MiniDTM项目报告书

1 项目概述

1.1 文档说明

本文档为"MiniDTM: 一个精简型数据交易市场平台"的项目报告书，包含项目概述、技术框架、功能模块与系统设计、运行展示、心得与建议五部分，全文共8567字，作者朱晨阳，学号3200103432

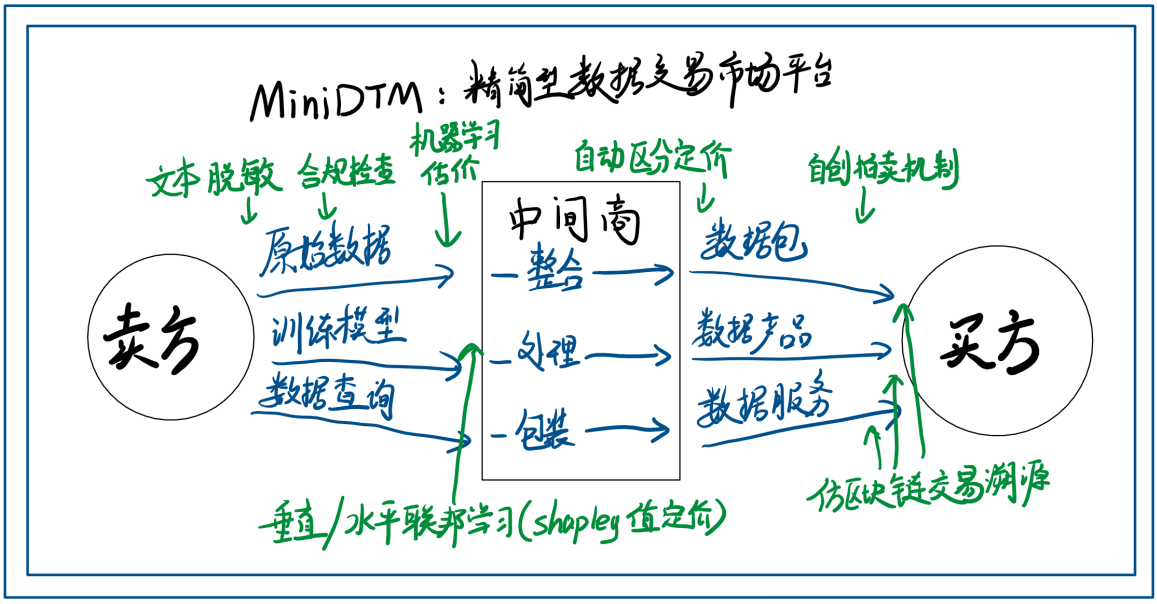
1.2 需求分析

需要实现一个面向三类用户：卖家、中间商、买家的数据要素交易平台。其中卖家提供数据与模型，中间商管理与分配交易，买家可以根据要求购买产品，平台需要能够支持上述功能，保证可行性与安全性，并提供其他服务。

在实现基本的出售-分配-购买工作流的基础上，需要突出数据要素的特征，融合课程所学知识，实现数据交易平台的独特功能如定价、合规、脱敏溯源等，并加入自我的创意元素对某些方面进行独创的改良，力争实现数据交易市场可信性、公平性、盈利性、安全性、高效性的设计要求。

1.3 平台概述

平台工作流如下图所示



- 对于数据要素的三种主要交易形式(数据、模型、服务)，设计三条主线，串联三类用户，对于每类用户各自提供独特的图形界面，前端界面与后端数据库贯通，构建平台功能基础，实现数据要素的交易流动与平台运行。

卖方机制	描述
原始数据出售	将脱敏、平台认证合规后的数据挂在市场上出售
训练模型出售	将模型相关参数挂在市场上出售
数据查询服务提供	将数据内容信息与查询接口挂在市场上出售

中间商机制	描述
数据打包	基于卖家的原始数据，打包后通过市场出售
模型处理	基于卖家的模型参数，处理（通过联邦学习）成数据产品通过市场出售
查询包装	基于卖家的接口查询，包装成数据服务后通过市场出售

买方机制	描述
数据包拍卖	以本人独创的拍卖形式参与竞拍中间商提供数据包
训练模型产品购买	以一口价形式购买中间商提供的数据服务
数据服务购买	以一口价形式购买中间商提供的数据服务

- 在三条主线中，插入数据市场平台独特功能，涵盖数据脱敏、合规检查、智能估价、模型训练、区分定价、独创拍卖、交易溯源功能，突出数据要素特征，融合课程所学的全部知识，结合自我创意进行功能拓展，实现功能丰富、可操作性性强、展示性好、可靠性完备、趣味性高、学习价值大的数据交易市场平台。

平台功能	描述
合规认证	平台作为第三方，对数据文本的合规性进行初步审查与认证
数据脱敏	提供文本数据批量脱敏服务
价格评估	根据数据量、准确性、历史价格通过机器学习自动评估初步价格
模型训练	基于卖家提供的模型(伪)进行水平或垂直联邦学习
交易机制	可选一口价或平台独创的动态保留价格的多轮二价拍卖机制
分类定价	帮助中间商进行精准刀法，区分定价，实现反套利
交易溯源	提供类似区块链的交易溯源，MD5水印，确保平台交易过程可溯源

2 技术框架

2.1 技术栈

语言：Python3

前端：PyQt5（一个python库，可通过pip install PyQt5获得）

后端：MySQL

代码量：约2400行

开发时间：约45小时

2.2 项目文件

文件树结构如下所示：

```
.
└── MiniDTM
    ├── database_funs.py
    ├── vertical_federated_learning.py
    ├── window_buyer.py
    ├── window_main.py
    ├── window_middleman.py
    ├── window_seller.py
    ├── hash_funs.py
    ├── horizonal_federated_learning.py
    ├── readme.txt
    ├── datas
    │   ├── history_price.csv
    │   ├── iris_test_1.csv
    │   ├── iris_test_2.csv
    │   ├── iris_train_1.csv
    │   ├── iris_train_2.csv
    │   ├── sensitive.txt
    │   ├── vertical_test.txt
    │   ├── vertical_train_1.txt
    │   └── vertical_train_2.txt
    ├── documents
    │   ├── MiniDTM.PDF
    │   └── MiniDTM项目报告书.md
    ├── images
    │   └── bg_son.jpg
    ├── requirements
    │   └── requirements.txt
    ├── sql
    │   └── minidtm.sql
    └── tools
        └── gen_history_price.py
```

其中MiniDTM文件夹下直接有8个.py文件，是项目全部代码，主函数在window_main.py中。datas文件夹下包含运行所需数据文件，缺一不可。documents文件夹下存放项目文档，images文件夹下存放图形界面所需文件，requirements文件夹下存放项目所需依赖组件的说明，sql文件夹下存放构建项目所用的mysql数据库的sql文件，方便快速导入搭建库运行。

2.3 运行环境

操作系统：开发采用Windows10环境，理论上在能够正常运行python3和MySQL的任何系统都可以

Python版本：开发采用3.10.5版本，理论上能够满足所有依赖库里最低需求即可，推荐3.6以上版本

Python依赖：开发时皆为2022年9月时pypi.org上的最新版本

MySQL版本：开发采用8.0.28版本

IDE: 开发采用VsCode，理论上记事本也可以

内存：开发时为64G，理论上能上网就够了

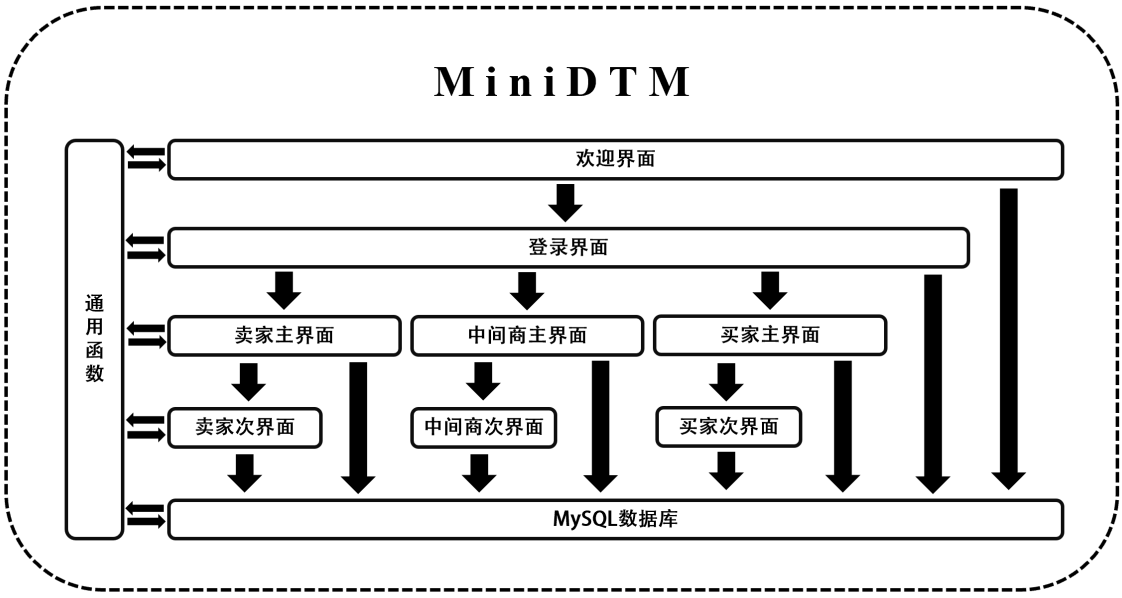
CPU：开发采用intel i9-10980HK，理论上能上网就够了

所在硬盘空余空间：开发时为200G，理论上10MB就够了

3 功能模块与系统设计

3.1 系统架构

以前端界面为主导，每个界面为一个类，该界面相关的所有功能函数都集成在类的成员函数里，实现较为优雅的面向对象程序设计。同时，部分通用的函数，如数据库操作相关的函数则抽象出来组成单独的模块，供所有界面类来调用。



该系统设计优点是前后端高度融合，便于调整改动，功能位置明晰。缺点也是前后端高度融合，整体存在一定冗杂性，被前端牵着鼻子走。但根据综合考量与实际开发情况，该架构是与需求目标和开发者能力高度匹配，较为适合该项目的设计。

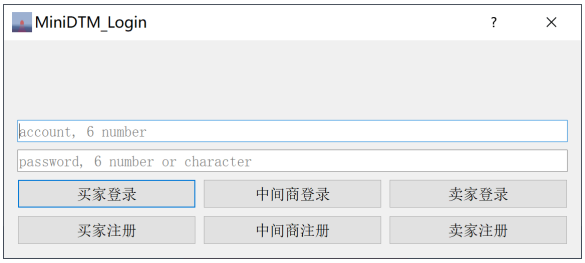
3.2 模块说明

3.2.1 欢迎与登录模块

该模块用于启动平台与用户登录。主要有用户管理、生成历史交易哈希区块的功能、查看作者信息与致谢信息功能。

- **用户管理**

点击欢迎页面的`Login/Register`按钮，即可打开登录界面`class Window_Login(QDialog)`



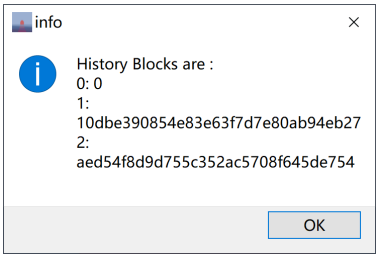
`注册：bt_register_clicked(self)`

三个注册按钮都被绑定至该函数。函数被调用时，首先获取账号与密码输入框的输入，判断是否符合格式要求。如否，则弹窗提示输入无效。若有效，则获取按钮发送者，然后根据发送者是买家、中间商还是卖家注册按钮，调用`db_add_user`，在数据库中对应的账号表内插入数据。操作成功后弹窗提示注册成功。

`登录：bt_login_clicked(self)`

三个登录按钮都被绑定至该函数。函数被调用时，首先获取账号与密码输入框的输入，判断是否符合格式要求。如否，则弹窗提示输入无效。若有效，则获取按钮发送者，然后根据发送者是买家、中间商还是卖家登录按钮，调用`db_query_user`，在在数据库中对应的账号表内查询数据。如果查询到对应数据，则弹窗提示登录成功，然后打开对应的买家、中间商或卖家主界面，同时传递登录使用的账号作为身份标识给下一个界面。

- **生成历史交易哈希区块**



`gen_history_hash(self)`

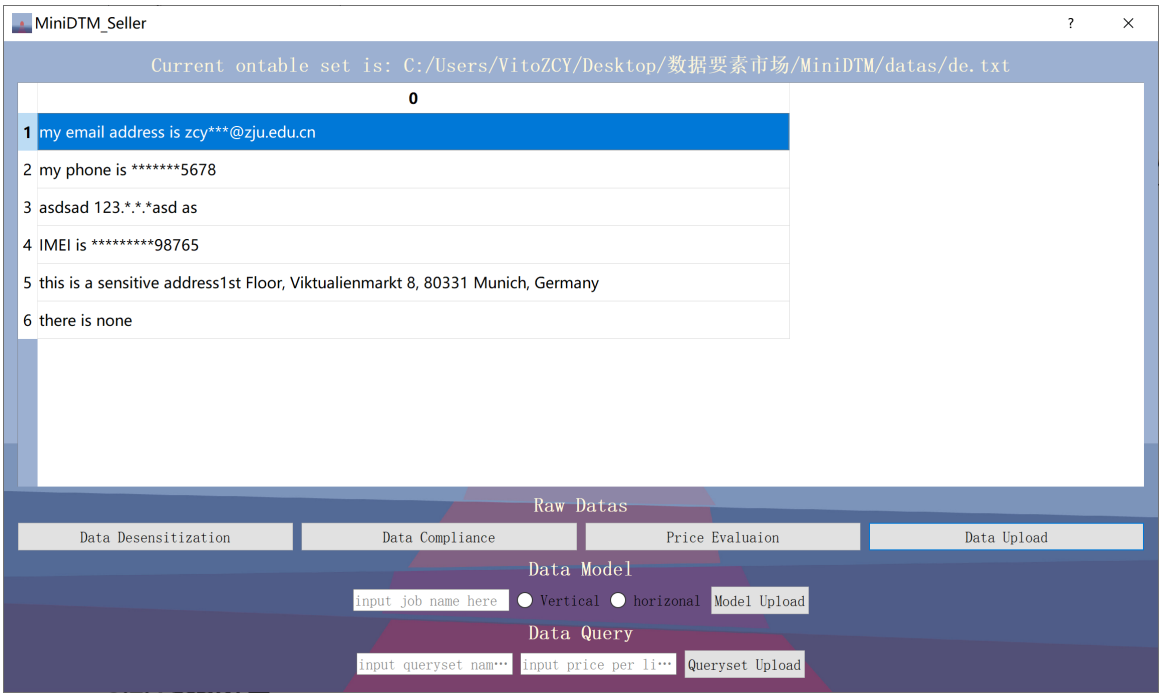
首先调用`db_get_table_datas`获取history表中所有历史数据，拆分出其已有的哈希块。如果最近一次交易记录没有生成新的哈希块，则根据上一块的哈希值与最近一次交易记录，调用`genearteMD5`生成新的哈希块，并写入history表。然后弹窗显示所有已生成的哈希块。

- **查看作者信息与致谢信息**

`thanks_bt_clicked`、`info_bt_clicked`

直接弹窗显示作者信息与致谢信息

3.2.2 卖家模块



- **原始数据处理**

`rawdatas_upload(self)`：上传原始数据集

调用`upload_file_into_lines`来读取文件形成列表，调用`align_lines`将每一行长度对齐，缺失的用空字符串填补。将格式化的列表显示到表格中。在将数据集名上传至数据库之前，读取数据库中已有原始数据集检查重名的情况，如果不重名，则正常上传，否则警告上传失败。

`data_desensitization(self)`：数据脱敏

调用`upload_file_into_lines`来读取文件形成列表，调用`find_and_replace_sensitive_words`来对每一行进行查找并替换敏感词，脱敏后保存数据到本地，同时更新显示在表格上

`check_exist_sensitization(self)`：合规检查

调用`upload_file_into_lines`来读取文件形成列表，调用`find_and_get_sensitive_words`来获取每行中的敏感词，将敏感词显示在表格上，如果没有，则提示数据合规

`price_evaluaion(self)`：机器学习估价

读取自己生成的历史价格文件，调用`sklearn.linear_model`进行线性回归模型的训练，然后弹窗输入需要估计的准确率与大小，生成估计的价格

- **数据模型处理**

`model_upload(self)`：上传模型

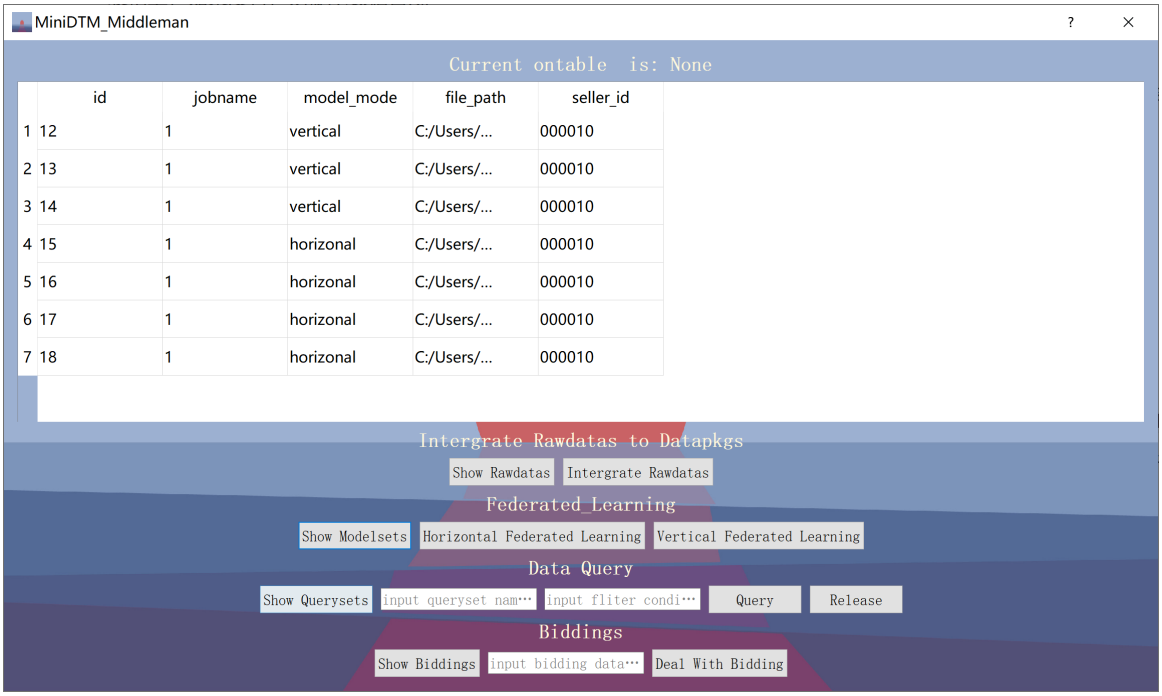
检查格式符合规范后，读取所上传的模型类型是垂直还是水平，然后将文件位置与卖家ID上传到modelsets表

- **数据查询集上传**

`queryset_upload`：上传数据集

读取输入框中的名字与价格，调用`upload_file_into_lines`与`align_lines`生成格式化的列表，调用`db_create_table`来创建表，调用`db_add_one_row_with_prefix`向表中插入数据，同时调用`db_add_one_row`更新queryset表，最后将读取的信息显示在表格上

3.2.3 中间商模块



`show_rawdatas` `show_modelsets``show_querysets` `show_biddings`：在表格中显示对应数据表

- **整合原始数据发布数据包**

`integrate2pkg(self)`

弹窗获取要整合的原始数据集名，启动`window_middleman_sub`子窗口，在子窗口中输入名字、accuracy等信息，然后根据自己设计的一条曲线(后文阐释)，对价格进行精准刀法生成子包，发布到market_datapkg表上

- **联邦学习**

`horizontal_federated_learning(self)`

读取分散的文件整合到一起后，调用`fun_horizional_federated_learning`对完整的数据进行SVM机器分类的联邦学习，并用蒙特卡洛算法计算shapley值，基于shapley值得到优化后的最终学习结果，并显示不同用户数据的贡献度百分比

`vertical_federated_learning(self)`

读取分散的文件整合到一起后，调用`fun_vertical_federated_learning`对完整的数据进行梯度下降法的逻辑斯蒂回归，完成后弹窗显示最终的成果

- **数据查询**

`query_datas(self)`

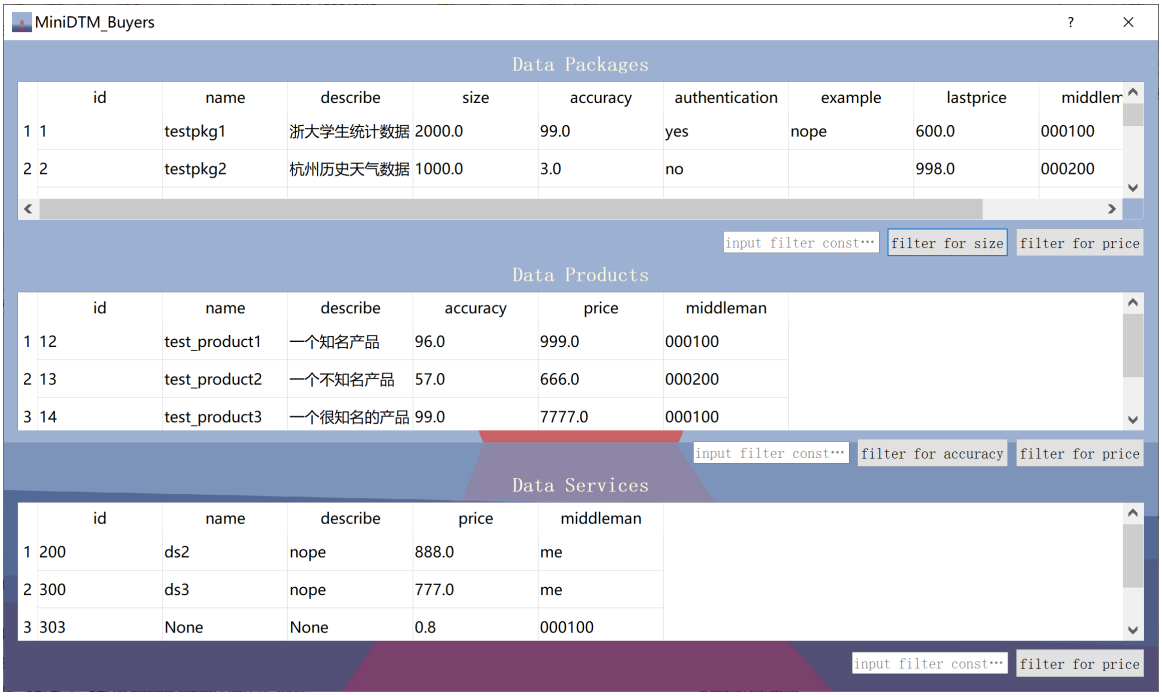
获取文本框输入要查询的数据集，从MySQL查询对应数据集的数据，再根据querysets中记录的价格与查询到的条数，计算得到价格弹窗提示，最后将查询结果显示在表格中

- **处理拍卖竞价**

`dealing_bidding(self)`

获取文本框输入要处理的拍卖目标，从MySQL查询对此所有的出价，输入保留价格，判断是否拍卖成功。如果成功，则在MySQL中删去该数据包，并在交易记录中加入该记录，弹窗提示成功拍得者的id与价格。如果不成功，则将本次的保留价格更新到MySQL公布。最后，将本轮拍卖所有记录给清除。

3.2.4 买家模块



- **数据包部分**

双击表格，可以自动获取对应行的包信息，弹出出价窗口，输入价格即可参与拍卖，在MySQL bidding表中加入竞拍记录，等待中间商处理

在过滤框中输入条件，点击对应按钮，即可进行筛选，留下符合条件的显示在表格中

- **数据产品与数据服务**

双击表格，可以自动获取对应行的对象信息，弹出提示窗口，提示价格与是否确定购买。如确定，则在对应表中删去该物品，刷新表格显示。同时在history表中加入交易记录

在过滤框中输入条件，点击对应按钮，即可进行筛选，留下符合条件的显示在表格中

3.3 平台特色说明

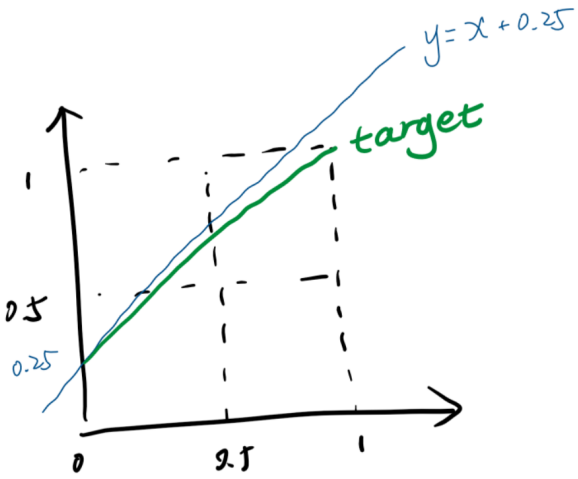
3.3.1 文本脱敏与合规检查

把第四天和第五天的文本脱敏作业缝了进来。整体实现逻辑难度不大，主要在于繁琐，因为要同步到PyQt的表格组件上，容易出现各种稀奇古怪的格式问题。因此自己创造了所谓"format_lines"，对于没有表头的数据，自动添加表头(MySQL里不能用数字开头作列名和表名，又是一个深坑)。对于每行长度不一的数据，添加空字符串进行列数对齐。开发过程中把具体函数都抽象出来成为可复用的通用组件，过程中又克服了一堆bug，但加快了后期开发速度

3.3.2 机器学习估价

主要基于accuracy和size两个属性，本意是基于平台原有历史交易数据，但限于数据量实在太少，只好自己用随机数生成了一堆数据，利用sklearn的线性回归工具进行训练，构成预测模型

3.3.3 精准刀法差别定价



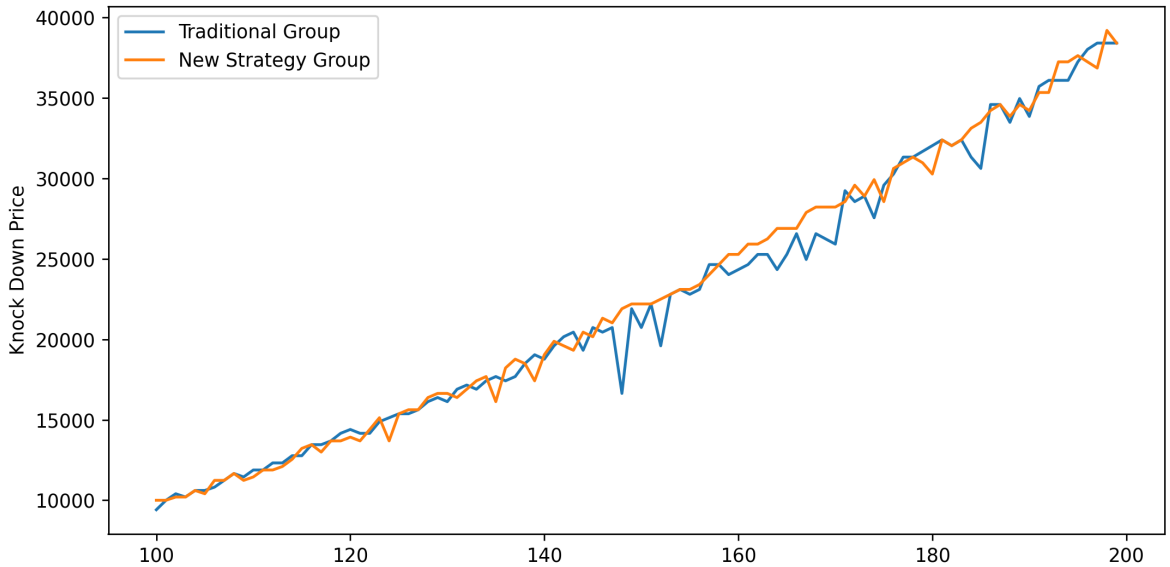
目标是把一个完整的集拆出价格更便宜的子集，目标要求反套利，应实现50%的数据60块钱，100%的数据100块钱这样的效果。因此要找到一条曲线，实现1.斜率小于1；2.斜率的斜率小于零

如上图所示，找到了-0.25x^2 + 2x +0.25这条神奇的曲线，在子集有原集一半“好”时，价格为原集的68.75%。且无论取60%“好”、70%“好”、80%“好”都可以满足要求。实际实现时，就先计算一个子集有原集的多少"好"，然后根据这条曲线计算出价格

3.3.4 自创拍卖机制

根据第二天的作业内容，把自己“创造”的这种拍卖机制应用了过来。整体就是对二价拍卖的修改，拍卖可重复多轮直至成功，成交规则仍与二价拍卖相同。所有人同时给出出价，设出价最高者为Vo，同时卖方事先设定一个保密的阈值Vs，选取Vo-Vs作为保留价格Vr，替代传统拍卖中事先确定的保留价格。如果第二高的出价高于保留价格，由出价最高者支付第二高的价格成交。但如果第二高的价格低于保留价格，则会公布本轮的保留价格，重新开始拍卖，直至成交。

根据自己设计的测试，假设共有100位竞拍者，每个人出价为r*r，r为1-x的随机数。共模拟进行100个不同商品的拍卖，x遍历100-199之间整数。分为新方案组与传统组两组，传统组为保留价格为10x的二价拍卖，交易费用为成交价格的3%，新方案组卖方设计的阈值Vs为5x，交易费用为成交轮的平均价格与实际成交价格的差值之差的4.5%。最终成交价格如下图所示：



如图所示，在最终成交价格上，新方案有微弱的优势。（实际算平均值也确实有比较稳定的微弱优势）

本身二价拍卖就是满足DSIC、福利最大化与计算高效的拍卖策略，这种对二价拍卖改动实际是破坏了福利最大化的，但实现了拍卖价格的微弱提高，保证了卖家福利的最大化。

3.3.5 联邦学习

把第三天作业和第七天作业缝了进来。本意是想所有数据同步到平台，但由于数据量极大，批量读取速度过慢，容易造成程序崩溃，故还是改为了目前只传文件路径的折中方案。由于计网等还没学，对通信一无所知，所以在自己能力范围内整体实现了一个"伪"联邦学习。

3.3.6 仿区块链交易溯源

对历史交易记录生成MD5。不是每次交易都生成，而是每次点击查看历史哈希值时，如果最近的一次交易没有生成，则会将所有没有加入到区块中的数据，和上一次的哈希值一起生成新的哈希值，实现像区块链一样的区块不断增长的效果。整体实现过程还是比较优雅的，本来想自己手搓MD5，发现调库实在太方便还是算了(:

3.4 数据库设计

buyer : 买家账号信息

account	password

middleman ： 中间商账号信息

account	password

seller : 买家账号信息

account	password

history：历史交易记录

id	name	price	seller	buyer	middleman	hash

bidding：拍卖出价记录

bid_id	datapkg_id	buyer_id	price

market_datapkg：市场上发布的数据包

id	name	describe	size	accuracy	authenication	example	lastprice	middleman

market_dataproduct：市场上发布的数据产品

id	name	describe	accuracy	price	middleman

market_dataservice：市场上发布的数据服务

id	name	describe	price	middleman

modelsets：模型集

id	jobname	model_mode	file_path	seller_id

querysets：可供查询的数据集

name	price	seller_id

rawdatas：可供使用的原始数据集

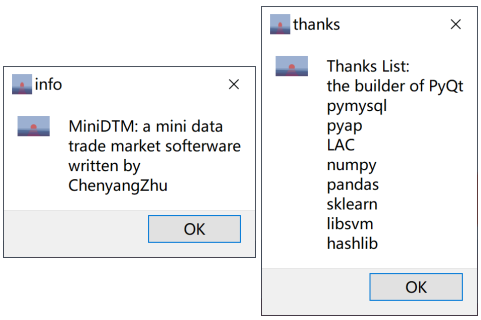
name	seller

以及用户自己上传的其他数据表。如果不带表头，则会自动生成row_index，一个自动增长的列来当主键，然后以row+i列作为列名

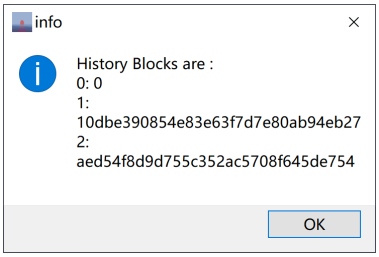
4 运行展示与分析

4.1 主界面与用户管理模块

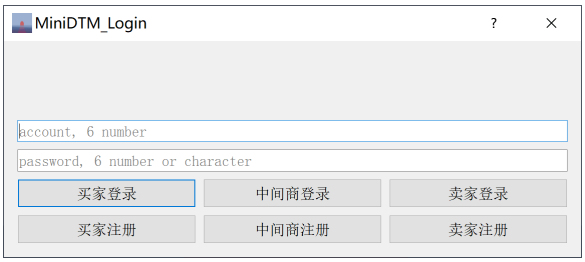
欢迎界面是用户点开应用所能看到的第一个窗口，包含顶部标题与底部四个按钮。点击AboutUs和Thanks按钮，即可分别看到作者信息与致谢信息



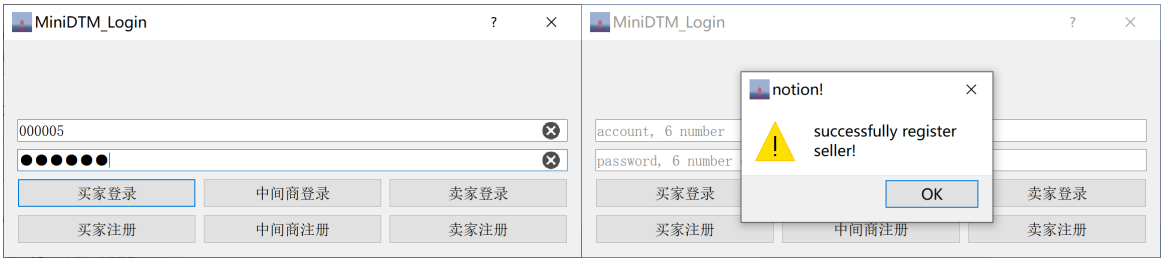
点击History Hash按钮，即可看到根据历史交易记录生成的哈希区块



点击第一个Login/Register按钮即可跳出登录页面



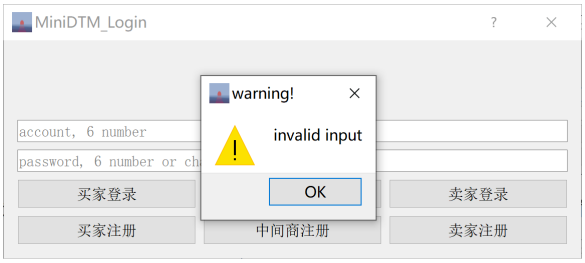
输入6位数字组成的账号，与6位数字和字母组成的密码，点击卖家注册，即可看到弹窗提示注册成功



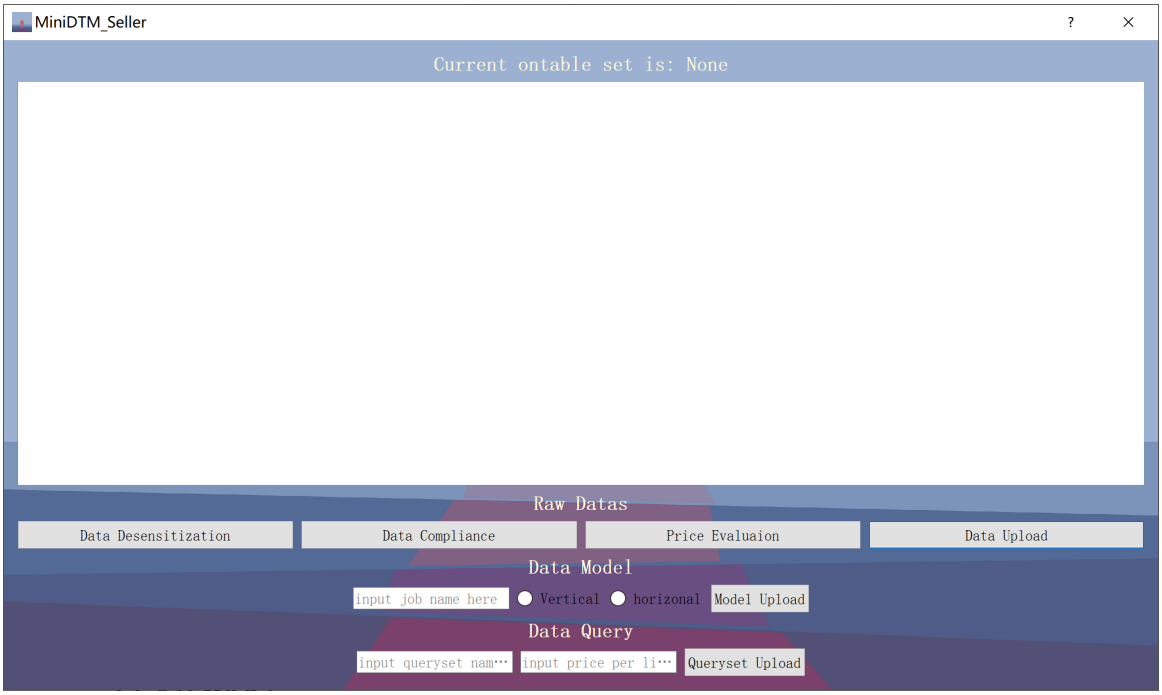
检查MySQL发现新数据添加成功

	account	password	encrypt_id
▶	000010	123456	NULL
	000050	123456	NULL

如果输入不符合格式规范，也会提示注册失败



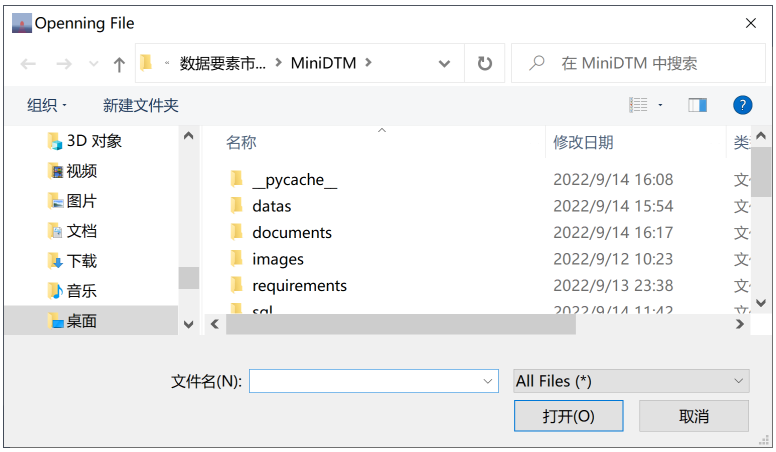
输入刚刚注册的账户，点击卖家登录，即可打开卖家主界面



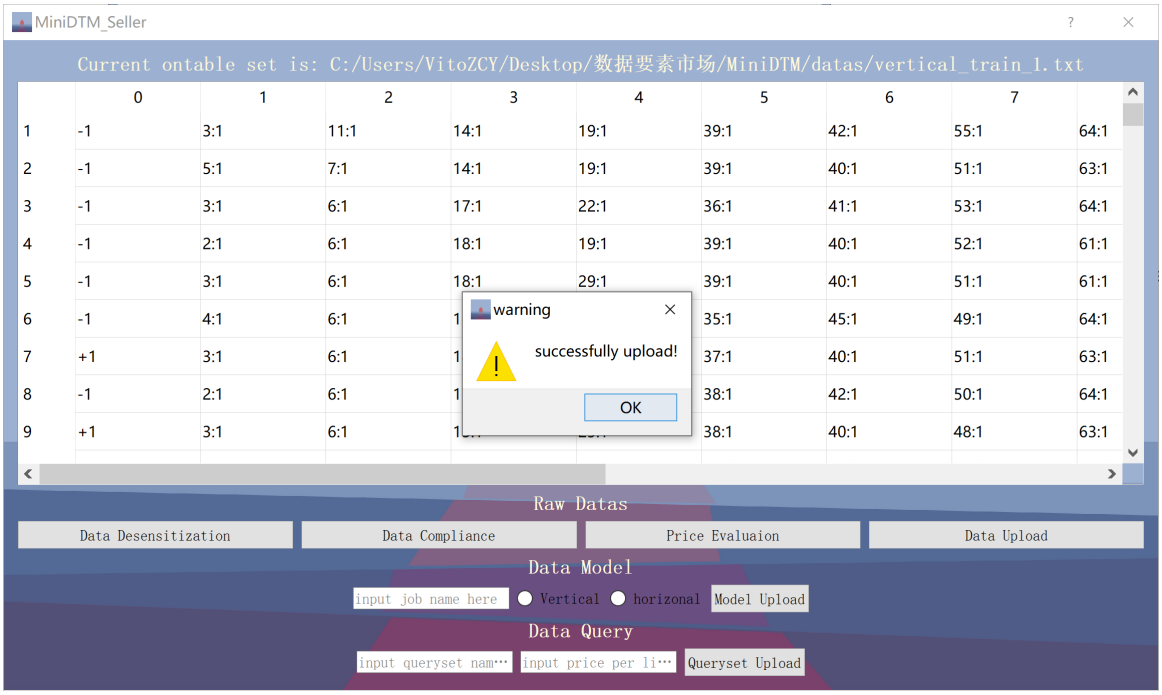
4.2 卖家模块

> Raw Datas

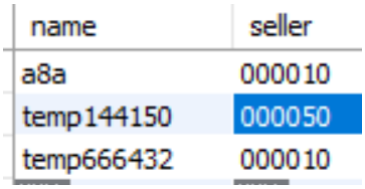
刚打开页面时，还未显示任何数据，故顶部提示表格中的数据集为None。作为卖家，可以向平台上出售数据集，点击Data Upload按钮，选择要传送的数据集



选取符合格式的文件后，数据集将自动显示在表格上，更新表格顶部标题

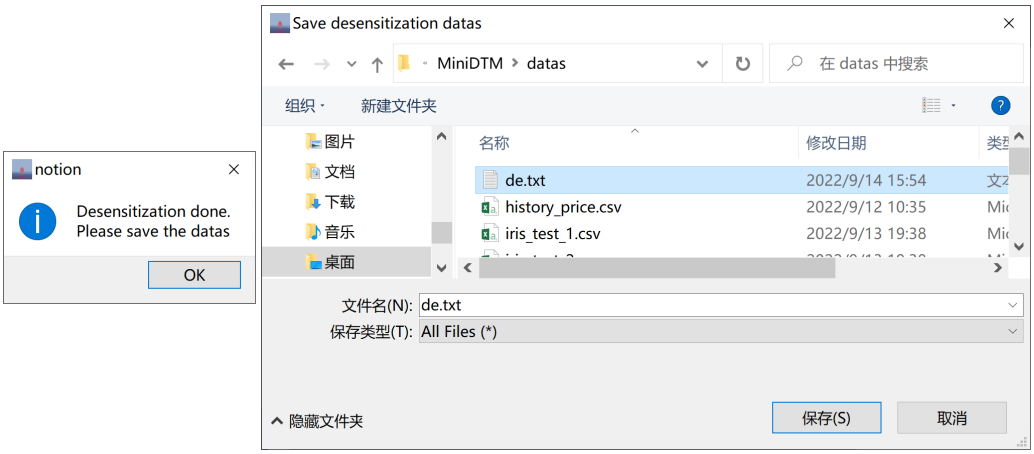


同时更新MySQL

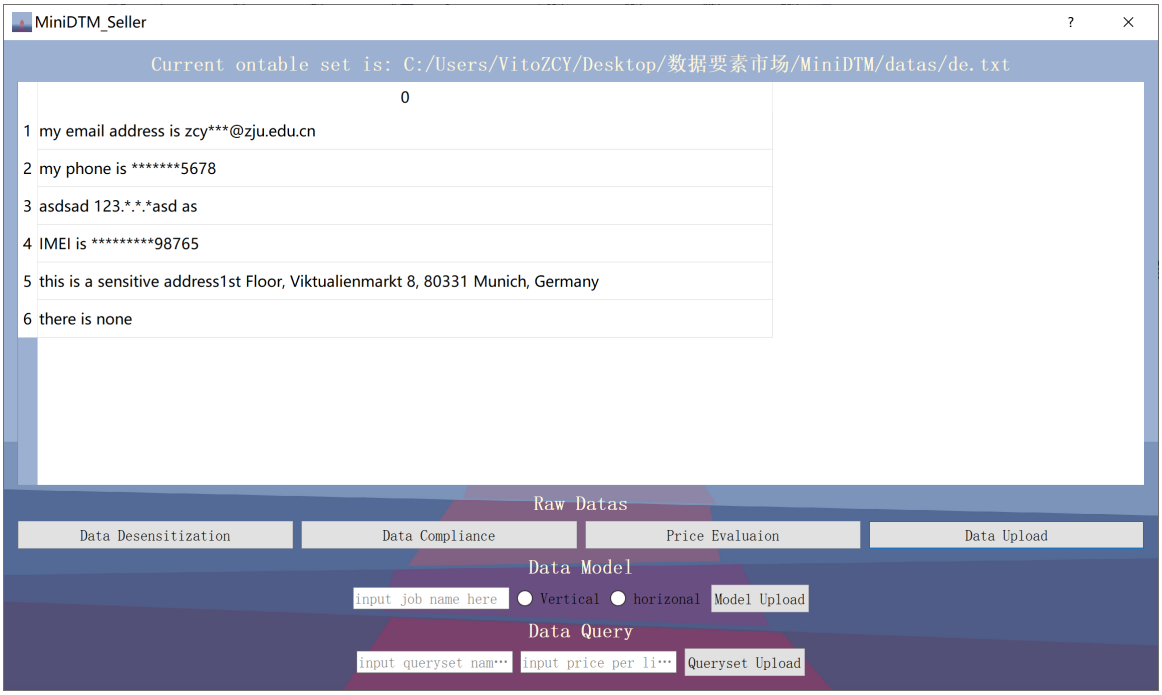


name	seller
a8a	000010
temp144150	000050
temp666432	000010

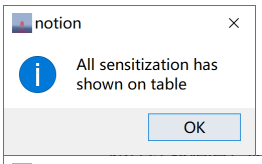
可以看到，对于原始数据集，共提供了数据脱敏、合规检查、价格评估、数据上传四个功能。点击数据脱敏按钮，选择准备好的含有敏感信息的数据，脱敏完成后将弹窗提示保存脱敏后数据



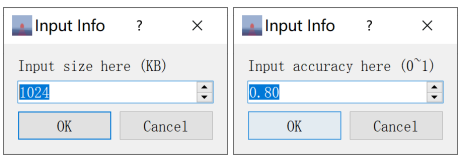
保存好数据后，脱敏后数据将自动同步更新到界面表格上



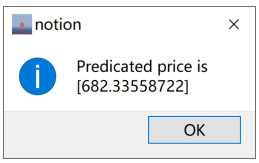
对于数据合规检查按钮Data Compliance，点击后同样是选择敏感数据文件，完成检查后将弹窗提示，点击确定后，会自动将检查到的敏感信息刷新到表格上。如果不存在敏感信息，则会提示合规，无敏感信息



点击Price Evaluation按钮，开启价格评估，根据弹窗先后输入size与accuracy

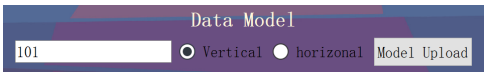


返回后台根据历史价格通过机器学习多项式回归生成的预测价格

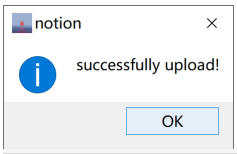


Data Model

对于数据模型，首先在输入框中输入任务名，选择垂直或者水平



点击Model Upload，弹窗提示上传成功，同时可在MySQL中看到刚上传的模型文件



id	jobname	model_mode	file_path	seller_id
2	123	vertical	C:/Users/VitoZCY/Desktop/数据要素市场/final/datas/vertical_test.txt	000010
3	1231231231	vertical	C:/Users/VitoZCY/Desktop/数据要素市场/final/datas/vertical_train_1.txt	000010
4	1234	vertical	C:/Users/VitoZCY/Desktop/数据要素市场/final/datas/vertical_train_2.txt	000010
5	20	horizonal	C:/Users/VitoZCY/Desktop/数据要素市场/final/datas/iris_test_1.csv	000010
6	20	horizonal	C:/Users/VitoZCY/Desktop/数据要素市场/final/datas/iris_test_2.csv	000010
7	20	horizonal	C:/Users/VitoZCY/Desktop/数据要素市场/final/datas/iris_train_1.csv	000010
8	20	horizonal	C:/Users/VitoZCY/Desktop/数据要素市场/final/datas/iris_train_2.csv	000010
9	101	vertical	C:/Users/VitoZCY/Desktop/数据要素市场/MiniDTM/datas/vertical_train_collect.txt	000050

同样的，对于job name将会有拼写检查

Data Query

输入queryset的name和每条数据的价格，点击QuerySet Upload按钮，(此处有拼写检查，如果不通过不会继续)，选择要上传的文件，即可在MySQL中创建新表，包含所上传的所有数据(运行速度较慢，可能会有卡顿)

row_index	row_0	row_1	row_2	row_3	row_4	row_5	row_6	row_7	row_8	row_9	row_10	row_11	row_12	row_13	row_14
1	-1	3:1	11:1	14:1	19:1	39:1	42:1	55:1	64:1	67:1	73:1	75:1	76:1	80:1	83:1
2	-1	5:1	7:1	14:1	19:1	39:1	40:1	51:1	63:1	67:1	73:1	74:1	76:1	78:1	83:1
3	-1	3:1	6:1	17:1	22:1	36:1	41:1	53:1	64:1	67:1	73:1	74:1	76:1	80:1	83:1
4	-1	2:1	6:1	18:1	19:1	39:1	40:1	52:1	61:1	71:1	72:1	74:1	76:1	80:1	95:1
5	-1	3:1	6:1	18:1	29:1	39:1	40:1	51:1	61:1	67:1	72:1	74:1	76:1	80:1	83:1
6	-1	4:1	6:1	16:1	26:1	35:1	45:1	49:1	64:1	71:1	72:1	74:1	76:1	78:1	101:1
7	+1	3:1	6:1	18:1	20:1	37:1	40:1	51:1	63:1	71:1	73:1	74:1	76:1	82:1	83:1
8	-1	2:1	6:1	17:1	24:1	38:1	42:1	50:1	64:1	71:1	73:1	74:1	76:1	82:1	83:1
9	+1	3:1	6:1	15:1	25:1	38:1	40:1	48:1	63:1	68:1	73:1	74:1	76:1	80:1	

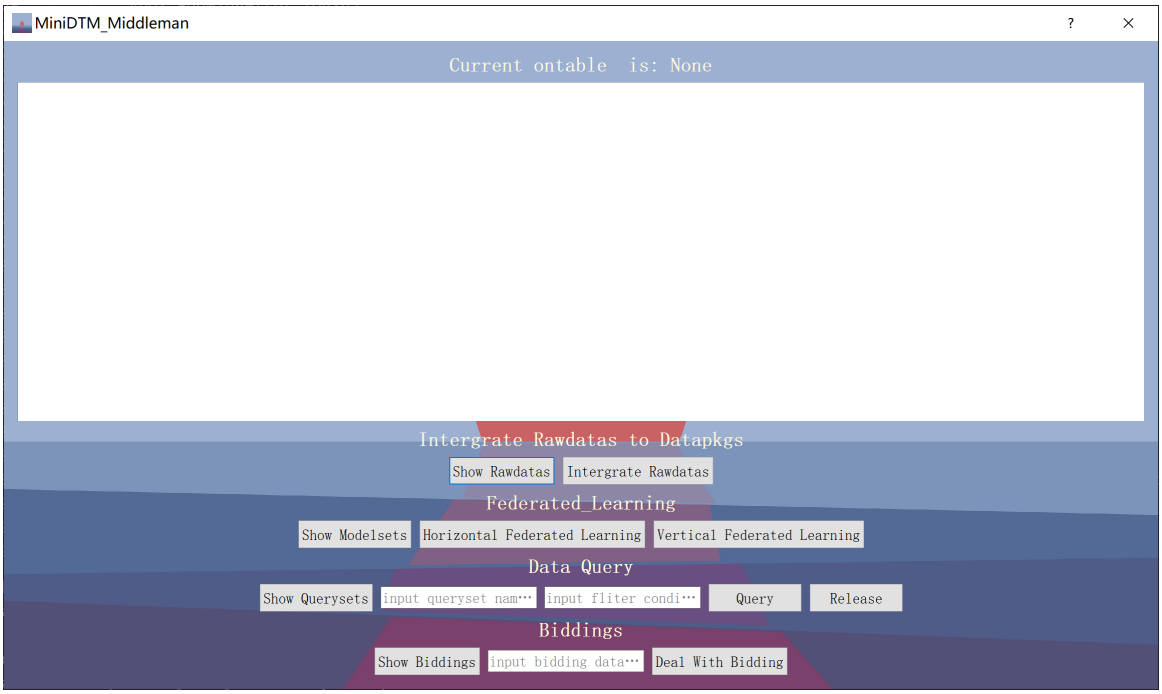
同时价格也被记录，供查询使用

name	price	seller
set101	10	000050
set9	123	000010

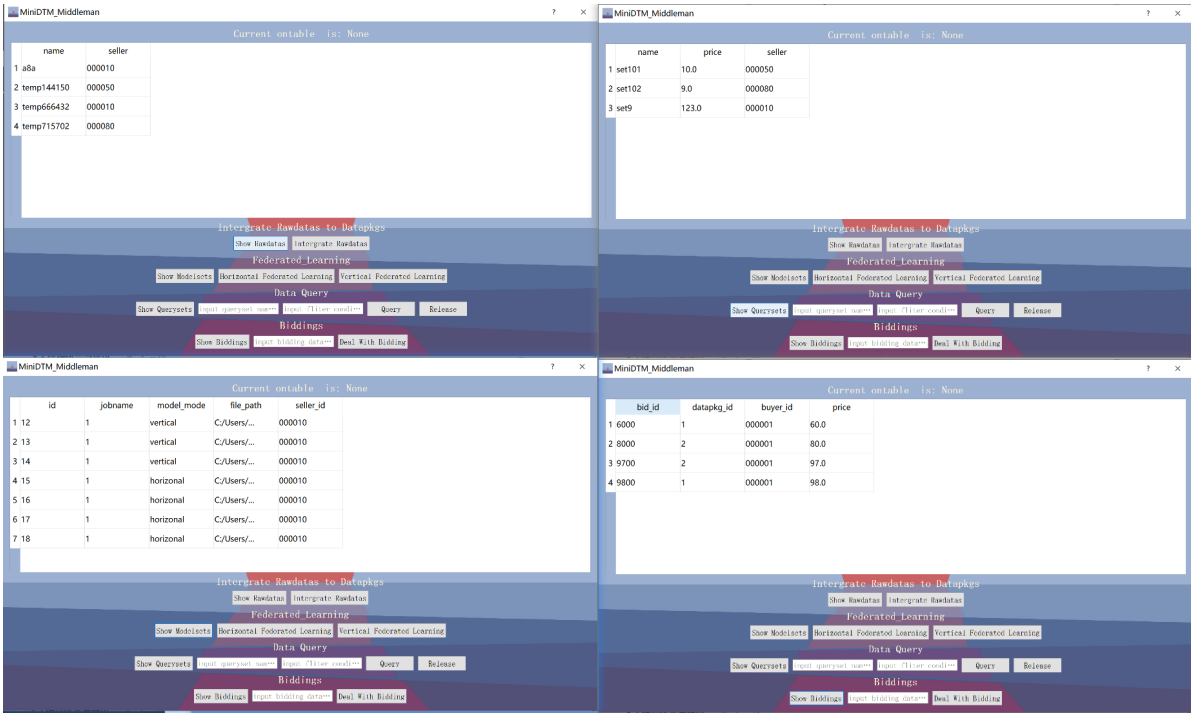
同时界面表格也将刷新为所上传的数据集

4.3 中间商模块

退出卖家界面，重新登录进入中间商模块

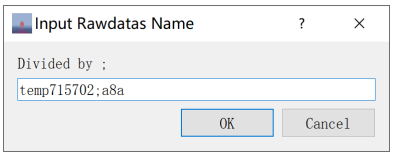


通过点击Show Rawdatas、Show Modelsets、Show Querysets、Show Biddings按钮，可以非常丝滑地调整表格显示的数据内容

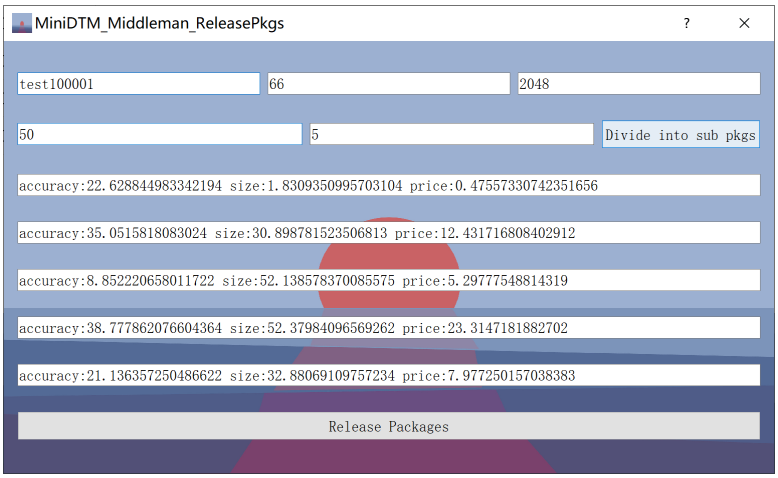


> Intergrate Rawdatas to Datapkgs

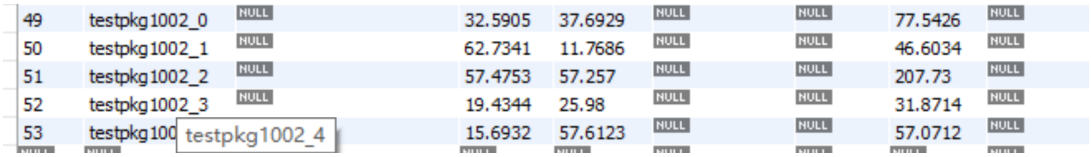
点击Intergrate Rawdatas按钮，弹窗提示输入要整合的原始数据集



根据框内原有提示，输入名字、accuracy、size、价格，以及要分成几份，点击Divide into sub pkgs，自动随机分成不同属性与价格的子包，设计了一条曲线实现反套利与买越多越便宜，实现全自动精准刀法



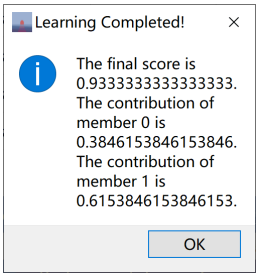
点击发布，即可在MySQL中看到发布的数据包



> Federated Learning

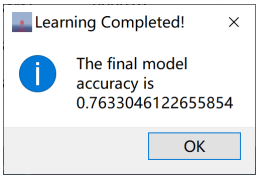
点击Horizonal Federated Learning按钮，即可获取库中所以标注为水平的文件，进行水平联邦学习

计算完毕后，弹窗显示最终得分，与根据shapley value所计算出的不同卖家模型的贡献百分百

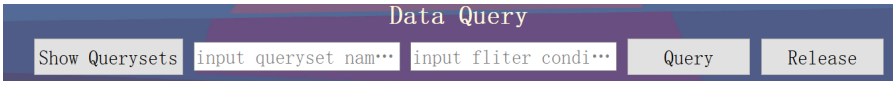


点击Vertical Federated Learning按钮，即可获取库中所以标注为垂直的文件，进行垂直联邦学习

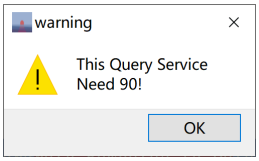
计算完成后，同样弹窗显示最终得分结果



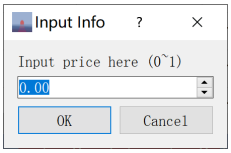
Data Query



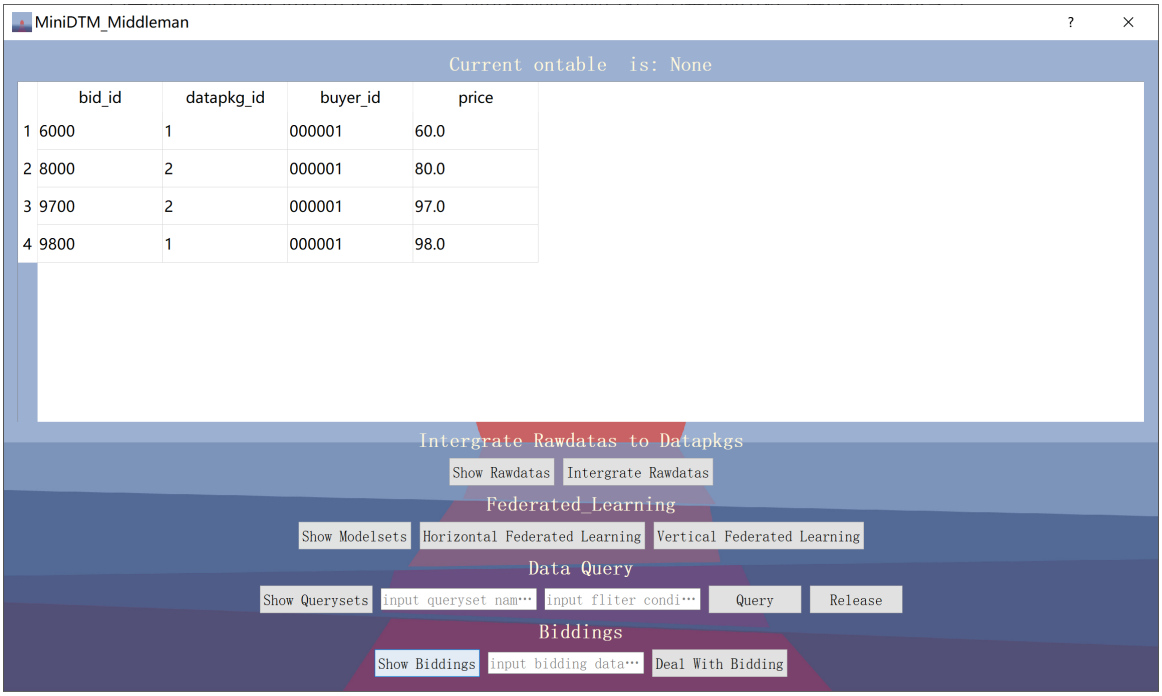
输入要查询的数据集名与筛选条件，点击Query按钮，即可得到弹窗提示自动计算出的价格



点击release，弹窗输入价格，即可将查询结果发布到市场上

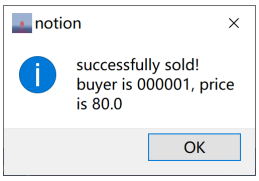


Biddings

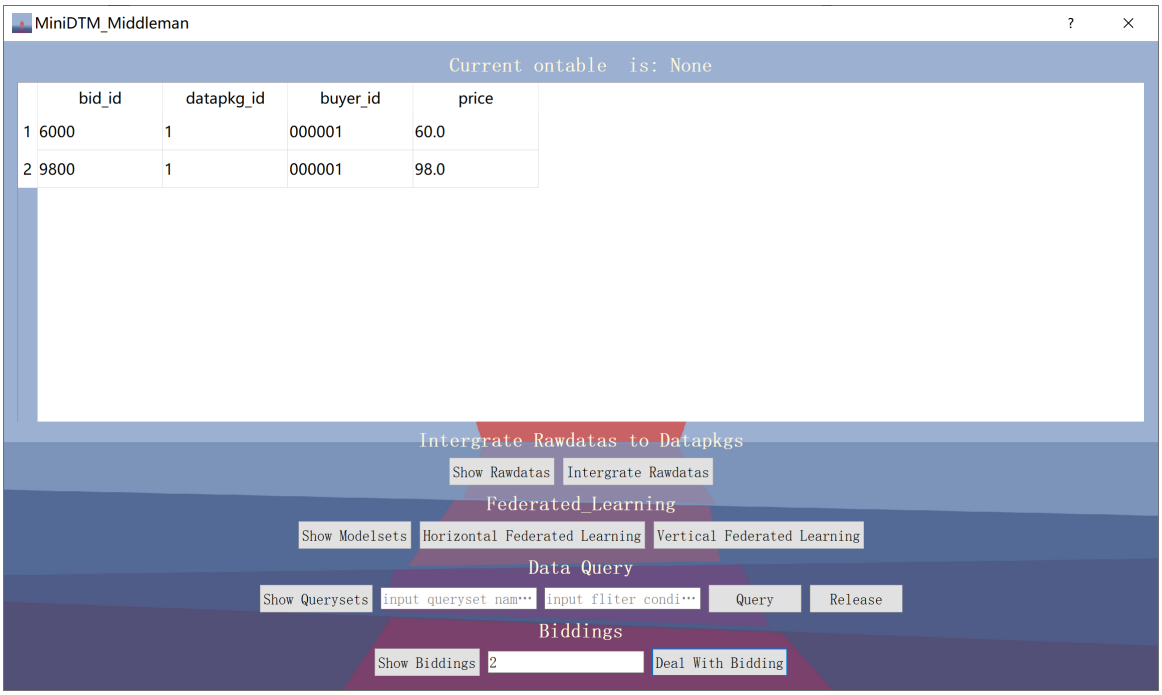


输入要处理的拍卖对象，即表格中的datapkg_id，弹窗输入保留价格

由于最高价减去保留价大于第二价，拍卖可以成功，弹窗提示成功，显示拍得者与成交价



拍卖结束后，表格中自动删去已结束的拍卖



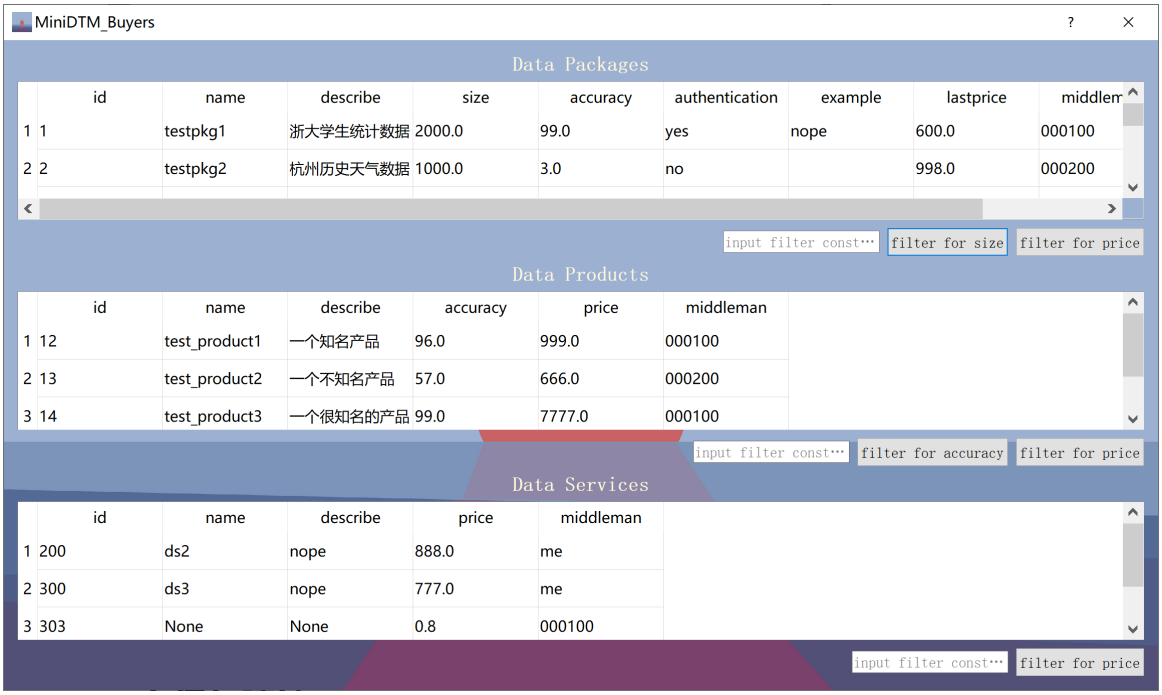
同时history中加入刚刚完成的拍卖数据

	id	name	price	seller	buyer	middleman	hash
▶	1	a1	1	NULL	000001	qwe	0
	3	1	1210	NULL	000005	000100	10dbe390854e83e63f7d7e80ab94eb27
	4	ds411	0.8	NULL	000002	000100	aed54f8d9d755c352ac5708f645de754
	5	ds4	666	NULL	000001	000200	NULL
	6	1	98	NULL	000001	000100	2b05831e7fd789ee639b4f6aad2f5cf3
	7	2	80	NULL	000001	000100	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

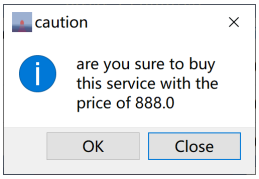
如果拍卖不成功，则会刷新lastprice，提示上一轮拍卖的最终保留价格，供以后参考

4.4 买家模块

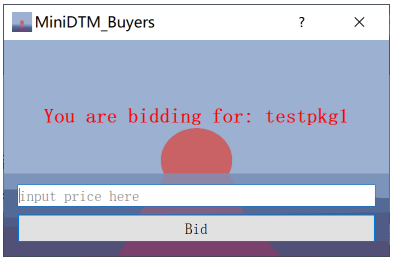
登录进入买家主界面，可以看到三类产品



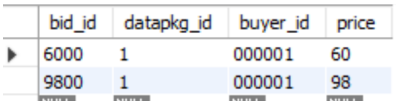
双击Data Products表或Data Services表，会自动获取当前点击表格对应的物品，弹窗提示购买确认



双击Data Packages表，会弹窗提示出价



出价会记录到bidding表中，等待中间商处理



bid_id	datapkg_id	buyer_id	price
6000	1	000001	60
9800	1	000001	98

5 心得与建议

5.1 大作业心得

肝，太肝了

2400行python，实在是没想到自己能写这么多，含泪也要把自己画的饼吃完

本来还有很多想做的，比如数据表的字段设计可以再优化优化，但最近洗澡时地漏总会被头发堵住，想想还是算了

Bug只有想不到，没有遇不到。比如数据库里account 拼成aacount 导致开新窗口时死活闪退，而PyQt经常闪退也没有错误提示，都要自己跟踪程序执行来定位错误，非常痛苦

从差不多6号开始到今天14号，一个多星期几乎每天都在写这个，不愿再笑

自夸时间

纯代码绘制前端（一般用QT会用图形拖拽的QT Designer，类似VB来画图），非常优雅。整体算不上漂亮，但起码属于能看，有个样子

整体代码也尽量写得美观，命名如其意。有些代码非常优美，比如flash_table_datas，会自动选取要用的列，而不是从第几列到第几列，改MySQL表时不会对前端显示造成bug，通用性非常好。

为表示成表格，花了非常多时间解决格式问题，包括字符串与数值、按行对齐等，抽象了很多通用函数。最终效果是前端表格显示数据切换非常丝滑，非常优雅。当然还是有一些屎山的地方，一部分是为了规范统一，另一方面是层层调用后不太敢乱动了……

项目工程量大，视图交互完备，前后端融合贯通，2400行代码，6个大窗口，N个小弹窗，11张数据表，一气呵成

把所学过的知识，所有写过的作业，全部给缝上去了，属实是一个究极缝合体，但还真给我缝成了:)，比如联邦学习部分数据分多少份都能跑，数据重复100份也能跑……

尽量向实际靠拢，在输入部分做了尽量多的拼写检查，但实际还是有不够的地方（深深体会到前端的痛苦了）

5.2 课程体会与建议

体会

第一天，挺好的，介绍了挺多，听得挺认真

第二天，挺不错的。但是作业直接懵逼。其实那本博弈论二十讲书挺好的，但一晚上看完显然不太可能……最后是上网看了一波解读，然后自己瞎想了一个二价拍卖的魔改

第三天，因为本人对数字不太敏感，讲概念还在认真听，一到公式推导就开始额……晚上作业又是直接懵逼，好在网上有一些类似的代码可以学习SVM分类是个啥玩意，然后跟着PPT里的蒙特卡洛算法写出了算shapley值的方法，经过炼丹调参最终在星期六实现了炼丹大成

第四天，上课讲的还比较有意思，但是作业又是直接懵逼。被逼速成了正则表达式

第五天，好像又是一堆数学推导……作业么……正则表达式也不够用了，自己谷歌找了一些库。然后这个平台太搞心态了……

第六天，忘了上课讲啥了，但是作业么……又是啥也不会的一天

第七天，今天的作业又是啥也不会……这次连谷歌都搜不到了……

第八天，开始设计大作业

第九天，感谢学长的钉钉直播让我终于看懂了纵向联邦学习的代码是在干啥……

建议

上课后来开始听得不够认真，主要在于上课讲的和作业写的基本没啥关系……

数据脱敏的作业完全就是在考验正则表达式和调库，如果连Python都是刚学的话可以直接暴毙了（还好暑假正好熟悉了python）。机器学习的作业如果没接触过ML的话也都是一脸懵逼

其实感觉还是第一天第二天的作业比较有意思，后面的作业和数据市场好像都没啥关系……

个人感觉，比如说数据脱敏，不如也让大家写小作文。第一天的小作文我是查阅了很多的网站写出来的，这个过程中是真正的去探索了什么是数据市场，感受是挺深的。而目前这个数据脱敏作业，完全就是在考正则表达式

同时，对于机器学习的作业，也希望有更详细的指导，全靠自学实在有点顶。总的来说，这几天的感受就是每天都在写作业……

另外这个蚂蚁平台实在是折磨……

最后是感觉课程内容还是挺丰富的，个人不太有兴趣的是公式推导感觉花了太多的时间，数学课即视感（最怕了），然后又和作业没啥关系，导致上课听得就慢慢不太认真了……

最后的大作业可能是希望同学们更多发挥自己的创意，给了大家比较大的自由性，但实际想写好的话工程量实在是有点大，希望明年可以给大家小组合作，或者给一些辅助的框架。全部从0开始，实在有点伤头发

老师人还是很好的，两位助教学长学姐也很棒，都会耐心解答我的疑问，尤其关于平台问题也会帮我找原因，非常感谢！完结撒花！