

1. Home	2
1.1 Handover Guide	4
1.2 Development	5
1.2.1 Development Process	6
1.2.2 Quality Assurance Guidelines	8
1.2.3 Frontend Workspace Structure Proposal	9
1.2.4 Set-up SSO Authorisation Guide	14
1.2.5 Deployment Process	15
1.3 Specifications	16
1.3.1 Project Description	17
1.3.2 Motivational Model	19
1.3.3 Personas	21
1.3.4 User Stories	28
1.3.5 Prototype	31
1.3.6 Test cases	33
1.3.7 Acceptance Criteria	40
1.4 Sprint Artefacts	41
1.4.1 Final Presentation	42
1.4.2 Sprint Plan	43
1.4.3 Client Communications	45
1.5 Resources	52
1.5.1 Team HA - Skills assessment	53
1.5.2 Contacts	54
1.5.3 Learning Resources	55
1.5.3.1 Accessing workflow related data	56
1.5.3.2 Docker	57
1.5.3.3 Git	59
1.5.3.4 GitHub	60
1.5.3.5 PyTest	61

Home

COMP90082 Hacking Materials User Interface (HA)

Project Overview

Materials science is a complex field that focuses on the study of physical materials and their properties. MatMiner is an existing Python library that combines materials data with Machine Learning strategies and models to study material properties without the need for time-consuming physical experimentation. Our project, Hacking Materials ("HA"), aims to build an easy and intuitive user interface to the [MatMiner library](#)
 * to eliminate the need for its users to have substantial Python or machine learning knowledge or experience.

Materials engineering is a field in which physical materials (e.g. metals, ceramics, polymers, composites, etc.) are studied to understand their composition, characteristics and properties. In many industries, such as mining, manufacturing and others, finding the right material for each job is essential for success. Materials engineers and scientists follow many different methods to compare candidate materials and make recommendations based on how they each satisfy the specific use case requirements. Some of these methods include physical experimentation, which can in some cases take up to 20 years to complete.

To avoid this, computer simulations (based on existing databases of known material properties and machine learning algorithms) can be used to compare materials in a much more efficient way. One such tool for this approach is the Python library [MatMiner](#), which allows easy access to ready-made datasets and integrates well with other machine learning Python libraries. However, to use this library, the user must have a substantial level of specialised knowledge in machine learning and programming, which most materials engineering do not have. To help them overcome this, the solution proposed by the client describes a simple and intuitive user interface that would act as a bridge between the user and the MatMiner library in the backend. The vision of this product is to make machine learning methodologies more accessible within the materials science and engineering industry as a whole, minimising the time and financial costs involved and leading to a more efficient industry.

Goal

Provide a user interface for standard processes in MatMiner: retrieving data from a database, users selecting features to be extracted from the database, performing simple machine learning tasks (Skikit Learn, Keras), and visualizing the results.

Team Members

				
Name	Dr Christian Brandl	Mauro Mello Jr		
Role	Client	Supervisor		

Useful links

Trello	Github	Slack
		

Tasks

- Edit this home page** - Click *Edit* in the top right of this screen to customize your Space home page
- Create your first page** - Click the *Create* button in the header to get started
- Brand your Space** - Click *Configure Sidebar* in the left panel to update space details and logo
- Set permissions** - Click *Space Tools* in the left sidebar to update permissions and give others access

Name	Yanan Liu (Rep)	Hongpei Lu	Jiahao Ju	Xinle Yu
Role	Team Representative Product Developer	Product Developer	Product Developer	Product Developer
Student ID	1289747	1275238	1128182	1294310
				1221568

Project Scope

At this stage in development, we are focusing on a minimal set of features applicable to all users of the product. We have worked with Dr. Brandl to identify a set of core features, and to cleanly separate these from "pro-user" and other nice-to-have features. These high-priority features are identified in the User Stories list.

For now, only the high-priority features are definitely in scope for development. Other nice-to-have features may be developed once the high priority features have been finished and a minimum viable product has been deployed.

This project is intended to be generally useful to a broad range of potential users, including:

- Educators, who may use it to demonstrate the Matminer library and its capabilities
- Students, who may use it to learn about these tools and experiment with them
- Materials engineers, who will need to be able to access more advanced features including downloading generated code and customising workflow components

Recent space activity

[Yanan Liu](#)



Handover Guide 11 •

[Rui ZHANG](#)



| Project Description 16 •

Deployment Process 17 •

Deployment Process delete 18 •

[Yanan Liu](#)



Set-up SSO Authorisation Guide 9:26

Handover Guide



HA-2022-handover.pdf

Source of truth is on Google Docs here: [Handover Document](#)

| Development

Development Process

Proposed by Red Back team and approved among three teams

Sprint Lifecycle

Sprint Kickoff

Attendees:

Everyone

Tasks:

- Pull user stories into the sprint from the product backlog
- Ensure user stories have complete definitions:
 - Do they have acceptance criteria?
 - Do they have test cases?
 - Are they ready for development? What are their dependencies?
- Re-estimate user stories with T-shirt size and priority
- Schedule kickoff meetings for user stories that are ready for development
- Review action items from the previous sprint's retrospective meeting

The aim of the sprint kickoff meeting is to define the high-level objectives for the sprint and consolidate learnings from previous sprints.

Feature Kickoff

Attendees:

The product owner, plus a subject matter expert for each domain the user story will involve. As an example, this might include:

- The product owner
- A frontend developer
- A backend developer
- A user experience (UX) designer
- Someone who worked on a related feature who may be able to give useful information

Tasks:

- Conduct a design discussion for the user story
- Break the user story down into multiple smaller tasks
- For each task, define:
 - Dependencies
 - Size (using **magic estimation**)
 - Relevant test cases and acceptance criteria
 - Assigned developer
- Create tasks in Trello

Development

Branching

- Use the format `feature/t-<ticket>` as a feature branch template, where `<ticket>` is the Trello card number
- Where branches have other purposes, prefixes other than "feature" may be used. For example:
 - "spike", for experimental work that shouldn't be merged
 - "fix", for fixes to existing features
- In general, follow [Git Flow](#) as a guide to using branches for development (we probably only need main, release, scratch, and feature branches)

Style

- Use auto-formatters and linters to maintain a consistent code style.
 - For Python, we use black and pylint
 - For TypeScript, we use Prettier

Tests

Where tests are available, **remember to run them before submitting a pull request(PR)**. Reviewers should check that tests pass before approving PRs for merge.

Code Reviews

Reviewers should:

- Verify that the pull request description includes:
 - A description of **what work was done**, and **why it was needed**. Usually, this just means a link or reference to the relevant user story or Trello card.
 - A description of **how the developer knows their work is correct**. The reviewer can then use this to cross-check the code.
- Verify that the tests pass. At a minimum, this means checking that the automated tests run by GitHub Actions passed
- Verify that all test cases and acceptance criteria identified in the relevant feature kickoff have been satisfied.

At least one developer from each of the other two teams **must** approve the pull request before it can be merged.

Sprint Retrospectives

Each sprint should end with a sprint retrospective. Each team has done this slightly differently, in the case of RedBack we have used a "Start / Stop / Continue" model.

The team representatives also conduct separate retrospectives to discuss team interactions, which used a "What went well / What didn't go well / What was confusing" model.

Releases

Versioning

Most of this is defined by the university. In general, though, we use [semantic versioning](#) and release by tagging releases on GitHub.

Quality Assurance Guidelines

Group cooperated work

Work Quality Guidelines

1. The code should be able to run. No syntax or compile-time errors.
2. If unit or integration tests are included in a pull request, they should pass
3. All preexisting tests should still pass with the new changes in the pull request
4. A branch should not be put up for a pull request if it has merge conflicts with the main. All conflicts should be resolved **before** requesting a review of a pull request.
5. The code should be understandable and contain documentation

Code Review Guidelines

1. Each PR must be reviewed by one member from each of the other two student teams.
2. Pull requests should:
 - a. Be assigned reviewers within **24 hours** of being submitted
 - b. Be given initial feedback by reviewers within **48 hours** of being submitted
3. Commit messages should describe the work you've done and the steps you took to verify the correctness of your work
4. Pull request descriptions should summarise the commit messages
5. Pull request changes should be reviewed by using the Review Changes button under the Files Changed tab, to encourage the use of **Accept** or **Review Changes** messages.
6. Pull requests should only be merged by the creator of the pull request
7. Branches that have been successfully merged to the main should be deleted by the creator of the pull request
8. Reviewers should check that pull requests follow the work quality guideline above

Acceptance Criteria definition guidelines

1. Acceptance criteria should be defined from the user's point of view
2. Acceptance criteria should contain a list of steps to test the desired functionality

Definition of "done" for a user story

1. Acceptance criteria should be defined for the user story and should pass
2. All related code has **passed code review and merged to the main**

Frontend Workspace Structure Proposal

Proposed by team Red Back and approved among three teams

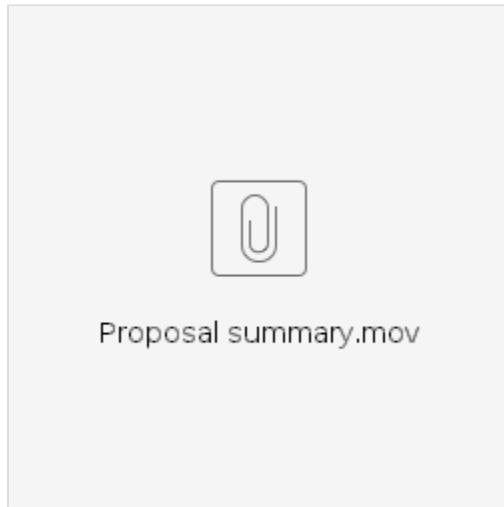
Proposed by	Ghina Yashar
Presented to	All frontend contributors from teams RedBack, BlueRing and BoxJelly
Proposal date	September 1 2022
Status	APPROVED - September 2 2022
Approvers	<ul style="list-style-type: none">■ Mamta Lopes(RedBack): 1/9/2022■ Felipe Lin (BoxJelly): 2/9/2022■ Rui Zhang (BlueRing): 2/9/2022

Proposed Structure

Please note: All the names used below can be replaced if needed, the focus of this proposal is more on the structure rather than the naming.

Summary video

If you don't like reading, please watch the video below for a quick overview of the proposed structure. The sample code snippets shown in the video are copied below as well.



Summary in writing

The structure I'm proposing would follow this rough directory tree:

Sample directory structure

```

|_assets
|_src
  |_components
    |_exampleComponent
      |_examples.tsx
      |_index.tsx
      |_test.tsx
      |_styled.tsx
    |_dropdownSelectStepType
      |_examples.tsx
      |_index.tsx
      |_test.tsx
      |_styled.tsx
    ...
  |_steps
    |_datasetSelection // (e.g.)
      |_index.tsx
      |_test.tsx
      |_HelpModal
        |_index.tsx
        |_styled.tsx
      ...
    ...
  ...
  |_sections
    |_appHeader
      |_index.tsx
    ...
    |_appBody
      |_InputPanel
        |_index.tsx
      ...
      |_ViewingWindow
        |_index.tsx
      ...
    |_index.tsx
  |_appFooter // amendment suggested by Felipe
  ...
  |_App.tsx
  ...
|_package.json
|_README.md
...

```

The main ideas of this are as follows:

Sections

By referring to the [low-fidelity prototype](#) created earlier in the project, we divide the main application page into 2 main sections:

- Header: the top bar, which does not need to have the context of what stage the user is up to and what's happening at any given point.
- Body: Includes 2 subsections that both need to know which stage the user is at (e.g. "Pre-process data" or "Apply machine learning"):
 - Left-side panel: named in the structure as `InputPanel`. Example code for this panel and how it shows the workflow steps is included in the sample code section below.
 - The main window on the right: named in the structure as `ViewingWindow`

AMENDMENT - 2 SEPTEMBER 2022

Felipe Lin (HA-BoxJelly) suggests potentially adding an `appFooter` as well. No objections to this so far.

Hacking Materials **SECTION: HEADER** ≡ ⌂

Step 2.1: Lorem ipsum i

Make a selection

Execute

Step 2.2: Lorem ipsum i

Lorem ipsum

Step 2.3: Lorem ipsum i

Make a selection

Execute

...

SECTION: BODY

Step 2.1:

Lore ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

Step 2.2:

Lore ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

...

< Go back Go

Hacking Materials ≡ ⌂

Step 2.1: Lorem ipsum i

Make a selection

Execute

Step 2.2: Lorem ipsum i

Lorem ipsum

Step 2.3: Lorem ipsum i

Make a selection

Execute

SUB-SECTION: INPUT PANEL ...

SUB-SECTION: VIEWING WINDOW

Step 2.1:

Lore ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

Step 2.2:

Lore ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

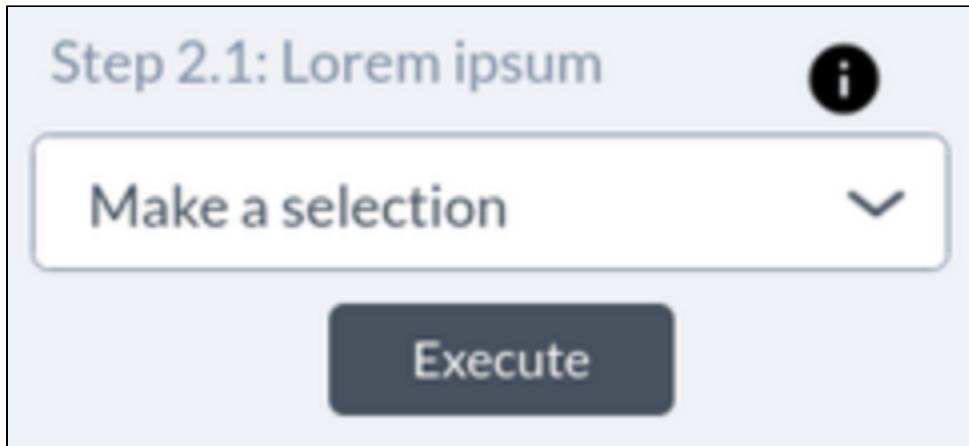
...

< Go back Go

Components

This folder will mainly contain all reusable components, e.g. Button, Tooltip, Modal, etc.

Notably, some of these reusable components would be "step types", e.g. `DropdownSelectStepType`. This example step type refers to the entire object shown below, including a step number, title, tooltip, dropdown list, button and whatever else may be needed. We would create this as a reusable component because many steps have similar requirements, e.g. selecting a dataset and selecting a featurizer should both be dropdown list type steps.



Steps

The word "steps" in this section refers specifically to the workflow steps that would be shown in the input panel, e.g. Dataset Selection step, Featurizer Selection step, etc.

A separate folder is created for these so that there would be a clear pattern that is easy to follow whenever more steps need to be added. Each step would use a step type component that is imported from the `/components` folder. For E.g. the `DatasetSelectionStep` would use the `DropdownSelectStepType`, as shown in the sample code snippet below.

Sample code snippets

Sample src/steps/datasetSelection/index.tsx

```
import DropdownSelectStepType from '../../../../../components/dropdownSelectStepType';
import HelpModal from './HelpModal';

...
const DatasetSelectionStep = (props) => {
    ...
    const STEP_KEY = "dataset_selection"

    const options = api_call_here() // calls backend API to get the dataset options

    const onSubmit = selected_value => send_to_backend() // send to backend using api

    return (
        <DropdownSelectStepType
            stepNumber={props.stepNumber}
            title="Select Dataset"
            description="bla bla"
            tooltipContent={HelpModal}
            options={options}
            onSubmit={onSubmit}
        />
    );
};
```

Sample src/sections/appBody/InputPanel/index.tsx

```
import DatasetSelectionStep from '../../../../../steps/datasetSelectionStep';
import FeatuirzerSelectionStep from '../../../../../steps/featuirzerSelectionStep';
...

const InputPanel = (props) => {
    ...
    const { stage } = props;

    if (stage === 1) {
        return (
            <div>
                <DatasetSelectionStep
                    stepNumber="1.1"
                    data={data}
                    handleChange={handleChange}>
                />
                <FeatuirzerSelectionStep
                    stepNumber="1.1"
                    data={data}
                    handleChange={handleChange}>
                />
                ...
            </div>
        );
    } else if (stage === 2) {
        return (
            <div>
                ...
            </div>
        );
    }
};

};
```

Set-up SSO Authorisation Guide



hacking_material...thentication.pdf

Deployment Process

Prerequisites:

- a. [Google API key for SSO login - Authenticate using API keys | Authentication | Google Cloud](#)
- b. [Microsoft API key for SSO login - Enable authentication in a web API by using Azure Active Directory B2C | Microsoft Learn](#)
- c. [Email Address](#) of assigned database administrator

Worth a read:

- [Connect over SSH with Visual Studio Code](#)
- [Docker - Visual Studio Marketplace](#)

There are three steps to deploying this product (see videos below)

1. Spin up a VM on Nectar ARDC, Google Cloud, AWS or Azure. Ensure you choose an **Ubuntu 22.04 image**.

Your browser does not support the HTML5 video element

2. Clone the GitHub repository and run the deployment script using the commands below.

```
sudo apt update && sudo apt upgrade && sudo apt install git  
git clone https://github.com/COMP90082-2022-SM2/HA-2022-SM2.git  
cd HA-2022-SM2/src/backend/  
git checkout deployment  
sudo chmod +x ./deploy_on_ubuntu22_04.sh  
../deploy_on_ubuntu22_04.sh
```

Your browser does not support the HTML5 video element

3. Point a domain name to the IP address of the VM (Unimelb IT department).

| Specifications

- | Project Description
- | Sprint Plan
- | Motivational Model
- | User Stories
- | Persona
- | Prototype
- | Business Case for Cross-Team Collaboration
- | Test cases
- | Acceptance Criteria

| Project Description

Background

- In computational materials engineering, the engineering design of new materials is accelerated by avoiding costly and lengthy experiments to demonstrate the properties of materials. Data mining methods discover better materials by searching computer-generated databases and making simulation predictions using high throughput and high-performance computing.
- Unfortunately, for some of the first movers, such as Tesla, translating this new thinking into engineering practice is still in its infancy. One obstacle is material engineers' software tools.
- Engineers with knowledge of the material domain need to effectively access and process this data in order to make informed decisions for the final machine learning strategy.

Goal

- Provide a user interface for standard processes in MatMiner: retrieving data from a database, users selecting features to be extracted from the database, performing simple machine learning tasks (Scikit Learn, Keras), and visualizing the results.

Sponsor

- **Dr Christian Brandl**
- Lecturer and sponsor at the University Of Melbourne.
- Completed his Ph.D. in Materials Science & Engineering at the École Polytechnique Fédérale de Lausanne EPFL and the Paul Scherrer Institute (Switzerland), followed by a post-doc in the Theoretical Division at the Los Alamos National Laboratory (USA).
- His research focuses on predictive atomistic simulation approaches to enable rational materials design.

In-scope

The product is to design and build the following:

- A web page tool to help the professional user and regular user or student who has no or very little experience in machine learning analysis method to complete material analysis;

Users requirements:

- Detailed analysis of user requirements;
- Design a beautified, user-friendly interface;
- Design a login page;
- Provide hints and guidance for new users;
- Design various machine learning models in Python that allow users to select features from a database and obtain a visual result;
- Allow to add new features;
- Provide code interface for the pro user;
- The workflow allows downloading;
- Machine learning model accuracy reaches a sufficient level;
- Complete the improvement suggestions that may be put forward by the clients;
- At least allow 30+ users to operate concurrently;
- Report the progress and results of the product;
- Runs on Melbourne Research Cloud;
- The final product results passed the test.

Out-of-Scope

- The team has no obligation to release back-end code to the public;
- Operating maintenance is not within the scope of the team's obligations;
- The interpretability of the machine learning model does not need to be guaranteed, which means that the team members can choose the black box model as the machine learning model;
- Due to the lack of resources, the backend server cannot guarantee the stability of multiple simultaneous access.

Stakeholders

Name	Position	Internal /External	Project Importance	Product Role
Dr. Christian Brandl	The person who originally wanted to start this product	External	High	Product Client

Mauro Mello Jr	The person who directs and supervises product team members on the product	Internal	Medium	Product Supervisor
Yanan Liu	Student of the University Of Melbourne enrolled in COMP90082	Internal	High	Team Representative Product Developer
Hongpei Lu	Student of the University Of Melbourne enrolled in COMP90082	Internal	High	Product Developer
Jiahao Ju	Student of the University Of Melbourne enrolled in COMP90082	Internal	High	Product Developer
Xinle Yu	Student of the University Of Melbourne enrolled in COMP90082	Internal	High	Product Developer
Rui Zhang	Student of the University Of Melbourne enrolled in COMP90082	Internal	High	Product Developer

Technologies

- Hosting
 - University of Melbourne Cloud Services
- Web Development Techniques
 - NodeJS
- Frontend
 - React
 - Typescript
- Backend
 - Python
 - Flask
 - Docker
- Machine Learning
 - [MatMiner](#)
 - [Scikit Learn](#)
- Admin & Collaboration Tools
 - Confluence
 - Trello
 - GitHub
 - Slack

Motivational Model

Assigned to [Yanan Liu](#), [Hongpei Lu](#)

Versions

Version ID	Description	Editor	Date
1.0	The first version is based on an initial understanding of the product and the first client meeting	Yanan Liu	2022-8-17
1.1	Add a goal model based on the initial do-be-feel list	Hongpei Lu	2022-8-18
2.0	All teams merged the result of the do be feel list and goal model	Yanan Liu Hongpei Lu	2022-8-19

Version 2.0

Group cooperated work [here](#)

Do-Be-Feel-Who List

Overall goal: Material engineers without a machine learning background can use the software fluently to perform the required data analysis

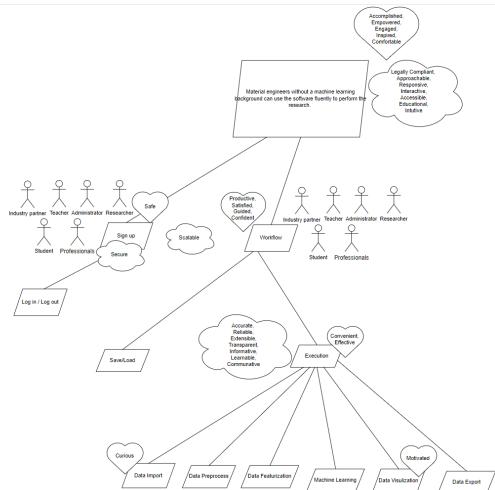
Who (users)	Do (functional goals)	Be (qualitative goals)	Feel (emotional goals)
Students	Add more database, machine learning method and plot types	Accessible	Accomplished
Administrators	Compare data using tables & plots	Accurate	Comfortable
Professionals	Data Pre-processing: Calculate descriptive statistics	Approachable	Confident
Industry Partners	Data Pre-processing: Consider anonymized data	Communicative	Convenient
Teachers	Data Pre-processing: Overview of the current import data	Educational	Curious
Researcher	Data Pre-processing: Reduces noise and eliminates ambiguity	Extensible	Effective
Code maintainers	Data Pre-processing: Standardizing data to bring it into the formatting range	Informative	Empowered
	Data Visualization: Data processing: Tabular data & Plotted Graph	Interactive	Engaged
	Edit python code directly in the interface	Intuitive	Guided
	Export input data	Learnable	Inspired
	Export jupyter notebook file	Legally Compliant	Motivated
	Export output data tables and figures	Reliable	Productive
	Featurization data: Add multiple composition-based features	Responsive	Safe
	Featurization data: Add multiple simple density features	Scalable	Satisfied
	Import Data: Create working spaces when importing	Secure	
	Import Data: Drag and drop import of files	Transparent (progress, error messages, notebook export...)	
	Import Data: Import data files (CSV, XES, Parquet) from local system		
	Log in/Log out		
	Machine Learning: Define input data and output data: Splitting data into training, test, and validation sets		
	Machine Learning: Determining model features and training the model: Configure and adjust hyperparameters for optimum performance		

	Machine Learning: Evaluate model performance and establish benchmarks: Continuous measurement and monitoring of model performance		
	Machine Learning: Evaluate model performance and establish benchmarks: Evaluate models using validation methods and validation datasets		
	Machine Learning: Get model results: The most important features of the current ML model		
	Machine Learning: Select the machine learning model to be used		
	Maintain software		
	save/load workflows		
	Sign up		

Version 2.0

Group cooperation work [here](#)

Goal model



| Personas

Assigned to [Xinle Yu](#)

Version	Description	Editor	Date
1.0	First version is based on an initial understanding of the project and the first client meeting	Xinle Yu	2022-8-19
1.1	Round off the sentences and adjust some description	Xinle Yu	2022-8-21
1.2	Combine three teams' work	Yanan Liu	2022-8-22
2.0	Adjust technical skills and edit the text for fitting the change of skills	Xinle Yu	2022-9-18
2.1	Combine three teams' work	Yanan Liu	2022-9-19
3.0	Update persona profiles based on feedback	Xinle Yu	2022-10-20

Persona 1: Student (By: BoxJelly)

Assol Anahita

age: 22

residence: Melbourne

education: Materials Science and Engineering

occupation: Student

marital status: Single

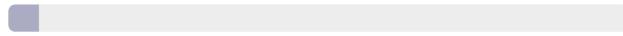


"It's SO time consuming to do material research and get decent results through just a semester."

Motivation: As a materials engineering graduate student, Assol gets frustrated and demotivated when she can't make sense of the data she has and feels that she is not really making real progress with her studies. She needs a tool that can speed up materials data retrieval and processing so she can focus in analysing the results to better understand the theory and concepts of materials engineering.

Comfort With Technology

PROGRAMMING WITH PYTHON



MACHINE LEARNING



CLOUD BASED STORAGE



MATERIALS SCIENCE



Criteria For Success:

Assol can perform materials data requests/retrievals and accurate materials property predictions supported by Machine Learning technology with easy to follow steps button clicks user interface.

Needs

- Easy-to-use interface materials science data processing and retrieval application
- A tool to predict property of a material with assistance of Machine Learning technology without prior knowledge of Python and Machine Learning programming

Wants

- A data mining application that helps her research projects
- A better understanding on how Machine Learning can help her to learn more about a material
- Ability to use ML algorithms as a black box
- Freedom to select features on her own terms
- A tool to accelerate research progress

Values

- Convenience
- Quickness
- Safety
- Understandable

Fears

- Spends hours working on a research project with very little progress because she neither has an adequate tool to do data mining, nor the programming skill to analyse the data herself
- Have to conduct countless experiments to figure out the properties of the materials

- Hard to choose suitable ML algorithms

Persona 2: Industry User (By: BlueRing)

Gray Zhou

age: 28

residence: Ningde, Fujian, China

education: Master of Material Engineering

occupation: R & D Engineer of Polymer

marital status: Single



"It is fantastic to apply a multi-function online tool with ML methods if it is efficient and reliable. Nobody will refuse a tool that can save time"

Motivation: Gray Zhou is a R & D Engineer of polymers in a battery factory. His work is searching for better materials for battery production. Gray spends a lot of time testing different materials, but some of tests are waste of time because of the poor performance observed. He needs a system that can predict some useful properties of materials so that he can remove samples with low predicted performance and boost the research. His company provides some ML tools, but they are awkward and only have limited functions.

Comfort With Technology

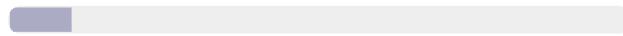
PROGRAMMING WITH PYTHON



MACHINE LEARNING



CLOUD BASED STORAGE



MATERIALS SCIENCE



Criteria For Success:

Provide a website or online-tool with quick, visual interface which can help him in daily development of new materials.

A successful product should help him save noticeable time on data processing and provide reliable prediction of properties.

Needs

- Retrieve and extract required data, process the data with ML methods to get some properties
- Provide graphs which can be modified with interface about predicted properties
- Help finding the material with best predicted properties

Wants

- Ability to interact with the graph to further compare serval materials in detail
- Upload data from his lab for predicting
- Explain what ML method the system applied and how it helps the prediction
- Continue his work on mobile devices without gaps of interaction

Values

- Easy to get started on both desktop and mobile
- Efficient back-end process
- Abilities to select functions and filter results
- Well organized visualization of interface and

Fears

- Not enough guidance in the web or tool so him may feel confused to find functions he wants.
- Lacking understand of what the system done, then reducing the confidence level of his report
- Frequently unable to access the system

gr dptts

• Frequently unable to access the system

Persona 3: Pro User (By: RedBack)

Alex

age: 45

residence: Melbourne

education: Masters Degree in Physics

occupation: Materials Engineer

marital status: Divorced without kids



"There has got to be a better way to do this."

Motivation : As an experienced Materials Engineer, Alex's job requires him to narrow down candidate materials by performing physical experiments to choose a material which can takes years to do. He needs a tool that can speed up the process by narrowing down candidate materials for experimentation using Machine Learning and simulations.

Comfort With Technology

PROGRAMMING WITH PYTHON



MACHINE LEARNING



CLOUD BASED STORAGE



MATERIAL SCIENCE



Criteria For Success:

Alex can find the right materials efficiently, with accurate results and that matches the client's requirements.

Needs

- Products to accelerate his workflow
- Access to wide variety of related tools and resources

Wants

- Suitable models and featurizers for different use cases
- Demonstrate reproducible results to his clients
- Share resources with others
- Refining generated workflow to reuse

Values

- Extensibility
- Accuracy
- Reliability
- Responsiveness
- Scalability
- Transparency

Fears

- Tool is too inflexible
- Losing access to progress on his work
- Not being able to verify his results
- Not having support with the tool

| User Stories

Assigned to Rui ZHANG

Versions

Version ID	Description	Editor	Date
1.0	First version based on initial understanding of the project and first client meeting	Rui ZHANG	2022-8-18
1.1	Added size estimation, priority, and justifications.	Rui ZHANG	2022-8-18
1.2	Supplement the remaining User Stories	Jiahao Ju	2022-8-18
1.3	Changed priority to Moscow priority. Added user story 5.	Rui ZHANG	2022-8-18
1.4	Delete unrelated stories. Add some general stories.	Jiahao Ju	2022-8-19
2.0	Merged user stories from all three teams	Rui ZHANG	2022-8-20
2.1	Reordered the user stories and grouped them by domain. Changed size to estimation of days	Rui ZHANG	2022-8-21
2.2	Regroup Action to new Epic and change original Epic to Domain	Yanan Liu	2022-8-21
2.3	Add epics summary table and explanation for size/priority	Rui ZHANG	2022-9-18
2.4	Add new US: US40 and US41 found during implementation	Jiahao Ju	2022-10-17
3.0	Fix order and some grammar. Update consistency with other teams	Rui ZHANG	2022-10-21

i Prioritization Technique

We used the MoSCoW prioritization classification.

Must have - must be included in the scope of the project, we defined this all the must have user stories can create a minimum viable product

Should Have - should be included in the scope of the project

Could Have - could be included in the scope of the project

Won't Have - will not be included in the scope of the project

Version 3.0 (cross-team effort)

Epics & Owning team allocation

	Epic	Total Size	Highest Priority within Epic	Assigned Team
1	Input Data	27	1 - Must Have	RedBack
2	Administration	31	1 - Must Have	BoxJelly
3	Machine Learning	16	1 - Must Have	BlueRing
4	Data Visualisation	13	1 - Must Have	BlueRing
5	Jupyter Notebook	20	2 - Should Have	Unassigned - stretch goal
6	External Data	8	2 - Should Have	Unassigned - stretch goal

User Stories

ID		Role		Action	Epic		Goal		Size (days)	Priority	Assigned Team
30	As a	general user	I want to	be able to view the citations for used featurizers	Input Data	so that	I could be know more about the source of the featurizer (legally compliant)		1	1 - Must have	RedBack

32	As a	general user	I want to	browse and select built-in featurizers	Input Data	so that	I can discover ways of manipulating my data	1	1 - Must have	RedBack
34	As a	general user	I want to	browse built-in datasets	Input Data	so that	I can discover data to experiment with	1	1 - Must have	RedBack
19	As a	student	I want to	quickly browse the Materials available in the database for retrieval and simulations	Input Data	so that	I can quickly perform queries.	3	2 - Should have	RedBack
21	As a	general user	I want to	be able to select datasets from existing databases	Input Data	so that	I do not have to worry about how the data is loaded	3	1 - Must have	RedBack
37	As a	general user	I want to	be able to preview the output of each execution	Input Data	so that	I could explore the data	1	1 - Should have	RedBack
41	As a	general user	I want to	view the columns of the saved dataset and select the column I want to featurize	Input Data	so that	I can provide the correct input to the featurizer	3	1 - Must have	RedBack
25	As a	general user	I want to	Select specific features from a dataset	Input Data	so that	I can improve the precision of my model	3	2 - Should have	RedBack
13	As a	Pro user	I want to	add new features	Input Data	so that	they can be reused in the future	5	2 - Should have	RedBack
28	As a	general user	I want to	be able to reference / view citation for original data sources	Input Data	so that	I can retrieve data.	1	3 - Could have	RedBack
18	As a	pro user	I want to	be able to apply new featurizers	Input Data	so that	I can create new features	3	3 - Could have	RedBack
1	As a	student	I want to	clean and tune data input	Input Data	so that	I have less noise on visualizations.	5	3 - Could have	RedBack
29	As a	student	I want to	save project specific data/checkpoints	Administration	so that	I can pick up where I left off for specific projects	1	1 - Must have	BoxJelly
35	As a	pro user	I want to	export model selections, parameters, and data flows	Administration	so that	I can save my work and share it with others	1	1 - Must have	BoxJelly
36	As a	pro user	I want to	import exported model selections, parameters, and data flows	Administration	so that	I can continue work I had previously saved	1	1 - Must have	BoxJelly
20	As a	student	I want to	Create an account using single-sign on, restricted to the *.unimelb.edu.au domain	Administration	so that	my research remains secure	3	1 - Must have	BoxJelly
23	As a	pro user	I want to	Control job execution	Administration	so that	I can start, view progress of, and cancel jobs related to my project	3	1 - Must have	BoxJelly
10	As a	pro user	I want to	be able to opt in to pro-user features	Administration	so that	I can access pro user features	5	1 - Must have	BoxJelly
38	As a	pro user	I want to	have my pro user settings persist on each visit	Administration	so that	I don't have to reconfigure settings to use the features I need	1	2 - Should have	BoxJelly
24	As a	student	I want to	receive provided hints and guidance for new users	Administration	so that	I can quickly learn how to use software	3	2 - Should have	BoxJelly
14	As a	pro user	I want to	easily find and read documentation on the pro features	Administration	so that	I can use them with ease	5	2 - Should have	BoxJelly
17	As a	pro user	I want to	Be kept informed about job status	Administration	so that	I can avoid polling my workspace to check for results	3	3 - Could have	BoxJelly
6	As a	pro user	I want to	have access to more processing power	Administration	so that	I can run more complex operations or use more data	5	3 - Could have	BoxJelly
31	As a	general user	I want to	able to select a Machine Learning model	Machine Learning	so that	I could use it to train and run the data	1	1 - Must have	BlueRing
33	As a	general user	I want to	browse built-in ML models	Machine Learning	so that	I can discover ways of manipulating my data	1	1 - Must have	BlueRing
39	As a	user	I want to	be able to select split ratio of data	Machine Learning	so that	to train and test the model	1	2 - Should have	BlueRing
26	As a	pro user	I want to	have the option to change the hyperparameters used in the machine learning model	Machine Learning	so that	I can fine tune my test results.	3	2 - Should have	BlueRing

15	As a	pro user	I want to	be able use additional ML models	Machine Learning	so that	I can improve accuracy	5	2 - Should have	BlueRing
7	As a	pro user	I want to	combine multiple ML models together	Machine Learning	so that	I can model more complex data manipulations	5	3 - Could have	BlueRing
40	As a	general user	I want to	select features used for x and y axis	Data Visualisation		I can plot the chart based on features I'm interested in	1	1 - Must have	BlueRing
22	As a	general user	I want to	see clear annotation or explanation of data points and features	Data Visualisation	so that	I can understand the results of the analysis	3	1 - Must have	BlueRing
8	As a	student	I want to	use different type of plotting graphs	Data Visualisation	so that	I have flexibility to visualize data according to my needs.	5	1 - Must have	BlueRing
9	As a	general user	I want to	able to view and plot the results of the model	Data Visualisation	so that	I could analysis and visualise the effects of the model	5	1 - Must have	BlueRing
12	As a	student	I want to	export my work to a Jupyter Notebook	Jupyter Notebook	so that	I can extend my work beyond the capability of the application	5	2 - Should have	TBD - after the completion of all assigned "should have's
2	As a	general user	I want to	attach comments to workflow objects	Jupyter Notebook	so that	I can document my work	5	3 - Could have	TBD - after the completion of all assigned "could have's
4	As a	Pro user	I want to	edit python code on the interface	Jupyter Notebook	so that	I can have control how the ML algorithms works	5	3 - Could have	TBD - after the completion of all assigned "could have's
5	As a	Pro user	I want to	upload my own script (in python) if possible	Jupyter Notebook	so that	I can extend the tool to support custom models and featurizers	5	3 - Could have	TBD - after the completion of all assigned "could have's
27	As a	pro user	I want to	be able to access new databases	External Data	so that	I can access addional data	3	2 - Should have	TBD - after the completion of all assigned "should have's
3	As a	Pro user	I want to	be able to add new datasets in the future	External Data	so that	if there's a new dataset that can be used on a new project, it can be added instantly	5	3 - Could have	TBD - after the completion of all assigned "could have's
11	As a	student	I want to	analyze the relationship between different features		so that	I can identify which features I need to select for my analysis	5	2 - Should have	TBD - after the completion of all assigned "should have's
16	As a	general user	I want to	add specific materials to the workflow for analysis		so that	compare the performance of the specific material my client or I choose with other material	3	3 - Could have	TBD - after the completion of all assigned "could have's

| Prototype

Assigned to [Yanan Liu](#)

Versions

Version ID	Description	Editor	Date
1.0	First version based on initial understanding of the project and client meeting Made in collaboration with team redback and boxjelly	Yanan Liu Ghina Yashar Felipe Leefu Huang Lin	2022-8-20

User Testing Prototypes

After we divided the workflow steps into three stages, it became unclear when and how the user would execute the workflow. After asking the client what he preferred, we realised that this is a more complex decision than we expected and it would need more consideration. To help the client reach a decision, we prepared three functional low-fi prototypes for user testing. Each prototype represented a different approach. The approaches were as follows:

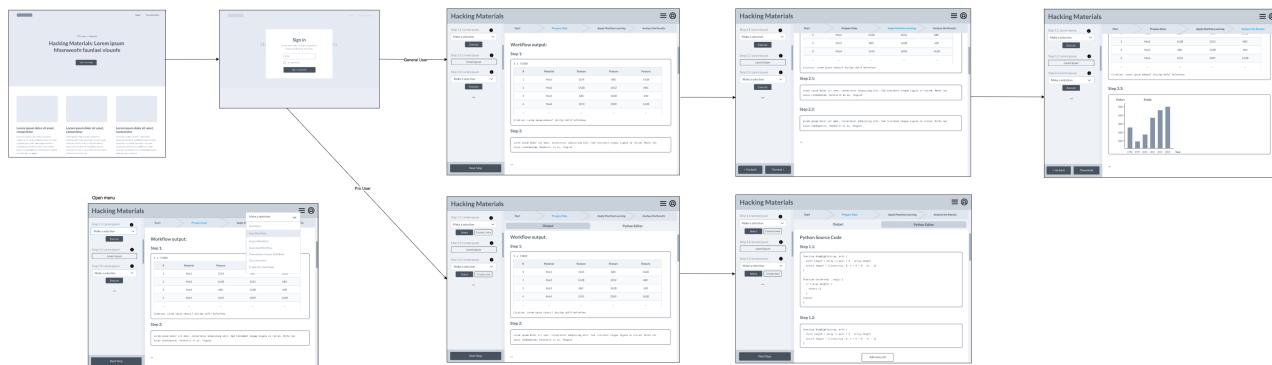
- Approach A - Run workflow steps individually. Prototype: <https://marvelapp.com/prototype/2g445gdg/screen/88805093>
- Approach B - Run the full workflow at once at the end. Prototype: <https://marvelapp.com/prototype/2g445gdg/screen/88805697>
- Approach C - Run each stage (which includes a number of steps) individually. Prototype: <https://marvelapp.com/prototype/2g445gdg/screen/88805736>

The prototypes were shared with the client, but as he was unavailable at the time, we decided to preemptively adopt Approach C as it offered the best balance between . Once the client was able to respond, he agreed with our decision. We then implemented Approach C, using the prototype as a guide.

Group cooperation work, work in marvel

Low-fidelity Prototype

Version 1.0



Descriptive Notes recorded by team redback

Descriptive Notes

- Landing page:
 - Static page with information about the app and project
 - Link to access the app
 - On click, it opens a login modal
 - Once the user is logged in, they're redirected to the app
- Single page app:

- Top bar:
 - User profile button at the top opens a menu to give the user the option to log out
 - Menu button at the top has options to import or save a workflow, download it in different formats, start over, a link to the documentation and a toggle to enable pro view.
- General user:
 - The workflow is divided into major and minor steps. Each major step would have its own page. The user can go back and forth between the major steps as needed.
 - Left panel:
 - All the minor steps are numbers and named to guide the user
 - Inputs can be of different types
 - Each step has a tooltip button that would open a modal with guidance information about the step
 - The steps and options in the left panel should always be the same no matter what selections the user made in previous steps. Any step that requires customised inputs would open in a modal.
 - Example 1: Step 3.1 might be "Selecting a plot type". As there is a known, limited list of different plot types, this step may be a drop-down menu that is displayed directly in the left panel.
 - Example 2: Step 3.2 might be customising the selected plot's configuration options. As different plot types may need different configuration options, these options will not be displayed in the panel directly. Instead, the panel will include only a button that says "Configure plot", which would open a modal with the specific options applicable to the selected plot type.
 - Pinned buttons at the bottom of the panel: navigate between the different major steps. Last step page may also have a button to download the full workflow.
 - Viewing window:
 - At the top of the viewing window, the user can see the progression of major steps with the current step highlighted.
 - The output of each minor step is labelled with the step number and contained inside a box. The output inside the box is the same output produced by running the python code, simply copied over for transparency.
 - The outputs from the previous pages are also always displayed, so it's not just the outputs of the current page.
 - Where a resource with citations is used, the citations will be automatically printed after the output of the step where the resource was selected.
- Pro user:
 - Left panel: has all the same options as a general user, plus additional buttons to configure their own settings as needed
 - Viewing window: the window has 2 tabs:
 - Output: same as the viewing window of the general user
 - Python source code:
 - An editable view of all the code generated by their selections, looks similar to a Jupyter notebook.
 - User can add new cells as desired
 - Brings up the following question: what happens if the user edits the code generated by one of the steps? This may lead to inconsistencies between what is shown in the step's input field and what the code now actually does. This is an implementation decision so is not a major concern right now, but one option that we decided to show in the prototype is that the step's input in the left panel would change to say "Custom" or something similar, indicating that the configuration was changed.

| Test cases

Versions

Version ID	Description	Editor	Date
1.0	Add test cases for the US 31	Hongpei Lu	2022-9-15
2.0	Add test cases for the US 08, 09, 40	Hongpei Lu	2022-10-16

US 31: As a general user, I want to be able to select a Machine Learning model.

US31, TC 01: Select a model (successful)

Test Type: Functional	Execution Type: Manual
Objective:	
Verify if a machine learning model is correctly selected.	
Setup:	
The IO stream has been established between the front end and the back end.	
Pre-Condition:	
1. The user has inputted the data or selected the data. 2. The user has selected the target feature.	
Notes:	
[1] Select the machine learning model in the interface. Must select a choice in the given place. Do not select the models that are out of scope in the given choices (Linear regression model and Random forest model). [2] Submit the information. * Application starts machine learning using the given input dataset and selected feature.	
Time constraint:	
Minimum: 10 min Maximum: 30 min	

US31, TC 02: Select a model (unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective:	
Verify if a machine learning model is correctly selected.	
Setup:	
The IO stream has been established between the front end and the back end.	

Pre-Condition:

1. The user has inputted the data or selected the data.
2. The user has selected the target feature.

Notes:

- [1] Try not to select the machine learning model in the interface.
[2] Submit the information.
* Application cannot learn without a selected machine learning model.

Time constraint:

Minimum: 10 min

Maximum: 30 min

US 40: As a general user, I want to select features used for the x-axis and y-axis.

US40, TC 01: Select an x-axis feature (successful)

Test Type: Functional	Execution Type: Manual
Objective:	
Verify if a feature for the x-axis is correctly selected.	
Setup:	
The IO stream has been established between the front end and the back end.	
Pre-Condition:	
<ol style="list-style-type: none"> 1. The user has inputted the data or selected the data. 2. The user has executed the machine learning model. 	
Notes:	
<p>[1] Select the feature for the x-axis in the interface. Must select a choice in the given place. Do not select the features that are out of scope in the given choices (The features are displayed in the interface). [2] Click the Execute button. * Application receives the x-axis feature value.</p>	
Time constraint:	
<p>Minimum: 1 min</p>	
Maximum: 5 min	

US40, TC 02: Select an x-axis feature (unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective:	
Verify if a feature for the x-axis is correctly selected.	

Setup:

The IO stream has been established between the front end and the back end.

Pre-Condition:

1. The user has inputted the data or selected the data.
2. The user has executed the machine learning model.

Notes:

- [1] Try not to select a feature for the x-axis in the interface.
 - [2] Click the Execute button.
- * Application cannot learn without a selected x-axis feature value.

Time constraint:

Minimum: 1 min

Maximum: 5 min

US40, TC 03: Select a y-axis feature (successful)

Test Type:	Execution Type:
Functional	Manual

Objective:

Verify if a feature for the y-axis is correctly selected.

Setup:

The IO stream has been established between the front end and the back end.

Pre-Condition:

1. The user has inputted the data or selected the data.
2. The user has executed the machine learning model.

Notes:

- [1] Select the feature for the y-axis in the interface.

Must select a choice in the given place.

Do not select the features that are out of scope in the given choices (The features are displayed in the interface).

- [2] Click the Execute button.

* Application receives the y-axis feature value.

Time constraint:

Minimum: 1 min

Maximum: 5 min

US40, TC 04: Select a y-axis feature (unsuccessful)

Test Type:	Execution Type:
Functional	Manual

Objective:

Verify if a feature for the y-axis is correctly selected.

Setup:

The IO stream has been established between the front end and the back end.

Pre-Condition:

1. The user has inputted the data or selected the data.
2. The user has executed the machine learning model.

Notes:

- [1] Try not to select a feature for the y-axis in the interface.
[2] Click the Execute button.
* Application cannot learn without a selected y-axis feature value.

Time constraint:

Minimum: 1 min

Maximum: 5 min

US 09: As a general user, I want to be able to view and plot the results of the model.

US09, TC 01: Plot the diagram (successful)

Test Type:	Execution Type:
Functional	Manual
Objective:	
Verify if a diagram can be displayed correctly.	
Setup:	
The IO stream has been established between the front end and the back end.	
Pre-Condition:	
<ol style="list-style-type: none"> 1. The user has inputted the data or selected the data. 2. The user has executed the machine learning model. 3. The user has selected features for the x-axis and the y-axis. 	
Notes:	
<p>[1] Click the Plot button. Must select an x-axis and the y-axis feature value before plotting the diagram. * Application receives the requests and plots the diagram.</p>	
Time constraint:	
Minimum: 1 min	
Maximum: 10 min	

US09, TC 02: Plot the diagram (unsuccessful)

Test Type:	Execution Type:
Functional	Manual
Objective:	
Verify if a diagram can be displayed correctly.	
Setup:	
The IO stream has been established between the front end and the back end.	

Pre-Condition:
1. The user has inputted the data or selected the data.
2. The user has executed the machine learning model.
Notes:
[1] The required choices are not selected: [1.1] Not select the x-axis feature. [1.2] Not select the y-axis feature. [2] Click the Plot button. * Application cannot plot the diagram without the x-axis and y-axis feature values.
Time constraint:
Minimum: 1 min
Maximum: 10 min

US 08: As a student, I want to use different types of plotting graphs

US08, TC 01: Select a type of regression model (successful)

Test Type: Functional	Execution Type: Manual
Objective:	
Verify if a regression model is correctly selected.	
Setup:	
The IO stream has been established between the front end and the back end.	
Pre-Condition:	
1. The user has inputted the data or selected the data. 2. The user has executed the machine learning model.	
Notes:	
[1] Select the regression model in the interface. Must select a choice in the given place. Do not select the regression models that are out of scope in the given choices (The regression model choices are displayed in the interface). [2] Click the Execute button. * Application receives the regression model.	
Time constraint:	
Minimum: 1 min	
Maximum: 5 min	

US08, TC 02: Select a type of regression model (unsuccessful)

Test Type: Functional	Execution Type: Manual
Objective:	
Verify if a regression model is correctly selected.	

Setup:
The IO stream has been established between the front end and the back end.
Pre-Condition:
1. The user has inputted the data or selected the data. 2. The user has executed the machine learning model.
Notes:
[1] Try not to select a regression model in the interface. [2] Click the Execute button. * Application cannot execute without a selected regression model value.
Time constraint:
Minimum: 1 min Maximum: 5 min

US08, TC 03: Create the regression plot (successful)

Test Type:	Execution Type:
Functional	Manual
Objective:	
Verify if a regression plot can be displayed correctly.	
Setup:	
The IO stream has been established between the front end and the back end.	
Pre-Condition:	
1. The user has inputted the data or selected the data. 2. The user has executed the machine learning model. 3. The user has selected features for the x-axis and the y-axis. 4. The user has selected the regression model.	
Notes:	
[1] Click the Plot button. Must select an x-axis, the y-axis feature value, and a regression model before plotting the diagram. * Application receives the requests and plots the diagram.	
Time constraint:	
Minimum: 1 min Maximum: 10 min	

US08, TC 04: Create the regression plot (unsuccessful)

Test Type:	Execution Type:
Functional	Manual
Objective:	
Verify if a regression plot can be displayed correctly.	
Setup:	
The IO stream has been established between the front end and the back end.	

Pre-Condition:

1. The user has inputted the data or selected the data.
2. The user has executed the machine learning model.

Notes:

[1] The required choices are not selected:

- [1.1] Not select the x-axis feature.
- [1.2] Not select the y-axis feature.
- [1.3] Not select the regression model.

[2] Click the Plot button.

* Application cannot plot the diagram without filling in the required choices.

Time constraint:

Minimum: 1 min

Maximum: 10 min

| Acceptance Criteria

User Story ID	User Story	Given	When	Then
33	Browse Machine Learning models	I have selected the target feature and I'm at the model selection step	I click on the dropdown menu	I should see a list of Machine Learning models with readable names
31	Select a Machine Learning model	I'm at the model selection step and I can see a list of Machine Learning models	I select one of the Machine Learning models and click on save	I should receive some feedback on whether the action is successful or not
39	be able to select split ratio of data	I've selected a dataset and saved my selection	I type in a ratio of train and test data and click the continue button	I should receive some feedback on whether the action is successful or not
26	Browse the hyperparameters used in the machine learning model	I successfully select one of the Machine Learning models	I receive the feedback that my selection is succeed	I should see some adjustable functionality for the hyperparameters of the model I selected
26	Be able to change the hyperparameters used in the machine learning model	I see some adjustable functionality for the hyperparameters of the model I selected	I adjust the values of these hyperparameters and click the continue button	I should receive some feedback on whether the action is successful or not
15	be able use additional ML models	I have selected the target feature and I'm at the model selection step	I click on the dropdown menu	I see enough Machine Learning models with readable names
7	Browse multiple models to be combined	I have selected the target feature and I'm at the model selection step	I receive the successful feedback from feature selection	I should see multiple dropdown menus that allow me to select multiple Machine Learning models to combine
7	Combine multiple models together	I see multiple dropdown menus that allow me to select multiple Machine Learning models to combine	I select several different models	I should receive some feedback on whether the action is successful or not
40	Browse features used for selecting x and y axis	I've selected a dataset and saved my selection	I click on the dropdown menu	I see a list of column names in the dropdown menu
40	Select features used for x and y axis	I am at the x and y axis selection step and I see a list of column names	I select two of the data columns as x and y axis and clicked on execution	I should receive some feedback on whether the action is successful or not
9	View and plot the results of the model	I successfully select x and y axis used for plotting	I click the plot button	I can see plots about the selected columns
8	Browse different regression models	I successfully select x and y axis used for plotting	I click on the dropdown menu of selecting regression models	I see several types of regression models
8	See different results of plotting graphs	I see several types of regression models	I choose one of the regression models and click on the plot button	I see a correct regression plot of my selection

Change Log

Date	Version	Author	Comment
2022-10-17	1.0	Jiahao Ju	First version.

| Sprint Artefacts

| Final Presentation

Date	2022-10-19 15:00
Presentation Recording	https://drive.google.com/file/d/1XFPP1e2R8pmjp7nLiCsyl5EVvZ9Cm50/view?usp=sharing
Presentation Slide (PDF)	 HA final presentation.pdf

| Sprint Plan

Assigned to @Rui Zhang, [Xinle Yu](#), [Hongpei Lu](#)

See details on Trello board [HERE](#).

Product Preparation (Aug 1 - Aug 7)

- Team form (Aug 1-3)
- Meeting with supervisor (Aug 4)
- Create a workspace on GitHub and Trello (Aug 5)
- Create Confluence pages (Aug 6)
- Draft of the workflow (Aug 7)

Sprint 1: Inception (Aug 8 - Aug 21)

- Kick-off Meeting with Client (Aug 12)
- Motivational Model (Aug 8 - Aug 14)
- User Story & Personas Of Users (Aug 15 - Aug 21)
 - Student: Box Jelly
 - Industry: Blue Ring
 - Pro-user: Red Back
- Prototype (Aug 20 - Aug 21)
- Development Environment (Aug 15 - Aug 18)
- Development workflow (Aug 19 - Aug 21)
- Documentation (Aug 8 - Aug 21)

Sprint 2: Development (Aug 22 - Sep 19)

- Retrospective of sprint 1 (Aug 22 - Aug 24)
- Sprint planning (Aug 23 - Aug 25)
- Sprint period (Aug 26 - Sep 19)
 - US 9 View and plot the results of the model
 - Frontend ([Rui ZHANG](#), 5 days)
 - US 22 See clear annotation or explanation of data points and features
 - Frontend ([Rui ZHANG](#), 5 days)
 - US 31 Select a Machine Learning model
 - Frontend ([Rui ZHANG](#), 2 days)
 - Machine learning ([Hongpei Lu Yanan Liu](#), 3 days each)
 - Backend ([Xinle Yu Jiahao Ju](#), 5 days each)
 - US 33 Browse built-in ML models
 - Frontend ([Rui ZHANG](#), 2 day)
 - Machine learning ([Hongpei Lu Yanan Liu](#), 10 days each)
 - Backend ([Xinle Yu Jiahao Ju](#), 5 days each)
 - Testing
 - Frontend ([Rui ZHANG](#), 3 days)
 - Machine learning ([Hongpei Lu Yanan Liu](#), 3 days each)
- Task Tracking (Aug 22 - Sep 19)
- Documentation (Aug 22 - Sep 19)

Sprint 3: Development (Sep 20 - Oct 21)

- Retrospective of sprint 2 (Sep 20 - Sep 23)
- Sprint planning (Sep 20 - Sep 23)
- Sprint period (Sep 24 - Oct 21)
 - US 8 Use different types of plotting graphs
 - Frontend ([Rui ZHANG](#), estimated 10 days)
 - US 9 View and plot the results of the model
 - Frontend ([Rui ZHANG](#), estimated 3 days)
 - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 3 days each)
 - Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
 - US 31 Select a Machine Learning model
 - Frontend ([Rui ZHANG](#), estimated 5 days)
 - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 3 days each)
 - Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
 - US 33 Browse built-in ML models
 - Frontend ([Rui ZHANG](#), estimated 5 days)
 - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 5 days each)
 - Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
 - US 39 Select split ratio of data
 - Frontend ([Rui ZHANG](#), estimated 5 days)

- Machine learning ([Hongpei Lu Yanan Liu](#), estimated 5 days each)
- Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
- Testing
 - Frontend ([Rui ZHANG](#), estimated 5 days)
 - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 5 days each)
 - Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
- Task Tracking (Sep 20 - Oct 21)
- Documentation (Sep 20 - Oct 21)

Sprint 4: Product (Oct 22 - Nov 4, Planned)

- Retrospective of sprint 3 (Oct 22 - Oct 26)
- Sprint planning (Oct 22 - Oct 26)
- Sprint period (Oct 24 - Nov 1)
 - Testing
- Task Tracking (Oct 22 - Nov 4)
- Prepare for final presentation (Oct 31 - Nov 4)
- Documentation (Oct 22 - Nov 4)
 - Release Note & release tag (Oct 31 - Nov 4)

| Client Communications

Time	Tool	Topic	Archive
Aug 3, 2022	Email	<ul style="list-style-type: none"> • The first communication with the Client; • Introduce the project team to the Client. • Invite the Client to join the first Client meeting. 	<p>COMP90082 HA Industry project Kick-Off Meeting [link]</p> <p> Yanan Liu <nanan.liu@student.unimelb.edu.au> Dr Christian Brandl, Eduardo Mauro, Hongpei me, Xinle, Rui ▾ Dear Dr Christian Brandl,</p> <p>My name is Yanan Liu and I am a student at the University of Melbourne comprising COMP90082 of the Master of FE. I am representing the student project team (BlueRing) assigned to your 2022 Industry Project. Thank you for providing us the opportunity to work with your company.</p> <p>We would like to organize the initial meeting to discuss the proposed project in more detail. Our proposed agenda items are: introductions; project overview; scope; expected deliverables/outcome; timelines; and our communications throughout the duration of the project.</p> <p>Here are possible dates available for our team:</p> <ul style="list-style-type: none"> • Friday at 11am-1pm & 1-2pm <p>Please let me know your preference or suggest alternatives.</p> <p>Given ongoing physical distancing restrictions, we anticipate most of our meetings will be virtual. When we schedule meetings, do you prefer Zoom, Teams, or another platform?</p> <p>Once the date/time is confirmed, I will send a calendar invite. If you want others from your team included, can you please forward their email addresses? To learn a little more about your project team, below are each of our names:</p> <p>Kind regards, Yanan Liu Graduate Researcher, Master of IP University of Melbourne yanan.liu@student.unimelb.edu.au</p> <p>Team Member 1: Hongpei me (hongpei.me@student.unimelb.edu.au) Team Member 2: Xinle Rui (xinle.rui@student.unimelb.edu.au) Team Member 3: Rui Zhang (rui.zhang@student.unimelb.edu.au)</p> <p>Again, thank you for hosting this project, and we look forward to meeting you soon and learning more about the project.</p> <p>Kind regards, Yanan Liu Graduate Researcher, Master of IP University of Melbourne yanan.liu@student.unimelb.edu.au</p>
Aug 4, 2022	Email	<ul style="list-style-type: none"> • The Client suggested a timeline for the first meeting with the project team. 	 <p>Christian Brandl <christian.brandl@unimelb.edu.au> to Yanan, Eduardo, Mauro, Hongpei, me, Xinle, Rui ▾</p> <p>Dear Yanan,</p> <p>Thanks for your email. It is great to see you taking on this project.</p> <p>Tomorrow, August 5th, I can meet at 4:15 pm (Melbourne time) using ZOOM.</p> <p>I am looking forward to meeting you and discussing a tool for materials' data exploration.</p> <p>Best regards Christian</p>

Aug 4, 2022	Email	<p>• The team members found the meeting time wrong and proposed to discuss the next week's meeting time the next day.</p> <p> Yanan Liu <yananli7@student.unimelb.edu.au> to Christian, Mauro ▾ Dear Dr Christian Brandl, Sorry for the late reply, we had some misunderstandings about the project schedule. The first official meeting should start next week. We need some time to coordinate with the two other teams participating in this project and ensure all team members are available. We will contact you again tomorrow with several options for meeting times next week. We apologise for the confusion caused by the previous email. Kind regards, Yanan Liu COMP90082, Master of IT University of Melbourne yananli7@student.unimelb.edu.au</p>
-------------	-------	--

Aug 29, 2022	Email	<p>• Team members wanted to discuss the frequency of Client meeting s with the Client.</p> <p>• Team members confirmed the language used in the project.</p> <p>• Team members suggested a possible schedule for that week's meeting.</p> <p>HA project meeting invitation [REDACTED]</p> <p> Yanan Liu yanli@student.unimelb.edu.au to Christian Meiss, cmeiss@unimelb.edu.au • Hi Christian,</p> <p>Thanks again for taking the time to check out our prototype and we look forward to getting feedback from your side. We have confirmed the languages that will be used in this project and look forward to being able to present them to you at the meeting. Also we want to discuss about current meeting frequency.</p> <p>Based on your response last week, we have scheduled this week's meeting for August 30th at 16:00 If you have any questions or comments in the meantime, please let me know. Thank you again and we look forward to working with you throughout the project!</p> <p>Zoom meeting info: Join from PC, Mac, iOS or Android: https://unimelb.zoom.us/j/8103054105?pwd=ZTg0UWh6RjU4VFlzZ0lYRngcfH46dz0 Passcode: 637115 Need to dial-in instead? Enter the meeting ID: 810 3054 105 via +61 3 7018 2009 or +61 2 8015 6011 Or join from a H.323/G.9800 room system: Dial: +61370182009 (or SIP: 8103054105@unimelb.edu.au) or 108.122.166.95 with meeting ID: 8103054105 and password: 637115 Help: https://unimelb.zoom.us/join?wiz=1 Legal: https://www.unimelb.edu.au/privacy</p> <p>The University of Melbourne collects your personal information via Zoom to facilitate virtual meetings, webinars and events. This may include your name, email address and any personal information you share via Zoom during the session. The University's General Privacy Statement details specific privacy collection notices provided to you at the time your personal information is collected for further detail how your personal information will be processed. Refer to Zoom's Privacy Statement for information about how Zoom collects and processes personal information.</p> <p>Kind regards, Yanan Liu yanli@student.unimelb.edu.au COMP 90082 2022 SM2 HA</p>

Aug 30, 2022	Email	<ul style="list-style-type: none"> The Client confirms the meeting time. 	<p>From: Christian Brandl <christian.brandl@unimelb.edu.au> Date: Tue, Aug 30, 2022 at 9:32 AM Subject: Re: HA project meeting Invitation To: Yanan Liu <yananl7@student.unimelb.edu.au> Cc: Mauro Mello Jr <mauro.mellojr@unimelb.edu.au>, Eduardo Araujo Oliveira <eduardo.oliveira@unimelb.edu.au></p> <p>Dear Yanan,</p> <p>I confirm our meeting at 4 pm today.</p> <p>Best regards Christian</p>
Sep 13, 2022	Email	<ul style="list-style-type: none"> Request the Client to hold a Client meeting on Friday of the current week based on the fortnightly meeting schedule. 	<p>Hacking Materials project meeting Schedule Inbox</p> <p> Yanan Liu <yananl7@student.unimelb.edu.au> to Christian, Mauro, eduardo.oliveira ▾</p> <p>Hi Christian. We send you our best wishes. As a result we were supposed to meet every fortnight, we would like to ask, would you like to have me Hope you have a good day!</p> <p>Regards, Yanan Liu</p> <p>COMP90082, Master of Information Technology University of Melbourne yananl7@student.unimelb.edu.au</p>

Sep 13, 2022	Email	<ul style="list-style-type: none">• The Client proposes two optional Friday Client meetings <p>On Tue, Sep 13, 2022 at 2:51 PM Christian Brandl <christian.brandl@unimelb.edu.au> wrote:</p> <p>Dear Yanan,</p> <p>Sure. I will be available either at 2 pm (1h) or at 4 pm (1h). Please let me know which time you prefer.</p> <p>Best regards Christian</p>
Sep 13, 2022	Email	<ul style="list-style-type: none">• Project team members chose one of the meeting times. <p> Yanan Liu <yananli7@student.unimelb.edu.au> to Christian, eduardo.oliveira, Mauro ▾</p> <p>Hi Christian,</p> <p>Thanks for your reply, We would like to have a meeting at 4 pm.</p> <p>Regards,</p> <p>Yanan Liu</p> <p>COMP90082, Master of Information Technology University of Melbourne yananli7@student.unimelb.edu.au</p>

Oct 11, 2022	Email	<ul style="list-style-type: none"> • Proto types share d with 3 approaches <p>----- Forwarded message ----- From: Christian Brandl <christian.brandl@unimelb.edu.au> Date: Wed, Oct 12, 2022 at 9:53 AM Subject: Re: Hacking Materials - Meeting invitation & status update To: Yanan Liu <yananl7@student.unimelb.edu.au> Cc: Eduardo Araujo Oliveira <eduardo.oliveira@unimelb.edu.au>, Mauro Mello Jr <mauro.mellojr@unimelb.edu.au></p> <p>Dear Yanan,</p> <p>Thanks for your update. Sorry that I have forgotten to reply to you with your option. Anyhow, I was about to opt for Approach C. You can use the following link (Book time with Christian Brandl: 1h Meetings regular hours • This link will expire on: January 10, 2023) to identify a suitable time slot in my agenda.</p> <p>Best regards Christian</p> <p>Christian Brandl Senior Lecturer dr sc Computational Materials Engineering Department of Mechanical Engineering Faculty of Engineering and IT Level 1, Melbourne Connect (Building 290) The University of Melbourne, VIC, 3010 Australia T: +61 3 8344 5331 E: christian.brandl@unimelb.edu.au blogs.unimelb.edu.au/materials-mechanics-modelling facebook.com/unimelb @nanoSimMat</p> <hr/> <p>From: Yanan Liu <yananl7@student.unimelb.edu.au> Sent: Tuesday, 11 October 2022 23:05 To: Christian Brandl <christian.brandl@unimelb.edu.au> Cc: Eduardo Araujo Oliveira <eduardo.oliveira@unimelb.edu.au>; Mauro Mello Jr <mauro.mellojr@unimelb.edu.au> Subject: Hacking Materials - Meeting invitation & status update</p> <p>Hi, Christian,</p> <p>Hope you had a good vacation. Would you have any available time to attend a meeting this week and discuss the latest progress of the project and some of the issues that exist?</p> <p>In our last email we sent you three different prototypes and since we didn't receive any responses from you, we opted to implement Approach C.</p> <p>Following list shows things that have been finished since last meeting</p> <ul style="list-style-type: none"> • Added support for viewing featurizer information, including: <ul style="list-style-type: none"> ◦ Citations ◦ Help text (generated from Python docs) ◦ Implementers • Added support for viewing the list of available dataset columns • Added support for Celery, a Python library for supporting long-running tasks (like machine learning) • Added a landing page with Google Sign-In support • Implemented several machine learning models • Added support for storing and retrieving a user's Workflow object, including: <ul style="list-style-type: none"> ◦ Selected dataset and featurizer ◦ Columns that should be featurized ◦ ML model to be applied ◦ Plot type to be used ◦ Tested deployment to University of Melbourne infrastructure (NeCTAR) <p>We have completed a lot of fragmented code and the next plan is to bring them together. Making ML models run under the Celery task execution system, and connecting the model outputs to the plotting system.</p> <p>Looking forward to hearing from you and wishing you a good day.</p> <p>Kind regards, Yanan Liu yananl7@student.unimelb.edu.au COMP 90082 2022 SM2 HA</p>
--------------	-------	---

Oct 18, 2022	Email	<ul style="list-style-type: none"> • Present at on Inv ite <p> THE UNIVERSITY OF MELBOURNE</p> <p>Presentation tomorrow 1 message</p> <p>Alastair Daivis <adaivis@student.unimelb.edu.au> To: christian.brandl@unimelb.edu.au</p> <p>Tue, Oct 18, 2022 at 11:39 AM</p> <p>Hi Christian - just writing to confirm that we will be giving a presentation on Zoom tomorrow at 3pm. I see you've already RSVP'd to the calendar event, but because the event was rescheduled the Zoom message in that invitation included an old date.</p> <p>Please find the updated Zoom invitation copied below.</p> <p>See you there!</p> <p>Alastair Daivis adaivis@student.unimelb.edu.au COMP90082 2022 SM2 HA RedBack</p> <p>Zoom meeting on Oct 19, 2022 03:00 PM Australia/Melbourne</p> <p>Join from PC, Mac, iOS or Android: https://unimelb.zoom.us/j/89614776080?pwd=UVhyTIRCZRhd2NVcnNNUDIRMIzUT0 Password: 125278</p> <p>Need to dial-in instead? Enter the meeting ID: 896 1477 6080 via +61 3 7018 2005 or +61 2 8015 6011</p> <p>Or join from a H.323/SIP room system: Dial: 89614776080@global.zoomcrc.com or SIP: 89614776080@zmau.us or 103.122.166.55 with meeting ID: 89614776080 and password: 125278</p> <p>Help: https://unimelb.service-now.com/it Legal: https://www.unimelb.edu.au/legal</p> <p>The University of Melbourne collects your personal information via Zoom to facilitate virtual meetings, webinars and events. This may include your name, email address and any personal information you share via Zoom during the session. The University's General Privacy Statement details how we collect and process personal information. Specific privacy collection notices provided to you at the time your personal information is collected further detail how your personal information will be processed. Refer to Zoom's Privacy Statement for information about how Zoom collects and processes personal information.</p>
--------------	-------	--

| Resources

GitHub: <https://github.com/COMP90082-2022-SM2/HA-BlueRing>

Trello: <https://trello.com/b/Q1AZdIXb/ha-bluering-sprints>

Matminer: https://nbviewer.org/github/hackingmaterials/matminer_examples/blob/main/matminer_examples/index.ipynb

LMS Project: https://canvas.lms.unimelb.edu.au/courses/126944/pages/hacking-materials-user-interface-code-ha?module_item_id=3996633

Dr. Christian Brandl information: <https://findanexpert.unimelb.edu.au/profile/826808-christian-brandl>

Motivational Model: <https://momo-staging.eresearch.unimelb.edu.au/dashboard>

| Team HA - Skills assessment

Email Address	Team	Matminer	React	Flask	Pandas	Scikit-learn	Numpy	Matplotlib	Seaborn	Plotly	Bokeh	Tensorflow	Keras	Pytorch	NodeJS	Pure HTML/CSS	Angular	Vue	Django	NodeJS	Any other suggestions for backend
Hongpei Lu	Bluering	0	0	2	3	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
Jiaheo Ju	Bluering	0	0	0	4	4	4	4	0	0	1	0	2	0	3	0	0	0	0	0	
Xinle Yu	Bluering	0	0	0	4	4	4	4	4	4	2	3	3	3	1	1	0	0	0	1	
Rui ZHANG	Bluering	1	2	3	4	5	4	0	0	1	0	1	5	3	2	1	0	0	0	0	0 Server/Cloud opt
Yanan Liu	Bluering	0	0	0	4	4	4	4	2	2	0	3	3	3	0	0	0	0	0	0	0
Dara O'Hehir	Bokehly	0	1	2	1	5	5	5	3	3	2	0	4	4	1	2	2	1	0	0	2
Yaocheng Xuan	Bokehly	0	0	0	4	5	5	5	0	0	0	0	1	1	0	0	0	0	0	0	
Tengfei Huang Lin	Bokehly	0	1	4	2	1	1	1	1	0	0	0	0	0	4	4	0	0	3	3	
Zhixue WANG	Bokehly	0	1	0	4	2	4	3	2	2	0	3	0	2	0	2	0	0	1	1	
Radimka Djan	Bokehly	0	0	0	3	4	4	4	3	3	1	1	3	3	2	2	0	0	3	0	
Marta Reith Lopes	Redback	0	1	3	0	3	5	5	4	0	0	0	1	1	0	0	4	0	0	0	
Ghina Yashar	Redback	1	5	0	4	4	4	4	0	4	0	4	4	4	2	5	2	1	0	2	
Chunbabuque Yang	Redback	1	0	0	3	4	3	2	2	0	0	0	3	3	1	0	1	0	0	0	
Sanjeevani Avasthi	Redback	2	1	1	5	5	5	4	2	4	0	3	3	4	0	3	0	1	0	0	
Alastair Davies	Redback	1	1	2	2	2	3	2	2	2	0	1	1	3	2	2	1	2	1	2	

Description	Files
Summary	
Distribution of skills	

| Contacts

Team Members

Name	Email address	Student ID
Yanan Liu (Team representative)	yananl7@student.unimelb.edu.au	1289747
Hongpei Lu	hongpeil@student.unimelb.edu.au	1275238
Jiahao Ju	jiahaoj1@student.unimelb.edu.au	1128182
Xinle Yu	xinley@student.unimelb.edu.au	1221568
Rui Zhang	rzzhan2@student.unimelb.edu.au	1294310

Client

Name	Email address
Dr. Christian Brandl	christian.brandl@unimelb.edu.au

Teaching Staff

Name	Email address
Mauro Mello Jr (Supervisor)	mauro.mellojr@unimelb.edu.au
Eduardo Oliveira	eduardo.oliveira@unimelb.edu.au

Team RedBack

Name	Email address
Alastair Daivis (Team representative)	adaivis@student.unimelb.edu.au
Sanjeevani Avasthi	savasthi@student.unimelb.edu.au
Mamta Lopes	mlopes@student.unimelb.edu.au
Ghina Yashar	gyashar@student.unimelb.edu.au
Chunbaixue Yang	chunbaixue@student.unimelb.edu.au
Confluence Page: Home	

Team BoxJelly

Name	Email address
Zhaoqi Wang (Team representative)	zhaoqiw@student.unimelb.edu.au
Dara O hEidhin	doheidhin@student.unimelb.edu.au
Yaoming Xuan	yaomingx@student.unimelb.edu.au
Radhimas Djan	djanr@student.unimelb.edu.au
Felipe Leefu Huang Li	fleefuhuangl@student.unimelb.edu.au
Confluence Page: Home	

| Learning Resources

Backend

- Pytest: <https://docs.pytest.org/en/7.1.x/getting-started.html>
 - SQLAlchemy: <https://docs.sqlalchemy.org/en/14/dialects/postgresql.html>
 - Flask :
 - <https://flask.palletsprojects.com/en/2.2.x/>
 - <https://www.tutorialspoint.com/flask/index.htm>
 - https://youtu.be/Z1RJmh_OqeA(Quick guide from setup, creating virtual env similar to the [Developer.md](#) , connecting to SQLAlchemy and the frontend)
 - <https://github.com/realpython/discover-flask>
-

Frontend

- React: <https://reactjs.org/tutorial/tutorial.html>
- Style components: <https://www.robinwieruch.de/react-styled-components/>
- Testing:
 - [RTL Cheat sheet](#)
 - [RTL + Jest tutorial](#)
 - [Jest cheat sheet](#)
- TypeScript:
 - [TS Cheat sheet](#)
 - [TS tutorial](#)
 - [TS + React tutorial](#)
- Storybooks:
 - [Tutorial \(start from step 2\)](#)
 - [Official "how to write stories" guide](#)

Accessing workflow related data

Written by RedBack

WorkflowConfigurations data

WorkflowConfigurations class in workflow_configurations.py is a data class in the backend containing all the configurations needed for executing each stage in the whole workflow. This class is created and saved to the database when a new user is created.

The choices made by user from the frontend are captured in this class and then saved in the Users table in the database.

When new choices are added in the frontend, the WorkflowConfigurations class should be extended as well to accomodate the changes, so choices made by the user can be consistent throughout the whole application.

save_workflow and load_workflow functions in workflow_data_handler.py can be used to in the backend to save and load the WorkflowConfigurations object to and from the database.

Dataset for machine learning and plotting

In the first stage of the whole workflow, featurized data will be created after the stage execution. This data is then also saved in the Users table in the database. Using save_featurized_data in workflow_data_handler.py will allow you to the dataframe to the database if there's any change to it.

To access the data for machine learning, or plotting, load_featurized_data in workflow_data_handler.py should be used to return the data.

Docker

Written by RedBack

Docker is a tool for managing isolated runtime environments (called "containers"). You can install Docker by following the Docker guide (linked below).

You can think of it as a combination of:

- A way to run virtual machines
 - But it's much more efficient
 - And makes it easier to access host machine resources, for example network ports and filesystems
- A way to build custom virtual machines, just like `make` can be used to build binaries or applications
 - The build instructions are described by the `Dockerfile` file

Bear in mind though that **containers are not virtual machines** - that analogy is useful for describing an overview of what Docker does, but it is not accurate.

The two most important things you can do with Docker are:

- Build container **images**, for example by running: `docker image build -t flask-docker .`
 - A container *image* is like a compiled program: it does nothing unless it is run
 - Unlike a program though, it doesn't exist in an obvious way on the filesystem. The Docker runtime is responsible for managing images.
 - `-t flask-docker` tells Docker to "tag" the image with the string "flask-docker". You can refer to Docker objects (including images) by their hash, but naming them makes it easier.
 - The `.` tells Docker to use the Dockerfile in the current directory. You could provide a path to another directory instead.
- Run a container image, for example by running `docker run -d -p 80:5000 flask-docker`
 - Running an image produces an object called a "container" - this is like a process, just as an image is like a compiled program.
 - The `-d` argument tells Docker to "detach" the container - to run it in the background and not show its output.
 - The `-p 80:5000` argument tells Docker to map port 80 on the host machine to port 5000 inside the container. If you're building a Flask application (which by default listens for requests on port 5000), this means that you would access the application by navigating to port 80 on the host machine.
 - `flask-docker` is the tag of the image we want to run.

Once a container is running, it is given a tag by the Docker runtime. This tag is separate to the tag of the image.

You can view running containers by running `docker ps`.

When you're finished with a container, you can kill it by running `docker kill <container-tag>`

For more information, you can go through the [Docker getting started guide](#), or refer to their [reference documentation](#).

Docker-Compose

An alternative to Docker approach detailed above

Docker-compose can automate the process of setting up & managing multiple containers. see [Install Docker Compose](#)

Before running the command below create a `.env` file in this directory and add the following lines

```
POSTGRES_USER="db_user"
POSTGRES_PASS="somehardpassword"
PGADMIN_USER="user@gmail.com"
PGADMIN_PASS="anotherhardpassword"
```

Replace the text inside the inverted commas with appropriate email/password/usernames

Then create a `.pgpassfile` in the `postgres-init` folder and add the following line

```
pg_db:5432:ha_db:username:password
```

Replacing username and password with the postgres username and password chosen above

For security, ensure `.env` & `.pgpassfile` is added to `.gitignore`

Running `docker-compose up -d` in this folder will set up 3 containers with persistant storage

- Flask app as defined in **Dockerfile**, available at `127.0.0.1:5000`.
- postgres database available at `127.0.0.1:5432`.
- PGAdmin, a UI to easily manage the database `127.0.0.1:5050`. [pgAdmin 4 Docs](#)

Using a Docker Development Container

While it's a little unusual, you can also use Docker to set up a **development container**. This is a container that you would use to create an isolated environment for software development - for example, if your main machine is a Windows machine, but you want to use a Linux development environment.

Visual Studio Code includes some tools for this: <https://code.visualstudio.com/docs/remote/containers>

In this kind of setup, you would create a **separate container** for your development environment, and connect to it from Visual Studio Code.

From the VSCode terminal, you would be able to access the container via a shell just as you could on a normal Linux machine.

Inside the container you could install Docker and develop as normal. However, there could be some issues with accessing web services hosted inside the development container. It may be simpler to use the **Windows Subsystem for Linux** instead: <https://code.visualstudio.com/docs/remote/wsl>

Git

Written by RedBack

Introduction

I strongly recommend learning how to use Git from the command-line. You can find good resources for doing this on the internet, but one of the best places to start in my opinion is the Atlassian Git tutorial.

The initial part of this tutorial is mainly about setting up a repository, which we have already done and might not be so useful to you. I would recommend starting with [Saving Changes](#), which introduces the "three trees" of Git. Understanding how those trees work together is fundamental to understanding how Git works and how to use it effectively.

A second, also excellent resource is [Thoughtbot's Git Tutorial](#), which comes with written tutorials, recorded videos, and example repositories. In particular, their [end-to-end feature example](#) very closely matches how I like to use Git.

Branching strategy

Branches are key to using Git effectively. While the tutorials above give some good advice on using them, every team has a slightly different branching strategy so I'll mention some good guidelines here:

- **Always use a branch.** Branches keep you safe by making sure you can get back to a given state of the repository. If you're about to do something risky (for example, a `git amend` or `git rebase` command), make a branch first so that if it goes wrong you can check out the "before" version of your code and undo the mistake.
- **Do not modify the history of shared branches.** If you are sharing work on a branch with other developers, using a command like `git commit --amend` or `git rebase` can cause your version of the branch to become different from the version that the other developers have.
- Adopt a **common strategy for naming branches**. I like to use the following formula:
 - Use a slash (/) to separate different parts of the branch name
 - A brief description of the **kind of work** being done on the branch (`feat` for new feature development, `style` for code style changes, `doc` for adding documentation, `tests` for adding unit tests, and so on)
 - A short description of the **actual work** being done on the branch, for example `feat/backend-featurizer-list-endpoint`

Resources

- The official [git tutorial](#)
- Atlassian's [git cheatsheet](#)
- [Dangit, Git!?](#) is a good resource for figuring out how to fix a mistake you've made using Git

Tools

While your IDE may have built-in Git support, I recommend using a stand-alone tool. They tend to be more stable and provide a good overview of the state of the repository.

I use [Sublime Merge](#), but other popular tools include:

- [SourceTree](#), by Atlassian
- [Tower](#)
- [tig](#), a command-line UI

Again though, I strongly recommend becoming familiar with Git's command-line interface. Often the GUI interfaces are convenient but hide information that can help you actually understand what they are doing, and knowing the command-line interface will help you use them more effectively.

Related to the above, if you are not yet familiar with a command-line editor I would strongly recommend learning one.

By far the most popular of these are Vim and Emacs, but both of these have quite steep learning curves. If you are looking for a command-line editor to get started with, the easiest is probably `nano`, which comes installed on most MacOS and Linux machines (I'm not sure about Windows, sorry!).

For those of you who are already confident with Git, some other tools you might find interesting / useful include:

- [Interactive Rebase Tool](#), a more convenient interface for using `git rebase -i`.
- [Trail](#), a tool I developed for rebasing feature branches on top of an amended commit after responding to code review feedback.

GitHub

Introduction

GitHub fills two key roles in our project:

- Code hosting server: it acts as a Git server from which we can push and pull commits, and by doing so share our work
- Code review platform: it acts as a way for us to share feedback on each other's work, facilitating quality control, collaboration, and knowledge distribution.

If you're familiar with Git already, most of GitHub should not be new to you. The exception to this is GitHub's **Pull Request workflow**, which covers the process of merging changes from a feature branch into the main branch.

With this in mind, I recommend reading GitHub's [documentation on this topic](#).

Tools

- GitHub's [command-line interface](#) is very good
- [GitHub Desktop](#) is a desktop application, which you may find more convenient than using GitHub's website.

PyTest

Pytest is a python testing framework that is used to write a small, readable test that can be scaled to a complex test such as testing API or database and UI. A testing framework is used to make sure our code behaves as we expect and ensure any changes to the code won't cause a regression in the program. The reason we use Pytest is because of the easiness of use of the library. Pytest can be installed onto your machine by manual download of Pytest packages or installed through the requirements.txt, which is located in the HA-2022-SM2\src\backend

Install command:

```
python -m pip install pytest  
HA-2022-SM2\src\backend > python pip install -r requirements.txt
```

In this project, Pytest is used to test several parts of the project. For example, it is used to test if the databases can be used to read and write. It is also used to test the program if the workflow is able to be updated, if a user is able to log in/register using test data, and many more. Pytest is integrated into our Continuous integration, where we test any code with Pytest before reviewing a pull request to make sure the code is able to work as expected.

For more info on how to learn Pytest, the following links are recommended:

- <https://docs.pytest.org/en/7.2.x/getting-started.html#get-started>
- <https://www.tutorialspoint.com/pytest/index.htm>