

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：

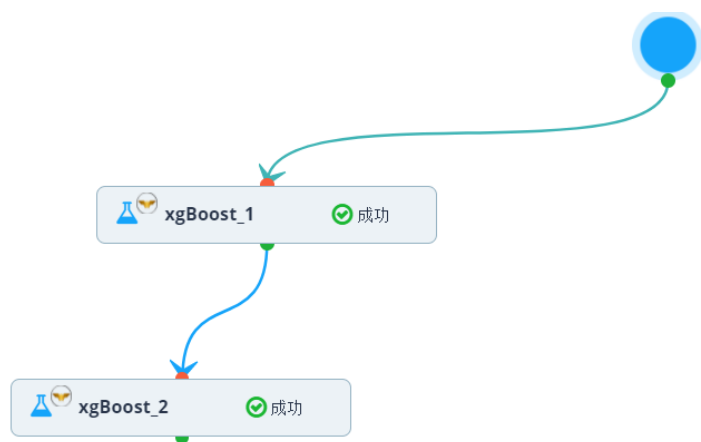
```
def fea_extra_one(a_sample_str):  
    # 这个函数里面可以用python原生的方法，也可以用Numpy  
    # 但尽量不要用python3和python2使用方法不一致的函数  
  
    # 1.解析原始数据  
    # *****  
    # 2.特征提取  
    # *****  
    # 3.是否加入标签  
    # 如果是训练集，请加上标签列。  
    # 如果是测试集，则不加。  
    fea_list = []  
    # 这个list里面的每一个元素就是一个特征  
    # 若是训练集，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：

参数配置

组件参数	描述说明
* xgboost_configurati	train.conf 配置文件
训练输入 ?	训练集路径 \${output_prefix}/train_fe at_libsvm.txt
测试数据集输入 ?	验证集路径（我设置就是训练集） \${output_prefix}/train_fe at_libsvm.txt
训练模型输出 ?	模型保存路径 \${output_prefix}/model

脚本名：

train.conf

脚本类型：

纯文本

```
1 booster=gbtree
2 objective=binary:logistic
3 learning_rate=
4 gamma=0.0
5 min_child_weight=1
6 max_depth=
7 subsample=
8 colsample_bytree=
9 eval_metric='logloss'
10 eval_metric='auc'
11 scale_pos_weight=1
12 num_round=
13 save_period=0
14 dsplit=row
```

- 进行 xgboost_2 模型配置，如下：



组件参数	
* xgboost_configuration	<div>predict.conf</div> <div>配置文件</div> <div>测试集的配置文件</div>
训练输入	
测试数据集输入	
训练模型输出	
预测输入	<div>测试集的数据路径 (200w)</div> <div>\$(output_prefix)/hcq/test_feat_libsvm.txt</div>
模型输入	<div>之前训练模型的路径</div> <div>\$(output_prefix)/hcq/model</div>
预测结果输出	<div>预测的结果保存路径</div> <div>\$(output_prefix)/hcq/predict_xgb.txt</div>

脚本名:

predict.conf

脚本类型:

纯文本

1 dsplit=row

2 task=pred

注意：

* jobname	<input type="text"/>	作业名称
* nworker	<input type="text" value="2"/> 这里千万别设置很大，可能是玄学	worker数
* vcores	<input type="text" value="1"/>	线程数
* memory_mb	<input type="text" value="5000"/>	内存大小
* other_yarn_args	<div><input type="text"/></div>	其他YARN参数

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。

```
. 777100
(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 ')
n 999858
```

代码如下：

```
1 import numpy as np
2 import sys
3 from pyspark import SparkContext
4 output_file=sys.argv[1]
5 input_file=sys.argv[2]
6 sc=SparkContext(appName="test")
7 rdd=sc.textFile(input_file)
8 result=rdd.collect()
9 result_filal=[]
10 for i,item in enumerate(result):
11     print(i,item)
12     item=float(item.split('\t')[1])
13     print(item)
14     if item<0.5:
15         result_filal.append(i+1)
16         #print(result_filal)
17 print('machine nums:',len(result_filal))
18 result_rdd=sc.parallelize(result_filal)
19 result_rdd.saveAsTextFile(output_file)
```