Machine Learning with Privacy Protections

Integration of AutoML and ARX

# Software Overview

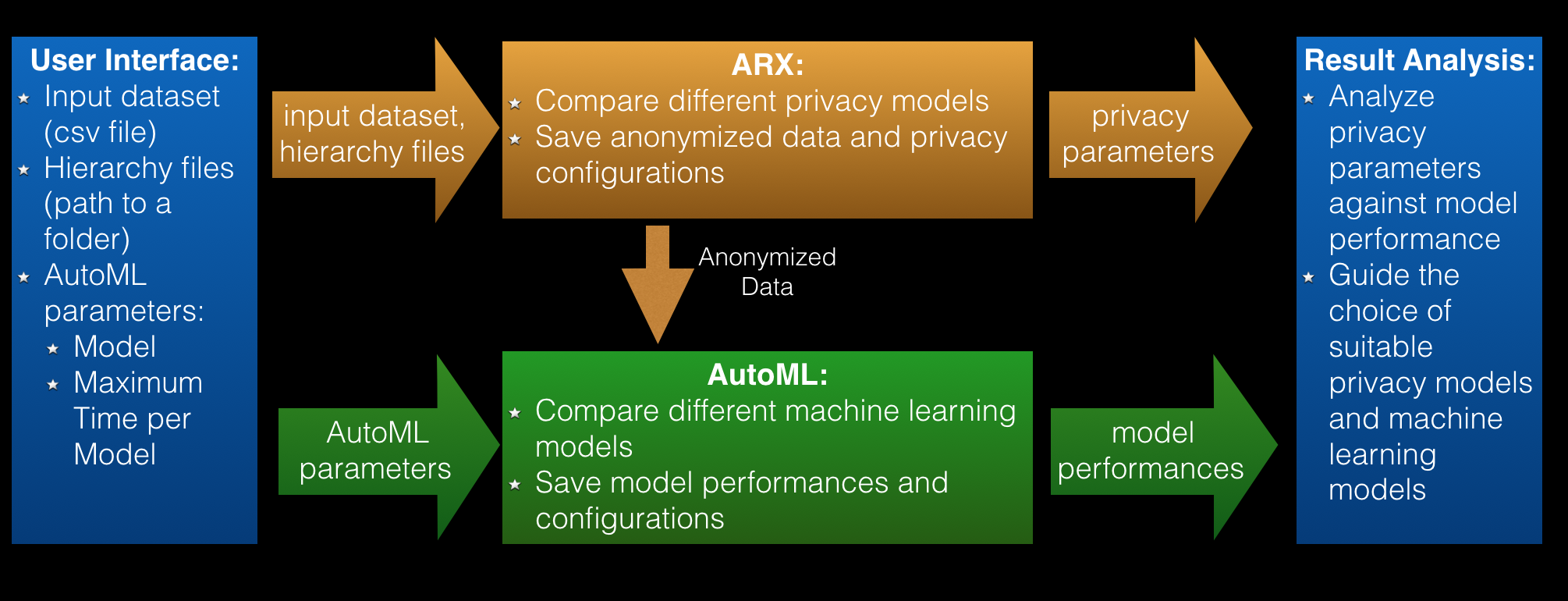
## Introduction

Machine learning techniques can help people get useful information from large datasets. However, if a dataset contains sensitive information, releasing it may result in the leakage of this information. Researchers have shown that removing personal identifiers (PID) cannot prevent re-identifications and attribute disclosures, and have designed different statistical privacy protection techniques for releasing an anonymized dataset.

To apply machine learning with privacy protections, a user has two different decisions to make: to choose the suitable machine learning model and the suitable privacy protection model. This software is designed to help the users on making these decisions. By combining AutoML and ARX, this software allows the user to test a dataset with different combinations of privacy models and machine learning models, and to compare the performances of them.

With this software, users can choose the suitable privacy protection models and machine learning models without the need of being an expert in both fields.

## Data Flow



## User Interface

#### Command Line Interface

For command line user interface of AutoML, please refer: <https://githubenterprise.parc.com/minhas/autoMLSimple>

With the integration of ARX, the command line interface needs 2 additional parameters:

-s Use this flag to indicate that the dataset contains sensitive data.

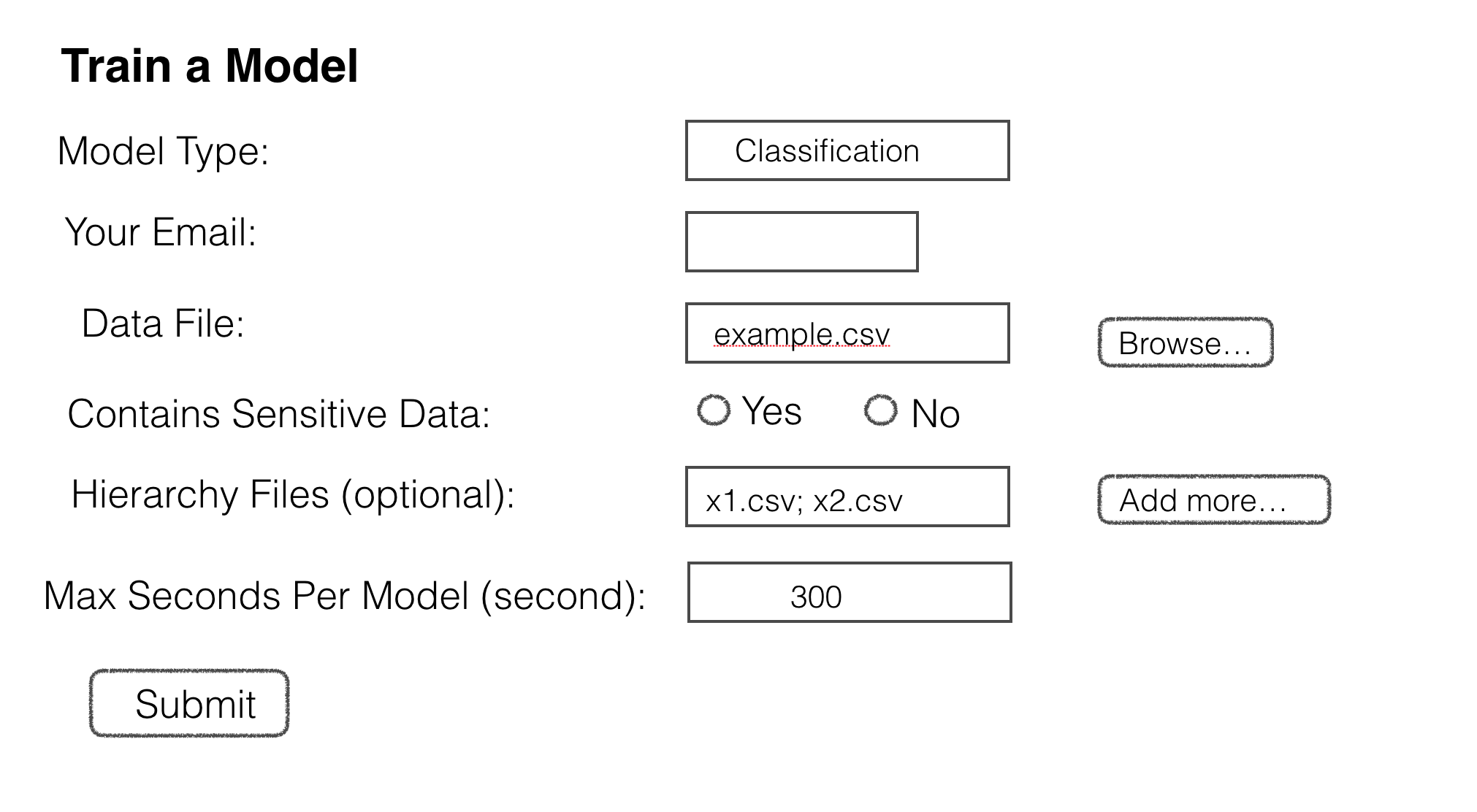
-f Specifies the folder that stores the hierarchy files. This folder will also be used to store the anonymized data files generated by ARX.

For example, the following command will run AutoML classification models on “data/example.csv” file, with privacy protections:

autoML.py –m classification –i data/example.csv –t 300 –s –f data

#### Graphical User Interface

In the graphical user interface, the user needs to upload the input data and the hierarchy files, input the parameters for AutoML (i.e., model type and max seconds per model), and specify whether the dataset contains sensitive data. To get the result report, the user needs to provide an email address.



#### Hierarchy Files

To protect sensitive information with generalization techniques, user needs to provide the hierarchy specifications of the columns to be protected. There are 3 ways to specify hierarchies:

1. Use hierarchy files. A hierarchy file is a “;”-separated csv file that specifies the generalization of each hierarchy. A hierarchy file should have the same name as the corresponding column name in the input dataset. Here is an example of the hierarchy file for column “marital-status\_\_cat”:

File Name: marital-status\_\_cat.csv

Content:

|  |  |
| --- | --- |
| Married-civ-spouse;Spouse present;\* | |
| Divorced;Spouse not present;\* |  |
| Never-married;Spouse not present;\* | |
| Separated;Spouse not present;\* |  |
| Widowed;Spouse not present;\* |  |
| Married-spouse-absent;Spouse not present;\* | |
| Married-AF-spouse;Spouse present;\* | |

1. Use ARX graphical user interface. In the ARX graphical user interface, user can add the hierarchy specifications manually. Please refer to ARX for detailed instructions.
2. Use default hierarchies. If the hierarchy for a protected column is not specified, the software will use default hierarchies: the column is set to either the true value, or the value “\*”. Using default hierarchies may result in lower utility of the anonymized dataset.

#### Result Report

Following is an example of the report returned by the software.

Results:

Privacy Criterion: None

        Number of models: 7

        Models: ['RandomForest: 0.816600', 'ExtraTrees: 0.811245', 'KNeighbors: 0.811245', 'QDA: 0.808568', 'AdaBoost: 0.797858', 'DecisionTree: 0.797858', 'GaussianNB: 0.784471']

        Ensemble Confusion Matrix (based on majority votes of top 5 models):

[[525  46]

[ 90  86]]

             precision    recall  f1-score   support

          0       0.85      0.92      0.89       571

          1       0.65      0.49      0.56       176

avg / total       0.81      0.82      0.81       747

Privacy Criterion 1: 2-anonymity

Average Re-identification Risk: 0.010050251256281407

Highest Re-identification Risk: 0.5

        Number of models: 7

        Models: ['KNeighbors: 0.765730', 'AdaBoost: 0.753681', 'DecisionTree: 0.753681', 'ExtraTrees: 0.753681', 'LDA: 0.753681', 'QDA: 0.753681', 'GaussianNB: 0.736278']

        Ensemble Confusion Matrix (based on majority votes of top 5 models):

[[572   0]

[175   0]]

             precision    recall  f1-score   support

          0       0.77      1.00      0.87       572

          1       0.00      0.00      0.00       175

avg / total       0.59      0.77      0.66       747

Privacy Criterion 2: 5-anonymity

Average Re-identification Risk: 0.008040201005025126

Highest Re-identification Risk: 0.2

        Number of models: 7

        Models: ['AdaBoost: 0.799197', 'ExtraTrees: 0.799197', 'LDA: 0.772423', 'QDA: 0.765730', 'DecisionTree: 0.759036', 'KNeighbors: 0.736278', 'GaussianNB: 0.716198']

        Ensemble Confusion Matrix (based on majority votes of top 5 models):

[[517  50]

[100  80]]

             precision    recall  f1-score   support

          0       0.84      0.91      0.87       567

          1       0.62      0.44      0.52       180

avg / total       0.78      0.80      0.79       747

Privacy Criterion 3: (0.01)-avg-reidentification-risk

Average Re-identification Risk: 0.008710217755443886

Highest Re-identification Risk: 1.0

        Number of models: 9

        Models: ['AdaBoost: 0.772423', 'DecisionTree: 0.772423', 'ExtraTrees: 0.772423', 'GradientBoost: 0.772423', 'LDA: 0.772423', 'QDA: 0.772423', 'RandomForest: 0.772423', 'KNeighbors: 0.742972', 'GaussianNB: 0.646586']

        Ensemble Confusion Matrix (based on majority votes of top 5 models):

[[571   3]

[167   6]]

             precision    recall  f1-score   support

          0       0.77      0.99      0.87       574

          1       0.67      0.03      0.07       173

avg / total       0.75      0.77      0.68       747

# Development Plan

Features to be implemented are highlighted.

## [Part I] Modifications on ARX (Completed)

**TODO:**

Currently, ARX decides the csv escape character depends on the version of system you are running. If the file is not generated on your current system, the program may return some error message. You may need to resave that file on your current system to make it work.

To solve this problem, modify getOs() function in Main.java to check for the format of the file instead of the current system.

### Developemnt Environment

1. The source code is available at <https://githubenterprise.parc.com/abrito/fhwa-privacy/tree/master/arxUI> .
2. This software requires two input parameters: the path to the input file, and the path to a folder for hierarchy files and ARX outputs.
3. This is the ARX part of the AutoML+ARX integration software. This part can be tested independent of AutoML with the following command:

**java –jar arx.jar [path\_to\_folder] [path\_to\_file]**

For example: **java arx.jar data data/example.csv**

(to have it work on Mac OS, use **java –XstartOnFirstThread –jar arx.jar [path] [example.csv]** instead)

For detailed instructions, please refer to the ReadMe.txt file under the arxUI directory.

1. This software is modified based on ARX version 3.4.1 (Download at <http://arx.deidentifier.org/?ddownload=1922>) .
2. This software is tested under Java(TM) SE Runtime Environment (build 1.8.0\_92-b1) (Download at <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html> ).
3. This software is developed under Java Development Kit 1.8.0\_92 (Download at <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> ).

### New Features (compared to the original version of ARX)

1. Pop up a window with instructions on how to use ARX
2. ARX automatically loads the input dataset when opened
3. ARX automatically loads the hierarchy files when opened
4. Allow user to save the anonymized data to the folder provided by user, the data is saved in the format compatible with AutoML
5. When user saves the anonymized data, ARX automatically saves the privacy configurations
6. When user exits ARX, pop up a window to remind the user to save anonymized data
7. Add brief instructions and recommended settings for each privacy criterion.

### Implementations

* 1. Class Name: **Main.java** (org.deientifier.arx.gui)
* Method Name: **main()** Status: Changed
* If there are more than two command line arguments, and the second one ends with “.csv”, call the loadFile() method to create a new project and load the input file.
* If the argument path doesn’t end with “\” or “/”, add “/” “\”
* Method Name: getOS() Status: New Method
  + Check for the current operating system.
  1. Class Name: **MainWindow.java** (org.deientifier.arx.gui.view.impl)
* New Attributes:
* **String outPath** – the path to store output files
* **String outSpec** – the path to the output specification file “output.txt”
* **int savedCnt** – Count of saved output files
* Method Name: **setOutPath(String outPath)** Status: New Method
* Store the [path] given by AutoML into the outPath variable
* Create the “output.txt” file under [path]
* Method Name: **getOutPath()** Status: New Method
* Return the outPath variable
* Method Name: **getSavedCnt()** Status: New Method
* Get the count of saved output files
* Method Name: **increaseSavedCnt()** Status: New Method
* Increase the count of saved output files by one
* Method Name: **getOutSpec()** Status: New Method
* Return the path to the output specification file "output.txt"
* Method Name: **getMenuFile()** Status: Changed
* Added new item in Menu: “File – Export to AutoML”
* The button only shows when the model has output, the window under the tab “analysis”, and the outPath variable is not empty (“”)
* Method Name: **getHelpMenu()** Status: Changed
* Added new item in help menu: “Help – AutoML Instructions”
* Method Name: **showAutoMLDialog()** Status: New Method
* Create a dialogue for AutoML instructions
  1. Class Name: **Controller.java** (org.deientifier.arx.gui)
* Method Name: **actionMenuFileExit()** Status: Changed
* Changed the choice items in the exit pop up window
* Added new choice: “save to AutoML and exit”
* When the user click save, save the output files to AutoML
* Method Name: **actionMenuFileExportDataToAutoML()** Status: New Method
* The method called when user clicks the button “Export to AutoML”
* Create a new WorkerAutoMLSaver class to save the outputs and privacy configurations
* Method Name: **increaseSavedCnt()** Status: New Method
* Increase the savedCnt variable in class MainWindow.java
* Method Name: **actionMenuFileLoad()** Status: New Method
* Create a new project
* Load the [example.csv] file into the new project
* If there is any file with the name [column\_name.csv] under [path] folder, load these files as hierarchy files, and set the corresponding columns to be quasi-identifying
* Pop out instructions on how to use AutoML
* Method Name: actionMenulHelpAutoML() Status: New Method
  1. Class Name: **WorkerAutoMLSaver.java** (New Class) (org.deientifier.arx.gui.worker)
* Save the privacy configurations and output files
  1. Class Name: **DialogAutoMLInfo.java** (New Class) (org.deidentifier.arx.gui.view.impl.menu)
* A dialogue for AutoML instructions (similar to DialogAbout.java class)
  1. Class Name: EditorCriterion.java (org.deidentifier.arx.gui.view.impl.menu)
* Method Name: createInstructions() Status: New Method
* Create a label with explanations for the privacy model
  1. Class Names: **EditorCriterionDifferentialPrivacy.java, EditorCriterionDDisclosurePrivacy.java, EditorCriterionDPresence.java, EditorCriterionKAnonymity.java, EditorCriterionKMap.java, EditorCriterionLDiversity.java, EditorCriterionRiskBased.java, EditorCriterionTCloseness.java** (org.deidentifier.arx.gui.view.impl.menu)
* Method Name: build() Status: Changed
* Added one line to construct the instruction label
  1. File Name: **messages.properties** (org.deientifier.arx.gui.resources)
* New Messages: AutoML.1 – AutoML.13
* Different pop up messages used to link ARX and AutoML

## [Part II] Modifications on User Interface

### Modifications on Graphical User Interface

1. Add the option to specify whether the dataset contains sensitive data.
   1. If the dataset contains sensitive information,
      1. Create a folder on the server to store hierarchy files and ARX outputs
      2. Run ARX with the following two input parameters: the path to the input data, and the path to the folder created in the previous step

e.g. **java –jar arx.jar [path\_to\_folder] [path\_to\_file]**

* 1. If the dataset does not contain sensitive information, run the original version of AutoML

1. Add the option to upload hierarchy files for datasets containing sensitive data
   1. Hierarchy files can only be uploaded if the dataset contains sensitive data
   2. Only csv files can be uploaded
   3. Store all the uploaded hierarchy files under the folder crated in Step 1)a.i.

### Modifications on Command Line User Interface

1. Add parser for flag “-s”. If “-s” is used, the dataset contains sensitive data.
   1. If “-s” is used with “-f”, run ARX with the following two parameters: path to the input data and path to the folder following “-f” parameter

e.g. **java –jar arx.jar [path\_to\_folder] [path\_to\_file]**

(to have it work on Mac OS, use **java –XstartOnFirstThread –jar arx.jar [path\_to\_folder] [path\_to\_file]** instead)

* 1. If “-s” is used without “-f”, create a new folder. Then, run ARX with the following two parameters: path to the input data and path to the created folder.

e.g. **java –jar arx.jar [path\_to\_folder] [path\_to\_file]**

(to have it work on Mac OS, use **java –XstartOnFirstThread –jar arx.jar [path\_to\_folder] [path\_to\_file]** instead)

* 1. If “-s” is not used, run the original version of AutoML

1. Add parser for flag “-f”. Usage: “-f [folder\_name]”.
   1. Check whether the string following “-f” is a valid folder name and store the folder name, otherwise return error message

## [Part III] Result Analysis

### Analyze anonymized datasets with AutoML

Write a script to analyze all the ARX output datasets with AutoML.

1. Output files of ARX:

**output.txt** – Stores the filenames of ARX’s output datasets. This file is reset every time the user opens a new ARX program.

**1.csv, 2.csv** … – The output data files after anonymization. When the user exports a new output data file, a new file is stored, and the file name is added to **output.txt**.

**1\_config.txt, 2\_config.txt** … -- The privacy configurations of the corresponding output files.

1. Pseudocode:

file = open\_file([folder\_name]/output.txt)

for each line in file

input\_data = open\_file(line)

run AutoML algorithms on input\_data, and save results

end for

### Integration of AutoML results and ARX privacy configurations

Combine the report returned by AutoML with the privacy configurations. The privacy configurations are stored as n\_config.txt, where n is the index of the output data.

For example, for output data 1.csv, the privacy configuration is stored in 1\_config.txt