

线上分析工具

线上分析工具

整体

功能

数据源

算法库

1. 标准差滤波算法

2. 中值绘图

Notebook·工作区

我的报告

本文档为线上分析工具设计思路，部分细节和技术方案待定

整体

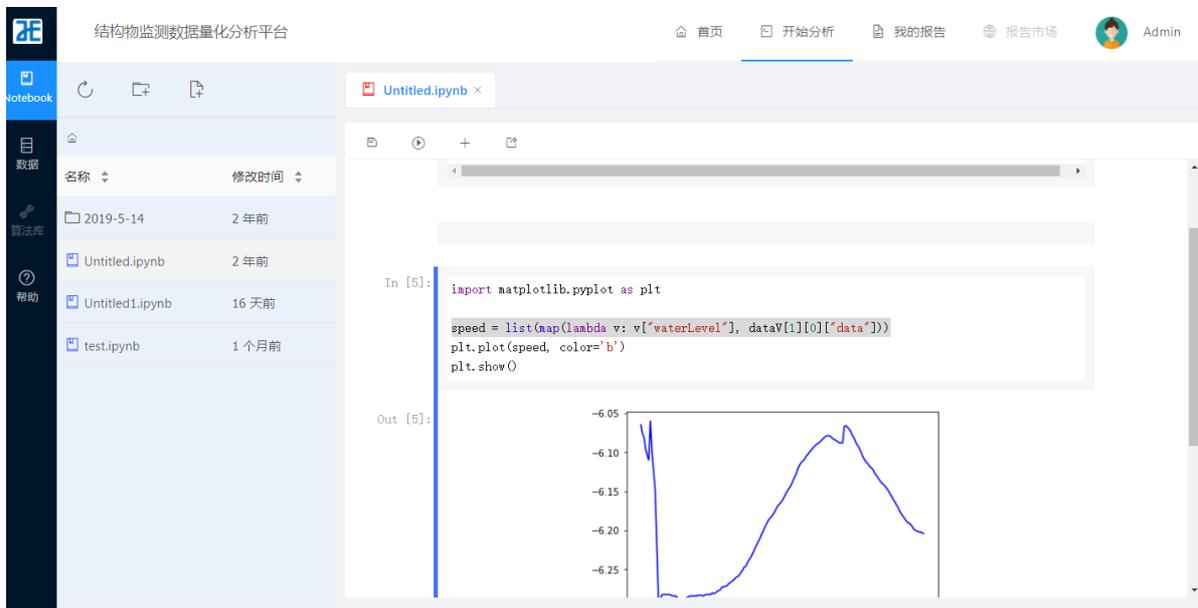
基于现有数据分析平台 `analyser` 开发可视化交互分析平台，做进一步功能扩展

组成:

web应用 <http://10.8.30.22/FS-Anxinyun/trunk/codes/web/analysis>; 包括web/api/编排器功能组合

jupyter服务 <http://10.8.30.22/FS-Anxinyun/trunk/codes/services/anxinyun-notebook>

现有功能界面:



The screenshot displays the user interface of the online analysis tool. At the top, there is a navigation bar with the logo 'AE' and the title '结构物监测数据量化分析平台'. The main interface is divided into three sections: a left sidebar, a central code editor, and a right output area. The sidebar contains a 'Notebook' section with a list of files: '2019-5-14' (modified 2 years ago), 'Untitled.ipynb' (2 years ago), 'Untitled1.ipynb' (16 days ago), and 'test.ipynb' (1 month ago). The central code editor shows a Jupyter Notebook cell with the following Python code:

```
In [5]: import matplotlib.pyplot as plt

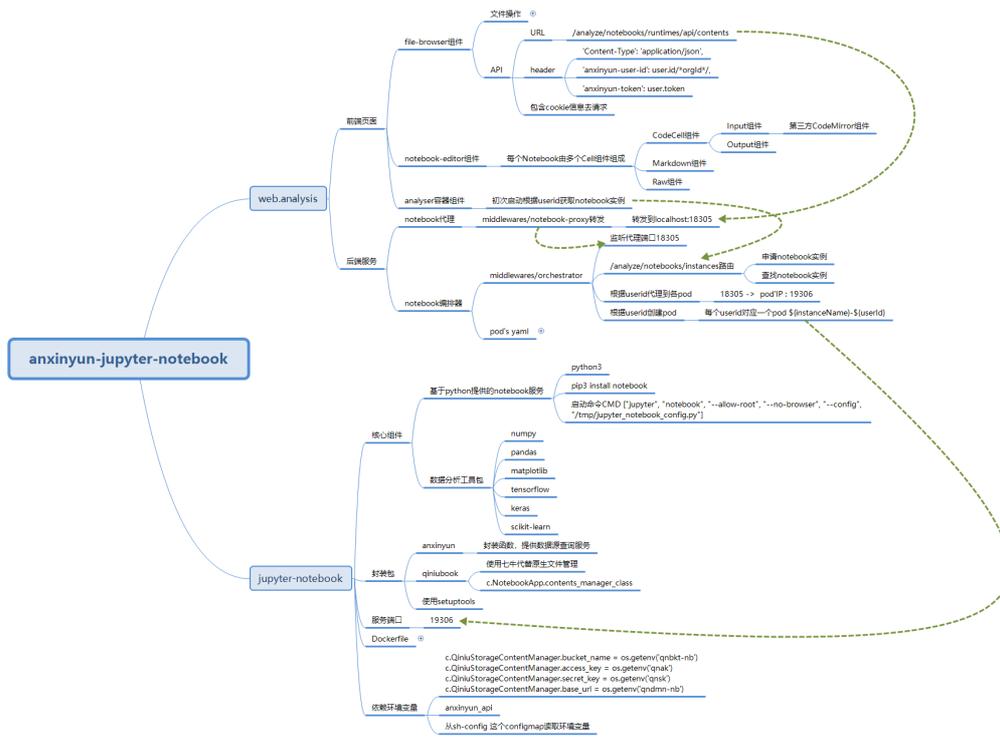
speed = list(map(lambda v: v["waterLevel*"], dataV[1][0]["data*"]))
plt.plot(speed, color='b')
plt.show()
```

The output area displays a line plot with a blue line. The y-axis ranges from -6.05 to -6.25, and the x-axis is unlabeled. The plot shows a sharp initial drop followed by a smooth, bell-shaped curve peaking around -6.10.

功能模块:

- 数据源: 列出当前用户所能访问的所有结构物数据源
- Notebook工作区: 文件目录、notebook工作区
- 算法库: 算法功能, 包括算法输入输出、算法代码块以及对应UI组成
- 我的报告: 工作区内执行的结果保存, 提供下载功能

现有架构图：by Tyr.Liu



功能

数据源

数据源分类

数据源			
安心云3.0			
安心云2.0			
上传数据CSV			

上传数据：

用户想要分析本地数据（CSV），需要首先将文件上传到安心云平台。上传过程除了需要提交文件外，还需提交相关的元数据信息。包括：

项目名称、数据类型(设备/主题)、数据文件格式（总结现有几种格式后提供选项）；

后端服务对文件进行再处理后存储到HDFS：

提取到设备名称（测点名称）、将数据格式转换成标准格式（平台3.0数据文件格式）；

注意：上传过程不创建3.0结构物，单独维护上传数据的项目-设备-监测类型-测点信息。

以上三种数据源均需实现以下接口：

1. 获取所有结构物
2. 查询结构物下的所有测点

/structures/{struct}/stations?factorId={factor}

3. 获取结构物所有设备

/structures/{struct}/devices?type={deviceType}

4. 获取设备数据

5. 获取测点数据

目前支持从API接口获取和HDFS数据获取。后续考虑支持HIVE接口查询

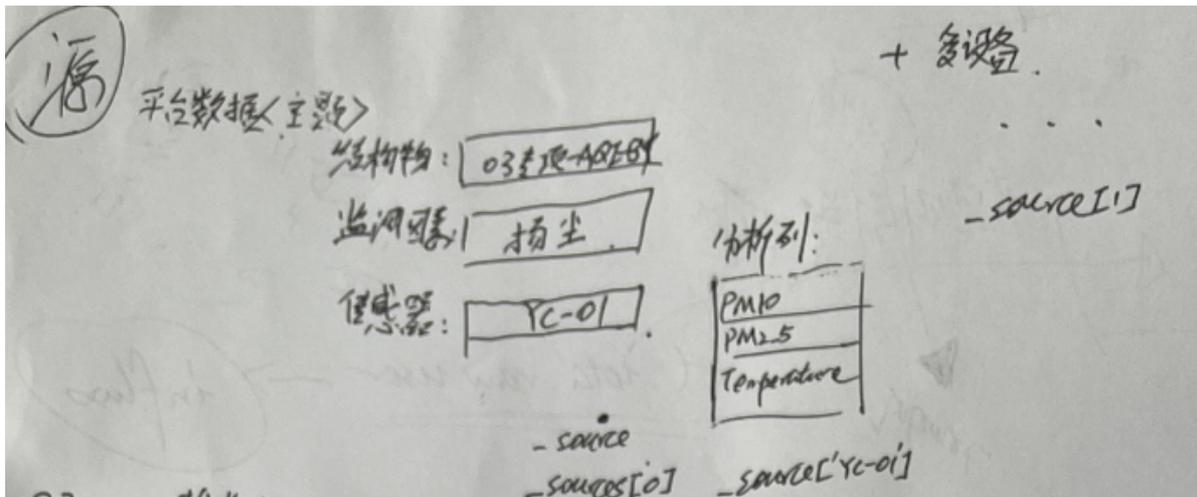
数据源从HDFS文档系统中的CSV文件获取，通过Pandas库读取CSV后输出DataFrame数据源。统一后的数据源格式如下：

数据列	描述
source	传感器名称或测点名称
time	数据时间
data1	数据列1
data2	数据列2
...	...

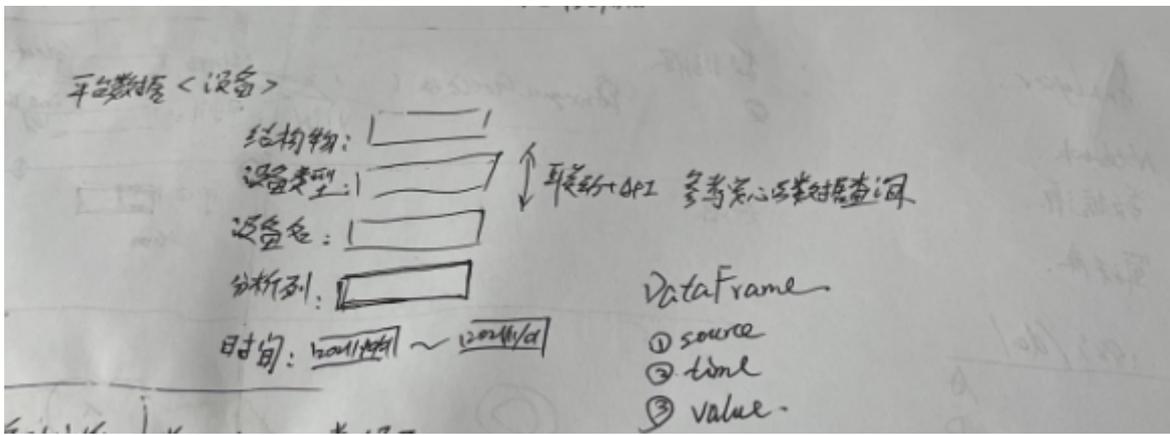
UI设计：

数据源对应的UI可参考目前平台实现，即可以通过结构物、监测因素过滤选择测点；通过结构物、设备类型选择设备。

测点选择设计草图：



设备选择设计草图：



- 可以选择多个点位
- 分析列不选默认输出所有数据列

数据源选定后，执行python单元格内容时，对应的参数将被赋值

```
_sources=DataApi.get_data(stations=${stationIds}, begin=${begin}, end=${end})
```

算法库

算法库包含具体算法的脚本代码和参数输入设计，这些算法通常包括过滤、计算、输出(图表)；参数输入设计采用动态UI和关系映射绑定，形成可视化的算法模块。

通过整理 [线下分析工具](#)，整理目前算法的类别，主要有：

分析方法	描述	Input	Output	举例
过滤方法	过滤数据	DataFrame	DataFrame	标准差倍数过滤
单列分析方法	单列数据的统计分析和绘图*	DataFrame - DataSeries	图表	热力图
多列分析方法	多列数据的统计分析和绘图	DataFrame	图表	实时趋势图
多点分析	数据的关联对比	DataFrame	图表	关联分析
综合分析	结构物下测点数据的统计	DataFrame	图表	数据连续性统计

举例来说明期望的算法定义和具体应用方式：

1. 标准差滤波算法

以下示例包含伪代码；

numpy、pandas、matplotlib为默认引入，分别记为别名 np/pd/plt

算法定义，代码 `std_filter`

```
# data -- 数据源: DataFrame
# field -- 数据列
# times -- 参数: 标准差倍数
# out -- 输出: 过滤后的DataFrame
def std_filter(data, field, times):
    bz=np.std(data[field])*times
    return data.loc[data[field]>bz]
```

UI: 该算法需要指定对取哪列数据做判断, 以及输入标准差倍数参数, 所以动态UI中应该包括如下两个输入框:

数据列: [--]

标准差倍数: [--]

使用:

工作区中, 插入单元格时选择该算法, 此时应该在Notebook下生成如下工作单元

```
from std_filter import *
for src in _sources.indexs
    _sources[i]=std_filter(src, ${data_filed}, ${times})
```

2. 中值绘图

算法定义

代码 `avg_draw`

```
# data -- 数据源: DataFrame
def avg_draw(data):
    fig, axs = plt.subplots(2, 1)
    axs[0].plot(data['time'], data['data1'], data['time'], data['data2'])
    axs[0].set_xlim(0, 2)
    axs[0].set_xlabel('time')
    axs[0].set_ylabel('data1')
    axs[0].grid(True)
    plt.show()
```

UI:

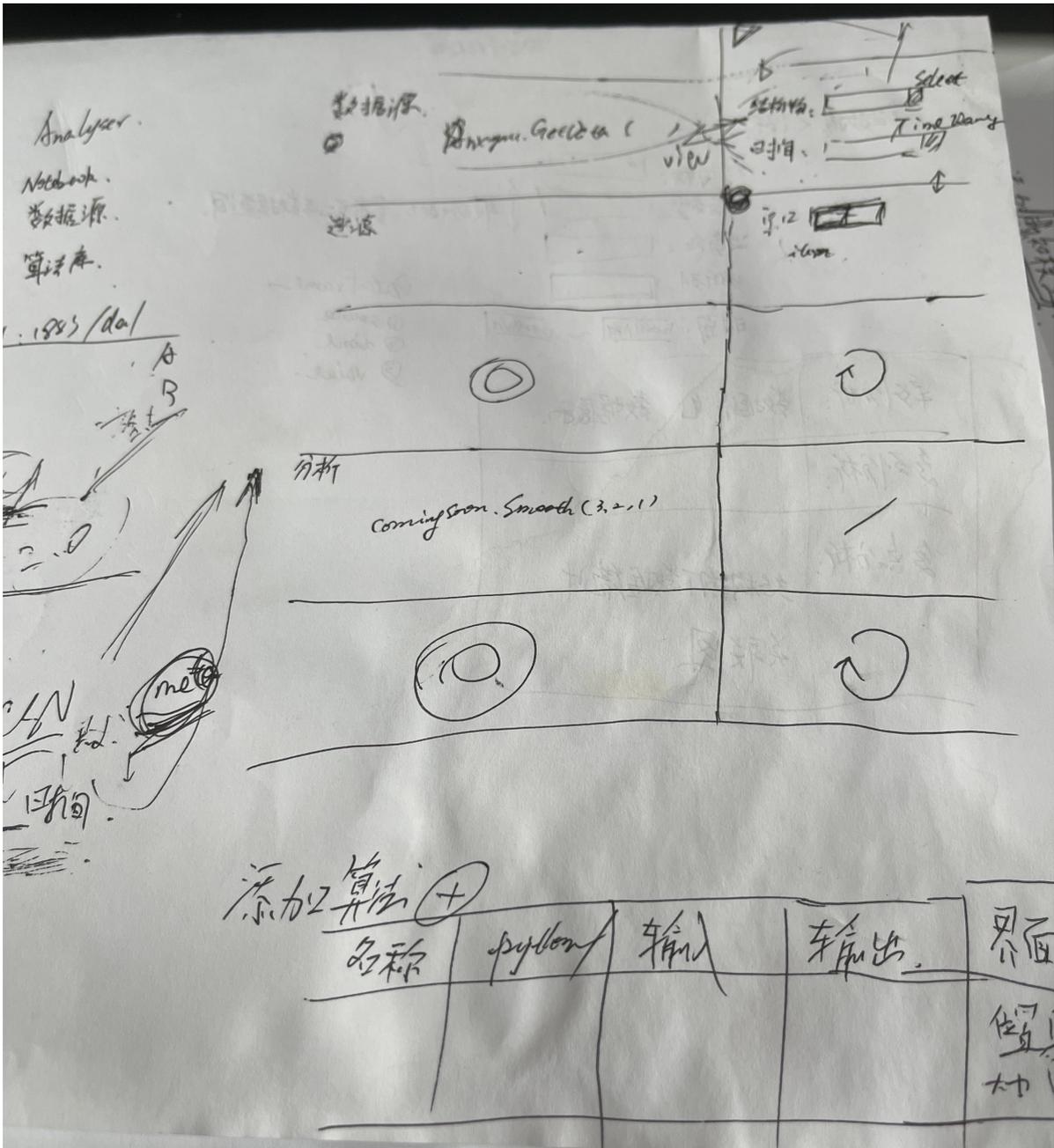
无参数

使用:

工作区中选择算法, 此时应该在Notebook下生成如下工作单元

```
from avg_draw import *
for src in _sources.indexs
    _sources[i]=avg_draw(_sources[i])
```

[Out]



工作区左侧为文件目录，目录下的文件可以进行新建、拷贝、重命名和删除等操作。每个用户可以创建自己的工作区，工作区的文件序列化到用户的七牛目录。

工作区右侧的内容，可以通过配置或不同用户权限，仅显示UI操作界面。

我的报告

工作区内执行完成后，可以保存当前页（html格式）到七牛，在“我的报告”菜单中，可以查看下载所有保存的报告