# 本周工作总结

1. 从以往代码中抽取工具类，实现对特定字段按时间的统计
2. 学习sparkml的官方案例和算法思想

# 工具类实现

1. StatisticUtil.scala

/\*\*  
 \* 输入按时间聚合过的dataframe，输出统计后的dataframe  
 \* 输出：前1个单位，前2个单位，前3个单位，。。。前10个单位  
 \*  
 \* @param spark  
 \* @param df 需要统计的dataframe  
 \* @param cols  需要统计的字段  
 \* @param timeCol 时间字段  
 \* @param idCol id字段  
 \* @param targetTime 要预测的时间点  
 \* @param statisticTime 统计的时间区间  
 \* @return  
 \*/  
 def statisticDf(  
 spark: SparkSession,  
 df: DataFrame,  
 cols: Array[String],  
 timeCol: String,  
 idCol: String,  
 targetTime: Int,  
 statisticTime: Int = 30) = {  
 import spark.implicits.\_  
 val colsSize = cols.length  
 df  
 .rdd  
 .map(x => {  
 val array = new ArrayBuffer[Double]  
 cols.foreach(col => array.append(x.get(x.fieldIndex(col)).toString.toDouble))  
 val key = x.get(x.fieldIndex(idCol)).toString  
 val value = Map(x.get(x.fieldIndex(timeCol)).toString.toInt -> array.toArray)  
 (key, value)  
 })  
 .combineByKey(  
 (v) => v,  
 (acc: Map[Int, Array[Double]], v: Map[Int, Array[Double]]) => acc ++ v,  
 (acc1: Map[Int, Array[Double]], acc2: Map[Int, Array[Double]]) => acc1 ++ acc2  
 )  
 .map(x => {  
 //id  
 val id = x.\_1  
 //某个id的统计值  
 val map = x.\_2  
 //将所有的特征保存在array中  
 var features = ArrayBuffer[Double]()  
 //计算前elem个单位的统计量和  
 var sum = new Array[Double](colsSize)  
 for (elem <- 0 to 30) {  
 val feature = map.get(targetTime - elem).getOrElse(new Array[Double](colsSize))  
 for (i <- Range(0, colsSize)) {  
 sum(i) += feature(i)  
 }  
 features.appendAll(sum)  
 }  
 (id, Vectors.dense(features.toArray))  
 })  
 .toDF("id", "features")  
 }

1. 调用

val features = statisticDf(  
 spark,  
 train, //训练集  
 Array("goods\_num", "goods\_click"), //需要统计的字段  
 timeCol="weekNum2", //时间字段  
 idCol="sku\_id" //商品标识字段,  
 targetTime=33, //需要抽取第33周之前的数据特征  
 statisticTime = 30) //需要抽取离第33周 0周、1周、2周、...、30周内的数量和

1. 使用效果

id: 商品标识符  
  
features: Vector[1周内销售量,1周内点击量,2周内销售量,两周内点击量........]  
  
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+  
|id |features |  
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+  
|SKpnOk5F|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0] |  
|SKTvUYPK|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.0,9.323333333333332,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333,4.0,15.313333333333333] |  
|SKtzguJM|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99,1.0,14.99] |  
|SKSo82Ys|[4.0,8.99,12.0,18.98,18.0,28.97,27.0,38.96,29.0,48.95,35.0,59.940000000000005,37.0,70.93,39.0,81.92,43.0,90.91,64.0,99.28833333333333,73.0,108.27833333333332,92.0,116.99,109.0,125.97999999999999,131.0,135.25,144.0,142.24,150.0,149.23000000000002,152.0,156.22000000000003,161.0,163.21000000000004,178.0,170.20000000000005,201.0,177.19000000000005,208.0,184.18000000000006,225.0,191.17000000000007,251.0,198.16000000000008,284.0,205.1485714285715,304.0,212.1385714285715,328.0,219.12857142857152,345.0,226.11857142857153,362.0,233.10857142857154,374.0,240.09857142857155,391.0,247.3302380952382,412.0,254.5202380952382] |  
|SKX3hl2C|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,6.99,1.0,6.99,1.0,6.99,1.0,6.99,3.0,13.98,3.0,13.98,4.0,24.97,6.0,31.96,8.0,38.95,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005,9.0,45.940000000000005] |  
|SKqIDAnH|[0.0,0.0,2.0,10.99,2.0,10.99,2.0,10.99,2.0,10.99,4.0,21.98,4.0,21.98,7.0,32.97,7.0,32.97,7.0,32.97,7.0,32.97,8.0,43.96,8.0,43.96,9.0,54.95,9.0,54.95,10.0,65.5,10.0,65.5,10.0,65.5,14.0,76.49,18.0,87.47999999999999,21.0,98.46999999999998,24.0,107.45999999999998,26.0,118.44999999999997,28.0,128.43999999999997,31.0,139.42999999999998,33.0,150.42,37.0,161.41,41.0,172.4,46.0,182.89000000000001,54.0,193.88000000000002,56.0,204.76000000000002] |  
|SKBxSM35|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99,3.0,28.99] |  
|SKi31xOm|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,6.99,1.0,6.99] |  
|SKc6QZUM|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0] |  
|SKk1NbpW|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,27.99,1.0,27.99,2.0,55.98,3.0,89.97,4.0,132.96] |  
|SKJuQn9Y|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.0,15.489999999999998,2.0,15.489999999999998,3.0,32.48,7.0,48.13666666666666,7.0,48.13666666666666,8.0,65.12666666666667,9.0,82.11666666666666,10.0,99.10666666666665,12.0,116.09666666666665,12.0,116.09666666666665,12.0,116.09666666666665,14.0,133.08666666666664,14.0,133.08666666666664,21.0,150.07666666666665,23.0,167.06666666666666,24.0,184.05666666666667,25.0,201.04666666666668,26.0,226.0366666666667,26.0,226.0366666666667,26.0,226.0366666666667,26.0,226.0366666666667,26.0,226.0366666666667,26.0,226.0366666666667,26.0,226.0366666666667,26.0,226.0366666666667] |  
|SKOpV7n5|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,10.99,1.0,10.99,1.0,10.99,1.0,10.99,1.0,10.99,1.0,10.99,1.0,10.99,1.0,10.99,1.0,10.99,2.0,24.98,6.0,34.97,11.0,43.626666666666665,13.0,54.61666666666667,15.0,65.60666666666667,16.0,76.59666666666666,17.0,89.58666666666666,18.0,102.57666666666665,23.0,112.89999999999999,25.0,123.38999999999999,28.0,136.38,33.0,144.70333333333332,35.0,153.69333333333333,36.0,162.68333333333334] |  
|SKBVMPQu|[0.0,0.0,0.0,0.0,1.0,15.99,1.0,15.99,1.0,15.99,1.0,15.99,1.0,15.99,1.0,15.99,1.0,15.99,1.0,15.99,1.0,15.99,1.0,15.99,1.0,15.99,2.0,31.98,2.0,31.98,2.0,31.98,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,3.0,47.97,4.0,66.96] |  
|SKNZT9dV|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0] |  
|SKeD3SbF|[6.0,13.99,12.0,27.78,14.0,43.77,15.0,56.760000000000005,26.0,72.97166666666666,35.0,89.16166666666666,39.0,105.90166666666666,45.0,121.71916666666665,50.0,136.95916666666665,53.0,154.94916666666666,56.0,171.60583333333332,58.0,185.59583333333333,59.0,199.58583333333334,62.0,212.57583333333335,62.0,212.57583333333335,63.0,223.56583333333336,63.0,223.56583333333336,64.0,239.55583333333337,69.0,251.54583333333338,79.0,264.20250000000004,83.0,276.52583333333337,83.0,276.52583333333337,83.0,276.52583333333337,84.0,287.5158333333334,84.0,287.5158333333334,85.0,300.5058333333334,85.0,300.5058333333334,85.0,300.5058333333334,85.0,300.5058333333334,85.0,300.5058333333334,85.0,300.5058333333334]|  
|SKzeaAyC|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,8.99,1.0,8.99,1.0,8.99,1.0,8.99] |  
|SKxBk76H|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0] |  
|SKFrVACT|[1.0,18.99,2.0,37.98,3.0,67.97,5.0,84.96,5.0,84.96,5.0,84.96,5.0,84.96,6.0,101.94999999999999,6.0,101.94999999999999,7.0,119.93999999999998,7.0,119.93999999999998,7.0,119.93999999999998,7.0,119.93999999999998,8.0,133.92999999999998,8.0,133.92999999999998,11.0,149.2533333333333,12.0,165.2433333333333,12.0,165.2433333333333,12.0,165.2433333333333,12.0,165.2433333333333,12.0,165.2433333333333,21.0,184.23333333333332,36.0,202.4083333333333,47.0,220.56499999999997,59.0,239.51749999999996,69.0,257.84149999999994,75.0,276.83149999999995,79.0,295.82149999999996,82.0,314.81149999999997,82.0,314.81149999999997,82.0,314.81149999999997] |  
|SKnSNB4s|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,26.99,1.0,26.99,1.0,26.99,1.0,26.99,1.0,26.99,1.0,26.99,1.0,26.99,1.0,26.99,1.0,26.99,1.0,26.99,1.0,26.99] |  
|SK3cRCOx|[0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.0,17.99,3.0,17.99,3.0,17.99,3.0,17.99,3.0,17.99,3.0,17.99,3.0,17.99] |  
+--------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+  
only showing top 20 rows

# spark ml库

AFTSurvivalRegressionExample$ 生存回归模型

ALSExample$ 交替最小二乘法

BinarizerExample$ 二分器

BisectingKMeansExample$

BucketedRandomProjectionLSHExample$

BucketizerExample$ 桶选择器

ChiSqSelectorExample$ 卡方检验

CountVectorizerExample$ 向量字符计数器

DataFrameExample$

DCTExample$ 离散余弦变换 离散余弦变换(DCT for Discrete Cosine Transform)是与傅里叶变换相关的一种变换，它类似于离散傅里叶变换(DFT for Discrete Fourier Transform),但是只使用实数。离散余弦变换相当于一个长度大概是它两倍的离散傅里叶变换，这个离散傅里叶变换是对一个实偶函数进行的（因为一个实偶函数的傅里叶变换仍然是一个实偶函数），在有些变形里面需要将输入或者输出的位置移动半个单位(DCT有8种标准类型，其中4种是常见的)。

DecisionTreeClassificationExample$ 决策树分类

DecisionTreeExample$ 决策树分类或回归

DecisionTreeRegressionExample$ 决策树回归

DeveloperApiExample.scala ElementwiseProductExample$ 向量与转换向量按位相乘

EstimatorTransformerParamExample$ 解释模型中的各个参数

GaussianMixtureExample$ GMM 高斯混合模型

GBTExample$

GeneralizedLinearRegressionExample$ 广义线性回归

GradientBoostedTreeClassifierExample$ 梯度提升分类树

GradientBoostedTreeRegressorExample$ 梯度提升回归树

IndexToStringExample$ String to index，string to original index

InteractionExample$ 向量相乘得向量 (1,2) (1,2) => (1,2,2,4)

IsotonicRegressionExample$ 保序回归

KMeansExample$ Kmeans聚类

LDAExample$ LDA主题聚类模型

LinearRegressionExample$ 线性回归

LinearRegressionWithElasticNetExample$ 使用弹性网络的线性回归

LogisticRegressionExample$ 逻辑回归

LogisticRegressionSummaryExample$ 逻辑回归的summary属性 LogisticRegression中Summary的使用

LogisticRegressionWithElasticNetExample$ 使用弹性网络的逻辑回归

MaxAbsScalerExample$ 列归一化工具，除以每一列的最大值的绝对值

MinHashLSHExample$ minHash(最小哈希)和LSH(局部敏感哈希)

MinMaxScalerExample$ 列归一化模型：(x-MIN)/(MAX-MIN)

ModelSelectionViaCrossValidationExample$ 通过交叉验证法进行模型选择

ModelSelectionViaTrainValidationSplitExample$ 通过训练集验证集训练模型 MulticlassLogisticRegressionWithElasticNetExample$ 多分类逻辑斯特回归

MultilayerPerceptronClassifierExample$ 多层感知机分类器

NaiveBayesExample$ 朴素贝叶斯

NGramExample$ [Hi, I, heard, about, Spark] ->[Hi I, I heard, heard about, about Spark]

NormalizerExample$ 将每一行的规整为1阶范数为1的向量，1阶范数即所有值绝对值之和。

OneHotEncoderExample$ Onehot编码工具模型

OneVsRestExample$ 将二分类器变为多分类器， 假如我有四类要划分（也就是4个Label），他们是A、B、C、D。于是我在抽取训练集的时候，分别抽取A所对应的向量作为正集，B,C,D所对应的向量作为负集；B所对应的向量作为正集，A,C，D所对应的向量作为负集；C所对应的向量作为正集， A,B,D所对应的向量作为负集；D所对应的向量作为正集，A,B,C所对应的向量作为负集，这四个训练集分别进行训练，然后的得到四个训练结果文件，在测试的时候，把对应的测试向量分别利用这四个训练结果文件进行测试，最后每个测试都有一个结果f1(x),f2(x),f3(x),f4(x).于是最终的结果便是这四个值中最大的一个。

PCAExample$ PCA降维算法

PipelineExample$ 使用pipeline包装转换操作

PolynomialExpansionExample$ 多项式扩展: (x,y) -> (x ,y)^3 变为8项式

QuantileDiscretizerExample$ 分位数离散化

RandomForestClassifierExample$ 随机森林分类模型

RandomForestExample$

RandomForestRegressorExample$ 随机森林回归模型

RFormulaExample$ 然后我们只要写一个类似这样的公式clicked ~ country + hour + my\_test，就代表clicked为label，coutry、hour、my\_test是三个特征 PS:自动onehot编码

SQLTransformerExample$ 通过SQLTransformer避免混乱的sql语句 val sqlTrans = new SQLTransformer().setStatement( "SELECT *, (v1 + v2) AS v3, (v1*  v2) AS v4 FROM **THIS**")

sqlTrans.transform(df).show()

StandardScalerExample$ 标准化

StopWordsRemoverExample$ 去除停用词

StringIndexerExample$ String to index ,前面有示例

TfIdfExample$ Tf-idf 示例 TokenizerExample$ 分词器

UnaryTransformerExample$ 自定义transformer extends UnaryTransformer

VectorAssemblerExample$ 包装成对象

VectorIndexerExample$ 对某些特征的值进行重新编号，如某个特征取值为（9,13,16） ，将转换为->(1,2,3)

VectorSlicerExample$ 选取特定的列 （向量切片）

Word2VecExample$ 将词转化为向量