

About The Workflow

1.基本目标

构建一个前端页面，用于展示一个具有不同属性节点的有向无环图。

2.数据类型

以JSON格式给出，形式为

属性名	类型	说明
id	string	该节点的id，为一长字符串且唯一。
name	string	该节点的名字。不唯一。
type	int	该节点的类型。0：数据实体，1：操作
category	int	该节点所属的分类。0：原始数据实体，1：数据集合实体，2：数据加工，3：结构化数据实体，4：外部算法实体，5：模型实体。不同分类的节点应该以不同颜色显示。
pre	list	该节点的前驱节点id
sub	list	该节点的后继节点id
stored_datas	JSON	仅数据实体具有的属性。以JSON的格式显示其存储的数据。其中，key是字段，value是字段值。
mapping_datas	JSON	仅操作具有的属性。以JSON的格式展示数据操作的前后映射。其中，key是该数据加工加工前的字段，value是数据加工加工后的字段。
infos	JSON	额外信息。

一个示范：

```
{
  "id": "6b2c3d01754127a1",
  "name": "数据集合节点1",
  "type": 0,
  "category": 1,
  "pre": [],
  "sub": [],
  "stored_datas": {
    "time": "2022-07-15",
    "data": 3
  },
  "infos": {
    "version": "0-0-1-2022-03-01"
  }
}
```

它所表示的节点展示出来应如下图所示：



在本例子中，绿色代表数据集合节点

3.功能

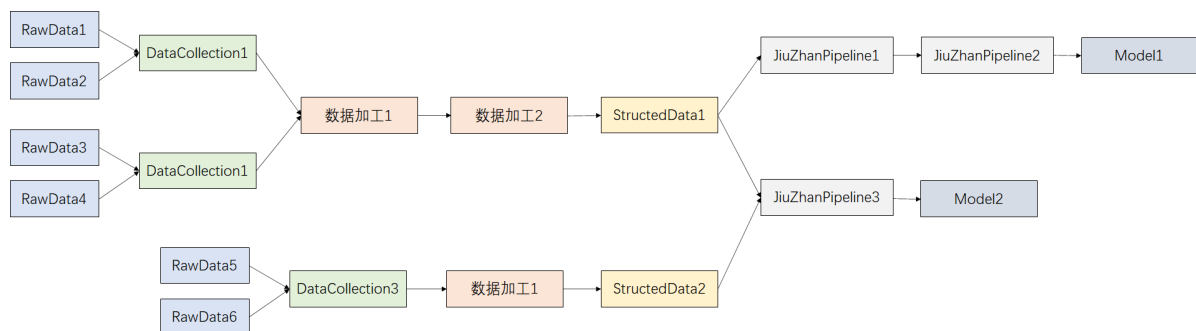
I 显示 workflow

一个 workflow 是按照以下顺序构成的：

原始数据 -> 数据集合 -> 数据加工-> 结构化数据 -> 外部算法 -> 模型

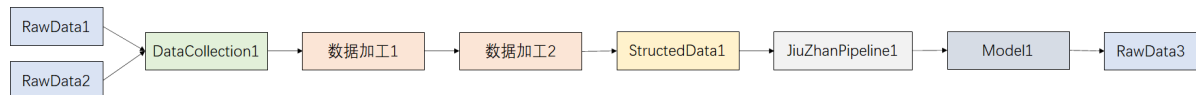
下图展示了一个示例。事实上，该图包含两个存在重叠部分的 workflow。

难点：前驱和后继关系已经给出，因此没必要考虑前驱和后继节点的安排。只需要考虑节点的位置即可。要求：尽量美观。

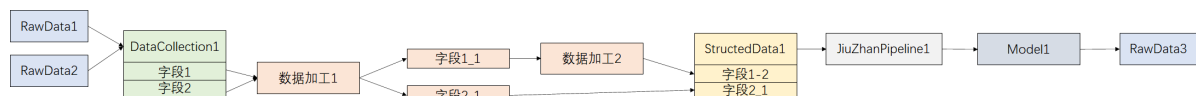


II 切换显示粒度

系统可以从不同粒度来展示 workflow。下图展示了**节点级别**的 workflow。**请注意！** workflow 之间可以符合嵌套的。下图中 RawData3 就是 Model1 的后继节点。



下图为上图在**字段级别**下的展示。**请注意！** 数据加工部分的字段结点后端不会给出，需要前端自行解析其前后关系（有待商榷，可以改成后端给）



III 显示详细信息

点击一个节点时，显示该节点的全部信息。请参考 Netron。

IV 过滤

NODE PROPERTIES

type

BatchNormalization

?

name

BatchNormalization_2

ATTRIBUTES

epsilon

0.00000999999747378752

+

momentum

0.8999999761581421

+

INPUTS

X

name: 685

scale

name: conv2.bn1.weight

+

B

name: conv2.bn1.bias

+

mean

name: conv2.bn1.running_mean

+

var

name: conv2.bn1.running_var

+

OUTPUTS

Y

name: 686