# 使用

## 安装依赖包

pip install -r requirements.txt

## 运行

双击目录下 app.bat 可以看到目前版本

## 说明文档

在 /doc/doc.md 中有目前版本的说明文档

# 普通梯度提升机 GBM

## 配置模型

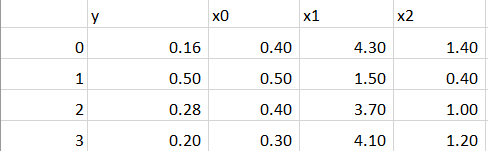
**"model\_parameters"**: {  
 **"n\_estimators"**: (**"弱学习器的个数"**, 100),  
 **"max\_depth"**: (**"树的最大深度"**, 4),  
 **"learning\_rate"**: (**"参数淘汰率"**, 0.01),  
 **"loss"**: (**"损失函数"**,**"ls"** )  
}

## 训练

### 训练输入

格式：excel表格

描述：第0列为样本序号，第1列为因变量y，其它列为自变量



样例数据：data/gbm/GBM\_TRAIN\_DATA.xlsx

### 训练输出

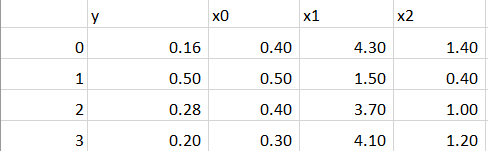
returnDic = {  
 **"系数"**: （模型系数）  
}

## 测试

### 测试输入

格式：excel表格

描述：第0列为样本序号，第1列为因变量y，其它列为自变量



样例数据：data/gbm/GBM\_TEST\_DATA.xlsx

### 测试输出

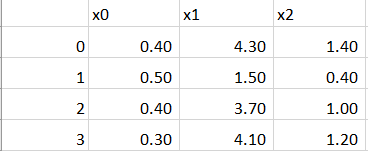
returnDic = {  
 **"mean\_squared\_error"**: **均方误差**,  
 **"mean\_absolute\_error"**: **平均绝对值误差**}

## 预测

### 预测输入

格式：excel表格

描述：第0列为样本序号，其它列为自变量



样例数据：data/gbm/GBM\_PREDICT\_DATA.xlsx

### 预测输出

returnDic = {  
 **"预测结果"**: **预测值**}

# 基于梯度提升机的I-BRT

## 配置模型

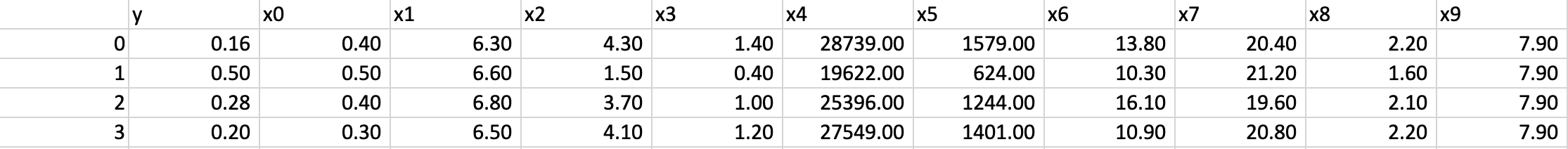
**"model\_parameters"**: {  
 **"n\_iter"**: (**"迭代次数"**, 10),  
 **"\_gamma"**: (**"正则项系数"**, 0.0),  
 **"\_lamda"**: (**"正则项系数"**, 1.0),  
 **"max\_depth"**: (**"基本树最大深度"**, 2)  
}

## 训练

### 训练输入

格式：excel表格

描述：第0列为样本序号，第1列为因变量y，其它列为自变量



样例数据：data/ibrt/IBRT\_TRAIN\_DATA.xlsx

### 训练输出

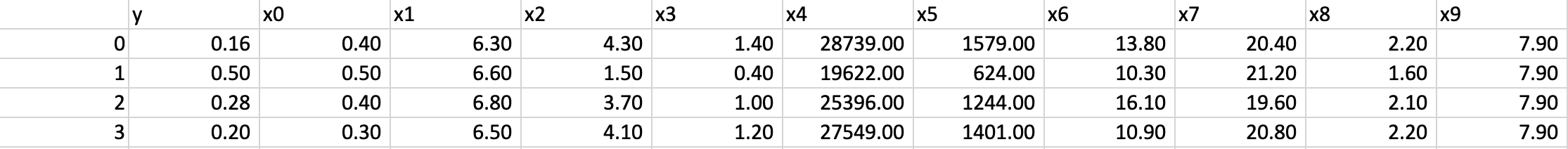
returnDic = {  
}

## 测试

### 测试输入

格式：excel表格

描述：第0列为样本序号，第1列为自变量y，其它列为因变量



样例数据：

data/ibrt/IBRT\_TEST\_DATA.xlsx

### 测试输出

returnDic = {

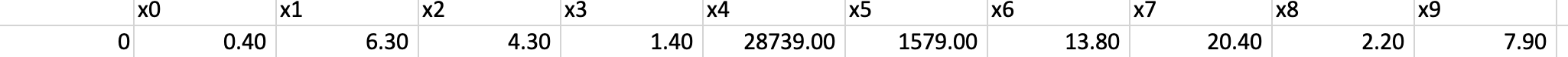
**"mean\_absolute\_error"**: **平均绝对值误差**}

## 预测

### 预测输入

格式：excel表格

描述：第0列为样本序号，其它列为因变量



样例数据：

data/ibrt/IBRT\_PREDICT\_DATA.xlsx

### 预测输出

returnDic = {  
 **"预测结果"**: **预测值**}

# 改进的自适应Lasso SALP

## 配置模型

**"model\_parameters"**: {  
 **"alpha"**: (**"惩罚项系数"**, 0.1),  
 **"max\_iter"**: (**"最大迭代次数"**, 500),  
 **"ex\_var\_per"**: (**"参数淘汰率"**, 0.25),  
 **"k"**: (**"重构样本数量"**, 10)  
}

## 训练

### 训练输入

格式：excel表格

描述：第0列为样本序号，第1列为自变量y，其它列为因变量



样例数据：data/salp/SALP\_TRAIN\_DATA.xlsx

### 训练输出

returnDic = {  
 **"排除特征"**:（排除的特征）   
 **"系数"**: （模型系数）  
}

## 测试

### 测试输入

格式：excel表格

描述：第0列为样本序号，第1列为自变量y，其它列为因变量



样例数据：

data/salp/SALP\_TEST\_DATA.xlsx

### 测试输出

returnDic = {  
 **"mean\_squared\_error"**: **均方误差**,  
 **"mean\_absolute\_error"**: **平均绝对值误差**}

## 预测

### 预测输入

格式：excel表格

描述：第0列为样本序号，其它列为因变量



样例数据：

data/salp/SALP\_PREDICT\_DATA.xlsx

### 预测输出

returnDic = {  
 **"预测结果"**: **预测值**}

# 面向纵向数据的RE-BET(MERT与其类似)

## 配置模型

**"parameter": {**

**"n": ("观测对象种类数量",1),**

**"epoch": ("迭代轮数",200),**

**"M": ("迪利克雷过程离散程度",10),**

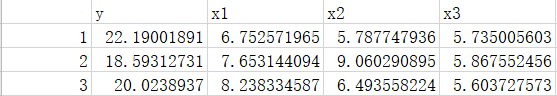
**"k": ("指定第k个变量作为随机效应变量",1),**  
}

## 训练

### 训练输入

格式：excel表格

描述：第0列为样本序号，第1列为自变量y，其它列为因变量



样例数据：data/rebet/data\_train.xlsx

### 训练输出

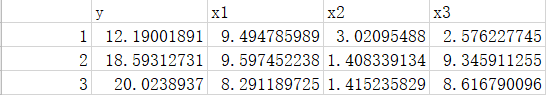
returnDic = {  
 **"随机效应协方差矩阵": str(D),**  
 **"误差方差": str(σ2)**  
}

## 测试

### 测试输入

格式：excel表格

描述：第0列为样本序号，第1列为自变量y，其它列为因变量



样例数据：

data/rebet/data\_test.xlsx

### 测试输出

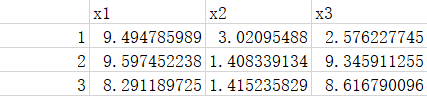
returnDic = {  
 **"mse"**: **均方误差**}

## 预测

### 预测输入

格式：excel表格

描述：第0列为样本序号，其它列为因变量



样例数据：

data/rebet/data\_predict.xlsx

### 预测输出

returnDic = {  
 **"预测结果"**: **预测值**}

# 多层线性模型 HLM

## 配置模型

无

## 训练

### 训练输入

格式：excel表格

输入分为**两个**数据文件，一个为某站点土壤的固定因素数据 ，假设为 soil\_data.xlsx；另一个为自变量和因变量数据，设为 erosion\_data.xlsx；

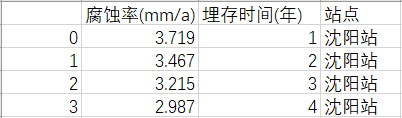
soil\_data,xlsx描述：第0列为样本序号，第1列为站点名称，之后的列为该站点土壤的固定因素，至少要包含 土壤 PH 值、全氮含量(%)、碳酸根离子浓度(%)、硫酸根离子浓度(%)、镁离子浓度(%)、钾离子浓度(%)、钠离子浓度(%)、土壤电阻(Ω) 8 种主要影响因素。



样例数据：data/gbm/SOIL\_DATA.xlsx

erosion\_data.xlsx描述：数据共**四列，**第 0 列为序号，第 1 列为腐蚀率 (浮点型)，第 2 列为埋存时间 (单位年，最好为整数)，第 3 列为站点名称 (string)。

注：soil\_data.xlsx中包含的站点才能进行测试和预测。



样例数据：data/gbm/GBM\_TRAIN\_DATA.xlsx

### 训练输出

returnDic = {   
 **"系数"**: （模型系数）

**"第一层随机误差的方差"**: （第一层随机误差的方差）

**"第二层随机误差的方差"**: （第二层随机误差的方差）  
}

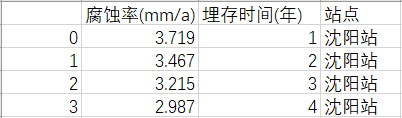
## 测试

### 测试输入

格式：excel表格

描述：数据共**四列，**第 0 列为序号，第 1 列为腐蚀率 (浮点型)，第 2 列为埋存时间 (单位年，最好为整数)，第 3 列为站点名称 (string)。

注：soil\_data.xlsx中包含的站点才能进行测试和预测。



样例数据：data/gbm/GBM\_TEST\_DATA.xlsx

### 测试输出

returnDic = {  
 **"mean\_squared\_error"**: **均方误差**,  
 **"mean\_absolute\_error"**: **平均绝对值误差**}

## 预测

### 预测输入

格式：excel表格

描述：数据共三列**，**第 0 列为序号，第 1 列为埋存时间 (单位年，最好为整数)，第 2 列为站点名称 (string)。

注：soil\_data.xlsx中包含的站点才能进行测试和预测。



样例数据：data/gbm/GBM\_PREDICT\_DATA.xlsx

### 预测输出

returnDic = {  
 **"预测结果"**: **预测值**}

# 基于ANFIS改进的数据填补方法 REANFIS

## 配置模型

**"model\_parameters"**: {  
}

## 训练

### 训练输入

格式：excel表格

描述：第0列为样本序号，第1列为因变量y，其它列为自变量



样例数据：data/rfanfis/RFANFIS\_TRAIN\_DATA.xlsx

### 训练输出

returnDic = {  
}

## 测试

### 测试输入

格式：excel表格

描述：第0列为样本序号，第1列为自变量y，其它列为因变量



样例数据：

data/rfanfis/RFANFIS\_TEST\_DATA.xlsx

### 测试输出

returnDic = {

**"mean\_square\_error"**: **均方误差**

**"root\_mean\_square\_error"**: **均方根误差**

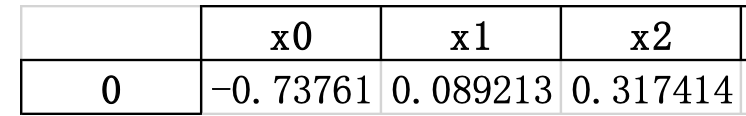
**"perc\_loss"**: **百分化平均绝对误差**}

## 预测

### 预测输入

格式：excel表格

描述：第0列为样本序号，其它列为因变量



样例数据：

data/rfanfis/RFANFIS\_PREDICT\_DATA.xlsx

### 预测输出

returnDic = {  
 **"预测结果"**: **预测值**}

# 遗传算法 GA

## 配置模型

**"parameter": {**

**"c": ("指定求函数的最大值或最小值",0),**

**"n": ("函数所含变量个数",1),**

**"ranges": ("各个变量的取值范围",[[-3, 3]]),**

**"precisions": ("精度",24),**

**"N\_GENERATIONS": ("迭代轮数",50),**

**"POP\_SIZE": ("种群大小",200),**

**"MUTATION\_RATE": ("变异概率",0.005),**

**"CROSSOVER\_RATE": ("交叉概率",0.8),**

**}**

## 预测

### 预测输入

格式：txt文件

描述：函数表达式，需用x[0],x[1]...表示自变量



样例数据：

data/ga/F.txt

### 预测输出

returnDic = {  
  **"最值": str(y),**

**"变量取值": str(x)**  
}

# 层次分析法 AHP

## 训练

无训练

## 测试

无测试

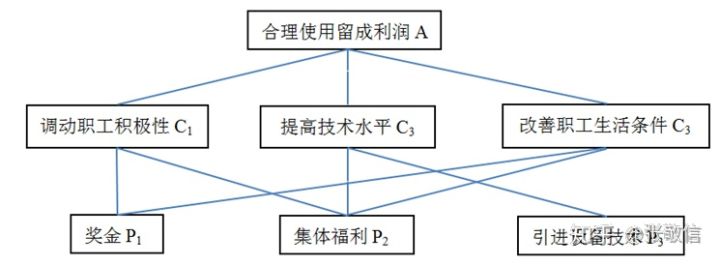
## 预测

### 输入

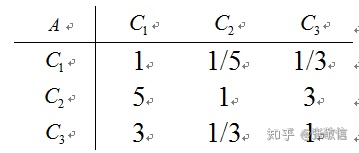
格式：文本

说明：层次分析法输入为关系矩阵，

如下例的输入在路径：data/ahp/ahpInput.txt

某工厂有一笔企业留成利润，需要决定如何分配使用。已经决定有三种用途：奖金、集体福利措施、引进技术设备。考察准则也有三个：是否能调动职工的积极性、是否有利于提高技术水平、考虑改善职工生活条件。建立如下图所示层次模型：

经过工厂决策人员讨论，得到如下判断矩阵：

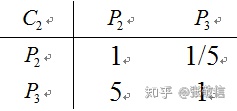
（1）第2层对第1层。

（2）第3层对第2层。

1）第3层对第2层第1个元素C1：

受C1支配的只有两个元素P1和P2，判断矩阵C{2}{1,1}为

2）第3层对第2层第2个元素C2：

受C2支配的只有两个元素P2和P3，判断矩阵C{2}{1,2}为

3）第3层对第2层第3个元素C3：

受C3支配的只有两个元素P1和P2，判断矩阵C{2}{1,3}为

其它样例路径：

data/ahp/ahpInput2.txt

### 输出

输出为各个决策的权重：

**"""合理使用留成利润[1]  
├── 提高技术水平[0.638]  
│ ├── 引进设备技术[0.833]  
│ └── 集体福利[0.167]  
├── 改善职工生活条件[0.258]  
│ ├── 奖金[0.333]  
│ └── 集体福利[0.666]  
└── 调动职工积极性[0.104]  
 ├── 奖金[0.249]  
 └── 集体福利[0.751]"""**