

End User and Technical Guides Submission

FIT3164: Data Science Project 2

Ensemble machine learning application for cancer
prediction using omics data

Group: MDS4

Supervisor : Dr. Ong Huey Fang

Project Manager : Raunaq Nawar

Technical Lead : Jia Chen Kuah

Quality Assurance : Han Wei Lim

Submission: 19 May 2023

Table of Contents

End User Guide	3
1. Overview	3
2. Launching the Web Application	3
3. Training the Model with User Data	4
3.1 Navigate to the Prediction page	5
3.2 Configuring the Input Parameters	6
3.3 Handling Input Errors	6
3.3.1 Handling CSV Input Errors	6
3.3.2 Handling Target Column Name Errors	7
3.3.3 Handling Feature Selection Input Errors	7
3.4 Training the Model	8
4. Interpreting the Result Page	9
4.1 Performance Metrics	9
4.2 Functionalities of the Buttons on the Result Page	10
5. Performing Predictions with Trained Model	10
Technical Guide	12
1. Recommended hardware	12
2. Software Requirement	13
3. Setup	13
4. Project Structure	13
5. Version Control	15
5.1 User	15
5.2 Developer	15
6. Miscellaneous	15

End User Guide

1. Overview

The purpose of this section of the user guide is to help users of our website use the ensemble machine learning model that we have created. To use our website and the model successfully, we presume users have successfully finished the steps specified in Section 2 of the Technical Guide, which relates to the hardware and software requirements.

2. Launching the Web Application

Using the PyCharm terminal, you can navigate to the project folder directory by accessing its corresponding location.

A screenshot of a PyCharm terminal window. The title bar shows 'Terminal: Local x + v'. The terminal content includes 'Windows PowerShell', 'Copyright (C) Microsoft Corporation. All rights reserved.', and a link to 'https://aka.ms/PSWindows'. The prompt is 'PS C:\Users\Kuah Jia Chen\Documents\Monash_Resources\Sem 1 2023\FIT3164\Project_FIT3164\Project>' with a cursor at the end.

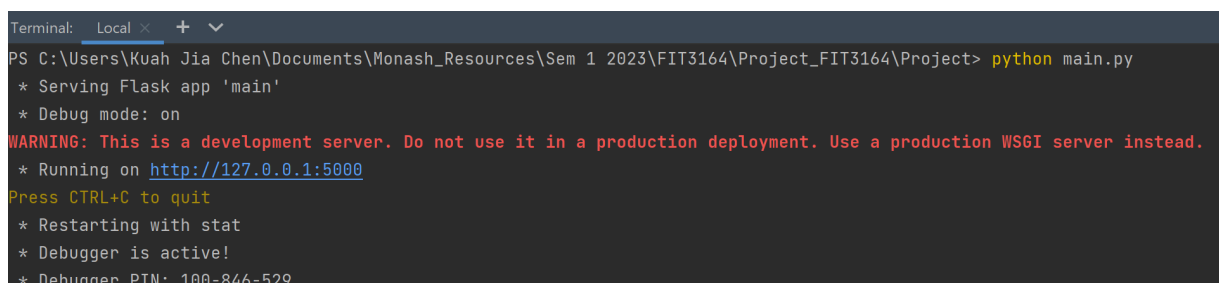
```
Terminal: Local x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Kuah Jia Chen\Documents\Monash_Resources\Sem 1 2023\FIT3164\Project_FIT3164\Project>
```

Image 1: PyCharm Terminal

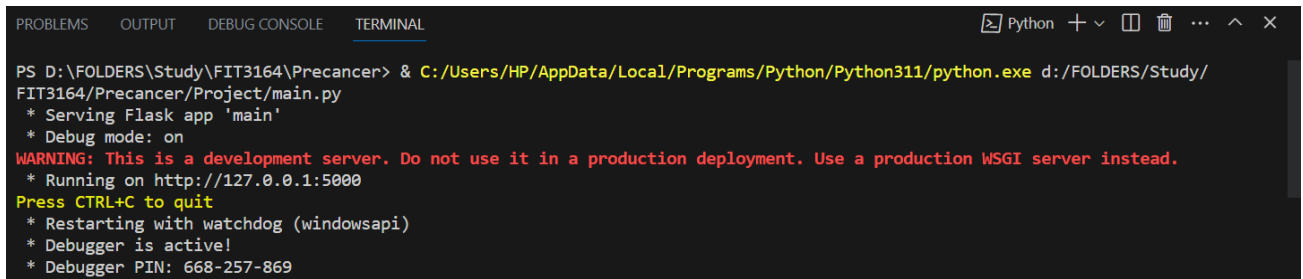
The project's Graphical User Interface (GUI) can be launched by running the required Python script file using the 'python main.py' command.

A screenshot of a PyCharm terminal window showing the execution of 'python main.py'. The output includes 'Serving Flask app 'main'', 'Debug mode: on', a warning about development vs production servers, the URL 'http://127.0.0.1:5000', and instructions to press CTRL+C to quit. It also shows 'Restarting with stat', 'Debugger is active!', and a debugger PIN.

```
Terminal: Local x + v
PS C:\Users\Kuah Jia Chen\Documents\Monash_Resources\Sem 1 2023\FIT3164\Project_FIT3164\Project> python main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 100-846-529
```

Image 2: PyCharm Terminal

The same thing can be done with Visual Studio Code as well instead of PyCharm, it varies with the users preference.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\FOLDERS\Study\FIT3164\Precancer> & C:/Users/HP/AppData/Local/Programs/Python/Python311/python.exe d:/FOLDERS/Study/
FIT3164/Precancer/Project/main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 668-257-869
```

Image 3: Visual Studio Code Terminal

To launch the web application, you first need to click on the URL <http://127.0.0.1:5000>. The local server address where your Flask application is executing is found at this URL. You can access the web application after clicking the link by launching a browser and going to the same address, <http://127.0.0.1:5000>. After doing this, the web application should appear in your web browser as shown in the following figure.

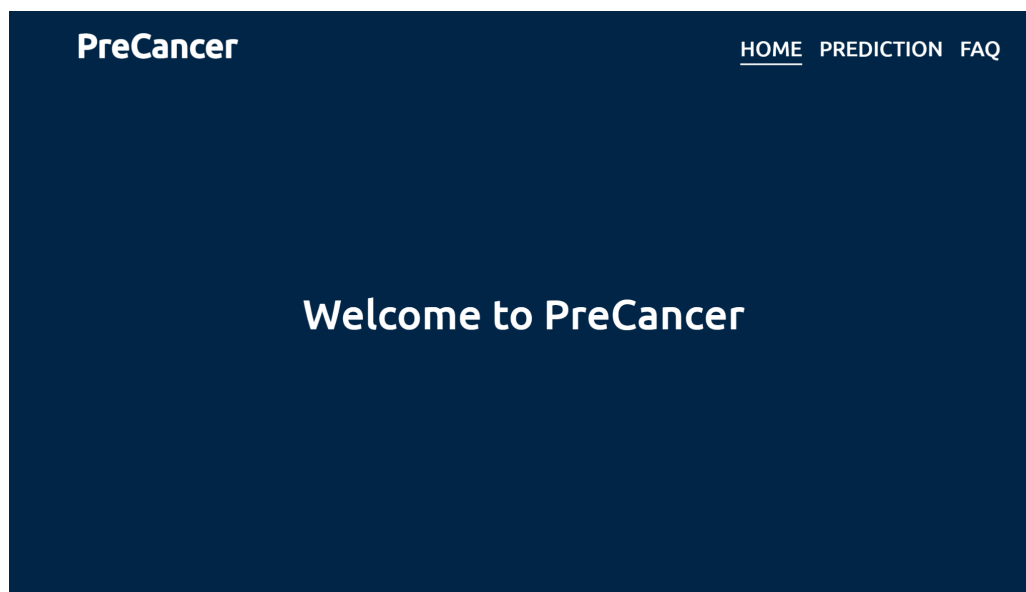


Image 4: PreCancer Homepage

3. Training the Model with User Data

This section will guide you through the process of training the machine learning model with your own data. To begin, you will need to navigate to the 'Prediction' page of the web application, where you can upload a CSV file containing your data. Once your data has been

uploaded, the model will use it to train and generate predictions based on your input. By following the steps outlined in this section, you can quickly and easily train the model with your own data and utilise its predictive capabilities to gain insights and make informed decisions.

3.1 Navigate to the Prediction page

To navigate to the 'Prediction' page, there are two ways to proceed. Firstly, you can scroll down from the Home Page and click on the button labelled 'Let's Get Started'.

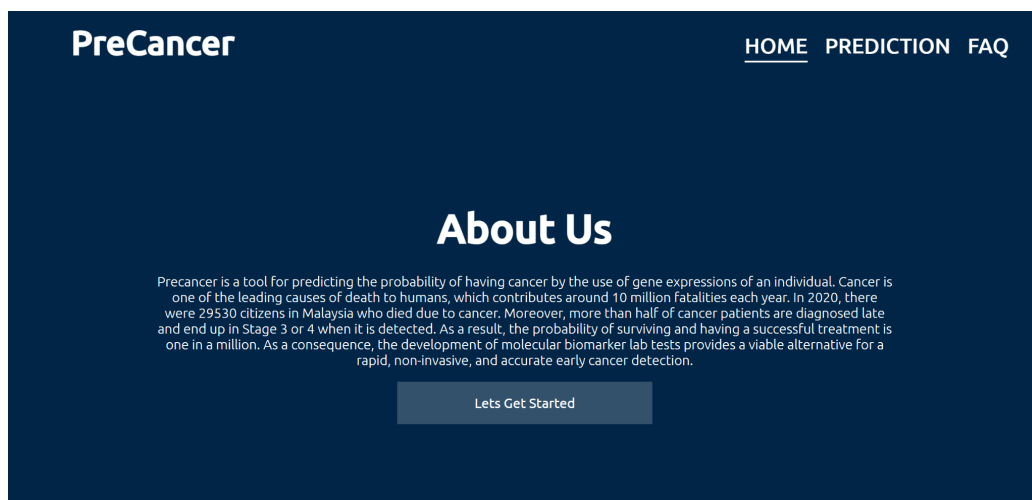


Image 5: PreCancer Homepage scroll down to About Us

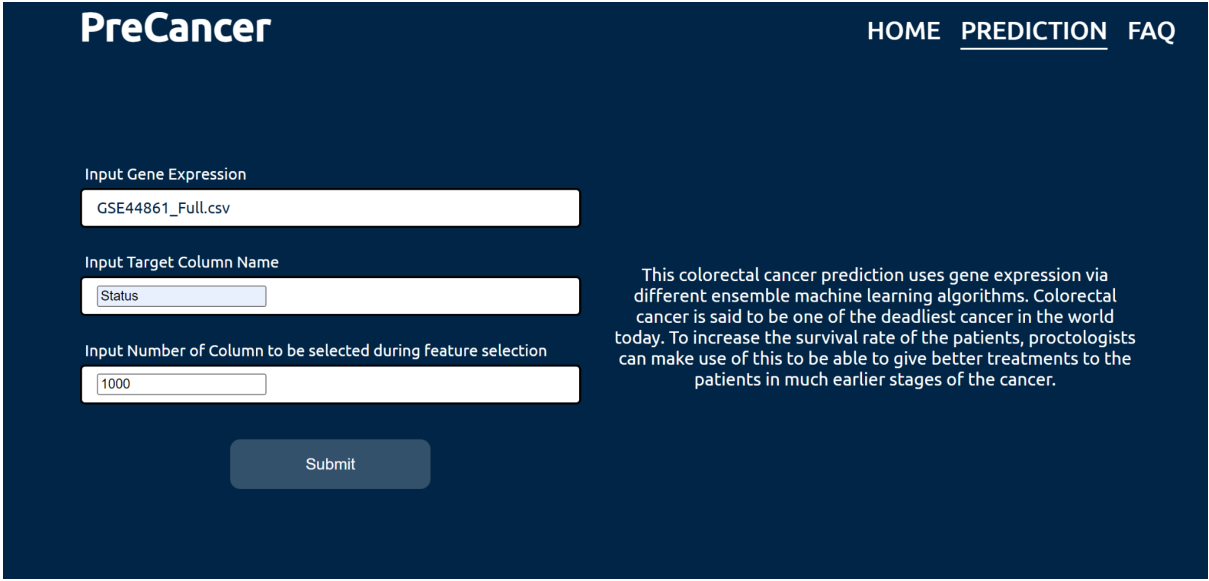
Alternatively, you can click on the 'Prediction' option that appears in the navigation bar on the top right corner of the website.

The image shows the 'Prediction' page of the PreCancer website. The header is identical to the previous image, with 'PreCancer' on the left and 'HOME', 'PREDICTION' (underlined), and 'FAQ' on the right. The main content area has a dark blue background. On the left, there are three input fields: 'Input Gene Expression' with a text box containing '.CSV' and an upload icon; 'Input Target Column Name' with a text box containing 'e.g., Status'; and 'Input Number of Column to be selected during feature selection' with a text box containing 'e.g., 100'. Below these fields is a dark blue 'Submit' button. On the right side of the page, there is a paragraph of text: 'This colorectal cancer prediction uses gene expression via different ensemble machine learning algorithms. Colorectal cancer is said to be one of the deadliest cancer in the world today. To increase the survival rate of the patients, proctologists can make use of this to be able to give better treatments to the patients in much earlier stages of the cancer.'

Image 6: PreCancer Prediction Page

3.2 Configuring the Input Parameters

After accessing the 'Prediction' page and following the instructions to upload their CSV file, users can input the necessary gene expression data, target column name, and select the number of columns to be used during the feature selection process. Once these input parameters have been configured, the web application will display a confirmation page summarising the selected parameters. An example of what this page looks like is shown in the following figure.



The screenshot shows the 'PreCancer' web application interface. At the top, there is a navigation bar with 'HOME', 'PREDICTION' (which is underlined), and 'FAQ'. The main content area has a dark blue background. On the left, there are three input fields: 'Input Gene Expression' with the value 'GSE44861_Full.csv', 'Input Target Column Name' with the value 'Status', and 'Input Number of Column to be selected during feature selection' with the value '1000'. Below these fields is a 'Submit' button. To the right of the input fields, there is a text block that reads: 'This colorectal cancer prediction uses gene expression via different ensemble machine learning algorithms. Colorectal cancer is said to be one of the deadliest cancer in the world today. To increase the survival rate of the patients, proctologists can make use of this to be able to give better treatments to the patients in much earlier stages of the cancer.'

Image 7: PreCancer Prediction Page with inputs

3.3 Handling Input Errors

Although the web application is designed to be user-friendly, errors may still occur when submitting input parameters. This section outlines how to handle errors that may arise during the process. It's important to note that any error messages will only appear after clicking the 'Submit' button.

3.3.1 Handling CSV Input Errors

During CSV file upload, two error messages may appear. The first error message appears when no CSV file is selected for upload. The second error message appears when a file that is not in CSV format is selected. Below are examples of these error messages:

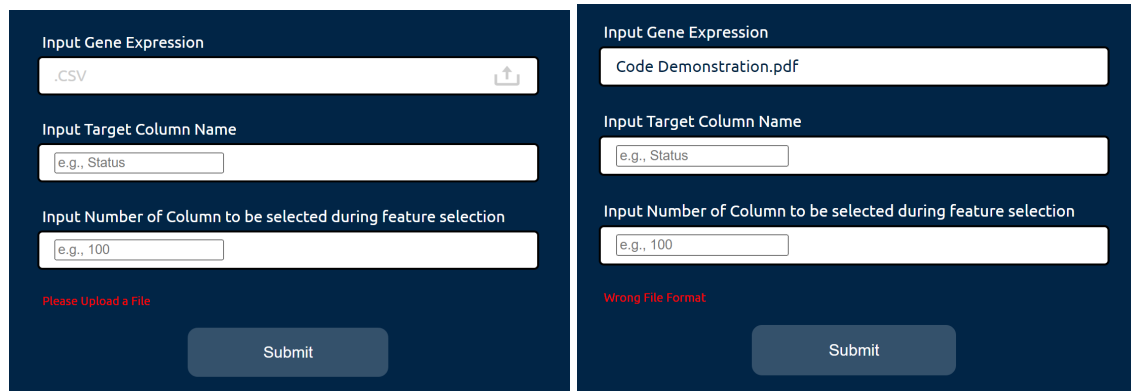


Image 8 and 9: Prediction Page no input error and wrong input error

Please double check that you have chosen the right file and that it is in CSV format if you encounter one of these issues. Please refer to the error message for more detailed advice if you're still having issues.

3.3.2 Handling Target Column Name Errors

After the user inputs the target column name and clicks the 'Submit' button, a validation check is performed to ensure that the target column name exists in the dataset. If the provided target column name does not exist, an error message will pop up, indicating that the input is invalid. The following figure illustrates an example of this situation.

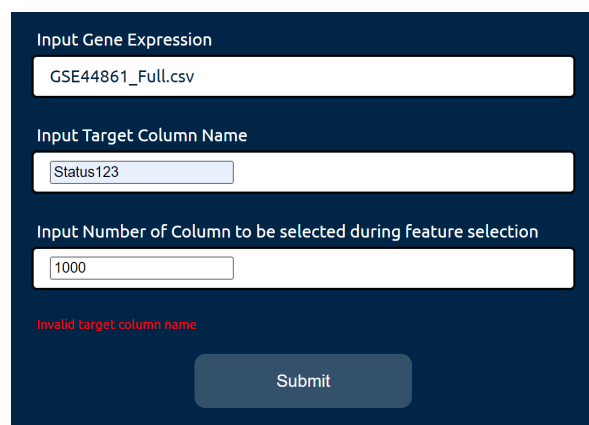


Image 10: Prediction Page wrong input error

Verify that the provided column name exactly matches the column name in the dataset if the target column name error message displays. You can verify this by opening the CSV file and comparing the header with the target column name inputted.

3.3.3 Handling Feature Selection Input Errors

If the user provides a number of columns to be selected during feature selection that exceeds the number of available columns (excluding the ID and Target columns), an error message will be displayed. The example below illustrates this situation.

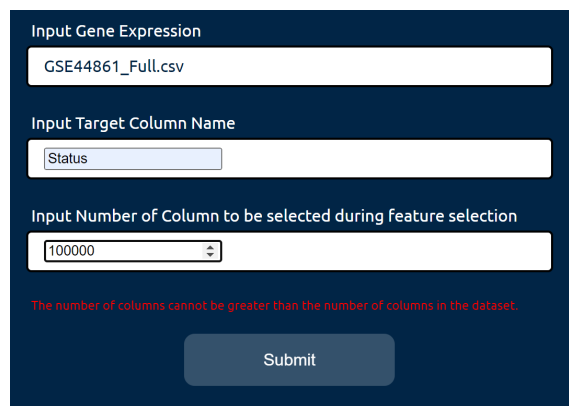
The image shows a web form titled "Prediction Page" with a dark blue background. It contains three input fields: "Input Gene Expression" with the value "GSE44861_Full.csv", "Input Target Column Name" with the value "Status", and "Input Number of Column to be selected during feature selection" with the value "100000". Below the third field, a red error message reads: "The number of columns cannot be greater than the number of columns in the dataset." At the bottom of the form is a grey "Submit" button.

Image 11: Prediction Page wrong input error

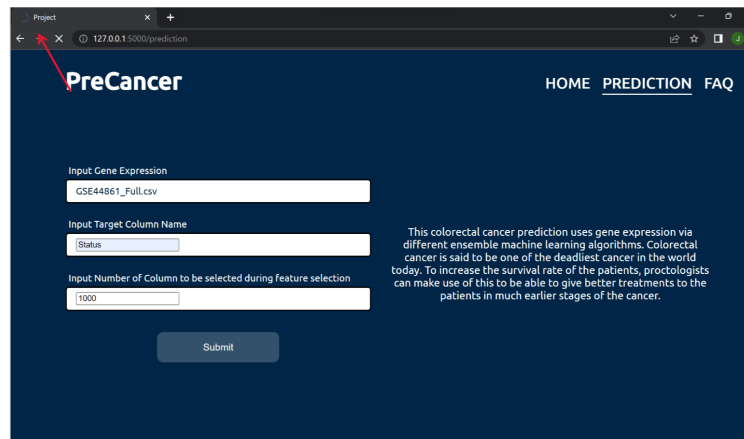
The user can resolve this error by entering a valid number of columns for feature selection that is less than or equal to the number of columns in the dataset.

3.4 Training the Model

When the user has accurately input all of the required parameters, they can train the model by clicking the "Submit" button. The software will process the provided data along with the chosen parameters to produce the predicted result. After that, users are taken to the results page, where they may see the predicted outcome. Depending on the size of the input data, this process could take a while. Please wait it out and don't close the browser while this is happening.

When training the model, it is important to note that the time required for training can vary depending on the number of features or columns in the input data. As the number of features increases, the training process may take longer to complete. This means that the time it takes to train the model is somewhat unpredictable and can be influenced by the complexity and size of the data.

The image below demonstrates the correct input parameters for the model to start training. Once all parameters are correctly provided, click on the 'Submit' button and wait for the model to start training. A loading symbol will appear, as indicated by a red arrow in the image, to signify that the model is currently being trained.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/prediction'. The page has a dark blue header with the 'PreCancer' logo on the left and navigation links 'HOME', 'PREDICTION', and 'FAQ' on the right. The main content area is white and contains three input fields: 'Input Gene Expression' with the value 'GSE44861_Full.csv', 'Input Target Column Name' with the value 'Status', and 'Input Number of Column to be selected during feature selection' with the value '1000'. A 'Submit' button is located below these fields. To the right of the input fields, there is a paragraph of text: 'This colorectal cancer prediction uses gene expression via different ensemble machine learning algorithms. Colorectal cancer is said to be one of the deadliest cancer in the world today. To increase the survival rate of the patients, proctologists can make use of this to be able to give better treatments to the patients in much earlier stages of the cancer.' A red arrow points to the 'Submit' button.

Image 12: Prediction Page loading tab

4. Interpreting the Result Page

In this section, we will provide an overview of the information displayed on the result page after the training of our proposed bagging algorithm. This includes the accuracy, specificity, sensitivity, and AUC of the model. Additionally, we will explain the three buttons available on the page: "Predict again", "Download Preprocessed Dataset", and "Predict Using Trained Model". Further details on the functionality of each button will be provided in the following subsections.

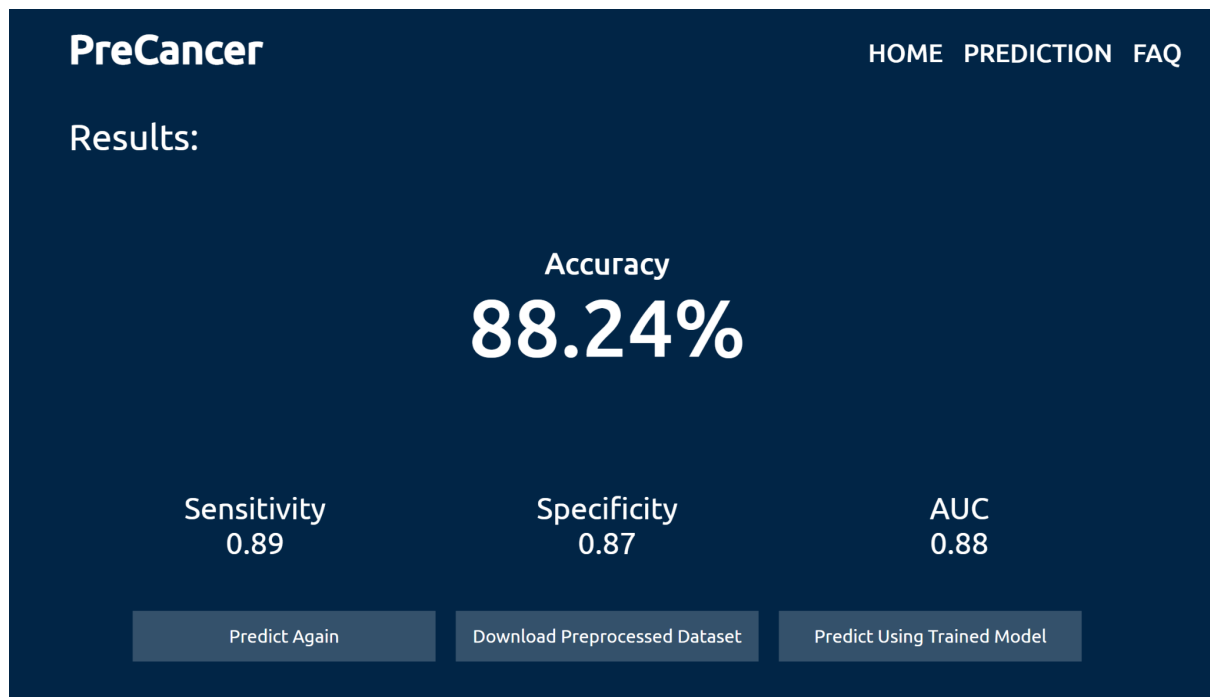


Image 13: Results

4.1 Performance Metrics

Once the result page is loaded, the user will be able to view the performance metrics of our ensemble machine learning model, which include accuracy, specificity, sensitivity, and AUC. These metrics indicate the model's overall performance and can be used to rate the reliability of the predictions made by the model.

4.2 Functionalities of the Buttons on the Result Page

Here are the explanation for the functionalities of the buttons on the result page:

Predict again: By selecting the "Predict again" button, the user may input new parameters and retrain their model using a different dataset. This feature enables the user to train numerous models for comparison while experimenting with various input setups.

Download Preprocessed Dataset: The user can obtain the preprocessed dataset that was used to train the model by clicking the "Download Preprocessed Dataset" button. Researchers who want to use the preprocessed dataset to train their own models or conduct further studies will find this capability to be especially helpful.

Predict Using Trained Model: The user will be taken to a different page where they can upload a new dataset and use the trained model to generate predictions after clicking the "Predict Using Trained Model" button. Users who wish to analyse a new dataset without a class label but are unwilling to completely retrain the model can benefit from this feature.

5. Performing Predictions with Trained Model

If the user clicks on the 'Predict Using Trained Model' button on the result page, they will be redirected to a new page that allows them to input a dataset without class labels. The user can then get the predicted labels for each row of this dataset using the trained model. The only input required on this page is the dataset itself, and the error handling for the input CSV file is the same as in the 'Prediction' page, as mentioned earlier.

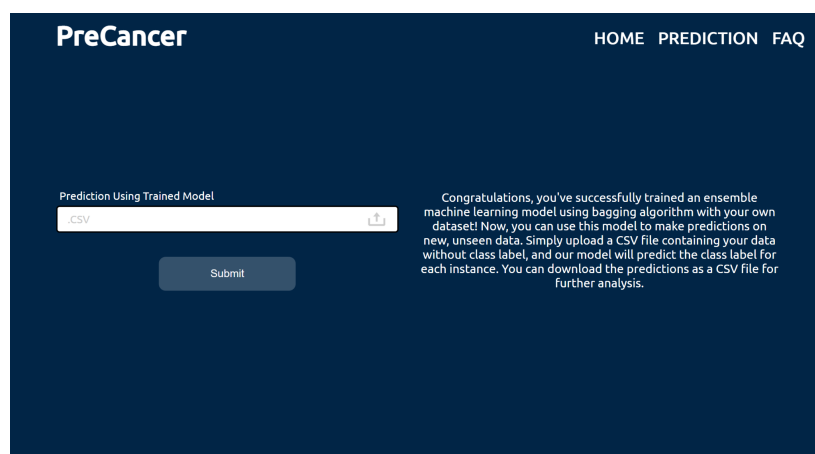


Image 14: Prediction using trained model

After clicking the "Submit" button, the prediction process will start and once it is completed, the page will display three buttons. The **"Make a new prediction again"** button allows users to upload another dataset without class labels for new predictions. The **"Download predicted class label"** button allows users to download a CSV file with predicted labels and IDs. Finally, the **"Train new model"** button redirects users back to the 'Prediction' page where they can input a new CSV, target column name, and number of columns to be selected during the feature selection process to train a new model.

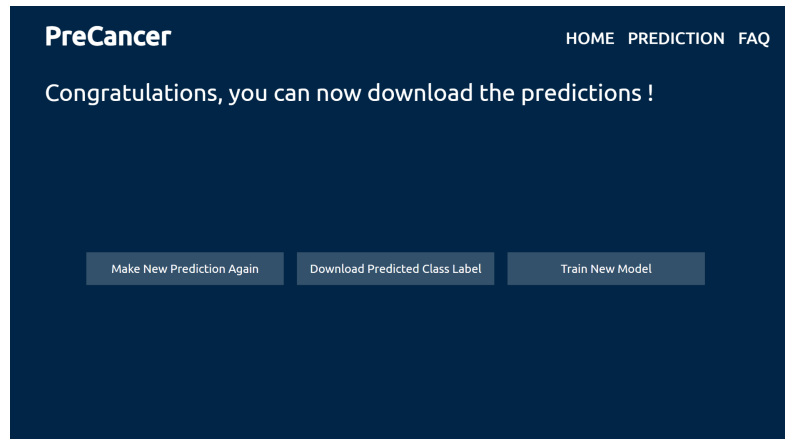


Image 15: Predictions can be downloaded

Technical Guide

This technical guide section shows the required hardware and software for the project.

1. Recommended hardware

To achieve optimal results across our entire workflow, it is advised to have hardware specifications similar to those presented in Table 1 and Table 2 for Window and MacOS respectively. This recommendation is based on our extensive use of this hardware configuration for various tasks, including preprocessing, training models, conducting tests, and operating our graphical user interface (GUI).

Window:

Hardware Component	Hardware Specification
Central Processing Unit (CPU)	<ul style="list-style-type: none">• 9th Gen Intel(R) i3 Processor and Above• 2.4GHz and Above
Random Access Memory	16 GB
Hard Disk Drive	750 MB available space
Operating System	Window 7+ with 64-bit

Table 1: Window Hardware Specification

MacOS:

Hardware Component	Hardware Specification
Central Processing Unit (CPU)	<ul style="list-style-type: none">• Intel Core Duo Processor and Above• 2.4GHz and Above
Random Access Memory	16 GB
Hard Disk Drive	750 MB available space
Operating System	MacOS X 10.11 (El Capitan) or later

Table 2: MacOS Hardware Specification

2. Software Requirement

Below is a list of the main software required to run our project program:

1. Python 3.8 and above
<https://www.python.org/downloads/>
2. PyCharm
<https://www.jetbrains.com/pycharm/download/>
3. Browser:
<https://www.google.com/chrome/> (Google Chrome)
<https://www.mozilla.org/en-US/firefox/new/> (Firefox)

3. Setup

To set up this project, please follow the instructions below:

1. Install the required softwares mentioned above.
2. Clone the project repository from the following URL:
<https://github.com/rnawar/PreCancer>
3. On the terminal from PyCharm, run the following command to install required libraries
`pip install -r requirements`
4. Run the “main.py” file in PyCharm as explained in Section 1
5. Please note that the datasets used for testing our project are included in the repository. These are mainly Colorectal Cancer datasets.
6. With the setup completed, you can now proceed to run the program as instructed in the End User Guide's Section 1 to verify that all software dependencies are correctly configured.

4. Project Structure

Our project has been compiled in the structure below:

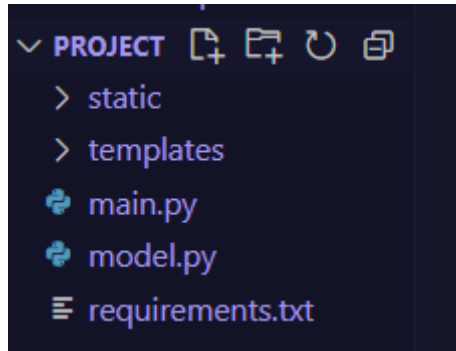


Image 16: Project Structure

For the **static** file, it contains folders and files that will not change when running the project which are the CSS and Javascript files.

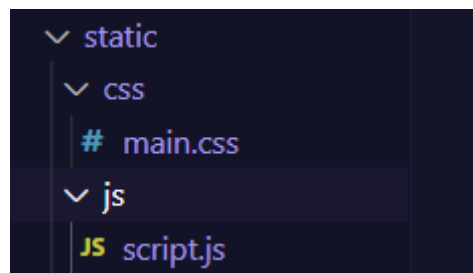


Image 17: Static Folder

Templates folder stored all the webpage templates and design in the format of HTML.

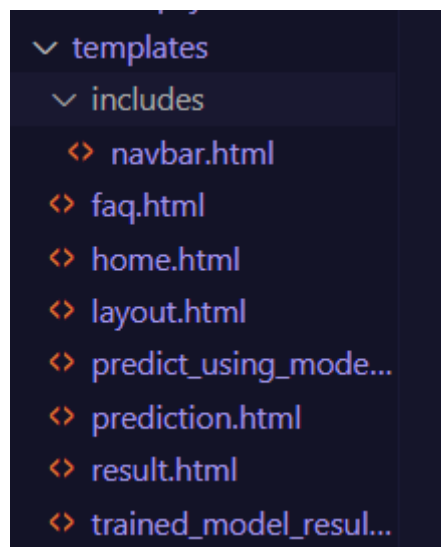


Image 18: Templates Folder

The **requirement.txt** is the file that contains libraries that the user/developer will need to install before running the project.

model.py contains our ensemble machine learning model and functions that integrate into our webpage.

main.py is the file that the user/developer uses to startup the local server and access to the webpage. More information refer to the [End User Guide](#).

5. Version Control

5.1 User

Any updates from our project will always be pushed to GitHub. The updates include bug fixes, model improvements, user interface adjustments, and more. Users will need to pull the folder from our project's GitHub repository to experience the latest features. However, users are not able to push their modified files into the GitHub repository because the repository is only available for developers.

5.2 Developer

Developers will always need to pull files from GitHub to avoid any overwrite issues. They also need to ensure that merging or committing the updated files is feasible before pushing them into the repository, ensuring that the webpage remains functional for users. The push messages from developers need to be concise and clear so that other developers can understand the changes made to the project.

6. Miscellaneous

If any issues occur while using our product, kindly follow the suggested methods for troubleshooting:

1. Look for error codes in the terminal or error messages on the webpage.
2. Verify that the uploaded file is in the appropriate file format.
3. Ensure that the program is running continuously.
4. Seek solutions for encountered problems from websites such as Stack Overflow, Google, etc.

Please contact us and report the issue if the above suggestions are unable to resolve the problem you are facing.

- Email:
 - jkua0008@student.monash.edu (Jia Chen)
 - hlim0033@student.monash.edu (Han Wei)
 - rnaw0003@student.monash.edu (Samantha)