FIT3164 Semester 1, 2024:
Final Project Report

Team: MDS02

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

Word Count: *8249*

Table of Contents

1 Introduction

Lymphedema is regarded as a chronic condition characterized by the accumulation of lymphatic fluid, leading to swelling, primarily in the arms or legs (Mayo Clinic, 2022). It commonly occurs following the removal of lymph nodes during cancer treatments, particularly breast cancer. Despite the significant impact on patients' quality of life, there is currently no cure for lymphedema. However, early detection and timely intervention are crucial for effective management and to prevent progression to more severe stages. Traditional methods for diagnosing lymphedema, such as limb measurements, MRI scans, and CT scans, require patients to visit healthcare facilities, often causing delays in diagnosis and treatment. To address this challenge, our project explores the use of machine learning approaches, which excels in analyzing large datasets and identifying patterns, to predict lymphedema.

The project aimed to explore and analyze existing literature and data sources related to lymphedema prediction, develop an enhanced machine learning algorithm specifically for predicting lymphedema in breast cancer survivors, and create a prototype for both prediction and visualization of lymphedema. The project's scope involved using clinical data and patient profiles to implement the machine learning model, proposing enhancements to current algorithms, and developing a prototype with user-friendly interfaces for displaying and visualizing the results. This project was executed through several phases, including project management planning, design and prototype development, progress reporting, conducting a literature review, developing the final prototype, and comprehensive reporting.

The methodology encompasses several key phases. Initially, data collection and preprocessing are performed to gather relevant data, handle class imbalances, and prepare for model training. This includes obtaining a dataset with patient information, such as blood test results and therapy details, from Bundang Seoul National University Hospital (Trinh et al., 2023), and using resampling techniques to address class imbalances. Baseline performance metrics are then established using traditional machine learning algorithms like logistic regression, decision trees, and artificial neural networks (ANN) to guide optimization and highlight strengths and weaknesses. Extensive experimentation follows with various machine learning and ensemble techniques, focusing on the RUSBoost algorithm, which effectively handles class imbalance and improves accuracy (Trinh et al., 2023). Finally, the predictive model is integrated into a Shiny-based web application, ensuring user-friendliness and practicality for clinicians and patients. The web application facilitates data input, prediction, and result visualization, providing a comprehensive and accessible tool for early detection and ongoing monitoring. The significance of this project lies in its potential to transform the early detection and management of lymphedema. By providing a reliable predictive tool, healthcare providers can implement preventive measures and personalized treatment plans more effectively. The web application will facilitate continuous monitoring and timely interventions, ultimately improving patient outcomes and quality of life.

This report typically discusses a few sections, namely project background, project outcomes, software methodology, software deliverables, critical discussion on our project, and finally a cohesive conclusion to wrap up the entire report.

2 Project Background

2.1 Brief Background of Project

Figure 1: Lymphedema swelling

Figure 2: Lymphedema delayed infections

Lymphedema is a persistent medical condition resulting from the buildup of lymphatic fluid, causing swelling, especially in the arms or legs, as shown in Figure 1 (PhysioMotion, n.d.). This condition frequently arises following lymph node removal or damage during cancer treatments such as surgery or radiation therapy (Trinh et al., 2023). Prompt detection and treatment are vital for effectively managing lymphedema, as delays can lead to severe complications, including infections, decreased mobility, and a lower quality of life (Fu et al., 2018). Figure 2 shows the infections (dots) of lymphedema due to delays in detection and treatment (Sheehan, 2023). However, forecasting the onset of lymphedema is challenging due to the numerous risk factors involved and individual variability in patient responses (Fazeli et al., 2017).

Traditional diagnostic methods for lymphedema, such as limb measurements, MRI, CT scans, and ultrasound, necessitate patient visits to healthcare facilities, leading to delays and increased costs. These methods are often impractical for widespread early detection due to their expense and time consumption (Fu et al., 2018). Recent advancements have focused on leveraging machine learning techniques for early lymphedema prediction. Machine learning, a subset of artificial intelligence, excels at analyzing large datasets and identifying patterns, making it a promising tool for medical predictions (Trinh et al., 2023). Besides, in Figure 3, we can also see that lymphedema diagnosis is increasing yearly, hence the need for an effective tool to detect and predict lymphedema (Markets and Markets, 2019).

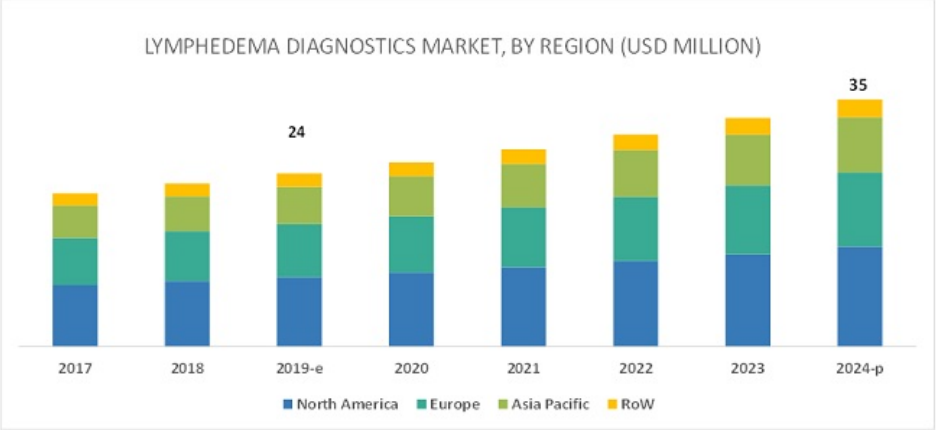

Figure 3: Lymphedema diagnostics market, by region (USD million)

2.2 Rationale of Project

The rationale for this project lies in the necessity to improve early lymphedema detection and management through advanced machine learning techniques. Previous studies have demonstrated the feasibility of using machine learning algorithms to predict lymphedema. For instance, Fu et al. (2018) developed an artificial neural network (ANN) model based on real-time symptom reporting, achieving high accuracy, sensitivity, and specificity. Fazeli et al. (2017) utilized data mining techniques to identify lymphedema risk factors and predict its occurrence among breast cancer patients. Despite these advancements, there remains a need for models that can utilize diverse data sources, such as blood tests and therapy data, to enhance prediction accuracy and early detection (Trinh et al., 2023).

By incorporating a broader range of data, including complete blood counts (CBC), serum tests, and other therapy information, this project aimed to develop more robust predictive models as compared to the proposed predictive model in Trinh et al. (2023). Thus, the lymphedema predictive model that we have built will be integrated into a user-friendly web application using Shiny in R, providing an accessible tool for healthcare providers and patients for early detection and intervention. This approach not only addresses the limitations of traditional diagnostic methods but also leverages the strengths of machine learning to offer a practical and efficient solution for managing lymphedema.

Hence, this project seeks to enhance existing research by developing an enhanced machine learning model for lymphedema prediction by incorporating RUSBoost technique, utilizing varied studies (research papers) and data sources, thus integrating the predictive model that we have built into an interactive web-based application. This initiative is expected to improve early detection and facilitate timely interventions for individuals at risk of developing lymphedema. In summary, the main contributions of this project can be summarised as:

I. Developed a more robust predictive model as compared to all the research papers or studies, by using the RUSBoost algorithm, which effectively handles class imbalances. This model significantly improves the prediction accuracy for lymphedema.

II. Conducted initial experiments with traditional machine learning algorithms like logistic regression, decision trees, and ANN. From there, we established baseline performance metrics that guided our optimization efforts, especially in addressing class imbalance, ensuring the robustness and accuracy of our predictive model.

III. The robust lymphedema predictive model was then seamlessly integrated into a Shiny-based web application prototype. This application allows for user-friendly data input (uploading own dataset), making prediction, dataset visualization, prediction result visualization, and prediction results download, thus making it practical for clinical use. The visualization tools also facilitate and enhance the understanding of underlying patterns and trends in the data and prediction results.

IV. Users are allowed and able to download the prediction results for further analysis and record-keeping. Additionally, the application includes detailed model descriptions that offer transparency and build user trust in the prediction system.

V. The webpage prototype user interface is designed to be intuitive and easy to navigate, ensuring that healthcare providers and patients can use the application effectively. Features like automated data validation and error handling further improve user experience, where all of these were not included in the previous studies or research papers.

2.3 Literature Review and Background Work

2.3.1 Introduction to Lymphedema and its Challenges

Lymphedema, characterized by the accumulation of lymphatic fluid leading to swelling, poses significant challenges in the medical field. Primarily affecting the arms or legs, it commonly occurs following cancer treatments such as surgery or radiation therapy, particularly in breast cancer patients. Early detection and effective management of lymphedema are crucial to prevent progression to more severe stages. This literature review provides an updated perspective on the advancements in lymphedema prediction using machine learning models, incorporating the latest research and practical implementations.

2.3.2 Predictive Models Using Symptom-Based Data

Fu et al. (2018) developed an artificial neural network (ANN) model based on real-time symptom reports to predict lymphedema among breast cancer survivors. Achieving a high accuracy of 93.75%, sensitivity of 95.65%, and specificity of 91.03%, this study demonstrated the potential of machine learning in enhancing early detection through symptom-based data. This approach underscores the importance of real-time monitoring and reporting of symptoms to improve early detection rates. While Fu et al. (2018) focused on real-time symptom reports, our project integrated a broader range of clinical data, including blood tests and therapy information. This shift allowed us to leverage more comprehensive data, potentially improving the robustness and accuracy of our predictive model.

2.3.3 Incorporating Diverse Clinical Data

Fazeli et al. (2017) utilized data mining techniques, including neural networks and support vector machines (SVM), to identify risk factors and predict lymphedema in breast cancer patients. Analyzing data from 933 patients, the study found that factors such as heaviness, type of surgery, body mass index, and radiotherapy were significant predictors. The SVM model showed a sensitivity of 82.87% and accuracy of 77.49%. This research highlights the need for incorporating diverse clinical data to improve predictive accuracy and underscores the value of a multi-faceted approach in predicting lymphedema. Fazeli et al. (2022) further expanded on this by collecting data from various sources, including blood tests and therapy records, to build models that could predict lymphedema with higher accuracy. This study demonstrated the effectiveness of using comprehensive clinical data for early detection, reinforcing the findings of their earlier work and emphasizing the need for diverse data inputs. Similar to Fazeli et al. (2022), our project also incorporated diverse clinical data. However, we went further by implementing advanced machine learning techniques like RUSBoost, which combined resampling and boosting to handle class imbalance more effectively. This approach improved the model's performance and addressed one of the key challenges highlighted in their research.

2.3.4 Advances in Data Types and Machine Learning Techniques

Trinh et al. (2023) proposed using new data types, such as complete blood count (CBC) tests, serum tests, and therapy information, to develop predictive models for lymphedema. Collecting data from 2,137 patients, they employed various machine learning algorithms, including random forest and gradient boosting. The random forest model exhibited the best performance, with a balanced accuracy of 87.0%, sensitivity of 84.3%, and specificity of 89.1%. This study introduced a web application for the rapid screening of lymphedema, enhancing accessibility for medical practitioners and highlighting the potential of integrating diverse data types into predictive models. Trinh et al. (2023)'s inclusion of diverse data types aligns with our approach. However, our use of the RUSBoost algorithm provided a novel method to address class imbalance, enhancing model accuracy and reliability. Additionally, we developed a Shiny-based web application with advanced features like batch data processing, visualization tools, and result downloads, making it more practical for clinical use.

2.3.5 Limitations and Future Directions

Bell et al. (2022) emphasized the limitations of relying on patient-reported data for lymphedema prediction due to potential errors and inaccuracies. The study highlighted the need for more reliable data sources, such as clinical tests and therapy records, to improve prediction accuracy and early detection. This finding is critical as it suggests a shift towards integrating more objective data sources in predictive models. We addressed the limitations identified by Bell et al. (2022) by incorporating objective clinical data, such as blood tests and therapy information, rather than relying solely on patient-reported data. This integration helped improve the accuracy and reliability of our predictive model.

Chang et al. (2016) developed a scoring system using logistic regression to predict arm lymphedema risk in breast cancer patients. Their model considered factors such as axillary lymph node dissection, history of hypertension, surgery on the dominant arm, radiotherapy, and early postoperative complications. The model achieved a sensitivity of 81.20%, specificity of 80.90%, and an AUC of 0.877, demonstrating the potential of logistic regression in prediction. This study highlights the utility of traditional statistical methods and their relevance in the era of machine learning. While Chang et al. (2016) utilized logistic regression, our project incorporated more advanced machine learning techniques, such as decision trees, ANN for the baseline performance development (will be discussed more later in this report), and ultimately RUSBoost. This allowed us to capture complex patterns in the data and improve prediction accuracy beyond what traditional methods could achieve.

2.3.6 Synthesis and Critical Analysis

These studies highlight the evolution of lymphedema prediction models from symptom-based data to more comprehensive datasets, incorporating clinical and therapy-related information. The advancements in machine learning algorithms, particularly ensemble methods like random forest and boosting techniques, have significantly improved the accuracy and reliability of predictive models. In conclusion, the reviewed literature underscores the importance of integrating diverse clinical data and employing advanced machine learning techniques to improve lymphedema prediction. Our project not only builds on these insights but also addresses the practical challenges of implementation, paving the way for more effective and accessible lymphedema management tools.

2.4 What We Have Done Differently from Previous Studies/Research Papers

Our project builds upon these advancements by developing an improved machine learning model for predicting lymphedema using the RUSBoost algorithm, which effectively handles class imbalance issues that was not well discussed in Trinh et al. (2023). The methodology involved several key phases. However, in this part, we will discuss more about how did we change and improve certain things from the research papers that we have studied, for the project entirely:

1. **Class Imbalance Handling**

   The primary distinction lay in the approach to address class imbalance issues. We obtained a dataset from Bundang Seoul National University Hospital, consisting of patient information, blood tests, and therapy details. The dataset contained 2,137 patient records, with a significant class imbalance of 1,781 patients without lymphedema and 356 with lymphedema. We addressed this imbalance using various resampling techniques to ensure the model's robustness (Trinh et al., 2023). Our model incorporated both resampling techniques and RUSBoost prior to the training phase. This preprocess step ensured a more balanced dataset before the model training began, which then enhanced the learning process and improved overall model performance.

2. **Baseline Performance Development**

   In our project, we have performed initial experiments that were not performed in other research papers, with traditional machine learning algorithms, such as logistic regression, decision trees, and ANN, thus establishing baseline performance metrics. These experiments provided insights into the strengths and weaknesses of each algorithm, guiding further optimization. The baseline performance metrics were useful while we were experimenting solutions to the class imbalance issues.

3. **Classification Probability Threshold**

   For the classification probability threshold used for decision-making, our model employed a standard classification probability of 0.5, meaning that any instance with a predicted probability above 0.5 was classified as positive. On the other hand, Trinh et al. (2023) used a lower cut-off point of 0.25, which resulted in a more lenient classification threshold and potentially higher sensitivity at the expense of specificity. In contrast, Trinh et al. (2023) applied their solution post-training by implementing a cut-off point strategy on a trained random forest model to maximize balanced accuracy. This method adjusted the classification threshold after the model had been trained, aiming to optimize the balance between sensitivity and specificity. Hence, in our case, we optimized sensitivity instead of specificity as false positives (predicting a patient has lymphedema when it is actually not) are more acceptable than false negatives. Thus, we will have a more thorough and strict model.

4. Webpage Prototype Input Dataset

The predictive model was integrated into a Shiny-based web application on R, designed to be user-friendly and informative. In Wei et al. (2021), the webpage prototype that they have integrated the predictive model allows users to enter data individually, including shoulder, arm, chest sizes, and symptoms. However, our project's webpage prototype has improved and requires uploading a dataset with a completed template, allowing multiple row inputs for efficient access to prediction results for numerous lymphedema patients. Results are now presented in a streamlined table format rather than individually with explanations and advice.

5. Webpage Prototype Results and Information

The webpage prototype that was developed by Wei et al. (2021) was somehow simple and did not allow users to visualize the results with further insights, or even save the results into users' local devices. However, our project's webpage prototype has included supplementary features like visualization tools, result downloads, chart generation based on selected features, and model descriptions. These enhancements enrich the user experience by providing insights through graphical representations, allowing users to save and further analyze results, and offering transparency about the prediction system.

These methodological differences highlighted the distinct strategies for handling class imbalances, classification criteria, and also improving the overall project's webpage prototype quality. Our approach aimed to enhance model robustness and predictive performance across a balanced dataset. The research paper's approach sought to fine-tune the classification outcomes to achieve optimal balanced accuracy. The project successfully delivered an open-access web application that facilitates the quick screening of lymphedema, catering to both medical professionals and patients. The application utilizes regular blood test results and therapy information to predict the risk of lymphedema, providing a practical and versatile tool for early detection and ongoing monitoring.

In short, the literature supports the transition towards using comprehensive clinical data in predictive models for lymphedema. Our project builds on this foundation, employing advanced machine learning techniques and integrating the model into an accessible web application to improve early detection and management of lymphedema.

3 Project Outcomes

3.1 What Has Been Implemented

The project successfully developed an open-access web application designed for the quick screening of lymphedema, catering to both medical professionals and patients. This application utilizes regular blood test results and information from therapies such as radiotherapy and chemotherapy to predict the risk of lymphedema. The core of the predictive capability is a model developed using the RUSBoost algorithm, implemented using the R programming language. This algorithm employs random forest models, iteratively trained as weak learners over 30 iterations, forming a robust classifier through a weighted majority voting scheme. The dataset used for training the model, obtained from Trinh et al. (2023), contained patient information, blood test results, and therapy data from Bundang Seoul National University Hospital. It consisted of 33 columns and included a total of 2137 patients, out of which 1781 were diagnosed with lymphedema and 356 were not. However, the dataset exhibits class imbalance, with a ratio of approximately 4:1, where patients with lymphedema significantly outnumbered those without. To address this issue, techniques such as undersampling, oversampling, and ensemble learning were employed. The user-friendly interface, created using the Shiny R package, ensures that the application is easy to use and informative. It provides comprehensive information about lymphedema, offers clear instructions on how to use the web page, includes visualizations of the predicted results, information about the predictive model, etc. Users can input datasets directly into the application to obtain predictions and have the option to download the results in Excel format or view them in detail on the web application, making the tool both practical and versatile.

3.2 Results Achieved/Product Delivered

3.2.1 Algorithm Performance

According to Table 1, we can see that the RUSBoost-based predictive model demonstrates impressive performance across various metrics, indicating its effectiveness in predicting lymphedema risk. With an accuracy of 89.6%, balanced accuracy of 85.1%, AUC of 93.6%, sensitivity of 78.2%, precision of 66.7%, specificity of 92%, and an F1 score of 72%, the model highlights strong predictive capabilities, effectively differentiating between patients with and without lymphedema. Moreover, integration testing has confirmed the successful integration of the model into the web application, ensuring that users can reliably use the platform to predict lymphedema risk. This seamless integration enhances the accessibility and usability of the application, providing medical professionals and patients with a reliable tool for early detection of lymphedema.

Table of Summary for Machine Learning Models (Oversampling + Undersampling + RUSBoost)

Model	Accuracy	Balanced Accuracy	AUC	Sensitivity	Precision	Specificity	F1
RUSBoost + over + under	0.896488	0.8512	0.9357994	0.7826	0.6667	0.9198	0.72

Table 1: Final model performance

3.2.2 Functional Prototype

The fully functional web application is hosted on a server and can be accessed at https://mds02.shinyapps.io/web_prototype/_w_14772920/#tab-9168-3. The application features three main tabs:

1. **Home** (Figure 4): This tab provides information about lymphedema for users who may not be familiar with the condition. It includes educational material and step-by-step instructions on how to use the website to predict lymphedema.
2. **Prediction** (Figure 5): This tab allows users to input a dataset containing all the patient information, enabling the model to forecast the likelihood of lymphedema in patients. Once the data is entered, the application uses the trained model to predict the risk of lymphedema and displays the results. Users can then download the prediction results in Excel format or view the results within the application.
3. **About Model** (Figure 6): This tab offers detailed information about the predictive model, including its development, the algorithms used, and the performance of it. This transparency helps users understand the model's workings and its reliability.

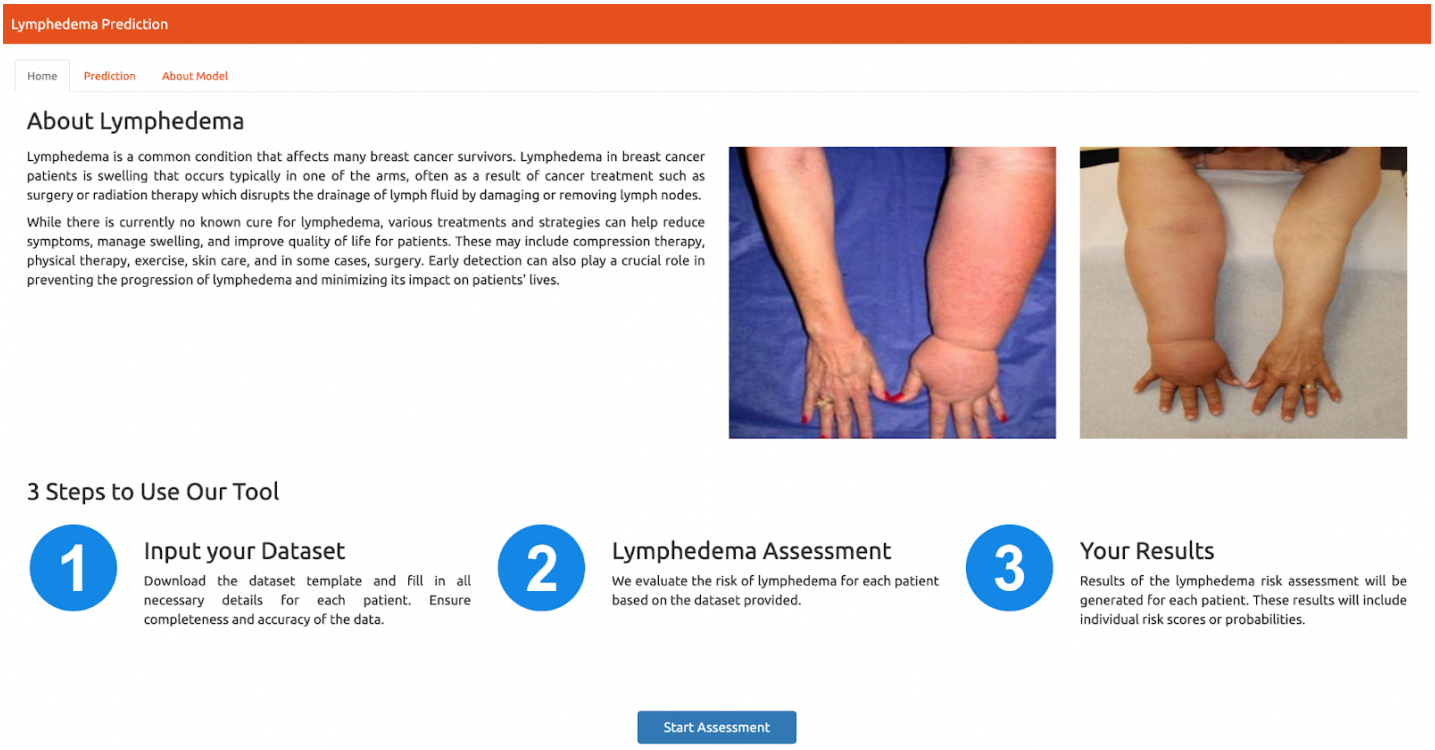

Figure 4: Website Home Tab

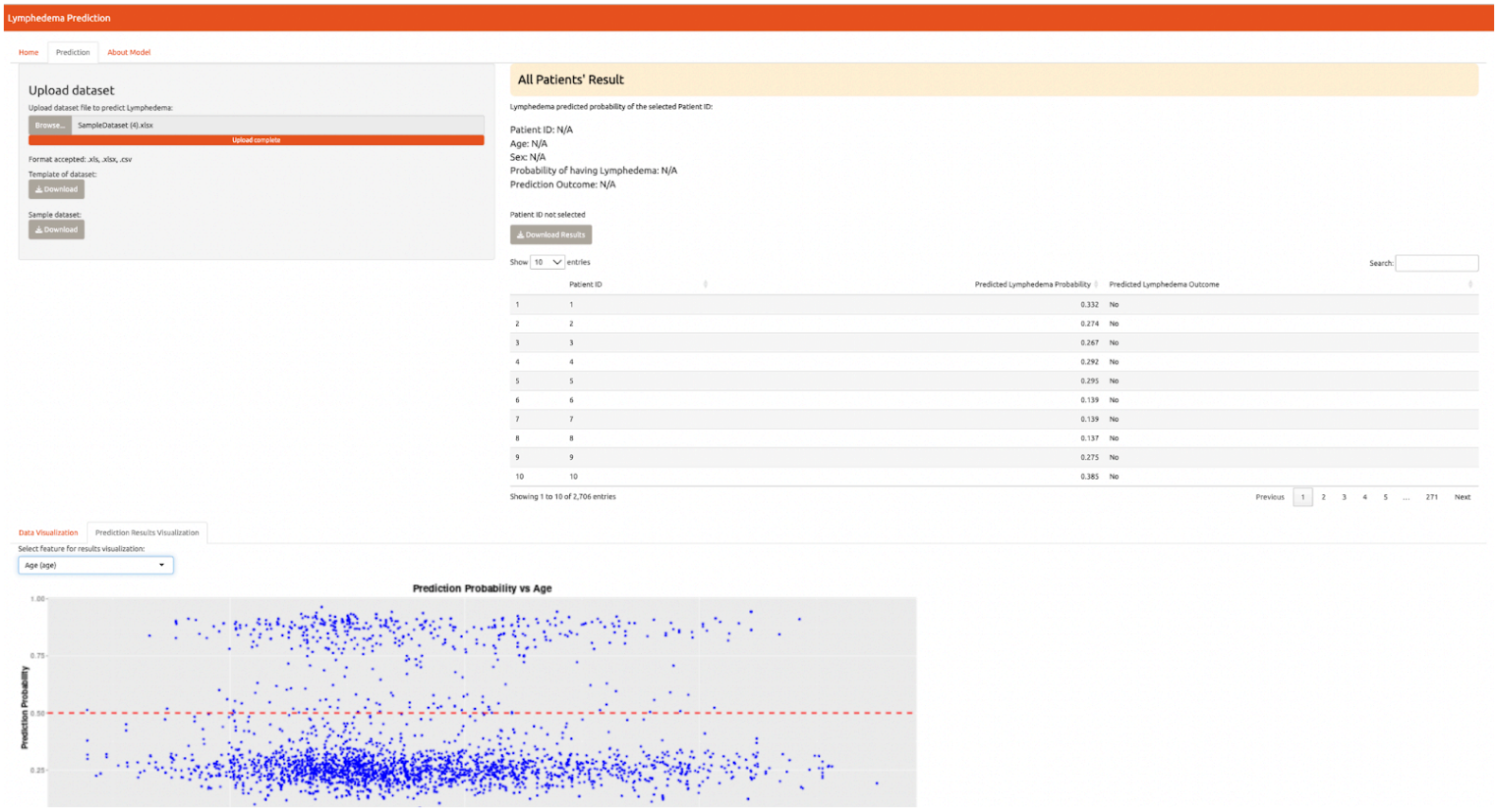

Figure 5: Website Prediction Tab

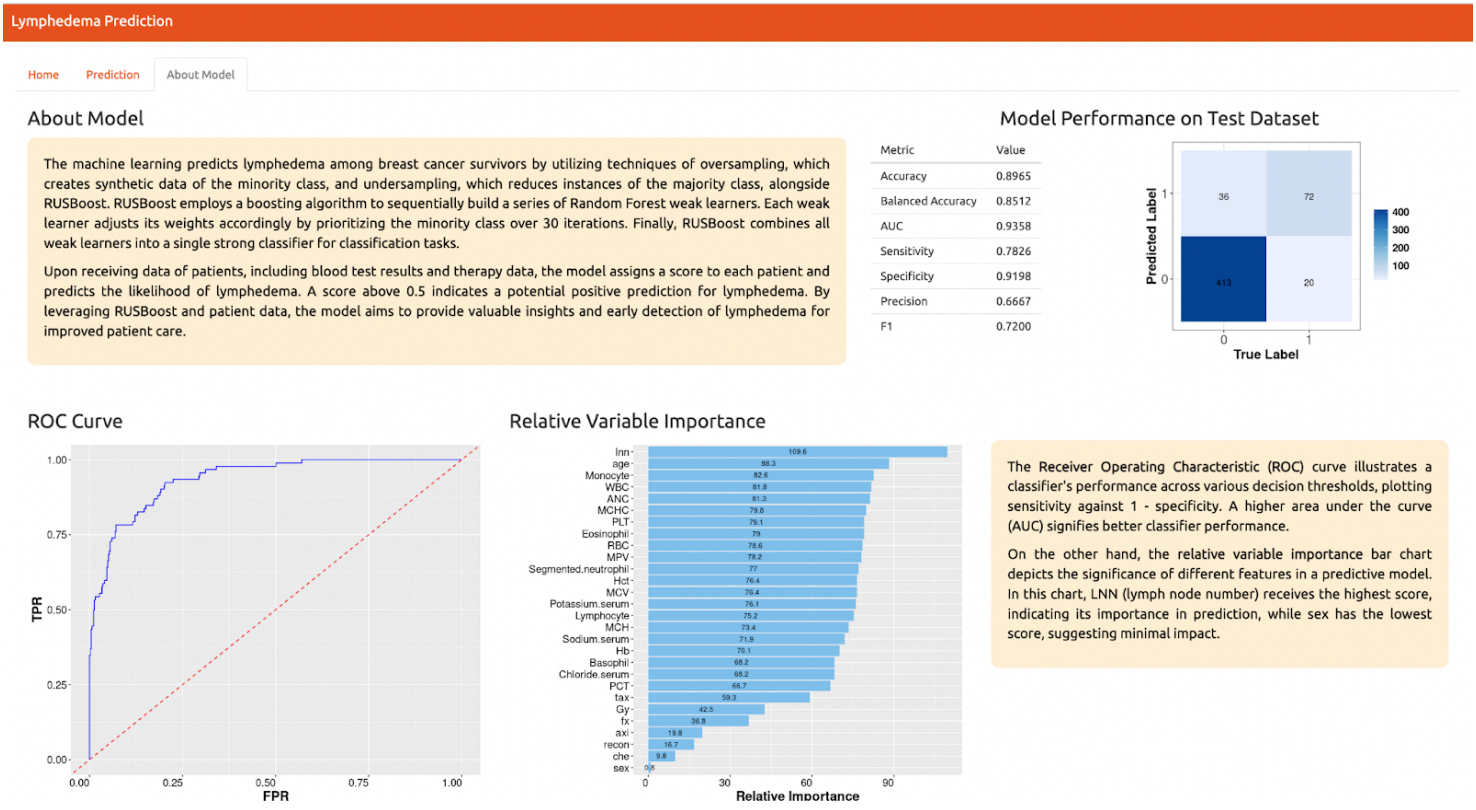

Figure 6: Website About Model Tab

3.3 Discussion of Degree of Success of Outcome Achieved

This project has been extremely successful, and a number of factors have contributed to its effectiveness. First off, creating a user-friendly interface ensures accessibility for both patients and medical professionals, making the web application simple to use and intuitive. This feature not only improves the user experience but also promotes engagement and platform usage. The utility of the webpage is increased by the inclusion of a wide range of features. From educational materials about lymphedema to step-by-step instructions on using the prediction tool, the webpage caters to a variety of user needs, offering valuable information and useful tools for managing and screening for lymphedema.

Moreover, the integration of the predictive model into the webpage contributes significantly to its value. Outstanding performance metrics, such as high sensitivity, specificity, and accuracy, shows how reliable and accurate the model is at predicting the risk of lymphedema. This feature instills confidence in users, enabling them to make informed decisions about their health and well-being based on the model's outputs.

Overall, the degree of success achieved in this project is notable, with the friendly user interface, comprehensive webpage content, and reliable predictive model results collectively contributing to its effectiveness.

3.4 User Requirements Satisfaction

The project's requirements were met through:

1. **Relevant Literature and Reliable Data Sources**

   The first objective of this project was to investigate relevant literature and data sources related to lymphedema prediction. Extensive literature review was conducted to understand the current state of lymphedema prediction, particularly among breast cancer survivors. The research paper by Trinh et al. (2023) provided a solid foundation, detailing the development of a predictive model for lymphedema among breast cancer survivors. This paper offered valuable insights into the methodologies and algorithms used in lymphedema prediction, serving as a critical reference point for our project.

   To train our model, we utilized a dataset from Bundang Seoul National University Hospital, as referenced in Trinh et al. (2023). This dataset included comprehensive patient information, blood tests, and therapy data, encompassing 33 columns and a total of 2137 patients. Out of these, 1781 patients were diagnosed with lymphedema and 356 were not, providing a robust basis for developing a predictive model. The dataset's reliability and richness in relevant features made it suitable for our machine learning purposes.

2. **Reliable Machine Learning Model**

   Our goal was to propose and develop an improved machine learning algorithm to predict lymphedema among breast cancer survivors. We employed the RUSBoost algorithm, a technique that combines random under-sampling of the majority class with boosting, to address the class imbalance in our dataset.

The overall performance of our model is commendable, surpassing the benchmark metrics established by the research paper in most areas. This is especially important given the medical context of predicting lymphedema, where high accuracy and reliability are critical for early intervention and effective management. Accurate predictions can significantly improve patient outcomes by enabling timely and appropriate treatment, thus reducing the risk of complications associated with lymphedema.

3. **Intuitive User Interface**

   We successfully developed an intuitive user interface using the Shiny R package, ensuring that the web application is easy to use, attractive, and fully functional. The usability of the interface is a key feature, allowing users to seamlessly input data, predict lymphedema risk, and visualize the results. Additionally, the web page includes several other features to enhance user experience such as offering information about lymphedema for users who may not be familiar with the condition, providing details about our predictive model and its development, including visualizations to provide more insights and identity trends, etc. These features collectively make the web application a comprehensive tool for both medical professionals and patients, enhancing its practical utility.

3.5 Justification of Decisions Made

The entire software was developed in R because we are more familiar with the R language, which made the development of the machine learning model easier and more efficient. R is particularly advantageous for machine learning due to its extensive libraries and packages specifically designed for data analysis and statistical modeling, such as caret, randomForest, e1071, ebmc, and more. Additionally, R has robust visualization packages like ggplot2 which is essential for creating insightful data visualizations. These resources made it easier for our team to implement, test, and refine the predictive model, ensuring high performance and accuracy.

The RUSBoost algorithm was selected for its ability to effectively handle the significant class imbalance in our dataset. Traditional machine learning models often struggle with imbalanced datasets, while RUSBoost addresses this issue by combining random under-sampling of the majority class with boosting techniques, enhancing the model's sensitivity and overall performance. By iteratively training weak learners (random forest models) and integrating their predictions through a weighted majority voting scheme, RUSBoost improves the model's accuracy, making it particularly suitable for medical predictions where early and reliable detection is critical.

To train our machine learning model, the decision to use regular blood tests and therapy data instead of relying solely on symptoms was driven by the limitations of symptom-based models in early detection. Symptoms-based features, such as swelling and pain in the arm or breast are often absent in the early stages of lymphedema. Patients' blood tests and therapy data can be acquired prior to the onset of visible lymphedema symptoms, providing earlier and potentially more accurate predictions, which would significantly benefit patients.

For the web application, we chose the Shiny R package based on its compatibility with our R-based model, facilitating seamless integration and rapid development. Shiny's capabilities allowed us to create an interactive, user-friendly interface that supports data input, prediction, and visualization, making the application accessible and informative for both medical professionals and patients.

3.6 Discussion of All Results

3.6.1 Validation of Accuracy and Reliability in Our Lymphedema Prediction Model

To verify the accuracy and reliability of our model, we compared its performance with a scholar research paper where the authors developed a model that serves a similar purpose using the same dataset from Bundang Seoul National University Hospital, as obtained by Trinh et al. (2023). We utilized the same metrics used in the paper as benchmarks for comparison. According to Table 2, our results exceeded the benchmarks in most metrics, including accuracy, AUC, precision, specificity, and F1 score. However, there were slight differences in the balanced accuracy and sensitivity, where our model exhibited slightly lower performance. Possible reasons for this discrepancy could be caused by differences in algorithms, data preprocessing methods, feature engineering techniques, or model hyperparameter tuning between our implementation and the one described in the research paper. These variations may have impacted the model's ability to achieve optimal balance between sensitivity and specificity. It is important to acknowledge the limitations of our model, as it highlights room for improvement. Future work could focus on improving the model's sensitivity while maintaining high overall performance, possibly by looking into feature selection or model optimisation strategies in more detail.

Table of Comparison between Our Model and Research Paper's Model

Model	Accuracy	Balanced Accuracy	AUC	Sensitivity	Precision	Specificity	F1
RUSBoost + over + under (Our Model)	0.896488	0.8512	0.9357994	0.7826	0.6667	0.9198	0.72
Random forest (Research Paper Model)	0.8558226	0.8785707	0.9337538	0.9130435	0.5454545	0.8440980	0.6829268

Table 2: Comparison of performance between our final model and research paper's model

3.6.2 Model Comparison: Evaluating Performance Across Various Algorithm

When selecting the best algorithm for our model, we conducted a thorough comparison of the metrics obtained from several experimented algorithms. According to Table 3, our analysis showed that RUSBoost appeared as the top-performing algorithm, as it has the best overall performance despite exhibiting the lowest accuracy among others. However, we observed a significant drawback in terms of precision, which was notably low. This could be due to the imbalanced nature of our dataset, where positive cases, such as patients with lymphedema, outnumbered negative cases. In such scenarios, models tend to excessively predict the positive class, leading to a higher frequency of false positive predictions and consequently a decrease in precision. To address this issue, we incorporated resampling

techniques such as oversampling and undersampling into our final model, aiming to rectify the class imbalance issue and enhance precision. From Table 2, looking at our final model's performance, it is proven that by incorporating oversampling and undersampling, the precision has improved.

Table of Comparison between Experimented Models

Model	Accuracy	Balanced Accuracy	AUC	Sensitivity	Precision	Specificity	F1
Logistic regression	0.8151571	0.54727	0.7816888	0.14130	0.38235	0.95323	0.2063492
Decision tree	0.8428835	0.58990	0.6722185	0.20652	0.61290	0.97327	0.3089431
C5.0(DT)	0.818854	0.64888	0.7463445	0.39130	0.46154	0.90646	0.4235294
ANN	0.8336414	0.60594	0.7531955	0.26087	0.52174	0.95100	0.3478261
RUSBoost	0.8003697	0.8452	0.9155854	0.9130	0.4565	0.7773	0.6086957

Table 3: Comparison of performance between Models

3.7 Limitations of Project Outcomes

1. **Lack of Flexibility in User Input Datasets**

   Users are constrained to adhere strictly to the provided dataset template and guidelines, limiting their ability to upload datasets based on their preferences or specific data collection practices. This necessity arises from the compatibility requirements essential for the seamless integration of user-provided datasets with the predictive model. Deviating from the prescribed dataset format risks incompatibility issues, potentially resulting in errors or inaccuracies in the model's predictions.

2. **Difficulty in Interpreting Visualizations**

   Users with limited knowledge of data visualization techniques may struggle to fully interpret the visualizations provided by the application. For instance, the visualization in Figure 7 may be complicated and overwhelming due to the presence of numerous data points, potentially resulting in a cluttered and messy appearance. As a consequence, users may encounter obstacles in comprehending the significance and implications of the predicted results, hindering their ability to make informed decisions based on the provided information.

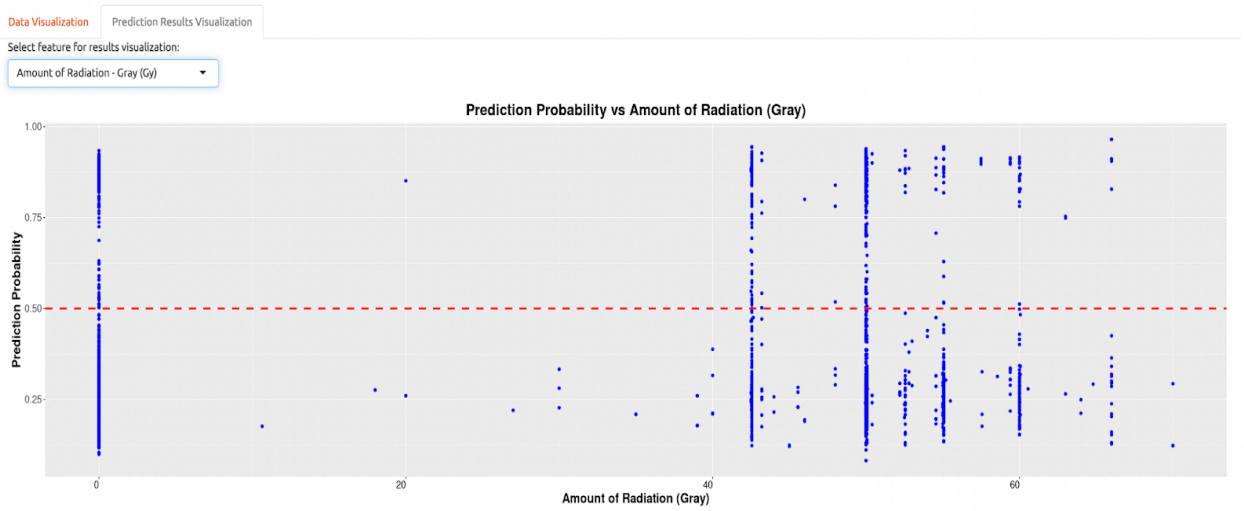

Figure 7: Prediction Results Visualization

3. **Performance Degradation**

   The application may experience performance degradation under high loads, such as a large number of concurrent users or requests, especially since it is hosted on a free hosting platform with limited resources. Users may experience delays or difficulties in accessing the application during peak usage periods, impacting their overall experience and satisfaction.

4. **Limited Prediction Scope**

   The model currently predicts only whether a patient has or does not have lymphedema, without providing insights into the severity or stages of the condition. This limitation may restrict the utility of the application for healthcare professionals who require more detailed predictions for patient management and treatment planning.

5. **Lack of Transparency in Result Accuracy**

   The application does not explicitly display the degree of accuracy or confidence level associated with the predicted results. This lack of transparency may lead users to question the reliability of the predictions and reduce trust in the application's predictive capabilities.

6. **Dependency on Data Quality**

   The predictive accuracy of the model heavily relies on the quality and completeness of the input data. If the dataset contains missing values, the model will encounter errors and will not be able to generate predictions. Users must ensure that the dataset is complete, requiring an additional step for them to clean the data since our model does not handle data cleaning. This could potentially reduce the usefulness of the application for certain users.

3.8 Discussion of Possible Improvements and Future Works

1. **Enhance Dataset Flexibility**

   The application could be upgraded to allow users to input data more freely without having to follow strict guidelines and fixed attributes, while ensuring compatibility with the predictive model. To achieve this, the model could focus on enriching the training dataset with additional features, allowing it to predict outcomes even with a broader range of different input attributes. Moreover, implementing robust data preprocessing techniques that would automatically clean and format the dataset inputted by the users, reducing the burden on users and minimizing the risk of errors that could impact the predictive accuracy.

2. **Simplify Visualizations**

   The current complexity of visualizations can make it difficult for users, especially those with limited knowledge of data visualization techniques, to derive meaningful insights from the data. Improving the clarity and simplicity of these visualizations, along with adding tooltips and explanations, would make the information more accessible and easier to interpret. This would allow users to gain valuable insights with greater ease and efficiency.

3. **Improve Infrastructure for Performance**

   Upgrading to scalable cloud services, implementing caching mechanisms, and optimizing the application code would ensure reliable performance even during peak usage periods. This would improve the overall user experience by providing faster and more stable access to the application.

4. **Expand Prediction Capabilities**

   Currently, the model only predicts the presence or absence of lymphedema. Expanding its capabilities to predict the stages or severity levels of lymphedema would provide more comprehensive insights for healthcare professionals, aiding in better patient management and treatment planning. This improvement would make the application more valuable and adaptable for its users.

5. **Include Confidence Metrics**

   Since this is a medical-related prediction, it is important to include confidence metrics that demonstrate the accuracy and reliability of the results. By incorporating confidence intervals alongside the predicted outcomes, users would help users understand how reliable and certain the predictions are. This transparency is essential for building trust in the application, as users, including medical professionals and patients, need assurance that they can rely on the accuracy of the predictive results for informed decision-making.

6. **Additional Enhanced Features**

   Enhancing the web page prototype with more features would improve its utility and user engagement. For example, adding a FAQ section that addresses common questions and concerns, a clinic finder that helps users locate healthcare facilities specializing in lymphedema treatment, and a discussion forum that provides a

platform for users to share experiences and connect with others. These features would make the application more comprehensive and user-friendly.

7. **Gather Feedback on UI/UX**

   Gathering feedback on the user interface (UI) and user experience (UX) is crucial for continuous improvement. Collecting and analyzing user feedback would provide valuable insights into how the application can be refined to better meet user needs. Iterative improvements based on this feedback would ensure the application remains instinctive, easy to use, and efficient.

8. **Collaborate with Healthcare Institutions**

   Exploring potential collaborations with healthcare institutions could greatly enhance the application's capabilities and reach. Partnerships could provide access to more variety of datasets, enable validation of the model, and create opportunities for clinical trials or studies. These partnerships would aid in improving the predictive model and broadening its utility, ultimately benefiting a wider range of users.

4 Methodology

4.1 Software Architecture

Our software operates as a web-based application, which provides users a suite of assessment tools for diagnosing lymphedema and visualizing data. The overall design of our software architecture is illustrated in Figure 8 below, comprising two main components: the frontend and the backend. The separation into frontend and backend components improves maintainability and responsiveness. The frontend ensures a responsive and interactive user experience, allowing users to input data and interact with tools. The backend handles data processing, analysis, and storage. It communicates with the frontend by processing requests and returning results.

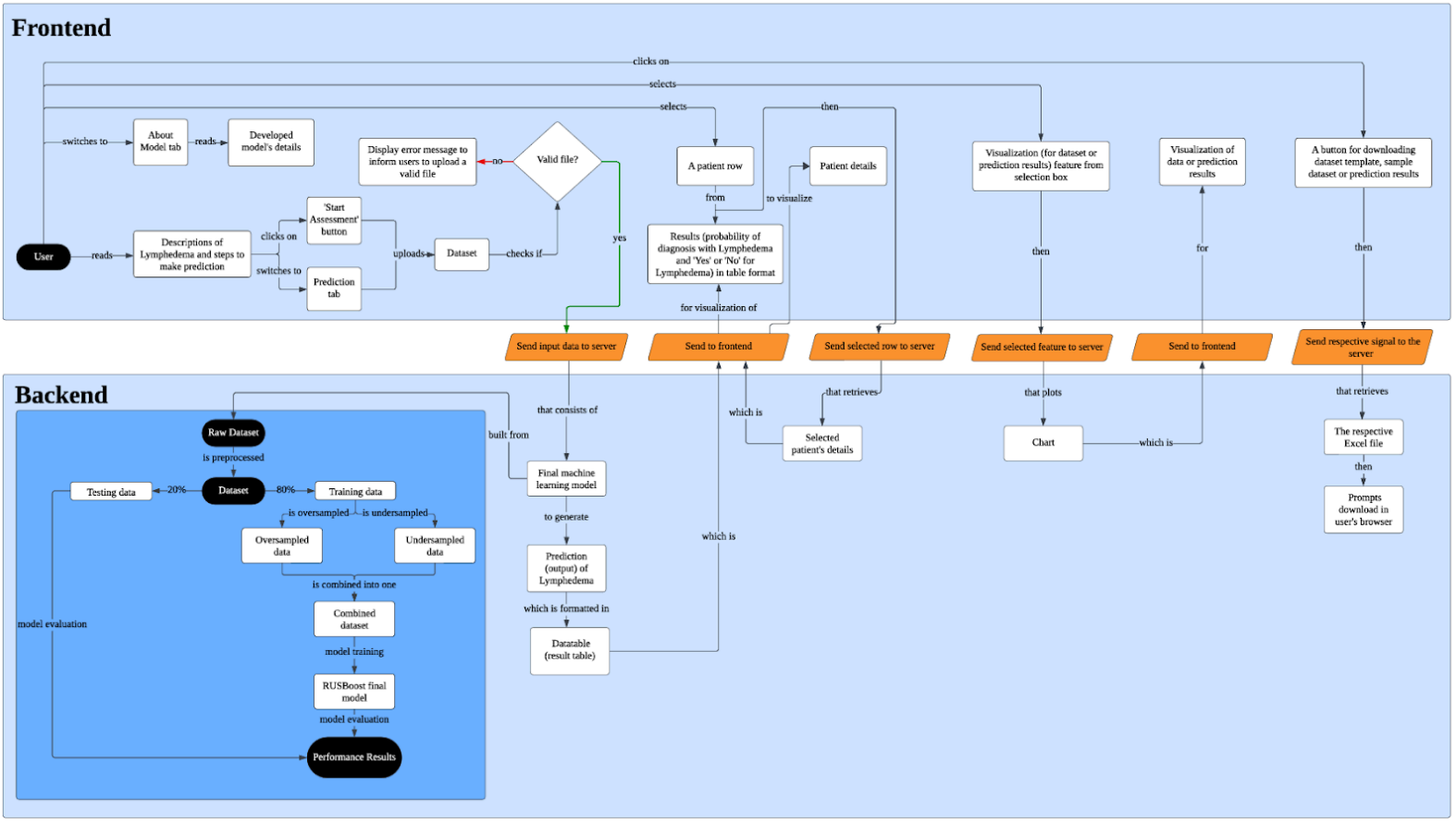

Figure 8: Overall software architecture

4.1.1 Software Frontend

The software frontend acts as the user interface, allowing users to interact with the application. Table 4 outlines the step-by-step workflow of the software frontend based on Figure 8, detailing how users interact with the lymphedema prediction tool. It covers the process from visiting the website and starting the assessment to uploading data, viewing predictions, retrieving detailed patient information, downloading results, and visualizing data and prediction outcomes.

Step	Description
1	Users visit the website to read the description of lymphedema and the steps for using the prediction tool.
2	Users have the option to either click on the “Start Assessment” button or switch to the Prediction tab.
3	Users download our dataset template and fill it with their data.
4	Users upload the dataset for prediction.
5	The input dataset will be sent to the backend. If invalid file is uploaded or the dataset contains invalid data format, error messages will be displayed. Otherwise, if the dataset is valid, the prediction process will be implemented. The prediction results will then be automatically generated and displayed in a table.
6	Users can select a specific row from the result table, which triggers a request to the backend to retrieve the corresponding patient details. Then, detailed information about the selected patient will be displayed.
7	Users click on the “Download Results” button. Then, this action signals the backend to retrieve the respective Excel file and triggers the download in the users’ web browsers.
8	In the Data Visualization or Prediction Results Visualization tab, users can select a feature from the selection box to visualize either the dataset or the results. The selected feature is then sent to the backend to retrieve the respective chart, which is subsequently displayed in the frontend.
9	Users can navigate to the About Model tab from the navigation bar to learn more about our model.

Table 4: Frontend workflow

4.1.2 Software Backend

Table 5 outlines the operational sequence of the software backend based on Figure 8, which serves as the computational core for predicting lymphedema. It details how the backend processes datasets received from the frontend, runs the prediction model, formats and returns results, retrieves specific patient details, generates downloadable files, and creates charts for visualization, ensuring accurate and reliable outputs for users.

Step	Description
1	The backend obtains a dataset from the frontend.
2	The received dataset is used in the prediction process.
3	The machine learning model generates the prediction probabilities and outcomes (“yes” or “no”), which are formatted into a data table.
4	The result table is sent back to the frontend for display of results.
5	Upon receiving the selected row from the result table, the backend retrieves the corresponding patient details from the uploaded dataset and sends them to the frontend.
6	Upon receiving a signal to download the results, the backend writes the table that contains the uploaded dataset and the prediction results to an Excel file and sends the file to the frontend.
7	Upon receiving the selected feature for visualization of the uploaded dataset or prediction results, the backend plots the respective chart and sends it to the frontend.

Table 5: Backend workflow

4.1.3 Deviations from Initial Software Design

By comparing the initial and updated software designs shown in Figure 9 in the appendix and Figure 8 above, significant modifications have been implemented. Originally, users entered data individually, including age, gender, clinical questions, and symptoms. The current design has transitioned to uploading a dataset that mandates the completion of all data features in a given template. Unlike the initial version, which only allowed single data inputs, the current design accommodates multiple row inputs. This transition aims to provide clinical usage, which enables medical professionals to efficiently access prediction results for numerous lymphedema patients. Furthermore, the presentation of results has shifted from a single result with explanations and advice to a streamlined table format.

Furthermore, the current design incorporates supplementary features such as visualization tools, result downloads, and the model's description. These new features enrich the user experience by providing additional functionalities beyond basic prediction. The visualization tools enable users to gain insights from the uploaded dataset or prediction results through graphical representations, enhancing their understanding of the underlying patterns and trends. Result downloads allow users to save and further analyze the prediction outcomes for their records or additional processing. Furthermore, the inclusion of the model descriptions offers transparency and educates users about the underlying workings of the prediction system, which encourages trust and confidence in the software.

In the backend, the current design incorporates the added functionality of data visualization, which enables the generation of charts based on user-selected features for analysis purposes. Regarding the machine learning model, there has been a transition from utilizing stacking ensemble learning, which combines various models like artificial neural network (ANN), decision tree, and logistic regression through stacking, to employing resampling and RUSBoost techniques in constructing the final model, which ultimately demonstrated better performance.

4.2 Software Specification, Libraries, and Tools Utilization

The software specification, libraries, and tools utilized are detailed in Tables 6, 7, and 8, respectively.

Initially, the proposed programming languages consisted of Python, R, JavaScript, HTML, and CSS. However, the final implementation adopted R as the primary programming language. Notably, there was a transition in both the frontend and backend development from the initially proposed React and Django frameworks to utilizing the Shiny package in R for both components. This was due to Shiny being an open-source R package that offers a sophisticated web framework for creating web applications using R. It empowers users to transform their analyses into interactive web applications, eliminating the need for expertise in HTML, CSS, or JavaScript (Posit, n.d.). Additionally, since R was the primary programming language for our model development, integrating the model into the server of the webpage was more straightforward.

Software Type	**Software *Option***
Operating System	Microsoft Windows 10; Apple macOS Ventura 13.0
Preprocessing Dataset	Microsoft Excel
Programming Languages	R 4.3.3
Frontend Web Framework	**R:** Shiny
Backend Web Framework	**R:** Shiny
Web Application Category	Single Page Application (SPA) with multiple tabs
Integrated Development Environment (IDE)	Visual Studio Code 1.88.0; RStudio 2023.12.1
Model Evaluation and Prototyping Platform	RStudio
Software Quality Control	Manual Testing
Documents/Files Storage	Google Drive
Collaboration of Code	GitHub
Collaboration of Writing	Google Docs

Table 6: Software specification

R Library	Details
dplyr	Select variables based on the required columns.
ggplot2	Plot charts.
lattice	Required library for loading “caret”.
caret	Plot confusion matrix.
openxlsx	Write a data table to Excel file.
ROCit	Find area-under-curve.
smotefamily	Implement oversampling on training dataset.
ROSE	Implement undersampling on training dataset.
ebmc	For RUSBoost model training.
shiny	Enable building interactive web applications with R.
shinythemes	Provide additional themes for enhancing the appearance of Shiny applications.
shinydashboard	Provide the creation of interactive dashboards within Shiny applications.
readxl	Read uploaded dataset into R.
ggpubr	Offer additional functionalities for publication-ready graphics.
ggpmisc	Provide additional statistical functionalities to enhance ggplot2 visualizations.
shinyBS	Offer additional Bootstrap components for Shiny applications.
shinyjs	Allow the window to scroll to top after switching the tab.

Table 7: External libraries of R

Tool	Details
Lucidchart	Develop the Work Breakdown Structure(WBS) and software design.
ProjectLibre	List down the WBS and develop the Gantt Chart.
Google Docs	Develop meeting minutes and take notes during meetings.
Google Drive	Store and share documents.
Google Chat	Formal communication between team members and supervisor.
Google Scholar / PubMed	Conduct academic research and find scholarly articles, theses, books, conference papers, and patents.
Trello	Develop Sprint Boards to plan, track and manage work.
Git	For version control that tracks code modifications history and allows multiple users to work on the same codebase concurrently.
Zoom	Conduct weekly meetings with the supervisor and rehearse for the presentation.
WhatsApp	Communication between team members.

Table 8: Project management, academic research and design tools

4.3 Data Collection and Preprocessing

Obtaining the dataset was challenging due to the sensitive nature of medical data, which included the privacy of individuals. After extensive research, we were able to access the dataset from a research paper published by Trinh et al. (2023). The dataset was collected from Bundang Seoul National University Hospital which included patient information, blood tests, and therapy data. It comprised 33 columns and 2137 patients, with 1781 patients diagnosed with lymphedema and 356 patients without lymphedema, which resulted in an approximate ratio of 4:1 in favor of those diagnosed with lymphedema.

During the preprocessing procedure, the dataset, initially containing 33 columns, underwent refinement by dropping 4 unused columns. These included the patient ID, date, name, and a column with an undefined designation (“int”). Additionally, data formats such as the presence of missing values were examined for each column to ensure data integrity. Following this, the dataset was split into 80% for training and 20% for testing before training the models.

4.4 Baseline Performance Development

In the initial phase of our project, we conducted experiments with several traditional machine learning algorithms that were initially proposed, such as logistic regression, decision trees, and ANN, to establish a baseline performance for predicting lymphedema. Each algorithm underwent rigorous testing to assess its predictive capabilities across a range of metrics. Overall, these findings provided valuable insights into the strengths and weaknesses of each machine learning algorithm, which guided further optimization efforts in our predictive modeling process.

4.5 Machine Learning Algorithms Experimentation to Solve Class Imbalance

4.5.1 Research Paper's Model

We reviewed the methodology presented in the referenced research paper and re-ran their model code. Their study employed several machine learning algorithms, including logistic regression, decision tree, C5.0, ANN, and random forest for lymphedema prediction. By re-evaluating these models, we aimed to validate the findings, understand the performance dynamics of each algorithm, and develop an improved machine learning model.

To address the class imbalance inherent in the dataset, the research paper employed a method to optimize the cut-off point by maximizing balanced accuracy. This method involved generating probability predictions for all instances and iteratively determining the optimal cut-off point. Balanced accuracy, as noted by Olugbenga (2023), is the arithmetic mean of sensitivity and specificity, which is useful for imbalanced data. Moreover, in machine learning, a classification cut-off point is a value used to classify the predicted class for each data point based on the probability generated by the model. The model does not assign the labels (e.g., positive or negative) directly but instead predicts the probability of each data point, which is then converted into distinct class labels (Evidently AI, n.d.). By fine-tuning this cut-off point, the models were better at distinguishing between lymphedema and non-lymphedema cases during the classification phase, thereby improving overall prediction performance. Figures 10 and 11 in the appendix show the flowchart and sample code for the implementation.

Overall, the re-evaluation of the research paper's model provided crucial insights into the strengths and limitations of each machine learning algorithm. It highlighted the importance of handling class imbalances and optimizing cut-off points to enhance predictive accuracy and fairness. These insights set a solid foundation for further refinement and optimization in our predictive modeling process.

4.5.2 Ensemble Learning

In the experiment with ensemble learning, we explored various techniques targeting class imbalance by achieving high sensitivity and overall robust performance in predicting lymphedema. Boosting, along with other ensemble techniques like bagging, has demonstrated notable effectiveness in managing imbalanced datasets. For example, AdaBoost is engineered to mitigate bias towards the majority class by prioritizing misclassified training instances, while Bagging employs bootstrap aggregating to train multiple classifiers using bootstrapped copies of the original dataset (Nanni, Fantozzi, & Lazzarini, 2015). Each ensemble method integrated multiple base learners, capitalizing on their strengths to enhance overall predictive capacity.

These ensemble learning techniques offered valuable enhancements in predictive performance for lymphedema detection. Nevertheless, during our experiments with ensemble methods, most of them showed extremely low sensitivity. This indicated that the challenge of class imbalance might not be entirely resolved because it was preferable to misclassify patients without lymphedema as having lymphedema. Therefore, there was a need to enhance sensitivity in order to address this concern effectively.

4.5.3 Resampling Methods

The assessments of oversampling or undersampling techniques with the baseline models were conducted to specifically address the class imbalance. Oversampling involves balancing the dataset by adding new samples to the minority class, achieved through either replicating existing minority samples or generating synthetic ones. Conversely, undersampling reduces the number of majority class samples, either randomly or based on statistical insights (Shelke, Deshmukh, & Shandilya, 2017). Each technique exhibited distinct performance characteristics indicative of its suitability for class imbalance.

During the experiment, oversampling yielded better accuracy but slightly lower sensitivity compared to undersampling. This could be attributed to information loss during undersampling, which randomly removed some data from the majority class. These observations underscore the importance of selecting appropriate sampling techniques to effectively address class imbalances and enhance predictive capabilities.

4.5.4 Final Model - Integration of Resampling and RUSBoost Techniques

After conducting thorough experimentation with various machine learning algorithms, we proposed a solution that not only addressed class imbalance but also significantly bolstered predictive capabilities, yielding superior performance across a spectrum of metrics. Our model integrated oversampling, undersampling, and RUSBoost techniques, as illustrated in Figure 12 in the appendix.

After splitting the dataset into training and testing datasets, the training data underwent both oversampling and undersampling individually. Oversampling augmented the minority class with synthetic data, while undersampling reduced the majority class by randomly eliminating some data points. Consequently, both datasets were almost balanced, and when combined, they yielded a more equitable overall distribution.

Subsequently, this combined dataset was integrated with the RUSBoost algorithm. Initially, RUSBoost balanced the dataset via random undersampling, followed by boosting iterations (Das, 2024). According to Vasilyeva (2018), the boosting builds an ensemble of weak learners by adjusting the weights of misclassified samples in each iteration. Initially, all training samples have equal weight. In subsequent iterations, weights are increased for misclassified samples, raising their chances of being included in the next learner's training. This process is especially effective for class imbalance problems as the minority class, often misclassified, receives progressively higher weights at each successive iteration. For our model, random forest models were iteratively trained as weak learners over 30 iterations, forming a strong classifier through the weighted majority voting scheme.

This integration of algorithms not only mitigated information loss but also enhanced model learning through duplicates and synthetic data, which effectively addressed class imbalance. Additionally, with the boosting algorithm in RUSBoost, our model iteratively focused on and corrected misclassifications, particularly benefiting minority class instances, thus enhancing overall accuracy and robustness. However, despite the presence of duplicates in the training dataset during the combination phase, the fact that performance validation was conducted on the testing dataset without duplicates might also introduce the potential for overfitting and bias, representing a limitation of our model.

4.6 Performance Metrics Evaluation

The models' performances were evaluated using a comprehensive set of metrics, which provided a detailed understanding of the model's effectiveness in various aspects. The performance metrics used in this study are detailed in Table 9 below:

Metric	Formula	Description
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Proportion of correctly predicted instances to the total instances.
Balanced Accuracy	$\frac{Sensitivity + Specificity}{2}$	Average of sensitivity and specificity. It accounts for imbalanced datasets by weighing each class equally to ensure the performance metric is not biased towards the majority class.
Sensitivity	$\frac{TP}{TP + FN}$	Proportion of actual positives that are correctly identified by the model.
Specificity	$\frac{TN}{TN + FP}$	Proportion of actual negatives that are correctly identified by the model.
Precision	$\frac{TP}{TP + FP}$	Proportion of predicted positives that are actually positive.
F1 Score	$2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity}$	Harmonic means of precision and sensitivity, providing a single metric that balances both false positives and false negatives. It is especially useful when the class distribution is imbalanced.

Note: TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

Table 9: Formula and description of the performance metrics

4.7 Webpage Prototype Development

Our webpage was meticulously crafted to offer essential functionalities such as prediction and visualization tools, supplemented with informative content about lymphedema, detailed instructions on utilizing the prediction tool, and descriptions of our model. We chose a web-based approach for accessibility, allowing users to access the tool from any device with an internet connection, which enhanced collaboration and ease of use.

Given the complexity of the datasets collected, we engineered our webpage to accommodate file uploads containing multiple rows of data, with the prerequisite that all features must be included for the prediction process. To streamline the user experience, we provided a downloadable input file template, enabling users to populate the required data effortlessly. The final machine learning model was integrated with the server of the webpage, ensuring an efficient prediction process. Additionally, we implemented the capability to download prediction results, conveniently attached to the uploaded input file. Furthermore, black box testing, integration testing, and usability testing were conducted to guarantee a flawless user experience.

5 Software Deliverables

5.1 Summary of Software Deliverables

5.1.1 What is Delivered

Our software is delivered as a fully functional website, which can be accessed online through web browsers without any additional software installation. The website includes the user interface, backend functionality, and integrated machine learning model. It is an online platform for users, especially the medical professionals to interact with predictive capabilities and visualizations. The model and website were developed using R programming language and Shiny R package, respectively. The complete source code, along with all necessary files and dependencies required for deployment, is accessible through our github repository: https://github.com/liawyihui/FIT3164_software/tree/main/web_prototype.

5.1.2 Description of Usage

As users access our webpage, they will observe a comprehensive description of lymphedema, followed by a step-by-step guide on utilizing the prediction tool as depicted in Figure 4 in Section 3.2.2 above.

To access the primary functionality of our webpage, users can switch to the Prediction tab, where they will be greeted with a message prompting them to upload a data file, as illustrated in Figure 13 in the appendix. Prior to uploading the dataset, users are advised to download the provided dataset template, depicted in Figure 14 in the appendix, and populate it with the required data. In the event that the uploaded dataset contains an invalid data format, such as missing values, error messages will be promptly displayed to notify the user.

Upon successfully uploading the dataset by browsing the file, users will be presented with the prediction results in table format, as shown in Figure 5 in Section 3.2.2. Furthermore, to utilize the visualization tools, users can simply select a feature from the selection box for the data or prediction results visualization, as exemplified in Figure 5 in Section 3.2.2. By switching to the About Model tab, users can read more about our model and its performance, as shown in Figure 6 in Section 3.2.2.

The summary and detailed descriptions of the features in the software are listed in Table 10 in the appendix.

5.2 Summary and Discussion of Software Qualities

5.2.1 Robustness

Our software prioritizes robustness by implementing comprehensive error-handling mechanisms. This includes validating user input to ensure data integrity and consistency. Additionally, we manage server errors effectively to minimize disruptions to the user experience. Furthermore, rigorous manual testing procedures are employed to identify and rectify any potential issues, ensuring a seamless user experience.

However, it is important to note that while we strive to address various error scenarios, unexpected errors or edge cases may still occur. These unforeseen errors could potentially impact the user experience and require prompt investigation and resolution to maintain the integrity and reliability of the software.

5.2.2 Security

Security measures are paramount in our software. To safeguard user data, we refrain from storing any uploaded data on the server. Instead, uploaded data is deleted at the end of each session, ensuring user privacy and data security. By adopting this approach, we mitigate the risk of unauthorized access to sensitive information.

5.2.3 Usability

Usability is a key focus of our software design. We achieve this by providing users with step-by-step guidance throughout the application. Our interface is designed to be intuitive and user-friendly, enabling users to navigate through the prediction process effortlessly. By prioritizing usability, we ensure that users can obtain prediction results efficiently.

However, a potential limitation is that users may encounter difficulties if they fail to provide a complete dataset for prediction, as our model requires all specified data features to be included in the input file. In such cases, the system will return an error message, which may lead to frustration and discourage users from further utilizing the software. This limitation highlights the importance of clear guidance for users to ensure successful interactions with the application. Moreover, users lacking knowledge of data visualization may face difficulties in interpreting the visualizations, posing a usability challenge for some users.

5.2.4 Scalability

Scalability is addressed through optimization techniques implemented in our codebase. One such technique involves avoiding duplications within the code, which enhances performance and reduces resource consumption. By optimizing our code, we ensure that our software can efficiently handle increasing loads and user demands as it scales over time. Nevertheless, it may still face challenges with performance degradation, particularly when handling a large number of users or requests. This is due to the free subscription plan for website hosting.

5.2.5 Documentation and Maintainability

Documentation and maintainability are vital aspects of software development. To facilitate ease of understanding and future updates, we adhere to best practices such as using meaningful variable names and providing code comments for each code block. Additionally, we break down complex code into smaller, manageable segments, making it easier to maintain and update the software in the long run. These practices ensure that our software remains well-documented, organized, and easily maintainable throughout its lifecycle.

5.3 Sample Source Code

This section contains the sample source code demonstrating the primary functionality of our software, which is the prediction tool. The reactive output expression that handles this functionality is included in a server function, as illustrated in Figure 15 in the appendix.

Figure 16 in the appendix shows the full reactive output expression code, ***output$pred.lymphedema***, for the prediction process, which renders a DataTable with lymphedema prediction results based on user-uploaded data files. The code validates the file format, normalizes the dataset, and uses a pre-trained model to make predictions. The results, including predicted probabilities and outcomes, are then formatted and displayed in the data table.

Figure 17 in the appendix demonstrates how the reactive output expression, ***output$pred.lymphedema***, defined within the server function, is called in the UI to display the DataTable, which is generated and formatted in the server function.

6 Critical Discussion

The execution of our project, "Lymphedema Prediction Using Machine Learning Approaches", was a multifaceted endeavor that involved several key phases and significant deviations from the initial project proposal. Our project aimed to develop an improved machine learning model using the RUSBoost algorithm and integrate it into a user-friendly web application for early detection and management of lymphedema. Here, we discuss the overall success of the project, the evolution of our approach, and the factors that influenced our decisions.

6.1 Project Overall Success

Our project was largely successful in achieving its primary objectives. The initial proposal focused on developing a predictive model using standard machine learning techniques, addressing class imbalances with basic resampling methods, and creating a simple web interface for individual data input and result display. In actuality, we developed a more sophisticated predictive model using the RUSBoost algorithm, which effectively handles class imbalances better than the basic techniques we originally planned. This change improved the model's accuracy and robustness. We also integrated this model into a Shiny-based web application, and provided a comprehensive tool for early detection and monitoring of lymphedema. The final product surpassed our initial expectations in terms of functionality and user experience. The addition of features such as data visualization tools, result downloads, and detailed model descriptions enhanced the usability and practicality of the application for healthcare providers.

6.2 Project Management Topics

We adopted an Agile methodology, which allowed flexibility and adaptability. Regular sprints and stand-up meetings helped us track progress and make necessary adjustments. This approach was beneficial when addressing data preprocessing and model integration challenges. For instance, the need for more sophisticated class imbalance handling techniques led us to allocate additional time and resources to implement the RUSBoost algorithm.

Effective project management was crucial to our success. Regular meetings, held twice a week, facilitated open communication and ensured alignment with project goals. These meetings allowed us to address challenges, such as the initial insufficiency of our dataset, by deciding to acquire more comprehensive data from additional healthcare providers. Task allocation leveraged each member's strengths, with data science-focused members handling data preprocessing and model development, while web development experts managed the Shiny application integration, and quality assurance handling software testing. This strategic division ensured efficient workflow and timely milestone completion. However, integrating the machine learning model with the Shiny application presented challenges, particularly converting the Python model to R while maintaining performance. Extensive troubleshooting revealed issues with data handling, which we addressed by incorporating automated data cleaning and validation steps. We also optimized the application for security and scalability, implementing data encryption and user authentication. Collaborative problem-solving, such

as code review sessions, improved our code quality and fostered team understanding. These efforts ensured the successful deployment of a robust, user-friendly tool for lymphedema prediction. Despite technical hurdles, our effective management and teamwork enabled us to deliver a comprehensive solution.

The project was divided into several phases, each with specific deliverables and deadlines. Initially, data collection and preprocessing were planned for the first month, model development for the second month, and integration and testing for the third month. Delays in data collection extended this phase, but overlapping development and testing phases kept the project on track. The scope expanded from developing a predictive model and a basic web interface to include advanced data preprocessing, enhanced visualization tools, and additional web application features. This expansion required careful resource management to deliver the enhanced features on time.

Moreover, we implemented effective risk management strategies, including regular progress reviews and proactive identification of potential bottlenecks. The team demonstrated adaptability by adjusting task allocations and timelines based on the evolving needs of the project. Despite technical challenges, such as the integration of the machine learning model with the web application, our team's commitment to finding solutions and continuous collaboration ensured the project's success. For example, when we encountered difficulties in integrating the model with the Shiny app, we scheduled additional troubleshooting sessions and consulted external resources, ultimately overcoming the issues through collaborative effort. Besides, we maintained a strong relationship with our supervisor, Dr. Ong Huey Fang. Her guidance and feedback were instrumental in refining our approach and ensuring that the project stayed aligned with its objectives. Regular updates and open communication helped in addressing concerns promptly and incorporating valuable insights into the project. For instance, Dr. Ong's suggestion to focus on enhancing the user interface led to the addition of visualization tools and result download features, significantly improving the application's usability.

6.3 Evolution of Thinking

Our thinking evolved significantly throughout the project. Initially, we focused solely on developing a predictive model using established machine learning techniques. However, as we progressed, we realized that a more comprehensive solution was necessary. Feedback from our supervisor and practical challenges highlighted the importance of advanced data handling, robust performance metrics, and a user-friendly interface. For instance, early testing revealed that handling class imbalances and missing data was more complex than anticipated. We had to refine our data preprocessing steps, incorporating techniques such as resampling and automated data validation to ensure high-quality input. Additionally, feedback emphasized the importance of making the tool scalable and easy to use in clinical settings. This led us to enhance the web application's features, including batch data processing, visualization tools, and result download options. Overall, this shift in focus allowed us to create a more robust, scalable, and user-friendly tool, addressing both technical and practical challenges and ultimately improving the project's impact and usability in real-world clinical settings.

6.4 Deviation from Initial Project Proposal

While our initial project proposal laid a solid foundation, we made significant deviations to enhance the project's outcome. Initially, we planned to use basic resampling techniques for class imbalance, however in our final software outcome, we have adopted RUSBoost for better performance and dataset balance. The reason for this is because basic resampling techniques, like oversampling and undersampling, often lead to overfitting or loss of valuable information, respectively. Oversampling can create redundant instances of the minority class, leading to overfitting, while undersampling can discard useful data from the majority class. RUSBoost, on the other hand, combines random undersampling with boosting, a powerful ensemble technique that improves model performance by focusing on difficult-to-classify instances. This combination allows RUSBoost to handle class imbalance more effectively without the drawbacks of basic resampling techniques. We have also conducted baseline performance experiments with logistic regression, decision trees, and ANN, providing insights for optimization and addressing class imbalance. The classification probability threshold that was adjusted from 0.25 to 0.5 has improved the sensitivity and reduced false negatives, making the model more reliable. Additionally, the web prototype was enhanced to support batch data processing, visualization tools, result downloads, and detailed model descriptions, making it more practical for clinical use by efficiently handling large patient volumes.

The current design of the prototype incorporates supplementary features as compared to the initial project proposal such as enhanced visualization tools, result downloads, and model descriptions. These features enrich the user experience by providing additional functionalities beyond basic prediction. The added enhanced visualization tools enable users to generate charts based on user-selected features and gain insights from the uploaded dataset or prediction results through graphical representations, enhancing their understanding of underlying patterns and trends. Result downloads that we have also added into our final prototype allow users to save and further analyze prediction outcomes, while added model descriptions also offer transparency and educate users about the prediction system, building trust and confidence.

Overall, despite the slight deviations, the project was more successful compared to the initial project proposal. The changes we implemented significantly enhanced the model's performance and the web application's usability. These deviations were necessary and beneficial, ultimately leading to a more effective and user-friendly tool for lymphedema prediction.

7 Conclusion

In conclusion, the development of our predictive model and accompanying web application marks a significant advancement in the field of lymphedema detection and management. Through careful experimentation, integration of techniques, and adherence to best practices in software development, we have successfully delivered a robust and user-friendly tool for lymphedema prediction for both medical professionals and patients.

Our project has shown success in various aspects. A reliable user experience is ensured by the software's robustness, which is maintained by extensive error-handling mechanisms and thorough testing. Security measures, such as not storing uploaded data on the server, prioritizing user privacy and data security, and aligning with industry standards and regulations. Our software is designed with usability in mind, with intuitive navigation and step-by-step guidance facilitating efficient interaction for users. However, issues like inflexibility in user input datasets, difficulty in interpreting visualizations, or performance degradation under high loads point to possible areas for improvement in our software.

Looking forward, our main focus is to expand the predictive capabilities of our model. While the current version accurately predicts the presence or absence of lymphedema, future iterations will aim to predict the stages of lymphedema (i.e. stages I, II, III). This enhancement will provide more comprehensive insights for healthcare professionals, enabling tailored patient management and treatment planning.

In summary, our project has made a significant contribution to the early diagnosis and treatment of lymphedema. With ongoing improvements and a commitment to future development, we aim to further improve patient outcomes and quality of life in the healthcare domain.

Appendix

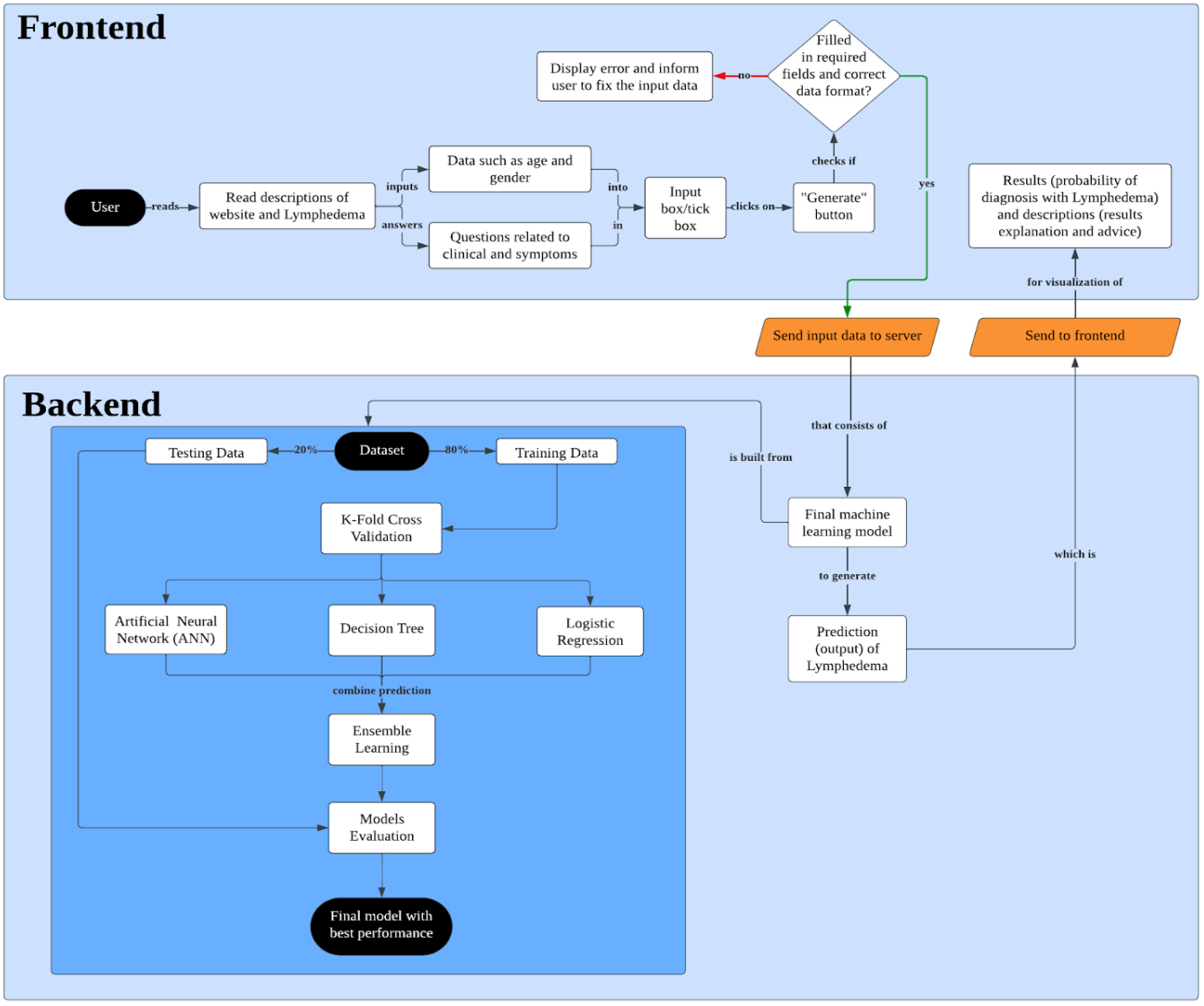

Figure 9: Initial software architecture

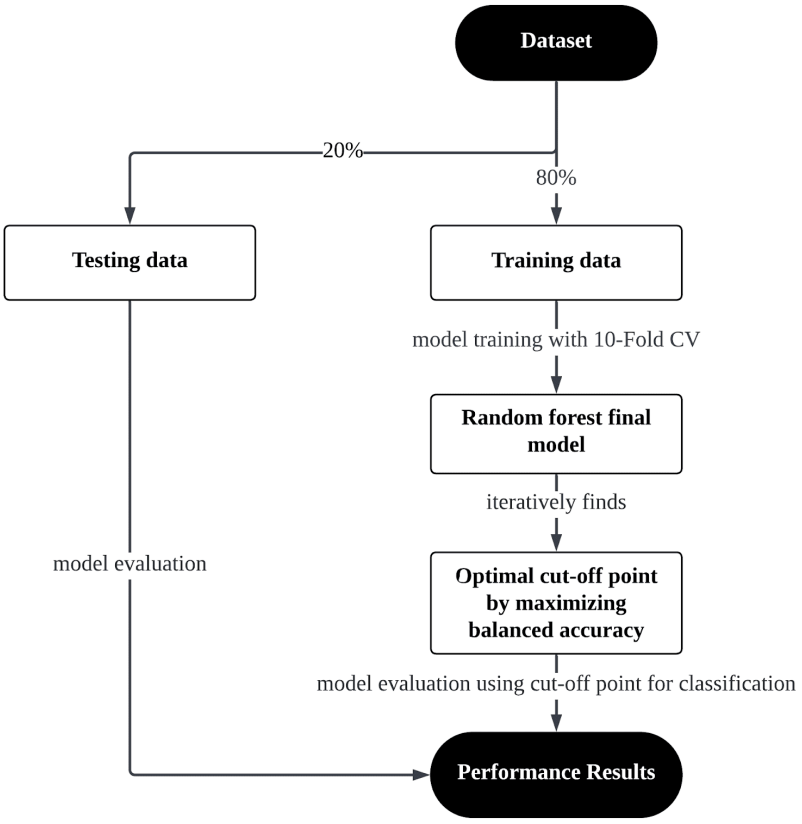

Figure 10: Research paper's model development flowchart

```r
#-------------------------------------------------------------------------------
control <- trainControl(method="repeatedcv", number=10, repeats=3)
# train the model
registerDoMC(cores=6)
LRmodel <- train(Endpoint~., data=train, method="LogitBoost", trControl=control, tuneLength=5)


#-------------------------------------------------------------------------------
# Find Cut-off value for probability to maximize balanced accuracy

# Get probability
pred_all_prob <- as.data.frame(LRmodel %>% predict(Table1, type = "prob"))

Table_cutoff <- data.frame( "Cutoff"             = seq(0.01, 1, by= 0.01),
                            "Balanced_Accuracy"  = 0)
for (i in (1:100)) {
  pred.LE <- as.factor(ifelse(pred_all_prob$`1`>Table_cutoff$Cutoff[i],"1","0"))
  Table2 <- table(factor(pred.LE, levels = c("0","1")), Table1$Endpoint)
  ConfMat <- confusionMatrix(Table2)
  Performance <- setDT(as.data.frame(ConfMat$byClass), keep.rownames = TRUE)[]
  Table_cutoff$Balanced_Accuracy[i] <- Performance[11,2]
}

cutoff <- Table_cutoff$Cutoff[which.max(Table_cutoff$Balanced_Accuracy)]

#-------------------------------------------------------------------------------
# Make prediction on test set
pred_train_prob <- as.data.frame(LRmodel %>% predict(train, type = "prob"))
pred_test_prob <- as.data.frame(LRmodel %>% predict(test, type = "prob"))
pred_all_prob <- as.data.frame(LRmodel %>% predict(Table1, type = "prob"))

pred_train <- as.factor(ifelse(pred_train_prob$`1`>cutoff,"1","0"))
pred_test <- as.factor(ifelse(pred_test_prob$`1`>cutoff,"1","0"))
pred_all <- as.factor(ifelse(pred_all_prob$`1`>cutoff,"1","0"))
```

Figure 11: Code snippet for model training, cut-off point determination, and classification in the research paper

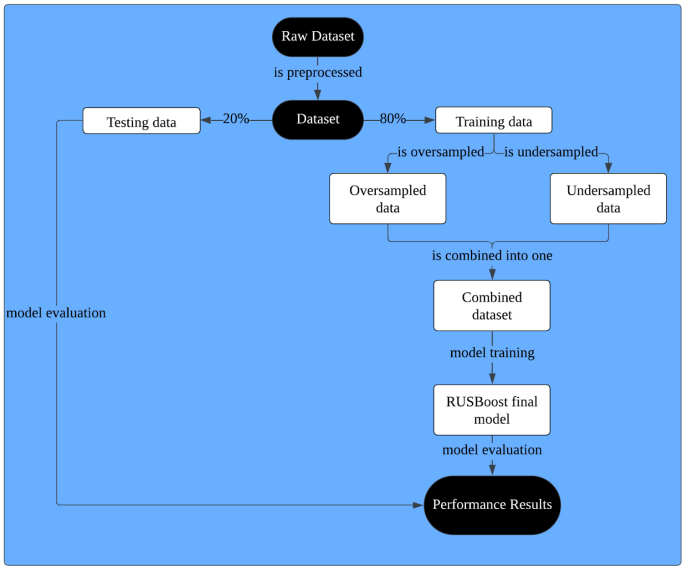

Figure 12: Final model training flowchart

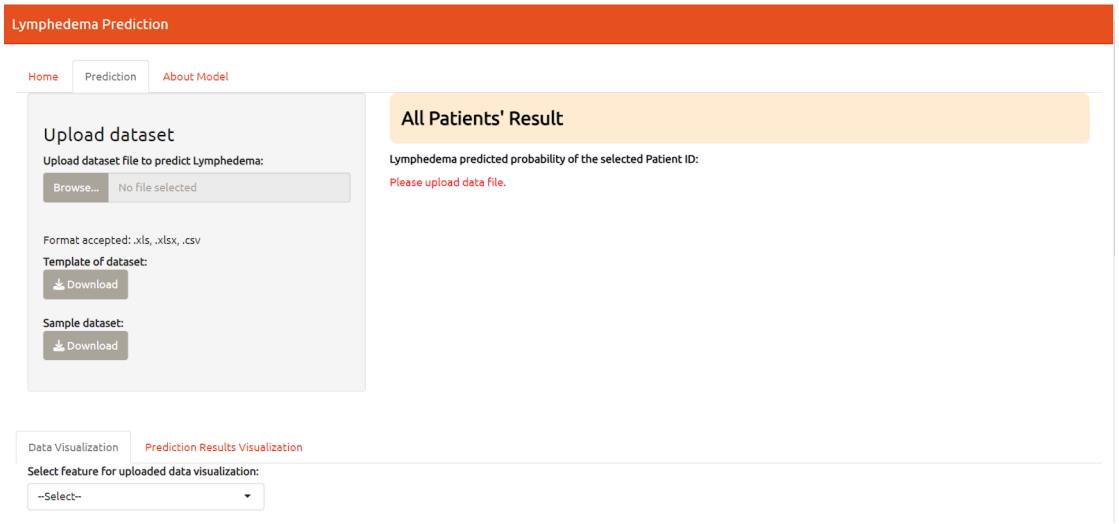

Figure 13: The view of webpage when user switches to the prediction tab

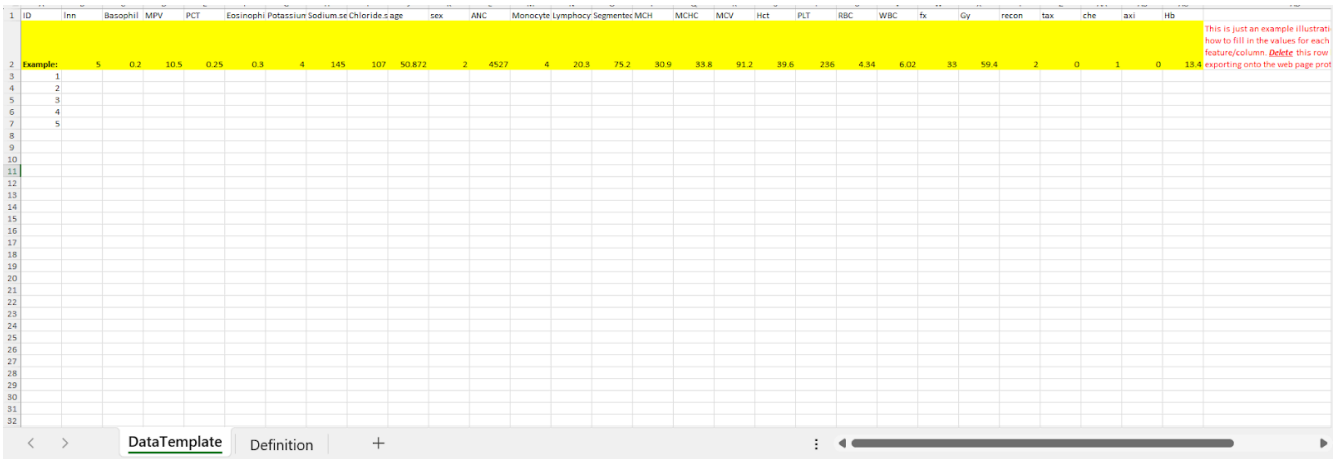

Figure 14: Example of data template

Feature	Description
Home Tab	The landing tab which is designed to provide users with introductory information about lymphedema and guidance on how to use the prediction tool.
Prediction Tab	Tab that grants access to the prediction tool functionality, allowing users to input data and receive predictions.
Dataset Upload	By clicking “Browse”, users can upload their dataset in .xls, .xlsx, or.csv format for prediction.
Download Dataset Template Button	A button allowing users to download a dataset template to ensure proper formatting for data upload.
Download Sample Dataset Button	A button allowing users to download a sample dataset for reference or testing purposes.
Prediction Results Table	A table that displays the prediction results generated by the prediction tool. The results table consists of patient ID, prediction probability, and prediction outcome. A prediction probability above 0.5 signifies a high risk of lymphedema for the patient, while a probability below this threshold indicates a lower risk.
Row Selection in Prediction Table	A feature that allows users to select a row within the prediction table for accessing detailed information about the selected patient.
Download Result Button	A button that allows users to download the prediction results for further analysis, storage, or processing.
Visualization Section	An area where users can select data features from selection boxes and visualize input data or prediction results to gain insights.
About Model Tab	Tab that provides comprehensive insights into the machine learning model used within the prediction tool, including details on model performance, background, ROC curve, and the significance of variables.

Table 10: Summary and detailed descriptions of the features in the software

```
287   server <- function(input, output) {
430
431     # Predict lymphedema for the user-input dataset
432   >   output$pred.lymphedema <- DT::renderDataTable( ···
498     )
499
```

Figure 15: Source code for integration of the reactive expression for prediction in the server function

```r
# Predict lymphedema for the user-input dataset
output$pred.lymphedema <- DT::renderDataTable(
  {
    validate(need(input$DataFile, "Please upload data file."))
    prediction_results$data <- NULL

    inFile <- input$DataFile
    file_ext <- tools::file_ext(inFile$name)

    validate(need(file_ext %in% c("xlsx", "xls", "csv"), "Error: Unsupported file format."))

    # Process excel file or csv file
    if (file_ext %in% c("xlsx", "xls")) {
      available_sheets <- excel_sheets(inFile$datapath)
      validate(need("DataTemplate" %in% available_sheets, "Error: Sheet 'DataTemplate' not found in the Excel file. Please rename the sheet."))
      DataTable <- read_excel(inFile$datapath, sheet = "DataTemplate")
    } else if (file_ext == "csv") {
      DataTable <- read.csv(inFile$datapath)
    }

    required_columns <- c("ID", "age", "sex", "lnn", "tax", "fx", "Gy", "recon", "che", "axi", "PLT", "PCT", "WBC", "ANC", "RBC", "MPV", "Eosinophil", "Basophil", "Monocyte",
    "Hct", "Segmented.neutrophil", "MCHC", "Hb", "Lymphocyte", "MCV", "MCH", "Potassium.serum", "Chloride.serum", "Sodium.serum")

    # Validation for columns
    validate(need(all(required_columns %in% colnames(DataTable)), "Error: Dataset is missing required column(s) or wrong column name(s)."))

    # Validation for data format
    validate(need(all(sapply(DataTable, function(x) !all(is.na(x)) && all(x != ""))), "Error: Dataset contains missing values."))
    validate(need(all(sapply(DataTable[, setdiff(colnames(DataTable), "ID")], function(x) all(is.numeric(x)))), "Error: Incorrect data format. Data must be numeric."))

    # Validation for categorical data
    validate(need(all(DataTable[["sex"]] %in% c(1, 2)), "Error: Data for 'sex' column must contain 1(Male) or 2(Female) only."))
    validate(need(all(DataTable[["recon"]] %in% c(0, 1, 2)), "Error: Data for 'recon' column must contain 0(No reconstruction), 1(TRAM flap) or 2(Implant) only."))
    validate(need(all(DataTable[["tax"]] %in% c(0, 1, 2)), "Error: Data for 'tax' column must contain 0(No taxane), 1(Type 1) or 2(Type 2) only."))
    validate(need(all(DataTable[["che"]] %in% c(0, 1)), "Error: Data for 'che' column must contain 0(No) or 1(Yes) only."))
    validate(need(all(DataTable[["axi"]] %in% c(0, 1)), "Error: Data for 'axi' column must contain 0(No) or 1(Yes) only."))

    # Normalize uploaded dataset
    Normalized_DataTable <- DataTable
    # Exclude the ID variable before normalizing
    independent_variables <- setdiff(required_columns, c("ID"))
    Normalized_DataTable[, independent_variables] <- scale(Normalized_DataTable[, independent_variables])

    # Make prediction
    Pred.prob <- predict(model, Normalized_DataTable, type = "prob")

    # Format and store results for displaying in the data table
    OutputTable <- data.frame(
      Patient.ID = as.factor(DataTable$ID),
      Predicted.Probability = round(Pred.prob, 3),
      Predicted.Lymphedema = ifelse(Pred.prob > 0.5, "Yes", "No")
    )

    # Format and store results for prediction results visualization
    results_plot <- data.frame(
      DataTable,
      Predicted.Probability = round(Pred.prob, 3),
      Predicted.Lymphedema = ifelse(Pred.prob > 0.5, "Yes", "No")
    )
    prediction_results$data <- results_plot

    # Format column names in the data table
    colnames(OutputTable) <- c("Patient ID", "Predicted Lymphedema Probability", "Predicted Lymphedema Outcome")

    OutputTable
  },
  selection = "single"
)
```

Figure 16: Source code for the reactive output expression of the prediction tool

```
  # Display prediction results
  mainPanel(
    verbatimTextOutput("txtout"), # txtout is generated from the server
    tableOutput("tabledata"), # Prediction results table
    p(strong("All Patients' Result"),
      style = "font-size:24px; text-align:justify; color:black; background-color:papayawhip; padding:15px; border-radius:10px"
    ),
    p(strong("Lymphedema predicted probability of the selected Patient ID:"),
      style = "text-align:left; color:black; padding:0px; border-radius:0px"
    ),
    tableOutput("pred.single"),
    uiOutput("downloadResultsButtonOutput"),
    DT::dataTableOutput("pred.lymphedema") # Call the expression from server function to retrieve prediction results
  )
), # mainPanel
```

Figure 17: Source code for calling the reactive output expression in the UI

References

Bell, L., Freeman, R., Jones, C., & Richards, S. (2022). Challenges in Predicting Lymphedema Using Patient-Reported Data. *Journal of Medical Systems,* 46, 12-23. doi:10.1007/s10916-022-01735-6

Chang, Y. J., Chen, Y. J., Wang, L., & Tsai, M. S. (2016). A scoring system to predict arm lymphedema risk for individual Chinese breast cancer patients. *Breast Care,* 11, 52-56. doi:10.1159/000444998

Das, B. (2024). RUSBoost. Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/37315-rusboost

Evidently AI (n.d.). How to use classification threshold to balance precision and recall. Retrieved from https://www.evidentlyai.com/classification-metrics/classification-threshold

Fazeli, M., Haghighat, S., & Kazemi, M. (2017). Predicting the risk of lymphedema in breast cancer patients by using data mining techniques. *Multidisciplinary Cancer Investigation, 1*(1), 0. doi:10.21859/mci-supp-83

Fazeli, M., Haghighat, S., & Kazemi, M. (2022). Predictive Models for Lymphedema in Breast Cancer Patients Using Data Mining Techniques. *Cancer Informatics,* 21, 1-14. doi:10.1177/1176935122111937

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.

Markets and Markets. (2019). Lymphedema diagnostics market by technology. Retrieved from https://www.marketsandmarkets.com/Market-Reports/lymphedema-diagnostics-market-145177203.html

Mayo Clinic (2022). Lymphedema. Retrieved from https://www.mayoclinic.org/diseases-conditions/lymphedema/symptoms-causes/syc-20374682

Nanni, L., Fantozzi, C., & Lazzarini, N. (2015). Coupling different methods for overcoming the class imbalance problem. *Neurocomputing, 158*, 48–61. https://doi.org/10.1016/j.neucom.2015.01.068

Olugbenga, M. (2023). Balanced Accuracy: When Should You Use It? Retrieved from https://neptune.ai/blog/balanced-accuracy#:~:text=Balanced%20Accuracy%20is%20used%20in,lot%20more%20than%20the%20other.

PhysioMotion. (n.d.). Lymphedema. Retrieved from https://www.physiomotion.com.hk/conditions/lymphedema

Posit (n.d.). Welcome to Shiny. Retrieved from https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html

Sheehan, M. (2023). Living with lymphedema. Retrieved from https://newsroom.osfhealthcare.org/living-with-lymphedema/

Shelke, M. S., Deshmukh, P. R., & Shandilya, V. K. (2017). A review on Imbalanced data handling using undersampling and oversampling technique. *International Journal of Recent Trends in Engineering & Research, 3*(4), 444–449. https://doi.org/10.23883/ijrter.2017.3168.0uwxm

Trinh, X., Chien, P. N., Long, N., Van Anh, L. T., Giang, N. N., Nam, S., & Myung, Y. (2023). Development of predictive models for lymphedema by using blood tests and therapy data. *Scientific Reports, 13*(1). https://doi.org/10.1038/s41598-023-46567-1

Vasilyeva, A. (2018). Using SMOTEBoost and RUSBoost to deal with class imbalance. Retrieved from https://medium.com/urbint-engineering/using-smoteboost-and-rusboost-to-deal-with-class-imbalance-c18f8bf5b805

Wei, X., Lu, Q., Jin, S., Li, F., Zhao, Q., Cui, Y., Jin, S., Cao, Y., & Fu, M. R. (2021). Developing and validating a prediction model for lymphedema detection in breast cancer survivors. *European Journal of Oncology Nursing,* 54, 102023. doi:10.1016/j.ejon.2021.102023

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.
- [sentences] paraphrase and change sentence structure.
- define agile methodology.

The outputs generated were modified and incorporated into this document.