Combinging Multiple Imputation and Cross-Validation for Predicting Survival of ECMO Treatment in ARDS Patients

Robert Edwards

2416963E

MASTER THESIS

Biostatistics

Acknowledgements

To my peers, alone we were sinking but together we swam.

To my family, for keeping me sane in the bipolar Scottish weather.

To Google, you da best.

Contents

1 Introduction

1.1 Discussion of the Context

- Description of Acute Respiratory Syndrome
- Description of ECMO treatment

1.2 Aims of the Proposed Research

Prediction in medical data can often be difficult; imbalanced class distributions and poor predictive covariates. If the sample size is small, then prediction becomes even more difficult. Some of these issues arise from the experimental design of the study but little can be rememdied post-hoc. Missing values in the data complicate analysis even further and are often handled either by dropping missing observations or filling in the missing value by the mean. Both methods can be valid if certain assumptions hold, but useful information is either lost to the analysis or the natural distribution of the data is effected.

Multiple imputation is another method for handling missing data that is not yet common in analysis of medical datasets **(Citation)**. This method both allows retention of observations in the analysis as well as accounts for the uncertainty of the imputed value. The advantages come at the cost of complexity and increased computation time. Multiple datasets must be imputed and results somehow pooled. This paper investigates the use of multiple imputation in increasing prediction performance in a medical dataset.

1.3 Questions of Interest

The three main questions of interest this paper aims to answer are:

1. Can ECMO treatment survival (`ECMO_Survival`) be accurately predicted by PreECMO biomedical markers?
2. What is the future expected performance of predictions?
3. Which biomedical markers are needed for accurate prediction and which can be dropped?

1.4 Study Population & Data Description

- Description of the study and variables invovled

The dataset is composed of 450 observations on patients with Acute Respiratory Distress Syndrome who underwent ECMO treatment. The response variable, `ECMO_Survival`, is a binary categorical variable for survival indication with levels "Y" and "N". 33 covariates are included in the analysis, two of which are categorical, and 31 continuous. The binary categorical variable `Gender` has two levels for "m", "f" and `Indication` is a seven level disease

indicator. `Age` is a contnuous variable included in the analysis. The remaining variables are biomedical markers from hospital measurements.

2 Methodology

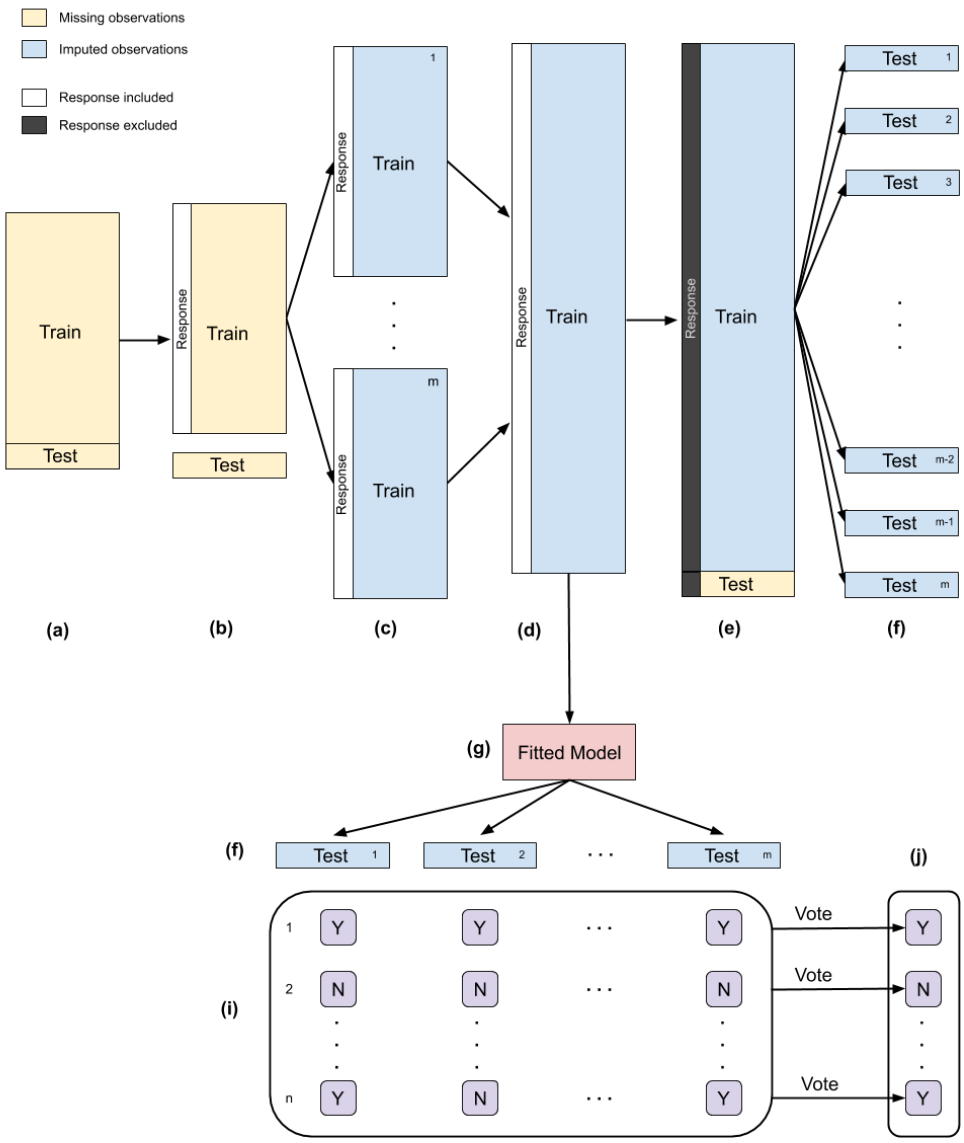

Figure 1: Outline of the algorithm used to pool predictions from multiple imputation. (a) Step 1. (b) Step 2. (c) Step 3. (d) Step 4. (e) Step 5. (f) Step 6.

2.1 PreProcessing

- Standardizing - only on continuous data
- Mean-centering
- Scaling
- Yeo-Johnson Transformation

2.2 Validation & Cross-validation

When building a classification model, it is important to asses its ability to produce valid predictions. If there are ample number of observations, one way to asses model performance is to randomly split the dataset into training, validation, and test sets. The training set is used to fit the model, which is then used to predict the classes for the observations in the validation set; the validation set is used to estimate prediction error and tune hyperparameters for model selection; the test set is used to estimate future prediction performance for the model/hyperparameters chosen. To simulate the model predicting on future, unseen data, the test set should be kept isolated. The model can overfit the data if feature manipulation and hyperparamter tuning are done before randomly splitting the data. If standarzation and transformation of the covariates is done on the entire dataset, information from the training set can "leak" into the test set and the true test error will be underestimated.

If there is insufficient data to split into three parts then a suitable alternative is K-fold cross-validation. It is one of the simplest and most widely used method for estimating prediction error [Hastie et al., 2009]. The data is randomly split into K folds, where the K^{th} fold is taken as the validation set and the the remaining $K-1$ folds are used for training the model. The procedure is then repeated K times and the prediction error averaged. K-fold cross validation is most useful on sparse datasets as it allows more observations to be used in training the model. The choice of K can effect the variability of the prediction error; if $K = 1$, the model will overfit the data and prediction error will be highly variable and if $K = n$ (the number of observation in the dataset), the model is fit with no validation set for training parameters. Typical values used are $K = 5$ & 10 [Hastie et al., 2009].

a training and a test set, respectively, preserving class proportions using the `createDataPartition()` from the **caret** package.

2.3 Models

There are many classification methods, some perform well on many types of data and others perform better on certain types of data. A variety of classification methods are explored toward the aim of predicting survival of ECMO treatment, including parametric methods with many assumptions and high bias as well as non-parametric methods with higher variability. The five explored on the ARDS dataset in this paper are: Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbors, and Random Forests.

2.3.1 Logistic Regression

Logistic regression is a widely used approach in machine learning and medicine for binary classification. It is a generalisation of linear regression that models the posterior probabilities of the Y classes. A logit link is used to ensure the posterior probabilities sum to one and are bounded by [0,1]. For two classes, the model has the form

$$\text{logit}\Big(\Pr(Y|X)\Big) = \log\frac{\Pr(Y=1|X=x)}{\Pr(Y=2|X=x)} = \mathbf{x}_i^T\boldsymbol{\beta}$$

The posterior probabilities are estimated by maximizing the log-likelihood function to find the parameter estimates, $\hat{\boldsymbol{\beta}}$, to obtain estimates of the probabilities:

$$\Pr(Y=1|X) = \frac{\exp(\mathbf{x}_1^T\hat{\boldsymbol{\beta}})}{1+\sum_{i=1}^{2}\exp(\mathbf{x}_i^T\hat{\boldsymbol{\beta}})}$$

2.3.2 LDA and QDA

Discriminant Analysis is a widely used set of classification methods. A generlization of Fisher's Linear Discriminant [FISHER, 1936], discriminant functions are created through a combination of the explanatory variables that characterize the classes.

Let $p(X|Y)$ be the densities of distributions of the observations for each class and let π_Y denote the prior probabilities of the classes; that is, the prior probability that a randomly sampled observation belongs to the Y^{th} class based on the class proportions. The posterior probabilities may be written using Bayes Theorem as:

$$p(Y|X) = \frac{p(X|Y)\ \pi_Y}{p(X)} \propto p(X|Y)\ \pi_Y \tag{1}$$

Suppose the class distribution for class Y is Multivariate Normal with mean μ_Y and covariance matrix Σ_Y, so that:

$$p(X|Y) = \frac{1}{(2\pi_Y)^{p/2}|\boldsymbol{\Sigma}_Y|^{1/2}}\exp\left[-\frac{1}{2}(X-\mu_Y)^T\boldsymbol{\Sigma}_Y^{-1}(X-\mu_Y)\right] \tag{2}$$

In comparing two classes, it is sufficient to look at the log-ratio:

$$\log\frac{\Pr(Y=1|X=x)}{\Pr(Y=2|X=x)} = \log\frac{p(X|Y=1)}{p(X|Y=2)} + \log\frac{\pi_1}{\pi_2} \tag{3}$$

and using Bayes Discriminant Rule stating that *an observation should be allocated to the class with the largest posterior probability.* From Equation (1), the posterior probability may be written as

$$p(Y|X) \propto \exp\left(Q_Y\right) \tag{4}$$

where

$$Q_Y = (X-\mu_Y)\Sigma_Y^{-1}(X-\mu_Y)^T + \log|\Sigma_Y| - 2\log\ \pi_Y \tag{5}$$

defines the Quadratic Discriminant Function for class Y. The Bayes Discriminant Rule is then: *allocated the observation to the class with the largest QDF*. This method of classification is called *Quadratic Discriminant Analysis* (QDA) because the decision boundaries between classes are elliptical and defined by Q_Y, an equation quadratic in X. If the covariance matrix, Σ_Y is assumed to be equal for each class then

$$L_Y = X\Sigma_Y^{-1}\mu_Y^T - \frac{1}{2}\mu_Y\Sigma_Y^{-1}\mu_Y^T - \log \pi_Y \tag{6}$$

defines the *Linear Discriminant Function*. This method has linear decision boundaries between classes defined by L_Y, an equation linear in X, and is known ad *Linear Discriminant Analysis* (LDA). The Bayes Discriminant Rule is then: *allocated the observation to the class with the largest LDF*.

There is a bias-variance trade-off; both assume the covariates are normally distributed, there is no multicollinearity, and the observations are independent [Cover, 1965]. LDA additionally assumes equal class covariances. Discriminant Analysis can only utilize continuous covariates with no missing observations. The bias from simple linear or quadratic class boundaries can be acceptable because it is estimated with less variance. Despite the many assumptions and limitations, both LDA and QDA are widely used and perform well on on a diverse set of classification tasks [Hastie et al., 2009], even when the classes are not normally distributed.

2.3.3 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a commonly used non-parametric classification method. To predict the class of a new observation, a distance matrix is constructed between all observations and the K nearest labelled observations to the new observation are considered. The new observation is then assigned the class label that the majority of its neighbors share. In case of only two classes, ties in class assignments are avoided by using odd values of K.

In the event of a tie, a class can be chosen at random. Various distance metrics may be used but it is common to use Euclidean distance to determine the closest training points, though it is advisable to scale variables so that one direction does not dominate the classification.

KNN is sensitive to the local sturcture of the data. As K increases, the variability of the classification tends to decrease at the expense of increased bias.

2.3.4 Random Forests

Random forests (Brieman, 2001) are one of the most successful general-purpose modern algorithms (Biau and Scornet, 2016). They are an ensemble learning method that can be applied to a wide range of tasks, namely classification and regression. A random forest is created by building multiple decision trees, where randomness is introduced during the construction of each tree. Predictions are made by classifying a new observation to the mode of the multiple decisions tree classifications. Random forests often make accurate and robust

predictions, even for very high-dimensional problems (Biau, 2012). See **(Appendix X)** for an explanation of the random forests algorithm.

- **State why random forests are good predictors**

2.4 Accuracy Metrics

These are the default metrics used to evaluate algorithms on binary and multi-class classification datasets in caret.

2.4.1 Accuracy, Sensitivity, and Specificity

Accuracy is the percentage of correctly classifies instances out of all instances. It is more useful on a binary classification than multi-class classification problems because it can be less clear exactly how the accuracy breaks down across those classes (e.g. you need to go deeper with a confusion matrix). Learn more about Accuracy here.

Don't use accuracy (or error rate) to evaluate your classifier! There are two significant problems with it. Accuracy applies a naive 0.50 threshold to decide between classes, and this is usually wrong when the classes are imbalanced. Second, classification accuracy is based on a simple count of the errors, and you should know more than this. You should know which classes are being confused and where (top end of scores, bottom end, throughout?)

Table 1: Confusion matrix for two classes.

		Observed	
		N	Y
Predicted	N	a	b
	Y	c	d

For the two class confusion matrix in Table 1 accuracy metrics are defined as:

$$\text{sensitivity} = \frac{a}{a+c}$$
$$\text{specificity} = \frac{d}{b+d}$$
$$\text{accuracy} = \frac{a+d}{a+b+c+d}$$

where sensitivity is a measure of how accurately non-survival is predicted, specificity is a measure of how accurately survival is predicted, and accuracy is a measure of how well both survival and non-survival are predicted. While sensitivity and specificity state the accuracy each class prediction, accuracy is a poor measure for model performance in an imbalanced dataset. On the ARDS datasets, for example, if `ECMO_Survival` is predicted to be "Y" for

all cases, then the accuracy is 75% but the prediction is no better than the baseline likelihood of the class percentages.

2.4.2 Cohen's Kappa

Kappa or Cohen's Kappa is like classification accuracy, except that it is normalized at the baseline of random chance on your dataset. It is a more useful measure to use on problems that have an imbalance in the classes. Let p_o be the accuracy, the relative observed agreement between observed and predicted classes and let p_e be the probability of chance agreement based on the class probabilities. Cohen's Kappa is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

If all the observations are predicted correctly then $\kappa = 1$. If the observations are predicted no better than expected by the class probabilities, p_e then $\kappa = 0$. If all the observations are predicted incorrectly, then $\kappa = -1$. A positive κ indicates that the model predicts better than would be expected by chance whereas a negative κ indicates that the model predicts worse than would be expected by chance.

$$p_o = \frac{a + d}{a + b + c + d}$$
$$p_e = p_{o,Y} + p_{o,N}$$

$$p_{o,Y} = \frac{a + d}{a + b + c + d} \cdot \frac{a + c}{a + b + c + d}$$

$$p_{o,N} = \frac{c + d}{a + b + c + d} \cdot \frac{b + d}{a + b + c + d}$$

2.5 Missing Data

Missing data is a common problem that must be dealt with in machine learning, statistics, and medicine. Understanding the missing mechanism for the missing observations is important in the analysis. [RUBIN, 1976] defined three types of missing data mechanisms: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). The data are said to be missing completely at random (MCAR) if the probability of being missing is the same for all cases. This implies the causes of the missing data are unrelated to the data itself. While MCAR is conveinient because it allows many complexities that arise because data are missing to be ignored, it is typically an unrealistic assumption [van Buuren, 2012]. The data is said to be MAR if the probability of being missing is the same only within groups defined by the observed data. MAR is a more general and more

realistic assumption than MCAR. If neither MCAR nor MAR applies, then the probability of being missing depends on an unknown mechanism and said to be MNAR. Most simple approaches to dealing with missing data are only valid under MCAR assumption. Modern methods to dealing with missing data begin from the MAR assumption.

2.6 Imputation Methods

2.6.1 Complete Case Analysis

Complete case canalysis is a convenient method for handling missing data and is the default method in many statistical packages. If there is a missing value in an observation, it is dropped from the analysis. This is often a poor appraoch as complete cases analysis assumes MCAR. In sparse datasets a complete case analysis can cause an analysis to be underpowered and if MCAR does not hold, can severely bias estimates of means, regression coefficients, and correlations [van Buuren, 2012].

The ARDS dataset considered in this paper has 268/450 observations with missing data.

2.6.2 Mean Imputation

Another common method for handling missing data is mean imputation; the missing value is replaced by the mean of the covariate or the mode for categorical data. Mean imputation is a simple and attractive solution because it retains more of the data. Mean imputation distorts the distribution of the variables toward the mean. If MCAR assumtion does not hold it will underestimate the variance and produce biased estimates other than the mean [van Buuren, 2012]. [van Buuren, 2012] suggests mean imputation should only be used only when there are few missing values, and should be generally avoided.

- Can further distort the distribution if the variable is not normally distributed.
- Mean imputatio is implemented in this paper because although it is often a poor method of choice for imputing missing values, it is commonly done.
- Variables are transformed via Yeo-Johnson Transformation to reduce distributional distortions.

2.6.3 Multiple Imputation

The aim when imputing data is to recreate the dataset and recreate the missing data as if it were never missing. Multiple imputation is a method that accounts for the uncertainty in the imputed values. The analysis begins with the observed, incomplete dataset. The dataset is imputed multiple times to create $m > 1$ complete datasets. The imputed values are drawn from a distribution specifically modeled for each missing entry. The m datasets are analyzed

using the same method that would have been used had the data been complete. The results will differ because of the variation in the input data caused by the uncertainty in the imputed values.

Multiple imputation can handle data that is both MAR and MNAR.

There is uncertainty as to the true value of the unseen data, and that uncertainty should be included in the analysis. Multiple imputation is a method created by Donald Rubin wherein multiple datasets are imputed, the analysis is conducted on each dataset, and the results are pooled using "Rubin's Rules" [RUBIN, 1976].

- Details of the **MICE** algorithm can be found in Appendix B.

2.6.4 Fully Conditional Specification

2.6.5 Predictive Mean Matching

Predictive Mean Matching (PMM) is a semi-parametric imputation approach. It is similar to the regression method except that for each missing value, it fills in a value randomly from among the a observed donor values from an observation whose regression-predicted values are closest to the regression-predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996). The PMM method ensures that imputed values are plausible; it might be more appropriate than the regression method (which assumes a joint multivariate normal distribution) if the normality assumption is violated (Horton and Lipsitz 2001, p. 246). PMM is fairly robust to transformations of the target variables [van Buuren, 2012], yielding similar results for a Yeo-Johnson transformation or no transformation.

2.7 Ensemble Multiple Imputation

Two approaches have been proposed for pooling results from several SVMs **(Belache et al. 2014)** and Cox regression **(Zavrakidis 2017)** from multiply imputed datasets. The method is to concatenate the m imputed datasets and fit a classifier, and optimize, to the resulting set; this accounts for the variability of the parameter estimates as well as the variability of the training observations in relation to the imputed values **(Belache et al. 2014)**. The second proceedure fits separate classifiers to each imputed data set and get the pooled (i.e. avereaged) performance of the m classifiers. Results from both studies either show similar results between appraoches **(Zavrakidis 2017)** or slightly better performance with the first approach **(Belache et al. 2014)**. For simplicity and the sake of computational costs, this paper, only considers the first approach.

The steps in the ensemble approach for multiply imputed data in k-fold cross-validation are as follows:

1. Randomly partition the training data into k folds

2. Define the k^{th} as the test set and the remaining $k-1$ folds as the training set
3. Impute the training set m times, with the response variable `ECMO_Survival` included, to create m imputed training sets
4. Concatenate the m imputed training sets into one extended training set
5. A model is fitted to the extended training set
6. The test set is concatenated with the extended training set
7. Impute the combined test and extended training set, with the response variable `ECMO_Survival` excluded, to create m imputed combined test and extended training sets
8. Extract the m test sets
9. Make m predictions on the m imputed test sets
10. Take the majority vote of the m predictions as the prediction for the fitted model
11. Validate the prediction against the test set by calculating Cohen's Kappa (note there are no missing values for the response variable in the data)
12. Repeat steps 2-11 k times and validate the fitted model on each training set against the test set for each fold
13. Average the k calculated Cohen's Kappas as the estimated in-sample accuracy metric

2.7.1 Number of Imputations

Flexible Imputation Book
"The classic advice is to use a low number of imputation, somewhere between 3 and 5 for moderate amounts of missing information. Several authors investigated the influence of m on various aspects of the results. The picture emerging from this work is that it is often beneficial to set m higher, somewhere in the range of 20-100 imputations.

"Rubin's Rules" provide a simple method for pooling parameters estimates from multiple imputation for linear and generalized linear models but to the author's knowledge, there has been insufficient work on estimating the required number of imputations for estimating posterior probabilities in classification problems. In his book, **(Van Buuren)** states

> "Theoretically it is always better to use higher m, but this involves more computation and storage. Setting m very high (say m=200) may be useful for low-level estimands that are very uncertain, and for which we want to approximate the full distribution, or for parameters that are notoriously different to estimates, like variance components. On the other hand, setting m high may not be worth the extra wait if the primary interest is on the point estimates (and not on standard errors, p-values, and so on). In that case using m=5-20 will be enough under moderate missingness."

2.8 Voting

There has been sufficient exploration into pooling of posterior probabilities resulting from classification problems **(Citation 1) (Citation 2) Citation 3)**. But not all statistical

methods produce posterior probabilities and the comparison of pooled models from multiple imputation is an area ripe for more analysis. Indeed, others have pooled predictions from various machine learning methods by taking the majority vote **(Zavrakidis) (Citation 2)**, and comparing prediction accuracy.

2.8.1 Majority Vote

The combination can be implemented using a variety of strategies, among which majority vote is by far the simplest, yet it has been found to be just as effective as more complicated schemes [Lam and Suen, 1995].

(Alexandre et al. 2001) There has been some interest on the comparative performance of the sum and product rules (or the arithmetic and geometric means) (Kittler et al., 1996; Tax et al., 1997; Kittler et al., 1998). The arithmetic mean is one of the most frequently used combination rules since it is easy to implement and normally produces good results.

In (Kittler et al., 1998), the authors show that for combination rules based on the sum, such as the arithmetic mean, and for the case of classifiers working in different feature spaces, the arithmetic mean is less sensitive to errors than geometric mean.

In fact (Alexandre et al. 2001) show that for classification problems with two classes, that give estimates of the a posteriori probabilities that sum to one the combination rules arithmetic mean (or the sum) and the geometric mean (or the product) are equivalent.

2.9 Feature Selection

- PCA for feature selection
- Quick description of PCA
- Maximizing the Area Under the ROC Curve

3 Results

3.1 Exploratory Data Analysis

To get an idea of the distribution of the data, the following summary statistics were obtained for the categorical variable `ECMO_Survival` (Table 2) and for the continuous variables (Table 3).

Table 2: Numbers of survivors and nonsurvivors of ECMO treatment.

ECMO_Survival	n	Percent %
N	109	24.22
Y	341	75.78

Table 2 shows that out of the 450 individuals, only 75.78% of the individuals in the study sample survived ECMO treatment (341 survived vs 109 did not survive).

Table 3: Number of males and females.

Gender	n	Percent %
m	305	67.78
w	145	32.22

Table 3 shows that out of the 450 individuals, only 67.78% of the individuals in the study sample are male (305 male vs 145 female).

Table 4: Number of each disease type indication.

Indication	n	Percent %
1	66	14.67
2	181	40.22
3	31	6.89
4	28	6.22
5	71	15.78
6	12	2.67
7	61	13.56

Table 4 shows the distribution of each disease type indication.

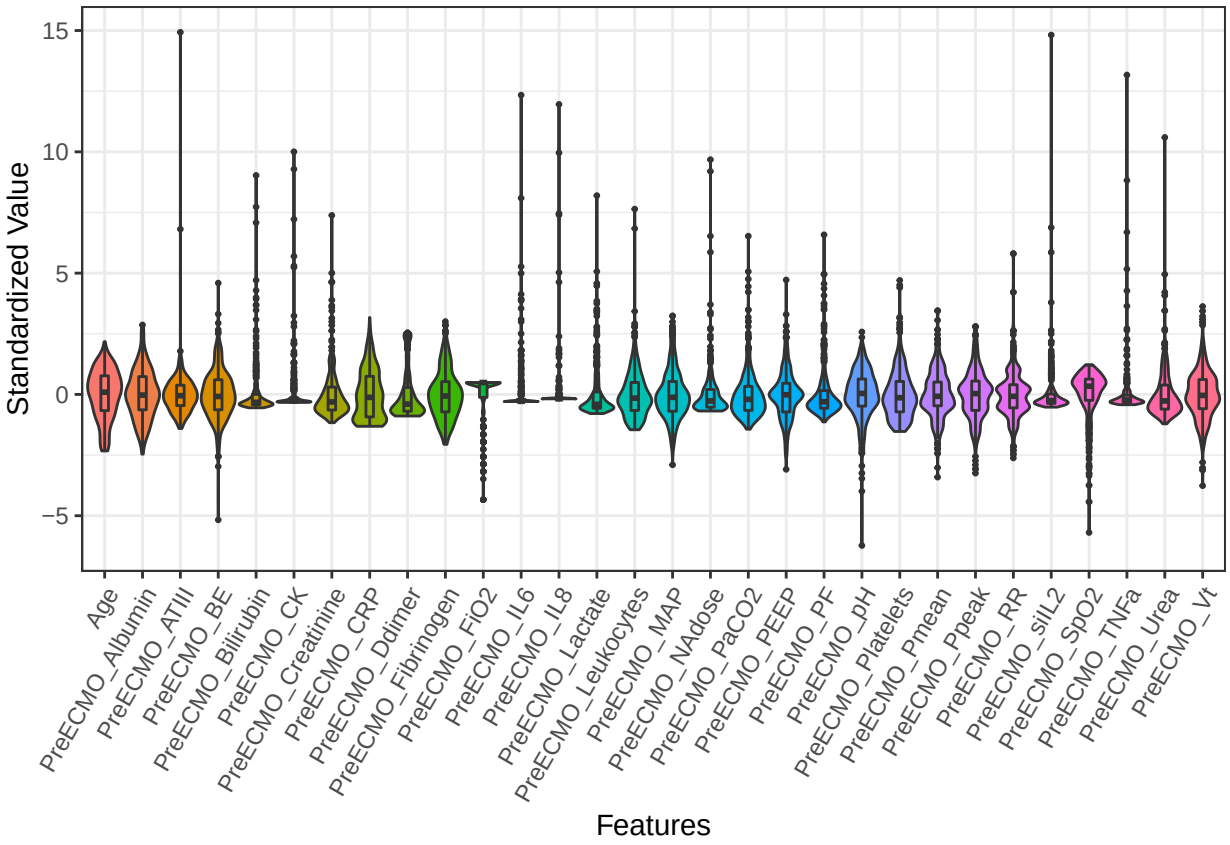

Figure 2: Violin plot of continuous variables.

3.2 Missing Data Patterns

Before imputation, and indeed multiple imputation, it is important to inspect the missingness patterns in the data and check assumptions. Figure 3 shows the missingness patterns in the dataset, where a black bar represents a missing value. Table **??** provides some measures about variable dependence in the dataset. The first row shows the probability of observed values for each variable. The following are coefficients that give insight into how the variables are connected in terms of missingness. **Influx** is the ratio of the number of variables pairs $(Y_j,\ Y_k)$ with Y_j missing and Y_k observed, divided by the total number of observed data. For a variable that is entirely missing, influx is 1, and 0 for if the variable is complete. **Outflux** is defined in the opposit manner, by dividing the number of pairs $(Y_j,\ Y_k)$ with Y_j observed and Y_k missing, by the total number of complete cells. For a completely observed variable, outflux will have a value of 1 and 0 if completely missing. Outflux gives an indication of how useful the variable will be for imputing other variables in the dataset, while influx is an indicator for how easily the variable can be imputed. Table 5 shows that all variables will be useful during impuation except `PreECMO_Albumin`. A high outflux variable might turn out to be useless for the imputation procedure if it is unrelated to the incomplete variables, while the usefulness of a highly predictive variables is severely limited by a low outflux value (Van Buuren 2012). **Mention 3)**

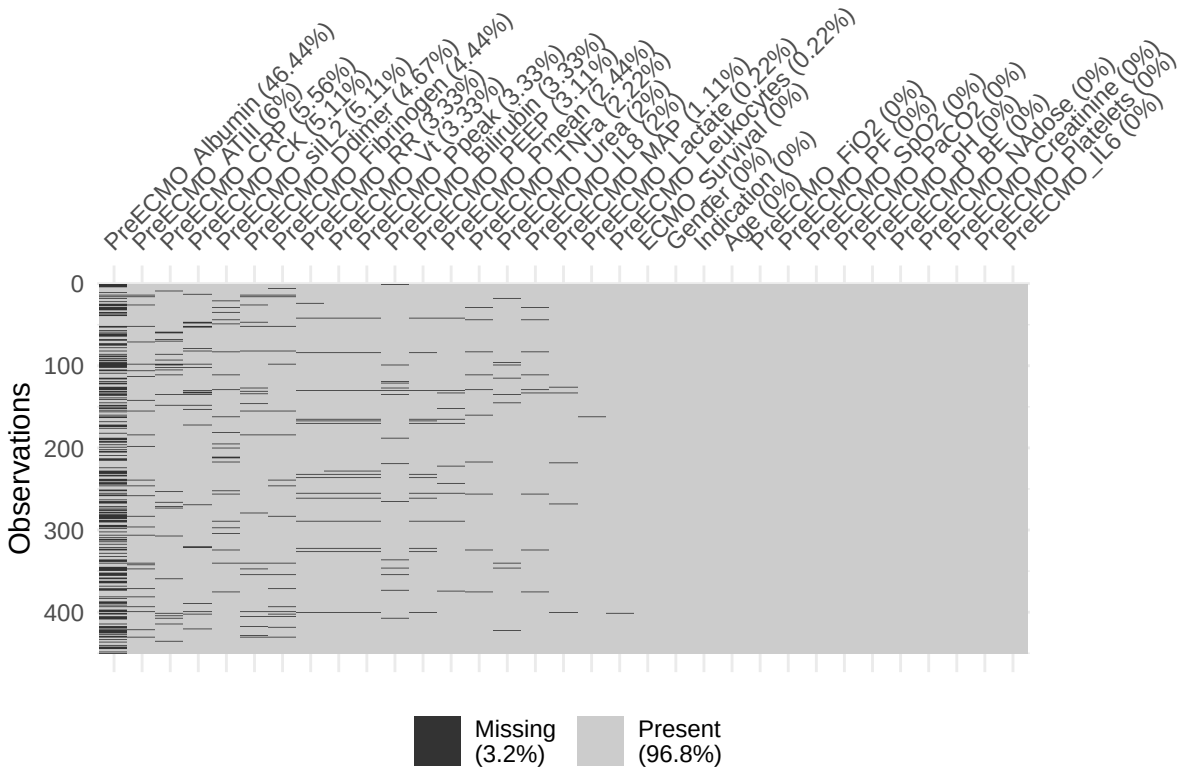

Figure 3: Visual representation of missing observations in the ARDS dataset.

- It can be difficult or impossible to determine if the data are MCAR. Figure 3 shows that many missing values occur in observations with other missing values. Missing values could be conditionally dependent on other variables, in which case the data would be MAR. The missing values could also be due to some unknown mechanism at the time of recording (*i.e.* a failure of the measurement device) that happens to effect multiple readings (the biomarkers are measured from blood samples and measurements are likely done in batches). In this case the data would be MCAR. Without more information, this analysis assumes the data is MCAR.

3.3 Experiments and Results

The experimentation phase of this study involved three methods for handling missing data: (a) complete case analysis with the variable `PreECMO_Albumin` dropped from the analysis due to 46.44% missingness, (b) mean imputation on variables with missing values, (c) imputation via the MICE algorithm implemented with PMM.

Table 6 shows the averaged Kappa from each analysis in 10-fold cross-validation. In complete case analysis and mean imputaion, LDA is the highest performer. While for predictive mean-matching with $m = 9$ logistic regression has the higest averaged Kappa.

Table 5: Missing pattern statistics for variables in dataset.

	Proportion	Influx	Outflux
ECMO_Survival	1.00	0.00	1.00
Gender	1.00	0.00	1.00
Indication	1.00	0.00	1.00
Age	1.00	0.00	1.00
PreECMO_RR	0.97	0.03	0.85
PreECMO_Vt	0.97	0.03	0.85
PreECMO_FiO2	1.00	0.00	1.00
PreECMO_Ppeak	0.97	0.03	0.85
PreECMO_Pmean	0.98	0.02	0.90
PreECMO_PEEP	0.97	0.03	0.85
PreECMO_PF	1.00	0.00	1.00
PreECMO_SpO2	1.00	0.00	1.00
PreECMO_PaCO2	1.00	0.00	1.00
PreECMO_pH	1.00	0.00	1.00
PreECMO_BE	1.00	0.00	1.00
PreECMO_Lactate	1.00	0.00	0.99
PreECMO_NAdose	1.00	0.00	1.00
PreECMO_MAP	0.99	0.01	0.97
PreECMO_Creatinine	1.00	0.00	1.00
PreECMO_Urea	0.98	0.02	0.94
PreECMO_CK	0.95	0.05	0.87
PreECMO_Bilirubin	0.97	0.03	0.91
PreECMO_Albumin	**0.54**	**0.46**	**0.26**
PreECMO_CRP	0.94	0.05	0.88
PreECMO_Fibrinogen	0.96	0.04	0.85
PreECMO_Ddimer	0.95	0.04	0.86
PreECMO_ATIII	0.94	0.06	0.84
PreECMO_Leukocytes	1.00	0.00	0.99
PreECMO_Platelets	1.00	0.00	1.00
PreECMO_TNFa	0.98	0.02	0.93
PreECMO_IL6	1.00	0.00	1.00
PreECMO_IL8	0.98	0.02	0.93
PreECMO_siIL2	0.95	0.05	0.87

Table 6: Averaged Cohen's Kappa for each model fitted in cross-validation. The tuned parameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=13, respectively.

	Logit	LDA	QDA	KNN	RF
Complete Case	0.139	0.205	0.038	0.053	0.035
Mean	0.191	0.220	0.040	0.136	0.085
PMM	0.179	0.124	0.106	0.088	0.136

3.3.1 Validation on Test Set

Using the parameters values learned obtained from 10-fold cross-validation in Table 6, models were fit to the full training set and validated against the test set. K-nearest neighbors and random forests were fit with parameters $K = 5$ and $mtry = 13$, respectively.

Table 7: Complete case analysis accuracy metrics. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=11, respectively.

	Sensitivity	Specificity	Accuracy	Kappa
Logit	0.20	0.814	0.658	0.015
LDA	0.20	0.847	0.684	0.054
QDA	0.00	0.966	0.722	-0.048
KNN	**0.30**	**0.847**	**0.709**	**0.161**
RF	0.05	0.966	0.734	0.022

Table 8: Mean imputation accuracy metrics (m=1). The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=11, respectively.

	Sensitivity	Specificity	Accuracy	Kappa
Logit	0.222	0.894	0.732	0.137
LDA	0.148	0.894	0.714	0.051
QDA	0.111	0.882	0.696	-0.008
KNN	0.222	0.824	0.679	0.050
RF	**0.185**	**0.965**	**0.777**	**0.197**

Table 9: MICE via predictive mean matching accuracy metrics (m=91). The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=13, respectively.

	Sensitivity	Specificity	Accuracy	Kappa
Logit	**0.222**	**0.906**	**0.741**	**0.153**
LDA	0.148	0.906	0.723	0.067
QDA	0.111	0.882	0.696	-0.008
KNN	0.222	0.847	0.696	0.077
RF	0.148	0.941	0.750	0.116

Table 10: MICE via predictive mean matching accuracy metrics (m=99). The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=13 and mtry=15, respectively.

	Sensitivity	Specificity	Accuracy	Kappa
Logit	**0.333**	**0.906**	**0.768**	**0.274**
LDA	0.185	0.906	0.732	0.111
QDA	0.111	0.894	0.705	0.006
KNN	0.185	0.882	0.714	0.080
RF	0.185	0.929	0.750	0.144

4 Discussion

4.1 Model Performance

Logistic Regression
For complete-case analysis, mean imputation, and predictive mean-matching, logistic regression does not meet the "one in ten rule", a rule of thumb stating that a logistic regression models give stable estimates for the covariates if there are at least 10 observations of the least frequent class per covariate.

LDA
Can perform better than logistic regression when the covariates are normally distributed **(CITATION)**, which they are in this case after Yeo-Johnson transformation.

QDA

K-Nearest Neighbors

Random Forests Fails

- Sparsity - When the data are very sparse, it's very plausible that for some node, the bootstrapped sample and the random subset of features will collaborate to produce an invariant feature space. There's no productive split to be had, so it's unlikely that the children of this node will be at all helpful.
- One surprising consequence is that trees that work well for nearest-neighbor search problems can be bad candidates for forests without sufficient subsampling, due toa lack of diversity. **(Tang et al. 2018)**
- Data are not axis-aligned - Suppose that there is a diagonal decision boundary in the space of two features, x_1 or x_2. Even if this is the only relevant dimension to your data, it will take an ordinary random forest model many splits to describes that diagonal boundary. This is because each split is oriented perpendicular to the axis of either x_1 or x_2.
- XGBoost, Rotation forest (PCA rotation) may do better

4.2 Important Features for Prediction

- Mention poor model performance
- Correlation heatmap
- Results from PCA

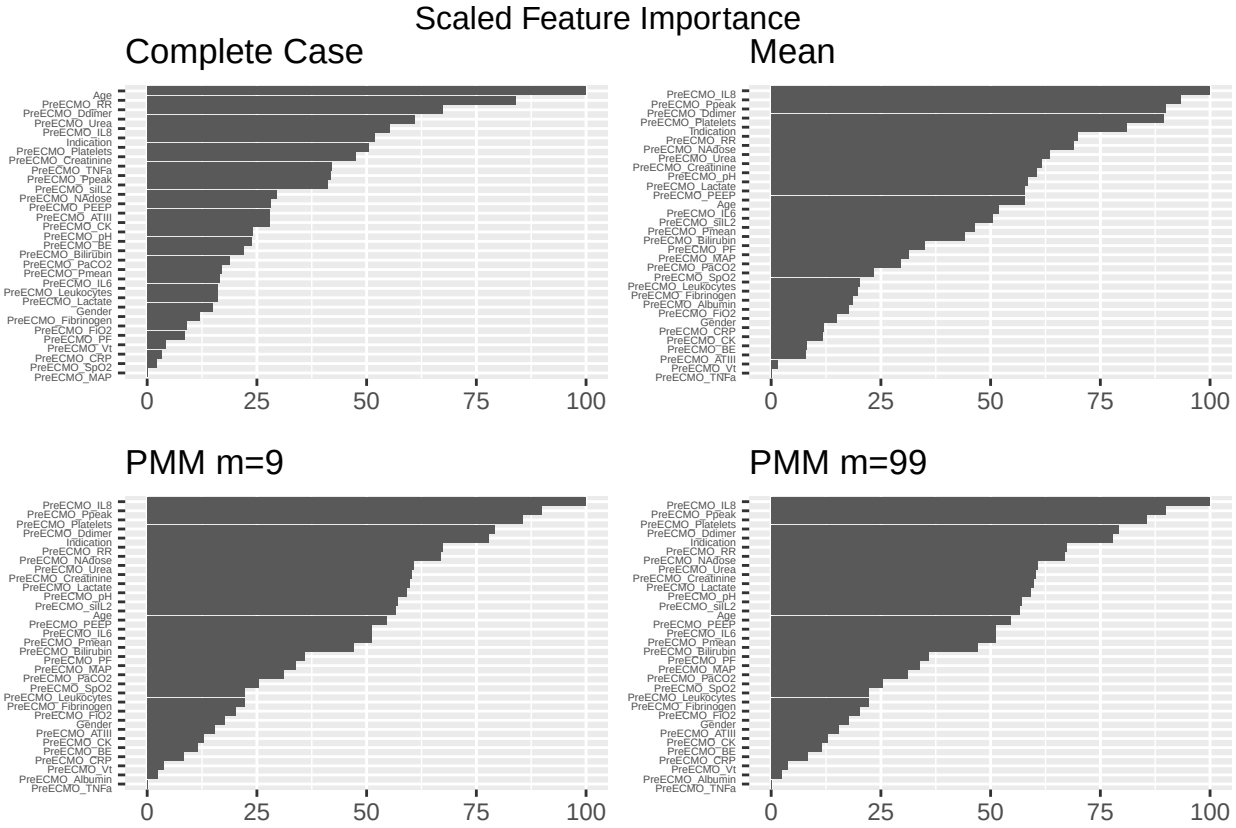

Figure 4: Ordered feature importance from LDA model

- **Refer to Figure ?? in Appendix A for feature importance plots for the logit model**

4.3 Conclusion

- Summary of proceedure
- Summary of results
- Possible improvements and future work

5 Appendices

5.1 A. Additional Exploratory Data Analysis

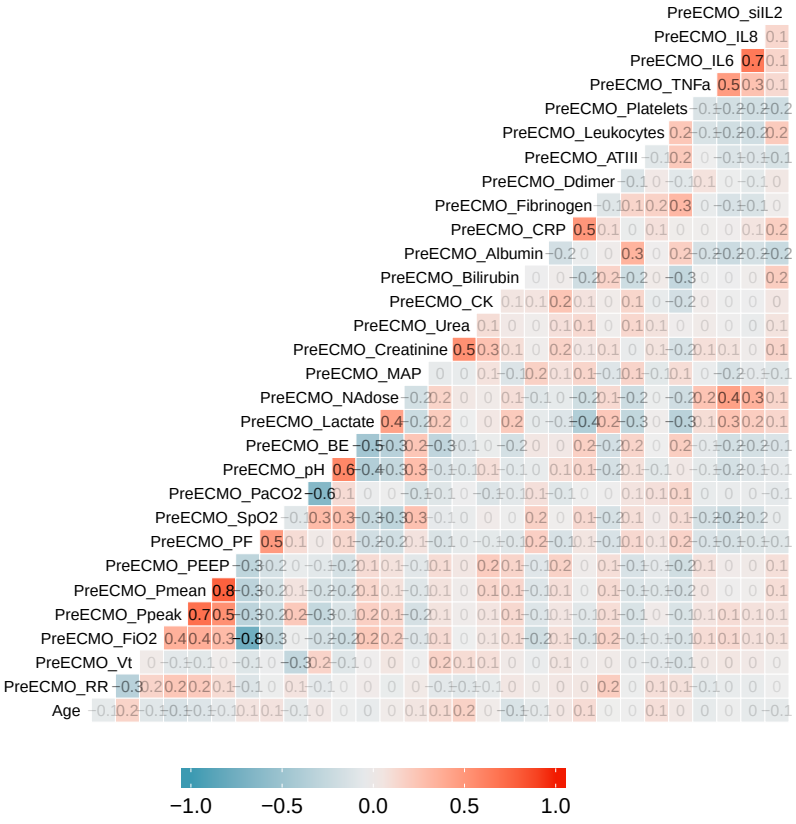

Figure 5: Heatmap of standardized and transformed variables.

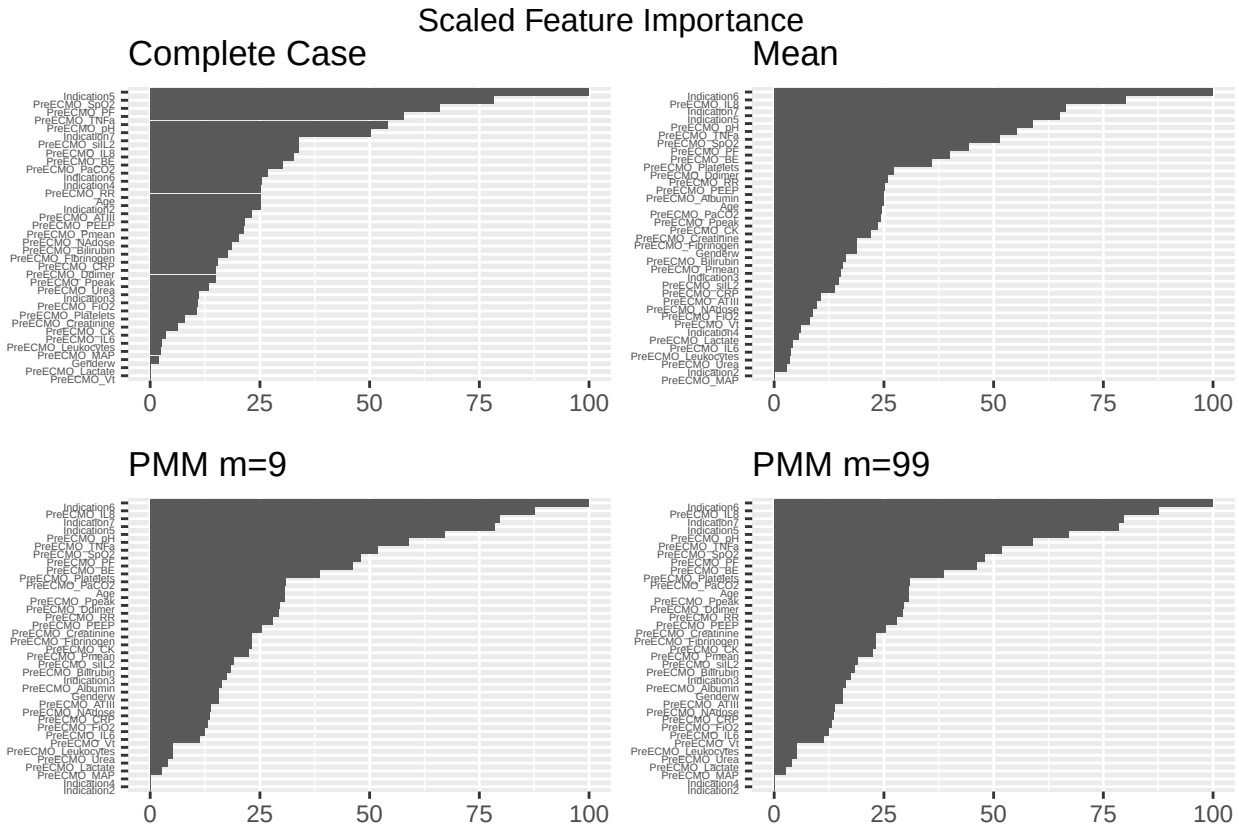

Figure 6: Ordered feature importance from Logit model

5.2 B. Algorithms

5.2.1 Random Forests Algorithm

The random forests algorithm depicted is adapted from [Hastie et al., 2009].

1. For ($b = 1$ to B):
   (a) Draw a bootstrap sample $\mathbf{Z}*$ of the size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repreating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select $mtry$ variables at random from the p covariates.
      ii. Pick the best covariate/split-point among the $mtry$.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_B\}_1^B$

Let $\hat{Y}_b(x)$ be the class prediction of the b^{th} random-forest tree. Then a new observation, x, is classified as:

$$\hat{Y}_{\text{rf}}^B(x) = \text{majority vote } \left\{\hat{Y}_b(x)\right\}_1^B$$

Algorithm 1: Random Forest Classifier

5.2.2 MICE Algorithm

The MICE algorithm is adapted from [van Buuren, 2012].

1. Specify an imputation model $P(Y_j^{\text{mis}}|Y_j^{\text{obs}}, Y_{-j}, R)$ for variable Y_j with $j = 1, ..., p$
2. For each j, fill in starting imputation Y_j^0 by random draws from Y_j^{obs}
3. Repeat for $t = 1, ..., T$:
4. Repeat for $j = 1, ..., p$:
5. Define $Y_{-j}^t = (Y_1^t, ..., T_{j-1}^t, Y_{j+1}^{t-1}, ..., Y_p^{t-1})$ as the currently complete data except Y_j
6. Draw $\phi_j^t \sim P(\phi_j^t|Y_j^{\text{obs}}, Y_{-j}^t, R)$.
7. Draw imputations from $Y_j^t \sim P(Y_j^{\text{mis}}|Y_j^{\text{obs}}, Y_{-j}^t, R, \phi_j^t)$.
8. End repeat j.
9. End repeat t.

Algorithm 2: Multiple Imputation via Chained Equations

5.3 C. Additional Missing Data Diagnostics

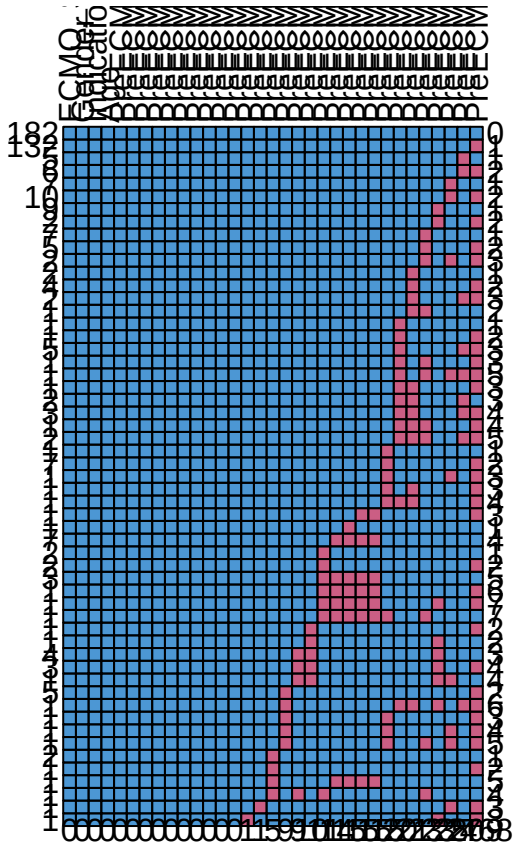

Figure 7: Missing data patterns. Each row corresponds to a missing data pattern (1=observed, 0=missing). Rows and columns are sorted in increasing amounts of missing information. The last column and row contain row and column counts, respectively.

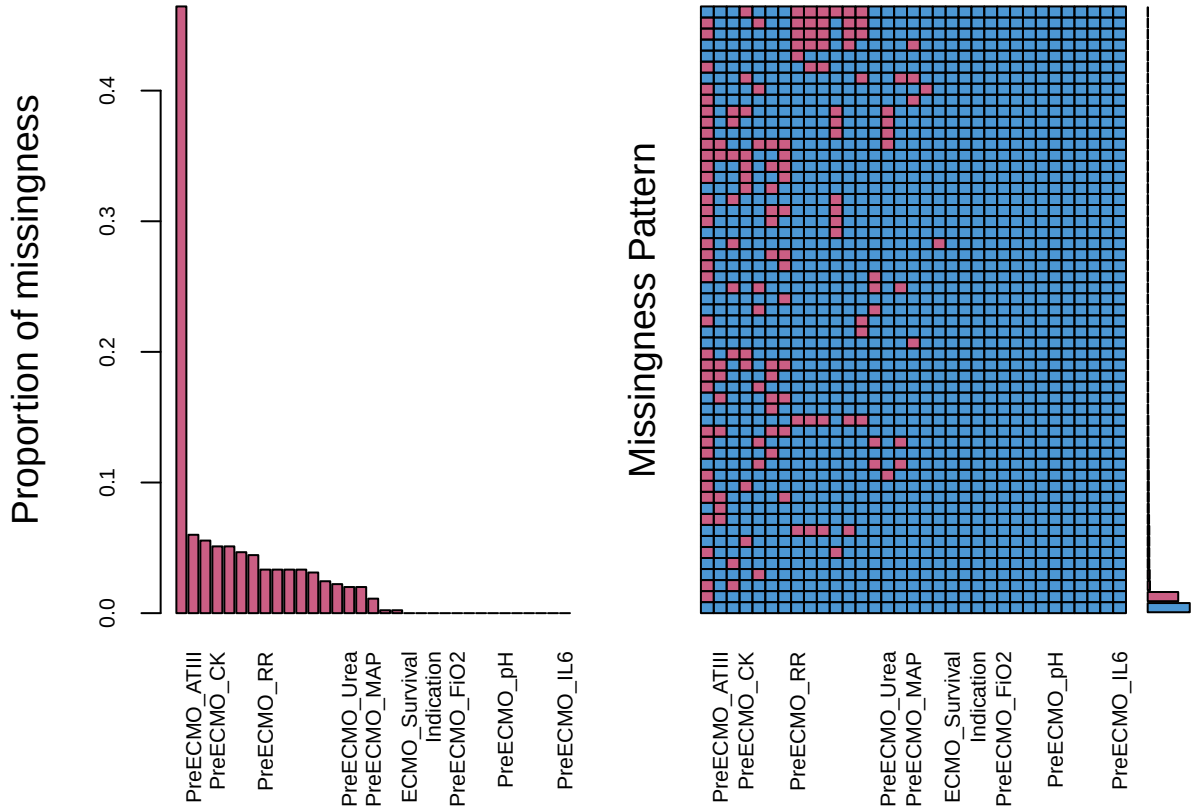

Figure 8: Missing data patterns. Each row corresponds to a missing data pattern (1=observed, 0=missing). Rows and columns are sorted in increasing amounts of missing information. The last column and row contain row and column counts, respectively.

```
Variables sorted by number of missings:
          Variable       Count
   PreECMO_Albumin 0.464444444
     PreECMO_ATIII 0.060000000
       PreECMO_CRP 0.055555556
        PreECMO_CK 0.051111111
     PreECMO_siIL2 0.051111111
    PreECMO_Ddimer 0.046666667
PreECMO_Fibrinogen 0.044444444
        PreECMO_RR 0.033333333
        PreECMO_Vt 0.033333333
     PreECMO_Ppeak 0.033333333
 PreECMO_Bilirubin 0.033333333
      PreECMO_PEEP 0.031111111
     PreECMO_Pmean 0.024444444
      PreECMO_TNFa 0.022222222
      PreECMO_Urea 0.020000000
```

```
       PreECMO_IL8 0.020000000
       PreECMO_MAP 0.011111111
   PreECMO_Lactate 0.002222222
PreECMO_Leukocytes 0.002222222
     ECMO_Survival 0.000000000
            Gender 0.000000000
         Indication 0.000000000
               Age 0.000000000
       PreECMO_FiO2 0.000000000
         PreECMO_PF 0.000000000
       PreECMO_SpO2 0.000000000
      PreECMO_PaCO2 0.000000000
         PreECMO_pH 0.000000000
         PreECMO_BE 0.000000000
    PreECMO_NAdose 0.000000000
PreECMO_Creatinine 0.000000000
 PreECMO_Platelets 0.000000000
       PreECMO_IL6 0.000000000
```

5.3.1 Visual Insepction of Imputations

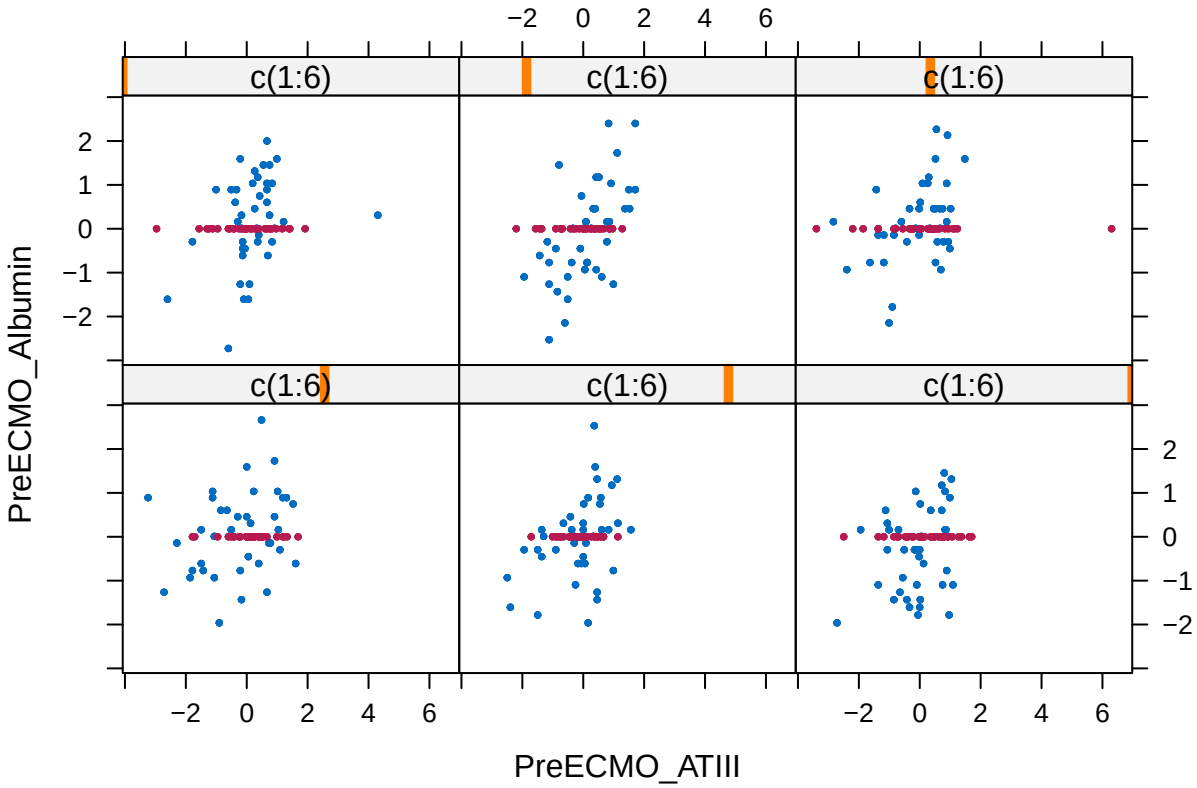

Figure 9: Scatterplot of each imputed dataset

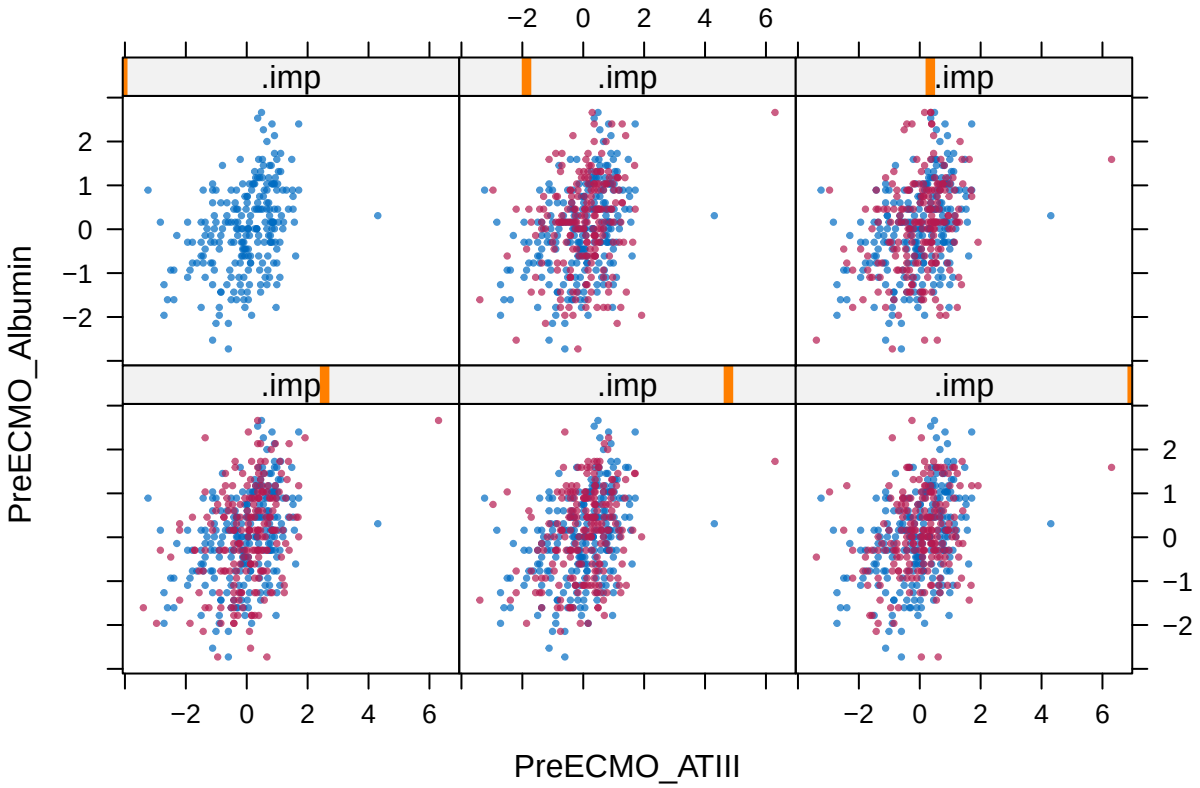

Figure 10: Scatterplot of each imputed dataset

- **xyplot checking distributions of original and imputed data for MEAN imputation**
- **xyplot checking distributions of original and imputed data for PMM imputation**
- **density plot of original and imputed data for MEAN imputation**
- **density plot of original and imputed data for PMM imputation**

This plot compares the density of observed data with the ones of imputed data. We expect them to be similar (though not identical) under MAR assumption.

5.3.2 Convergence Monitoring

- **Plot of convergence**

5.4 D. R Code

The code organization is described in Figure 11. `libraries.R` contains all the libraries used in the analysis. `functions.R` contains functions used in `training.R` and `model-evaluation.R`. The ensemble cross-validation algorithm is done in the `crossValidation()` function. The data is initially cleaned and split into test and training sets in `preprocess.R`. The cleaned datasets are saved to `processed-data.RData` for use in `training.R` and in creating tables and figures in the thesis rmarkdown. The training data is loaded into `training.R` where each of the five classification methods are trained via ensemble cross-validation. This is done for the four imputation methods: complete case analysis, mean imputation, MICE using PMM for $m = 9$, and MICE using PMM for $m = 99$ imputed datasets. The trained models for each imputation method are saved into separate `trained-models.RData`. The methods are then then fit to the full training set in `model-evaluation.R` using the trained parameters found in `training.R`. The final fitted models are evaluated on the test set and the fitted models and performance metrics are saved to `metrics.RData`.

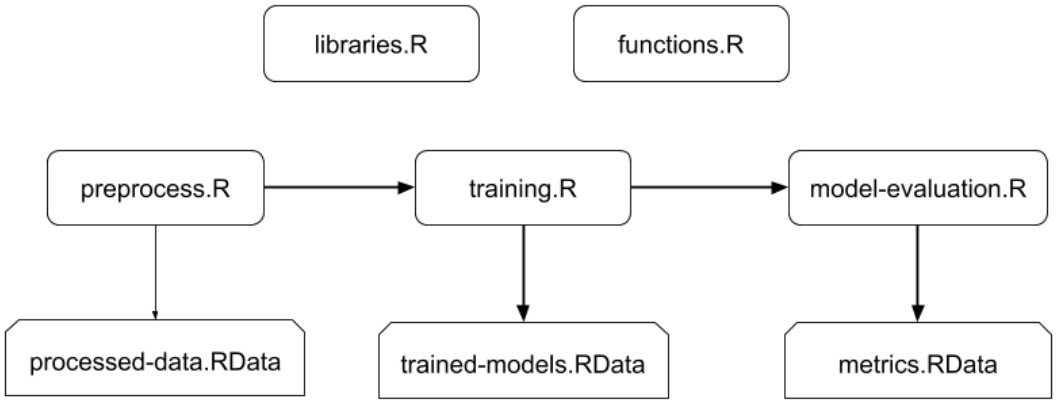

Figure 11: Flowchart of code structure.

References

T. M. Cover. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, EC-14 (3):326–334, June 1965. ISSN 0367-7508. doi: 10.1109/PGEC.1965.264137.

R. A. FISHER. THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. *Annals of Eugenics*, 7(2):179–188, 1936. doi: 10.1111/j.1469-1809.1936.tb02137.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009. ISBN 978-0-387-84884-6. URL https://books.google.co.uk/books?id=eBSgoAEACAAJ.

Louisa Lam and Ching Y. Suen. Optimal combinations of pattern classifiers. *Pattern Recognition Letters*, 16(9):945–954, September 1995. ISSN 0167-8655. doi: 10.1016/0167-8655(95)00050-Q. URL http://www.sciencedirect.com/science/article/pii/016786559500050Q.

DONALD B. RUBIN. Inference and missing data. *Biometrika*, 63(3):581–592, December 1976. ISSN 0006-3444. doi: 10.1093/biomet/63.3.581. URL https://doi.org/10.1093/biomet/63.3.581.

Stef van Buuren. *Flexible Imputation of Missing Data*. Chapman & Hall, London, second edition edition, 2012.