FIT3164 Semester 1, 2024:
User Guides

Team: MDS02

Liaw Yi Hui (32023707)
Chan Jia Xin (31859089)
Pang Eason (32024584)

Table of Contents

1 End User Guide

1.1 Lymphedema Prediction Web Page Prototype Layout

Figure 1 portrays the Home Tab of the Lymphedema prediction web page prototype developed by our team, where it houses some useful and background information about the Lymphedema disease, and also the brief introduction to the steps in utilizing the main Lymphedema prediction tool in the Prediction Tab (Figure 2 & 3). Moving on, inside the Prediction Tab of the web page prototype as shown in Figure 2 & 3, it is where our main Lymphedema prediction tool sits, where uploading and downloading datasets, prediction results and outcome, datasets' visualizations as well as prediction results' visualizations will be shown in this particular tab. Furthermore, inside the About Model Tab of the web page prototype as we can see in Figure 4, it actually has the introduction on how we have achieved our Lymphedema predictive model, the ROC curve of the predictive model, the predictive model performance on the testing dataset, as well as the relative variable importance of each feature or variable inside the dataset that we have utilized to train our Lymphedema predictive model. All of the functions or purposes of buttons and menu options in all three figures or tabs will be explained in Section 1.4, including the steps on utilizing each of them.

1.1.1 Home Tab

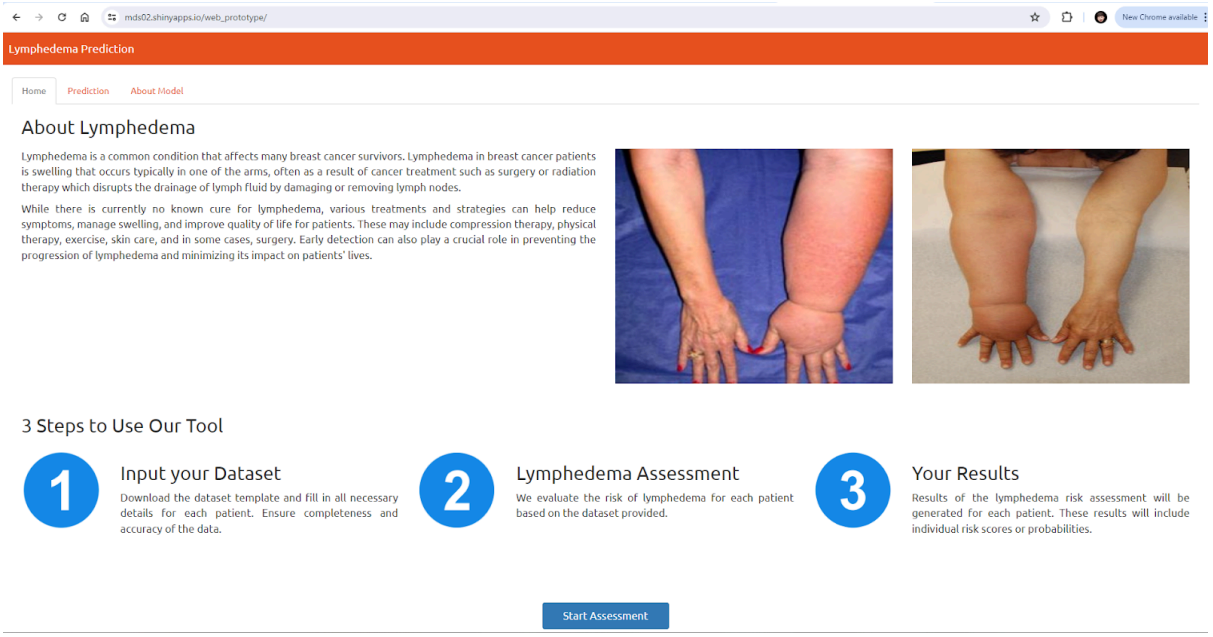


Figure 1: Lymphedema prediction web page prototype layout (Home Tab)

1.1.2 Prediction Tab

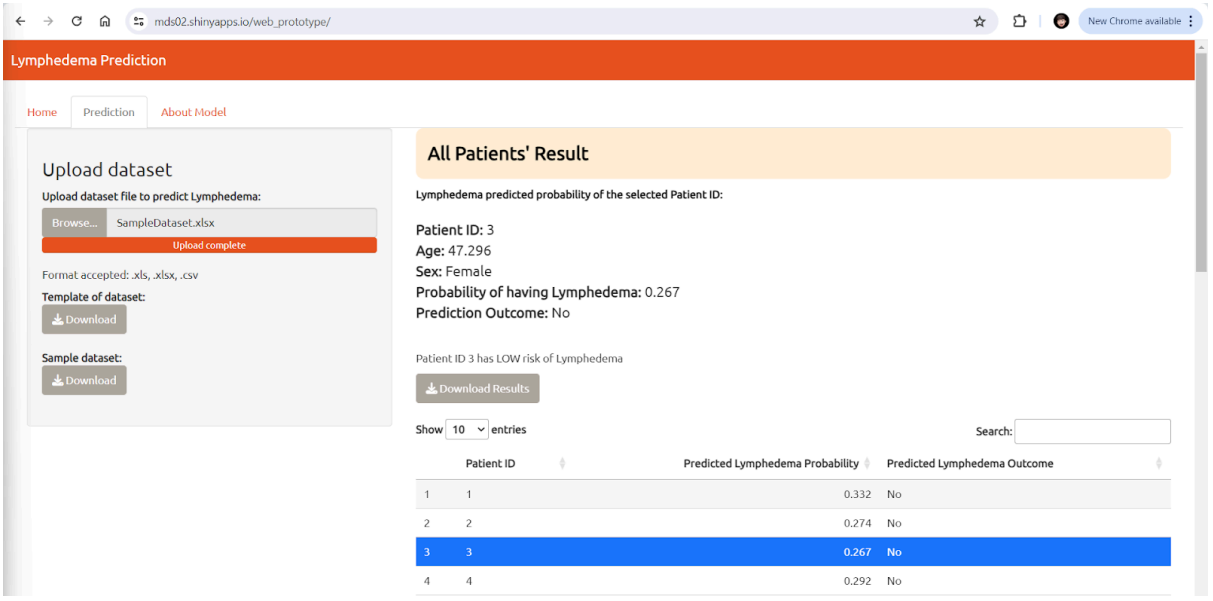


Figure 2: Lymphedema prediction web page prototype layout part 1 (Prediction Tab)

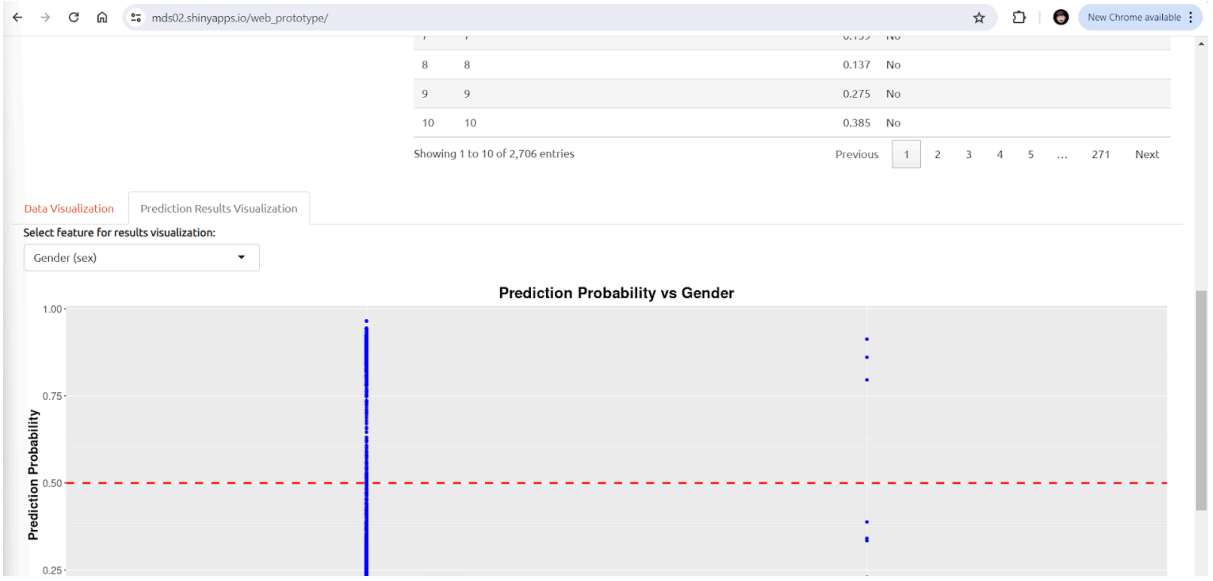


Figure 3: Lymphedema prediction web page prototype layout part 2 (Prediction Tab)

1.1.3 About Model Tab

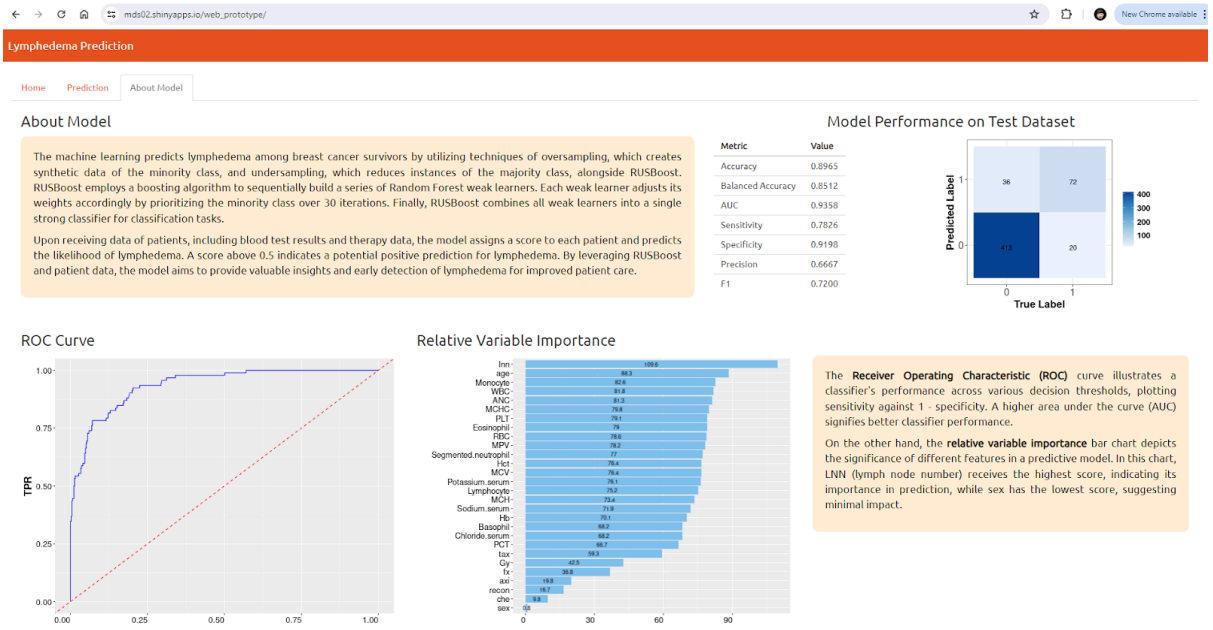


Figure 4: Lymphedema prediction web page prototype layout (About Model Tab)

1.2 Lymphedema Model Prediction Results/Outcomes

Due to the complexity in navigation of the Prediction Tab of the Lymphedema Prediction Web Page Prototype as compared to the Home and About Model Tabs, we will be delving into the Prediction Tab to explain in detail about each of its components or sections. Inside the Prediction Tab, it consists of 5 components or sections, which collectively allow users to upload datasets for prediction, read and understand prediction results, and also perform various data visualizations on the input dataset and prediction results. The purposes of each component or section in the Prediction Tab are included below in Table 1.

Component/Section	Figure	Purpose
Upload Dataset	5	Allow users to browse their local devices to upload their own dataset to perform Lymphedema prediction, in the format of either .xls, .xlsx and .csv. A template of dataset (i.e., empty dataset) will also be available for download to guide users in inputting their own dataset adhering to the requirements of the accepted input dataset, such as sequence of columns and format of values inputted. Definition of each feature or column will also be provided inside the template of dataset. If a user does not have his/her own dataset, we also provided a sample dataset to be downloaded, so that the user can still try out our Lymphedema prediction web page prototype.
All Patients' Result Information Box	6	Patients' details such as Patient ID, age, sex, probability of having lymphedema and the prediction outcome will be summarized and shown here. There is also a download button for downloading the Lymphedema prediction results in .xlsx format after making the prediction.
All Patients' Result Table	7	Display all of the predicted lymphedema probabilities and the predicted lymphedema outcome for each patient or Patient ID. Searching through the Patient ID, the probability and the outcome using the search box is also allowed. The size or number of showing entries of the result table could also be altered, default will be 10.
Input Dataset Visualization	8 & 9	Portraying the number of patients or people (count) and percentages of each feature or column in the input dataset. Provide a good visual representation of the distribution using histograms and pie charts.

Prediction Results Visualization	10	Portraying the relationship between the Lymphedema prediction probability against each feature or column using a scatter plot. With a horizontal dotted red line at probability of 0.5, which is the cut-off point between the prediction outcomes Yes and No, we can observe the number of patients and groupings for each Yes and No outcome. With that, it provides users a good visual representation to visualize which value or category of one particular feature or column has contributed to the most or least Lymphedema prediction probability, respectively for each feature and column.

Table 1: Purpose of each component or section in the Prediction Tab

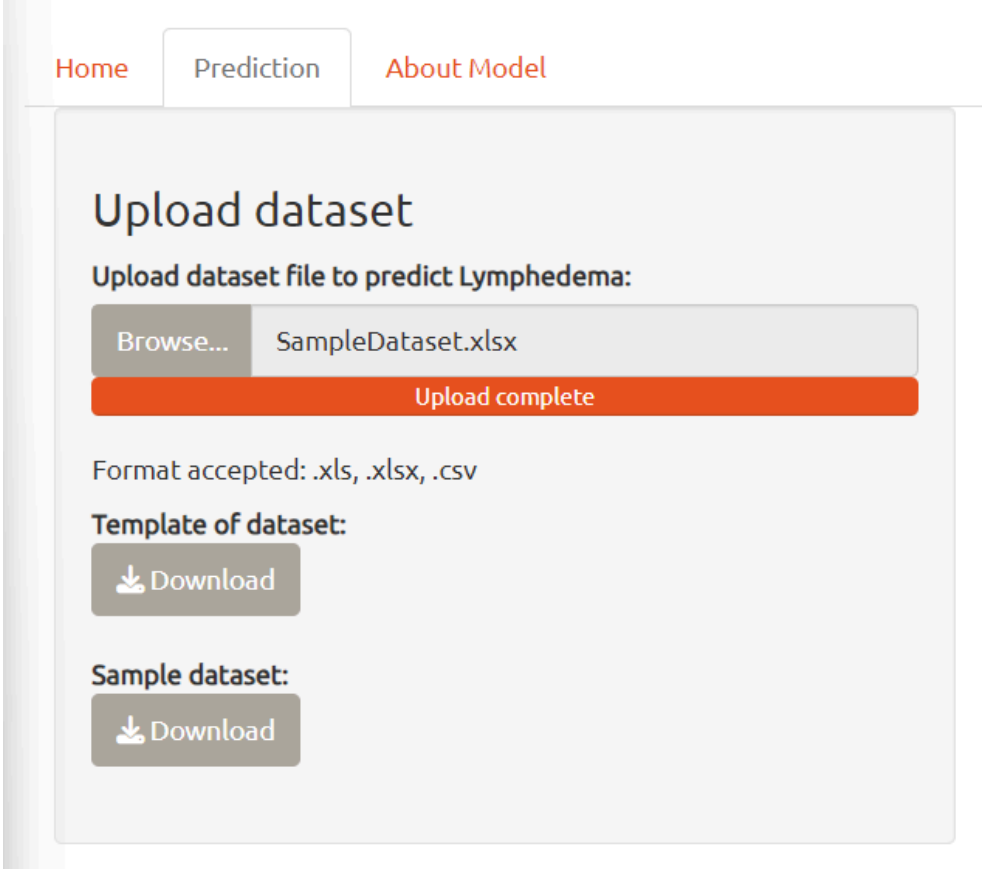


Figure 5: Upload dataset component/section

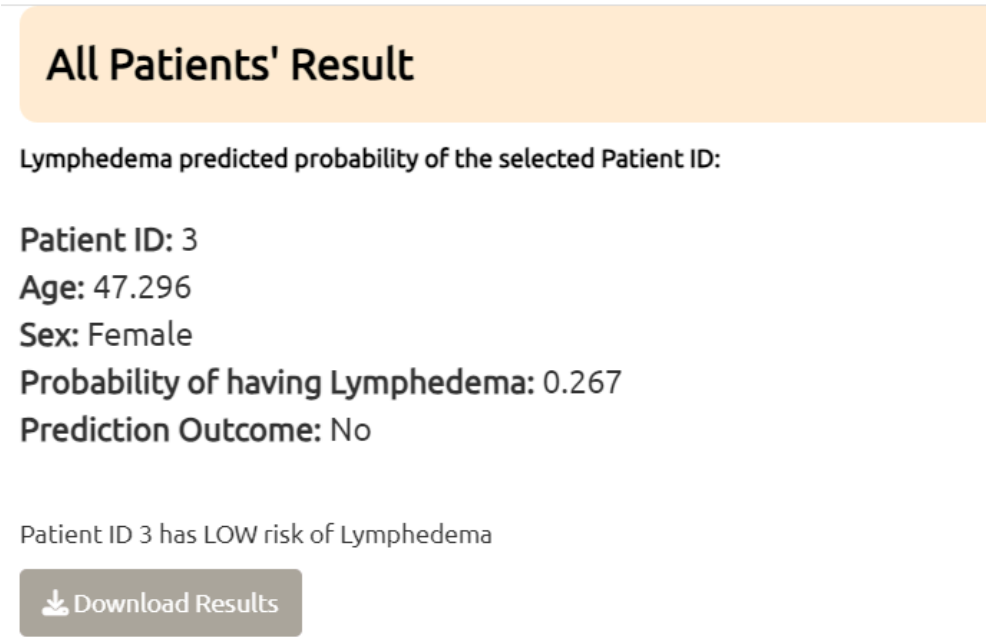


Figure 6: All patients' result information box component/section

Show 10 entries Search:

	Patient ID	Predicted Lymphedema Probability	Predicted Lymphedema Outcome
1	1	0.332	No
2	2	0.274	No
3	3	0.267	No
4	4	0.292	No
5	5	0.295	No
6	6	0.139	No
7	7	0.139	No
8	8	0.137	No
9	9	0.275	No
10	10	0.385	No

Showing 1 to 10 of 2,706 entries Previous 1 2 3 4 5 ... 271 Next

Figure 7: All patients' result table component/section

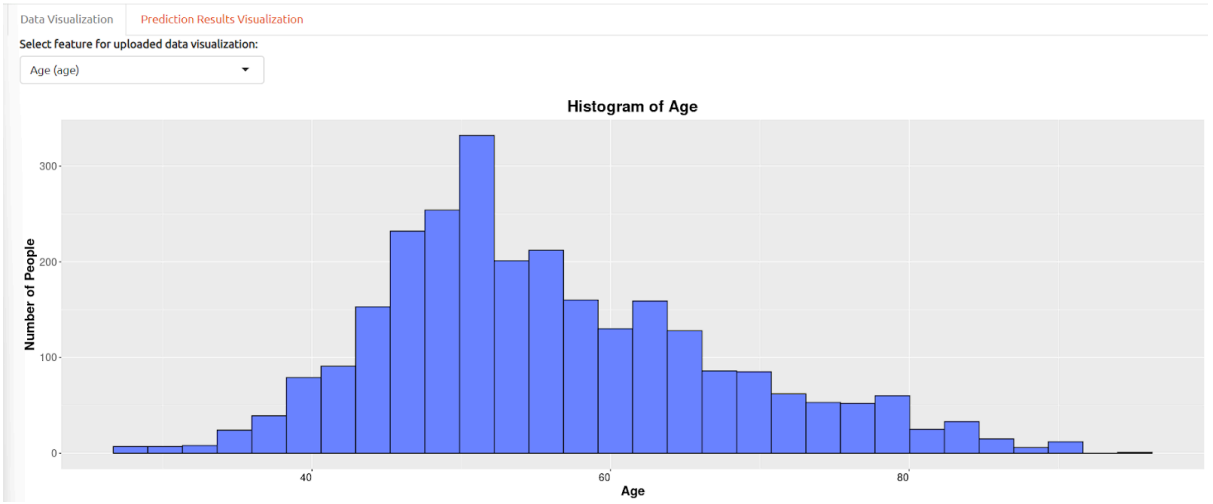


Figure 8: Input dataset visualization component/section (histogram)

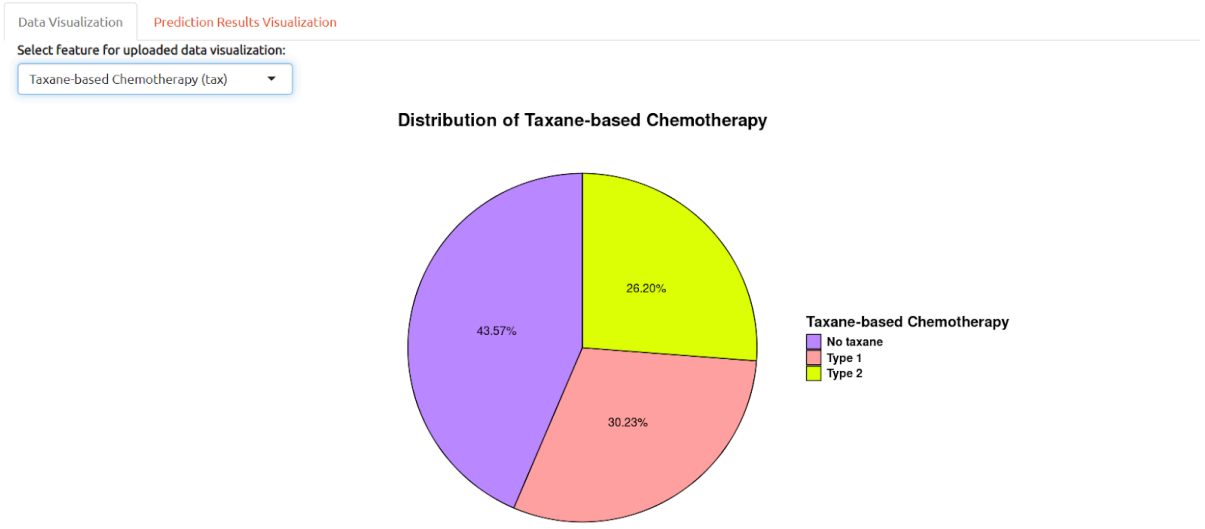


Figure 9: Input dataset visualization component/section (pie chart)

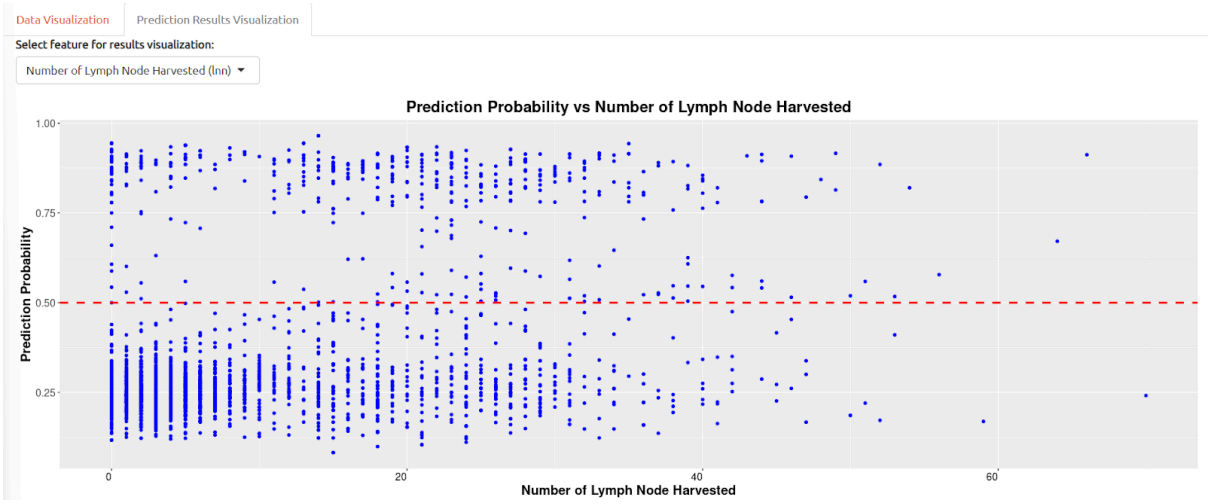


Figure 10: Prediction results visualization component/section (scatter plot)

1.3 How to Access the Lymphedema Prediction Web Page Prototype

In order to access the Lymphedema Prediction Web Page Prototype, it is as straightforward as accessing any other web pages. You can access the web page prototype by browsing this link:

https://mds02.shinyapps.io/web_prototype/

in your preferred web browser's address bar, such as Google Chrome, Microsoft Edge, Safari, etc. Click enter and you will be redirected automatically to the Lymphedema Prediction Web Page Prototype. Rest assured that your experience with the prototype is secure and private as we do not retain any data or personal information of the users during the browsing session.

1.4 Lymphedema Prediction Web Page Prototype Use Menus

There are several menu option buttons across all three tabs in the Lymphedema Prediction Web Page Prototype. The functions, purposes and usage steps description of each menu option button are included below in Table 2.

Menu Option Button	Function/Purpose	Step to Use
Home	Redirect to the Home Tab where the brief information about Lymphedema and the steps to utilize the prediction tool is at.	Click on the "Home" button at the top of the web page, you will be redirected to the Home Tab automatically.
Prediction	Redirect to the Prediction Tab where the main Lymphedema prediction tool is at.	Click on the "Prediction" button at the top of the web page, you will be redirected to the Prediction Tab automatically.
About Model	Redirect to the About Model Tab where the description, performance and the relative variable importance about the prediction model that we have trained and integrated is at.	Click on the "About Model" button at the top of the web page, you will be redirected to the About Model Tab automatically.
Start Assessment	Redirect to the Prediction Tab where the main Lymphedema prediction tool is at.	Click on the "Start Assessment" button at the bottom of the Home Tab, you will be redirected to the Prediction Tab automatically.
Browse	Allow users to upload dataset that will be transferred to our backend predictive model to be utilized for the Lymphedema prediction.	Click on the "Browse" button in the upload dataset section in the Prediction Tab, select the dataset file to be uploaded from your local device/machine and click open. Note that the accepted format of the dataset file uploaded will be .xls, .xlsx and .csv only.

Download Template of Dataset	Allow users to download an empty dataset with predefined features or columns, that rows are to be filled in later with the users' own data records. The definition of each feature or column is also included in the downloaded template of dataset.	Click on the "Download" button below "Template of dataset" in the upload dataset section in the Prediction Tab, the template of dataset will be downloaded to your local device/machine automatically.
Download Sample Dataset	If users do not have their own Lymphedema dataset or data records, they can download this sample dataset to try out the Lymphedema prediction tool.	Click on the "Download" button below "Sample dataset" in the upload dataset section in the Prediction Tab, the sample dataset will be downloaded to your local device/machine automatically.
Download Results	After the Lymphedema prediction is performed and completed, users have the option to download the results of the prediction made where the results will be appended to the users' original input dataset.	Click on the "Download Results" button in the All Patients' Result section in the Prediction Tab, the results of the prediction made will be downloaded to your local device/machine automatically.
Show Entries	Allow users to select the number of entries to be shown in one page of the prediction results table.	Select between 10, 25, 50 and 100 entries above the prediction results table in the All Patients' Result section in the Prediction Tab, for the number of entries to be shown in the prediction results table.
Search Field	Allow users to search numeric or non-numeric values (i.e., column "Patient ID", "Predicted Lymphedema Probability" and "Predicted Lymphedema Outcome").	Directly type into the search field box above the prediction results table in the All Patients' Result section in the Prediction Tab for either numeric or non-numeric values to be searched (e.g., "25" or "Yes").

Sorting Arrows in the Prediction Results Table	Allow users to sort the values of each column inside the prediction result table regardless that they are numeric or non-numeric.	Click on the small arrow beside each column in the prediction results table in the All Patients' Result section in the Prediction Tab to sort the columns either in ascending or descending order. This applies for all three columns inside the prediction results table (i.e, "Patient ID", "Predicted Lymphedema Probability" and "Predicted Lymphedema Outcome").
Individual Row inside the Prediction Results Table	Allow users to manually select the individual row or patient inside the prediction results table, so that the individual detail of each patient will be shown inside the All Patients' Result Information Box.	Select or click directly on the specific/individual row or patient that the user wishes to observe inside the prediction results table in the All Patients' Result section in the Prediction Tab, the selected row or patient will turn blue after the selection is made, the individual detail of each patient will also be shown inside the All Patients' Result Information Box.
Data Visualization	Allow users to perform data visualization on the input dataset, such as the distribution, number of patients and percentages of each feature or column, by using histograms or pie charts.	Click on the "Data Visualization" button in the data visualization section in the Prediction Tab, and select manually using the drop down bar for the feature in the input dataset to be visualized.
Prediction Results Visualization	Allow users to perform data visualization on the results of the prediction made, such as the relationship between the Lymphedema prediction probability against each feature or column, by using scatter plots.	Click on the "Prediction Results Visualization" button in the data visualization section in the Prediction Tab, and select manually using the drop down bar for the feature in the input dataset to be visualized against the prediction probability.

Table 2: Function and usage step description of each menu option button

1.5 General Steps on Utilizing the Lymphedema Prediction Tool

1. Access the Lymphedema Prediction Web Page Prototype at *https://mds02.shinyapps.io/web_prototype/*
2. You can read about Lymphedema and also the brief steps to utilize the Lymphedema Prediction Tool, in the Home Tab of the web page prototype.
3. Click on the "Start Assessment" button in the Home Tab, that will direct you automatically to the Prediction Tab.
4. Click and download the template of dataset and fill in the patients' records data according to the example and definition of each feature or column given inside the template of dataset. If you do not have your own patients' records data but you wish to try out our Lymphedema prediction tool, skip this step and move on to Step 4.
5. If you have completed Step 3 (i.e, have your own patients' records data to be filled into the template of dataset), ignore this step and move onto Step 5. If not, click and download the sample dataset provided.
6. Click on the "Browse" button to select and upload the dataset from your local device or machine that you have gotten from either Step 3 or 4 above, so that it will be transferred to our backend predictive model for the prediction of Lymphedema to be carried out.
7. You can sort, search, select individual rows or patients in the prediction results table, by referring to the usage description in Section 1.4 of this document above.
8. You can also download the prediction results by clicking on the "Download Results" button on top of the prediction results table.
9. You can also perform some data visualizations on the input dataset itself, or even on the prediction results, by hovering to the section below the prediction results table. Detailed usage steps were also documented in Section 1.4 of this document above.
10. You can then read about the description, performance and the relative variable importance of the predictive model that we have trained and integrated onto the web page prototype to perform the Lymphedema prediction, by clicking on the About Model Tab, that will then direct you to that particular tab.
11. You can exit the Lymphedema Prediction Web Page Prototype by clicking on the "X" at the top of your browser to close your browser as you normally would, and all data associated with your session will be automatically deleted once it ends.

1.6 Input Data Requirements

There are some specific requirements that the input data must meet. Firstly, it should be provided in one of the accepted formats, including .xls, .xlsx, or .csv, allowing for compatibility with the system. Secondly, the input data must adhere strictly to the provided dataset template, with no alterations such as adding or removing columns or changing column or sheet names. Lastly, every column in the dataset must be thoroughly completed, without any missing values, guaranteeing comprehensive data coverage.

1.7 Limitations and Constraints on the Input Data

The input data is subject to certain limitations and constraints to ensure consistency and accuracy in the prediction process. Users are required to adhere to a predefined set of columns and must follow the template provided. They can only input values of their choice within fixed columns and are unable to modify column or sheet names to their liking. While this approach may restrict users' ability to customize the dataset structure according to their preferences, these constraints are essential to maintain the integrity of the dataset and facilitate seamless processing by the machine learning model.

1.8 Limitations of the Lymphedema Prediction Web Page Prototype

Limitations of the Lymphedema Prediction Web Page Prototype include interpretation challenges, as some visualizations and results generated may be difficult to interpret for users lacking a background in data science or medical knowledge, potentially hindering the understanding of the lymphedema prediction results. Additionally, the prototype may lack advanced customization capabilities, such as adjusting visualization settings or selecting specific algorithms for prediction, which could hinder users from tailoring the tool to their preferences. Lastly, there might be performance degradation under heavy loads, especially with a large number of concurrent users or requests, as the prototype is hosted on a free hosting platform.

1.9 How to Exit the Lymphedema Prediction Web Page Prototype

Exiting the Lymphedema Prediction Web Page Prototype is simple and similar to closing any other webpage. You can close your browser as you normally would, and all data associated with your session will be automatically deleted once it ends. This ensures that your privacy and information remain protected, as no data is stored after you leave the webpage. Rest assured that your experience with the prototype is secure and private.

2 Technical Guide

2.1 Software Installation

2.1.1 Git

Git is required to clone repositories and track changes in source code during software development. Download Git for your respective operating system by following the steps in the links provided in Table 3.

Operating System	Links to install Git
MacOS	https://git-scm.com/download/mac
Windows	https://git-scm.com/download/win
Linux and Unix	https://git-scm.com/download/linux

Table 3: Summary of test objectives with their respective test phases

After downloading Git, open command prompt and execute the following command to activate Git and verify its correct installation:

```
git --version
```

2.1.2 R and RStudio

R is a programming language and RStudio is an integrated development environment (IDE) for the R programming language. Both are required for training the model and developing the website of our project. Download R and RStudio for your respective operating system by following the steps in the links provided in Table 4.

Operating System	Link to install R	Link to install R
MacOS	https://cran.rstudio.com/	https://posit.co/download/rstudio-desktop/
Windows		
Linux and Unix		

Table 4: Summary of test objectives with their respective test phases

For using R in Visual Studio Code, download the extensions of R, R Debugger and R Extension Pack as shown in Figure 11.

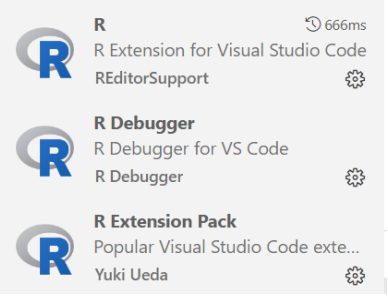


Figure 11: Extensions required for working in Visual Studio Code

2.2 Git Clone Repository and Commit Changes

2.2.1 RStudio

1. Open command prompt and set the location you want to store the folder.

   cd [PATH_TO_DIRECTORY]

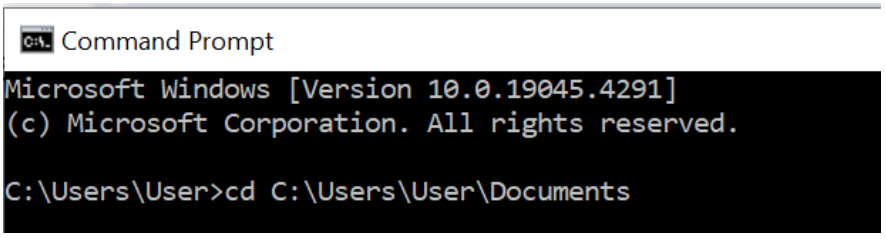


Figure 12: Code to set the location for cloning the repository

2. Clone the folder repository from GitHub.

   git clone https://github.com/liawyihui/FIT3164_software.git

Figure 13: Code to clone the repository

3. Navigate to the folder directory and pull changes from the repository before editing the files.

   git pull

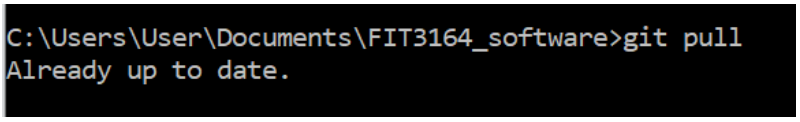


Figure 14: Code to pull changes from the repository

4. Open RStudio Desktop and navigate to the directory where you cloned the Git repository using setwd().

   setwd("[YOUR_PATH]/FIT3164_software/web_prototype")

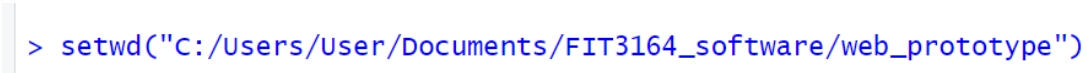


Figure 15: Code to navigate to the working directory

5. Open any file to edit or run (File > Open File > Select a file).

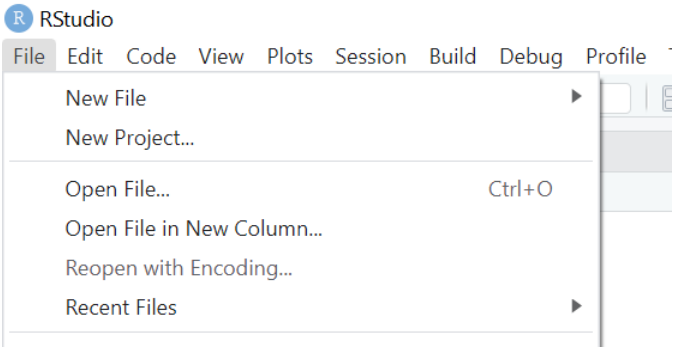


Figure 16: Steps to open a file

6. After editing the file(s), save the file and commit the changes to Git. Make sure to navigate to the directory before commiting the changes.

```
git add .
git commit -m "YOUR MESSAGE"
git push
```

```
C:\Users\User\Documents>cd C:\Users\User\Documents\FIT3164_software

C:\Users\User\Documents\FIT3164_software>git add .

C:\Users\User\Documents\FIT3164_software>git commit -m "Update comment"
[main aa78df8] Update comment
 1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\User\Documents\FIT3164_software>git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 471 bytes | 471.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/liawyihui/FIT3164_software.git
   fcb4dd1..aa78df8  main -> main
```

Figure 17: Code to commit changes to the repository

2.2.2 Visual Studio Code

1. Open a new terminal(Terminal > New Terminal) and set the location you want to store the folder.

   `cd [PATH_TO_DIRECTORY]`

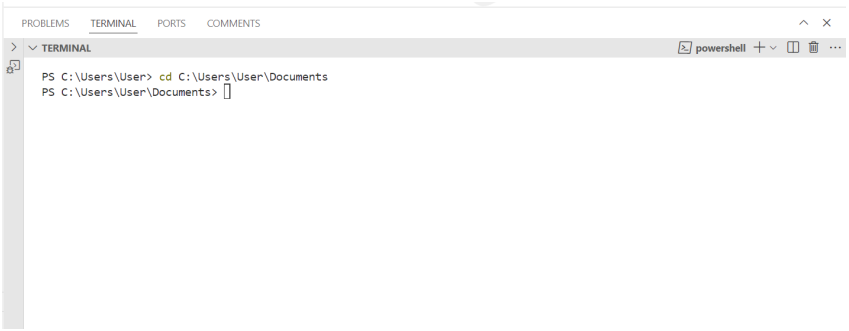


Figure 18: Code to set the location for cloning the repository

2. Clone the folder repository from GitHub.

   `git clone https://github.com/liawyihui/FIT3164_software.git`

```
PS C:\Users\User\Documents> git clone https://github.com/liawyihui/FIT3164_software.git
Cloning into 'FIT3164_software'...
remote: Enumerating objects: 961, done.
remote: Counting objects: 100% (353/353), done.
remote: Compressing objects: 100% (135/135), done.
remote: Total 961 (delta 221), reused 316 (delta 185), pack-reused 608
Receiving objects:  91% (875/961), 225.84 MiB | 10.86 MiB/s
Receiving objects: 100% (961/961), 226.46 MiB | 11.34 MiB/s, done.
Resolving deltas: 100% (614/614), done.
```

Figure 19: Code to clone the repository

3. Navigate to the folder directory and pull changes from the repository before editing the files.

   `git pull`

```
PS C:\Users\User\Documents\FIT3164_software> git pull
Already up to date.
```

Figure 20: Code to pull changes from the repository

4. Open a new R terminal (Click on the dropdown button > select ‘R Terminal’).

Figure 21: Steps to open R terminal in Visual Studio Code

5. In the R terminal, navigate to the directory where you cloned the Git repository using setwd().

```
setwd("[YOUR_PATH]/FIT3164_software/web_prototype")
```

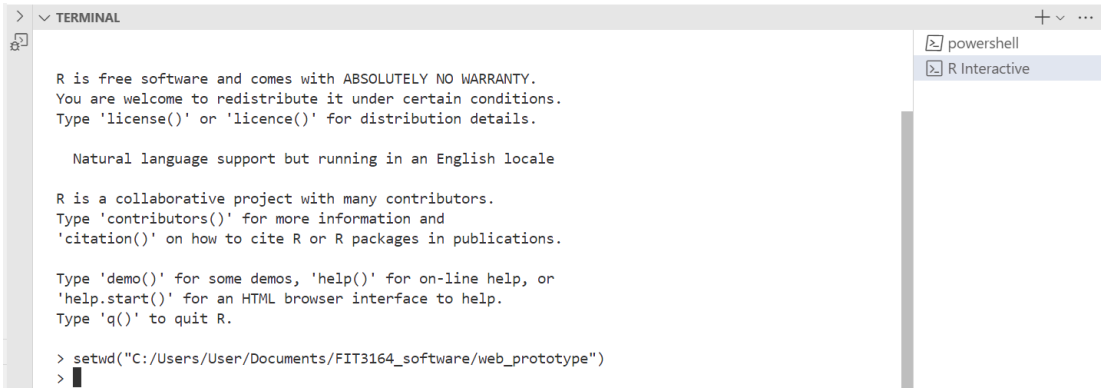


Figure 22: Code to navigate to the working directory in R terminal

6. Open the folder to edit or run files (File > Open Folder > Select the folder).

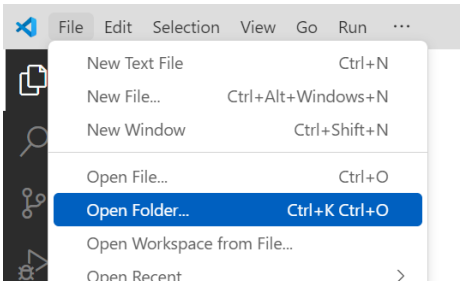


Figure 23: Steps to open the working folder in Visual Studio Code

7. After editing the file(s), save the file and commit the changes to Git. Make sure to navigate to the directory before commiting the changes.

```
git add .
git commit -m "YOUR MESSAGE"
git push
```

```
PS C:\Users\User\Documents\FIT3164_software> git add .
PS C:\Users\User\Documents\FIT3164_software> git commit -m "Update comment"
[main a1bcd78] Update comment
 2 files changed, 6 insertions(+), 4 deletions(-)
PS C:\Users\User\Documents\FIT3164_software> git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 685 bytes | 685.00 KiB/s, done.
Total 6 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To https://github.com/liawyihui/FIT3164_software.git
   aa78df8..a1bcd78  main -> main
```

Figure 24: Code to commit changes to the repository

2.3 R Packages Installation

The method to install the required packages is the same for both RStudio and Visual Studio Code.

1. Open the R terminal.

2. Install the required packages for model training.

```
packages <- c("dplyr", "ggplot2", "caret", "openxlsx",
"ROCit", "smotefamily", "ROSE", "ebmc")
install.packages(packages)
```

```
> packages <- c("dplyr", "ggplot2", "caret", "openxlsx", "ROCit", "smotefamily", "ROSE", "ebmc")
> install.packages(packages)
```

Figure 25: Code to install the required packages for model training

3. Try to run the libraries to verify the correct package installations.

```
library(dplyr)
library(ggplot2)
library(lattice)
library(caret)
library(openxlsx)
library(ROCit)
library(smotefamily)
library(ROSE)
library(ebmc)
```

4. Install the required packages for website development.

```
packages <- c("shiny", "shinythemes", "shinydashboard",
"readxl", "ggpubr",
+                "ggpmisc", "openxlsx", "shinyBS", "shinyjs")
install.packages(packages)
```

```
> packages <- c("shiny", "shinythemes", "shinydashboard", "readxl", "ggpubr",
+               "ggpmisc", "openxlsx", "shinyBS", "shinyjs")
> install.packages(packages)
```

Figure 26: Code to install required packages for website development

5. Try to run the libraries to verify the correct package installations.

```
library(shiny)
library(shinythemes)
library(shinydashboard)
library(readxl)
library(ggplot2)
library(ggpubr)
library(ggpmisc)
library(ebmc)
library(openxlsx)
library(shinyBS)
library(shinyjs)
```

2.4 Running R Script

Running the scripts for model training and website are the same for both RStudio and Visual Studio Code. All the files including the R scripts("final_model.R" and "app.R") are located in the "web_prototype" folder. Hence, run setwd() to this folder before running the scripts.

2.4.1 Model Training

1. Open the R terminal and navigate to the folder directory.

   ```
   setwd("[YOUR_PATH]/FIT3164_software/web_prototype")
   ```

2. Run the R script file called "final_model.R" (Ctrl + A on the script file > Ctrl + Enter / Press "Run" button at the top right).

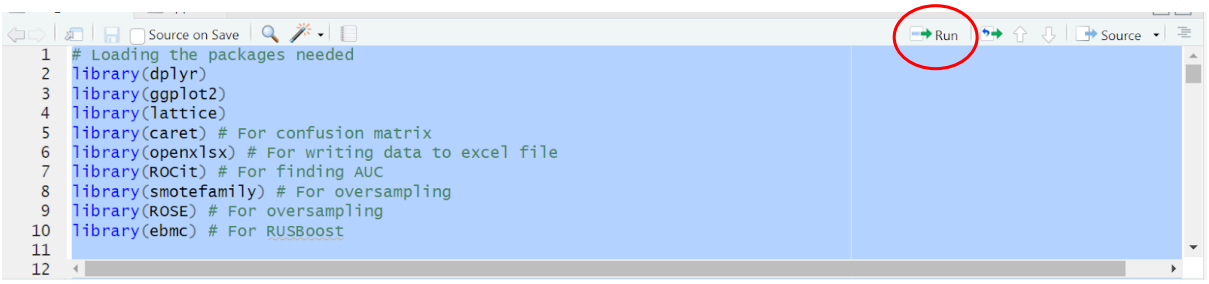


Figure 27: Running the "final_model.R" script file

2.4.2 Website

1. Open the R terminal and navigate to the folder directory.

   ```
   setwd("[PATH_TO_FOLDER]/FIT3164_software/web_prototype")
   ```

2. Run the R script file called "app.R" (Ctrl + A on the script file > Ctrl + Enter / "Press Run App" button at the top right). After execution, the localhost of the website will open in a new browser window.

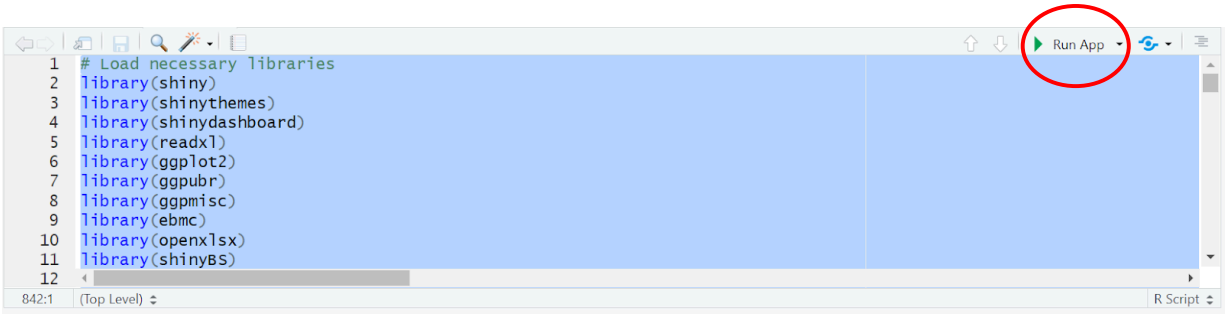


Figure 28: Running the "app.R" script file

3. Deploy new updates online.

   ```
   rsconnect::deployApp(appDir =
   "[YOUR_PATH]/FIT3164_software/web_prototype")
   ```

4. Open the website through the URL (https://mds02.shinyapps.io/web_prototype/) to see the updates.

FIT3164 Semester 1, 2024:
Test Report

Team: MDS02

Liaw Yi Hui (32023707)
Chan Jia Xin (31859089)
Pang Eason (32024584)

Table of Contents

1 Introduction

This project focuses on developing an improved machine learning model that is used for lymphedema prediction, specifically in breast cancer survivors. As outlined by Fu et al. (2018), lymphedema occurs due to the accumulation of lymph fluid in the body when lymph nodes are surgically removed during breast cancer treatment which leads to swelling. The lymph nodes act as drains in a sink which provide fluid clearance. Nevertheless, blockages hinder this process that lead to fluid retention. Existing methods for predicting lymphedema often depend on subjective manual assessments like limb measurements and medical imaging. Therefore, our project aims to develop an improved machine learning model that allows lymphedema predictions for early detection and intervention. Following this, the developed model is integrated into our website for predicting lymphedema, thereby advancing cancer care. Moreover, our website offers comprehensive information about lymphedema, alongside features for visualizing both data and prediction results.

During the model and website development phases, comprehensive testing procedures are implemented to ensure the efficacy and reliability of the lymphedema prediction platform. The testing phase is conducted iteratively throughout the entire development process which aligns with the Agile methodology applied in our project. This iterative testing approach allows us to continuously assess the performance of the model and website functionalities to ensure the final outcome aligns closely with the project goals. Moreover, stakeholders' testing and feedback play a crucial role in refining the user interface and functionality of the website. By continuously testing and refining the model and website functionalities, we can promptly identify and address any issues or deficiencies, ultimately enhancing the overall performance and user experience of the lymphedema prediction platform.

This report covers several key topics related to testing of our software such as test plan, test approach, black box testing, integration testing, usability testing, recommendation for improvements and test limitations.

2 Test Plan

2.1 Test Objectives

Test objectives serve as the guiding principles for the testing activities which outline specific goals and targets that the testing process aims to achieve. They ensure that the developed system or software meets the specified requirements and functions effectively. The summary of the test objectives are listed in Table 1.

Test Objective	Test Phases
Machine learning model is integrated correctly into the frontend.	Integration Testing
Input data files can be uploaded and correctly passed to the server.	Integration Testing
The frontend allows successful dataset uploads.	Blackbox Testing, Usability Testing
The frontend displays error messages when an invalid input data file format is detected.	Blackbox Testing, Integration Testing,
Frontend can predict and display results correctly.	Blackbox Testing, Integration Testing, Usability Testing
The frontend can display visualizations of both the uploaded dataset and prediction results correctly.	Blackbox Testing, Usability Testing
The website is user-friendly and meets stakeholder's expectations.	Usability Testing

Table 1: Summary of test objectives with their respective test phases

2.2 Test Approach

Our project focuses on black box, integration, and usability testing of the developed software. Black box testing examines the software's external behavior and functionality without knowledge of its internal implementation. Furthermore, integration testing ensures that the frontend, backend, and machine learning model collaborate correctly by verifying smooth and accurate data flow to ensure precision and integrity of the results. Usability testing involves end users to confirm that the software meets their requirements and expectations.

For the test approach, we employed manual testing to assess the functionality and usability of our web page prototype. This involved systematically navigating through the various features and functionalities of the web page to identify any bugs, errors, or inconsistencies. Through manual testing, we were able to evaluate the user interface, data input/output processes, and overall user experience. While automated testing frameworks such as PyTest were not utilized in this instance, manual testing provided valuable insights into the performance and reliability of our web page, allowing us to iteratively improve and refine its functionality.

3 Blackbox Testing

Black box testing evaluates the functionality of an application without having a detailed understanding of its internal structure and implementation. During this process, input values are compared directly with output values. Furthermore, black box testing prioritizes understanding the user experience. Thus, testers do not need extensive technical knowledge to perform it. It provides broad testing coverage, which is especially valuable compared to white box testing because white box testing can be so detailed that testers might overlook the overall functionality (Ashtari, 2022).

According to Imperva (n.d.), black box testing has several testing techniques which include Equivalence Class Partitioning, Boundary Value Analysis, Decision Table Testing, State Transition Testing, and Error Guessing. For our software testing, it primarily focuses on Equivalence Class Partitioning. Equivalence Class Partitioning operates under the assumption that both the input and output domains of a program can be divided into finite classes, which include valid and invalid cases. Within each partition, all scenarios share the same functionality or behavior (Nidhra and Dondeti, 2012). This focus aligns with our specific requirement of processing data files containing patient information to generate visualizations and prediction results for each row in the file.

3.1 Prediction Component

For testing the prediction component of our software, it primarily focuses on validating the correctness of the input file. The equivalence class partitions and corresponding test cases are listed in Table 2 and Table 3 respectively.

Valid Equivalence Classes	Invalid Equivalence Classes
Valid file type (.xls, .xlsx, .csv)	Invalid file type
Valid sheet name for “.xls” and “.xlsx” file types	Invalid sheet name for “.xls” and “.xlsx” file types
No missing columns	Missing columns
Valid column names	Invalid column names
Valid data format	Invalid data format

Table 2: Valid and invalid equivalence class partitions for prediction component

Test ID	Test Description	Test Method	Expected Output	Actual Output
T1	Test valid file type, sheet name, data format, column names and no missing columns	Upload a correct input data file with valid file type, data format and column names without any missing columns	Generate and display the prediction results	Generate and display the prediction results as shown in Figure 1
T2	Test invalid file type	Upload a pdf file	Show error message of “Error: Unsupported file format.”	Show error message of “Error: Unsupported file format.” as shown in Figure 2
T3	Test invalid sheet name for “.xls” and “.xlsx” file types	Upload a dataset of “.xlsx” file with any sheet name other than “DataTemplate” (correct sheet name)	Show error message of "Error: Sheet 'DataTemplate' not found in the Excel file. Please rename the sheet."	Show error message of "Error: Sheet 'DataTemplate' not found in the Excel file. Please rename the sheet." as shown in Figure 3
T4	Test file with missing columns	Upload a file with a few missing columns	Show error message of "Error: Dataset is missing required column(s) or wrong column name(s)."	Show error message of "Error: Dataset is missing required column(s) or wrong column name(s)." as shown in Figure 4
T5	Test invalid column names	Upload a file with typos for some column names	Show error message of "Error: Dataset is missing required column(s) or wrong column name(s)."	Show error message of "Error: Dataset is missing required column(s) or wrong column name(s)." as shown in Figure 5
T6	Test invalid data format (missing values or “NA”s)	Upload a file that contains missing values or “NA”s	Show error message of “Error: Dataset contains missing values."	Show error message of “Error: Dataset contains missing values." as shown in Figure 6
T7	Test invalid data format (non-numeric for the columns except “ID”)	Upload a file that contains non-numeric in some columns	Show error message of "Error: Incorrect data format. Data must be numeric."	Show error message of "Error: Incorrect data format. Data must be numeric." as shown in Figure 7
T8	Test the “axi” column for invalid data formats by checking for any integer other than 0 or 1.	Upload a file where the “axi” column contains any integer other than 0 or 1	Show error message off "Error: Data for ‘axi’ column must contain 0(No) or 1(Yes) only."	Show error message of "Error: Data for ‘axi’ column must contain 0(No) or 1(Yes) only." as shown in Figure 8

T9	Test the “che” column for invalid data formats by checking for any integer other than 0 or 1.	Upload a file where the “che” column contains any integer other than 0 or 1	Show error message of "Error: Data for ‘che’ column must contain 0(No) or 1(Yes) only."	Show error message of"Error: Data for ‘che’ column must contain 0(No) or 1(Yes) only." as shown in Figure 9
T10	Test the “recon” column for invalid data formats by checking for any integer other than 0, 1 or 2.	Upload a file where the “recon” column contains any integer other than 0, 1 or 2	Show error message of "Error: Data for ‘recon’ column must contain 0(No reconstruction), 1(TRAM flap) or 2(Implant) only."	Show error message of "Error: Data for ‘recon’ column must contain 0(No reconstruction), 1(TRAM flap) or 2(Implant) only." as shown in Figure 10
T11	Test the “tax” column for invalid data formats by checking for any integer other than 0, 1 or 2.	Upload a file where the “tax” column contains any integer other than 0, 1 or 2	Show error message of "Error: Data for ‘tax’ column must contain 0(No taxane), 1(Type 1) or 2(Type 2) only."	Show error message of"Error: Data for ‘tax’ column must contain 0(No taxane), 1(Type 1) or 2(Type 2) only." as shown in Figure 11
T12	Test the “sex” column for invalid data formats by checking for any integer other than 1 or 2.	Upload a file where the “sex” column contains any integer other than 1 or 2	Show error message of "Error: Data for ‘sex’ column must contain 1(Male) or 2(Female) only."	Show error message of "Error: Data for ‘sex’ column must contain 1(Male) or 2(Female) only." as shown in Figure 12

Table 3: Test cases for prediction component

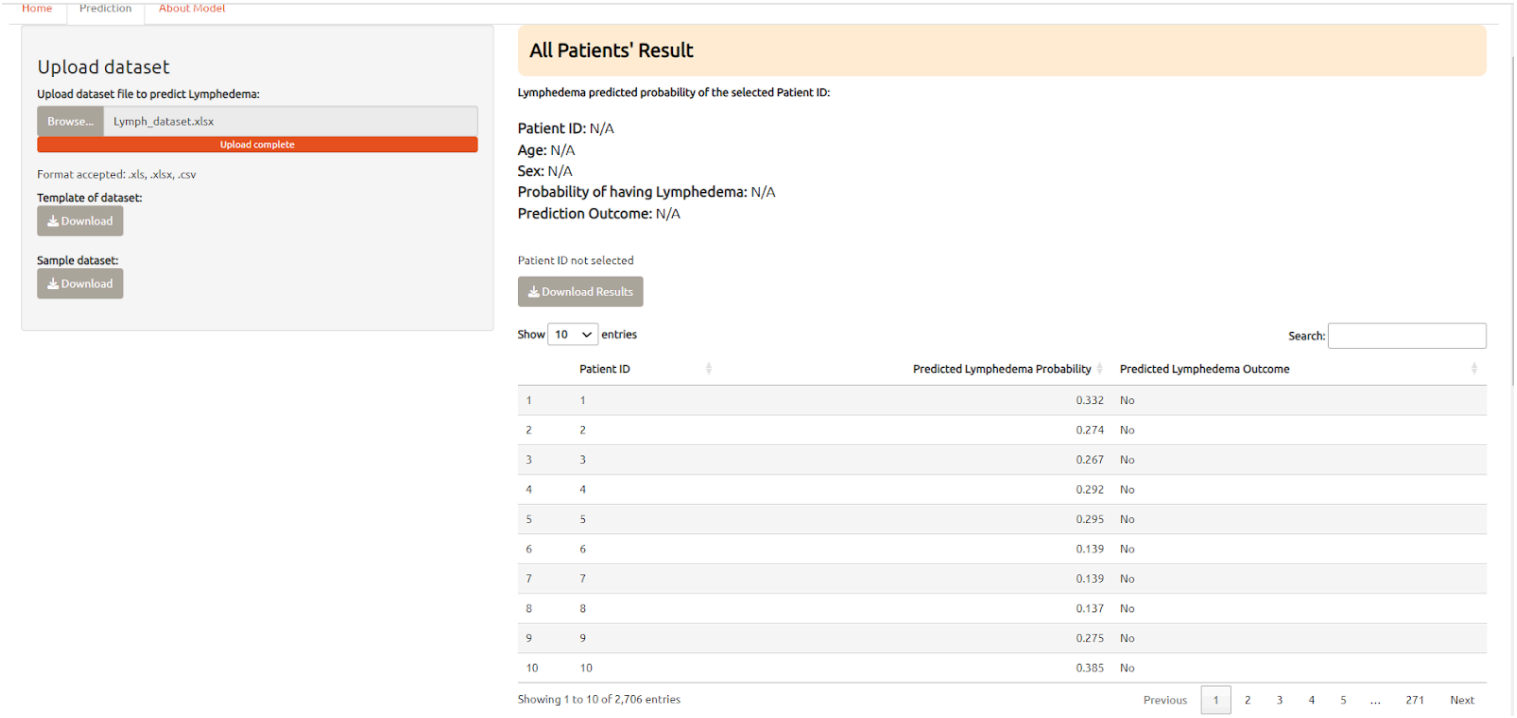


Figure 1: Testing result for valid input data file

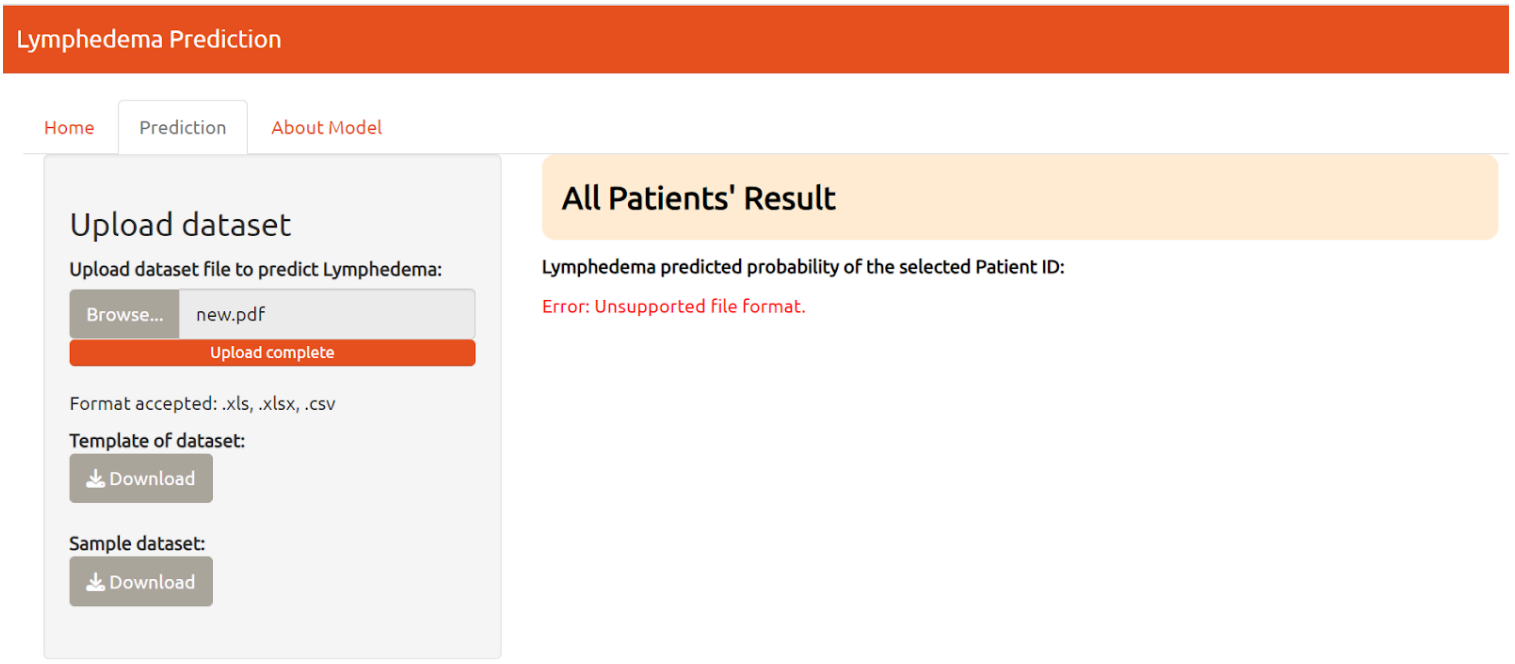


Figure 2: Testing result for invalid file type (pdf file)

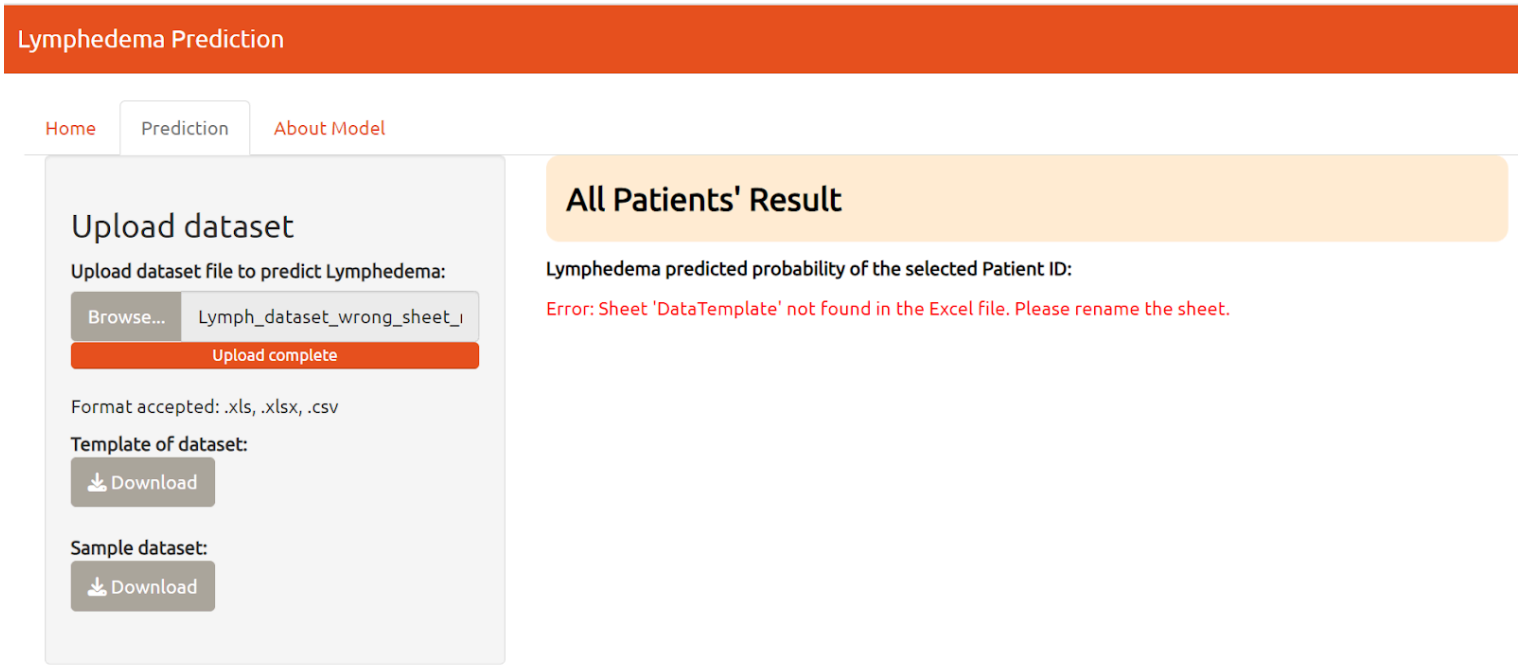


Figure 3: Testing result for invalid sheet name ("Template1" instead of "DataTemplate")

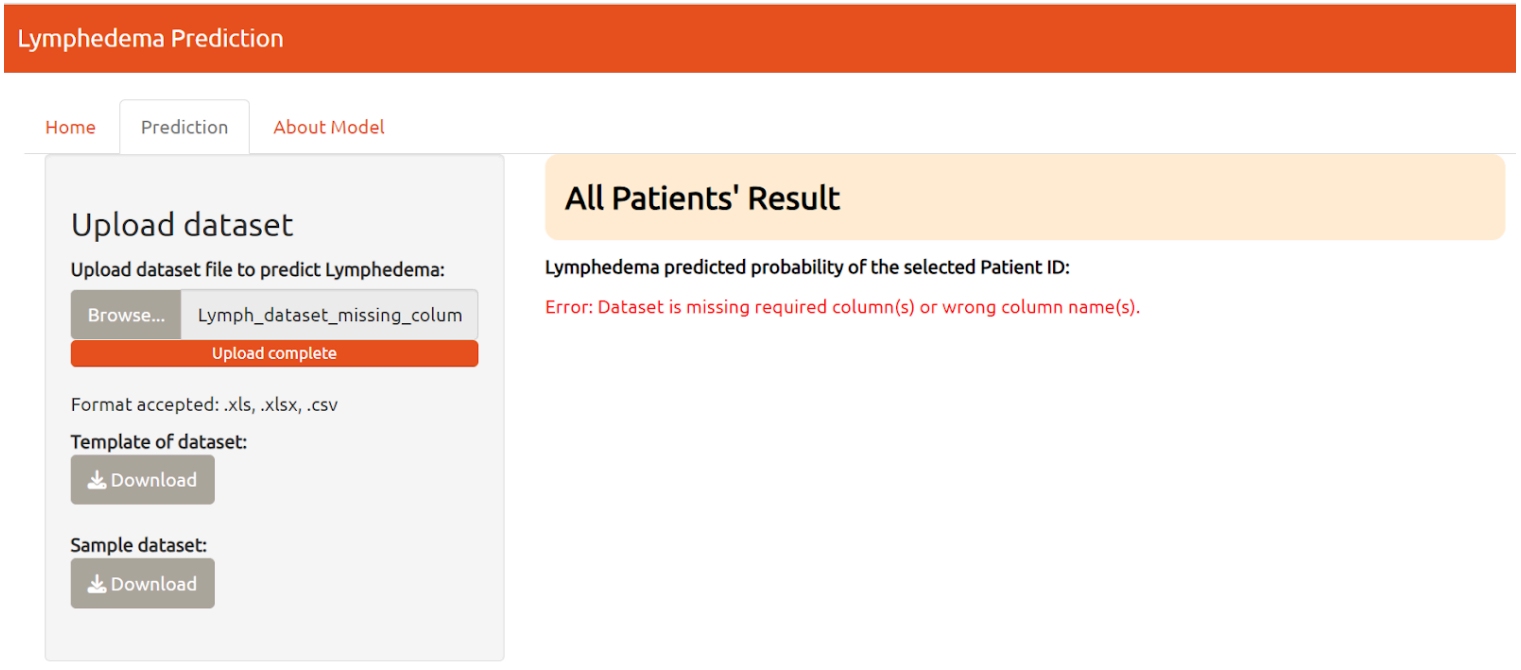


Figure 4: Testing result for missing columns of "PLT"

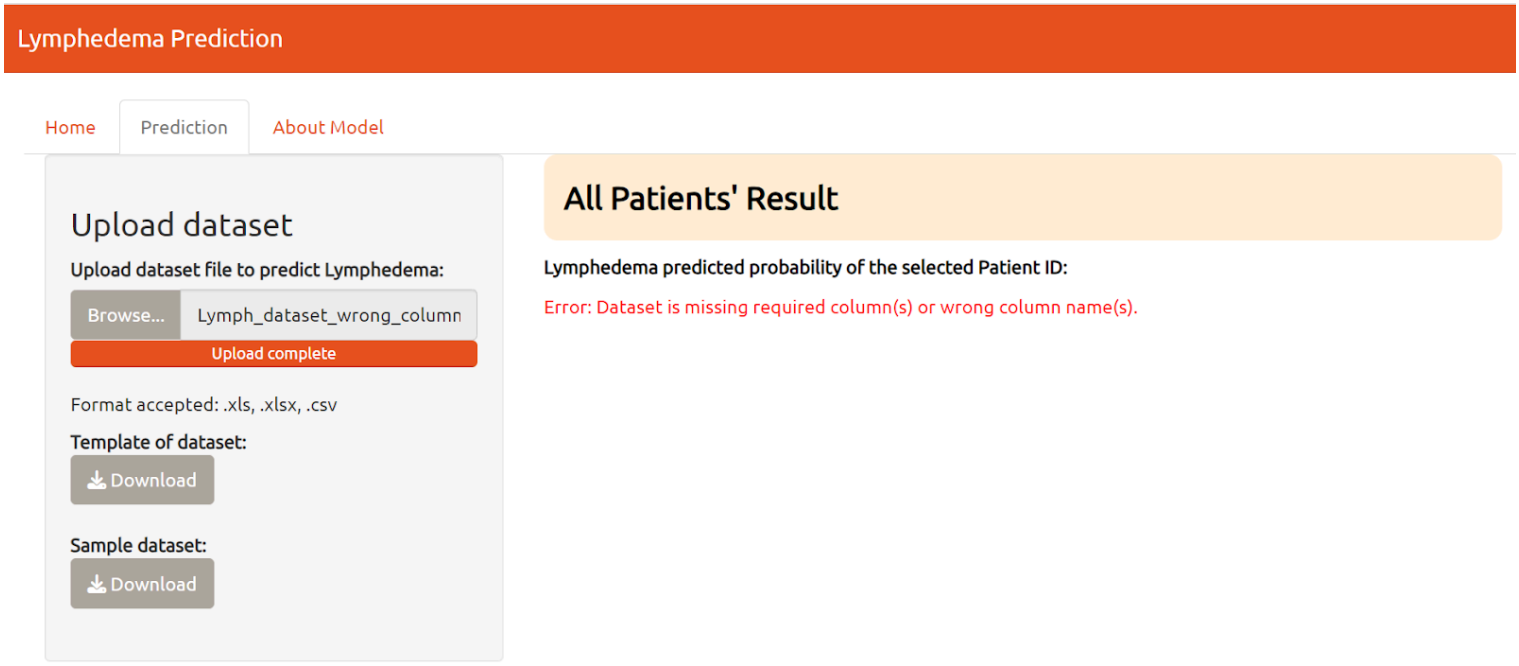


Figure 5: Testing result for invalid column names

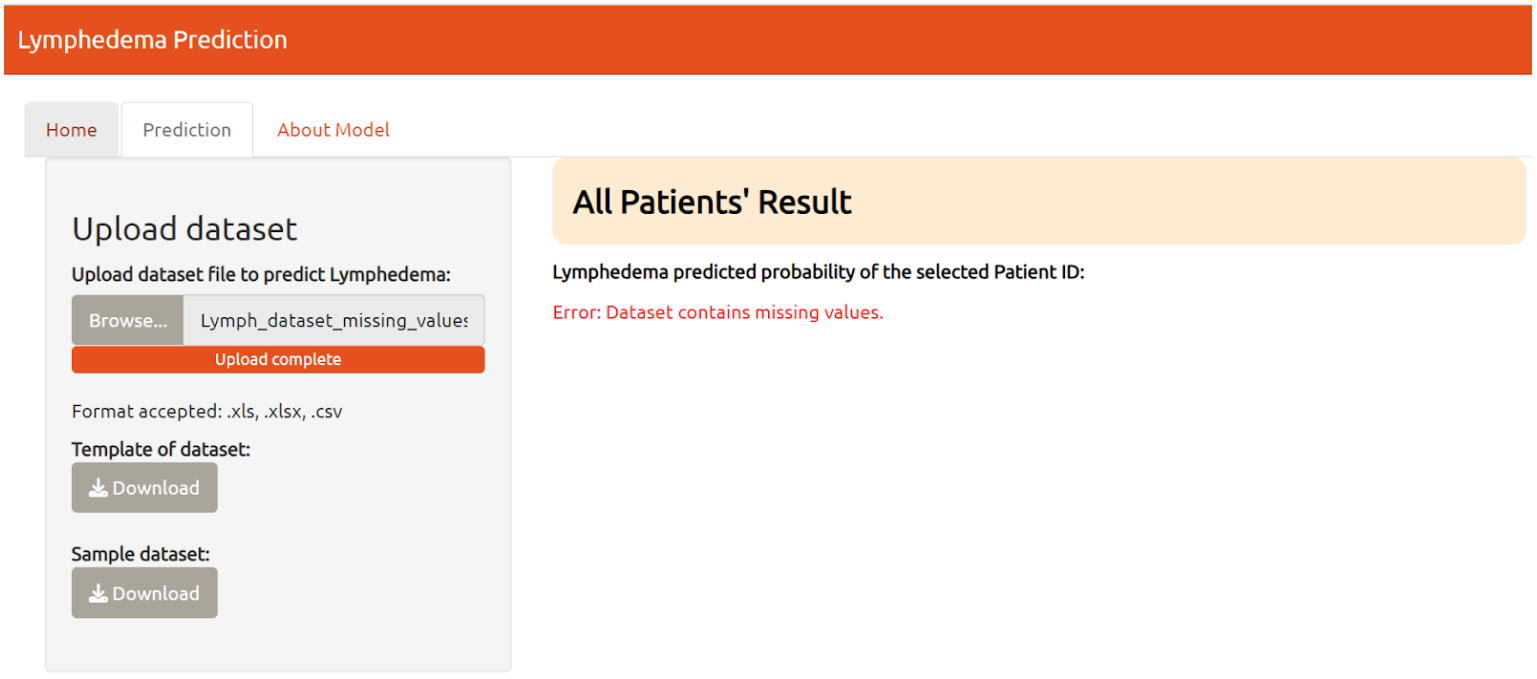


Figure 6: Testing result for invalid data format (missing values)

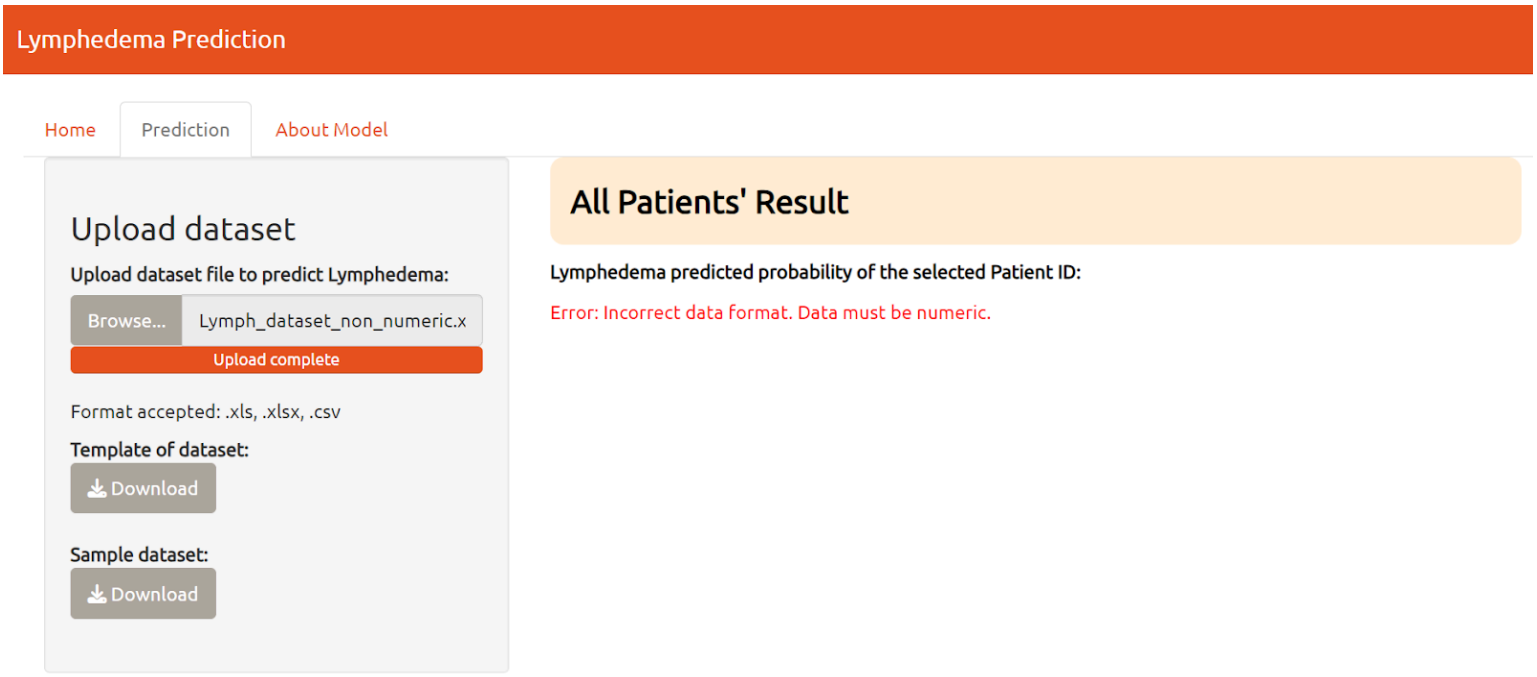


Figure 7: Testing result for invalid data format (contains non-numeric)

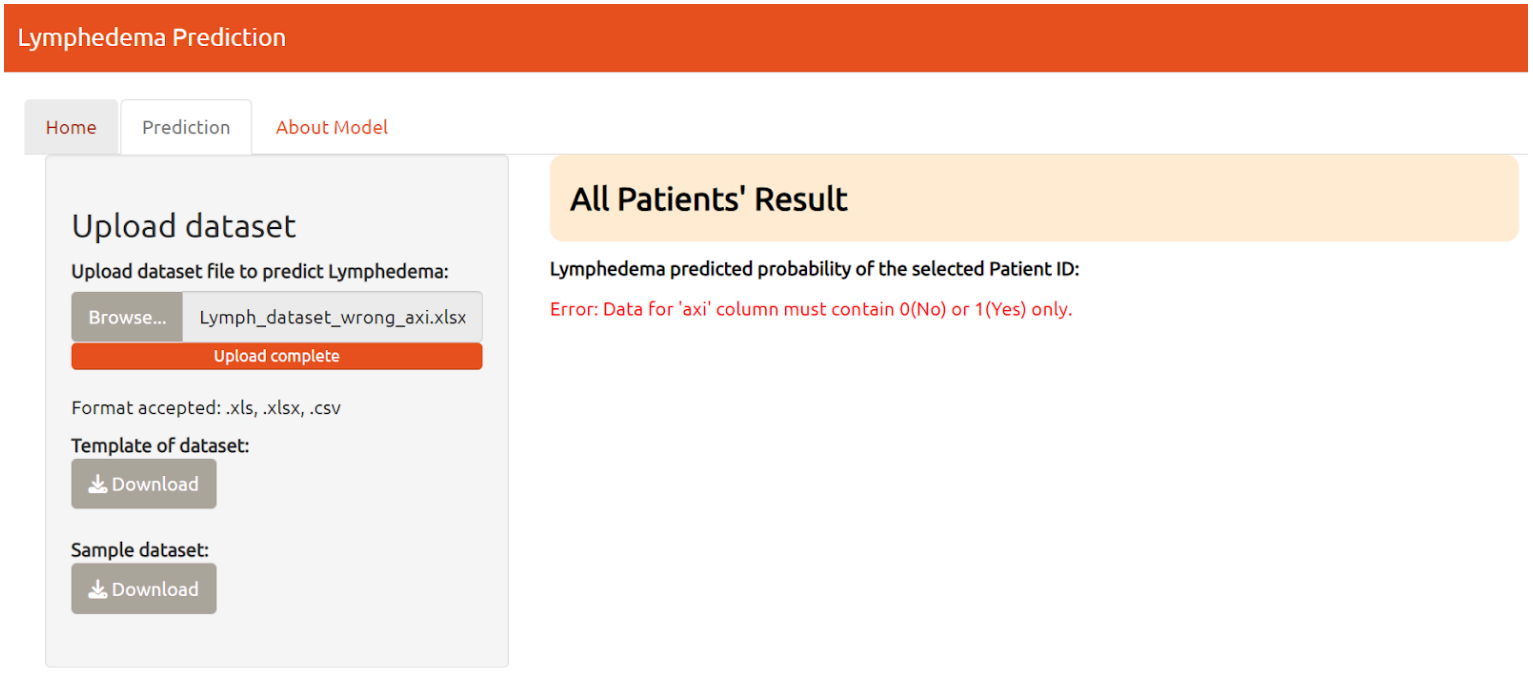


Figure 8: Testing result for invalid data format (invalid values in "axi" column)

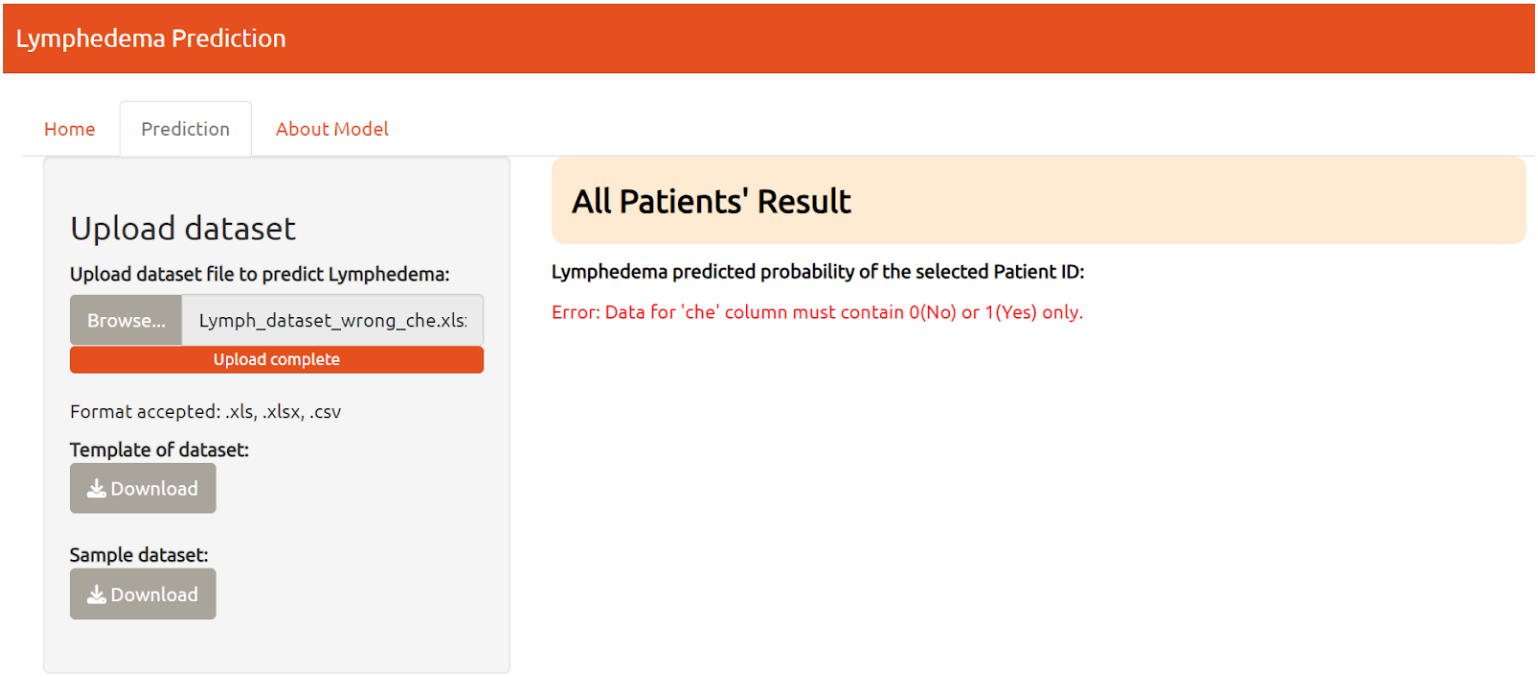


Figure 9: Testing result for invalid data format (invalid values in "che" column)

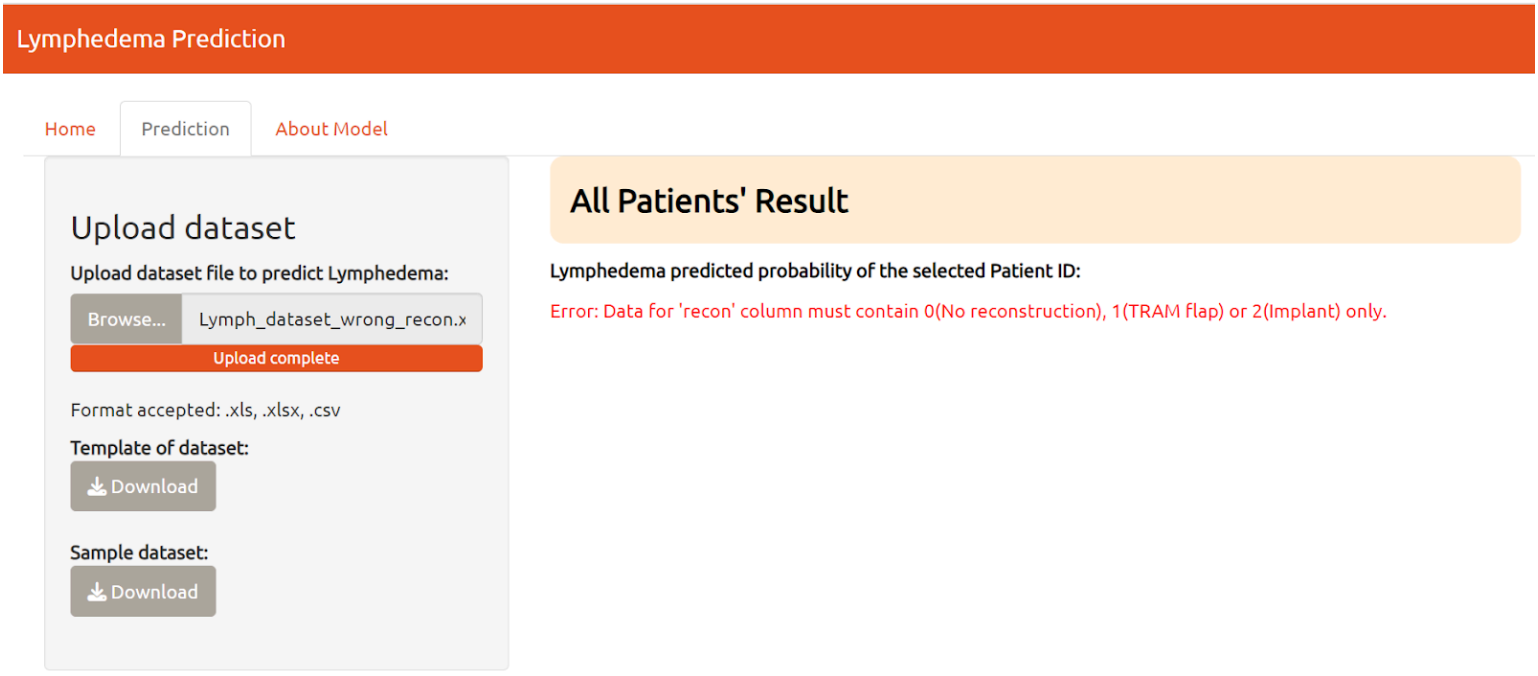


Figure 10: Testing result for invalid data format (invalid values in "recon" column)

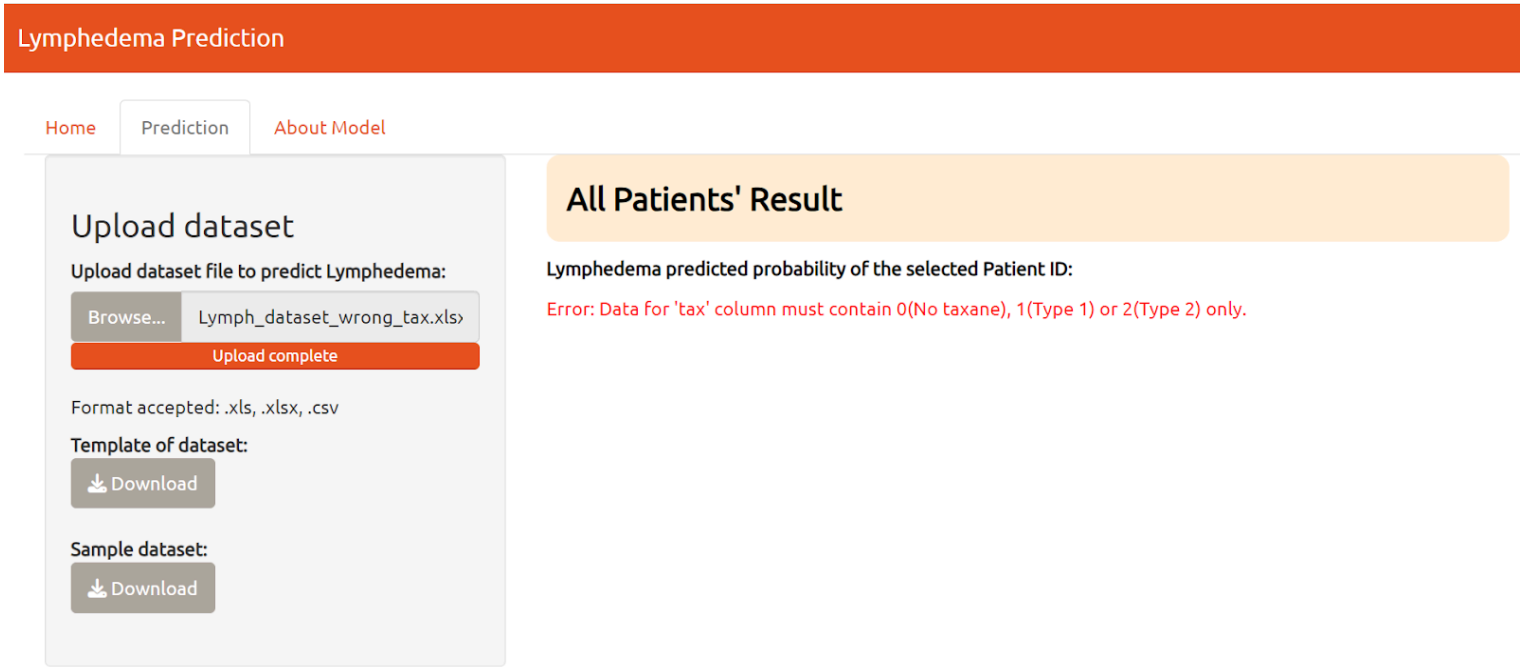


Figure 11: Testing result for invalid data format (invalid values in "tax" column)

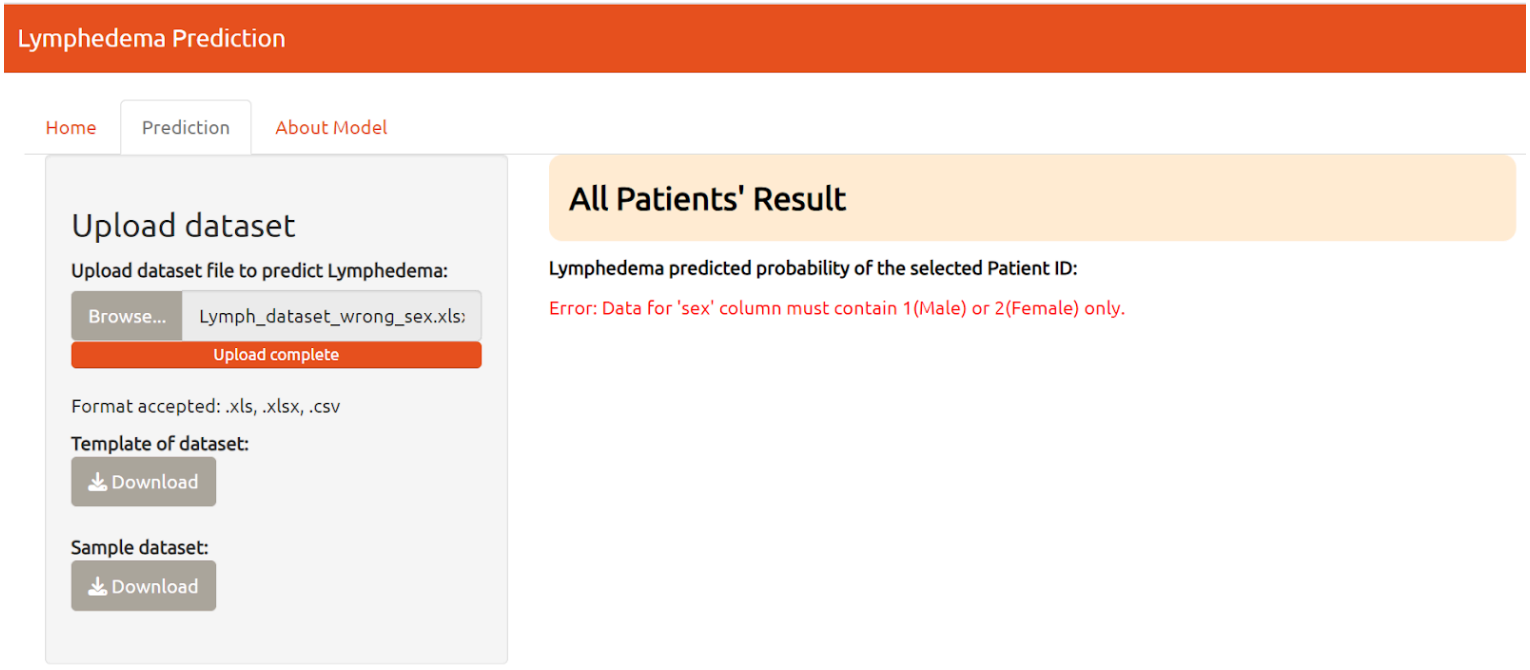


Figure 12: Testing result for invalid data format (invalid values in "sex" column)

3.2 Visualization Component

For testing the data and prediction results visualizations of our software, it focuses on validating the correctness of the input file and the selected feature. The equivalence class partitions and corresponding test cases are listed in Table 4 and Table 5 respectively.

Valid Equivalence Classes	Invalid Equivalence Classes
Valid file type (.xls, .xlsx, .csv)	Invalid file type
Valid sheet name for “.xls” and “.xlsx” file types	Invalid sheet name for “.xls” and “.xlsx” file types
No missing columns	Missing columns
Valid column names	Invalid column names
Valid data format	Invalid data format

Table 4: Valid and invalid equivalence class partitions for visualization components

Test ID	Test Description	Test Method	Expected Output	Actual Output
T13	Test valid file type, sheet name, data format, column names and no missing columns	Upload a correct input data file with valid file type, data format and column names without any missing columns and select a feature for visualization	Generate and display the prediction results	Generate and display the prediction results as shown in Figures 13 and 14
T14	Test invalid file type	Upload a pdf file	Show error message of “Error: Unsupported file format.”	Show error message of “Error: Unsupported file format.” as shown in Figures 15 and 16
T15	Test invalid sheet name for “.xls” and “.xlsx” file types	Upload a dataset of “.xlsx” file with any sheet name other than “DataTemplate” (correct sheet name)	Show error message of "Error: Sheet 'DataTemplate' not found in the Excel file. Please rename the sheet."	Show error message of "Error: Sheet 'DataTemplate' not found in the Excel file. Please rename the sheet." as shown in Figures 17 and 18
T16	Test file with missing columns	Upload a file with a few missing columns and for data visualization, select the feature with missing column	Show error message of "Error: Dataset is missing required column(s) or wrong column name(s)." for prediction results visualization and "Error: Please ensure the selected feature is in the uploaded file." for data visualization	Show error message of "Error: Dataset is missing required column(s) or wrong column name(s)." for prediction results visualization and "Error: Please ensure the selected feature is in the uploaded file." for data visualization as shown in Figures 19 and 20 respectively
T17	Test invalid column names	Upload a file with typos for some column names and for data visualization, select the feature with wrong column name	Show error message of "Error: Dataset is missing required column(s) or wrong column name(s)." for prediction results visualization and "Error: Please ensure the selected feature is in the uploaded file." for data visualization	Show error message of "Error: Dataset is missing required column(s) or wrong column name(s)." for prediction results visualization and "Error: Please ensure the selected feature is in the uploaded file." for data visualization as shown in Figures 21 and 22 respectively

T18	Test invalid data format (missing values or “NA”s)	Upload a file that contains missing values or “NA”s	Show error message of “Error: Dataset contains missing values."	Show error message of “Error: Dataset contains missing values." as shown in Figures 23 and 24
T19	Test invalid data format (non-numeric for the columns except “ID”)	Upload a file that contains non-numeric in some columns	Show error message of "Error: Incorrect data format. Data must be numeric."	Show error message of "Error: Incorrect data format. Data must be numeric." as shown in Figures 25 and 26
T20	Test the “axi” column for invalid data formats by checking for any integer other than 0 or 1.	Upload a file where the “axi” column contains any integer other than 0 or 1	Show error message off "Error: Data for ‘axi’ column must contain 0(No) or 1(Yes) only."	Show error message of "Error: Data for ‘axi’ column must contain 0(No) or 1(Yes) only." as shown in Figures 27 and 28
T21	Test the “che” column for invalid data formats by checking for any integer other than 0 or 1.	Upload a file where the “che” column contains any integer other than 0 or 1	Show error message of "Error: Data for ‘che’ column must contain 0(No) or 1(Yes) only."	Show error message of"Error: Data for ‘che’ column must contain 0(No) or 1(Yes) only." as shown in Figures 29 and 30
T22	Test the “recon” column for invalid data formats by checking for any integer other than 0, 1 or 2.	Upload a file where the “recon” column contains any integer other than 0, 1 or 2	Show error message of "Error: Data for ‘recon’ column must contain 0(No reconstruction), 1(TRAM flap) or 2(Implant) only."	Show error message of "Error: Data for ‘recon’ column must contain 0(No reconstruction), 1(TRAM flap) or 2(Implant) only." as shown in Figures 31 and 32
T23	Test the “tax” column for invalid data formats by checking for any integer other than 0, 1 or 2.	Upload a file where the “tax” column contains any integer other than 0, 1 or 2	Show error message of "Error: Data for ‘tax’ column must contain 0(No taxane), 1(Type 1) or 2(Type 2) only."	Show error message of"Error: Data for ‘tax’ column must contain 0(No taxane), 1(Type 1) or 2(Type 2) only." as shown in Figures 33 and 34
T24	Test the “sex” column for invalid data formats by checking for any integer other than 1 or 2.	Upload a file where the “sex” column contains any integer other than 1 or 2	Show error message of "Error: Data for ‘sex’ column must contain 1(Male) or 2(Female) only."	Show error message of "Error: Data for ‘sex’ column must contain 1(Male) or 2(Female) only." as shown in Figures 35 and 36

Table 5: Test cases for visualization components

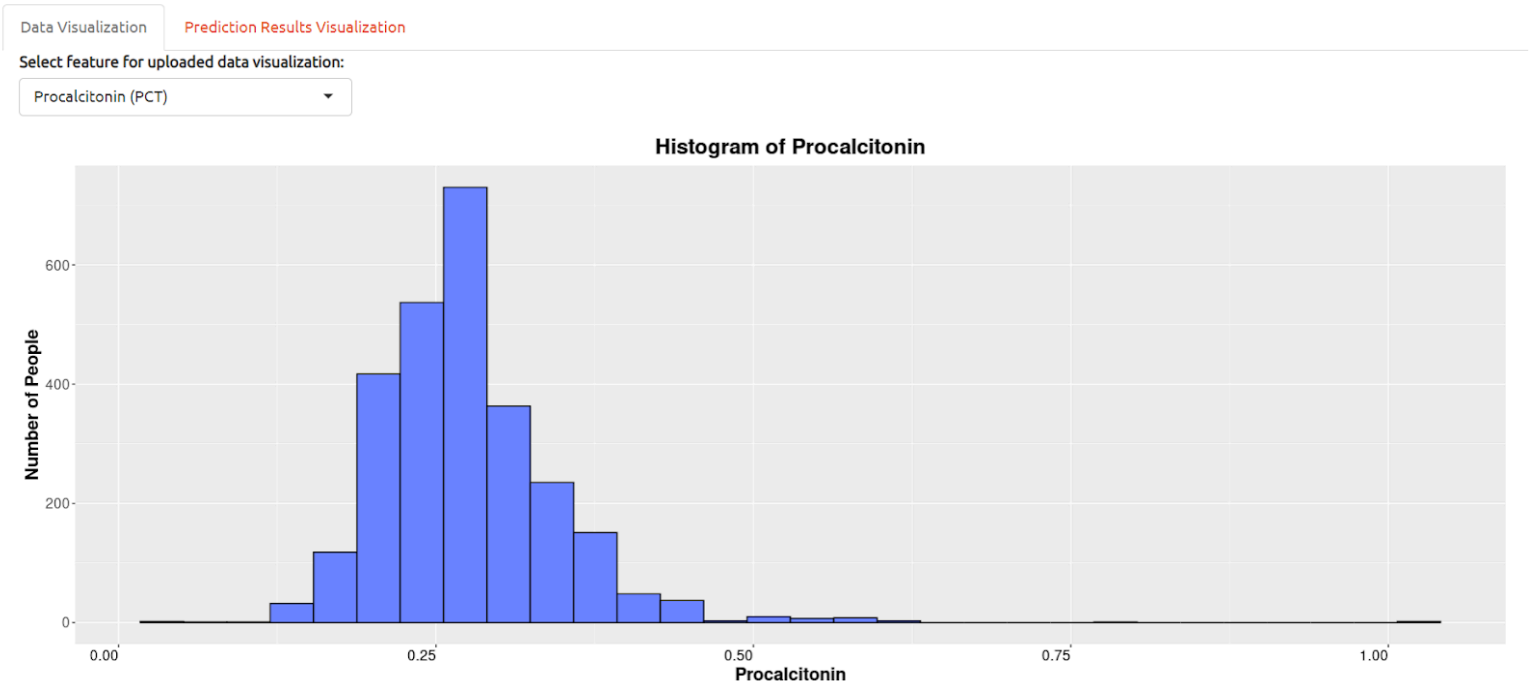


Figure 13: Testing result for valid input data file (for data visualization)

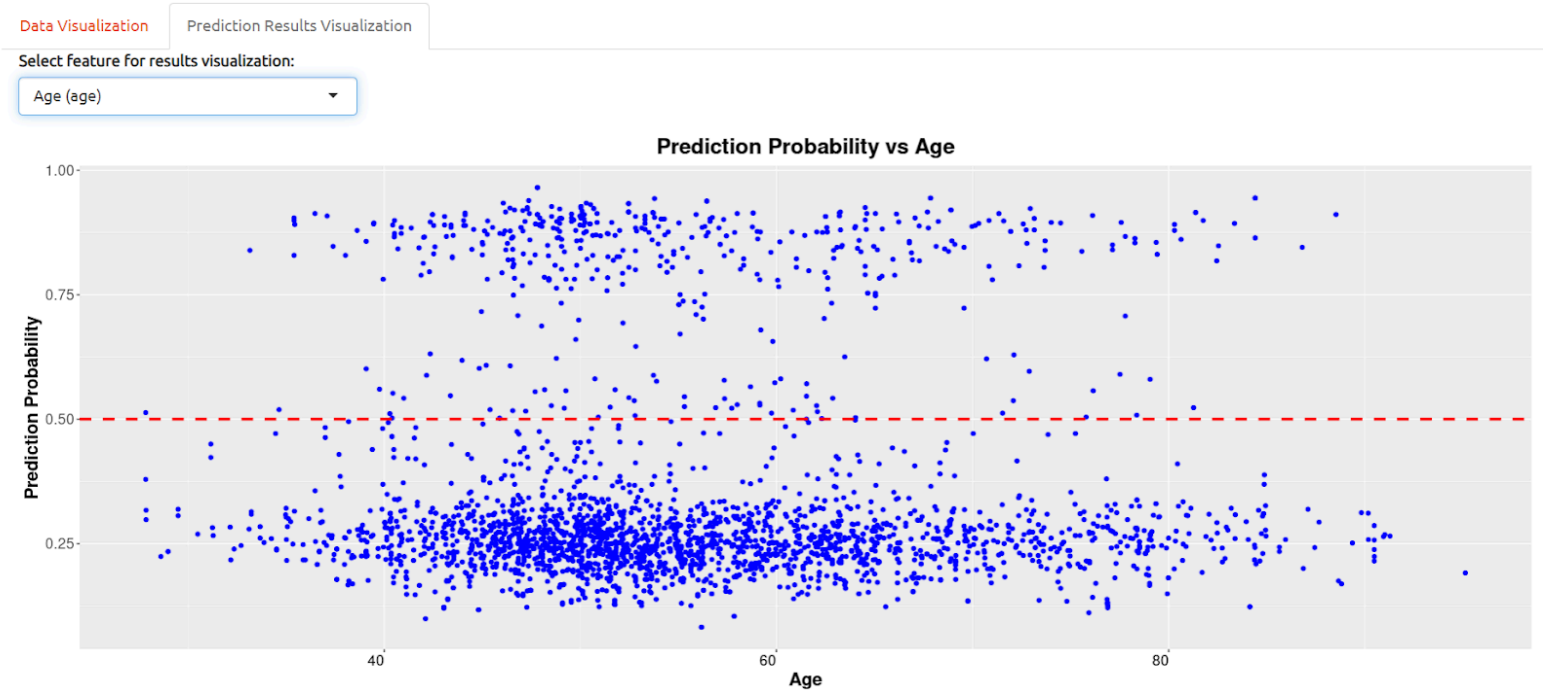


Figure 14: Testing result for valid input data file (for prediction results visualization)

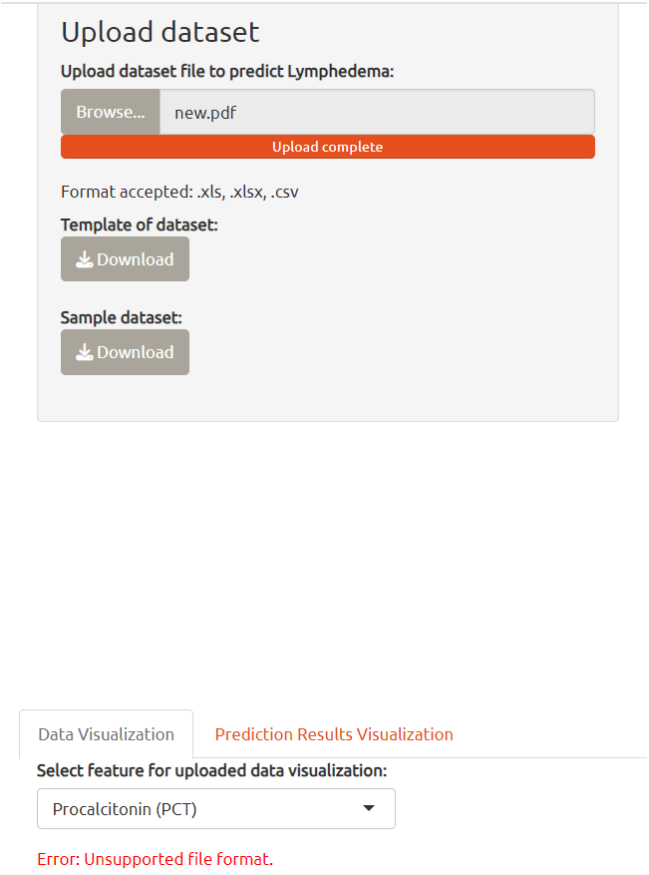


Figure 15: Testing result for invalid file type for prediction results visualization

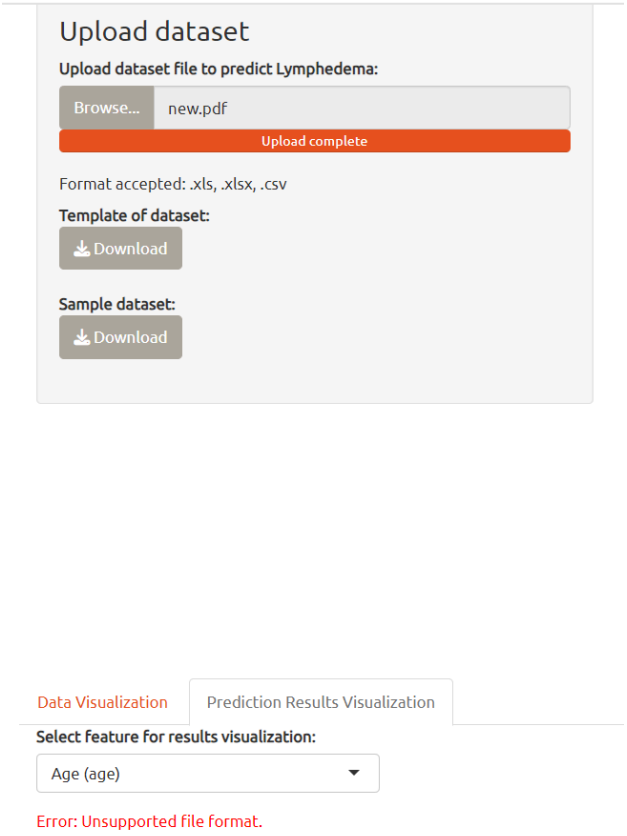
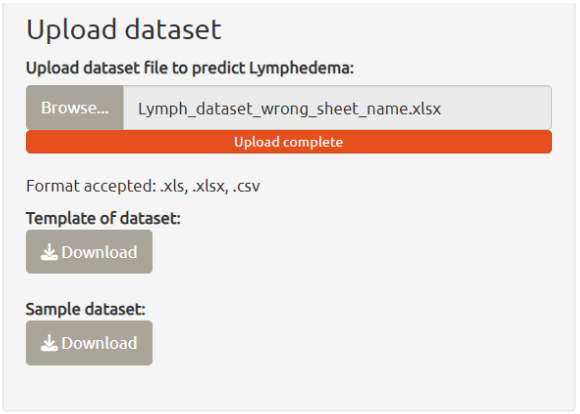


Figure 16: Testing result for invalid file type for prediction results visualization



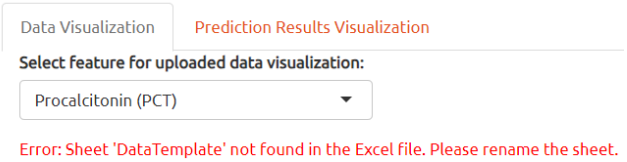
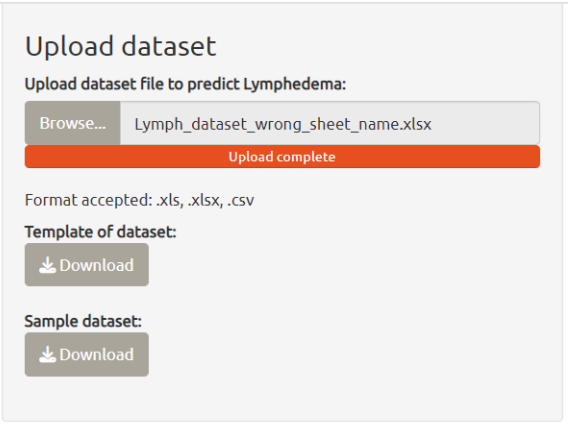


Figure 17: Testing result for invalid sheet name for data visualization



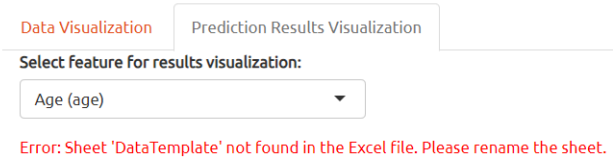


Figure 18: Testing result for invalid sheet name for prediction results visualization

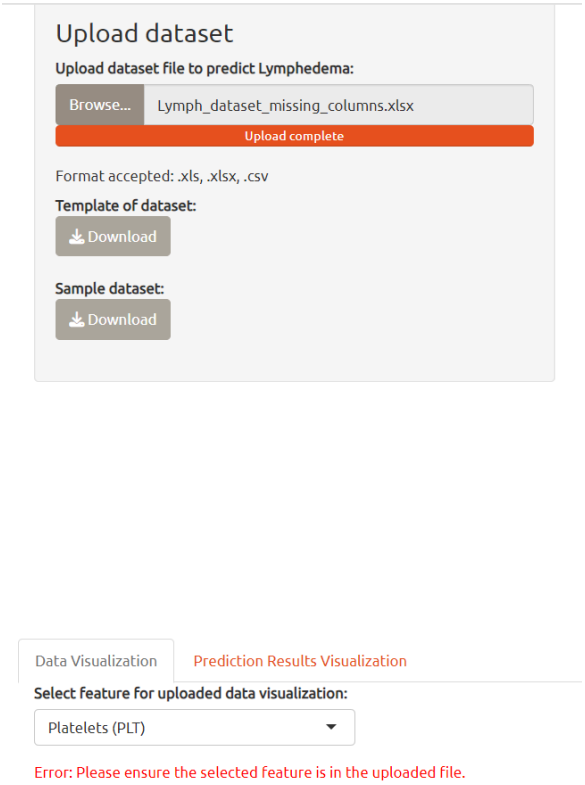


Figure 19: Testing result for missing columns of "PLT" for data visualization

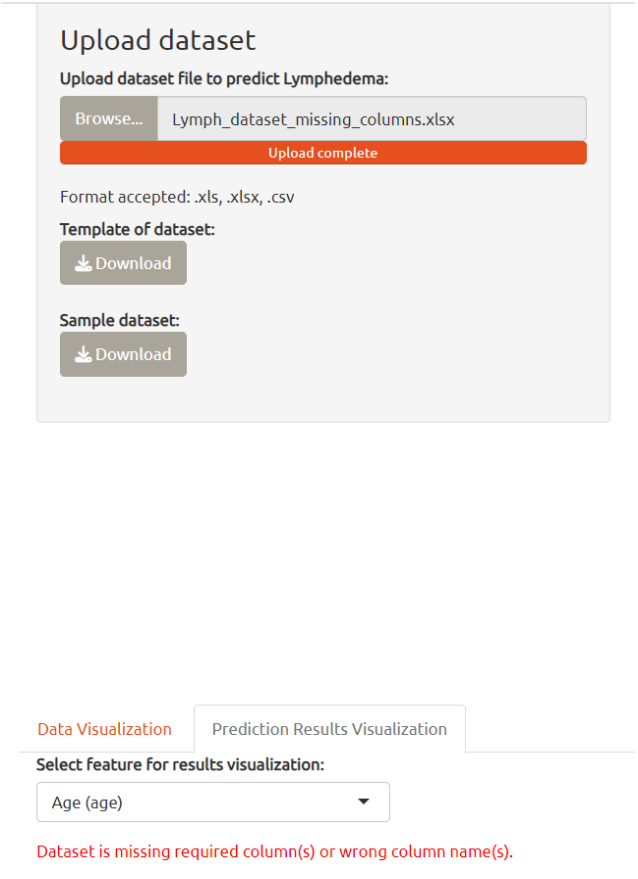


Figure 20: Testing result for missing columns of "PLT" for prediction results visualization

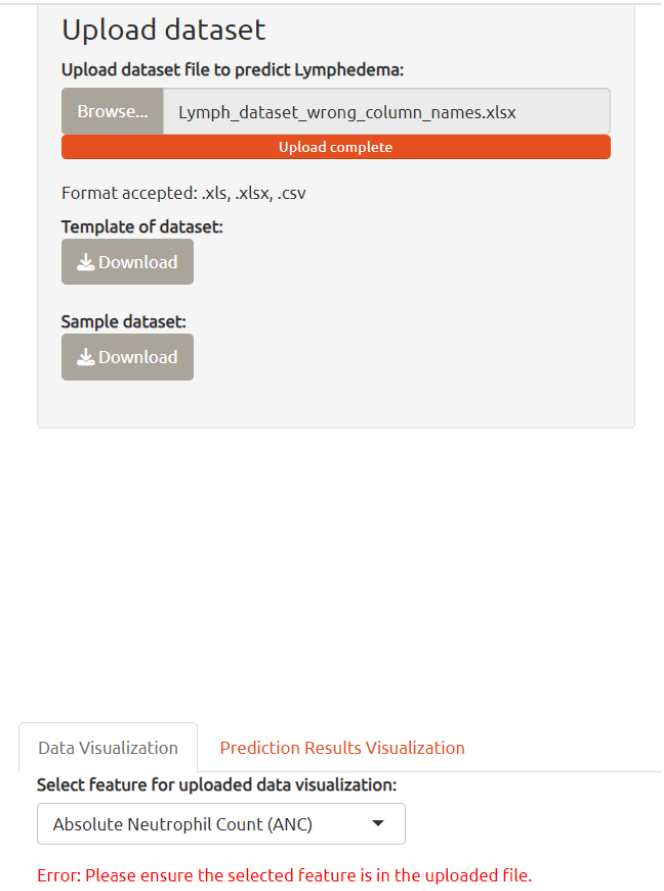


Figure 21: Testing result for invalid column names for data visualization

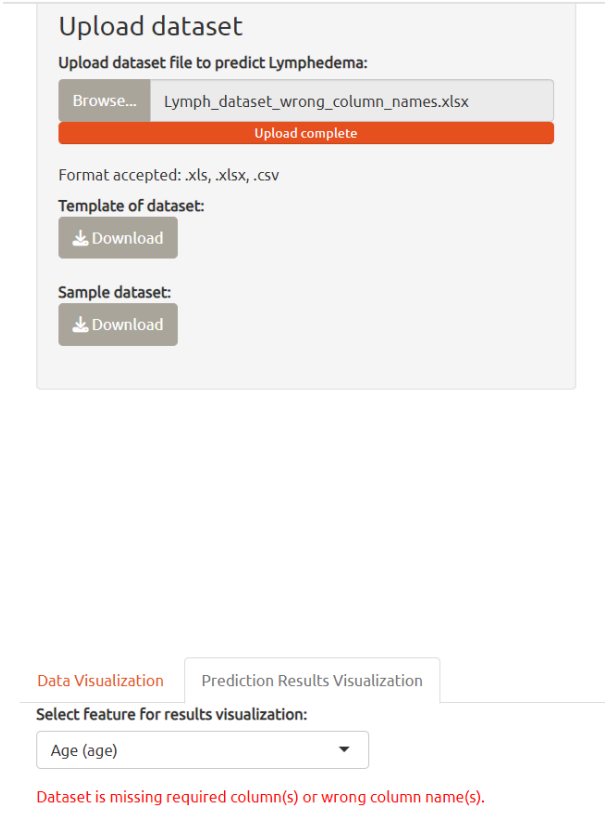


Figure 22: Testing result for invalid column names for prediction results visualization

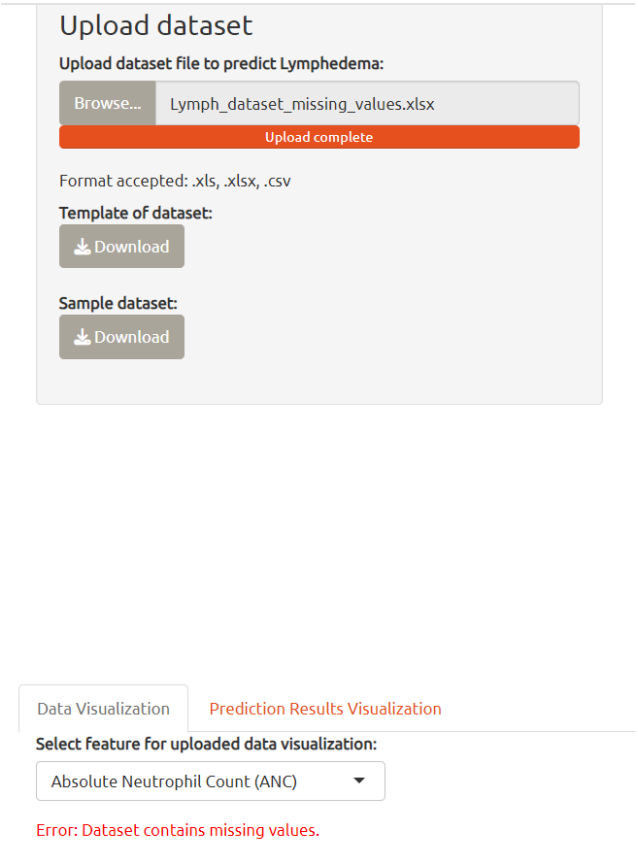


Figure 23: Testing result for invalid data format (missing values) for data visualization

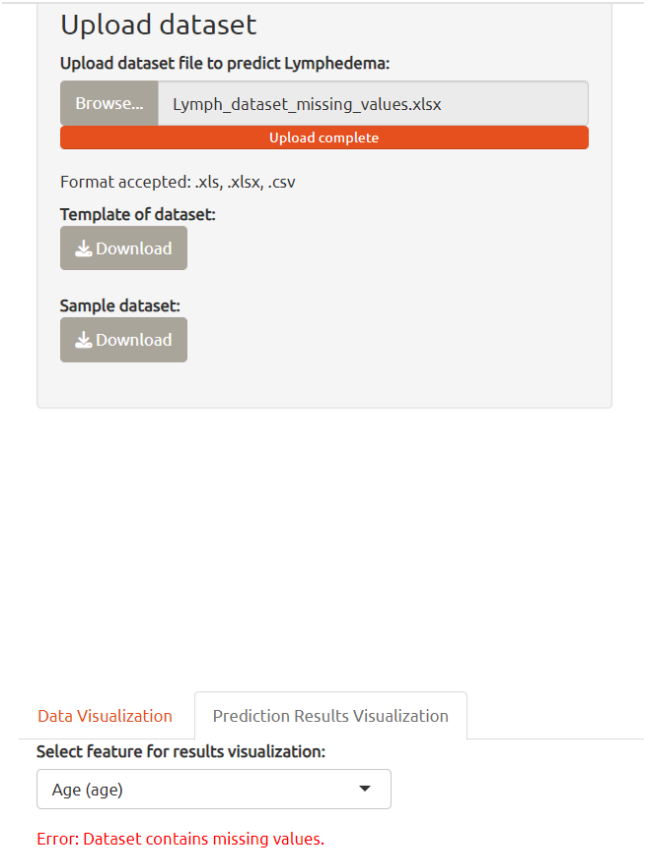


Figure 24: Testing result for invalid data format (missing values) for prediction results visualization

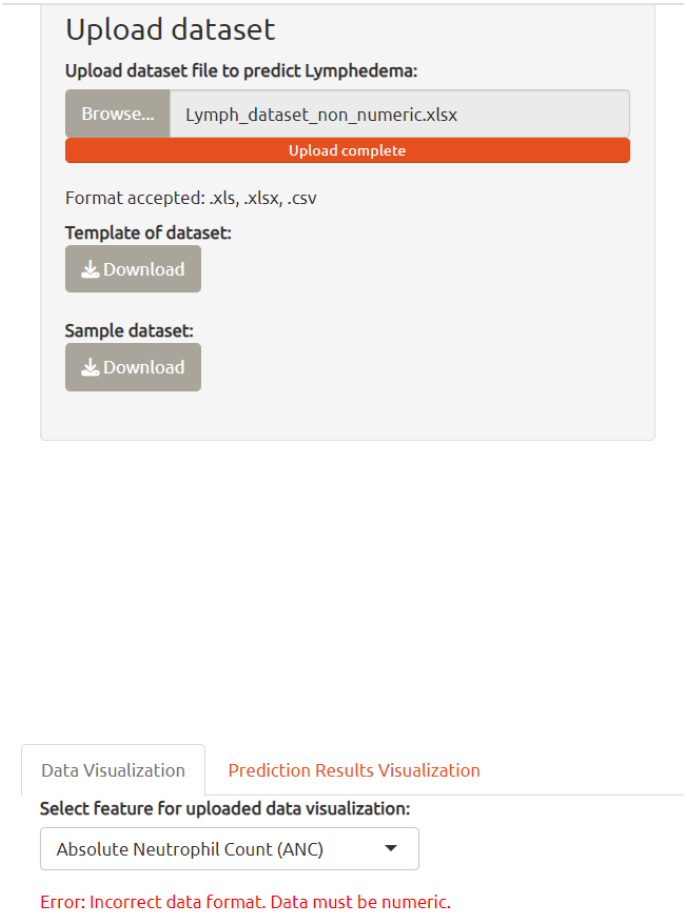


Figure 25: Testing result for invalid data format (contains non-numeric) for data visualization

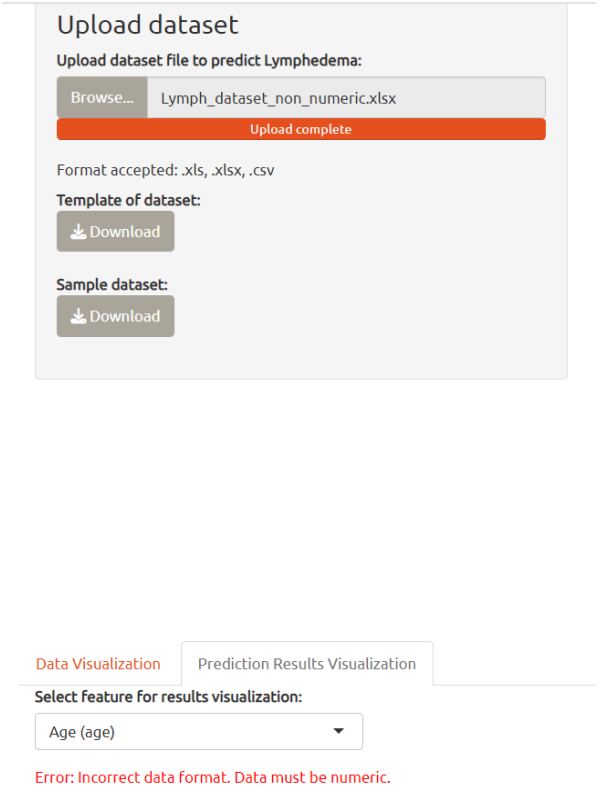


Figure 26: Testing result for invalid data format (contains non-numeric) for prediction results visualization

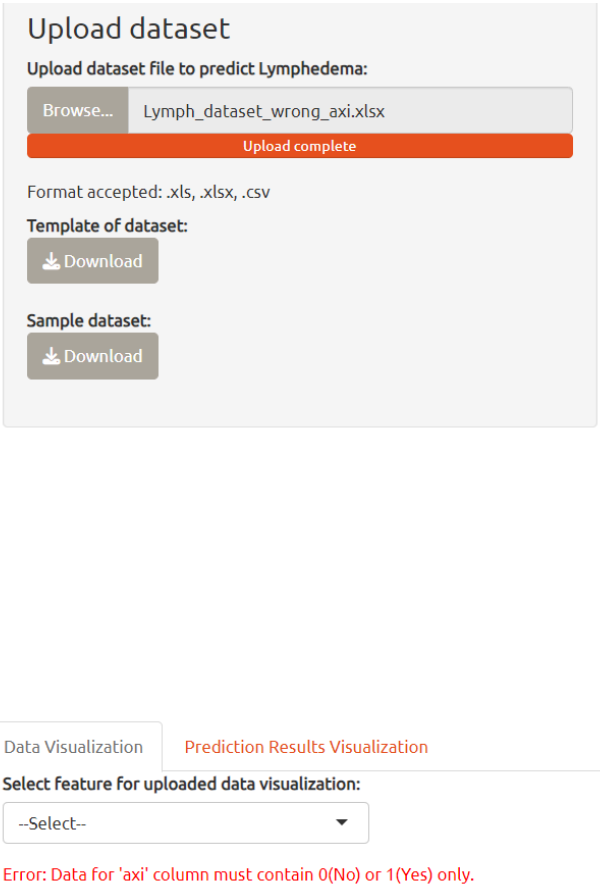


Figure 27: Testing result for invalid data format (invalid values in "axi" column) for data visualization

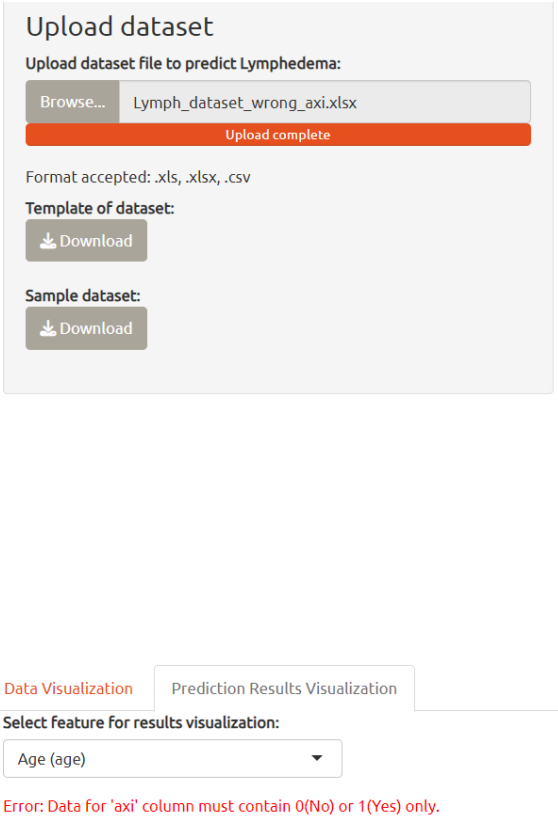


Figure 28: Testing result for invalid data format (invalid values in "axi" column) for *prediction results visualization*

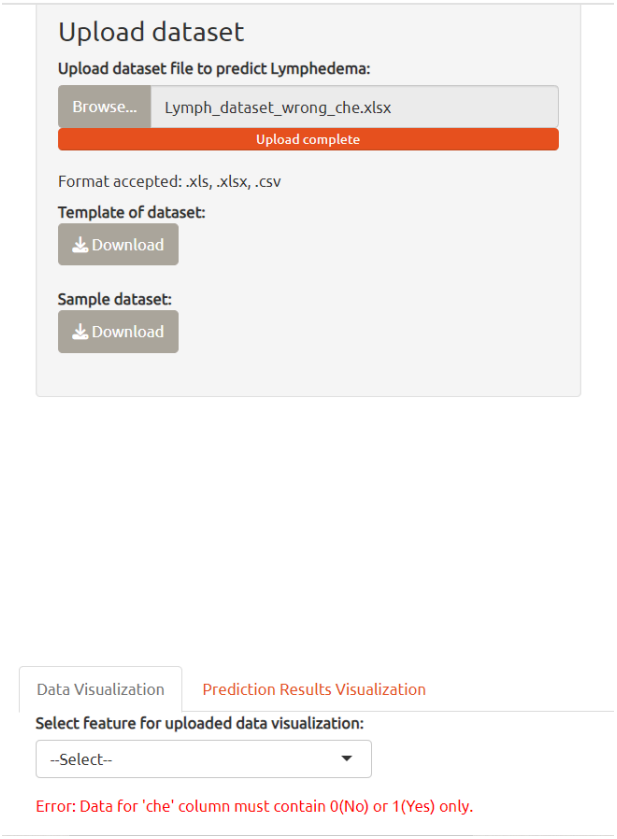


Figure 29: Testing result for invalid data format (invalid values in "che" column) for *data visualization*

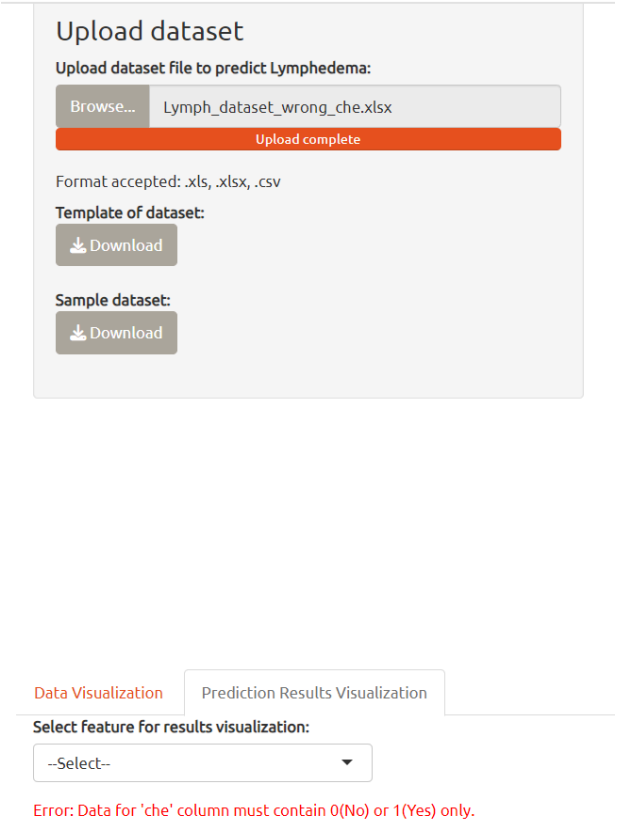


Figure 30: Testing result for invalid data format (invalid values in "che" column) for *prediction results visualization*

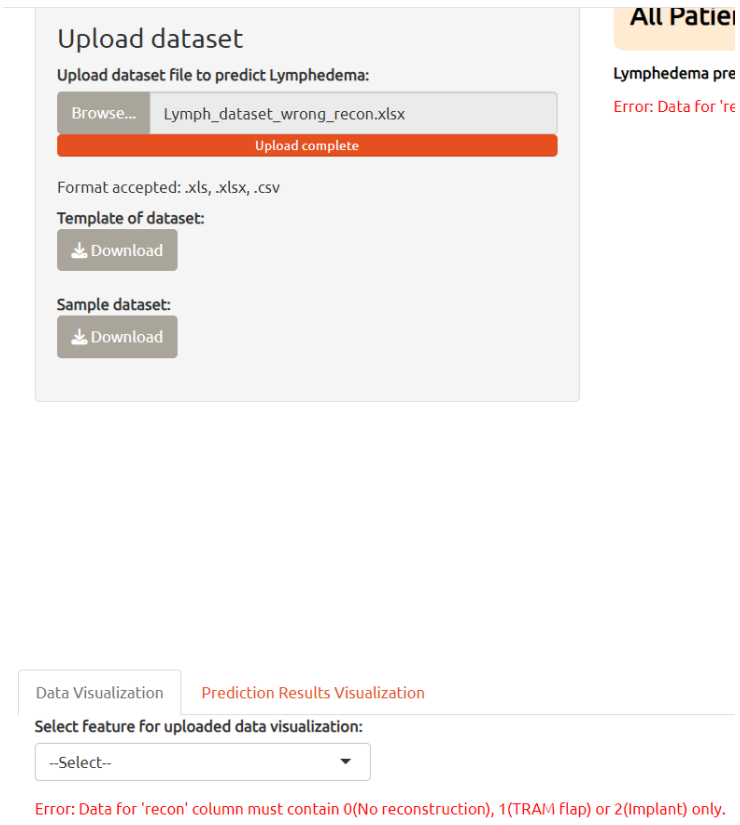


Figure 31: Testing result for invalid data format (invalid values in "recon" column) for *data visualization*

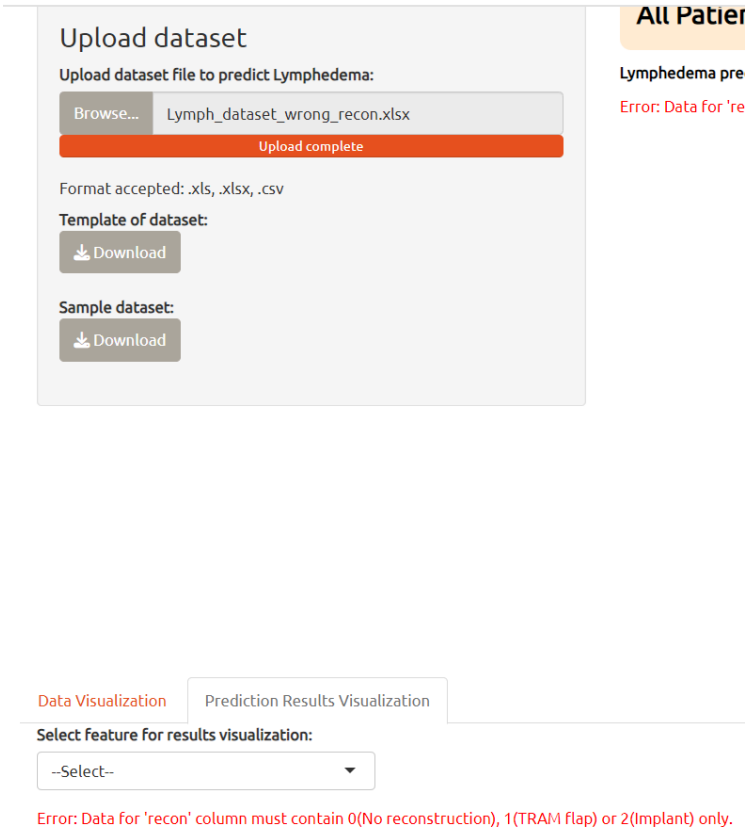


Figure 32: Testing result for invalid data format (invalid values in "recon" column) for *prediction results visualization*

Figure 33: Testing result for invalid data format (invalid values in "tax" column) for *data visualization*

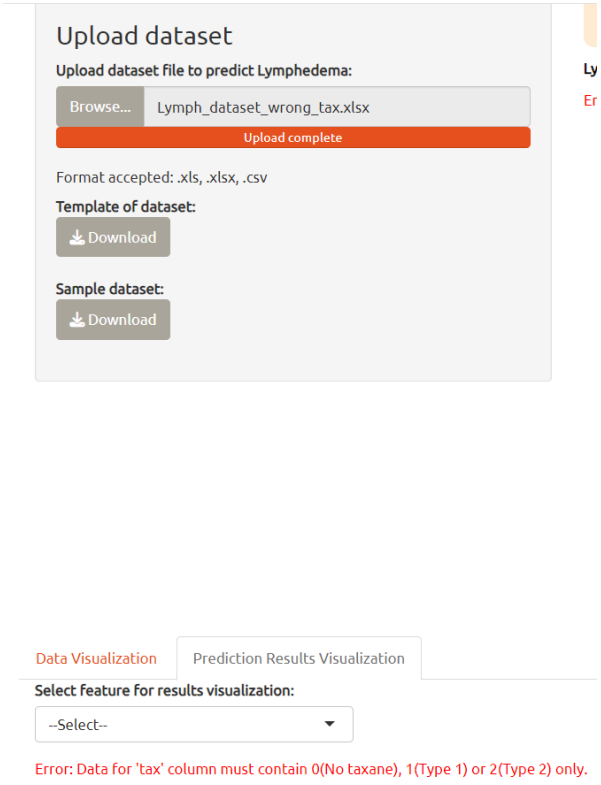


Figure 34: Testing result for invalid data format (invalid values in "tax" column) for *prediction results visualization*

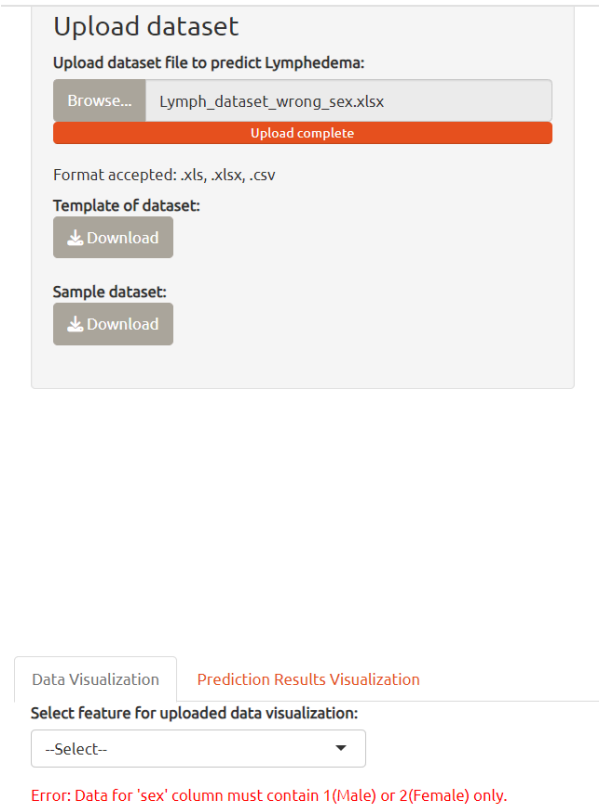


Figure 35: Testing result for invalid data format (invalid values in "sex" column) for *data visualization*

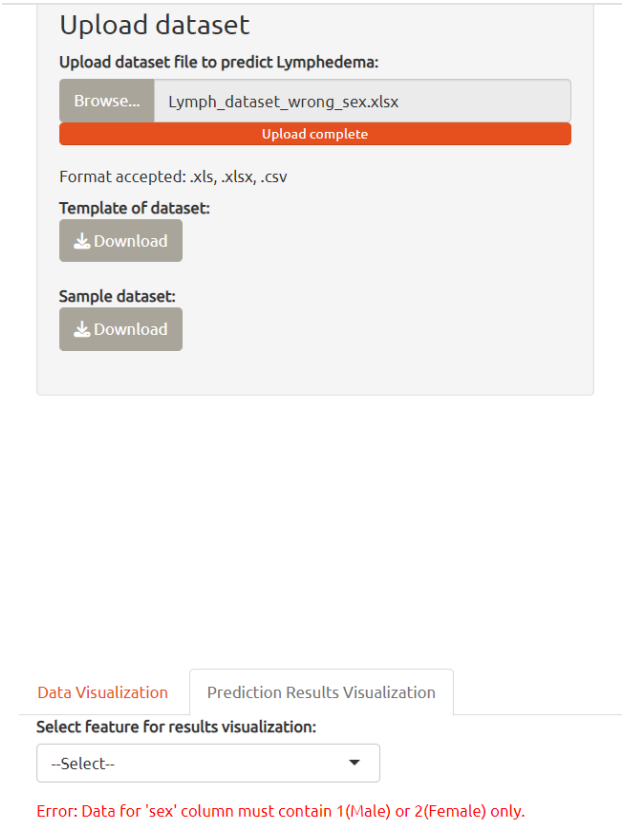


Figure 36: Testing result for invalid data format (invalid values in "sex" column) for *prediction results visualization*

4 Integration Testing

According to Katalon (2024), integration testing is a software testing method that involves gradually combining and testing software components as a cohesive unit. While these components typically function independently without issues, they may encounter problems when integrated. The goal of integration testing is to uncover any defects arising from conflicts between code in software modules as they are integrated together.

One of the reasons why integration testing is crucial is because different programmers may have varying logic and development approaches, leading to functional issues when modules are integrated. Integration testing ensures alignment of the code behind these components, resulting in a functional application. (Katalon, 2024).

There are various types of integration testing, each serving different purposes in the software development life cycle. However, one of the most commonly used and effective approaches is top-down testing. According to Awati (2022), this is an incremental approach that begins testing at the highest-level module and gradually progresses to the lower modules. Each module undergoes individual testing before being integrated to check the overall functionality of the software.

For our integration testing, we chose to use the top-down approach as we believe that it is the most suitable for our needs. We would be able to easily identify the issues that might occur during each integration stage, which reduces the time required for us to identify and fix it otherwise. Furthermore, by using the top-down approach, we are able to separate the integration by the importance of each module, to allow us to focus more on the important modules and ensure that we could have an earlier prototype that is functional.

Test ID	Test Description	Test Method	Expected Output	Actual Output
T25	Test the integration of the web page and the model	Verify that the model prediction is able to be displayed on the web page	Display the prediction results	The prediction is displayed correctly on the web page as shown in Figure 37
T26	Test the accuracy of the model prediction after integration	Apply the same dataset to the model before and after integration to check if the prediction results is the same	The prediction results before and after integration should remain the same	The prediction results before and after integration is the same as shown in Figure 38
T27	Test the integration of the upload dataset component and the model	Upload a dataset using the uploading component	The model is able to correctly process the uploaded dataset to produce a prediction	The prediction of the uploaded dataset is processed by the model and the prediction is correctly displayed as shown in Figure 39
T28	Test the integration of the predicted results visualization and the model	Verify that the predicted results visualization accurately reflects the output generated by the model	The predicted results visualization should reflect the output generated by the model	Visualization accurately reflects the model's output as shown in Figure 40
T29	Test the compatibility of the model after integration	Use various different browser to verify if the model still works as intended	The model works seamlessly on various browsers	The model displays prediction results correctly on all browsers as shown in Figure 41 and 42
T30	Test the performance of the model with larger datasets after integration	Upload a large dataset that exceeds typical dataset size and measure the processing time	The large dataset does not cause the web page to crash and the model is able to correctly process the large dataset within a reasonable timeframe	The large dataset is successfully uploaded and the predicted results are correctly displayed with a slight increase in processing time as shown in Figure 43

Table 6: Test cases for integration testing

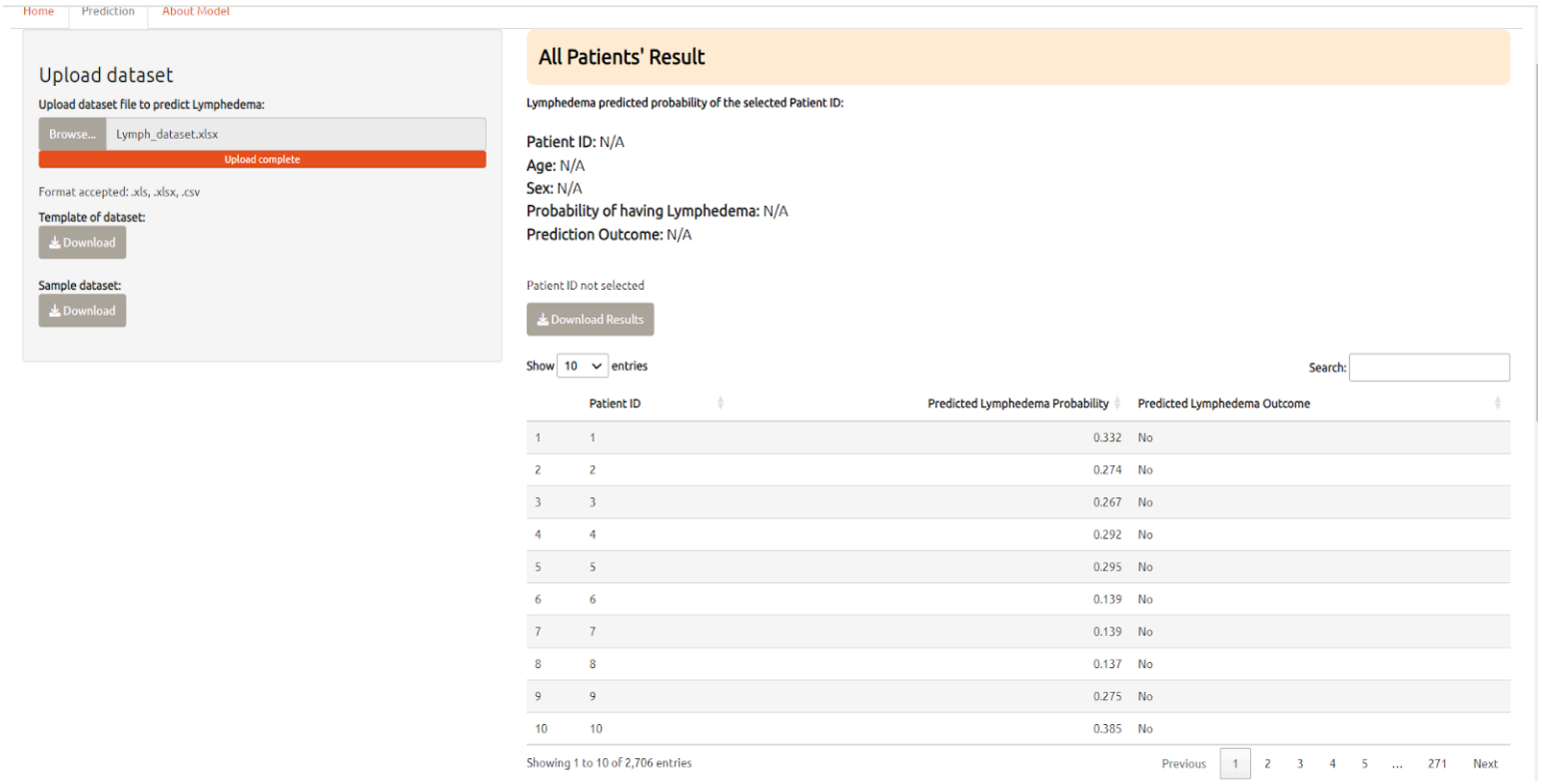


Figure 37: Testing the integration of the web page and the model

	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1		Lymphocyte	Segmented.ne	MCH	MCHC	MCV	Hct	PLT	RBC	WBC	fx	Gy	recon	tax	che	axi	Hb	Prediction.Results.in.R	Predicted.Lymphedema
145	9	35.8	52.7	23.9	31.3	76.5	33.9	240	4.43	5.33	0	0	2	0	0	0	10.6	0	No
146	7	27.3	63	30.1	32.4	93.1	44.2	271	4.75	7.76	0	0	2	0	0	0	14.3	0	No
147	.3	22.6	72.3	30.8	34	90.6	37.6	218	4.15	4.64	25	50	2	2	1	0	12.8	0	No
148	.7	33.3	60.6	30.2	35.2	85.9	40.1	293	4.67	7.63	16	42.56	2	2	1	0	14.1	1	Yes
149	.7	33.3	60.6	30.2	35.2	85.9	40.1	293	4.67	7.63	16	42.56	2	2	1	0	14.1	1	Yes
150	.2	35	57.6	30.2	33.4	90.3	34.4	246	3.81	4.06	16	42.56	2	2	1	1	11.5	0	No
151	.5	18.1	75.8	27.9	32.2	86.6	38.2	287	4.41	7.79	0	0	2	2	0	0	12.3	0	No
152	.5	18.1	75.8	27.9	32.2	86.6	38.2	287	4.41	7.79	0	0	2	2	0	0	12.3	0	No
153	.7	30.2	60.4	29.3	31.8	92	40.2	338	4.37	6.48	0	0	2	2	0	0	12.8	0	No
154	5	21.3	73	32.1	34.4	93.3	40.4	238	4.33	7.26	0	0	2	2	0	0	13.9	0	No
155	5	49.4	44.6	32.5	33.5	97.1	40	227	4.12	5.42	0	0	2	2	0	0	13.4	0	No
156	.2	27.6	62.5	30.4	33.6	90.4	39.6	317	4.38	8.48	0	0	2	2	0	0	13.3	0	No
157	.7	45.7	47.8	30.3	33.6	90.2	36.9	261	4.09	4.88	0	0	2	2	0	0	12.4	0	No
158	.2	26.5	68.5	30.7	32.5	94.6	45.3	328	4.79	6.14	0	0	2	2	0	0	14.7	0	No
159	.8	30.2	62.2	28	33.3	84.2	40.9	321	4.86	5.9	0	0	2	2	0	0	13.6	0	No
160	.8	30.2	62.2	28	33.3	84.2	40.9	321	4.86	5.9	0	0	2	2	0	0	13.6	0	No
161	.1	26.1	64.8	30.3	33.4	90.8	36.5	275	4.02	5.41	0	0	2	2	0	0	12.2	1	Yes
162	.7	31.7	61.5	30.8	33.9	90.6	38.6	210	4.26	4.41	0	0	2	2	0	0	13.1	1	Yes
163	.7	31.7	61.5	30.8	33.9	90.6	38.6	210	4.26	4.41	0	0	2	2	0	0	13.1	1	Yes
164	6	24.9	66.8	31.3	32.5	96.3	41.6	289	4.32	6.66	25	50	2	2	1	1	13.5	0	No
165	6	24.9	66.8	31.3	32.5	96.3	41.6	289	4.32	6.66	25	50	2	2	1	1	13.5	0	No
166	.8	55.7	26.6	32.3	32.7	98.7	38.2	271	3.87	2.87	25	50	2	2	1	1	12.5	1	Yes
167	.2	20.2	74.8	29.8	33.7	88.4	43.6	323	4.93	9.11	16	42.56	2	2	1	1	14.7	0	No
168	.2	16	77.5	31.9	35.1	90.9	37	219	4.07	8.64	0	0	2	2	0	0	13	0	No
169	.7	32.6	60.2	30	32.4	92.6	41.1	193	4.44	4.69	0	0	2	2	0	0	13.3	0	No
170	.1	17.5	75.8	29.4	31.8	92.4	39.9	196	4.32	4.46	0	0	2	2	0	0	12.7	0	No
171	.1	17.5	75.8	29.4	31.8	92.4	39.9	196	4.32	4.46	0	0	2	2	0	0	12.7	0	No
172	4	46.3	49.1	32	35	91.4	40.3	261	4.41	5.49	0	0	2	2	0	0	14.1	0	No
173	.3	35.8	52	30	32.2	93.1	39.4	329	4.23	4.78	25	50	2	2	1	1	12.7	0	No
174	.9	24.6	67.7	29.9	33.2	90	38	201	4.22	7.78	0	0	2	2	0	0	12.6	1	Yes
175	.9	26.6	64.5	30.1	32.3	93.2	42.4	222	4.55	7.93	0	0	2	2	0	0	13.7	0	No
176	.9	26.6	64.5	30.1	32.3	93.2	42.4	222	4.55	7.93	0	0	2	2	0	0	13.7	0	No
177	.7	39.9	53.2	27.1	31.6	86	38	351	4.42	5.84	16	42.56	2	2	1	1	12	1	Yes
178	.3	16.3	76.2	28.5	32.1	88.6	35.8	311	4.04	11.22	25	50	2	2	1	1	11.5	0	No
179	.5	39.2	50.6	25.3	30.5	82.9	37.7	264	4.55	4.8	0	0	2	2	0	0	11.5	0	No
180	.2	35.5	58.3	30.9	32.7	94.5	41.6	201	4.4	6.51	16	42.56	2	2	1	1	13.6	1	Yes
181	.4	31.4	58.5	29.6	32.8	90.1	40.2	254	4.46	4.78	16	42.5	2	2	1	0	13.2	0	No
182	.4	31.4	58.5	29.6	32.8	90.1	40.2	254	4.46	4.78	16	42.5	2	2	1	0	13.2	0	No
183	.3	26.1	68.8	29.9	33.7	88.6	40.3	248	4.55	5.13	0	0	2	2	0	0	13.6	0	No
184	.3	26.1	68.8	29.9	33.7	88.6	40.3	248	4.55	5.13	0	0	2	2	0	0	13.6	0	No
185	.6	47.8	42.5	32.4	34.5	94	43.5	239	4.63	6.65	0	0	2	2	0	0	15	0	No

Figure 38: Comparing the prediction results before and after integration

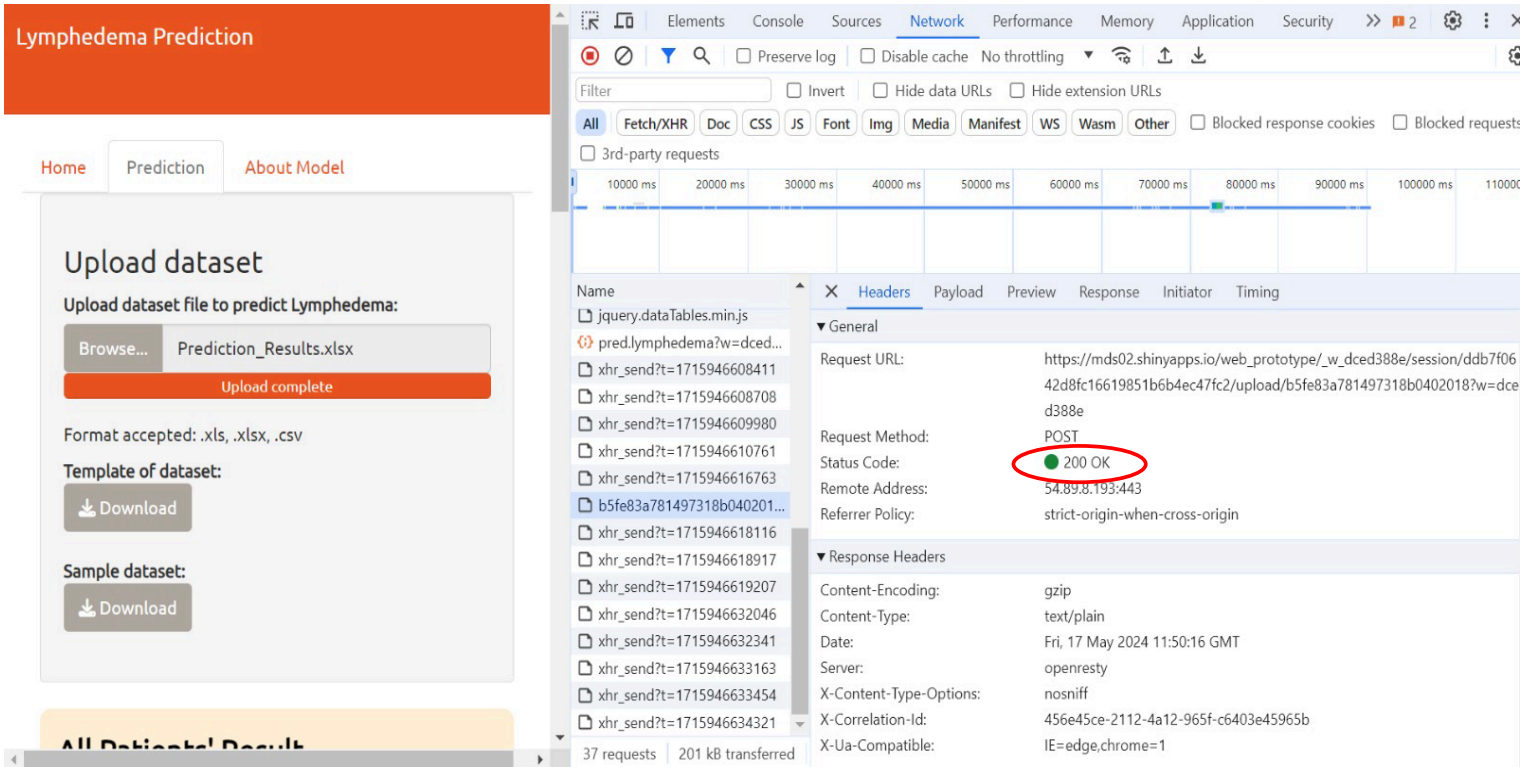


Figure 39: Testing the integration of the upload dataset component and the model

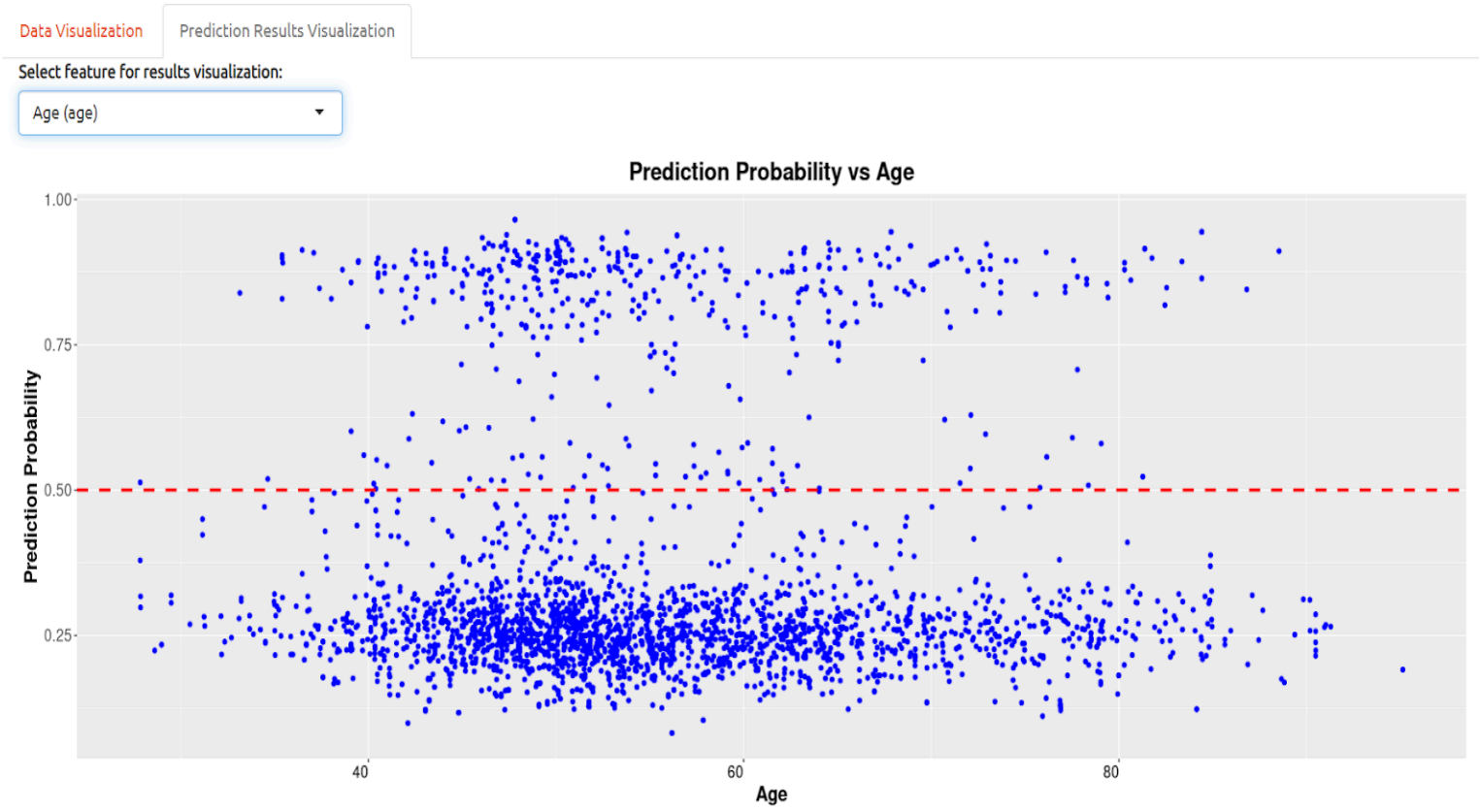


Figure 40: Testing the integration of the predicted results visualization and the model

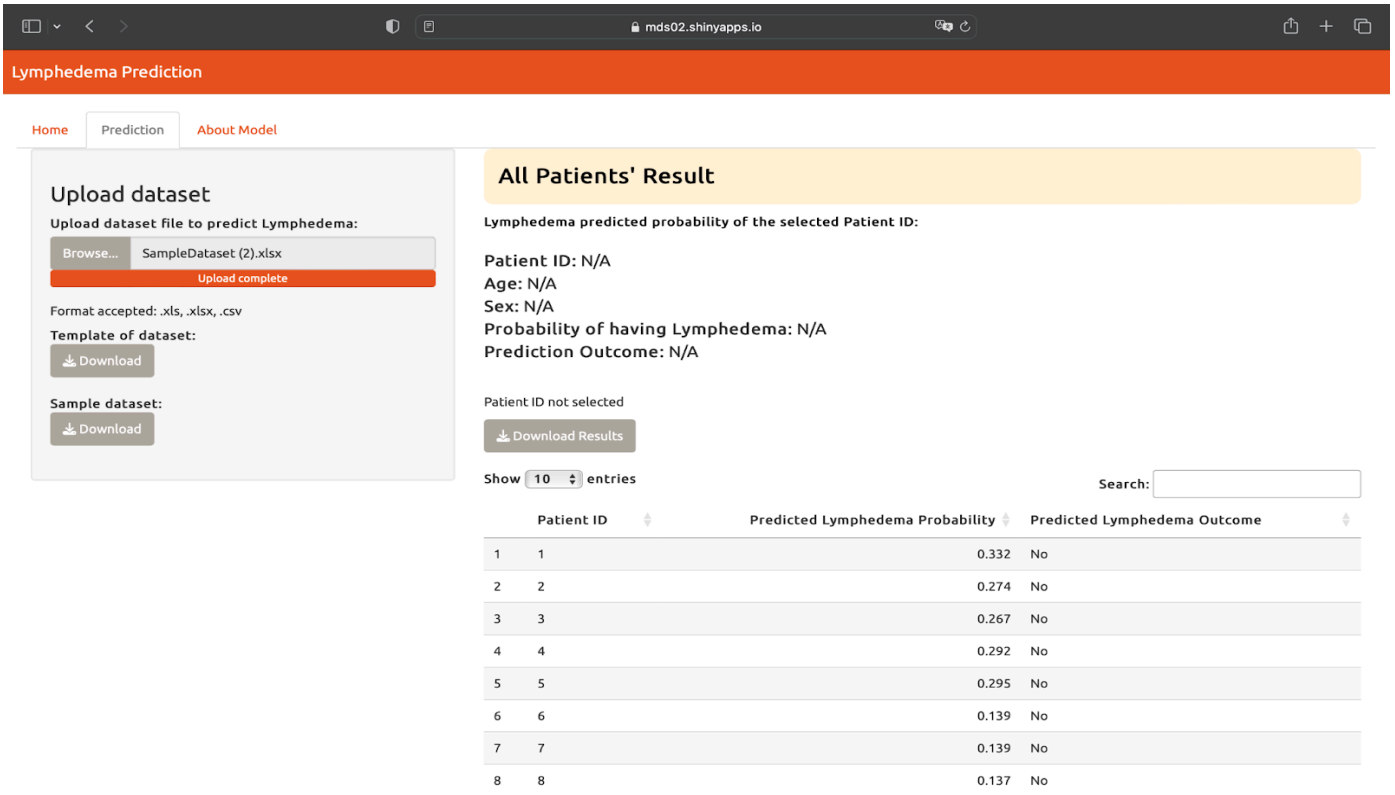


Figure 41: Testing the compatibility of the model after integration (web page in Safari)

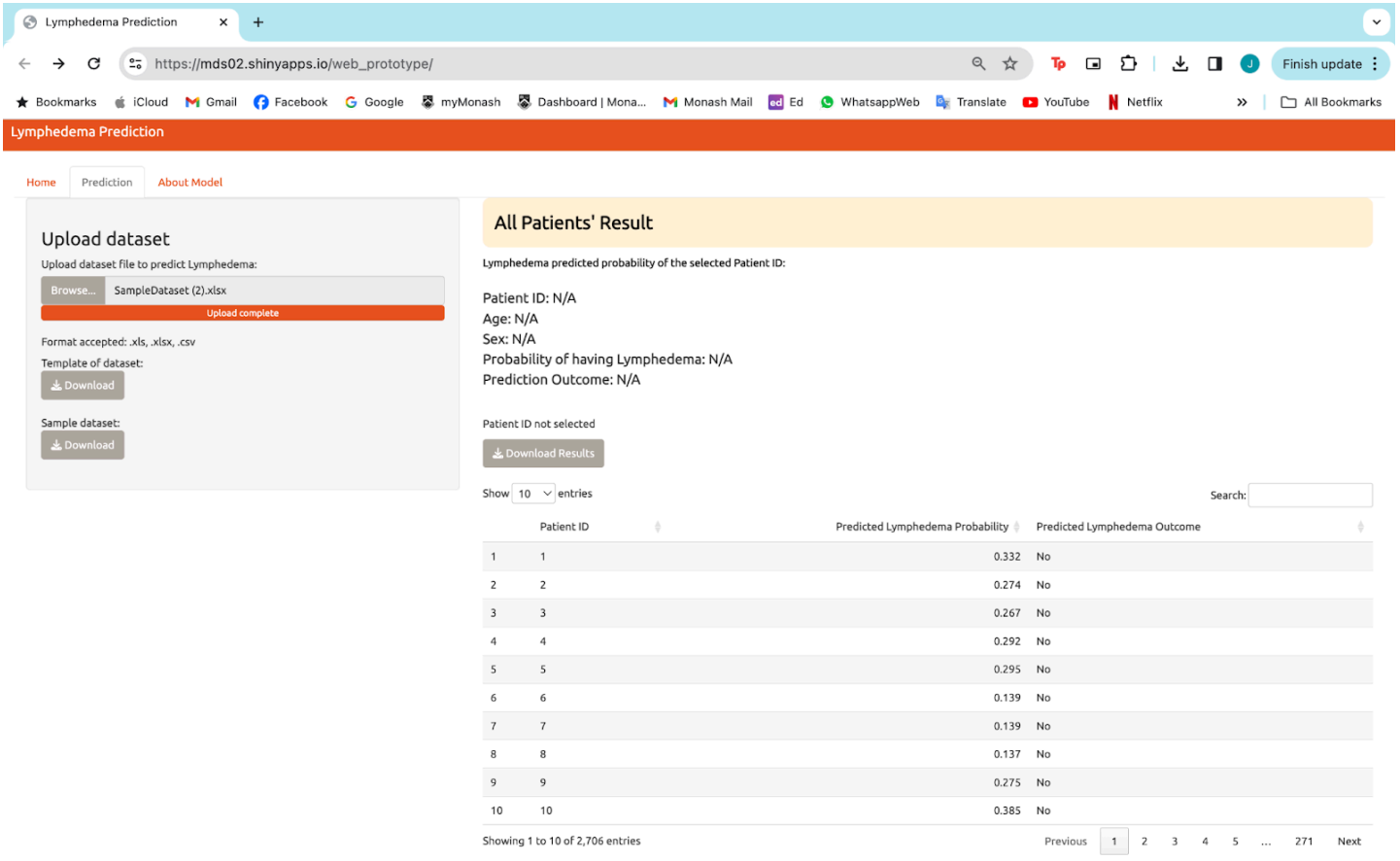


Figure 42: Testing the compatibility of the model after integration (web page in Google Chrome)

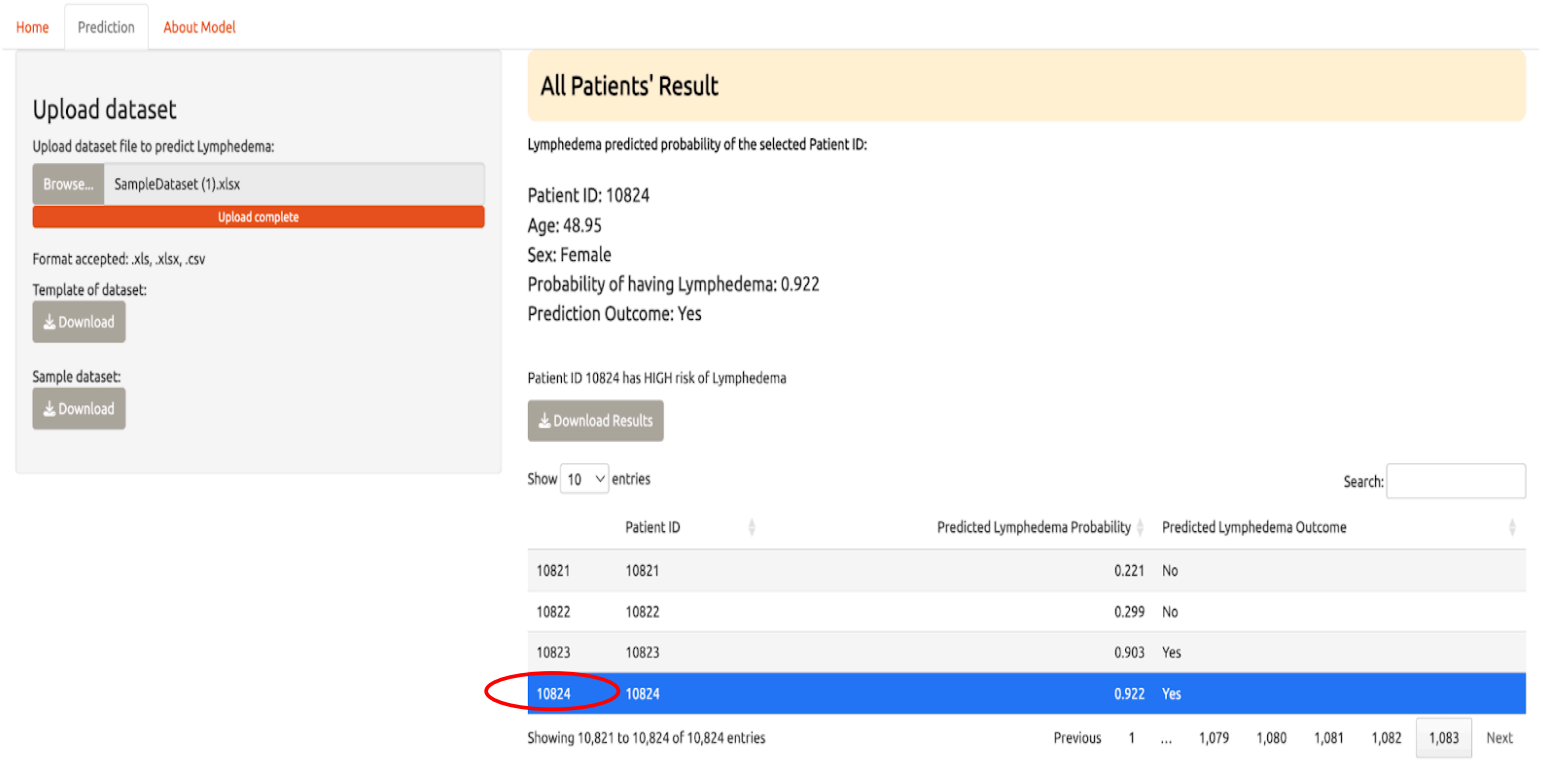


Figure 43: Testing the performance of the model with larger datasets after integration

5 Usability Testing

Usability testing is a crucial step in software and product development, designed to assess how easily and effectively users can interact with a system. It involves observing real users as they complete specific tasks, enabling designers and developers to pinpoint usability problems and identify areas for enhancement (Moran, 2019). The main objective is to make sure the product is intuitive, efficient, and enjoyable to use, emphasizing aspects like navigation ease, instruction clarity, and overall user experience. By gathering direct feedback, usability testing reveals issues that might be missed by the development team, reduces user errors, and boosts user satisfaction. This process is vital for developing products that fulfill user needs and expectations, which in turn leads to higher adoption rates and success in the market. Ultimately, usability testing ensures the final product is not only functional but also delivers a smooth and pleasant user experience.

For our Lymphedema prediction web page prototype, we'd typically test aspects such as the ease of uploading the input dataset, the clarity of the Lymphedema's background information and steps to use our tool as shown, the clarity of the results displayed, the clarity of the description and performance of our predictive model portrayed and the overall user navigation and experience of our web page prototype.

We will be performing the usability testing by manual testing from either our development team and also real users where appropriate. As for real users, we will be requesting a few of our peers to try out our web page prototype. Engaging a third-party team is to ensure unbiased feedback received. The testing technique that we will be utilizing but not limited to is task-based testing, where users perform specific tasks and we observe any usability issues. The tests performed are documented below in Table 7.

Test ID	Test Description	Test Method	Expected Output	Actual Output
T31	Test the ease of reading the description of Lymphedema and steps to use our tool in the Home Tab as shown in Figure 44	Ask real users to read and understand the description of Lymphedema and steps to use our tool	Received positive feedback on the ease of reading and understanding	Received positive feedback on the ease of reading and understanding
T32	Test the ease of understanding the instructions in using the template of dataset provided as shown in Figure 45	Ask real users to fill in the template of dataset based on the given instructions and definition of features or columns provided in the template of dataset	Template of dataset successfully filled up correctly	Template of dataset successfully filled up correctly
T33	Test the ease of uploading the input dataset without much complexity as shown in Figure 46	Ask real users to upload an input dataset themselves based on the prompted instructions in the prototype without given much instructions verbally	Input dataset successfully loaded completely	Input dataset successfully loaded completely
T34	Test the ease of understanding the results information box and table after prediction is made as shown in Figure 47	Ask real users to read and understand the information and results in the results information box and table	Received positive feedback on the ease of reading and understanding	Received positive feedback on the ease of reading and understanding
T35	Test the ease of searching through a particular patient or record and navigating through different pages in the results table as shown in Figure 48	Ask real users to search for a particular patient or record using the search bar, and try to click on the page number to navigate through different pages in the results table	Search performed generates correct searched results and page number navigated through correct different pages in the result table	Search performed generates correct searched results and page number navigated through correct different pages in the result table
T36	Test the ease of understanding the visualizations for the input dataset as shown in Figure 49	Ask real users to interpret and understand the visualizations provided for the input dataset	Received positive feedback on the ease of reading and understanding	Real users mentioned that there it could be quite challenging to understand the visualizations without context

T37	Test the ease of understanding the visualizations for the prediction results as shown in Figure 50	Ask real users to interpret and understand the visualizations provided for the prediction results	Received positive feedback on the ease of reading and understanding	Real users mentioned that there it could be quite challenging to understand the visualizations without context
T38	Test the ease of reading the description and performance of the predictive model in the About Model Tab as shown in Figure 51	Ask real users to read and understand the description and performance of the predictive model	Received positive feedback on the ease of reading and understanding	Received positive feedback on the ease of reading and understanding
T39	Test the ease of navigating through different tabs in the prototype as shown in Figure 52	Ask real users to navigate through different tabs in the prototype back and forth (i.e., Home, Prediction, About Model, Data Visualization and Prediction Results Visualization Tabs)	Tabs could be navigated through easily back and forth	Tabs could be navigated through easily back and forth
T40	Test the ease of reading regarding to the font size, font style and layout of the entire prototype as shown in Figure 53	Ask real users to glance and read through the contents across the entire prototype	Received positive feedback on the ease of reading and understanding regarding to the font size, font style and layout	Received positive feedback on the ease of reading and understanding regarding to the font size, font style and layout
T41	Test the suitability of the color selection for the buttons, main bar and visualizations of the entire prototype as shown in Figure 54	Ask feedback from the real users on their thoughts on the color selection for the buttons, main bar and visualizations of the entire prototype	Received positive feedback on the color selection for the buttons, main bar and visualizations of the entire prototype	Received positive feedback on the color selection for the buttons, main bar and visualizations of the entire prototype
T42	Test the responsiveness and load time of the prototype as shown in Figure 55	Load the web page across multiple devices (of the development team) and note the responsiveness and load time for all of the features (uploading dataset, visualizations, results, etc.)	Web page features and components loads within a few seconds and buttons are responsive almost instantly	Web page features and components loads within a few seconds and buttons are responsive almost instantly

Table 7: Test cases for usability testing

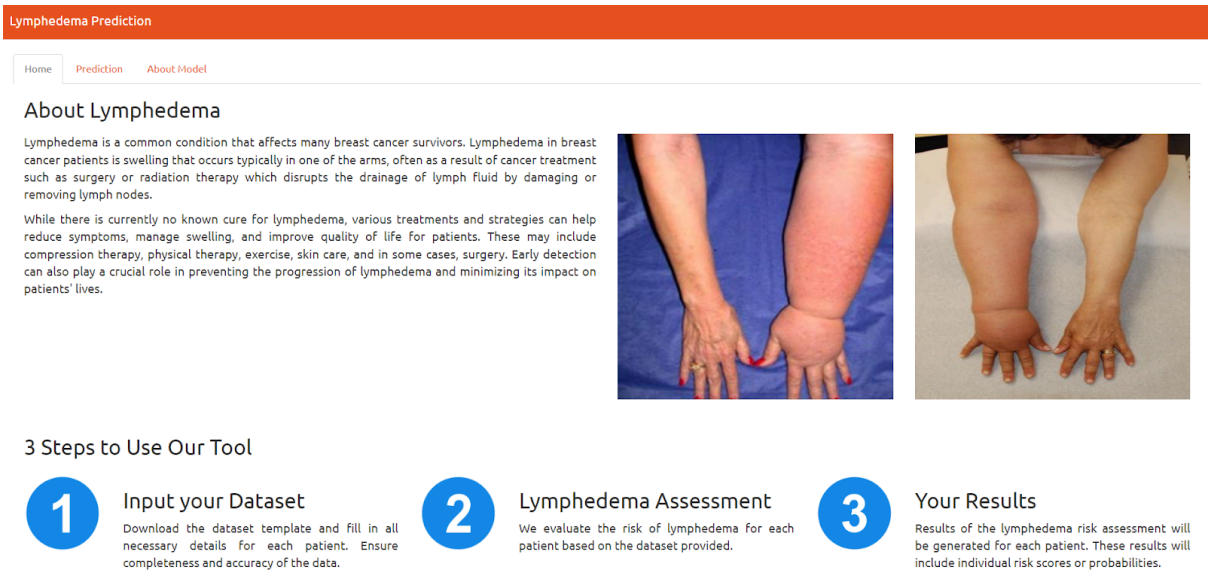


Figure 44: Testing ease of reading in Home Tab

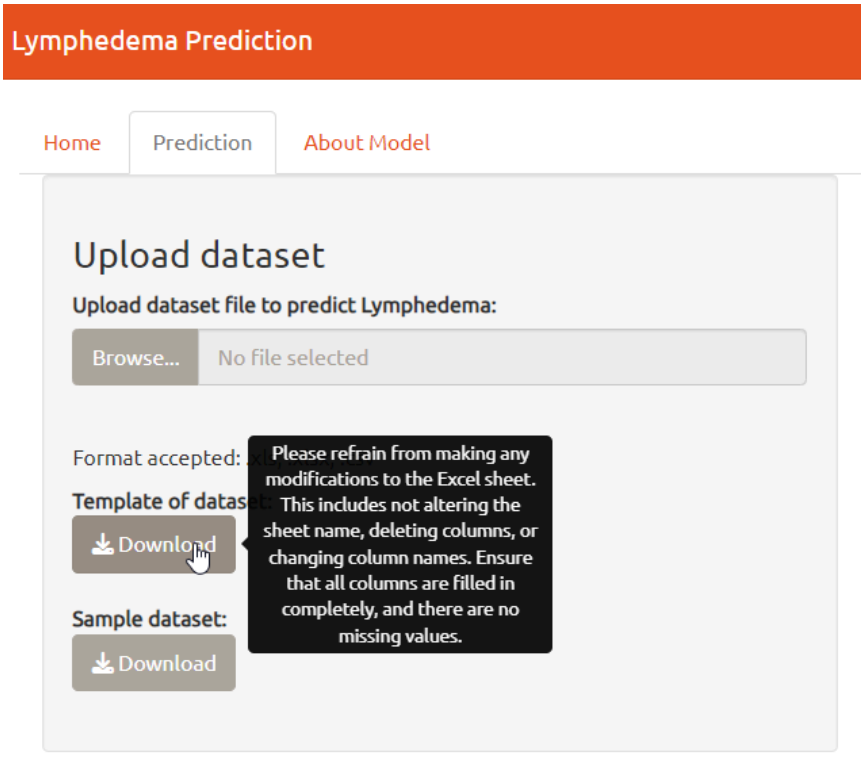


Figure 45: Testing ease of understanding instructions for template of dataset

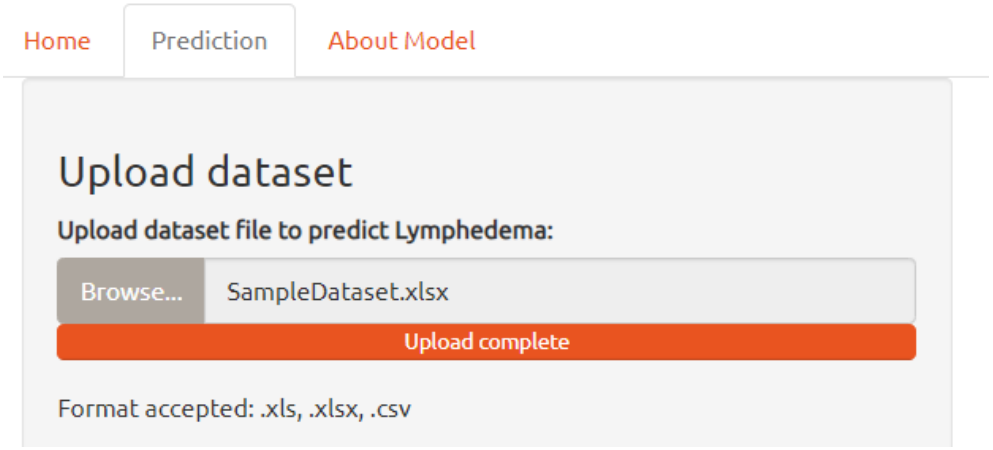


Figure 46: Testing ease of uploading input dataset

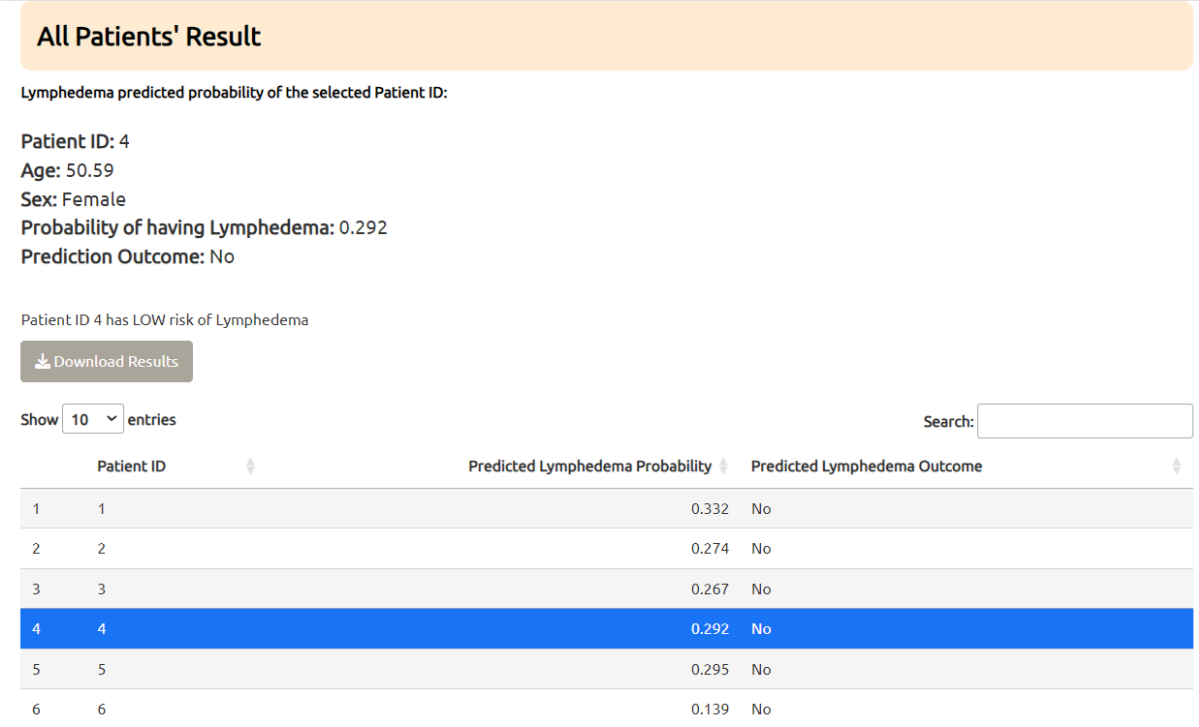


Figure 47: Testing ease of understanding results information box and table

Show 10 entries Search: 23

	Patient ID	Predicted Lymphedema Probability	Predicted Lymphedema Outcome
22	22	0.231	No
23	23	0.239	No
31	31	0.233	No
61	61	0.239	No
65	65	0.237	No
66	66	0.237	No
69	69	0.238	No
82	82	0.232	No
89	89	0.236	No
92	92	0.237	No

Showing 1 to 10 of 355 entries (filtered from 2,706 total entries) Previous 1 2 3 4 5 ... 36 Next

Figure 48: Testing ease of searching for particular patient or record and navigation through different pages in the results table

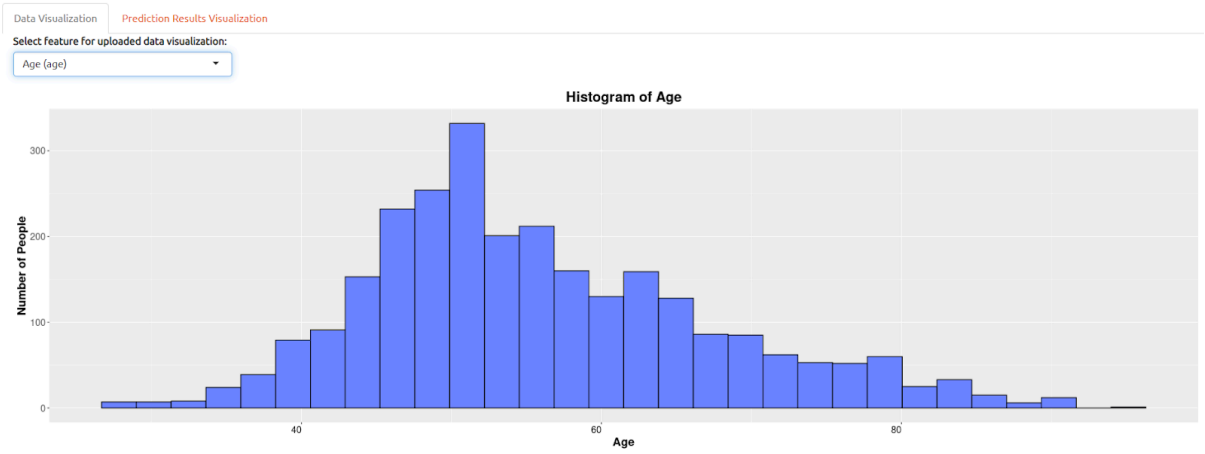


Figure 49: Testing ease of understanding the visualizations for the input dataset

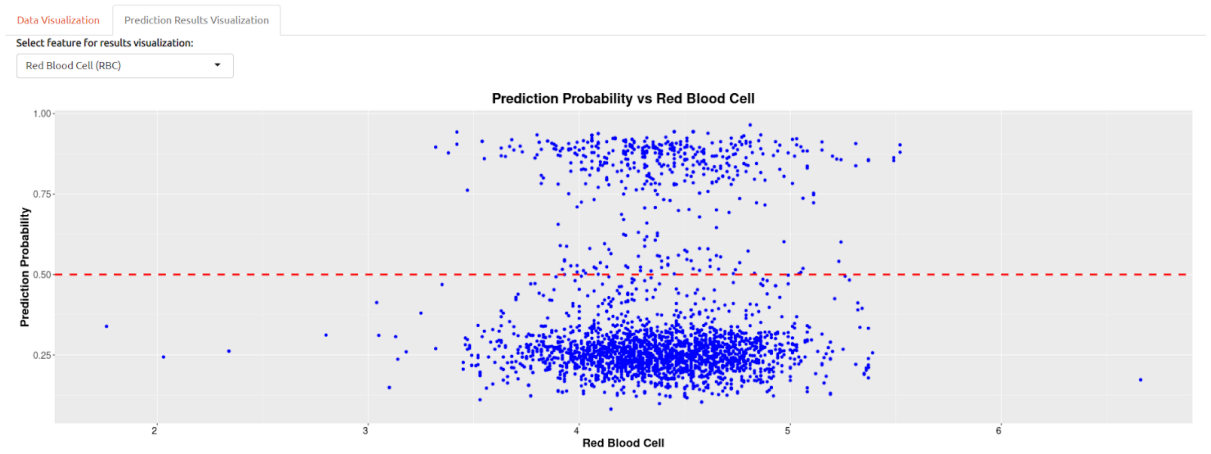


Figure 50: Testing ease of understanding the visualizations for the prediction results

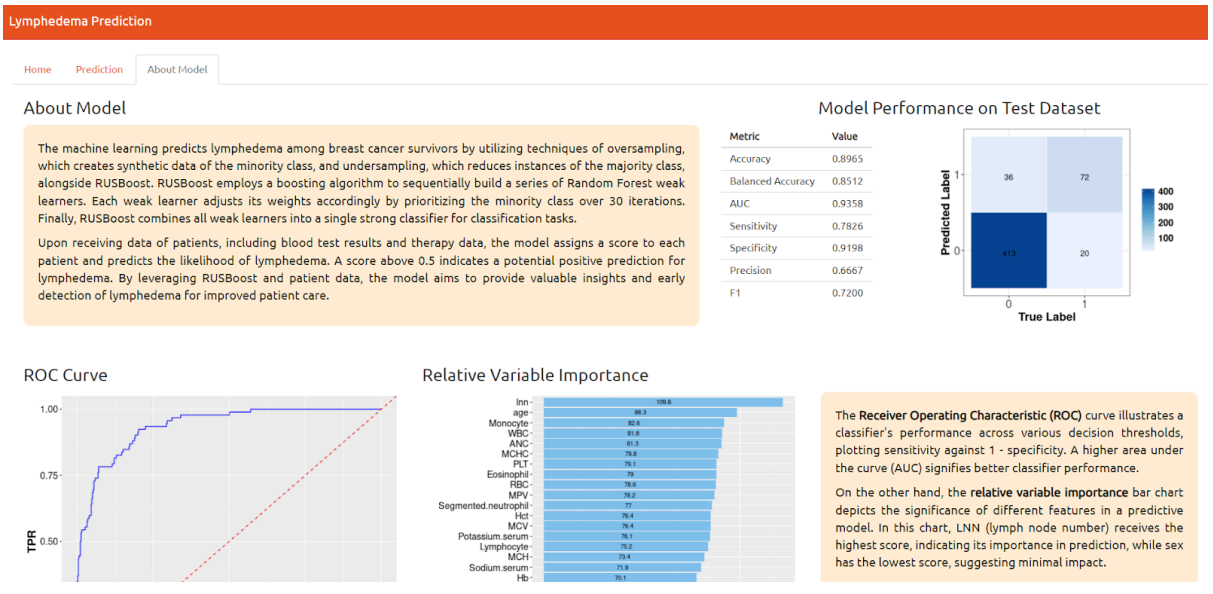


Figure 51: Testing ease of reading in About Model Tab

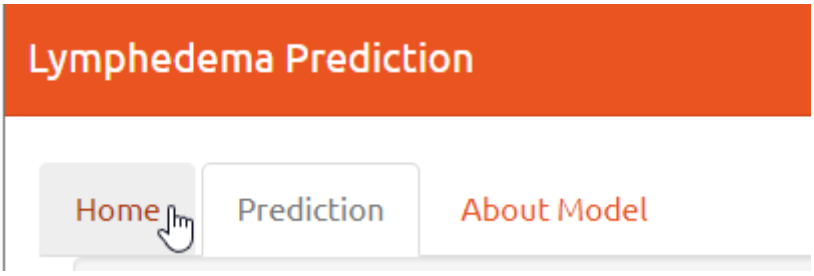


Figure 52: Testing ease of navigating through different tabs

Figure 53: Testing ease of reading regarding to font size, font style and layout of the entire prototype

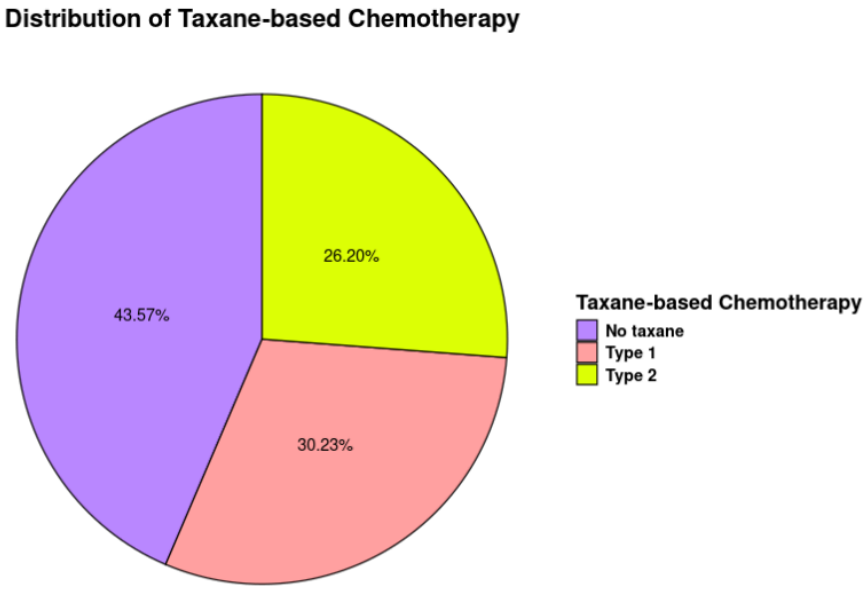


Figure 54: Testing suitability of the color selection for the buttons, main bar and visualizations of the entire prototype

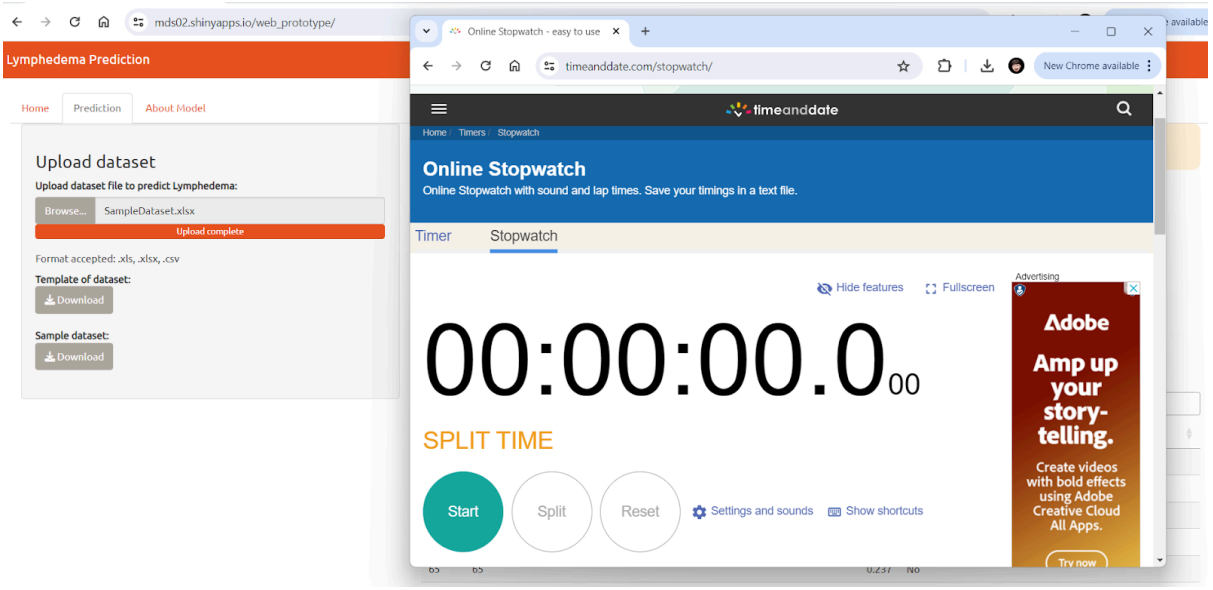


Figure 55: Testing responsiveness and load time of the prototype

6 Recommendation for Improvements

While we were conducting the usability testing (task-based testing) involving real users to perform specific tasks and we observed any usability issues, we received constructive feedback where they suggested several improvements for our Lymphedema prediction web page prototype. Based also on the test results documented above in Table 7, the real users suggested that the visualizations for the input dataset as well as the prediction results should include several contexts explaining the charts or graphs, as the one without contexts that we have right now could be deemed to be challenging to interpret and understand. As committed developers, we swiftly addressed the suggestions and made potential enhancements to our web page prototype, hence the newly refined version of our web page prototype is now reflected.

Besides, incorporating interactive elements such as tooltips, progress bars, and informative pop-ups can significantly enhance the user experience by guiding users through the process. Personalization features, such as allowing users to create profiles and save their data, can lead to more tailored predictions and enable tracking over time. Including a feedback mechanism where users can provide their input or report issues directly through the web page can help to continuously improve the service. Adding educational content about lymphedema, its symptoms, and management strategies, along with a comprehensive FAQ section, is also deemed to be a valuable resource for users.

Moreover, on the backend, regularly updating and retraining the model is crucial to maintain its relevance and accuracy. This could be achieved by continuously updating the dataset with new patient data ensures the model adapts to evolving symptoms and patterns. Implementing a pipeline for continuous training and validation also allows the model to stay current. Furthermore, monitoring and evaluating model performance using metrics like precision, recall, F1-score, and AUC-ROC helps in gauging the model's effectiveness and making necessary adjustments.

In addition, adding analytics and reporting capabilities can also provide valuable insights. User analytics can help track interactions and gather insights into how the web page is being used, leading to informed improvements. Offering custom reports that users can generate and share with healthcare providers can make the tool more practical and useful in real-world settings.

7 Limitations of Testing Process

The testing process for the lymphedema web page prototype faces several notable limitations affecting its thoroughness and reliability. A primary issue is the reliance on manual testing. Manual tests must be repeated with every system modification or enhancement, making the testing process time-consuming whenever we have modification to our software or system, and all of these testing processes will also be human error-prone, thus reducing efficiency and accuracy of our testing processes.

Besides, the load time and responsiveness testing of the web page prototype also vary across different devices or computers. This is largely influenced by the computational power of individual computers. This variability makes it challenging to consistently assess the system's performance, as acceptable loading times on one machine may differ significantly on another due to varying hardware and environmental factors, thus introducing uncertainty in evaluating the system's efficiency and effectiveness across different computing environments.

Furthermore, there are also certain edge cases or rare user scenarios to be untested as test coverage gaps exist. There is also discrepancy between the test results and the actual user experience as there is a gap between the testing data or process, with the actual real-world usage behaviors by real users. Hence, this will definitely lead to potential bugs or performance issues to be present even after all of the testing processes have been completed and passed in this document. Therefore, it is important to ensure that the testing data and environment that we are utilizing could actually simulate the real-world data, scenarios and environments as much as possible, representing a wide range of real-world use cases. Thus, this could help to eliminate incomplete or skewed insights into the software's performance.

8 Conclusion

In conclusion, our comprehensive testing process has successfully verified the functionality, performance, and usability of the lymphedema prediction web page prototype. Through a systematic approach outlined in our test plan, we met our test objectives by applying blackbox testing to the prediction and visualization components, ensuring they work as intended. Integration testing further validated the smooth interaction between different components, showcasing the robustness of our system. Usability testing has provided our team with insights into user experience that are valuable to us, and allows us to enhance our prototype to improve the ease of use.

There were some limitations in the testing process, primarily due to constraints in resources and time, which may have affected the thoroughness of testing in certain aspects. Moving forward, we plan to do continuous testing and make iterative improvements to tackle these limitations and further improve the prototype. Our findings and recommendations highlight the significance of thorough testing in achieving a reliable and user-friendly tool for predicting lymphedema in breast cancer survivors.

References

Ashtari, H. (2022). Black Box vs. White Box Testing: Understanding 3 Key Differences. Retrieved from https://www.spiceworks.com/tech/devops/articles/black-box-vs-white-box-testing/

Awati, R. (2022). Integration Testing or Integration and Testing (I&T). Retrieved from https://www.techtarget.com/searchsoftwarequality/definition/integration-testing

Fu, M. R., Wang, Y., Li, C., Qiu, Z., Axelrod, D., Guth, A. A., Scagliola, J., Conley, Y., Aouizerat, B. E., Qiu, J. M., Yu, G., Van, C. J. H., Haber, J. & Cheung, Y. K. (2018). Machine learning for detection of lymphedema among breast cancer survivors. *mHealth*, *4*, 17. doi:10.21037/mhealth.2018.04.02.

Imperva (n.d.). Black Box Testing. Retrieved from https://www.imperva.com/learn/application-security/black-box-testing/

Katalon (2024). What is Integration Testing? Definition, How-to, Examples. Retrieved from https://katalon.com/resources-center/blog/integration-testing

Moran, K. (2019). Usability testing 101. Retrieved from https://www.nngroup.com/articles/usability-testing-101/

Nidhra, S. (2012). Black Box and White box testing Techniques - A literature review. *International Journal of Embedded Systems and Applications, 2*(2), 29–50. https://doi.org/10.5121/ijesa.2012.2204

Use of generative AI declaration
We hereby acknowledge the use of ChatGPT to shorten and improve sentences. The prompts entered include:

- [sentences] shorten and improve the sentences.
- Review and audit mechanisms for Lymphedema prediction project
- Pseudocode for ANN, Decision Tree and Logistic Regression

The outputs generated were modified and incorporated into this document