PAKDD2020

AUTH	TIME	VERSION
IvanaXu	2020-03-18 23:00:10	V1.0

一、解决方案

1、特征提取

根据磁盘smart_n_normalized、smart_nraw等SMART数据，仅选用归一化后，**考虑运行时间**，样本数据每1000000分块读取，保留非空非唯一值的变量。

> project/feature/feature.py

2、数据采集

考虑数据量，月磁盘SMART数据以及上月止坏磁盘好坏作为**观察**，取下月始30天磁盘好坏作为**表现**，定义是否出现磁盘问题为**好坏目标**，按月汇总SUM、MAX、MEDIAN、MIN值。

并完成以下预处理：

- 增加主键

```
data["mk"] = [
  f"{i}{j}{k}" for i,j,k in zip(data["manufacturer"], data["model"],
data["serial_number"])
]
```

- 日期格式

```
data["dt"].apply(lambda x: datetime.datetime.strptime(str(x), "%Y%m%d"))
```

> project/feature/collect.py

3、构建样本

根据201707-201805数据作为训练样本，保存至**data_t01.h5**，201806数据作为验证样本，保存至**data_v01.h5**。

并根据A、B榜要求，如设置**i_model = "A2"**筛选model=2数据。

> project/feature/build_data.py

4、建模过程

- 训练样本、测试样本，占比1:1

```
d01t = pd.read_hdf(f"../user_data/tmp_data/data_t01.h5", key="data")
d01t = pd.DataFrame(d01t)

data_x = d01t[var_l]
data_y = d01t["bad"]
x_train, x_test, y_train, y_test = train_test_split(data_x, data_y, test_size=0.5)
```

- 验证样本

  考虑预测数据好坏占比，为保证验证完整性，切割为小样本验证、大样本验证。

```
d01v = pd.read_hdf(f"../user_data/tmp_data/d_out_201806.h5", key="data")
# S 0.3
d01v1 = pd.DataFrame(d01v).sample(frac=0.3)
# B 0.7
d01v2 = pd.DataFrame(d01v).sample(frac=0.7)
```

- 构建模型

  通过xgboost构建模型。

- 参数调整

```
params = {
    'booster': 'gbtree',
    'objective': 'binary:logistic',
    'eval_metric': 'auc',
    'max_depth': 4,
    'lambda': 10,
    'subsample': 0.75,
    'colsample_bytree': 0.75,
    'min_child_weight': 2,
    'eta': 0.012,
    'seed': 0,
    'nthread': 8,
    'silent': 1
}
```

- 最佳cutoff

  以最大F1值为目标，根据metrics.f1_score函数找出训练、测试的最佳cutoff。

  并观察验证样本F1值，调整至佳。

> project/model/basic_model.py

5、结果输出

如上，对预测数据进行相同预处理，并带出**"manufacturer", "model", "serial_number", "dt"**等主要字段。根据保存模型m001.model以及最佳cutoff值Nr，将预测结果处理为如下示例：

```
A,1,disk_1,2018-08-15
A,1,disk_123,2018-08-16
A,1,disk_1,2018-08-17
A,2,disk_456,2018-08-14
```

若出现mk主键重复，根据时间顺序排序最早以去重，并通过csv导出至predictions.csv。

> project/code/output.py

二、运行说明

project/code 路径下：

```
# /bin/bash
sh main.sh
```

三、代码规范

提交代码文件夹结构：

```
project
  ├── README.md
  ├── code
  |   ├── main.sh
  |   ├── output.py
  |   └── requirements.txt
  ├── data
  |   ├── round1_testA
  |   |   └── disk_sample_smart_log_test_a.csv
  |   ├── round1_testB
  |   |   └── disk_sample_smart_log_test_b.csv
  |   └── round1_train
  ├── feature
  |   ├── build_data.py
  |   ├── collect.py
  |   └── feature.py
  ├── model
  |   └── basic_model.py
  ├── prediction_result
  |   ├── predictions.csv
  |   └── predictions.csv.zip
  └── user_data
      ├── model_data
      |   └── m001.model
      └── tmp_data
          ├── d_out_201707.h5
          ├── d_out_201708.h5
          ├── d_out_201709.h5
```

```
    ├── d_out_201710.h5
    ├── d_out_201711.h5
    ├── d_out_201712.h5
    ├── d_out_201801.h5
    ├── d_out_201802.h5
    ├── d_out_201803.h5
    ├── d_out_201804.h5
    ├── d_out_201805.h5
    ├── d_out_201806.h5
    ├── d_smart_dl.h5
    ├── data_t01.h5
    ├── data_v01.h5
    └── value.h5
```