# 背景 – 纵向数据分割场景和树模型

关于纵向树模型：

为纵向数据集特化的树模型算法

树模型可以提供解释性，并且训练高校

SF提供两种树模型：

可证安全算法SS-XGB，使用秘密分享，不泄露信息。网络条件好时性能高

纵向联邦算法SGB，明密文混合，网络要求不高，计算量大。算力高就性能高

# 使用 – sf纵向树模型

## 使用步骤

经典三步：

准备，通信、安全设备、训练数据

训练，设置参数并且执行

评估，计算相应指标

## SGB使用步骤

教程：[决策树模型 | SecretFlow v1.6.1b0 | 隐语 SecretFlow](https://www.secretflow.org.cn/zh-CN/docs/secretflow/v1.6.1b0/user_guide/mpc_ml/decision_tree)

准备：设置HEU和数据

sk\_keeper是私钥拥有方（标签持有方）

evaluator是计算方（其他参与方）

## 算法使用的区别

准备阶段：

SS-XGB需要重点关注SPU、MPC

SGB需要关注HEU和联邦

训练阶段：

SS-XGB参数比较少，全是训练参数

SGB参数多，18个训练参数，6个安全和联邦参数

训练结果

SS-XGB为密态模型，每一方只有一部分数据，预测结果为MPC密文。目前不支持模型保存

SGB预测结果为单方面明文，标签持有方可以单独评估

# 原理和实现 – 理论到工程

## 算法发展

SS-XGB是第一个实现可证安全的树模型算法。

经典算法到MPC算法：

1、确定要保护的数据：标签和特征（数据源）、数据处理中间特征的顺序、数据分布

2、准备安全原语（MPC算子）：秘密分享和初始化、加减乘除、sigmoid和argmax

3、改造数据结构和算法

SGB是已经开源的性能最好的纵向联邦树模型算法

SGB算法(SecureBoost)

经典算法到FL算法：

1、确定要保护的数据：标签（数据源）、数据处理中间特征的顺序

2、准备安全原语：半同态加法

3、改造数据结构和算法

## 实践：修改源代码实现预期功能

三个方面的开发：组件封装、算法修改、模型导出

### 组件封装

修改算法后，需要修改组件的相关定义

组件封装三步：

1、定义组件主体

名字、描述、版本号

2、定义参数和输入输出

需要引入新参数、删掉旧参数

3、定义调用方法

一般不会改变

主要是使用方法的步骤，建立安全设备 – 准备训练数据 – 准备训练参数

建立spu：

if ctx.spu\_configs is None or len(ctx.spu\_configs)==0:

raise CompEvalError(“spu config not found”)

if len(ctx.spu\_configs)>1:

raise CompEvalError(“support only 1 spu”)

spu\_config=next(iter(ctx.spu\_configs.values()))

spu\_device=SPU(spu\_config{“cluster\_def”}, spu\_config{“link\_desc”})

y=load\_table(ctx, train\_dataset, load\_labels=True)

x=load\_table(ctx, train\_dataset, load\_features=True)

### 算法修改

阅读算法代码，掌握流程

修改算法和对应模块

修改单元测试

### 模型导出

阅读代码确定是否有导出功能

按照接口存放、读取模型

自定义模型序列化方法

# sf优势

1、设备抽象：sgb不会感知heu协议，ss-xgb不会感知mpc协议

2、分层架构：原语，可以优化重点算子

3、分层架构：算法，python实现，易读易维护

4、开源共建

# 作业

参考homework 2：

练习ss-xgb, 检查auc

注意：如果发现could not convert string to float这种报错，是没有对string类型正确编码导致的，需要执行这种东西进行转换：

data['month'] **=** encoder.fit\_transform(data['month'])