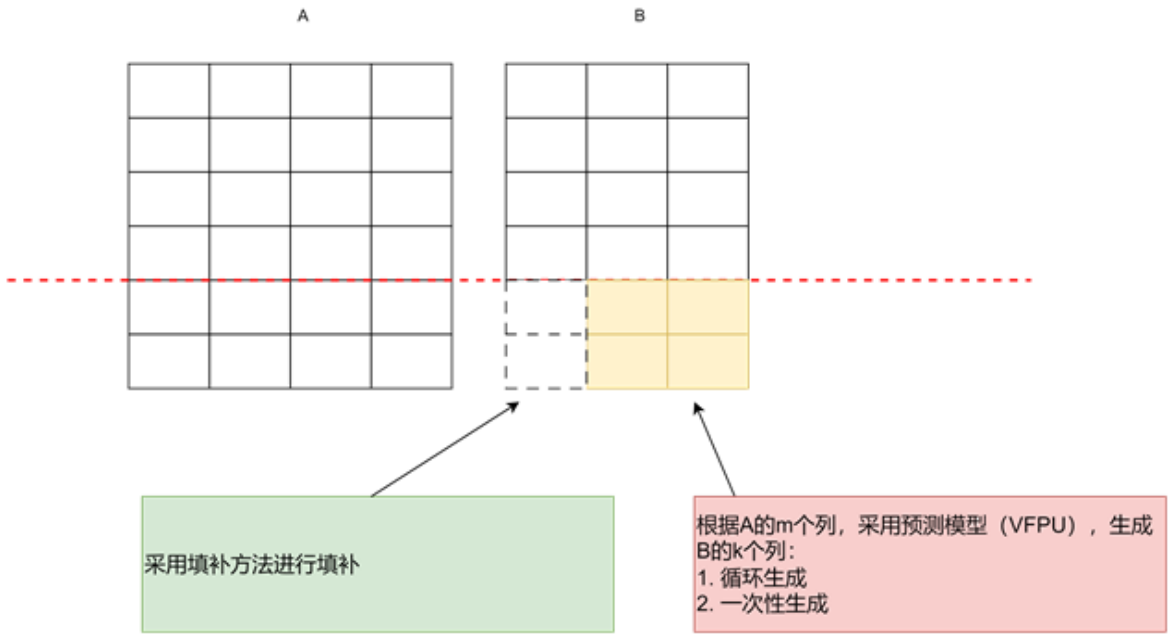



1、A这边有B的1 到 n，是逐个用半监督生成，还是先用gan网络生成，一部分，再用半监督生成一部分

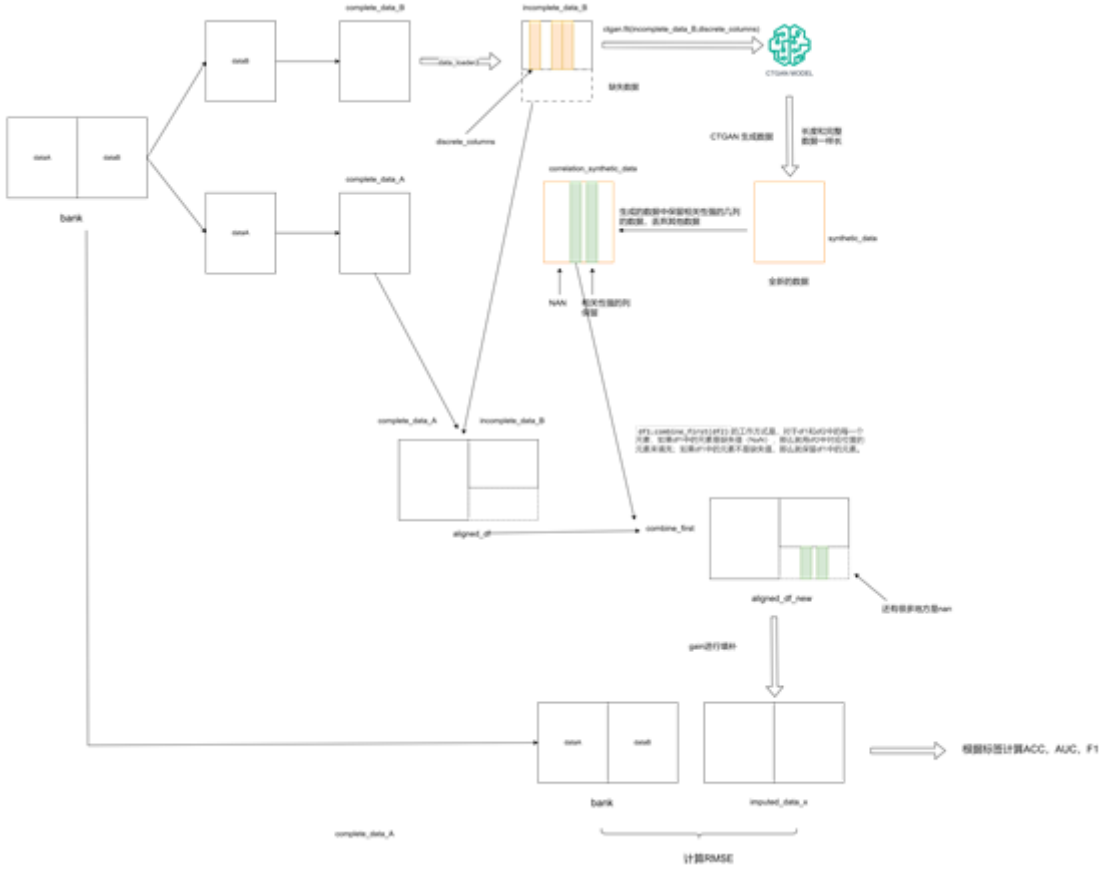
2、做一个基础验证：

- 全部用gan网路生成（何杭轩在做）
- （1）一列一列全部用半监督（循环），（2）或者一次性做半监督同时的生成（结合gan网络，一次性生成，可以作为谭和朱的研究）
- 先用gan网络生成一部分，再用半监督生成一部分。

放到第四章涉及到实验验证，全部验证完，每个，哪一种方案是最可能的，最有效的，用一个数据集，

现在不用完整的实验，先做个初步验证，这几种方案需要比较一下，考虑哪一种相对说可行一点的。基于之前实验的基础

# 实验笔记




实验分三类文件：

1. 相关性实验：correlation_experimen_0.2，correlation_experimen_0.5
2. 随机实验：random_experiment_0.2，random_experiment_0.5
3. 对比试验：comparative_experiment_0.2，comparative_experiment_0.5

0.2 0.5 表示缺失率

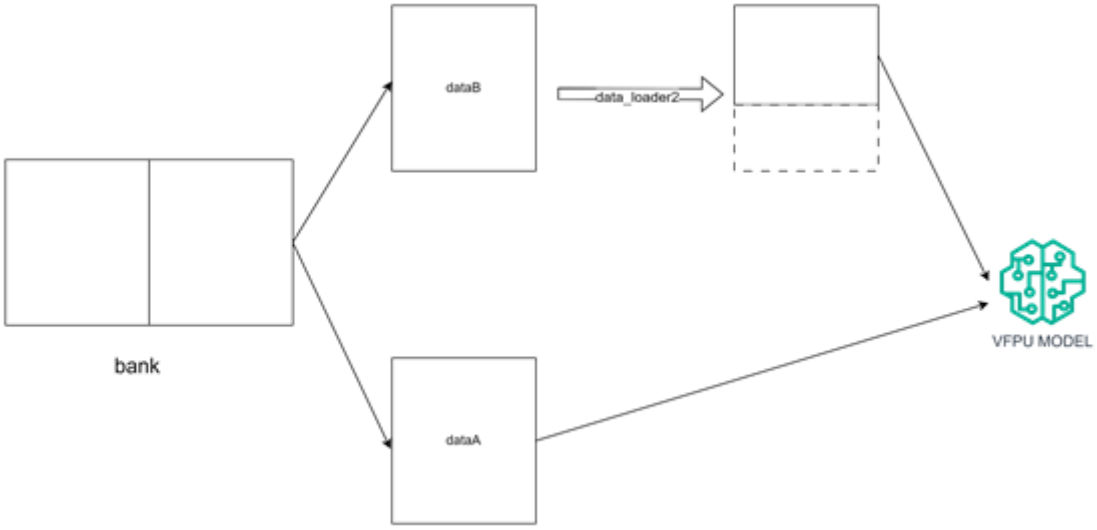
按照缺失率的大小将部分行丢弃
返回原始数据，缺失后数据，掩码矩阵

整理代码

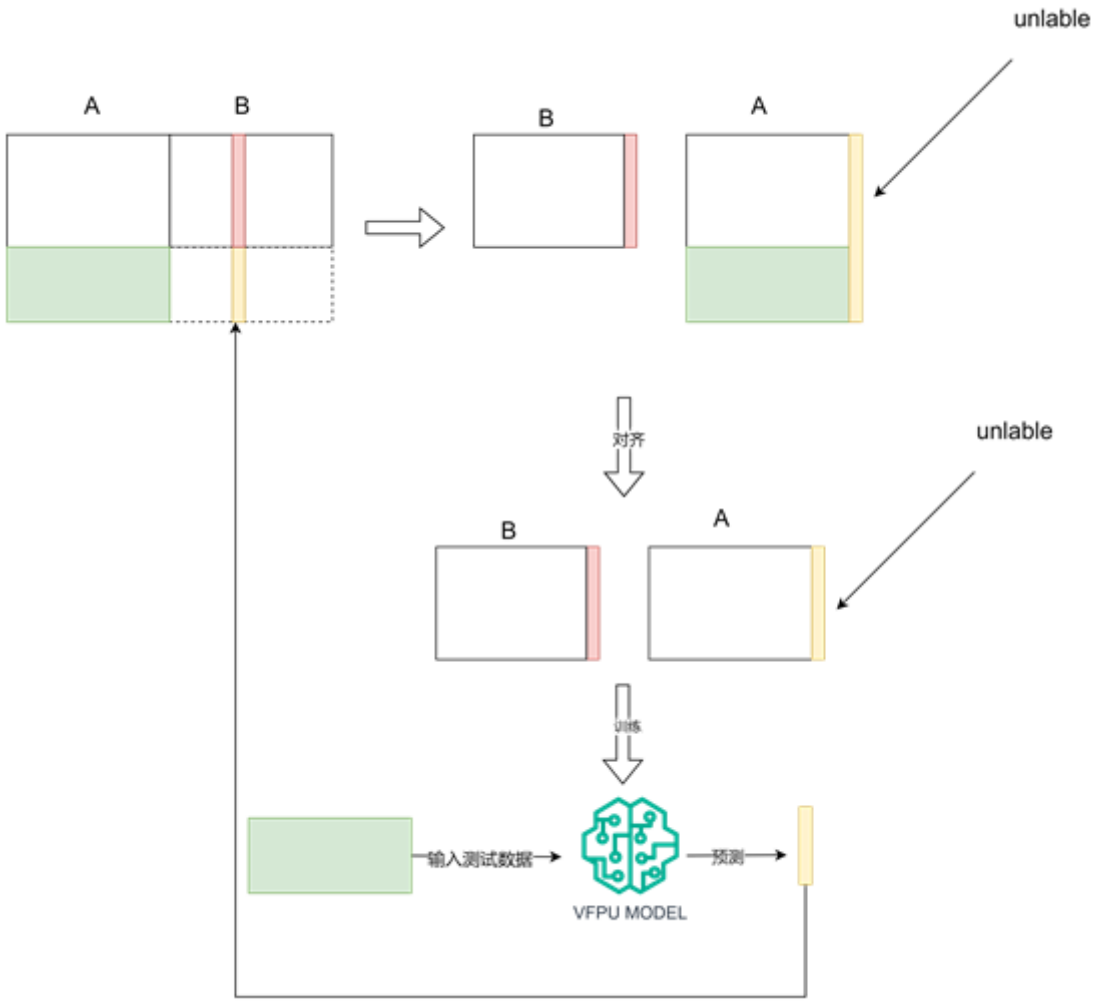
实验设计

有两个数据集A和B，A的大小是100x11，最后一列是标签列；B的大小是50x10，没有标签，创建一个VFPU分类器，VFPU有fit和predict方法，vfpu.fit(DataA, DataB)，训练好模型后，利用vfpu生成B的剩下50行数据

# VFPU_GEN 用半监督方法生成数据的实验设计




# 实现的一些细节




使用一个二分类器来做回归任务。可以通过将y的值转化为类别标签来实现。具体来说，可以将y的值按照某种方式映射到0和1，然后在预测时，将预测的概率转化回原来的值。

一个可能的实现方式，首先，我们需要定义一个阈值，例如0，将y<0的值映射到类别0，y>=0的值映射到类别1。在预测时，我们可以将预测的概率直接作为预测的值。注意，这种方法的效果取决于y值的分布和映射方式。

```
<PYTHON>import numpy as np
```

```python
class RegressorWrapper:
    def __init__(self, clf):
        self.clf = clf

    def fit(self, XA, XB, y):
        self.y_min = y.min()
        self.y_max = y.max()
        y_norm = (y - self.y_min) / (self.y_max - self.y_min)
        y_binary = np.where(y_norm >= 0.5, 1, 0)
        self.clf.fit(XA, XB, y_binary)

    def predict_proba(self, XA, XB):
        y_pred_proba = self.clf.get_out_of_bag_score()
        y_pred = y_pred_proba * (self.y_max - self.y_min) + self.y_min
        return y_pred
```