**TF-IDF介绍**

特征抽取：从原始数据中抽取特征。

TF-IDF: 词频-逆向文件频率。是一种在文本挖掘中广泛使用的特征向量化方法，它可以体现一个文档中词语在语料库中的重要程度。

词语：用t表示

文档：用d表示

语料库：用D表示

词频TF(t,d)：词语t在文档d中出现的次数。

文件频率DF(t,D)：包含词t的文档的个数。

如果我们只使用词频来衡量重要性，很容易过度强调在文档中经常出现，却没有太多实际信息的词语，比如“a”，“the”，以及“of”。如果一个词语经常出现在语料库中，意味着它并不能很好的对文档进行分区。TF-IDF就是在数值化文档信息，衡量词识提供多少信息以及分区文档。

IDF(t,D) = log (D+1/DF(t,D)+1)

此处|D|是语料库中总的文档数，公式中使用log函数，当词出现在所有文档中时候，它的IDF值就为0。加1是为了避免分母为0的情况。

那么IF-IDF度量值表示为：

TF-IDF(t,d,D)=TF(t,d) \* IDF(t,D) = TF(t,d) \* log (D+1/DF(t,D)+1)

在Spark ML库中，TF-IDF被分为两部分：TF和IDF

TF: HashingTF是一个transfomer，在文本处理中，接受词条的集合，然后把这样集合转化为固定长度的特征向量。

TF: Estimator，在一个数据集上应用的fit() 方法，产生一个IDFModel，用该模型接收特征向量，然后计算每一个词在文档中出现的频次。

举个例子：

假如，有一篇文章，词语数量=100，但是词语“机器学习”出现了3次，那么“机器学习”一词在999份文件中出过，并且文件总数为9999999份的话，逆向文件频率就是log(999+1/9999999+1)=4。最后的TF-IDF值就是0.03\*4=0.12

Spark源码中的IDF:

org.apache.spark.ml.feature.IDF

org.apache.spark.ml.feature.HashingTF