算法文档

一、分类算法

1.逻辑回归

算法接口：
inputDatabas: 输入数据库
inputTable: 数据表
outputDatabase: 输出数据库
outputTable: 输出表
numClasses: 待分类种类数目
intercept: 是否使用截距，默认 false
validate: 是否验证训练集，默认 true
train: 训练集占，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class
com.vigortech.bigdata.LogisticRegressionApplication --driver-memory 5g --executor-memory 5g
--driver-cores 10 --num-executors 10 --verbose
/usr/myfiles/algos/LogisticRegression/target/scala-2.10/LogisticRegression_Demo-assembly-2.0.j
ar '{"inputDatabase": "default","inputTable": "test_table","outputDatabase":
"default","outputTable": "test_predictions", "numClasses": 3,"intercept": "false","validate":
"true","train": 0.7,"test": 0.3,"targetName": "target","algo": "LR", "trackId": "流水号 1"}'

2.支持向量机

算法接口：
inputDatabase: 输入数据库
inputTable: 输入表
outputDatabase: 输出数据库
outputTable: 输出表
stepSize: 搜索步长，0 到 1
miniBatch: 批量系数，0 到 1
iterations: 迭代次数
regParam: 正则系数，0 到 1

updater: 正则函数，L1 或 L2
train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class com.vigortech.bigdata.SVMApplication --driver-memory 5g --executor-memory 5g --driver-cores 10 --num-executors 10 --verbose /usr/myfiles/algos/SVM/target/scala-2.10/SVM_Demo-assembly-2.0.jar '{"inputDatabase": "default","inputTable": "breast_cancer","outputDatabase": "default","outputTable": "test_predictions", "stepSize": 1.0,"miniBatch": 1.0,"iterations": 100,"regParam": 0.01,"updater": "L1","train": 0.7,"test": 0.3,"targetName": "class","algo": "SVM", "trackId": "流水号 2"}'

3.素朴贝叶斯

算法接口：
inputDatabase: 输入数据库
inputTable: 输入表
outputDatabase: 输出数据库
outputTable: 输出表
lambda: lambda 系数，0 到 1
modelType: 模型类型，bernoulli 或 multinomial
train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class com.vigortech.bigdata.NaiveBayesApplication --driver-memory 5g --executor-memory 5g --driver-cores 10 --num-executors 10 --verbose /usr/myfiles/algos/NaiveBayes/target/scala-2.10/NaiveBayes_Demo-assembly-1.0.jar '{"inputDatabase": "default","inputTable": "test_table","outputDatabase": "default","outputTable": "test_predictions", "lambda": 1.0,"modelType": "multinomial","train": 0.7,"test": 0.3,"targetName": "target","algo": "NB", "trackId": "流水号 3"}'

4.决策树分类

算法接口：
inputDatabase: 输入数据库
inputTable: 输入表

outputDatabase: 输出数据库
outputTable: 输出表
numClasses: 待分类种类数目
impurity: 分割标准，gini，entropy 或 variance
maxDepth: 树最大深度，0 到 20
maxBins: 最大分箱数目，0 到 20
train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class com.vigortech.bigdata.DecisionTreeClassification --driver-memory 5g --executor-memory 5g --driver-cores 10 --num-executors 10 --verbose /usr/myfiles/algos/DecisionTrees/target/scala-2.10/DecisionTreeClassification_Demo-assembly-1.0.jar '{"inputDatabase": "default","inputTable": "test_table","outputDatabase": "default","outputTable": "test_predictions", "numClasses": 3,"impurity": "gini","maxDepth": 5,"maxBins": 3,"train": 0.7,"test": 0.3,"targetName": "target","algo": "DTC", "trackId": "流水号4"}'

5.随机森林

算法接口：
inputDatabase: 输入数据库
inputTable: 输入表
outputDatabase: 输出数据库
outputTable: 输出表
numClasses: 待分类种类数目
impurity: 分割标准，gini，entropy 或 variance
maxDepth: 树最大深度，0 到 20
maxBins: 最大分箱数目，0 到 20
numTrees: 决策树数量，0 到 100
featuresSubsetStrategy: 特征抽样策略，默认 auto
train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class com.vigortech.bigdata.RandomForestApplication --driver-memory 5g --executor-memory 5g

--driver-cores 10 --num-executors 10 --verbose
/usr/myfiles/algos/RandomForest/target/scala-2.10/RandomForestApplication_Demo-assembly-1.0.jar '{"inputDatabase": "default","inputTable": "test_table","outputDatabase": "default","outputTable": "test_predictions", "numClasses": 3,"impurity": "gini","maxDepth": 5,"maxBins": 3,"numTrees": 50,"featureSubsetStrategy": "auto","train": 0.7,"test": 0.3,"targetName": "target","algo": "RF", "trackId": "流水号 5"}'

6.GBDT

算法接口：
inputDatabase: 输入数据库
inputTable: 输入表
outputDatabase: 输出数据库
outputTable: 输出表
numClasses: 待分类种类数目
defaultParams: classification 或 regression，默认 classification
maxDepth: 树最大深度，0 到 20
iterations: 迭代次数，0 到 100
train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class com.vigortech.bigdata.GBDTApplication --driver-memory 5g --executor-memory 5g --driver-cores 10 --num-executors 10 --verbose /usr/myfiles/algos/GBDT/target/scala-2.10/GBDTApplication_Demo-assembly-1.0.jar '{"inputDatabase": "default","inputTable": "breast_cancer","outputDatabase": "default","outputTable": "test_predictions", "numClasses": 2,"defaultParams": "classification","maxDepth": 5,"iterations": 50,"train": 0.7,"test": 0.3,"targetName": "class","algo": "GBDT", "trackId": "流水号 6"}'

二、回归算法

1.线性回归

算法接口：
inputDatabase: 输入数据库
inputTable: 输入表
outputDatabase: 输出数据库
outputTable: 输出表
iterations: 迭代次数
stepSize: 训练步长，0 到 1

train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class
com.vigortech.bigdata.LinearRegressionApplication --driver-memory 5g --executor-memory 5g
--driver-cores 10 --num-executors 10 --verbose
/usr/myfiles/algos/LinearRegression/target/scala-2.10/LinearRegression_Demo-assembly-1.0.jar
'{"inputDatabase": "default","inputTable": "concrete","outputDatabase": "default","outputTable":
"test_predictions", "iterations": 50,"stepSize": 0.000001,"train": 0.7,"test": 0.3,"targetName":
"concrete_compresive_strength","algo": "Linear", "trackId": "流水号 7"}'

2.Ridge

算法接口：
inputDatabase: 输入数据库
inputTable: 输入表
outputDatabase: 输出数据库
outputTable: 输出表
iterations: 迭代次数，0 到 100
stepSize: 训练步长，0 到 1
regParam: 正则项系数，0 到 1
train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class
com.vigortech.bigdata.RidgeRegressionApplication --driver-memory 5g --executor-memory 5g
--driver-cores 10 --num-executors 10 --verbose
/usr/myfiles/algos/Ridge/target/scala-2.10/RidgeRegression_Demo-assembly-1.0.jar
'{"inputDatabase": "default","inputTable": "concrete","outputDatabase": "default","outputTable":
"test_predictions", "iterations": 50,"stepSize": 0.000001,"regParam": 2.0,"train": 0.7,"test":
0.3,"targetName": "concrete_compresive_strength","algo": "Ridge", "trackId": "流水号 8"}'

3.LASSO

算法接口：
inputDatabase: 输入数据库

inputTable: 输入表
outputDatabase: 输出数据库
outputTable: 输出表
iterations: 迭代次数，0 到 100
stepSize: 训练步长，0 到 1
regParam: 正则相系数，0 到 1
train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class com.vigortech.bigdata.LASSOApplication --driver-memory 5g --executor-memory 5g --driver-cores 10 --num-executors 10 --verbose /usr/myfiles/algos/LASSO/target/scala-2.10/LASSO_Demo-assembly-1.0.jar '{"inputDatabase": "default","inputTable": "concrete","outputDatabase": "default","outputTable": "test_predictions", "iterations": 50,"stepSize": 0.000001,"regParam": 2.0,"train": 0.7,"test": 0.3,"targetName": "concrete_compresive_strength","algo": "LASSO", "trackId": "流水号 9"}'

4.决策树回归

算法接口：
inputDatabase: 输入数据库
inputTable: 输入表
outputDatabase: 输出数据库
outputTable: 输出表
impurity: 分割标准，gini，entropy 或 variance
maxDepth: 树最大深度，0 到 20
maxBins: 最大分箱数目，0 到 20
train: 训练集占比，0 到 1
test: 测试集占比，0 到 1
targetName: 待分类的字段名称
algo: 算法识别名称
trackId: 唯一流水号

Spark 调用命令：
spark-submit --master yarn --deploy-mode client --class com.vigortech.bigdata.DecisionTreeRegression --driver-memory 5g --executor-memory 5g --driver-cores 10 --num-executors 10 --verbose /usr/myfiles/algos/DecisionTreeRegression/target/scala-2.10/DecisionTreeRegression_Demo-assembly-1.0.jar '{"inputDatabase": "default","inputTable": "concrete","outputDatabase": "default","outputTable": "test_predictions", "impurity": "variance","maxDepth": 5,"maxBins":

3,"train": 0.7,"test": 0.3,"targetName": "concrete_compresive_strength","algo": "DTR", "trackId": "流水号 10"}'