Machine Learning for Understanding Aging

Team: sdmay20-41

Website: http://sdmay20-41.sd.ece.iastate.edu/

Aria Sheets / Report Manager
Ian Simon / Chief Engineer
Jacob Laing / Chief Engineer
Nathan Carter / Test Engineer
Samantha Williams / Meeting Scribe
Scott Rose / Meeting Facilitator

Client/Advisor: Dr. Julie Dickerson

Overview

- Goal:

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

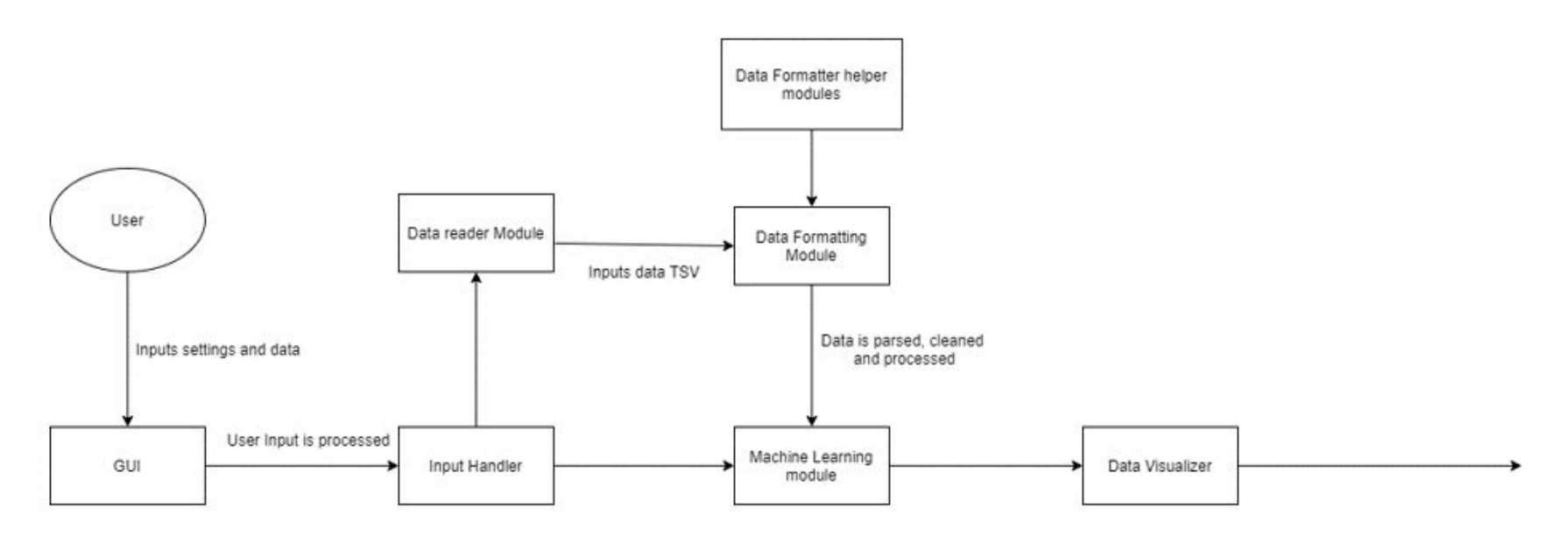
- Target Audience:

  We hope this tool will be used to help gerontologists create machine learning proof of concept observations using the MIDUS data set without needing to know how to create a machine learning model from scratch.

Problem Statement

Through this project, we want to create a tool that would help gain insight into increasing life expectancy and overall quality of life.

Conceptual Sketch



Functional Requirements

- The program accurately assesses patterns in data related to aging.
- Users can continue to input data to increase the accuracy of the program.
- Users can implement their own modifications to our program that affect how their data is read.
- The program outputs aspects of the input data that affect the experience of aging.

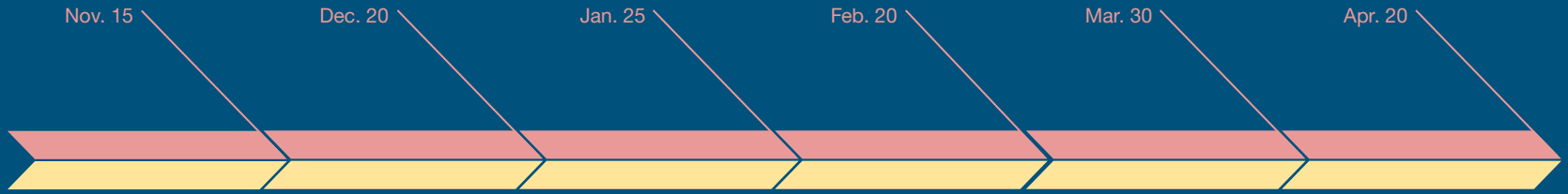
Constraints & Considerations

- Project needs to be completed by May 2020.
- The data that is inputted into our program must be in a specific format or the program won’t run correctly.
- The datasets we use must be anonymized to protect the personal information of the subjects.
- Speed of the program is dependent on how fast the user’s machine is.

Potential Risks & Mitigation

- Risk 1: Inexperience with machine learning.
  - Mitigation: Spend time studying machine learning before we begin programming. Practiced machine learning tutorials and read various papers provided by our faculty advisor prior to implementing our project.
- Risk 2: Working with personal health data.
  - Mitigation: We used MIDUS data set which has fully anonymized data. Gained certification from Collaborative Institutional Training Initiative (CITI) on the use of personal data in research studies.
- Risk 3: Inconsistency in data formatting.
  - Mitigation: Implemented a plugin architecture that will allow users to modify the way data is read into the program. Additionally included several parameters for user to interact with to change the behavior of the program without having to do any additional implementations.

Project Timeline



Inception

Requirements are gathered for the project and research is done.

Design

The project is fully conceptualized and tested.

Data Preparation & Parser

We prepare our data for being parsed through. We then develop the data parser so that we can traverse the MIDUS data

Data Formatter

We develop a data formatter to take in the parsed data and output it in a way that can be input into Tensorflow

Machine Learning Module

We develop our machine learning module that reads in the formatted data.

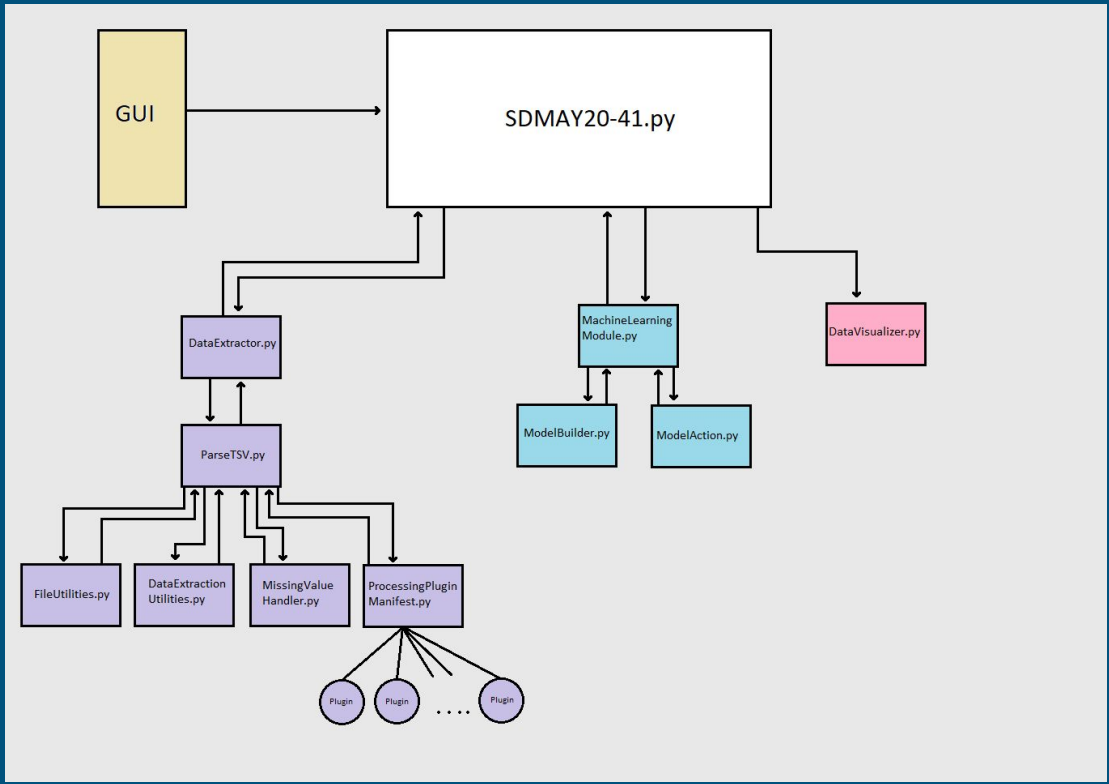
Data Visualizer, GUI, & Completion

We create the data visualizer and GUI that allows our program to be easily operated by scientists and researchers.

Functional Decomposition

1. GUI
    a. Create a user-friendly GUI to input data and select runtime parameters
2. Data Preparation & Data Parser
    a. Parsing multiple MIDUS TSV data files
3. Data Formatter
    a. Format data in such a way so that its easily usable
    b. Must have the ability for users to add their own preprocessing methods
    c. Must be able to be used by TensorFlow
4. Machine Learning Module
    a. Set up machine learning training
    b. Test machine against new data
5. Data Visualizer & Completion
    a. Create a data visualizer in Plotly
    b. Project is completed

Detailed Design



Hardware, Software, & Technology Platforms

- Language
  - Python 3.7.4
- Continuous Integration
  - GitLab
- Machine Learning Framework
  - TensorFlow 2.0.0
- Project Tracking
  - Trello
- Data Visualization Tool
  - Plotly 4.6.0
- GUI Library
  - PyQt5 5.14.2
- Testing
  - Python “unittest” library

Test Plan - Functional & Non-Functional

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 us developers to determine the ease-of-use of the program.

Test Plan - Continuous Integration (CI) / Pipeline

Test Runner

- Project has a test runner that 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 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
- Never merge a branch into master if the pipeline is failing

Integration Test

- Affective Reactivity to Daily Stressors Is Associated With Elevated Inflammation
  - Links a person’s stress reactivity to their inflammation levels

Integration Test

Positive Affectivity

Column Name	Column Description
B2DC7	IN_GOOD_SPIRITS
B2DC8	CHEERFUL
B2DC9	EXTREMELY_HAPPY
B2DC10	CALM_AND_PEACEFUL
B2DC11	SATISFIED
B2DC12	FULL_OF_LIFE
B2DC21	CLOSE_TO_OTHERS
B2DC22	LIKE_YOU_BELONG
B2DC23	ENTHUSIASTIC
B2DC24	ATTENTIVE
B2DC25	PROUD
B2DC26	ACTIVE
B2DC27	CONFIDENT

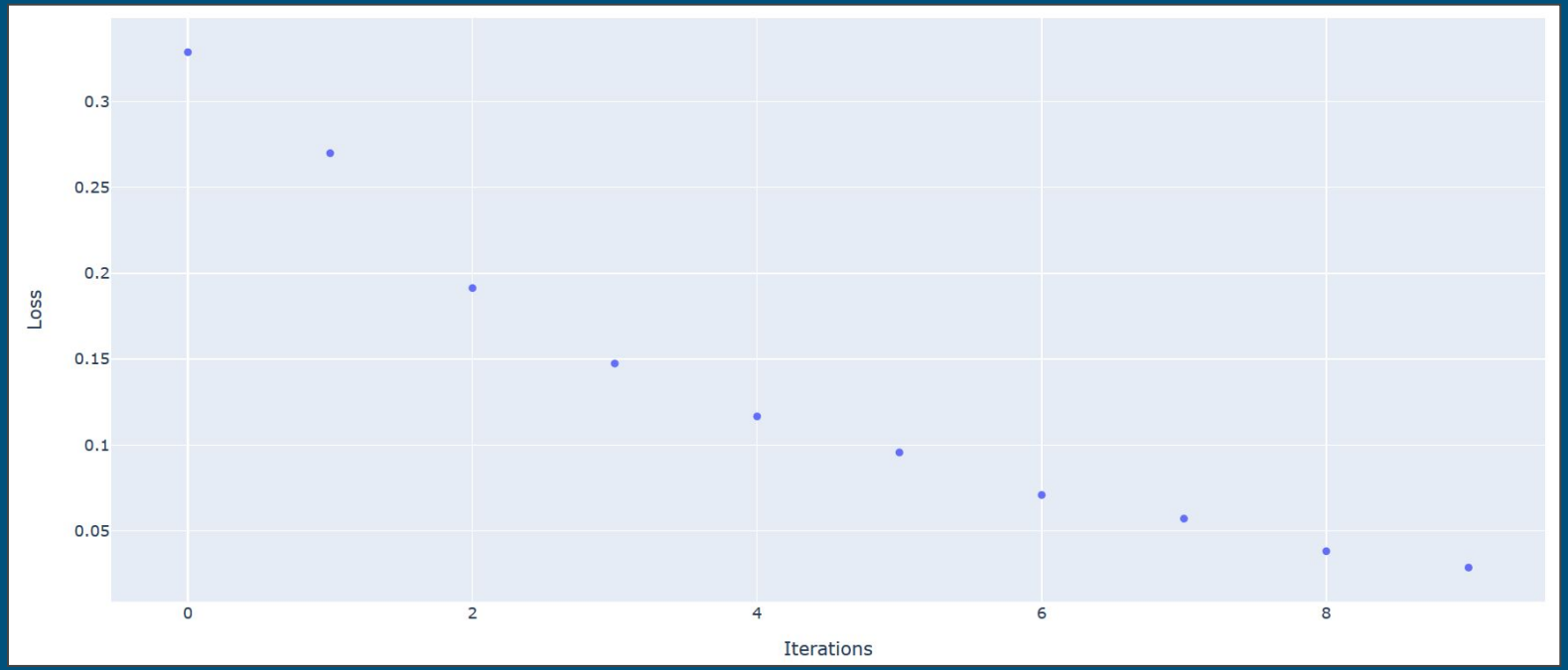
Negative Affectivity

Column Name	Column Description
B2DC1	RESTLESS_OR_FIDGETY
B2DC2	NERVOUS
B2DC3	WORTHLESS
B2DC4	SO_SAD_NOTHING_COULD_CHEER_YOU_UP
B2DC5	EVERYTHING_WAS_AN_EFFORT
B2DC6	HOPELESS
B2DC13	LONELY
B2DC14	AFRAID
B2DC15	JITTERY
B2DC16	IRRITABLE
B2DC17	ASHAMED
B2DC18	UPSET
B2DC19	ANGRY
B2DC20	FRUSTRATED

Inflammation

Column Name	Description
B4BSIL6R	Blood Serum Soluble IL6 Receptor (pg/mL)
B4BCRP	Blood C-Reactive Protein (ug/mL)

Integration Test



Prototype Implementations

Reformatting/Reducing the MIDUS dataset.

- Compared the different files and narrowed the data down using only data sets associated with IDs present in all three files.
- User specifies columns of interest to use in the machine learning program, as well as how they want to deal with missing/unanswered values
- User can handle missing or incomplete data.
- Outputs a data array for the machine learning component.
- Prototype was successfully implemented

Prototype Implementations

Tensorflow component

- Takes in health data, labels and user input information.
- User will be able to train their own Machine Learning algorithm or use one previously trained.
- The machine learning algorithm will use the data to predict certain health attributes specified by the user.

Machine Learning

- Used Keras to develop the machine learning algorithm.
- Data was split into training and testing subsets at a 9:1 ratio
- The training data was further split into 10 equal parts
- The machine learning algorithm is then trained once on each training set.

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

Project Demo

Click to start the video

Future Prospects of the Project

Our highly modifiable code has left options open for future expansion of the project:

- Users can add and implement new processing methods and parameters with relative ease to fit their needs.
- Expansion to other sets of data beyond MIDUS.
- Allow for users to modify the machine learning module.

Task Contributions of Each Member

- Nathan Carter: Researched Machine learning. Researched python data visualization tools. Developed user handler helper component allowing the user to decide what that want to do with missing or incomplete data. Helped develop methods for the data extraction component. Helped develop tests and spent time debugging code. Developed the Machine learning component with Tensorflow. This includes splitting the data up into testing and training subsets, and developing the neural network with Keras.
- Jacob Laing: Created a function within the data formatter to take the parsed data and combine it into a singular array to output. Wrote tests for that function and for the DataExtractor class that pulls the data from the TSV files. Created a user friendly GUI to make the program easy to run for everyone.
- Scott Rose: Researched Ridge Regression, Created Components for the Data Parser and Formatter, Helped configure the team VM, Created experiments to test TensorFlow and SciKit Learn, Created a test plan for our project. Created the initial framework for the project. Created the testing framework for the project. Contributed to the data extraction modules. Created unit tests for data extraction module. Created the plugin system for data extraction. Created a data processing plugin for the system test of our project. Created a write up for the system test we conducted.

Task Contributions of Each Member cont.

- Aria Sheets: Set up Trello and Slack as communication and project progress for the project. Keeps track of the documents our team makes. Parsed through the MIDUS codebooks to retrieve min and max values of data into the form of a TSV file. Researched into and developed the GitLab continuous integration pipeline. Worked on incorporating the Plotly component.
- Ian Simon: Contacted MIDUS team to gain information on how to read their DDI files. Programmed processExtractedColumns which normalizes data for use of the machine learning component. Worked on adding parameters to the data formatter in order to properly structure data for input to the machine learning module. Implemented functionality for how the program handled the missing data cells.
- Samantha Williams: Worked on the importing and storage of data into our objects and methods. Created ColumnParameters class. Created MetaData singleton class to keep track of run time specifications. Tested MetaData and extractWantedColumns. Formatted the output of the program. Added parameters for method of reduction of the data. Added the plotly scatter plot for visual output component.

Thank You