dix 平台下运行 xgboost

作者：带个傻逼出去浪

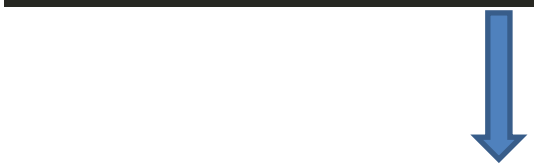
注：所有程序都是在《dix 平台（高校版）简易教程》下进行修改的，由于本人也是第一次使用 dix 平台，若部分操作有问题，欢迎指出！
感谢各位大佬们的鼎力相助！！！！！

1、dix 平台下 xgboost 所支持的数据格式是 Libsvm 数据格式。

- 训练集格式保存成：label+feature 的顺序
  [label feature1:6.666 feature2:6.666 ...]
- 测试集格式保存成：id+feature 的顺序
  [id feature1:6.666 feature2:6.666 ...]

将下面这段代码改成 Libsvm 的格式：

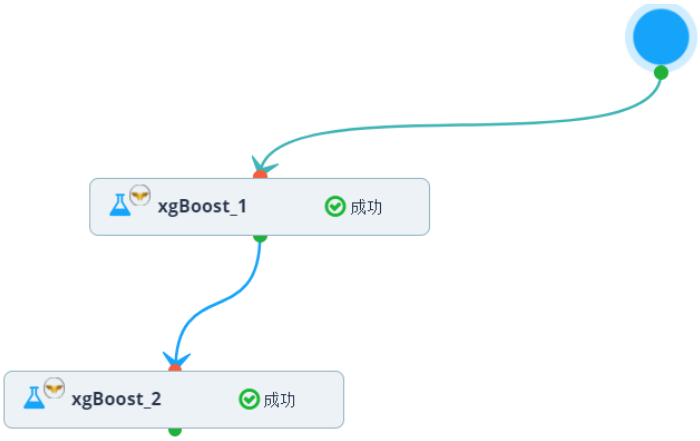
```
def fea_extra_one(a_sample_str):
    # 这个函数里面可以用python原生的方法，也可以用Numpy
    # 但尽量不要用python3和python2使用方法不一致的函数

    # 1.解析原始数据
        # *******
    # 2.特征提取
        # *******
    # 3.是否加入标签
        # 如果是训练集，请加上标签列。
        # 如果是测试集，则不加。
    fea_list = []
        # 这个list里面的每一个元素就是一个特征
        # 若是训练集，将标签放在的fea_list最后
    fea_str_list = [str(item) for item in fea_list]
    fea_str = ' '.join(fea_str_list)
    return fea_str
```



```
feat_str_list= ['{}:{}'.format(i + 1, j) for i, j in enumerate(feat_list)]
feat_str_list = [str(label)] + feat_str_list
feat_str_list = ' '.join(feat_str_list)
```

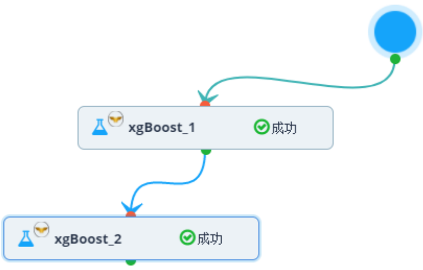
2、dix 平台下 xgboost 模型搭建

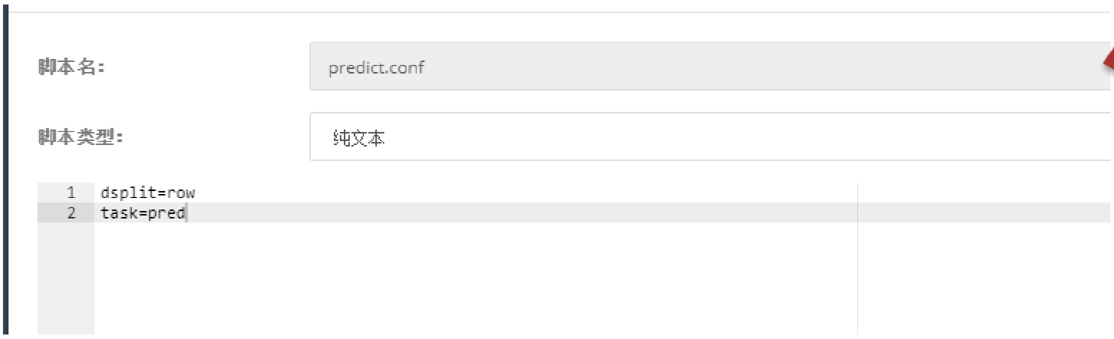
- 选择两个 xgboost 模型，如下：



- 进行 xgboost_1 模型配置，如下：

- 进行 xgboost_2 模型配置，如下：





注意：!!

3、接下来，就是结果输出

注意：xgboost　输出的是这种数据（id\t prob），在《dix 平台（高校版）简易教程》上面数据 split 一下就好了。

```
(1999903, u'1999904\t0.99978 ')
0.99978
(1999904, u'1999905\t0.986494 ')
0.986494
(1999905, u'1999906\t0.999789 ')
0.999789
(1999906, u'1999907\t0.999463 ')
0.999463
(1999907, u'1999908\t0.99984 ')
0.99984
(1999908, u'1999909\t0.999862 ')
0.999862
(1999909, u'1999910\t0.999751 ')
0.999751
(1999910, u'1999911\t0.999871 ')
0.999871
(1999911, u'1999912\t0.99906 ')
0.99906
(1999912, u'1999913\t0.97777 ')
0.97777
(1999913, u'1999914\t0.999226 ')
0.999226
(1999914, u'1999915\t0.999833 ')
0.999833
(1999915, u'1999916\t0.999858 ')
```

代码如下：

```
import numpy as np
import sys
from pyspark import SparkContext
output_file=sys.argv[1]
input_file=sys.argv[2]
sc=SparkContext(appName="test")
rdd=sc.textFile(input_file)
result=rdd.collect()
result_filal=[]
for i,item in enumerate(result):
    print(i,item)
    item=float(item.split('\t')[1])
    print(item)
    if item<0.5:
        result_filal.append(i+1)
        #print(result_filal)
print('machine nums:',len(result_filal))
result_rdd=sc.parallelize(result_filal)
result_rdd.saveAsTextFile(output_file)
```