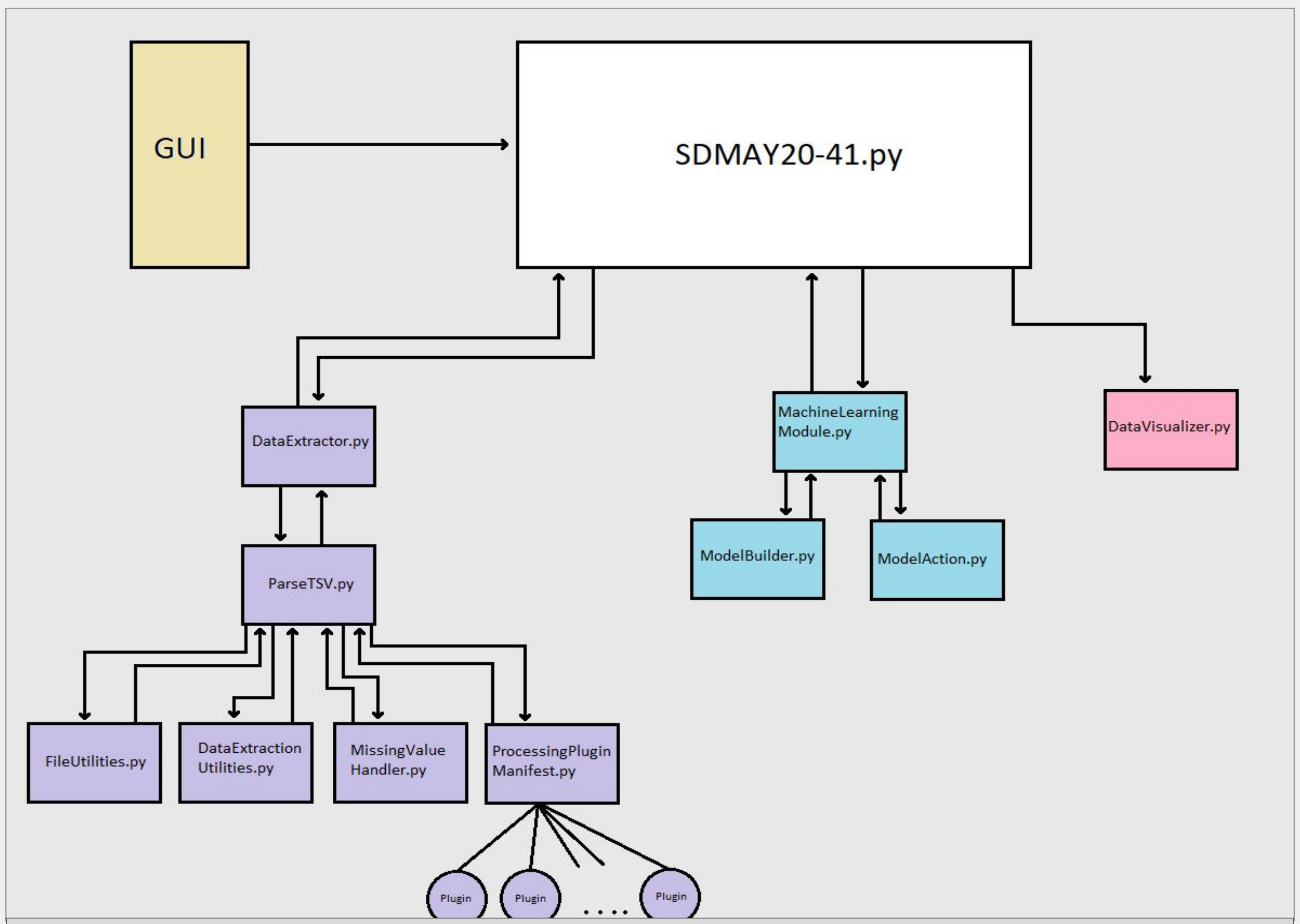
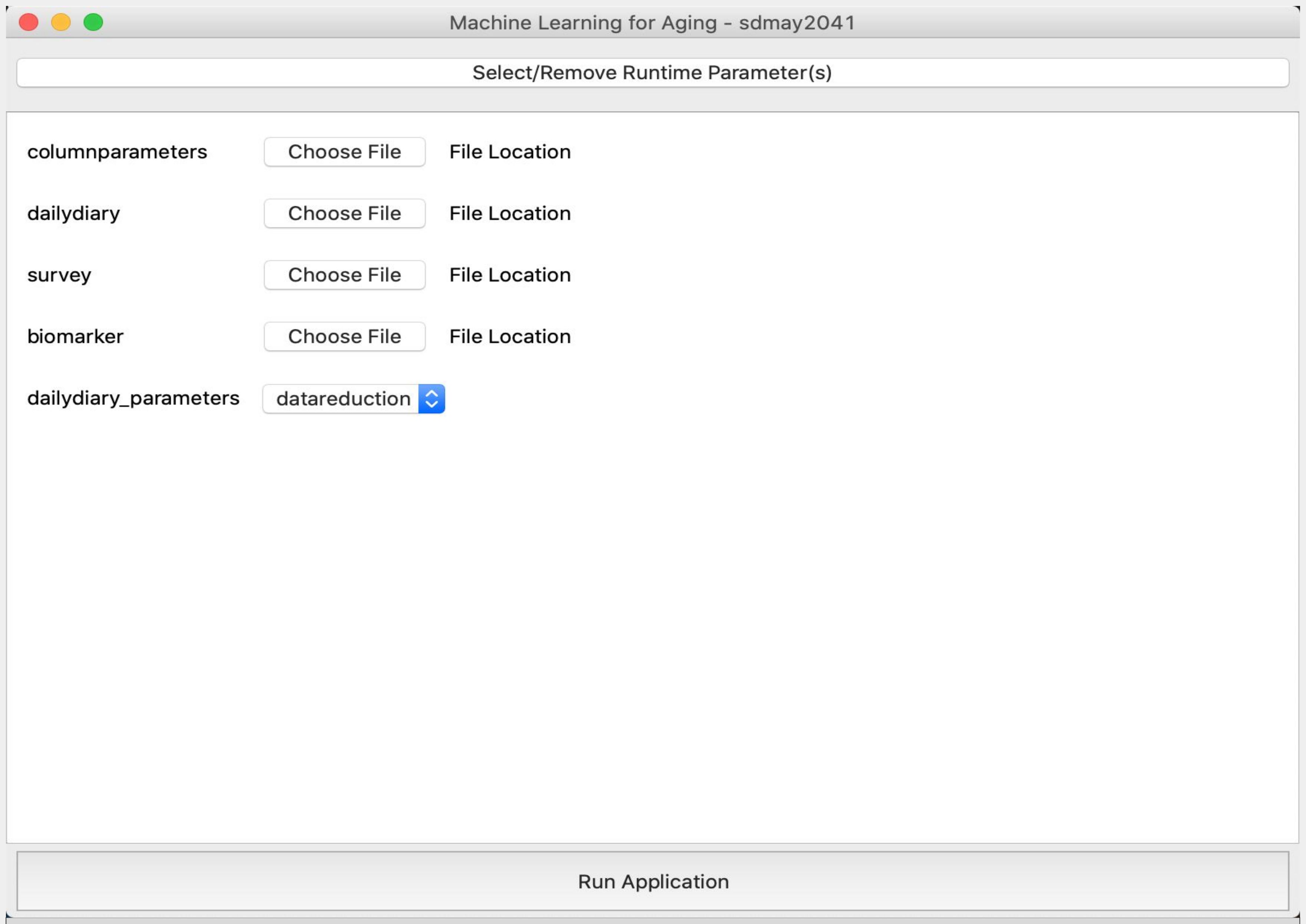
Machine Learning for Understanding Aging



Detailed design of the product


Machine Learning for Aging - sdmay2041
Select/Remove Runtime Parameter(s)
Choose File
Choose File
Choose File
Choose File
Run Application

columnparameters File Location

dailydiary File Location

survey File Location

biomarker File Location

dailydiary_parameters datareduction

User Interface

Introduction/Motivation

Problem Statement

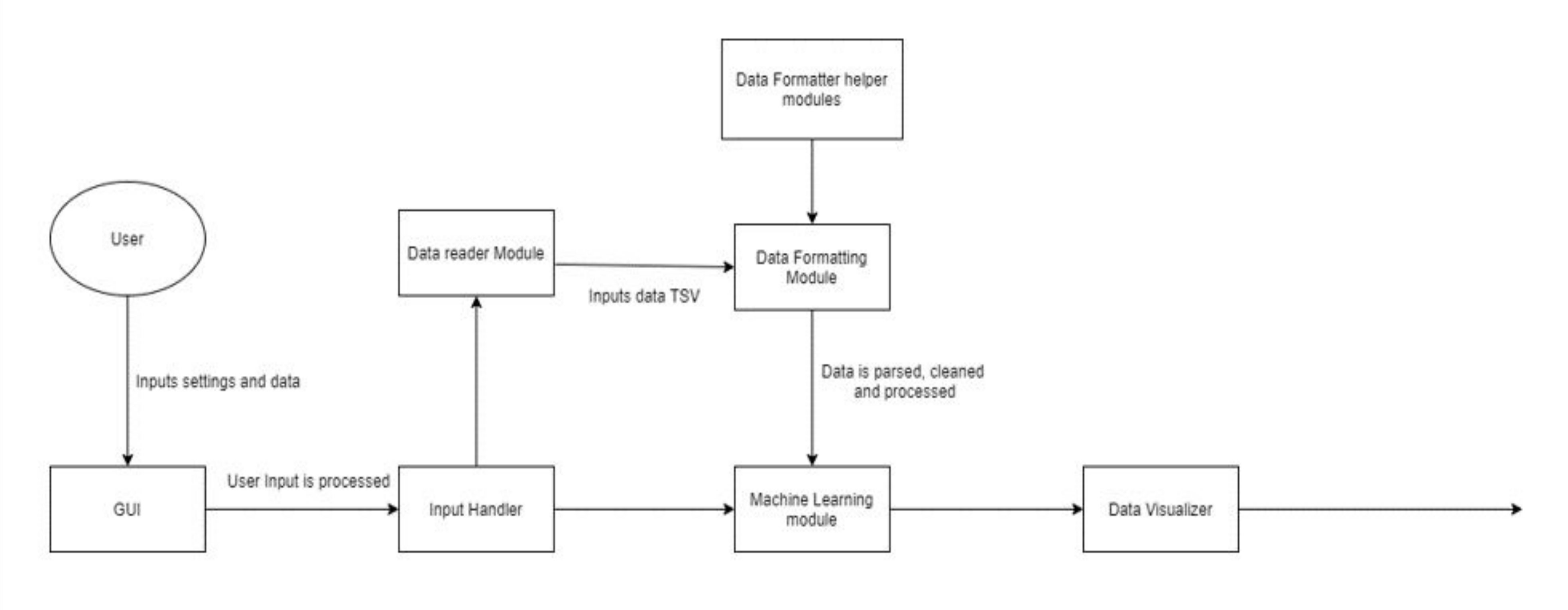
Human aging is a topic that has been studied throughout history. Scientists, doctors, sociologists, and the general public all want to know what characteristics indicate a decline in health and what actions can be taken to slow down this decline. Knowing this information can allow people to increase their life expectancy and overall quality of life. Creating a tool that would assist the study of this topic could lead to a faster discovery of what leads to a higher quality of life later in life.

Solution

Using health data collected by ICPR within the study Midlife in the United States (MIDUS) project, create a tool that will help gerontologists analyze their data using machine learning based observations about aging.

Engineering Standards & Design Practices

1. The product must ensure the privacy of the people whose data we are using to use in the creation of our machine-learning program.
   a. All Personal Health Information (PHI) must be anonymized.
   b. Any PHI that is transmitted must be encrypted.
2. The product must be accessible, and the information output must be easily understandable.
3. The product must be fast to learn, and up to today's standards of machine learning.
4. The product must output results that are accurate so that others can use the data we obtain through our program with reliability.
5. The product must be created considering the knowledge that we have learned from taking the *CITI Program's Social/Behavioral Research Course*[7].



Conceptual design of the product

Testing & Test Plan

Testing Environment

Test Runner
- Test runner is ran by the developer before any merge request is made
- Runs all tests using one command
- Ensures that the merge requests don't break previously-working functionality

GitLab Continuous Integration (CI) / Pipeline
- GitLab has a built-in CI tool that can be used by making a ".gitlab-ci.yml" file in the repository.
- This tool compiles/builds our project and then runs the test runner to ensure no breaking tests
- Pipeline runs on all branches

Testing Strategy

Functional
- Unit Testing
  - Add tests for each merge request that adds functionality
- Integration Testing
  - Recreate "Affective Reactivity to Daily Stressors is Associated with Elevated Inflammation" study.
- System Testing
  - Ensure all parts are working as intended by using the program hands-on.
- Acceptance Testing
  - Weekly Meeting

Non-Functional
- Performance Testing
  - Testing with big and small data.
  - Ensuring test times are relatively low.
- Compatibility Testing
  - Testing on multiple virtual machines with different operating systems
    - Windows 7+
    - Linux
    - MacOS
- Usability Testing
  - Hands-on testing by developers to determine the ease-of-use

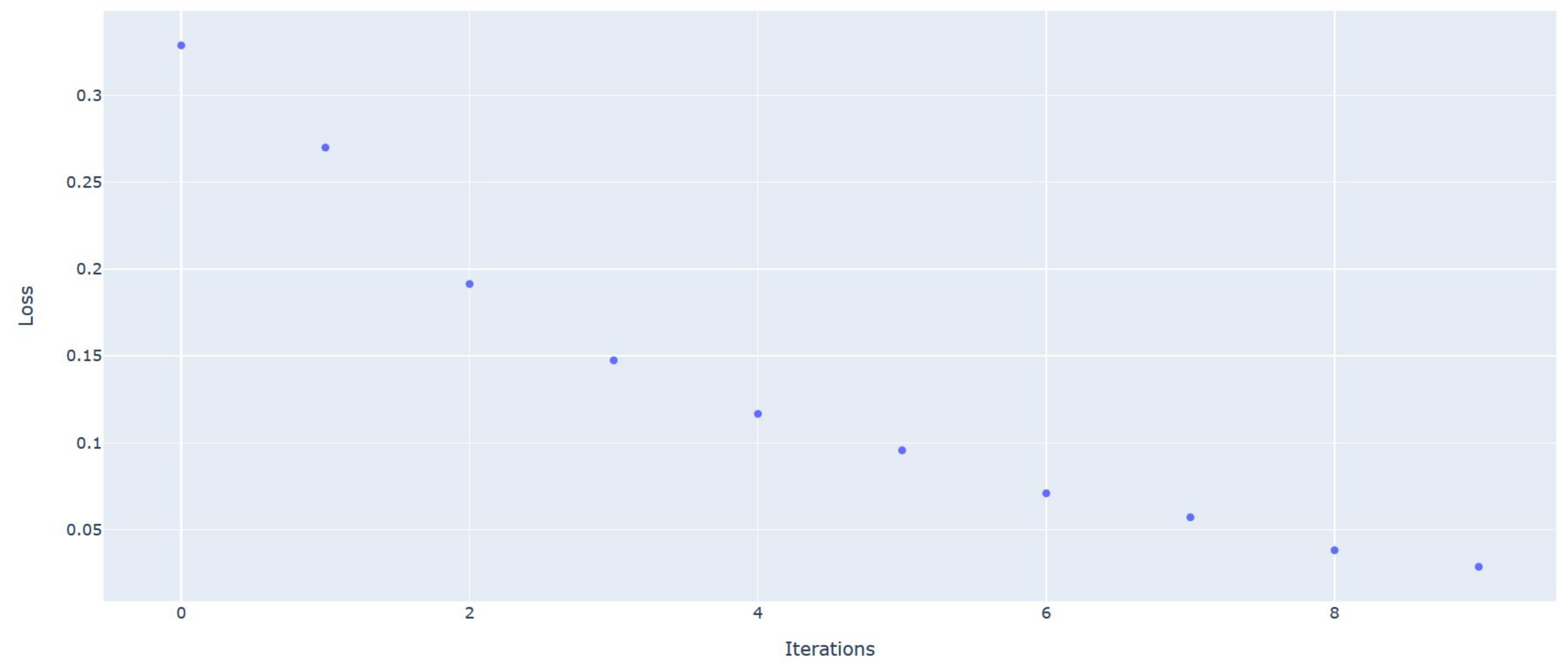
Design Requirements

Functional Requirements

- The program accurately assesses patterns in data related to the MIDUS Datasets.
- Users can continue to input data to increase the accuracy of the program.
- The program outputs aspects of the input data that affect the experience of aging.
- Project completed by May 2020.
- The program must have extension points that will allow users to add their own data processing methods.

Non-Functional Requirements

- Written in Python
- Clear, well-documented code.
- Privacy of the subjects included in test data must be considered.
- Appropriate size of training data is used to properly train the program.
- The results of the running program are outputted in a user friendly format.



Example of graph displaying decreasing mean squared error values for each iteration of the machine learning component. This represents the machine learning component verifying associations between the input data and the comparison data

Intended Users

The program is meant for gerontologists or scientologists who want to verify their data findings using machine learning. The user can specify characteristics of the program to improve the machine learning run if they choose to.

Technical Details

Technical Stack

Language
- Python 3.7.4

CI/CD Pipeline
- GitLab Continuous Integration

Machine Learning Framework
- TensorFlow 2.0.0

GUI Libraries
- Plotly 4.6.0
- PyQt5 5.14.2

Modules

GUI
The program features a GUI that assists users in selecting their runtime parameters for the program, and displays the output of the program into graphs.

Data Extraction
The datasets that this program is designed to run with have thousands of columns and tens of thousands of rows. The Data Extraction modules cuts the massive amount of input data, and normalizes it into a format that is usable by our machine learning module.

Machine Learning
Our machine learning module performs a lasso regression on the data passed to it from the Data Extraction module. The results of the lasso regression is then passed to the GUI to be represented in a graph.

Team Information

Project Identifier / Team

sdmay20-41

Team Members / Roles

1. Aria Sheets	1. Report Manager
2. Ian Simon	2. Chief Engineer
3. Jacob Laing	3. Chief Engineer
4. Nathan Carter	4. Test Engineer
5. Samantha Williams	5. Meeting Scribe
6. Scott Rose	6. Meeting Facilitator

Client / Advisor

Dr. Julie Dickerson