链滴

Spark mllib API- tree

作者：Zing

原文链接：https://ld246.com/article/1461564483922

来源网站：链滴

spark在tree这个模块中提供了DecisionTree、RandomForest、GradientBoostedTrees三种算法。属于分类/回归 树模型。

三种算法均可用于回归预测。其中决策树和决策森林可用于二元或多元分类，GBT只能用于二元分类。

随机森林和GBT均属于组合模型，解决模型过拟合问题。

##DecisionTree 决策树

- 类：pyspark.mllib.tree.DecisionTree

决策树算法，训练决策树模型，提供分类和回归。

- 方法：

trainClassifier(data, numClasses, categoricalFeaturesInfo, impurity='gini', maxDepth=5, maxBins=32, minInstancesPerNode=1, minInfoGain=0.0)

训练用于分类的二叉树模型。

- data：训练数据集，格式为LabeledPoint的RDD，LabeledPoint中的Label是整数。
- numClasses：分类的个数。
- categoricalFeaturesInfo：向量中为分类属性的索引表。任务没有出现在该列表中的特征将以连续值处理。{n:k}表示第n个特征，是0-k的分类属性。
- impurity：纯度计算，支持“entropy”和“gini”（默认）
- maxDepth：决策树的最大深度，默认5
- maxBins：每个特征分裂时，最大划分(桶)数量，默认32
- minInstancesPerNode：需要保证节点分割出的左右子节点的最少的样本数量达到这个值，认1
- minInfoGain：当前节点的所有属性分割带来的信息增益都比这个值要小，默认0.0

- 方法：

trainRegressor(data, categoricalFeaturesInfo, impurity='variance', maxDepth=5, maxBins=32, minInstancesPerNode=1, minInfoGain=0.0)

训练用于回归的二叉树模型。

- data：训练数据集，格式为LabeledPoint的RDD，LabeledPoint中的Label是实数。
- categoricalFeaturesInfo：向量中为分类属性的索引表。任务没有出现在该列表中的特征将以连续值处理。{n:k}表示第n个特征，是0-k的分类属性。
- impurity：纯度计算，支持“variance”（默认）
- maxDepth：决策树的最大深度，默认5
- maxBins：每个特征分裂时，最大划分(桶)数量，默认32
- minInstancesPerNode：需要保证节点分割出的左右子节点的最少的样本数量达到这个值，认1
- minInfoGain：当前节点的所有属性分割带来的信息增益都比这个值要小，默认0.0

- 类：pyspark.mllib.tree.DecisionTreeModel(java_model)

- 方法： call(name, *a)

调用java模型

- 方法： depth()

获取决策树的深度

- 方法： load(sc, path)

从指定path加载决策树模型

- 方法： numNodes()

获取决策树的节点数量，包括叶子节点

- 方法： predict(x)

预测一个或多个样本的label值

- 方法： save(sc, path)

将决策树模型持久化到指定path

- 方法： toDebugString()

以string输出整个模型的信息

##RandomForest 随机森林

- 类：pyspark.mllib.tree.RandomForest

- 方法：

trainClassifier(data, numClasses, categoricalFeaturesInfo, numTrees, featureSubsetStrategy='auto', impurity='gini', maxDepth=4, maxBins=32, seed=None)

训练一个用于二元或多元分类的随机森林

- data：训练数据集，格式为LabeledPoint的RDD，LabeledPoint中的Label是整数。
- numClasses：分类的个数。
- categoricalFeaturesInfo：向量中为分类属性的索引表。任务没有出现在该列表中的特征将以连续值处理。{n:k}表示第n个特征，是0-k的分类属性。
- numTrees：随机森林中，树的数量。
- featureSubsetStrategy：特征子集采样策略，支持"auto"（默认），"all","aqrt","log2","onthird"
- impurity：纯度计算，支持“entropy”和“gini” （建议）
- maxDepth：树的最大深度。
- maxBins：每个特征分裂时，最大划分(桶)数量，默认32
- seed：用于引导和选择特征子集的随机种子。

- 方法：

trainRegressor(data, categoricalFeaturesInfo, numTrees, featureSubsetStrategy='auto', impurity='variance', maxDepth=4, maxBins=32, seed=None)

训练一个用于回归预测的随机森林

- data：训练数据集，格式为LabeledPoint的RDD，LabeledPoint中的Label是实数。
    - categoricalFeaturesInfo：向量中为分类属性的索引表。任务没有出现在该列表中的特征将以连续值处理。{n:k}表示第n个特征，是0-k的分类属性。
    - numTrees：随机森林中，树的数量。
    - featureSubsetStrategy：特征子集采样策略，支持"auto"（默认），"all","aqrt","log2","on third"
    - impurity：纯度计算，支持“variance”
    - maxDepth：树的最大深度。
    - maxBins：每个特征分裂时，最大划分(桶)数量，默认32
    - seed：用于引导和选择特征子集的随机种子。

- 类：pyspark.mllib.tree.RandomForestModel(java_model)
    - 方法： call(name, *a)

调用java模型

    - 方法： load(sc, path)

从指定path加载决策树模型

    - 方法： numTrees()

获取随机森林中树的数量

    - 方法： predict(x)

预测一个或多个样本的label值

    - 方法： save(sc, path)

将决策树模型持久化到指定path

    - 方法： toDebugString()

以string输出整个模型的信息

    - 方法： totalNumNodes()

获得森林中所有树的节点总和

##GradientBoostedTrees（GBT） 梯度提升决策树

这是一种模型组合的方法，利用简单模型的组合克服过拟合等问题。常用于推荐系统。

- 类：pyspark.mllib.tree.GradientBoostedTrees
    - 方法：

trainClassifier(data, categoricalFeaturesInfo, loss='logLoss', numIterations=100, learningRate 0.1, maxDepth=3, maxBins=32)

训练一个用于二元分类预测的梯度提升决策树模型。

    - data：训练数据集，格式为LabeledPoint的RDD。label必须为0或1.

● categoricalFeaturesInfo：向量中为分类属性的索引表。任务没有出现在该列表中的特征将会以连值处理。{n:k}表示第n个特征，是0-k的分类属性。

● loss：损失函数，梯度提升计算时需要最小化的该函数。支持“logLoss” (默认)，“leastSquaresError”，“leastAbsoluteError”

● numIterations：提升的迭代次数，默认100.

● learningRate：学习率，取值(0,1]

● maxDepth：树的最大深度

● maxBins：每个特征分裂时，最大划分(桶)数量，默认32

● 方法：

trainRegressor(data, categoricalFeaturesInfo, loss='leastSquaresError', numIterations=100, learningRate=0.1, maxDepth=3, maxBins=32)

训练一个用于回归预测的梯度提升决策树模型。

● data：训练数据集，格式为LabeledPoint的RDD。label为实数。

● categoricalFeaturesInfo：向量中为分类属性的索引表。任务没有出现在该列表中的特征将以连续值处理。{n:k}表示第n个特征，是0-k的分类属性。

● loss：损失函数，梯度提升计算时需要最小化的该函数。支持“logLoss” (默认)，“leastSquaresError”，“leastAbsoluteError”

● numIterations：提升的迭代次数，默认100.

● learningRate：学习率，取值(0,1]

● maxDepth：树的最大深度

● maxBins：每个特征分裂时，最大划分(桶)数量，默认32

● 类： pyspark.mllib.tree.GradientBoostedTreesModel(java_model)

● 方法： call(name, *a)

调用java模型

● 方法： load(sc, path)

从指定path加载决策树模型

● 方法： numTrees()

获取随机森林中树的数量

● 方法： predict(x)

预测一个或多个样本的label值

● 方法： save(sc, path)

将决策树模型持久化到指定path

● 方法： toDebugString()

以string输出整个模型的信息

● 方法： totalNumNodes()

获得森林中所有树的节点总和