINISHED

```
import  org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
val evaluator: BinaryClassificationEvaluator = new BinaryClassificationEvaluator().setLabelCol("label")
val accuracy: Double = evaluator.evaluate(predictions)

import org.apache.spark.ml.evaluation.BinaryClassificationEvaluator
evaluator: org.apache.spark.ml.evaluation.BinaryClassificationEvaluator = binEval_b24ec7cbfd04
accuracy: Double = 0.9926910299003322
Took 2 sec. Last updated by anonymous at January 25 2017, 11:11:24 PM.
```

FINISHED

```
model.write.overwrite().save("/home/anindya/zeppelin-spark-projects/spark-ml-lr-wisconsin-breastcancer/model")
```

Took 1 sec. Last updated by anonymous at January 25 2017, 11:11:25 PM.

FINISHED

Took 0 sec. Last updated by anonymous at January 25 2017, 11:11:25 PM.