FIT3164 Semester 1, 2024:
Progress Status Summary Report

Lymphedema Prediction using Machine Learning Approaches

Team: MDS02

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

Table of Contents

Overall Team's Progress

1 Brief recap or summary of project

Lymphedema in breast cancer patients is swelling that occurs typically in the arm or chest area following treatments like radiation, which disrupt the drainage of lymph fluid by damaging or removing lymph nodes (Mayo Clinic, 2017). Our main objective in this project is to propose and build an improved machine-learning predictive model as compared to the currently available predictive model by research, to predict lymphedema among breast cancer patients. We then integrate the predictive model into a web page prototype for usage and visualization purposes.

2 Tasks accomplished over the previous 5 weeks

During week 3, when conducting research on our project, we discovered a research paper with a new dataset which was closely aligned with our project's objectives. We analyzed the dataset, developed the base models, and assessed the models' performance. In week 4, we thoroughly reviewed the research paper and executed the code provided in the paper to compare the performance with our findings. This exploration revealed significant disparities in performance metrics such as accuracy and sensitivity due to class imbalance issues.

During week 5, our focus shifted towards studying ensemble learning and data augmentation and their common approaches with experiments. Additionally, we updated our software framework to include the step for class imbalance. Furthermore, we summarized the important features and visualizations for our webpage prototype. In week 6, further research on ensemble learning and resampling to discover the common approaches, gaps and improvements was conducted. Moreover, the prototype webpage was initialized using the library shiny from R. In week 7, after intensive research in the preceding week, we shifted our focus from ensemble learning to tackling the class imbalance challenge. We conducted further research to brainstorm innovative solutions to address this issue and continued developing our webpage.

In summary, the efforts of our team have brought the project forward significantly. With our current progress on optimization of our models and the ongoing web page prototype development, we are assured to deliver a robust solution aligned with our project's goals.

3 Degree of completion in comparison to previous set goal or plan

During semester break, the project development phase was delayed to accommodate our unavailability and it was updated in the original plan as shown in Figure 1. The project has achieved a 100% completion rate compared to the original plan which marks a significant progress. This calculation is based on comparing the actual progress to the initial goal after the delay adjustment.

4 Summary of progress and contribution of each team member

The team has made excellent progress, with each member contributing actively to the project and completing their assigned tasks on time and with high quality. Tasks were distributed evenly among team members to ensure a balanced workload. Some tasks were collaborative, while some were tackled individually, showcasing the variety of skills and abilities within the team.

Yi Hui primarily focuses on technical aspects, leading the development of the machine learning model. Jia Xin leads project management and provides support to everyone in researching, development of machine learning model and webpage. Eason handles webpage development. However, everyone collaborates and shares tasks, not solely focusing on specific skills.

5 Short overview of what still needs to be completed to the end of the semester

In short, continuous refinement to our proposed improved Lymphedema predictive model is still ongoing, typically in solving the current class imbalance challenge we have, to ensure a highly robust solution is being developed. The integration of the predictive model into the web page prototype is also being accomplished, with continual improvements on the prototype's user interface and additional features available on it.

Individual Team Member's Progress

Team Member (name and ID): Liaw Yi Hui (32023707)

Task attempted and degree of completion over the previous 5 weeks

Task attempted / completed	Completion (%)	Time taken (days or part of)	Comment, eg: reason for not completing if any
1. Discovered and studied the new research paper with analysis and preprocessing of the new dataset.	**100%**	**3 days**	
2. Executed the predictive models code from the research paper.	**100%**	**3 days**	
3. Reviewed and studied the machine learning libraries that have been used.	**100%**	**2 days**	
4. Developed base models and compared the performance results with the research paper's performance results.	**100%**	**7 days**	
5. Researched and developed ensemble learning and oversampling models for the dataset.	**100%**	**7 days**	
6. Researched further on ensemble learning to identify gaps and improvements in existing techniques.	**100%**	**3 days**	
7. Investigated and decided whether to focus further on ensemble learning or class imbalance issues for the final model.	**100%**	**14 days**	
8. Researched common approaches for handling class imbalance issues.	**100%**	**3 days**	
9. Brainstormed a new innovation to handle class imbalance issues.	**100%**	**10 days**	
10. Added F1 score metric for performance evaluation.	**100%**	**1 day**	
11. Developed and executed the new innovation algorithm for handling class imbalance issues and analyzed the performance results.	**100%**	**5 days**	
12. Explored the method to set up the web page and make it publicly available.	**100%**	**2 days**	

Team Member (name and ID): Eason Pang (32024584)

What has been attempted and degree of completion over the previous 5 weeks

Task attempted / completed	Completion (%)	Time taken (days or part of)	Comment, eg: reason for not completing if any
1. Searched up materials/tutorials online for implementation of the machine learning predictive model and web page prototype.	**100%**	**5 days**	
2. Jotted or listed down the machine learning predictive model's framework or steps of the new research paper we found.	**100%**	**5 days**	
3. Reviewed again the machine learning libraries that we have selected.	**100%**	**6 days**	
4. Researched on ensemble learning and data augmentation.	**100%**	**2 days**	
5. Listed down the important features and visualizations to be included in the web page prototype.	**100%**	**5 days**	
6. Conducted a comparative study on the features of the web page prototypes of existing (research paper) and proposed (ours).	**100%**	**4 days**	
7. Performed thorough research to decide whether to focus on ensemble learning or solving class imbalance issues for enhancing our predictive model.	**100%**	**14 days**	
8. Implemented the front-end of the web page prototype.	**100%**	**14 days**	
9. Implemented the back-end of the web page prototype.	**100%**	**14 days**	
10. Hosted the web page prototype online for public usage.	**100%**	**2 days**	

Team Member (name and ID): Chan Jia Xin (31859089)

What has been attempted and degree of completion over the previous 5 weeks

Task attempted / completed	Completion (%)	Time taken (days or part of)	Comment, eg: reason for not completing if any
1. Continued searching for new datasets that are suitable and reliable for our project.	**100%**	**3 days**	
2. Delved into the research paper as well as the dataset provided to understand their methodology.	**100%**	**3 days**	
3. Summarised the research paper's methodology and drew out their machine learning predictive model's framework.	**100%**	**3 days**	
4. Familiarized myself with Trello for our project management and created our sprint boards to streamline task management.	**100%**	**2 days**	
5. Reviewed and read the machine learning libraries that we have utilized.	**100%**	**2 days**	
6. Revamped our machine learning workflow (framework) from previous semester to address each stage comprehensively.	**100%**	**5 days**	
7. Researched on ensemble learning and data augmentation.	**100%**	**3 days**	
8. Explored common approaches for handling imbalanced datasets in conjunction with ensemble methods.	**100%**	**5 days**	
9. Incorporated undersampling on the dataset before developing the machine learning models and compared the results with others.	**100%**	**7 days**	
10. Looked into research papers addressing class imbalanced issues and examined their methodologies for mitigating this issue.	**100%**	**5 days**	
11. Tried finding gaps in class imbalanced issues to come up with a new innovation (added value).	**60%**	**8 days**	Did not succeed in generating a new innovation.

12. Researched on Shiny package in R for web development.	**100%**	**4 days**	
13. Developed the UI for the web page and included visualizations related to the machine learning model.	**70%**	**7 days**	Yet to include more visualizations.
14. Conducted research on Association Rule Mining.	**100%**	**3 days**	

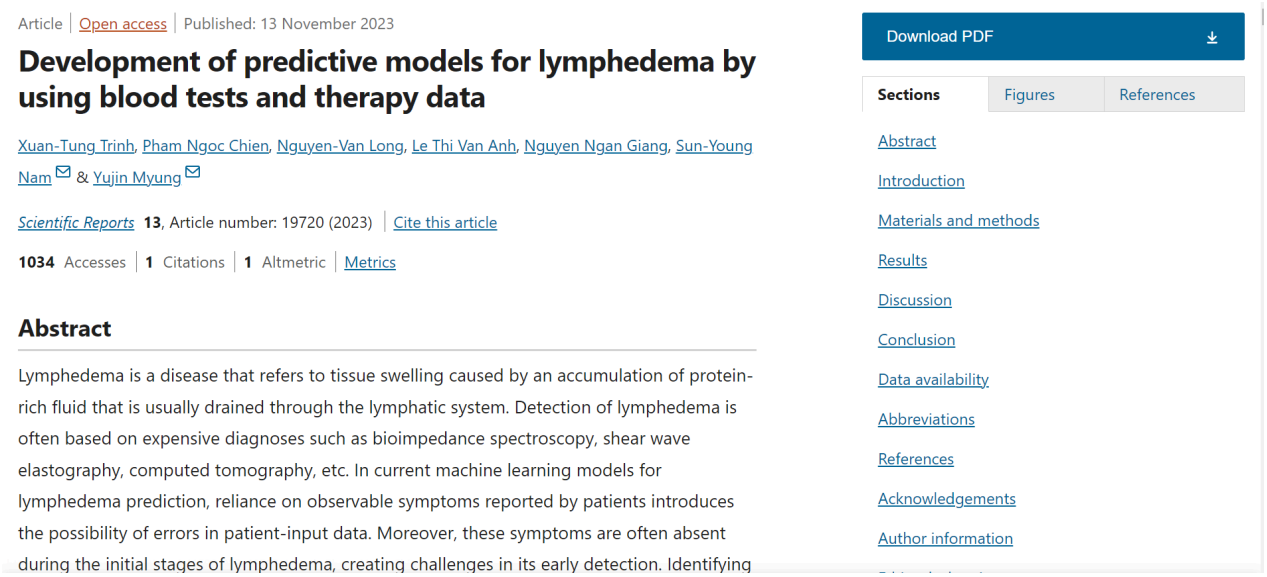
Appendix

Project plan:

		Name	Duration	Start	Finish	Predecessors
1		**Project Planning and Initial Concept**	**30 days**	**8/21/23 8:00 AM**	**9/19/23 5:00 PM**	
2		Studying and researching current algorithms	10 days	8/21/23 8:00 AM	8/30/23 5:00 PM	
3		Conducting discussions with stakeholders	30 days	8/21/23 8:00 AM	9/19/23 5:00 PM	
4		Defining functionalities and requirements	10 days	8/21/23 8:00 AM	8/30/23 5:00 PM	
5		Defining risk and approaches for risk management	10 days	8/21/23 8:00 AM	8/30/23 5:00 PM	
6		Developing project plan and schedule	10 days	8/21/23 8:00 AM	8/30/23 5:00 PM	
7		Collecting relevant data	15 days	9/1/23 8:00 AM	9/15/23 5:00 PM	2
8		Cleaning, processing and normalizing collected data	3 days	9/16/23 8:00 AM	9/18/23 5:00 PM	7
9		**Project Design**	**33 days**	**10/1/23 8:00 AM**	**11/2/23 5:00 PM**	
10		Machine learning algorithm selection	17 days	10/1/23 8:00 AM	10/17/23 5:00 PM	
11		Desiging software architecture (frontend and backend)	17 days	10/1/23 8:00 AM	10/17/23 5:00 PM	
12		Proposing prototype design	13 days	10/18/23 8:00 AM	10/30/23 5:00 PM	
13		Proposing project with literature review	16 days	10/18/23 8:00 AM	11/2/23 5:00 PM	7;10
14		**Project Development**	**55 days**	**3/7/24 8:00 AM**	**4/30/24 5:00 PM**	
15		Developing baseline performance	14 days	3/7/24 8:00 AM	3/20/24 5:00 PM	10;8
16		Developing machine learning algorithm	13 days	3/21/24 8:00 AM	4/2/24 5:00 PM	8;15
17		Evaluating, validating and optimizing machine learning model	25 days	4/3/24 8:00 AM	4/27/24 5:00 PM	16
18		Developing code for webpage prototype	24 days	4/3/24 8:00 AM	4/26/24 5:00 PM	
19		Integrating frontend and backend	1 day	4/27/24 8:00 AM	4/27/24 5:00 PM	18
20		Reviewing and debugging code	3 days	4/28/24 8:00 AM	4/30/24 5:00 PM	19
21		**Software Testing**	**15 days**	**5/1/24 8:00 AM**	**5/15/24 5:00 PM**	
22		Unit, integration, system and user acceptance testings	15 days	5/1/24 8:00 AM	5/15/24 5:00 PM	20
23		Developing documentation for the user guide	15 days	5/1/24 8:00 AM	5/15/24 5:00 PM	20
24		Developing test report	15 days	5/1/24 8:00 AM	5/15/24 5:00 PM	20
25		**Project Finalization**	**20 days**	**5/16/24 8:00 AM**	**6/4/24 5:00 PM**	
26		Writing final report	20 days	5/16/24 8:00 AM	6/4/24 5:00 PM	24
27		Improving overall project quality	20 days	5/16/24 8:00 AM	6/4/24 5:00 PM	24

Figure 1: Listings of Work Breakdown Structure with schedule (from previous semester with delay due to semester break adjustment)

Liaw Yi Hui's evidence of progress (according to sequence in table):



Article | Open access | Published: 13 November 2023

Development of predictive models for lymphedema by using blood tests and therapy data

Xuan-Tung Trinh, Pham Ngoc Chien, Nguyen-Van Long, Le Thi Van Anh, Nguyen Ngan Giang, Sun-Young Nam & Yujin Myung

Scientific Reports **13**, Article number: 19720 (2023) | Cite this article

1034 Accesses | **1** Citations | **1** Altmetric | Metrics

Abstract

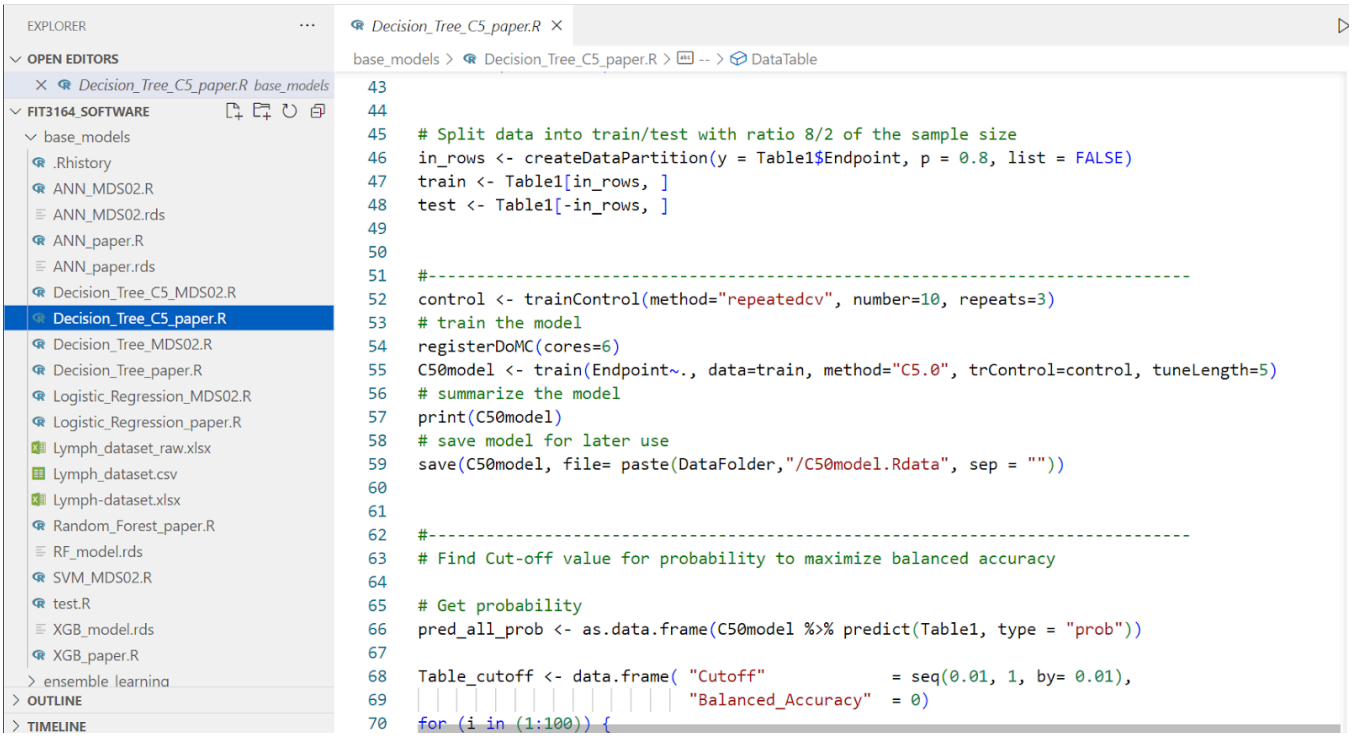
Lymphedema is a disease that refers to tissue swelling caused by an accumulation of protein-rich fluid that is usually drained through the lymphatic system. Detection of lymphedema is often based on expensive diagnoses such as bioimpedance spectroscopy, shear wave elastography, computed tomography, etc. In current machine learning models for lymphedema prediction, reliance on observable symptoms reported by patients introduces the possibility of errors in patient-input data. Moreover, these symptoms are often absent during the initial stages of lymphedema, creating challenges in its early detection. Identifying

Download PDF

Sections | Figures | References

Abstract
Introduction
Materials and methods
Results
Discussion
Conclusion
Data availability
Abbreviations
References
Acknowledgements
Author information

Figure 2: A new research paper with available dataset discovered



EXPLORER

OPEN EDITORS
Decision_Tree_C5_paper.R base_models

FIT3164_SOFTWARE
base_models
.Rhistory
ANN_MDS02.R
ANN_MDS02.rds
ANN_paper.R
ANN_paper.rds
Decision_Tree_C5_MDS02.R
Decision_Tree_C5_paper.R
Decision_Tree_MDS02.R
Decision_Tree_paper.R
Logistic_Regression_MDS02.R
Logistic_Regression_paper.R
Lymph_dataset_raw.xlsx
Lymph_dataset.csv
Lymph-dataset.xlsx
Random_Forest_paper.R
RF_model.rds
SVM_MDS02.R
test.R
XGB_model.rds
XGB_paper.R
ensemble learning
OUTLINE
TIMELINE

Decision_Tree_C5_paper.R

base_models > Decision_Tree_C5_paper.R > -- > DataTable

```
43
44
45  # Split data into train/test with ratio 8/2 of the sample size
46  in_rows <- createDataPartition(y = Table1$Endpoint, p = 0.8, list = FALSE)
47  train <- Table1[in_rows, ]
48  test <- Table1[-in_rows, ]
49
50
51  #-------------------------------------------------------------------------------
52  control <- trainControl(method="repeatedcv", number=10, repeats=3)
53  # train the model
54  registerDoMC(cores=6)
55  C50model <- train(Endpoint~., data=train, method="C5.0", trControl=control, tuneLength=5)
56  # summarize the model
57  print(C50model)
58  # save model for later use
59  save(C50model, file= paste(DataFolder,"/C50model.Rdata", sep = ""))
60
61
62  #-------------------------------------------------------------------------------
63  # Find Cut-off value for probability to maximize balanced accuracy
64
65  # Get probability
66  pred_all_prob <- as.data.frame(C50model %>% predict(Table1, type = "prob"))
67
68  Table_cutoff <- data.frame( "Cutoff"            = seq(0.01, 1, by= 0.01),
69                              "Balanced_Accuracy"  = 0)
70  for (i in (1:100)) {
```

Figure 3: A predictive model code from the selected research paper

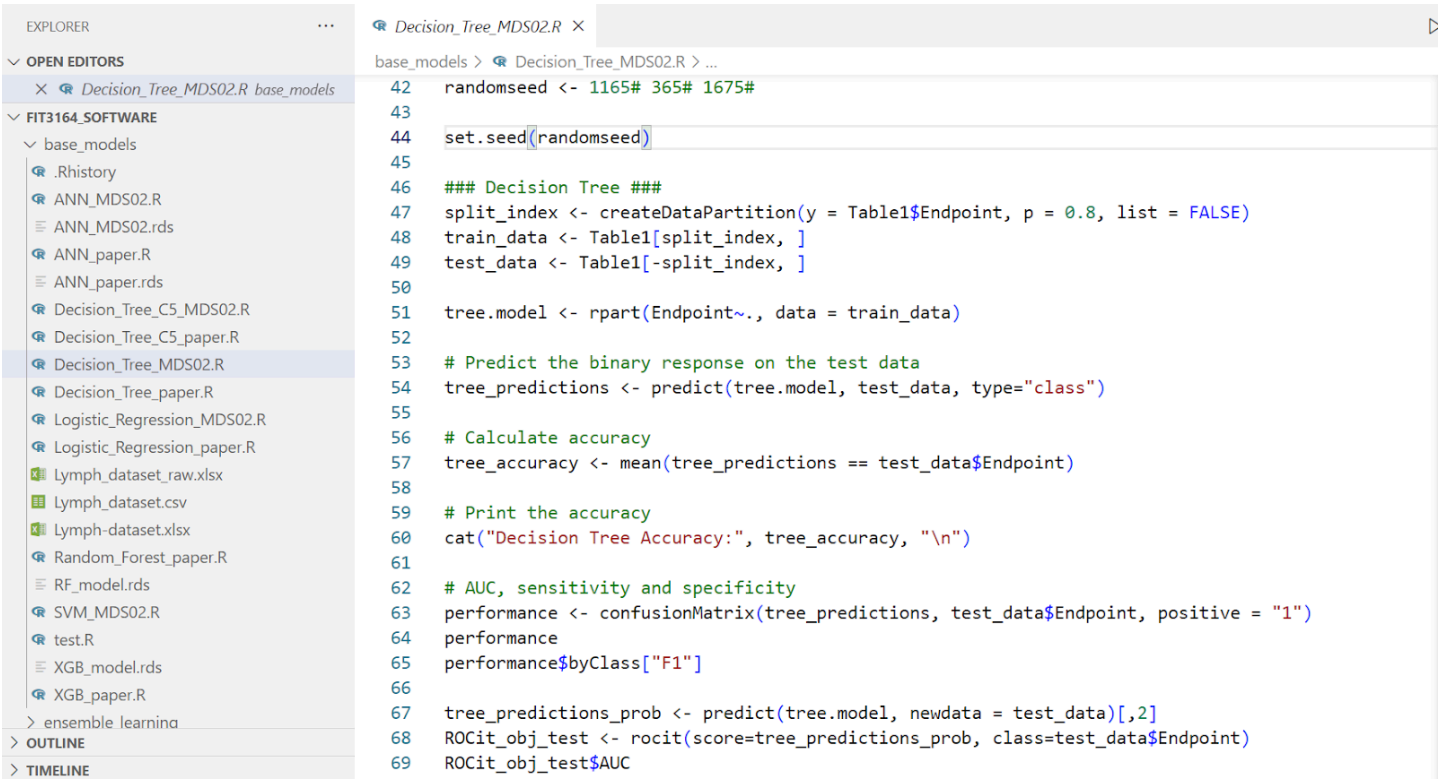


Figure 4: Code for a base model(Decision Tree)

Table of Summary for Machine Learning Models (Research Paper)

Model	Accuracy	Balanced accuracy	AUC	Sensitivity	Precision	Specificity	F1
Logistic regression	0.5489834	0.6937155	0.6935218	0.9130435	0.2625000	0.4743875	0.4077670
Decision tree	0.7615527	0.64892999	0.6722185	0.47826087	0.8915452	0.81959911	0.40552995
C5.0(DT)	0.8299445	0.8197686	0.888012	0.8043478	0.5000000	0.8351893	0.6166667
ANN	0.6802218	0.7036409	0.7531955	0.7391304	0.3133641	0.6681514	0.4401294
Random forest	0.8558226	0.8785707	0.9337538	0.9130435	0.5454545	0.8440980	0.6829268
XGB	0.9685767	0.9292147	0.9688922	0.8695652	0.9411765	0.9888641	0.9039548

Table of Summary for Machine Learning Models (Traditional)

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

Figure 5: Performance results for our base models and models from the research paper

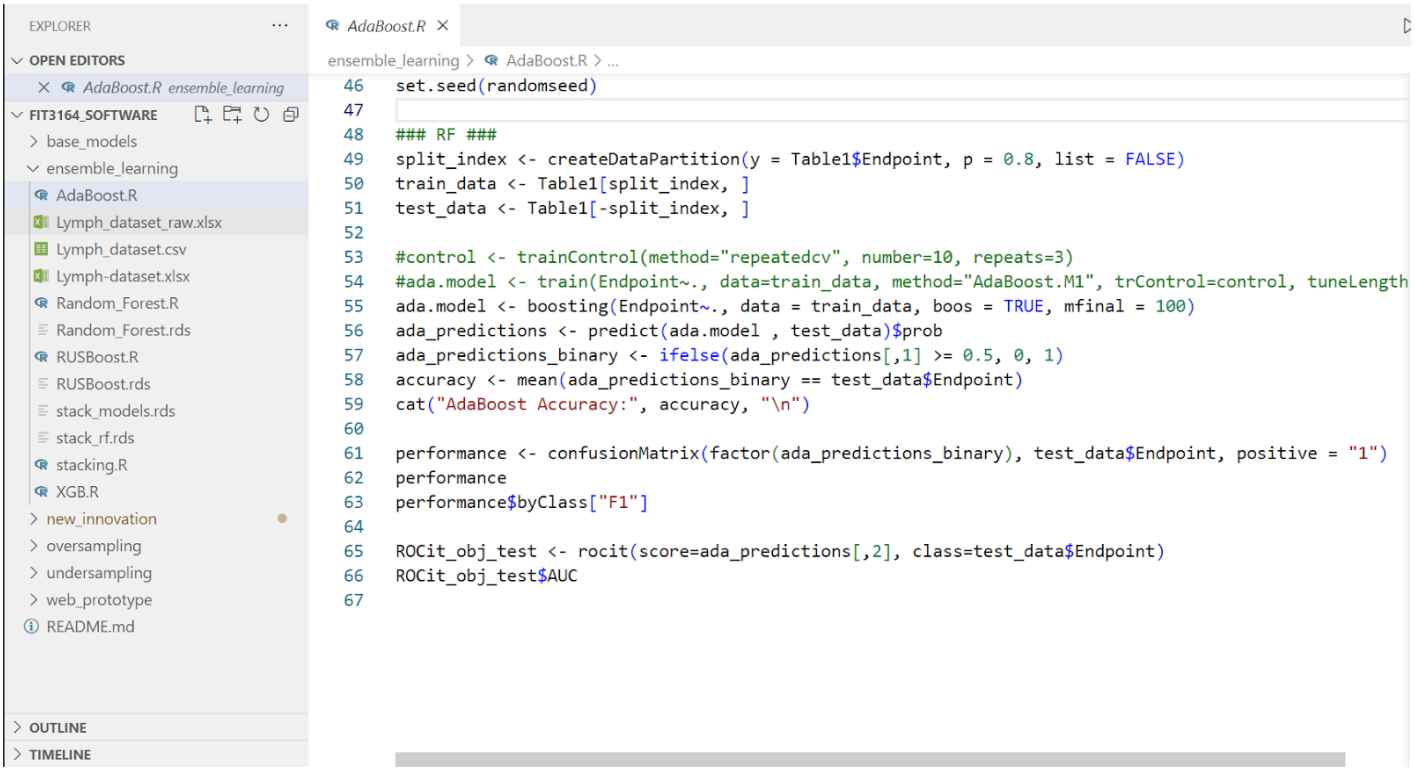


Figure 6: Code for an ensemble learning model(AdaBoost)

Table of Summary for Machine Learning Models (Ensemble Learning)

Model	Accuracy	Balanced Accuracy	AUC	Sensitivity	Precision	Specificity	F1
Stack(LR, C5.0, ANN)	0.8743068	0.73414	0.8483708	0.52174	0.66667	0.94655	0.5853659
XGBoosting	0.8909427	0.7787	0.8616975	0.6087	0.7089	0.9488	0.6549708
Random Forest	0.8927911	0.68910	0.9330396	0.38043	0.97222	0.99777	0.546875
RUSBoost	0.8003697	0.8452	0.9155854	0.9130	0.4565	0.7773	0.6086957
AdaBoost	0.8890943	0.76034	0.8761499	0.56522	0.72222	0.95546	0.6341463

Figure 7: Performance results for ensemble learning models

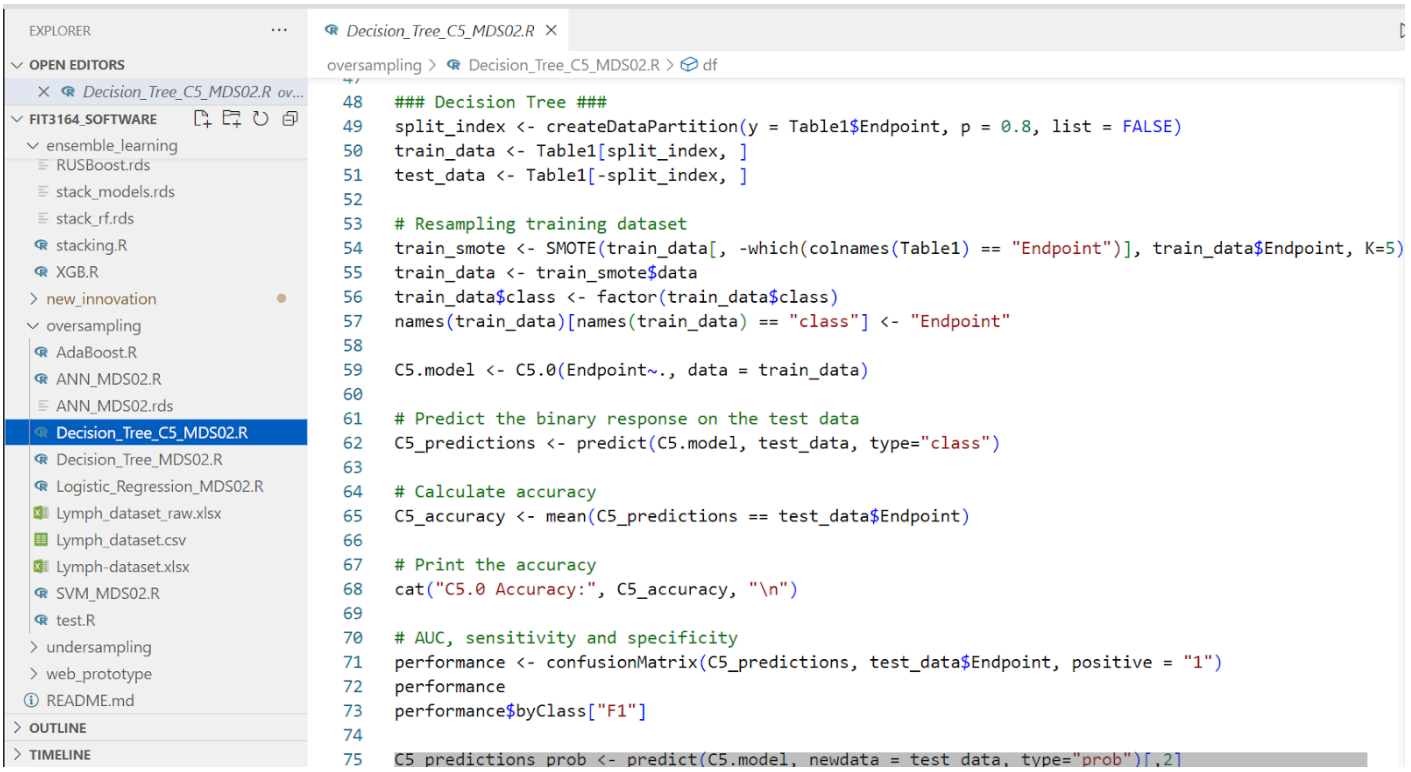


Figure 8: Code for a model(C5.0) with oversampling

Table of Summary for Machine Learning Models (Traditional with SMOTE)

Model	Accuracy	Balanced Accuracy	AUC	Sensitivity	Precision	Specificity	F1
Logistic regression	0.7338262	0.7359	0.7802605	0.7391	0.3617	0.7327	0.4857143
Decision tree	0.7707948	0.68043	0.7464293	0.54348	0.37879	0.81737	0.4464286
C5.0(DT)	0.8151571	0.7417	0.8118524	0.6304	0.4677	0.8530	0.537037
ANN	0.7264325	0.6926	0.7461993	0.6413	0.3391	0.7439	0.443609

Figure 9: Performance results for base models with oversampling

Stacking:

- Allows a training algorithm to ensemble several other similar learning algorithm predictions.
- Regression, density estimations, distance learning, and classifications.
- Less widely used than Bagging and Boosting
- Steps:
    1. Split the training set into two disjoint sets.
    2. Train several base learners on the first part.
    3. Test the base learners on the second part.
    4. Using the predictions from 3) as the inputs, and the correct responses as the outputs, train a higher level learner.
- Methods:
    - **Stacked Models**
    - Blending
    - Super Ensemble

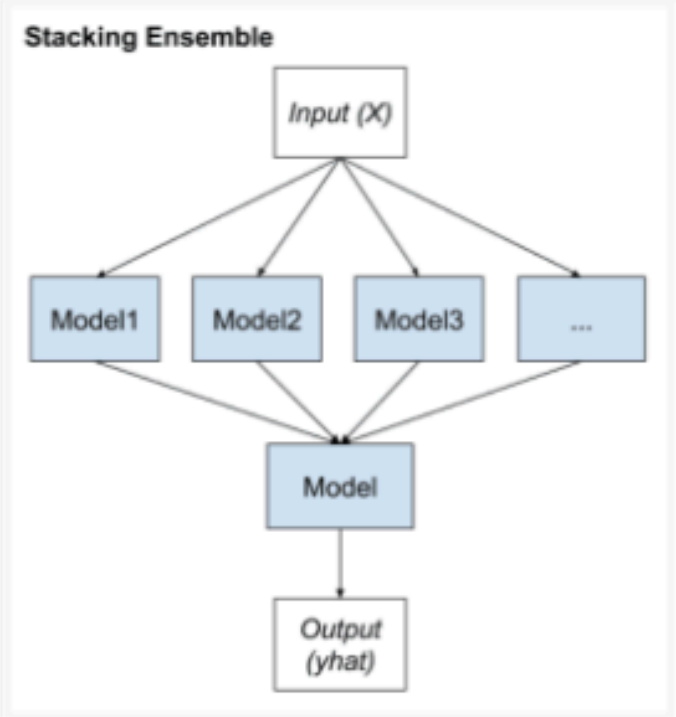


Figure 10: A part of research findings of ensemble learning

Research paper:
https://www.sciencedirect.com/science/article/pii/S0925231215001411
Common methods to solve class imbalance:

- Undersampling of the majority class
  - Randomly select a fraction of records from the majority class
- Oversampling of the minority classes
  - Randomly duplicate the records in the minority classes to increase the cardinality of the classes themselves - SMOTE
  - RAMOBoost - Oversamples the minority class using an adaptive weight adjustment procedure that shifts the decision boundary towards the difficult-to-learn examples from both the minority and majority classes
- Ensemble methods(boosting)
  - Reduce the bias towards the majority class by focusing on misclassified training patterns
- Cost-sensitive learning
  - A different cost is assigned to false negative and false positive patterns
- Asymmetric classification
  - Assign different weights and adjust the error penalties of each class.
- Dimension reduction (Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA))
  - Reduce the number of input variables (or features) in a dataset while preserving important information

Proposed approach (HardEnsemble):

1. Oversampling of the minority class, based on a variant of SMOTE
2. Undersampling of the majority class using an editing algorithm
3. Application of boosting (RUSBoost) to each new built training set

Research paper:
https://www.sciencedirect.com/science/article/pii/S0925231213011429
Proposed approach (SUNDO):

1. Divide the imbalanced dataset into a training set (75%) and a validation set (25%). Maintain the same proportion of the two classes in each set.
2. Divide the training set into two subsets, one for the minority class and one for the majority class.
3. Set the target percentages for the minority and majority classes.
4. Undersample the majority class and oversample the minority class to meet the target percentages. Oversampling adds synthetic samples to the minority class.
5. Merge the undersampled majority class and the oversampled minority class to create a new balanced training set.
6. Train the chosen classifier on the balanced training set.
7. Use the imbalanced validation set to evaluate the performance of the classifier.

Figure 11: Research findings of common approaches for class imbalance

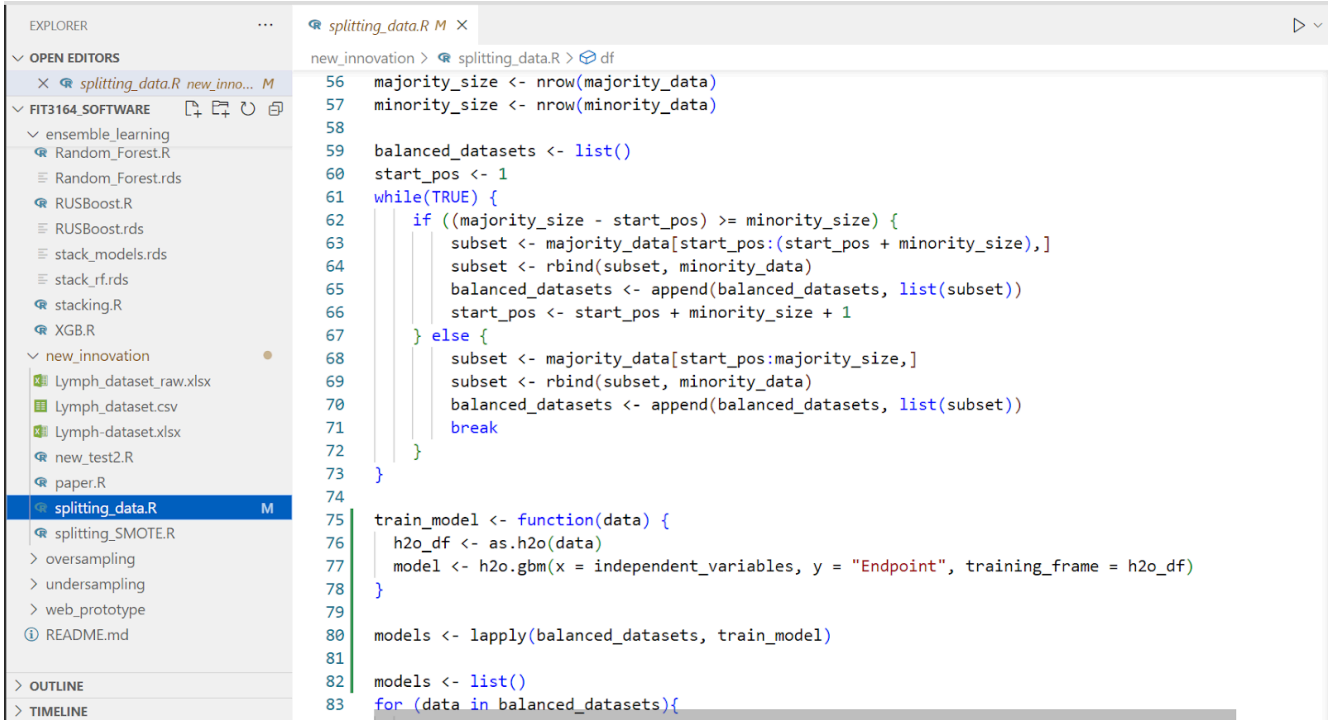


Figure 12: Code for new innovation algorithm

Table of Summary for Machine Learning Models (Splitting data - equal size to minority class)

Model	Accuracy	Balanced Accuracy	AUC	Sensitivity	Precision	Specificity	F1
Gradient Boosting Machine	0.8299	0.5	0.1607921	0.0000	NA	1	NA
Random Forest	0.8299	0.5	0.1607921	0.0000	NA	1	NA

Table of Summary for Machine Learning Models (Splitting data - half + SMOTE on minority data + RUSBoost)

Model	Accuracy	Balanced Accuracy	AUC	Sensitivity	Precision	Specificity	F1
Subset 1	0.6248	0.7221	0.860366	0.8696	0.2952	0.5746	0.4407713
Subset 2	0.7412	0.6885	0.7883945	0.6087	0.3500	0.7684	0.4444444

Table of Summary for Machine Learning Models (Splitting data - half + SMOTE on minority data + Random Forest)

Model	Accuracy	Balanced Accuracy	AUC	Sensitivity	Precision	Specificity	F1
Subset 1	0.6229	0.7253	0.8222741	0.8804	0.2956	0.5702	0.442623
Subset 2	0.7061	0.6803	0.7465987	0.6413	0.3189	0.7194	0.4259928

Figure 13: Performance results for new innovation algorithm

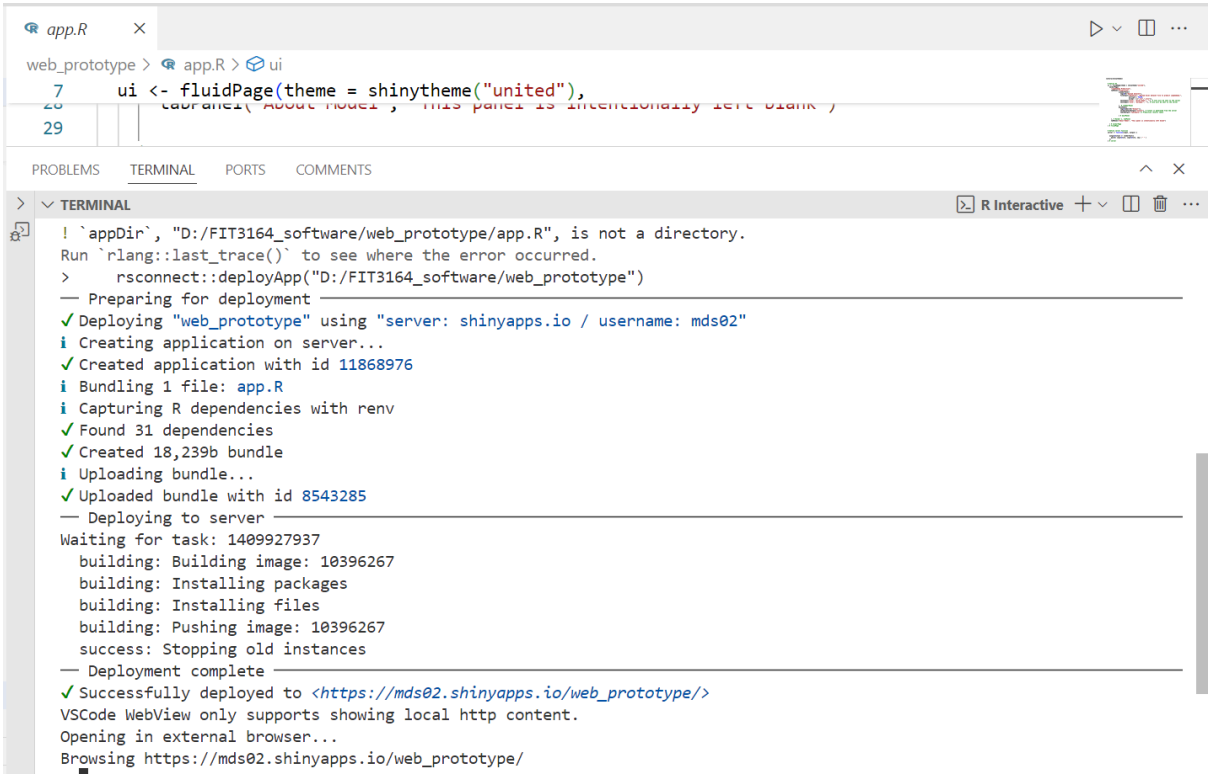


Figure 14: Exploration of method for deploying the app

Eason Pang's evidence of progress (according to sequence in table):

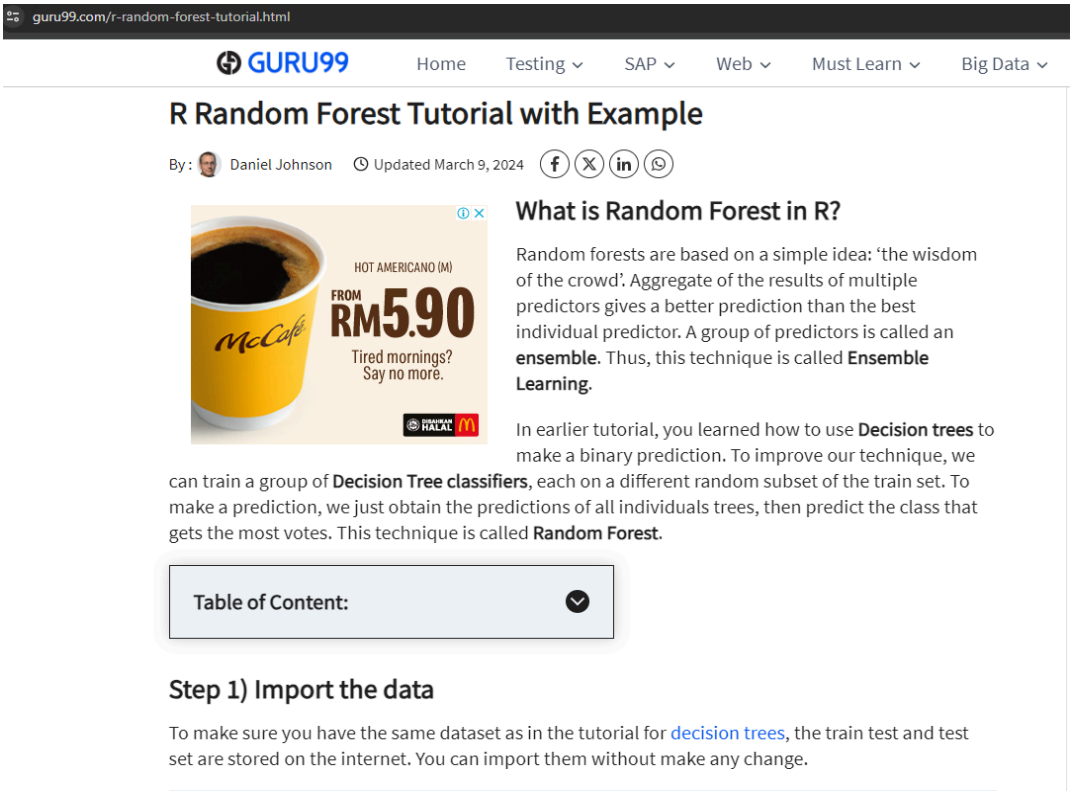


Figure 15: Found materials online for the implementation of our project

Enhancement to usual Lymphedema predictive models - proposed to use new data such as complete **blood count, serum,** and **therapy data**, to develop predictive models for predicting lymphedema (i.e., outcome is either yes or no). This approach aims to compensate for the limitations of using only observable symptoms data.

Machine learning algorithms used - Random Forest, Gradient Boosting, Decision Tree, C5.0, Logistic Regression, ANN

1. **Data collection** - collected data from 2137 patients, including 356 patients with lymphedema and 1781 patients without lymphedema
2. **Data cleaning** - missing data, inconsistent value. After cleaning the missing data, they obtained a data table of 28 parameters (variables). A dataset of 2706 rows and 29 columns was obtained after data cleaning. Among 29 columns, 16 columns are CBC test variables, three columns are serum test variables, nine columns are therapy variables, and one column was lymphedema status.
3. **Feature selection** - most important variables determined using **Random Forest** (**mean decrease accuracy**; variables that lead to largest decrease in accuracy when removed are considered more important).
4. Dataset is splitted into **80% for training** and **20% for testing** randomly, the splitting was repeated three times to obtain three random splits.
5. In the training process, machine learning algorithms were applied to the training set via **tenfold cross-validation**, in which the training data was randomly partitioned into 10 mutually exclusive subsets, with 9 subsets for training and one for internal validation.
6. Because the data of lymphedema and non-lymphedema patients in this study is **imbalanced** (17% of lymphedema and 83% of non-lymphedema), they adjusted class weights and decision threshold to deal with the imbalance problem. Trained models provide probability (between 0.0 and 1.0) of a data being lymphedema or not. The decision threshold is the probability to decide if there is lymphedema or not. It is set to 0.5 for balanced data, but in this study, it was adjusted to maximise balanced accuracy of trained models (between 0.17 and 0.30).
7. After training and obtaining trained models, they applied them on the external **validation (testing)** dataset to validate the application of the trained models.
8. **Model evaluation - performance metrics** used: Balanced accuracy, Precision, Sensitivity, Specificity, F1 Score, AUC
9. Based on the performance of developed models, chosen the best predictive model for developing a web application.
10. Create a **web application** using the Shiny package in R.

This study uses data collected from clinical tests such as blood tests and therapy data, making it more accurate than patient self-reports and limb circumference measurements

Figure 16: Machine learning predictive model's framework of the new research paper found

eFood
Vol. 1(1); February (2020), pp. 69–76
DOI: https://doi.org/10.2991/efood.k.191004.001; eISSN 2666-3066
https://www.atlantis-press.com/journals/efood

Research Article

Random Forest, Artificial Neural Network, and Support Vector Machine Models for Honey Classification

Cecilia Martinez-Castillo[1,2], Gonzalo Astray[1,*], Juan Carlos Mejuto[1], Jesus Simal-Gandara[2,*]

[1]Department of Physical Chemistry, Faculty of Food Science and Technology, University of Vigo - Ourense Campus, Ourense E32004, Spain
[2]Nutrition and Bromatology Group, Department of Analytical and Food Chemistry, Faculty of Food Science and Technology, University of Vigo - Ourense Campus, Ourense E32004, Spain

ARTICLE INFO

Article History
Received 14 May 2019
Accepted 29 September 2019

Keywords
Food authenticity
honey
Galician honeys
classification models

ABSTRACT

Different separated protein fractions by the electrophoretic method in polyacrylamide gel were used to classify two different types of honeys, Galician honeys and commercial honeys produced and packaged outside of Galicia. Random forest, artificial neural network, and support vector machine models were tested to differentiate Galician honeys and other commercial honeys produced and packaged outside of Galicia. The results obtained for the best random forest model allowed us to determine the origin of honeys with an accuracy of 95.2%. The random forest model, and the other developed models, could be improved with the inclusion of new data from different commercial honeys.

1. INTRODUCTION

In the past years, the quality of food products has been an important characteristic for consumers [1]. The definition of honey explains that it is a natural product which is produced by *Apis mellifera* (honey bees) from the nectars of different kinds of plants [2–4] or other secretions [3,4] and that has high viscosity, sweetness, and a particular aroma [5]. Honey can be categorized as blossom honey or honeydew honey [4]. Honey is composed of different sugars and other elements such as enzymes, organic acids, vitamins, or are

to limit, even eliminate, the risks of falsification [10] to ensure the food authenticity. The most common methods to adulterate honey is through addition of cheap sweeteners (corn syrup and maltose syrup, among others) or through use of honeybees that are fed sugar or other types of sucrose [6,11,12]. These two methods are in line with the assertion of Cotte et al. [10] that also reports another method of fraud consisting of misuse of the name of origin by mixing (voluntarily or not) different honeys of diverse varieties.

Figure 17: Reviewed again the machine learning libraries that we have selected

What is ensemble learning?
Ensemble learning is a machine learning technique that **enhances accuracy** and **resilience** in forecasting by **merging predictions from multiple models**. It aims to **mitigate errors or biases** that may exist in individual models by leveraging the collective intelligence of the ensemble.

Different ways on doing ensemble learning
Bagging:
- Mainly applies in classification and regression
- Increases the accuracy of models through decision trees which reduces variance to a large extent
- Methods:
  - Bagged Decision Trees
  - **Random Forest**
  - Extra Trees

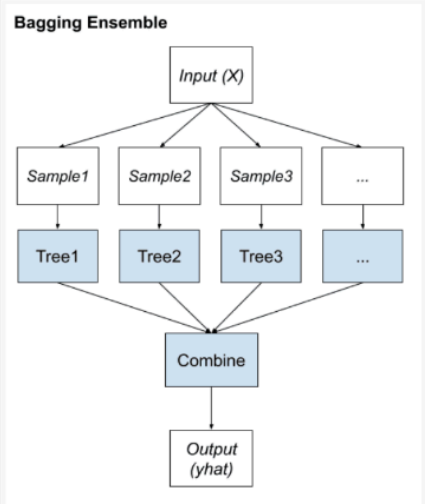


Figure 18: Part of research on ensemble learning and data augmentation

Elements and features that we could possibly include in our Lymphedema prediction website prototype:

- Users' demographic and clinical information input form
- Prediction result (likelihood, normal distribution, some charts, etc.)
- Explanation of results in text (educational content, factors that influences the results, definition and importance of features used)
- Graphical visualisation (visualise trends, importance of each feature like correlation heatmap, distribution of symptoms among all patients, indicating different age groups, etc.)
- Clinic finder
- Resources library (resources such as articles, videos, or downloadable materials related to lymphedema prevention, management, and treatment options)
- Platform/forum for users to connect with others who are affected by lymphedema, share experiences, ask questions, and provide support)
- FAQ section (for common queries about lymphedema or the predictive model or process used)
- Feedback form (for users to provide comments, suggestions, or report any issues on the website)

Figure 19: Listed down important features and visualizations that could be included in the web page prototype

Table of comparison for features available inside the website prototype:

Features	Research Paper	MDS02
Dataset uploading	Yes	Pending for trial
Template of dataset	Yes	Pending for trial
Users' demographic and clinical information input form	No	Pending for trial
Visualization for each feature used	Yes	Yes
Visualization for results (normal distribution, visualize trends, distribution of symptoms among all patients, indicating different age groups, etc.)	No	Yes
Results score (out of 1)	Yes	Yes
Summary/description of model	Yes	Yes
Relative importance of each feature used (heatmap visualization or correlation)	Yes	Yes

Figure 20: Part of the comparative study on the features of the web page prototypes of existing (research paper) and proposed (ours)

Research paper:
https://www.sciencedirect.com/science/article/pii/S0925231215001411
Common methods to solve class imbalance:

- Undersampling of the majority class
  - Randomly select a fraction of records from the majority class
- Oversampling of the minority classes
  - Randomly duplicate the records in the minority classes to increase the cardinality of the classes themselves - SMOTE
  - RAMOBoost - Oversamples the minority class using an adaptive weight adjustment procedure that shifts the decision boundary towards the difficult-to-learn examples from both the minority and majority classes
- Ensemble methods(boosting)
  - Reduce the bias towards the majority class by focusing on misclassified training patterns
- Cost-sensitive learning
  - A different cost is assigned to false negative and false positive patterns
- Asymmetric classification
  - Assign different weights and adjust the error penalties of each class.
- Dimension reduction (Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA))
  - Reduce the number of input variables (or features) in a dataset while preserving important information

Proposed approach (HardEnsemble):

1. Oversampling of the minority class, based on a variant of SMOTE
2. Undersampling of the majority class using an editing algorithm
3. Application of boosting (RUSBoost) to each new built training set

Research paper:
https://www.sciencedirect.com/science/article/pii/S0925231213011429
Proposed approach (SUNDO):

1. Divide the imbalanced dataset into a training set (75%) and a validation set (25%). Maintain the same proportion of the two classes in each set.
2. Divide the training set into two subsets, one for the minority class and one for the majority class.
3. Set the target percentages for the minority and majority classes.
4. Undersample the majority class and oversample the minority class to meet the target percentages. Oversampling adds synthetic samples to the minority class.
5. Merge the undersampled majority class and the oversampled minority class to create a new balanced training set.
6. Train the chosen classifier on the balanced training set.
7. Use the imbalanced validation set to evaluate the performance of the classifier.

Figure 21: Part of research findings to decide whether to focus on ensemble learning or solving class imbalance issues for enhancing our predictive model

```
# define UI
ui <- fluidPage(theme = shinytheme("united"),
  navbarPage(
    "Lymphedema Prediction",
    tabPanel("Prediction",
             sidebarPanel(
               tags$h3("Upload dataset"),
               fileInput("DataFile", "Upload Excel dataset file to predict Lymphedema:",
                         multiple = FALSE,
                         accept = c(".xls",".xlsx", ".csv")),
               tags$h5("Format accepted: .xls, .xlsx, .csv"),
               tags$strong("Template of dataset:"),
               div(downloadButton("DownloadData", "Download"), style = "margin-bottom: 20px;"),
             ), # sidebarPanel
             mainPanel(
               tags$label(h3("Output")),
               verbatimTextOutput("txtout"), # txtout is generated from the server
               tableOutput('tabledata') # Prediction results table

             ) # mainPanel

    ), # Navbar 1, tabPanel
    tabPanel("About Model",
             fluidRow(
               column(12, tags$h3("Model Performance"))
             ),
             fluidRow(
               column(12, tableOutput("table"))
             ),
             fluidRow(
               column(4, tags$h3("ROC Curve")),
               column(4, tags$h3("____")),
               column(4, tags$h3("____"))
```

Figure 22: Part of code for the front-end of the web page prototype

```r
# define server function
server <- function(input, output) {

  # uploading dataset
  datasetInput <- reactive({
    inFile <- input$DataFile
    DataTable <- read.csv(inFile$datapath, sheet = 1)
    return(DataTable)
  })

  # table of input dataset
  output$DataTable <- renderTable({
    datasetInput()
  })

  # downloadable csv template of selected dataset
  output$DownloadData <- downloadHandler(
    filename = function() {
      paste("DataTemplate", "csv", sep=".")
    },
    content = function(file) {
      file.copy("DataTemplate.csv", file)
    },
    contentType = "ExcelFile"
  )

  # predicting lymphedema for the selected patient
  # to be added

  # output section
  output$txtout <- renderText({
```

Figure 23: Part of code for the back-end of the web page prototype

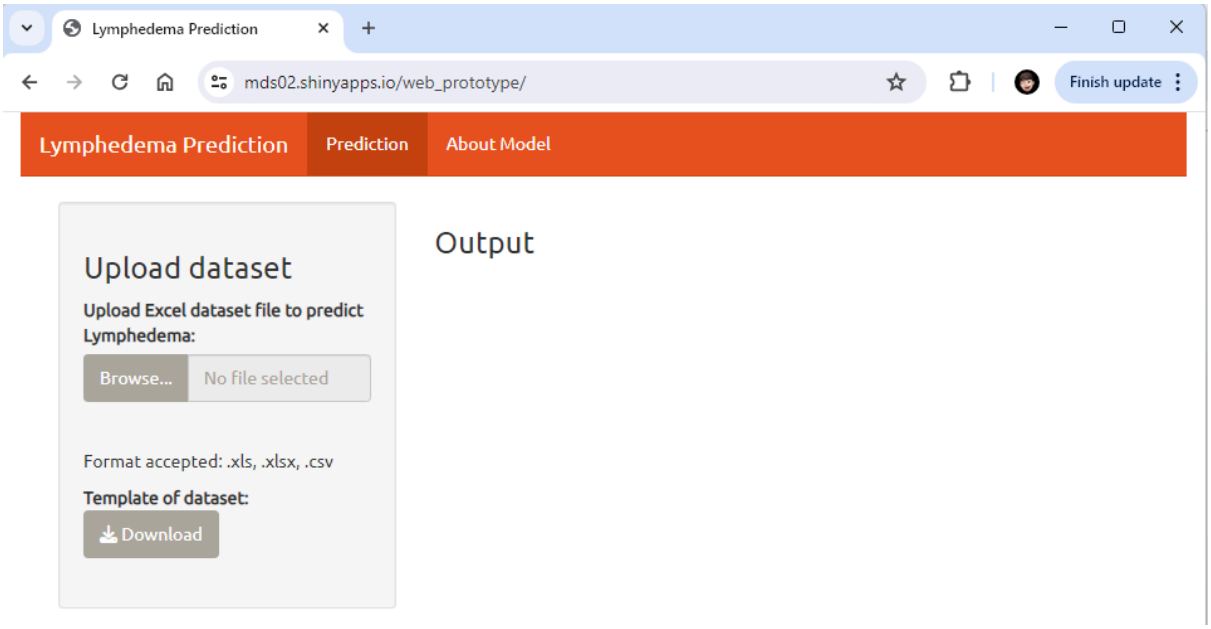
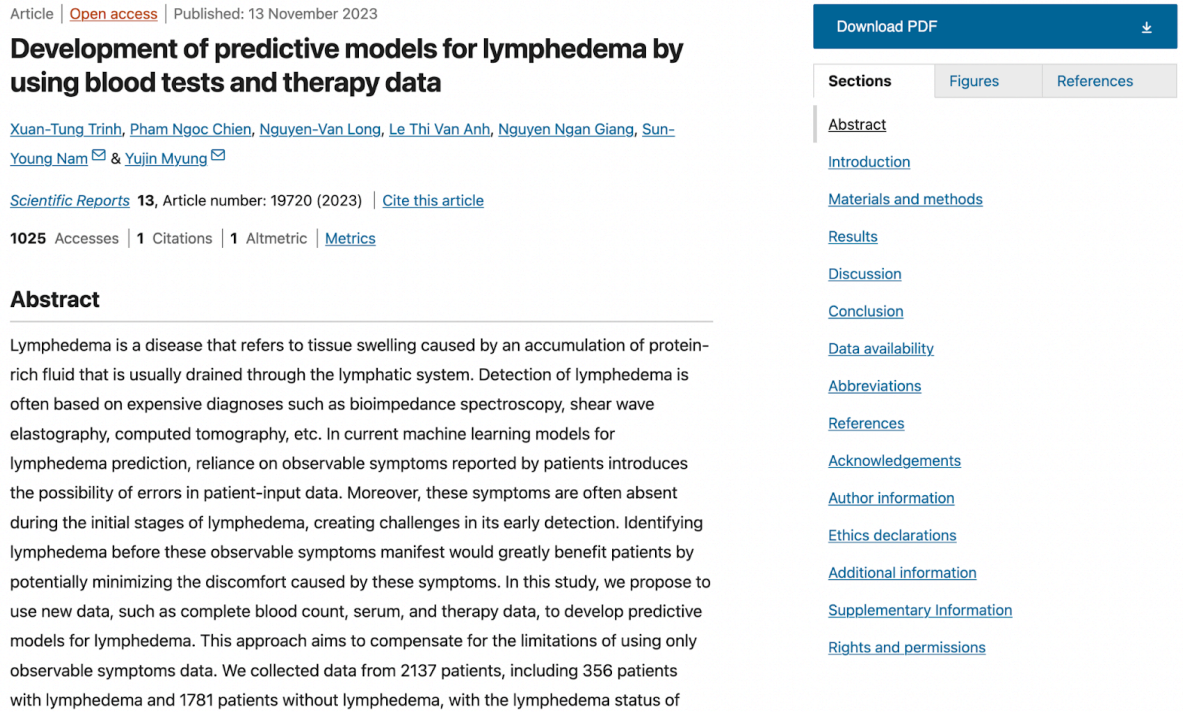


Figure 24: Hosted the web page prototype online for public usage

Chan Jia Xin's evidence of progress (according to sequence in table):



Article | Open access | Published: 13 November 2023

Development of predictive models for lymphedema by using blood tests and therapy data

Xuan-Tung Trinh, Pham Ngoc Chien, Nguyen-Van Long, Le Thi Van Anh, Nguyen Ngan Giang, Sun-Young Nam & Yujin Myung

Scientific Reports **13**, Article number: 19720 (2023) | Cite this article

1025 Accesses | **1** Citations | **1** Altmetric | Metrics

Abstract

Lymphedema is a disease that refers to tissue swelling caused by an accumulation of protein-rich fluid that is usually drained through the lymphatic system. Detection of lymphedema is often based on expensive diagnoses such as bioimpedance spectroscopy, shear wave elastography, computed tomography, etc. In current machine learning models for lymphedema prediction, reliance on observable symptoms reported by patients introduces the possibility of errors in patient-input data. Moreover, these symptoms are often absent during the initial stages of lymphedema, creating challenges in its early detection. Identifying lymphedema before these observable symptoms manifest would greatly benefit patients by potentially minimizing the discomfort caused by these symptoms. In this study, we propose to use new data, such as complete blood count, serum, and therapy data, to develop predictive models for lymphedema. This approach aims to compensate for the limitations of using only observable symptoms data. We collected data from 2137 patients, including 356 patients with lymphedema and 1781 patients without lymphedema, with the lymphedema status of

Download PDF

Sections | Figures | References

Abstract
Introduction
Materials and methods
Results
Discussion
Conclusion
Data availability
Abbreviations
References
Acknowledgements
Author information
Ethics declarations
Additional information
Supplementary Information
Rights and permissions

Figure 25: Delved into the research paper to understand their methodology

Enhancement to usual Lymphedema predictive models - proposed to use new data such as complete **blood count, serum,** and **therapy data**, to develop predictive models for predicting lymphedema (i.e., outcome is either yes or no). This approach aims to compensate for the limitations of using only observable symptoms data.

Machine learning algorithms used - Random Forest, Gradient Boosting, Decision Tree, C5.0, Logistic Regression, ANN

1. **Data collection** - collected data from 2137 patients, including 356 patients with lymphedema and 1781 patients without lymphedema
2. **Data cleaning** - missing data, inconsistent value. After cleaning the missing data, they obtained a data table of 28 parameters (variables). A dataset of 2706 rows and 29 columns was obtained after data cleaning. Among 29 columns, 16 columns are CBC test variables, three columns are serum test variables, nine columns are therapy variables, and one column was lymphedema status.
3. **Feature selection** - most important variables determined using **Random Forest** (**mean decrease accuracy**; variables that lead to largest decrease in accuracy when removed are considered more important).
4. Dataset is splitted into **80% for training** and **20% for testing** randomly, the splitting was repeated three times to obtain three random splits.
5. In the training process, machine learning algorithms were applied to the training set via **tenfold cross-validation**, in which the training data was randomly partitioned into 10 mutually exclusive subsets, with 9 subsets for training and one for internal validation.
6. Because the data of lymphedema and non-lymphedema patients in this study is **imbalanced** (17% of lymphedema and 83% of non-lymphedema), they adjusted class weights and decision threshold to deal with the imbalance problem. Trained models provide probability (between 0.0 and 1.0) of a data being lymphedema or not. The decision threshold is the probability to decide if there is lymphedema or not. It is set to 0.5 for balanced data, but in this study, it was adjusted to maximise balanced accuracy of trained models (between 0.17 and 0.30).
7. After training and obtaining trained models, they applied them on the external **validation (testing)** dataset to validate the application of the trained models.
8. **Model evaluation - performance metrics** used: Balanced accuracy, Precision, Sensitivity, Specificity, F1 Score, AUC
9. Based on the performance of developed models, chosen the best predictive model for developing a web application.
10. Create a **web application** using the Shiny package in R.

This study uses data collected from clinical tests such as blood tests and therapy data, making it more accurate than patient self-reports and limb circumference measurements

Figure 26: Machine learning predictive model's framework of the new research paper found

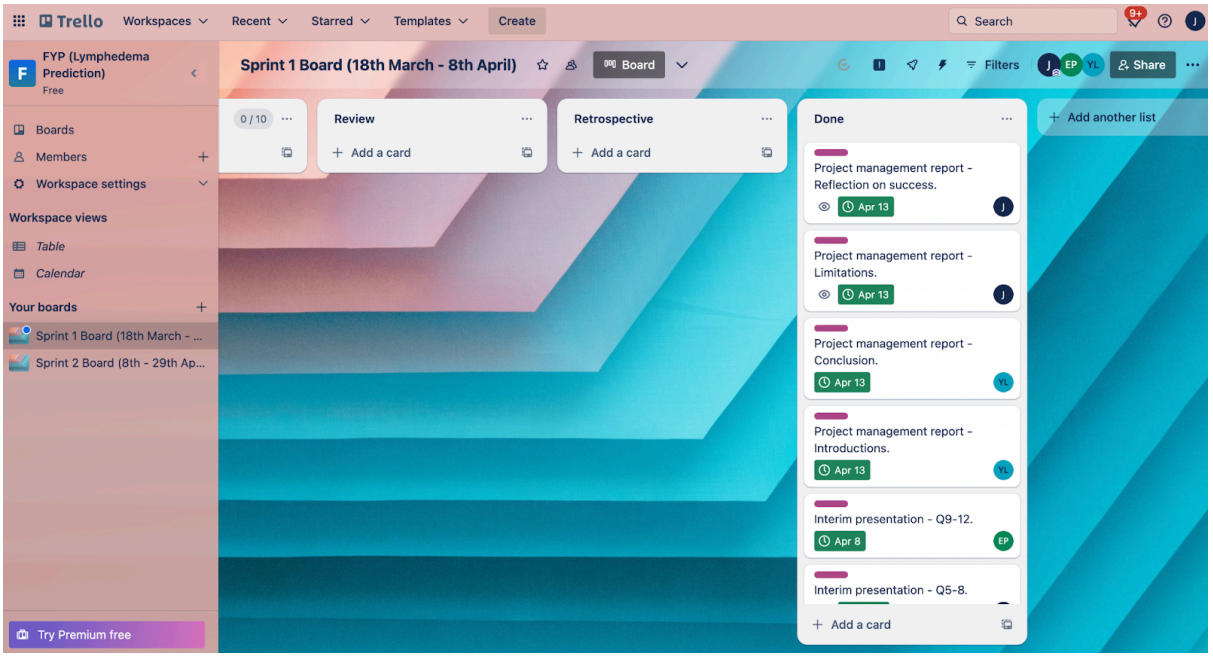
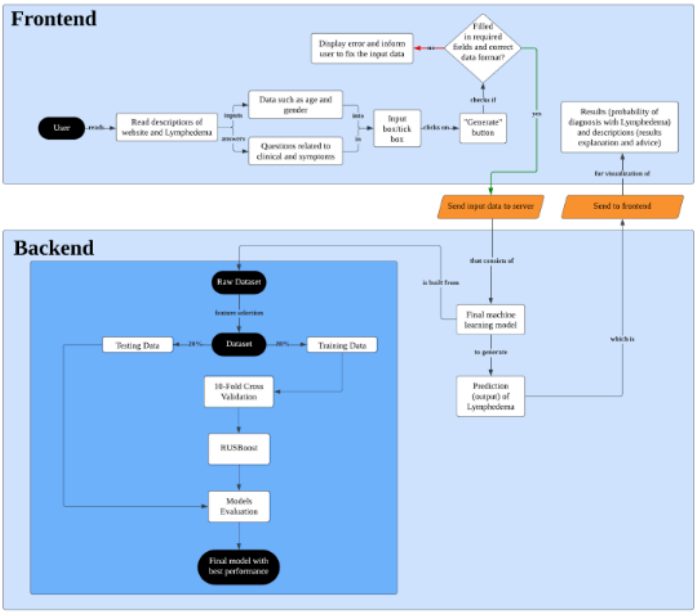


Figure 27: Trello Sprint Boards



Using the dataset provided by the paper, which has been preprocessed to remove any inconsistencies or errors, we can proceed with the following steps:

1. **Feature selection** - most important variables determined using **Random Forest** (**mean decrease accuracy**; variables that lead to largest decrease in accuracy when removed are considered more important).
2. Dataset is splitted into **80% for training** and **20% for testing** randomly.
3. In the training process, machine learning algorithms were applied to the training set via **tenfold cross-validation**, in which the training data was randomly partitioned into 10 mutually exclusive subsets, with 9 subsets for training and one for internal validation.
4. **RUSBoost** - adaptation of AdaBoost, but employs a technique of random under-sampling, aiming to balance the dataset by reducing the representation of the majority class at each boosting iteration.
5. After training and obtaining trained models, they applied them on the external **validation (testing)** dataset to validate the application of the trained models.
6. **Model evaluation - performance metrics** used: Balanced accuracy, Accuracy, Precision, Sensitivity, Specificity, AUC

Figure 28: Updated machine learning workflow (framework)

Techniques to handle class imbalance - resampling

1. **Oversampling** - increasing number of instances in the minority class
    - Duplicate or generate synthetic data
    - Pro: better learn the characteristics of the minority class
    - Con: overfitting
    - SMOTE (Synthetic Minority Over-sampling Technique)
2. **Undersampling** - reducing number of instances in the majority class
    - Pro: reduce overfitting risk
    - Cons: loss of information, biased model, poor performance on majority class
    - RUS

- Resample, and then ensemble to help with class imbalance

BalancedBaggingClassifier

Instead of randomly sampling instances with replacement (as in traditional Bagging), Balanced Bagging uses balanced bootstrapped samples. This means that each bootstrapped sample contains an equal number of instances from each class, ensuring that the minority class is adequately represented in each subset.

This classifier takes two special parameters “sampling_strategy” and “replacement”. The sampling_strategy decides the type of resampling required (e.g. ‘majority’ – resample only the majority class, ‘all’ – resample all classes, etc) and replacement decides whether it is going to be a sample with replacement or not.

An illustrative example is given below:

Figure 29: A part of research findings on common approaches to tackle imbalanced datasets

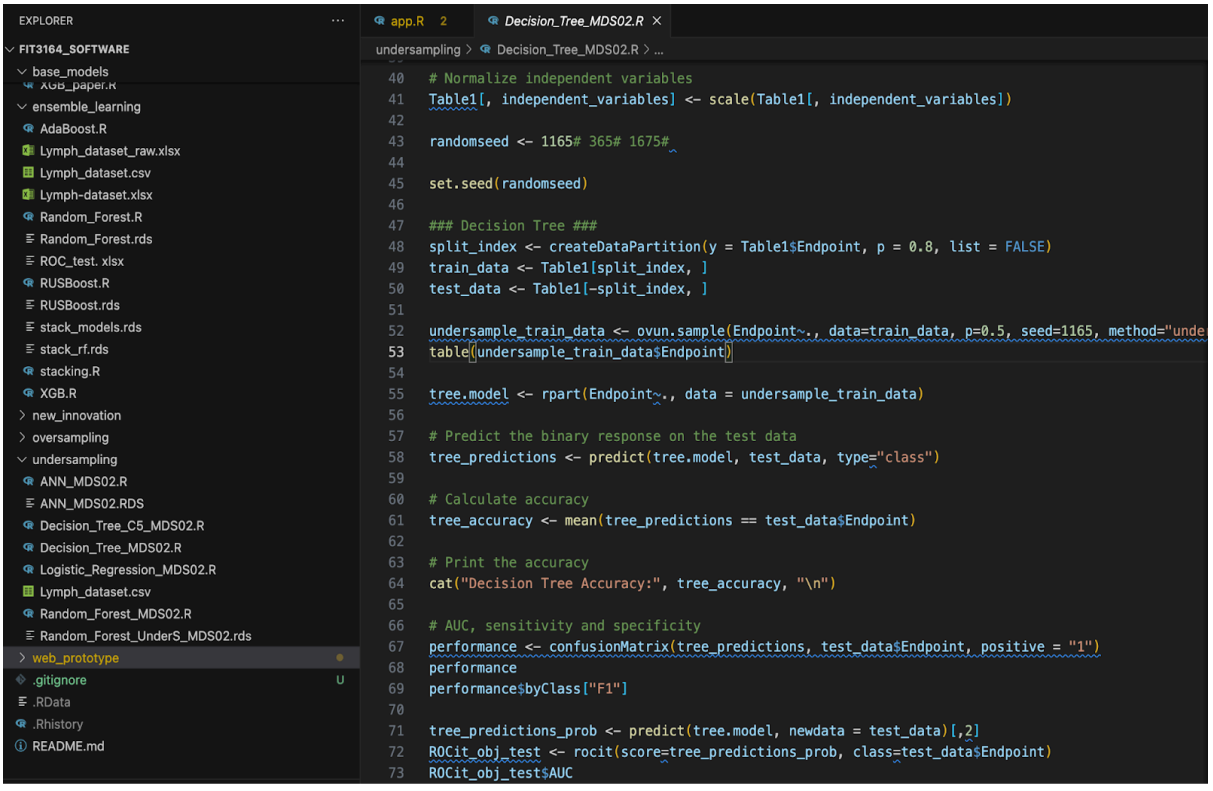
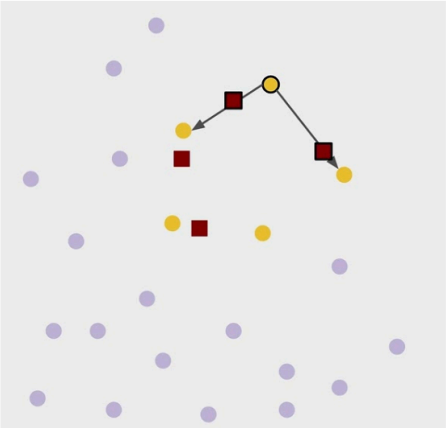


Figure 30: Code for a model with undersampling

Research paper:
https://www.researchgate.net/publication/348257059_Random_and_Synthetic_Over-Sampling_Approach_to_Resolve_Data_Imbalance_in_Classification



- SMOTE
1. Identify a data point from minority class
2. Compute its k nearest neighbour (typically k = 5) in the feature space.
3. Commonly measured using Euclidean distance
4. Create synthetic data between data point and its neighbours
5. Repeat 1-4

Research paper:
https://www.nature.com/articles/s41598-022-21046-1?utm_medium=affiliate&utm_source=commission_junction&utm_campaign=CONR_PF018_ECOM_GL_PHSS_ALWYS_DEEPLINK&utm_content=textlink&utm_term=PID5835937&CJEVENT=71ef2cb7fd2511ee83a373a90a18ba73

- SD-KMSMOTE method, based on the spatial distribution of minority samples
- SMOTE treats all minority class samples equally and does not consider the class information of the neighbouring samples.
- If selected minority class samples that are surrounded by majority class samples, newly synthesised samples will overlap with the surrounding majority class samples.

Figure 31: Research papers addressing class imbalanced issues

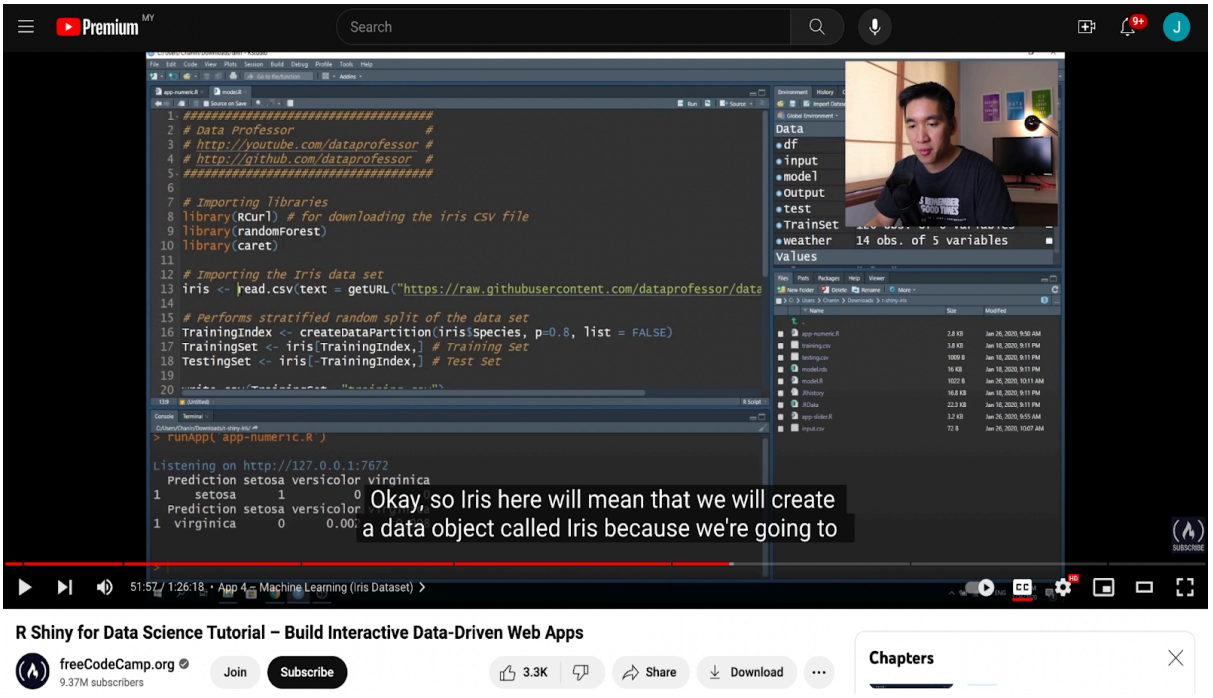


Figure 32: Learning a new package, Shiny in R to develop webpage

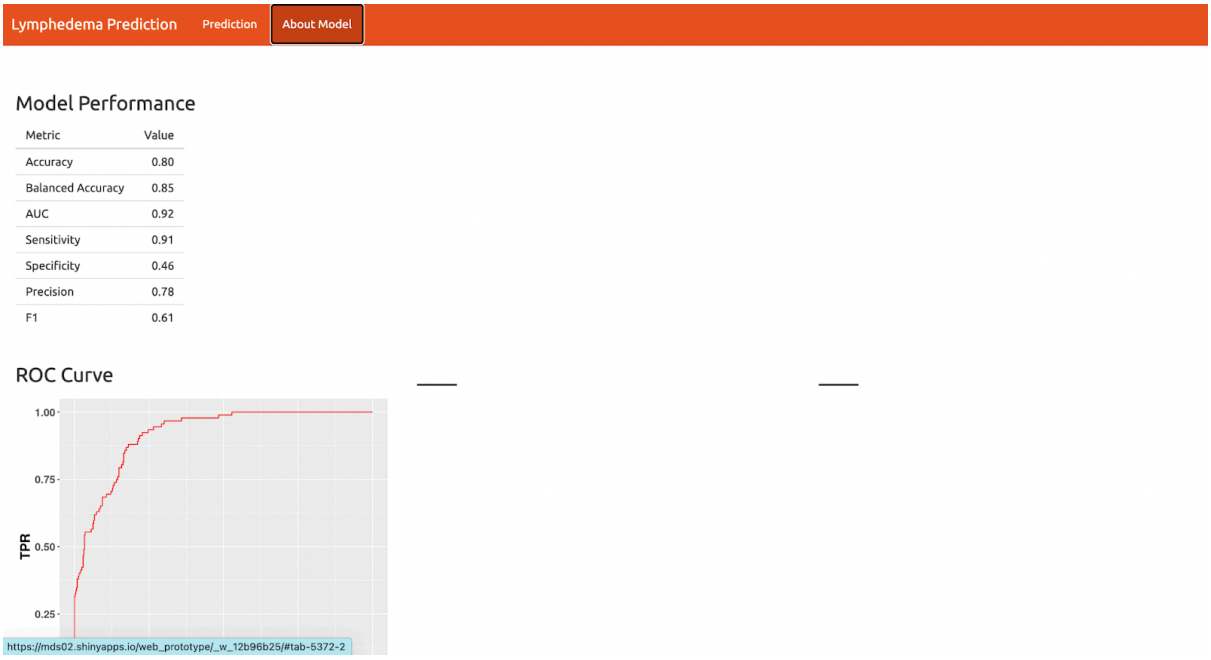


Figure 33: Webpage with visualizations

Association Rule Mining

- A technique used to discover interesting relationships or patterns in datasets.
- Focuses on uncovering associations between variables in datasets.
- Complement other techniques used to address class imbalance in machine learning, such as resampling methods (e.g., under-sampling, over-sampling).
- By providing insights into the relationships between variables, association rule mining can inform the selection of features to better handle imbalanced datasets.

Association rule mining can complement oversampling by:

1. Synthetic Data Generation: Association rule mining can provide insights into the relationships between different features in the dataset. This information can be used to generate synthetic data points that preserve these relationships.

Figure 34: A part of research findings on association rule mining

References

Lymphedema. (2017). Retrieved from https://www.mayoclinic.org/diseasesconditions/lymphedema/symptoms-causes/syc-20374682

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.

The outputs generated were modified and incorporated into this document.