1. **程序目录结构**（红色文件代表程序运行之后才会生成的文件）：

│ main\_image.py

│ main\_tabular.py

│ black\_box\_attack.py

│

├─raw\_data

│ ├─celebA

│ │ list\_attr\_celeba.csv

│ │ 000001.jpg

│ │ 000002.jpg

│ │ 000003.jpg

│ ├─UTKFace

│ │ 1\_0\_0\_20161219140623097.jpg.chip

│ │ 1\_0\_0\_20161219140627985.jpg.chip

├─data

│ ├─adult

│ │ adult.data

│ │ adult.names

│ │ adult.test

│ │ Index

│ ├─health

│ │ health.csv

│ ├─celebA

│ │ train\_data\_clip.pickle

│ │ test\_data\_clip.pickle

│ ├─UTKFace

│ │ train\_data\_clip.pickle

│ │ test\_data\_clip.pickle

├─models

│ models\_image.py

│ models\_health.py

│ models\_adult.py

├─process\_data

│ load\_image.py

│ load\_health.py

│ load\_adult.py

│ process\_celebA.py

│ process\_UTKFace.py

├─utils

│ GMM\_loss.py

│ gradient\_operations.py

│ trace\_QP.py

│ tools.py

├─infocensor\_utils

│ GMM.py

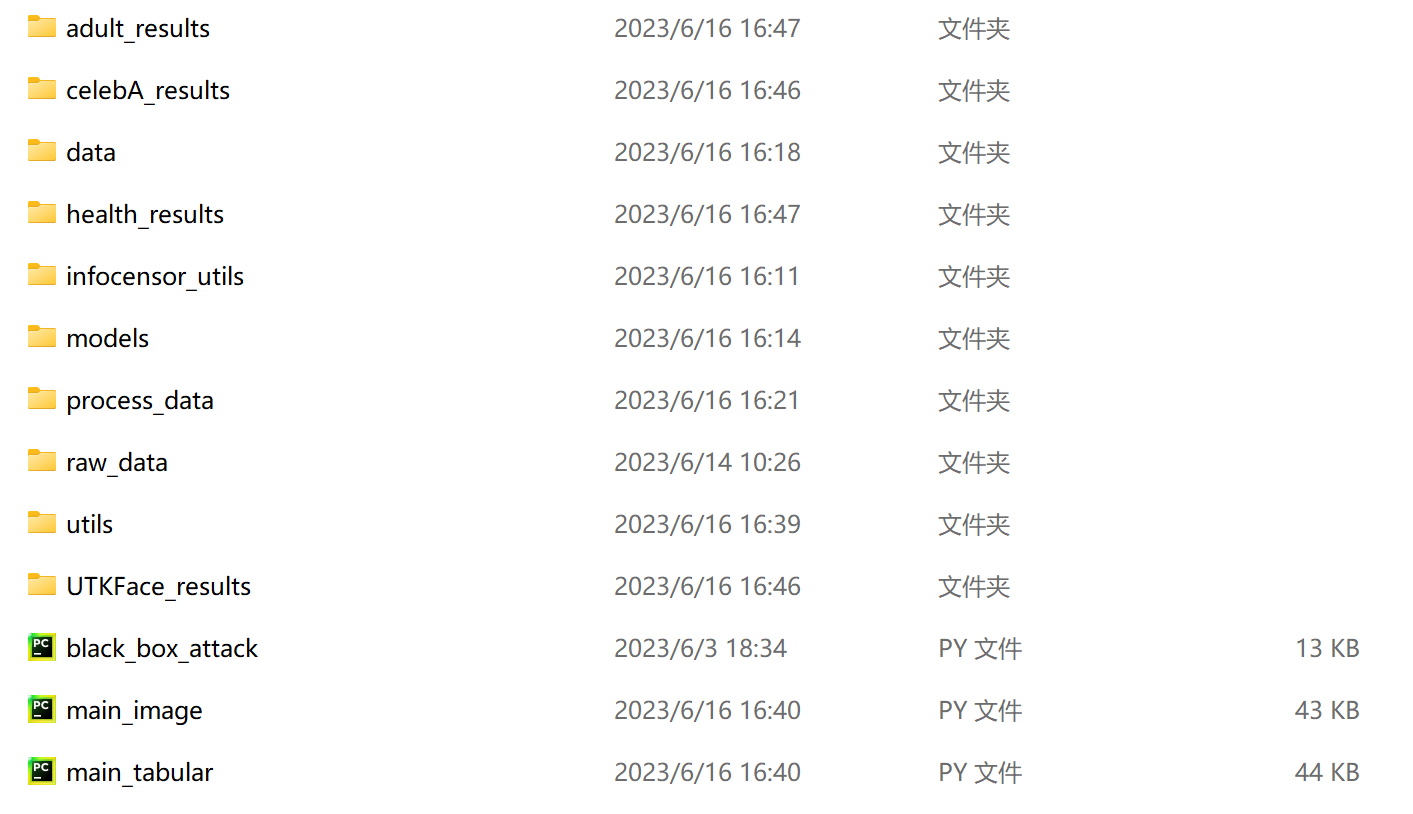
│ utils.py

├─celebA\_results

├─health\_results

├─adult\_results

1. **文件说明：**



**Main\_image.py** 文件是图像数据集的主要运行文件，负责拆分学习的模型训练过程，在训练过程中会同步进行DSA和DPA这两种攻击，其中可以通过参数调整数据集，目标任务和隐私任务，防御方法等，运行产生的结果会自动保存到数据集名称\_results文件夹中

**Main\_tabular.py**文件是表格数据集的主要运行文件，其功能与图像数据集一致

**Black\_box\_attack.py**基本黑盒攻击的运行文件，负责对main文件中生成的用户端模型发起黑盒攻击，其攻击结果保存在main文件生成的结果同目录中

**Raw\_data**文件夹负责保存celebA和UTKFace的原始图片数据，需要先经过处理后才能用于后续训练

**Data**文件夹负责保存处理后的数据文件，其中文本数据的原始数据文件直接存在其中

**Models**文件夹负责保存后续训练所需要用到的模型，两个图像数据集用的模型相同，两个表格数据有各自的模型

**Process\_data**文件夹负责对原始数据进行预处理以及对数据进行加载。其中process\_celebA.py和process\_UTKFace.py文件分别负责将celebA数据集和UTKFace数据的原图片处理为pickle文件。三个load文件则负责读取出数据用于后续训练。

**Utils**文件夹存有代码运行所需要的部分工具文件，其中GMM\_loss.py负责计算隐私损失函数，gradient\_operations.py负责进行部分梯度操作如梯度合并，梯度赋值等。Tools.py中负责一些常用操作比如曲线平滑，构建文件夹，保存图片等。Trace\_QP.py负责解算法中的二次规划问题，得出每一步的前进方向。

**Infocensor\_utils**文件夹保存有infocensor方法中所需的工具文件

**celebA\_results**文件夹保存celebA数据集训练得到的结果，模型等

1. **程序运行顺序**
2. 根据数据类型选择运行main\_image.py或main\_tabular.py
3. 运行black\_box\_attack.py对(1)中得到的模型进行攻击

**注意：**

图像数据运行前需要先按文件夹目录放入所需的原始数据