商品关键词提取(2)：TFIDF

由于TFIDF的求值需要根据全体数据求解，此处使用spark中TFIDF的相关模块

先分词，然后分别计算词的TF和IDF值

TF = 当前文档某关键词的个数/当前文档的关键词总个数

- 如果某文档共有100个词(含重复)，其中“python”出现了5次，那么该文档中“python”的TF值为：5/100=0.05

IDF = log(总文档个数/(含有某关键词的文档个数 + 1))，这里+1是为防分母为0

- 如共100篇文档，其中5篇含有“python”，那么“python”的IDF值：math.log(100/6)=2.81

TFIDF = TF * IDF

In [1]:
```
import math
math.log(100/6)
```

Out[1]: 2.8134107167600364

In [1]:

```
import os
# 配置pyspark和spark driver运行时 使用的python解释器
JAVA_HOME = '/root/bigdata/jdk'
PYSPARK_PYTHON = '/miniconda2/envs/py365/bin/python'
# 当存在多个版本时，不指定很可能会导致出错
os.environ['PYSPARK_PYTHON'] = PYSPARK_PYTHON
os.environ['PYSPARK_DRIVER_PYTHON'] = PYSPARK_PYTHON
os.environ['JAVA_HOME'] = JAVA_HOME
# 配置spark信息
from pyspark import SparkConf
from pyspark.sql import SparkSession

SPARK_APP_NAME = "TFIDF"
SPARK_URL = "spark://192.168.58.100:7077"

conf = SparkConf()    # 创建spark config对象
config = (
    ("spark.app.name", SPARK_APP_NAME),    # 设置启动的spark的app名称，没有提供，将随
    ("spark.executor.memory", "2g"),    # 设置该app启动时占用的内存用量，默认1g，指一
    ("spark.master", SPARK_URL),    # spark master的地址
    ("spark.executor.cores", "2"),    # 设置spark executor使用的CPU核心数，指一台虚拟
    ("hive.metastore.uris", "thrift://localhost:9083"),    # 配置hive元数据的访问，否

    # 以下三项配置，可以控制执行器数量
#     ("spark.dynamicAllocation.enabled", True),
#     ("spark.dynamicAllocation.initialExecutors", 1),    # 1个执行器
#     ("spark.shuffle.service.enabled", True)
#   ('spark.sql.pivotMaxValues', '99999'),  # 当需要pivot DF，且值很多时，需要修改，默
)
# 查看更详细配置及说明：https://spark.apache.org/docs/latest/configuration.html

conf.setAll(config)

# 利用config对象，创建spark session
spark = SparkSession.builder.config(conf=conf).enableHiveSupport().getOrCreate()
```

2.3.1合并商品信息中的文本数据为一个长文本

In [70]:

```
# 电子产品
sku_detail = spark.sql('select * from sku_detail')
electonic_product = sku_detail.where('category1_id < 6 and category1_id > 0')
# electonic_product.show()
from pyspark.sql.functions import concat_ws
sentence_df = electonic_product.select('sku_id','category1_id',\
            concat_ws(',',\
                      electonic_product.category1,\
                     electonic_product.category2,\
                     electonic_product.category3,\
                     electonic_product.name,\
                     electonic_product.caption,\
                     electonic_product.price,\
                     electonic_product.specification\
                     ).alias('summary'))
sentence_df.show(2,truncate=False)
```

```
+-------+-------------+-----------------------------------------------------------------
----------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------
------------------------------+
|sku_id|category1_id|summary
|
+-------+-------------+-----------------------------------------------------------------
----------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------
------------------------------+
|148    |3            |数码,数码配件,读卡器,随身厅 WPOS-3 高度集成业务智能终端 森锐手持触
摸屏收银机 打印/身份识别/读写卡,享包邮! 正品保证, 购物无忧!,2999.0
|
|463    |3            |数码,数码配件,读卡器,飞花令 安卓手机读卡器Type-c/USB接口OTG相机车
载读卡器TF/SD/MS多功能合一转接器 迷你车载内存卡读卡器【黑色】,您身边的私人定制: 【联系
客服告知型号】【来图设计看效果图】戳 !其他型号定制点击这里!!!,7.8,颜色:Type-C TF卡
读卡器【金色】,颜色:【安卓手机/电脑-2合一读卡器】蓝色,颜色:安卓+电脑 支持TF-SD卡【黑
色】,颜色:安卓+电脑+type-c 支持TF-SD卡【扁】,颜色:安卓+电脑+type-c 支持TF-SD卡【正】,颜
色:安卓手机TF读卡器-Mirco【金色】,颜色:手机相机SD/TF/MS卡读卡器【4合一金色】,颜色:电脑U
SB3.0+Type-c【2合一黑色】,颜色:迷你车载内存卡读卡器【黑色】,颜色:闪迪 TF转SD卡套|
+-------+-------------+-----------------------------------------------------------------
----------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------
------------------------------+
only showing top 2 rows
```

2.3.2CountVectorizer使用介绍

In [5]:

```
"d i n g".split()
```

Out[5]:

```
['d', 'i', 'n', 'g']
```

In [16]:
```
# Countvectorizer对数据集中的单词进行个数统计
# Input data: Each row is a bag of words with a ID
from pyspark.ml.feature import CountVectorizer
df = spark.createDataFrame([
    (0,'a b c g h'.split(' ')),
    (1,'a b b c a d e f'.split(' '))
],['id','words'])
# df.show()

# fit a CountVectorizerModel from the corpus(语料库).
# vocabSize: 最多保留的单词个数
# minDF: 最小的出现次数，即词频
cv = CountVectorizer(inputCol='words',outputCol='features',vocabSize=100,minDF=1.0)

model = cv.fit(df)
print('数据集中的词：', model.vocabulary)

result = model.transform(df)
result.show(truncate=False)
```

```
数据集中的词： ['b', 'a', 'c', 'f', 'g', 'e', 'd', 'h']
+---+---------------------+-----------------------------------------+
|id |words                |features                                 |
+---+---------------------+-----------------------------------------+
|0  |[a, b, c, g, h]      |(8,[0,1,2,4,7],[1.0,1.0,1.0,1.0,1.0])   |
|1  |[a, b, b, c, a, d, e, f]|(8,[0,1,2,3,5,6],[2.0,2.0,1.0,1.0,1.0,1.0])|
+---+---------------------+-----------------------------------------+
```

2.3.3分词并统计个数

In [27]:

```
import os
import jieba
import jieba.posseg as pseg
import codecs

def words(partitions):

    abspath = "/root/workspace/3.rs_project/project2/notebook"

    stopwords_path = os.path.join(abspath, 'keywordExtract/extract/baidu_stopwords.tx

    # 结巴加载用户词典
    userDict_path = os.path.join(abspath, "keywordExtract/extract/词典/all.txt")
    jieba.load_userdict(userDict_path)

    # 停用词文本
    stopwords_path = os.path.join(abspath, "keywordExtract/extract/baidu_stopwords.tx


    def get_stopwords_list():
        """返回stopwords列表"""
        stopwords_list = [i.strip()
                          for i in codecs.open(stopwords_path).readlines()]
        return stopwords_list

    # 所有的停用词列表
    stopwords_list = get_stopwords_list()

    # 分词
    def cut_sentence(sentence):
        # print(sentence,"*"*100)
        # eg:[pair('今天', 't'), pair('有', 'd'), pair('雾', 'n'), pair('霾', 'g')]
        seg_list = pseg.lcut(sentence)
        seg_list = [i for i in seg_list if i.flag not in stopwords_list]
        filtered_words_list = []
        for seg in seg_list:
            # print(seg)
            if len(seg.word) <= 1:
                continue
            elif seg.flag == "eng":
                if len(seg.word) <= 2:
                    continue
                else:
                    filtered_words_list.append(seg.word)
            elif seg.flag.startswith("n"):
                filtered_words_list.append(seg.word)
            elif seg.flag in ["x", "eng"]:  # 是自定一个词语或者是英文单词
                filtered_words_list.append(seg.word)
        return filtered_words_list

    for row in partitions:
        yield row.sku_id, cut_sentence(row.summary)

doc = sentence_df.rdd.mapPartitions(words)
doc = doc.toDF(["sku_id", "words"])
doc
```

Out[27]: DataFrame[sku_id: bigint, words: array<string>]

In [73]:

```
doc.show(2,truncate=False)
```

```
+-------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|sku_id|words
|
+-------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|148    |[数码, 数码配件, 读卡器, WPOS, 高度, 业务, 智能, 终端, 森锐, 触摸屏, 收银机, 身份, 包邮, 正品, 购物]
|
|463    |[数码, 数码配件, 读卡器, 飞花, 安卓, 手机, 读卡器, Type, USB, OTG, 车载, 读卡器, 转接器, 内存卡, 读卡器, 黑色, 私人, 联系, 客服, 型号, 效果图, 型号, 颜色, Type, 读卡器, 金色, 颜色, 安卓, 手机, 电脑, 读卡器, 蓝色, 颜色, 安卓, 电脑, 黑色, 颜色, 安卓, 电脑, type, 颜色, 安卓, 电脑, type, 颜色, 安卓, 手机, 读卡器, Mirco, 金色, 颜色, 手机, 读卡器, 金色, 颜色, 电脑, USB3, Type, 黑色, 颜色, 内存卡, 读卡器, 黑色, 颜色, 闪迪, 卡套]|
+-------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
only showing top 2 rows
```

In [74]:
```
from pyspark.ml.feature import CountVectorizer
# 6w * 20
# 这里我们将所有出现过的词都统计出来，这里最多会有6w * 20个词
cv = CountVectorizer(inputCol='words',outputCol='rawFeatures',vocabSize=60000*20,minD

cv_model = cv.fit(doc)
cv_result = cv_model.transform(doc)
cv_result.show()
```

```
+------+--------------------+--------------------+
|sku_id|               words|         rawFeatures|
+------+--------------------+--------------------+
|   148|[数码, 数码配件, 读卡器, W...|(42504,[7,10,36,9...|
|   463|[数码, 数码配件, 读卡器, 飞...|(42504,[0,2,3,5,1...|
|   471|[数码, 数码配件, 读卡器, 包...|(42504,[0,1,5,10,...|
|   496|[数码, 数码配件, 读卡器, 品...|(42504,[0,5,10,13...|
|   833|[数码, 数码配件, 读卡器, L...|(42504,[1,10,36,5...|
|  1088|[摄影, 数码相框, 青美, 壁挂...|(42504,[0,1,9,48,...|
|  1238|[数码, 数码配件, 读卡器, d...|(42504,[10,22,36,...|
|  1342|[数码, 数码配件, 读卡器, 绿...|(42504,[0,5,10,36...|
|  1580|[摄影, 数码相框, HNM, 英...|(42504,[1,2,4,9,1...|
|  1591|[数码, 数码配件, 读卡器, k...|(42504,[1,3,10,36...|
|  1645|[摄影, 数码相框, 爱国者, a...|(42504,[1,4,17,20...|
|  1829|[数码, 数码配件, 读卡器, 金...|(42504,[0,10,36,5...|
|  1959|[摄影, 数码相机, 理光, Ri...|(42504,[0,6,9,13,...|
|  2122|[手机, 手机配件, 移动电源, ...|(42504,[0,5,22,24...|
|  2142|[手机, 手机配件, 移动电源, ...|(42504,[0,5,22,23...|
|  2366|[手机, 手机配件, 移动电源, ...|(42504,[0,1,2,5,1...|
|  2659|[手机, 手机配件, 移动电源, ...|(42504,[0,1,2,5,9...|
|  2866|[手机, 手机, 通讯, 对讲机,...|(42504,[0,5,21,43...|
|  3175|[手机, 手机, 通讯, 对讲机,...|(42504,[0,5,21,23...|
|  3749|[手机, 手机, 通讯, 对讲机,...|(42504,[0,5,11,11...|
+------+--------------------+--------------------+
only showing top 20 rows
```

In [96]:
```
cv_result.select('rawFeatures').show(2,truncate=False)
```

```
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|rawFeatures
|
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
|(42504,[7,10,36,97,192,212,350,417,643,2906,4404,7553,13829,14270,24439],[1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0])
|
|(42504,[0,2,3,5,10,16,30,36,52,53,58,64,67,84,91,154,192,229,272,325,410,427,1282,1736,3412,7512,9897],[10.0,4.0,5.0,4.0,1.0,2.0,1.0,9.0,6.0,1.0,2.0,3.0,1.0,1.0,3.0,1.0,1.0,1.0,2.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0])|
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+
only showing top 2 rows
```

In [75]:
```
print(cv_model.vocabulary)
len(cv_model.vocabulary)# 42504
```

...

2.3.4IDF值计算

In [91]:
```
from pyspark.ml.feature import IDF
idf = IDF(inputCol='rawFeatures',outputCol='features')

idfModel = idf.fit(cv_result)
rescaledData = idfModel.transform(cv_result)

rescaledData.select('words','features').show()
```

```
+--------------------+--------------------+
|               words|            features|
+--------------------+--------------------+
|[数码, 数码配件, 读卡器, W...|(42504,[7,10,36,9...|
|[数码, 数码配件, 读卡器, 飞...|(42504,[0,2,3,5,1...|
|[数码, 数码配件, 读卡器, 包...|(42504,[0,1,5,10,...|
|[数码, 数码配件, 读卡器, 品...|(42504,[0,5,10,13...|
|[数码, 数码配件, 读卡器, L...|(42504,[1,10,36,5...|
|[摄影, 数码相框, 青美, 壁挂...|(42504,[0,1,9,48,...|
|[数码, 数码配件, 读卡器, d...|(42504,[10,22,36,...|
|[数码, 数码配件, 读卡器, 绿...|(42504,[0,5,10,36...|
|[摄影, 数码相框, HNM, 英...|(42504,[1,2,4,9,1...|
|[数码, 数码配件, 读卡器, k...|(42504,[1,3,10,36...|
|[摄影, 数码相框, 爱国者, a...|(42504,[1,4,17,20...|
|[数码, 数码配件, 读卡器, 金...|(42504,[0,10,36,5...|
|[摄影, 数码相机, 理光, Ri...|(42504,[0,6,9,13,...|
|[手机, 手机配件, 移动电源, ...|(42504,[0,5,22,24...|
|[手机, 手机配件, 移动电源, ...|(42504,[0,5,22,23...|
|[手机, 手机配件, 移动电源, ...|(42504,[0,1,2,5,1...|
|[手机, 手机配件, 移动电源, ...|(42504,[0,1,2,5,9...|
|[手机, 手机, 通讯, 对讲机,...|(42504,[0,5,21,43...|
|[手机, 手机, 通讯, 对讲机,...|(42504,[0,5,21,23...|
|[手机, 手机, 通讯, 对讲机,...|(42504,[0,5,11,11...|
+--------------------+--------------------+
only showing top 20 rows
```

In [97]:
```
rescaledData.select('rawFeatures','features').show(2,truncate=False)
```

...

In [114]:
```
# 利用idf属性，获取每一个词的idf值，这里每一个值与cv_model.vocabulary中的词一一对应
idfModel.idf.tolist()
# .toArray()
```

Out[114]:
```
[0.2425842894701456,
 1.2595826650830566,
 1.407338832221065,
 0.9269926353711626,
 1.8806329882184594,
 1.3972161773351852,
 2.174763724196774,
 1.938138708698929,
 1.4940382384442819,
 1.9751858790808754,
 1.4125004511528634,
 3.111596728371243,
 3.1048807864058063,
 1.9338751968072267,
 2.8187058732445966,
 2.221522680946395,
 3.312829126931913,
 2.2557200609674517,
 2.997714672905641,
```

In [100]:
```
keywords_list_with_idf = list(zip(cv_model.vocabulary,idfModel.idf.toArray()))
keywords_list_with_idf
```

Out[100]:
```
[('颜色', 0.2425842894701456),
 ('版本', 1.2595826650830566),
 ('黑色', 1.407338832221065),
 ('电脑', 0.9269926353711626),
 ('英寸', 1.8806329882184594),
 ('手机', 1.3972161773351852),
 ('套装', 2.174763724196774),
 ('智能', 1.938138708698929),
 ('办公', 1.4940382384442819),
 ('白色', 1.9751858790808754),
 ('数码', 1.4125004511528634),
 ('套餐', 3.111596728371243),
 ('鼠标', 3.1048807864058063),
 ('京东', 1.9338751968072267),
 ('内存', 2.8187058732445966),
 ('游戏', 2.221522680946395),
 ('型号', 3.312829126931913),
 ('高清', 2.2557200609674517),
 ('键盘', 2.997714672905641),
```

2.3.5TFIDF值的计算

```
# row.rawFeatures是一个向量类型
row.rawFeatures.indices
array([    7,    10,    38,    99,   195,   216,   356,   422,   647,
         2923,  4425,  7473, 13946, 14562, 24286], dtype=int32)
```

```
row.rawFeatures.values
```

```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
row.rawFeatures[7]
```

```
1.0
```

In [148]:

```
from functools import partial

def _tfidf(partition, kw_list):
    for row in partition:
        # 作为分母，大家都是一样的，所以去不去重彼此之间的相对值大小不变，按照开篇介绍
        words_length = len(set(row.words))    # 统计文档中单词总数

        for index in  row.rawFeatures.indices:
            word, idf = kw_list[int(index)]
            # row.rawFeatures[int(index)] 看cell上面的解释
            tf = row.rawFeatures[int(index)]/words_length   # 计算TF值
            tfidf = float(tf)*float(idf)    # 计算该词的TFIDF值
            yield row.sku_id, word, tfidf

# 使用partial为函数预定义要传入的参数
tfidf = partial(_tfidf, kw_list=keywords_list_with_idf)

keyword_tfidf = cv_result.rdd.mapPartitions(tfidf)
keyword_tfidf = keyword_tfidf.toDF(["sku_id","keyword", "tfidf"])
keyword_tfidf.show()
'''
cv_result
+------+--------------------+--------------------+
|sku_id|               words|         rawFeatures|
+------+--------------------+--------------------+
|   148|[数码, 数码配件, 读卡器, W...|(42504,[7,10,36,9...|
'''
```

```
+------+-------+--------------------+
|sku_id|keyword|               tfidf|
+------+-------+--------------------+
|   148|     智能|0.12920924724659527|
|   148|     数码|0.09416669674352422|
|   148|    读卡器|0.22691875231942266|
|   148|     正品| 0.1984394909665287|
|   148|   数码配件| 0.2300917543361638|
|   148|     购物| 0.2408169399138654|
|   148|     包邮| 0.2749585054126381|
|   148|    触摸屏|0.31885875801848024|
|   148|     高度| 0.3896366762502419|
|   148|    收银机|0.45724967270727696|
|   148|     终端|  0.515996300178136|
|   148|     业务|  0.537514526329006|
|   148|     身份| 0.6107553455735467|
|   148|     森锐| 0.6107553455735467|
|   148|   WPOS| 0.6672418695993603|
|   463|     颜色|0.08984603313709096|
|   463|     黑色|0.20849464181052813|
|   463|     电脑|0.17166530284651157|
|   463|     手机|0.20699498923484225|
|   463|     数码|0.05231483152418012|
+------+-------+--------------------+
only showing top 20 rows
```

In [149]:
```
1 keyword_tfidf.orderBy('tfidf',ascending=False).show()
```

```
+------+-------+-----------------+
|sku_id|keyword|            tfidf|
+------+-------+-----------------+
| 65304|     钥匙|16.866725445032152|
| 46934|    K22|15.970125524670596|
| 64669|     研钵|15.621139728147854|
| 23128|     木纹|13.619016619083395|
| 23350|     木纹|13.619016619083395|
| 46559|    XAD|13.354329091785555|
| 10349|     条线|13.255835415734486|
| 53158|     畸变|13.191231766589008|
|  4507|   W88D|13.136324307737405|
| 61486|     单排| 13.02289314127927|
| 65283|    抢答器|12.954087439685217|
| 51841|     纯铜| 12.68401586837053|
| 65127|    钥匙盘|12.663736560299217|
| 46847|    低碳钢| 12.38866944982758|
| 46643|X45X100|12.307564634298311|
| 65127|     铁环|12.149775344114998|
| 46349|    XAD|11.975349457307699|
| 66679|     卡位|11.935691239580011|
| 65127|     钥匙|11.869177165022624|
| 46848|     水道|11.650520419971981|
+------+-------+-----------------+
only showing top 20 rows
```

In [150]:
```
1 rescaledData.first()
```

Out[150]:
```
Row(sku_id=148, words=['数码', '数码配件', '读卡器', 'WPOS', '高度', '业务', '智能',
'终端', '森锐', '触摸屏', '收银机', '身份', '包邮', '正品', '购物'], rawFeatures=Sparse
Vector(42504, {7: 1.0, 10: 1.0, 36: 1.0, 97: 1.0, 192: 1.0, 212: 1.0, 350: 1.0, 417: 1.
0, 643: 1.0, 2906: 1.0, 4404: 1.0, 7553: 1.0, 13829: 1.0, 14270: 1.0, 24439: 1.0}), fea
tures=SparseVector(42504, {7: 1.9381, 10: 1.4125, 36: 3.4038, 97: 2.9766, 192: 3.4514,
212: 3.6123, 350: 4.1244, 417: 4.7829, 643: 5.8446, 2906: 6.8587, 4404: 7.7399, 7553:
8.0627, 13829: 9.1613, 14270: 9.1613, 24439: 10.0086}))
```

In [151]:
```
1 keyword_tfidf.registerTempTable('tempTable')
2 spark.sql('desc tempTable').show()
```

```
+--------+---------+-------+
|col_name|data_type|comment|
+--------+---------+-------+
|  sku_id|   bigint|   null|
| keyword|   string|   null|
|   tfidf|   double|   null|
+--------+---------+-------+
```

2.3.6将TFIDF结果存入hive中

In [152]:
```
sql = """CREATE TABLE IF NOT EXISTS sku_tag_tfidf(
sku_id INT,
tag STRING,
weights DOUBLE
)"""
spark.sql(sql)
```

Out[152]: DataFrame[]

In [153]:
```
spark.sql("INSERT INTO sku_tag_tfidf SELECT * FROM tempTable")
```

Out[153]: DataFrame[]

In [154]:
```
spark.sql("select * from sku_tag_tfidf").show()
```

```
+------+----+------------------+
|sku_id| tag|           weights|
+------+----+------------------+
|   148|  智能|0.12920924724659527|
|   148|  数码|0.09416669674352422|
|   148| 读卡器|0.22691875231942266|
|   148|  正品| 0.1984394909665287|
|   148|数码配件| 0.2300917543361638|
|   148|  购物| 0.2408169399138654|
|   148|  包邮| 0.2749585054126381|
|   148| 触摸屏|0.31885875801848024|
|   148|  高度| 0.3896366762502419|
|   148| 收银机|0.45724967270727696|
|   148|  终端|  0.515996300178136|
|   148|  业务|  0.537514526329006|
|   148|  身份| 0.6107553455735467|
|   148|  森锐| 0.6107553455735467|
|   148|WPOS| 0.6672418695993603|
|   463|  颜色|0.08984603313709096|
|   463|  黑色|0.20849464181052813|
|   463|  电脑|0.17166530284651157|
|   463|  手机|0.20699498923484225|
|   463|  数码|0.05231483152418012|
+------+----+------------------+
only showing top 20 rows
```

In [ ]:
```

```