# Spark数据挖掘常用API

# 一、理解数据

PCA

PCA训练模型以将向量投影到顶部k主要成分的较低维空间。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.PCA.html#pyspark.ml.feature.PCA>

## KMeans

以k-means ++进行K-means聚类（Bahmani等人的k-means ||算法）。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.clustering.KMeans.html#pyspark.ml.clustering.KMeans>

ChiSquareTest

针对标签上的每个特征进行Pearson的独立性测试。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.stat.ChiSquareTest.html#pyspark.ml.stat.ChiSquareTest>

## Correlation

使用指定的方法为Vectors的输入数据集计算相关矩阵。当前支持的方法：pearson（默认）

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.stat.Correlation.html#pyspark.ml.stat.Correlation>

## Summarizer

用于MLlib向量的向量化统计的工具。此程序包中的方法为DataFrames中包含的Vector提供了各种统计信息。此类使用户可以选择要为给定列提取的统计信息。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.stat.Summarizer.html#pyspark.ml.stat.Summarizer>

# 二、数据准备

## Binarizer

给定阈值对一列连续特征进行二值化。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.Binarizer.html#pyspark.ml.feature.Binarizer>

## ChiSqSelector

卡方特征选择，选择分类特征以用于预测分类标签。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.ChiSqSelector.html#pyspark.ml.feature.ChiSqSelector>

## Normalizer

使用给定的p范数对向量进行归一化，使其具有单位范数。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.Normalizer.html#pyspark.ml.feature.Normalizer>

## StandardScaler

通过去除均值并使用训练集中样本的列摘要统计量缩放到单位方差来对特征进行标准化。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.StandardScaler.html#pyspark.ml.feature.StandardScaler>

## MaxAbsScaler

通过除以每个特征中最大的最大绝对值，分别将每个特征重新缩放为[-1，1]范围。它不会移动/居中数据，因此不会破坏任何稀疏性。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.MaxAbsScaler.html#pyspark.ml.feature.MaxAbsScaler>

## MinMaxScaler

使用列摘要统计信息将每个特征分别线性地缩放到公共范围[min，max]，这也称为最小-最大归一化或重新缩放。特征E的重定比例值计算为：

重新缩放（e\_i）=（e\_i-E\_min）/（E\_max-E\_min）\*（max-min）+ min

对于E\_max == E\_min的情况，Rescaled（e\_i）= 0.5 \*（max + min）

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.MinMaxScaler.html#pyspark.ml.feature.MinMaxScaler>

# 三、建立模型

## 分类

### LogisticRegression

逻辑回归。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.LogisticRegression.html#pyspark.ml.classification.LogisticRegression>

<http://spark.apache.org/docs/latest/ml-classification-regression.html#logistic-regression>

### GBTClassifier

用于分类的梯度提升树（GBT）学习算法。它支持二进制标签（0，1），以及连续和分类功能。不支持多类标签。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.GBTClassifier.html#pyspark.ml.classification.GBTClassifier>

<http://spark.apache.org/docs/latest/ml-classification-regression.html#gradient-boosted-tree-classifier>

### DecisionTreeClassifier

决策树 学习算法进行分类。它支持二进制和多类标签，以及连续和分类功能。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.DecisionTreeClassifier.html#pyspark.ml.classification.DecisionTreeClassifier>

<http://spark.apache.org/docs/latest/ml-classification-regression.html#decision-tree-classifier>

### RandomForestClassifier

用于分类的随机森林学习算法。它支持二进制和多类标签，以及连续和分类功能。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.RandomForestClassifier.html#pyspark.ml.classification.RandomForestClassifier>

<http://spark.apache.org/docs/latest/ml-classification-regression.html#random-forest-classifier>

## 协同过滤-推荐

### ALS

交替最小二乘（ALS）矩阵分解。属于回归算法

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.recommendation.ALS.html#pyspark.ml.recommendation.ALS>

<http://spark.apache.org/docs/latest/ml-collaborative-filtering.html>

# 四、评估模型

## BinaryClassificationEvaluator

二进制分类的评估程序，需要输入列rawPrediction，label和可选的weight列。rawPrediction列的类型可以是double（二进制0/1预测，或标签1的概率），也可以是vector类型（原始预测，分数或标签概率的length-2向量）。

主要输出：areaUnderROC | areaUnderPR

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.evaluation.BinaryClassificationEvaluator.html#pyspark.ml.evaluation.BinaryClassificationEvaluator>

## MulticlassClassificationEvaluator

多类分类的评估程序，需要输入列：预测，标签，权重（可选）和概率概率（仅适用于logLoss）。

亦可用于二分类评估，输出f1 、 accuracy、PrecisionByLabel、truePositiveRateByLabel

主要输出：f1 | accuracy | weightedPrecision | weightedRecall | weightedTruePositiveRate | weightedFalsePositiveRate | weightedFMeasure | truePositiveRateByLabel | falsePositiveRateByLabel | PrecisionByLabel | PrecisionBy

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.evaluation.MulticlassClassificationEvaluator.html#pyspark.ml.evaluation.MulticlassClassificationEvaluator>

## MultilabelClassificationEvaluator

多标签分类的评估程序，需要两个输入列：预测和标签。

主要输出指标：subsetAccuracy | accuracy | hammingLoss | precision | re1 | f1Measure | precisionByLabel | recallByLabel | f1MeasureByLabel | microPrecision | microRecall | microF1Measure

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.evaluation.MultilabelClassificationEvaluator.html#pyspark.ml.evaluation.MultilabelClassificationEvaluator>

## RegressionEvaluator

回归评估器，需要输入列预测，标签和可选的权重列。

rmse-均方根误差（默认）

mse-均方误差

r2-度量标准

mae-平均绝对误差

var-解释方差。

1.平均绝对误差、均方误差和均方根误差 可以准确的计算出预测结果和真实的结果的误差大小，但却无法衡量模型的好坏程度。但是这些指标可以指导我们的模型改进工作，如调参，特征选择等。

2.r2的结果可以很清楚的说明模型的好坏，该值越接近于1，表明模型的效果越好。该值越接近于0，表明模型的效果越差。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.evaluation.RegressionEvaluator.html#pyspark.ml.evaluation.RegressionEvaluator>

# 五、优化模型

## ParamGridBuilder

用于基于网格搜索的模型选择中的参数网格的生成器。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.tuning.ParamGridBuilder.html#pyspark.ml.tuning.ParamGridBuilder>

## CrossValidator

K折交叉验证通过将数据集分成一组不重叠的随机划分的折叠进行模型选择，这些折叠用作单独的训练和测试数据集，例如k = 3折，K折交叉验证将生成3（训练，测试）数据集对，每个对都使用2/3的数据进行训练，并使用1/3的数据进行测试。每一折都被用作测试集一次。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.tuning.CrossValidator.html#pyspark.ml.tuning.CrossValidator>

<http://spark.apache.org/docs/latest/ml-tuning.html>

## TrainValidationSplit

验证超参数调整。将输入数据集随机分为训练集和验证集，并在验证集上使用评估指标来选择最佳模型。与相似CrossValidator，但仅将集合分割一次。

代码参考：

<http://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.tuning.TrainValidationSplit.html#pyspark.ml.tuning.TrainValidationSplit>

<http://spark.apache.org/docs/latest/ml-tuning.html>

# 六、算法模型对应评估器

LogisticRegression模型自带LogisticRegressionTrainingSummary评估，直接使用即可，代码参考：

<http://spark.apache.org/docs/latest/ml-classification-regression.html#logistic-regression>

决策树分类（DecisionTreeClassifier）模型，使用BinaryClassificationEvaluator评估器和MulticlassClassificationEvaluator评估器，代码参考：

<http://spark.apache.org/docs/latest/ml-classification-regression.html#decision-tree-classifier>

随机森林分类（RandomForestClassifier）模型，使用BinaryClassificationEvaluator评估器和MulticlassClassificationEvaluator评估器

代码参考：

<http://spark.apache.org/docs/latest/ml-classification-regression.html#random-forest-classifier>

梯度提升树分类（GBTClassifier）模型只能做二分类，因此，计算ROC面积和PR面积使用BinaryClassificationEvaluator评估器，做准确率、F1等统计可使用MulticlassClassificationEvaluator评估器

代码参考：

<http://spark.apache.org/docs/latest/ml-classification-regression.html#gradient-boosted-tree-classifier>

ALS模型, 使用RegressionEvaluator评估器

代码参考：

<http://spark.apache.org/docs/latest/ml-collaborative-filtering.html>

总结：

分类算法使用分类评估器，主要评估F1分数、精准率、ROC、PR曲线等，值都尽量接近1，则效果越好。

回归算法使用回归评估器，主要评估均方误差和R2，均方误差尽量小和R2尽量接近1，则效果越好。