Ensemble Multiple Imputation: An Ensemble Approach to using Multiple Imputation and Cross-Validation for Classification

Robert Edwards

2416963E

MASTERS THESIS

Biostatistics

Acknowledgements

To my peers, alone we are biased, together we are unbiased.

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

To my friends, for your unbiased indulgence in my regressive statistical puns.

To my non-Gaussian friends, you increased the variance on the objectively Normal days.

Contents

1 Introduction

Prediction in medical data can often be difficult due to a low number of observations and poor predictive covariates. If the response class distributions are imbalanced, then prediction becomes even more difficult. Some of these issues arise from the experimental design of the study or for reasons beyond the control of the researcher but little can be rememdied post-hoc. Missing values in the data complicate the 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 lost and these approaches bias estimates of means, regression coefficients, and correlations (1, 2). Several statistical methods have been proposed for handling missing data (2); simple proceedures include complete case analysis (CC) wherein all observations with missing data are excluded, single imputation methods that simply use the mean to replace the missing value, and more principled methods such as multiple imputation (MI). Multiple imputation is a widely used flexible method in datasets with missing values. MI is commonly used in medical studies (3–5) but how it is implemented is often not stated (6, 7).

1.1 Study Population & Data Description

This paper investigates predicting survival of patients diagnosed with Acute Respiratory Distress Syndrom (ARDS) after treatment with Extracorporeal Membrane Oxygenation (ECMO). ARDS is a common and often fatal cause of respiratory failure among patients who are critically ill, with an estimated global prevalence of 10% and a mortality of 25-40% (8–11). ARDS has many disease paths and is characterized by rapid onset of widespread inflammation in the lungs (10). ECMO is a complicated and invasive proceedure intended as a supportive care treatment, yet is quickly being adopted as a primary treatment for severe ARDS patients (12). ARDS patients who have undergone ECMO treatment show lower morbidity and mortality rates than those not considered for ECMO treatment (13). Follow-up studies (14) have reported favorable outcome in younger ARDS patients treated with ECMO with fewer comorbid conditions. Identifying the mortality risk of patients before treatment is of growing interest. Clinical studies in the U.S. (15) and U.K. (16) have identified two ARDS subphenotypes have been identified with distinct clinical and biological features that are thought to support predictive strategies.

The dataset is composed of 450 observations on patients with ARDS who underwent ECMO treatment. The response variable, `ECMO_Survival`, is a binary categorical variable for survival indication with levels "Y" and "N". There are 33 covariates included in the analysis, two of which are categorical, and 31 continuous. The categorical variable `Gender` has two levels, "m" and "f", and `Indication` is a seven level nominal categorical indicator of disease type. The continuous variable `Age` is also included in the analysis with a minimum age of 18 and a maximum of 83 with a median age of 53. The remaining variables are biomedical markers from hospital measurements.

1.2 Aims of the Proposed Research

The main questions of interest investigated in this paper 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?

To further the goals of this paper, multiple imputation is investigated for increasing prediction performance on ECMO treatment survival. This method allows observations with missing data to be retained and accounts for the uncertainty of the imputed values. The advantages come at the cost of complexity and increased computation time; multiple datasets are be imputed and results are to be pooled.

This paper begins by explaining how the data are cleaned and an explanation of the proceedure to pool results from MI in cross-validation. An explanantion of imputaiton methods follows. An explanation of the considered classification methods and how each is implemented then follows. Lastly, the results from predictions on each imputed data set are discussed.

2 Methodology

2.1 PreProcessing

Before analysis the data are standardised by mean-centering and scaling so the standard deviation is 1. The standardizing of variables in important in classification because variables measured at different scales do not contribute equally to the analysis. For example, the K-Nearest Neighbors method uses a distance metric to distinguish classes; a variable on a scale of 0 to 100 will be analyzed differently than a variable with a range of 0 to 1.

In addition to standardizing, the continuous variables are also transformed so the distributional form of the data is multivariate normal. Some nonparametric classification methods assume the data is multivariate normally distributed and can have better prediction performance if the assumption is true. Van Buuren (1) also suggests transforming the data toward normailty for multiple imputation. The data are transformed using the Yeo-Johnson transformation (17). The Yeo-Johnson transformation is similar to a Box-Cox transformation except it can accomodate covariate with zero and/or negative values.

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 (18). 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 (18–20).

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 methods explored on the ARDS dataset in this paper are: Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, K-Nearest Neighbors, and Random Forests.

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, let Y_i be independent Bernoulli random variables, then the model has the form

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

where $\mathbf{x}_i^T$ is the design matrix for observation i. 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:

$$\Pr(Y_i = 1|X_i) = \frac{\exp(\mathbf{x}_i^T\hat{\boldsymbol{\beta}})}{1 + \exp(\mathbf{x}_i^T\hat{\boldsymbol{\beta}})} \tag{2}$$

Logitistic Regression (Logit) is implemented with a logit link function using the `"glmnet"` method in the *caret* package (21). The parameters settings are `alpha = 1` and `lambda = 0` to suppress regularization.

LDA and QDA

Discriminant Analysis is a widely used set of classification methods. A generlization of Fisher's Linear Discriminant (22), discriminant functions are created through a combination of the explanatory variables that characterize the classes.

Let $p(X_i|Y_i)$ be the densities of distributions of the observations for each class where Y_i are independent Bernoulli random variables and let π_{Y_i} denote the prior probabilities of the Y_i^{th} class; that is, the prior probability that a randomly sampled observation belongs to the Y_i^{th} class based on the class proportions. The posterior probabilities may be written using Bayes Theorem as:

$$p(Y_i|X_i) = \frac{p(X_i|Y_i)\ \pi_{Y_i}}{p(X_i)} \propto p(X_i|Y_i)\ \pi_{Y_i} \tag{3}$$

Suppose the class distribution for class Y_i is Multivariate Normal with mean μ_{Y_i} and covariance matrix Σ_{Y_i}, so that:

$$p(X_i|Y_i) = \frac{1}{(2\pi_{Y_i})^{p/2}|\mathbf{\Sigma}_{Y_i}|^{1/2}}\exp\left[-\frac{1}{2}(X-\mu_{Y_i})^T\mathbf{\Sigma}_{Y_i}^{-1}(X-\mu_{Y_i})\right] \tag{4}$$

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

$$\log\frac{\Pr(Y_i=1|X_i)}{\Pr(Y_i=2|X_i)} = \log\frac{p(X_i|Y_i=1)}{p(X_i|Y_i=2)} + \log\frac{\pi_1}{\pi_2} \tag{5}$$

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

$$p(Y_i|X_i) \propto \exp\left(Q_{Y_i}\right) \tag{6}$$

where discriminant function is

$$Q_{Y_i} = (X_i - \mu_{Y_i})\Sigma_{Y_i}^{-1}(X-\mu_{Y_i})^T + \log|\Sigma_{Y_i}| - 2\log\ \pi_{Y_i} \tag{7}$$

for class Y. The Bayes Discriminant Rule is then: *allocated the observation to the class with the largest discriminant function, Q_Y.* 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 the discriminant function is defined as

$$L_Y = X\Sigma^{-1}\mu_Y^T - \frac{1}{2}\mu_Y\Sigma^{-1}\mu_Y^T - \log\ \pi_Y \tag{8}$$

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 discriminant function, L_Y.*

These methods come with disadvantages. Firstly, there is a bias-variance trade-off; both assume the covariates are normally distributed, there is no multicollinearity, and the observations are independent (23). 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 (18), even when the classes are not normally distributed.

LDA and QDA are implemented using the "lda" and "qda" methods in the *caret* package (21).

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

KNN is implemented using the "kknn" method in the *caret* package (21) with a grid search over the number of observations considered in classifying a new observation, $k = 3, ..., 19$. A Gaussian kernel and Euclidean distances are used.

Random Forests

Random Forests (24) are one of the most successful general-purpose modern algorithms(25). 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(26). See Figure 2 in Appendix B for an explanation of the random forests algorithm. Random Forests (RF) are implemented using the "rf" method in the *caret* package (21) with a grid search over the number of variables considered at each split, $mtry = 3, ..., 15$.

2.4 Accuracy Metrics

A classification method is typically assesed using a confusion matrix. Table 1 represents a confusion matrix for a binary classification.

Table 1: Confusion matrix for two classes.

		Observed	
		N	Y
Predicted	N	a	b
	Y	c	d

Accuracy is the percentage of correctly classifies instances out of all instances. It is often a poor performance metric to use alone. There are two significant problems with it. Firstly, 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. It does not provide information on which classes are being improperly classified or where. For the two class confusion matrix in Table 1 accuracy is defined as:

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

For binary classification, sensitivity and specificity provide more insight into performance of a classifier. These are defined as:

$$\begin{aligned}\text{sensitivity} &= \frac{a}{a+c}\\ \text{specificity} &= \frac{d}{b+d}\end{aligned}$$

Here, 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 of 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 proportions.

Cohen’s Kappa

Kappa or Cohen’s Kappa (27) is a classification performance metric that is normalized at the baseline of random chance on the dataset. It is a useful performance measure on problems with imbalanced classes. Kappa is defined as:

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

where p_o is simply the accuracy, the relative observed agreement between observed and predicted classes and p_e is the probability of chance agreement based on the class probabilities.

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

where

$$\begin{aligned}p_{o,Y} &= \frac{a+d}{a+b+c+d} \cdot \frac{a+c}{a+b+c+d}\\ p_{o,N} &= \frac{c+d}{a+b+c+d} \cdot \frac{b+d}{a+b+c+d}\end{aligned}$$

Kappa is used in this paper to compare the performance of different classifiers. A classifier with a larger Kappa is considered to predict better than a classifier with a lower Kappa. 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.

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 (28) 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 convenient because it allows many complexities that arise because of missing data to be ignored, it is typically an unrealistic assumption (1). 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

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 approach 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 (1).

Mean Imputation

Another common method for handling missing data is mean imputation; the missing value is replaced by the mean of the observed values (the mode for categorical data). This approach is satisfactory for a moderate amount of MCAR-generated missing values. However, it distorts the distribution of the data by reducing the variance of the imputed variables and the correlations between variables (29). Van Buuren suggests mean imputation should only be used only when there are few missing values, and should be generally avoided (1). Mean imputation (SI1) is implemented using the "mean" method in the *micemd* package with the number of imputations set as $m = 1$.

Multiple Imputation

Multiple imputation is a method that accounts for the uncertainty in the imputed values. The observed 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. Details of the *MICE* algorithm can be found in Algorithm 3.

Predictive Mean Matching

Many methods can be used to predict the missing value in the *MICE* algorithm (Algorithm 3) in Appendix B. Predictive Mean Matching (30, 31) (PMM) is used as the imputation method for MI of the ARDS. It is a versatile method robust to transformations of the target variable (1). PMM uses the observed values to make realistic imputations. Meaningless imputations are avoided because imputed values are always contained within the range of the observed data. PMM is also less vulnerable to model misspecification than other methods because the model is implicit, the distribution of missing values is based on the observed data rather than an explicit model (29). The algoritm as outlined by Van Buuren (1) pp. 57-58, is defined in Algorithm 4.

Let y be the vector of observed and imputed values where y_{obs} is the vector of n_1 observed values and $\dot{y}$ is the vector of n_0 imputed values. Then let X be the design matrix of predictors where X_{obs} indicates the subset of n_1 rows of X for which y is observed and X_{mis} is the subset of n_0 rows for which y is missing. Then Bayesian multiple imputation is specified as

$$\dot{y} = \dot{\beta}_0 + X_{\text{obs}}\dot{\beta}_1 + \dot{\epsilon} \qquad \text{where} \qquad \dot{\epsilon} \sim N(0, \dot{\sigma}^2) \tag{9}$$

where $\dot{\beta}_0$, $\dot{\beta}_1$, and $\dot{\sigma}$ are random draws from their posterior distribution, given the data. Imputations are drawn under the normal linear model using non-informative priors for each parameter.

Multiple imputation is implemented using the "pmm" method in the *micemd* package (32) with *maxit* $= 5$ and with the number of imputations set as $m = 9$ and $m = 99$.

2.7 Ensemble Multiple Imputation

While the topic of multiple imputation has been widely researched, how to best use multiple imputation in conjunction with cross-validation has not. Two approaches have been proposed for pooling results from several SVMs (33) and Cox regression (34) 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 (33). 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 (34) or slightly better performance with the first approach (33). For simplicity and the sake of computational costs, this paper, only considers the first approach as outlined in Figure 1.

The Algorithm 1 describes the steps of the ensemble approach for multiply imputed data in k-fold cross-validation:

1. Randomly partition the training data into k folds while retaining class proportions;
2. Define the k^{th} as the validation set and the remaining $k-1$ folds as the training set;
3. Impute the training set m times, with the response variable 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 validation set is concatenated with the extended training set;
7. Impute the combined validation and extended training set, with the response variable excluded, to create m imputed combined validation and extended training sets;
8. Extract the m validation sets;
9. Make m predictions on the m imputed validation sets;
10. Take the majority vote of the m predictions as the prediction for the fitted model;
11. Validate the prediction against the validation 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 performance.

Algorithm 1: One iteration in the cross-validation step of ensemble multiple imputation.

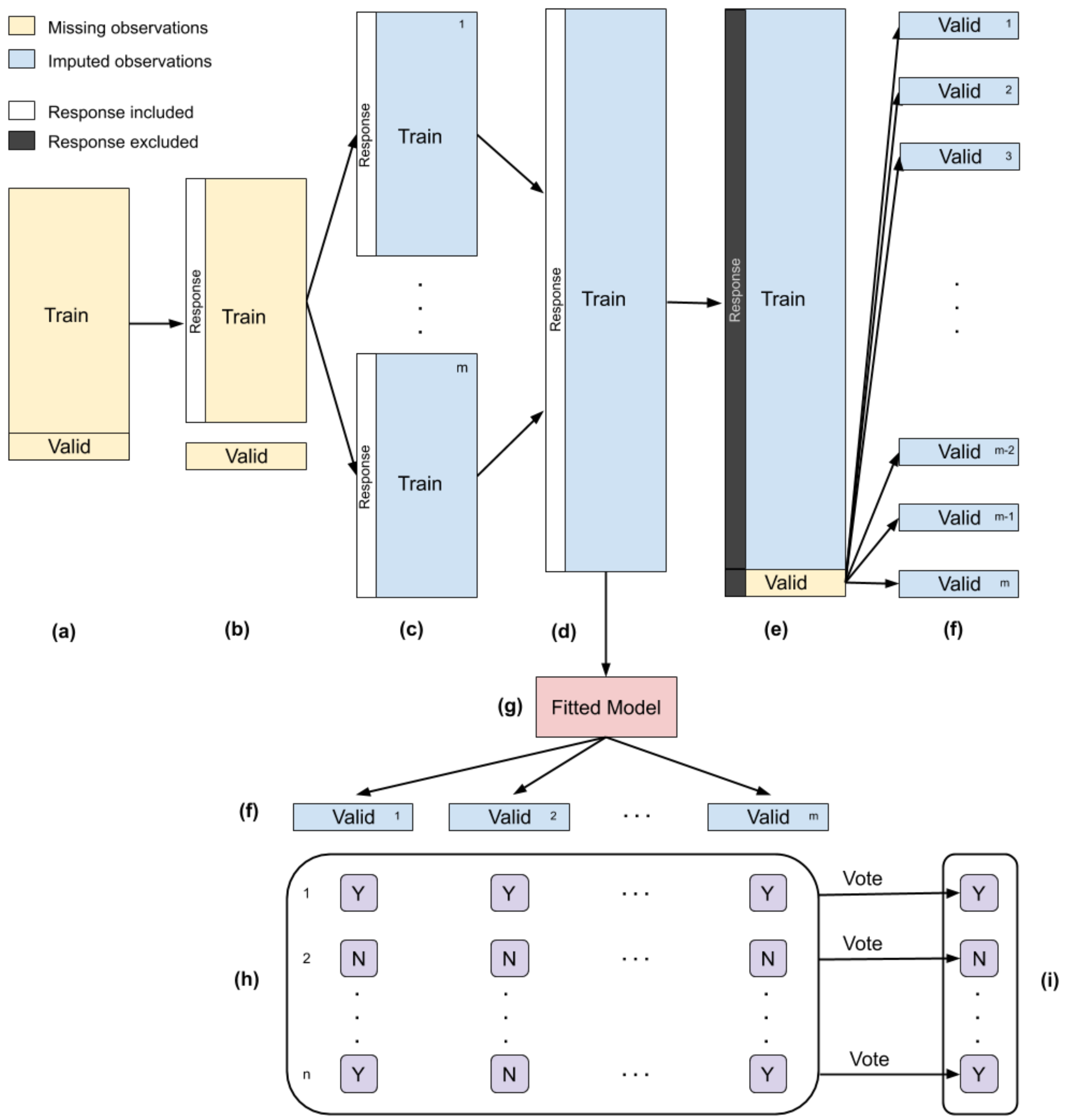

Figure 1: Outline of Kth step in the ensemble algorithm used to combine MI in cross-validation. (a) The Kth fold is taken as the test set (valid) and the remaining K-1 folds are taken as the training set. (b) Valid is separated from the analysis. (c) Train is imputed m times with the response included. (d) The m imputed datasets are 'stacked' to form one training set. (e) Valid is concatenated with the imputed and 'stacked' training set. (f) The test set is imputed m times using the imputed training set without the response included. (g) A model is fitted to the imputed and 'stacked' training set. (h) The fitted model makes predictions on each of the m valid sets. (i) The m predictions are pooled by a majority vote.

"Rubin's Rules" (28) 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. The classic advice for the choice of m is between 3 and 5 for moderate amounts of missing information but it is often beneficial to set m higher and create between 20-100 imputations (1).

There has been sufficient exploration into pooling of posterior probabilities resulting from classification problems (35, 36), but there has been little research into the pooling of predictions in classification problems on multiply imputed datasets (33). Pooling multiple predictions can be implemented using a variety of strategies, among which majority vote is one of the simplest, and has been found to be just as effective as more complicated schemes (37). Indeed, others have pooled predictions from various classification methods by taking the majority vote (33, 36) and comparing prediction performance.

This study involved four phases: (a) complete case analysis (CC) with the variable `PreECMO_Albumin` dropped from the analysis due to 46.44% missingness, (b) mean imputation (SI1) on variables with missing values, (c) imputation via the MICE algorithm implemented with PMM for $m = 9$ imputed datasets (MI9), and (d) imputation via the MICE algorithm implemented with PMM for $m = 99$ imputed datasets (MI99). In each phase the data is first randomly stratified into 75% training and 25% test sets, with the test set held-out. Five classification models are trained (Logit, KNN, LDA, QDA, RF) and tuned in 10-fold cross-validation using the ensemble imputation described in Figure 1. To determine the expected performance of future predictions, one iteration of the ensemble imputation is conducted for the full training set, the held-out test set, and the tuned classification methods.

2.8 Feature Selection

One of the goals of this analysis is to identify the variables most useful for accurate prediction. There are various methods that can be used for feature selection: stepwise selection, Recursive Feature Elimination (RFE), LASSO regularization, and principle Component Analysis (PCA). However, some of these methods are either highly criticized, dependent on the classification method considered, or cannot be integrated into the ensemble cross-validation approach used. Stepwise selection, while very common, is only applicable to regression models and it is often criticised (38); problems include falsely narrow confidence intervals for effects and predicted values (39) and multiple hypothesis testing inflating risks of capitalising on chance features of the data (40), such as noise covariates gaining entry into the model when the number of candidate variables is large (41). RFE is an iterative procedure analogous of backward feature selection. A new classifier is trained on a subset of the features and the importance of the feature is a measure of the change in performance. The training time scales linearly with the number of classifiers to be trained (42). Both logistic regression with LASSO regularization (43) and the analogous Sparse Discriminant Analysis (44) are embedded feature selection methods that are dependent on the classification method.

Principle Component Analysis (PCA) (45) is a feature extraction method that is independent of the classification method. The training set are orthogonally transformed into new uncorrelated variables called principle components that are linear combinations of the original variables. Feature extraction is accomplished by selecting the k largest principle components that contain a chosen percent of the variance in the original feature space.

PCA can also be used for feature selection by calculating the contribution of each variable to the extracted features (46). Let C_i be the contribution of a given variable on the principle component, V_i, and let λ_i be the eigenvalue of V_i, where $\mathrm{V}_i = \lambda_i \mathrm{C}_i$. The eigenvalues measure the amount of variation retained by each principle component. The total contribution of a variable, C_j, on explaining the variations retained by k extracted features, $\mathrm{V}_1, ..., \mathrm{V}_k$, is

$$\mathrm{C}_j = \sum_{i=1}^{k} |\lambda_{ij}\mathrm{C}_{ij}| = \sum_{p=1}^{k} |\mathrm{V}_{ij}| \tag{10}$$

The C_j are sorted in descending order where C_1 contributes the most variation to the extracted principle components among all the C_j for $j = 1, 2, ...p$, variables. Variables at the beginning of the sorted list are considered more important for the analysis than variables at the end. Here, any variable that contributes more than the expected average contribution – that is if all variables contributed equally – is selected as important for the analysis.

Variable selection via PCA is independent of the classification method and allows important variables to be identified outside of the classification analysis. The number of principle components retained, k, is based on the proportion of variance retained of the p principle components, where the variance threshold is chosen to be 80%.

$$0.8 \leq \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{p} \lambda_j} \tag{11}$$

The number of principle components retained is based on the proportion of variance. If the contribution of the $p = 30$ continuous variables were uniform, the expected value would be 1/p = 0.03. For a given component, an observation with a contribution larger than this cutoff could be considered as important in contributing to the component.

Variable selection is implemented using the `FactoMineR` package (47).

3 Exploratory Data Analysis

To get an idea of the distribution of the data, the following summary statistics were obtained for the categorical variables in Table 2 and for the continuous variables in Figure A1.

Table 2: Summary statistics for categorical variables.

Variable	Level	n	%
ECMO_Survival	N	109	24.22
	Y	341	75.78
Gender	m	305	67.78
	w	145	32.22
Indication	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 2 shows that the repsonse variable `ECMO_Survival` is imbalanced; of the 450 individuals, only 75.78% in the study sample survived ECMO treatment (341 survived vs 109 did not survive). The variable `Gender` is also imbalanced with only 67.78% of the individuals in the study sample are male (305 male vs 145 female). The distribution disease indication, `Indication` shows a majority are of level 2 and levels 3, 4, and 6 relatively rare occurances in this dataset.

Many of the standardised continuous variables in Figure A1 are highly skewed with a number of outliers. This can affect the performance of discriminant analysis classification methods that assume a distributional form for the data (18). The transformed continuous variables are shown in Figure A2

The heatmap in Figure A3 shows only a few variables with moderate to strong correlation. Only a few variables, `PreECMO_NAdose` and `PreECMO_Lactate`, are moderately correlated with many other variable. Feature selection methods based on the correlation matrix may not show strong feature importance for a subset of the variables.

3.1 Missing Data Exploration

Before imputation, and indeed multiple imputation, it is important to inspect the missingness patterns in the data and check assumptions. Figure 2 shows the missingness patterns in the dataset, where a black bar represents a missing value. Many missing values occur in observations with other missing values. The 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.

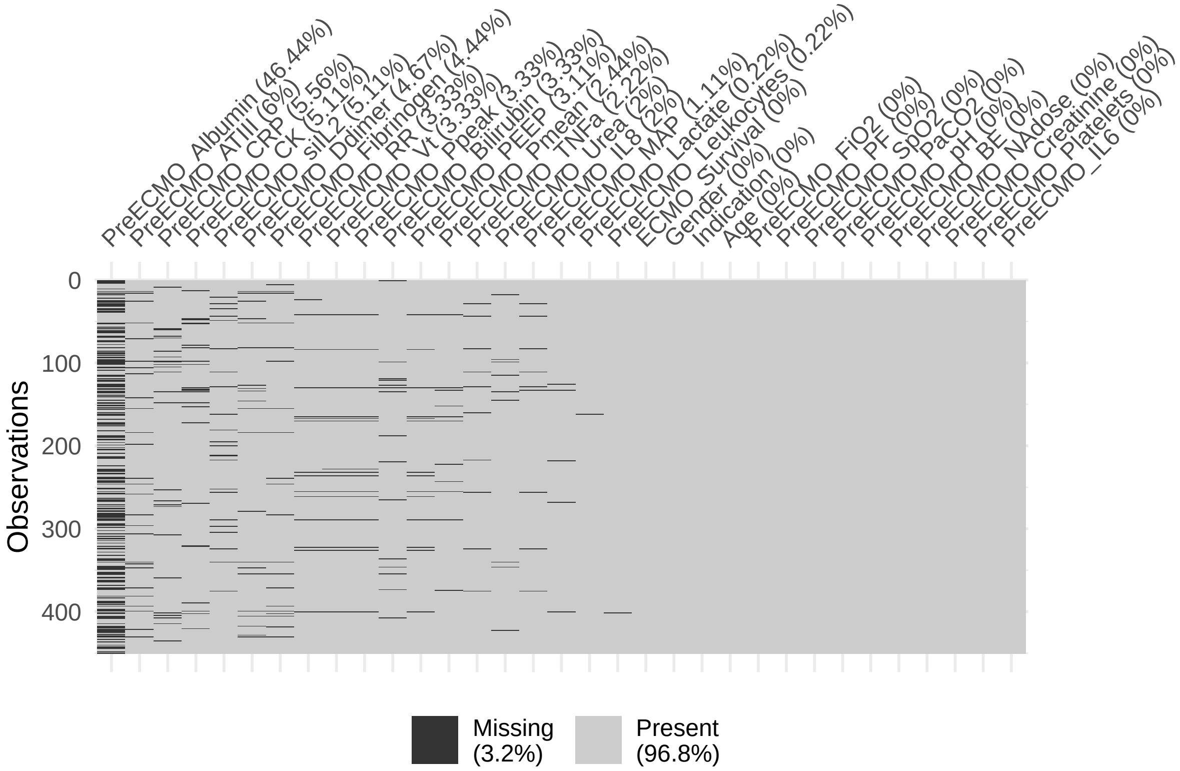

Figure 2: Visual representation of missing observations in the ARDS dataset.

From Figure 2, `PreECMO_Albumin` is seen to have 46.44% missingness. To conserve more observations for the training set, `PreECMO_Albumin` is dropped from the complete case analysis. Of the remaining variables only half contain missing values with moderate to low missingness up to 6%.

Table A1 provides some measures about variable dependence in the dataset. The first column 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 opposite 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 A1 shows that all variables will be useful during impuation with the exception of `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 (1).

4 Results

4.1 Prediction Performance

This study involved four phases: (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 for $m = 9$ imputed datasets, and (d) imputation via the MICE algorithm implemented with PMM for $m = 99$ imputed datasets.

The dataset was randomly stratified into 75% training and 25% test. The five classification models were trained in 10-fold cross-validation using the ensemble imputation approach. Table 3 shows the averaged Kappa from each analysis in 10-fold cross-validation. For CC and SI1, LDA had the highest averaged Kappa, while logistic regression had the highest averaged Kappa for MI9 and MI99.

Table 3: Averaged Cohen's Kappa for each model fitted in cross-validation. The tuned parameters for KNN and RF on each imputation emthod are (a) K=5 and mtry=11(b) K=5 and mtry=11(c) K=5 and mtry=13 (d) K=13 and mtry=15, respectively.

	Logit	LDA	QDA	KNN	RF
CC	0.139	0.205	0.038	0.053	0.035
SI1	0.191	0.220	0.040	0.136	0.085
MI9	0.179	0.124	0.106	0.088	0.136
MI99	0.185	0.158	0.037	0.127	0.177

Validation on Test Set

Using the parameters values learned in cross-validation (see Table 3), models were fit on the full training set and validated against the test set. Performance statistics are shown in Table 4. In complete case analysis, KNN with $K = 5$ performed the best with $\kappa = 0.161$. For the mean-imputed data, RF was the top performer with $\kappa = 0.197$. For both MI with $m = 9$ (MI9) and $m = 99$ (MI99), logistic regression outperformed the other classfication methods with $\kappa = 0.153$ and $\kappa = 0.274$, respectively.

The highest overall accuracy was 0.777 using RF on the mean-imputed dataset. However, the class-specific accuracies were 0.965 for survival and 0.185 for non-survival. The best predictor of non-survival was logistic regression on MI99.

4.2 Feature Selection

From the PCA analysis, at least 16 principle components were needed to explain 80% of the variance in the imputed training data and at least 15 principle components for the complete case analysis. The red dashed lines in Figure 3 indicate the expected average contribution of each variable to the selected principle components if each variable contributed equally to each principle component.

Table 4: Pooled performance results of trained models validated on test set.

		Sensitivity	Specificity	Accuracy	Kappa
CC	Logit	0.200	0.814	0.658	0.015
	LDA	0.200	0.847	0.684	0.054
	QDA	0.000	0.966	0.722	-0.048
	KNN	0.300	0.847	0.709	0.161
	RF	0.050	0.966	0.734	0.022
SI1	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
MI9	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
MI99	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

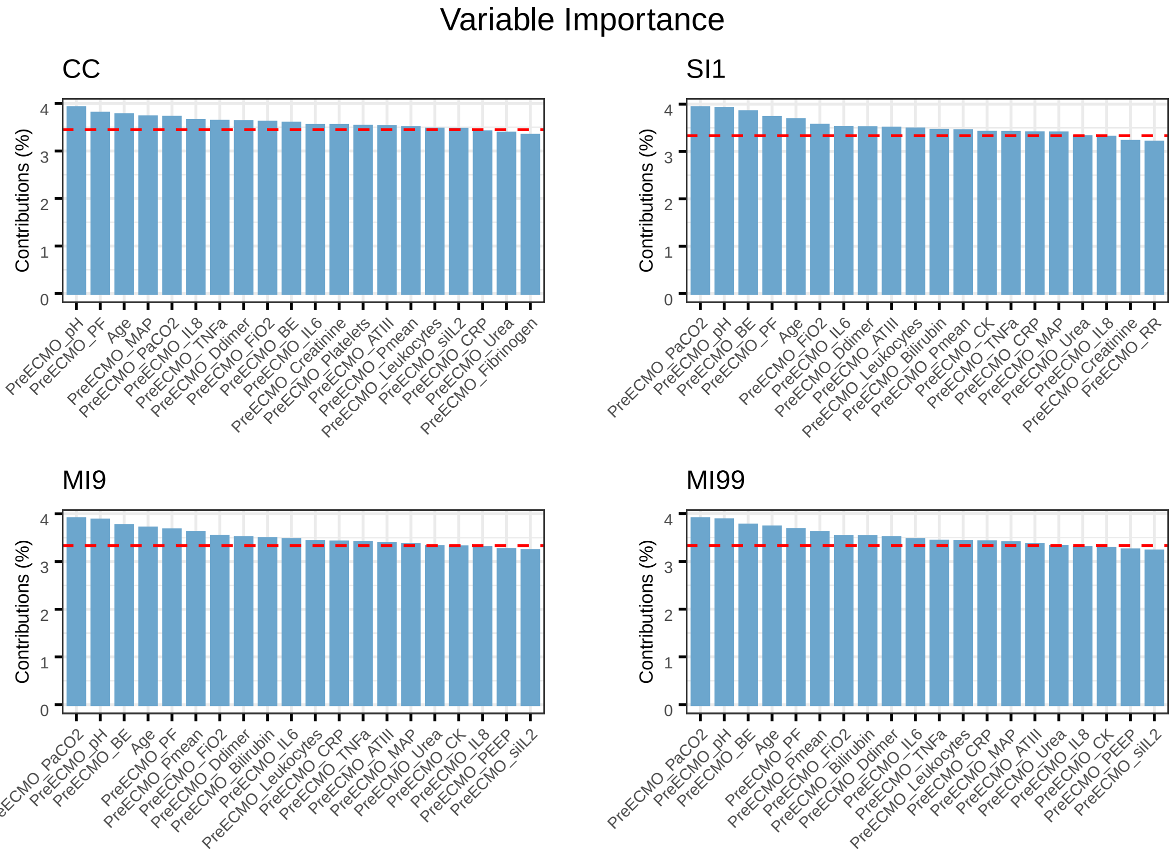

Figure 3: Contribution of variables to the principle components whose cumulative sum explains >80% of the variation in the data.

5 Discussion

Model performance on the imputed datasets were generally better than in complete case analysis. Non-parametric methods, KNN and RF, performed better on the complete case analysis and single mean imputation while logistic regression performed better on the multiply imputed datasets. All the methods in each analysis were able to predict survival with $> 80\%$ accuracy. The ability to predict non-survival was the limiting factor in the performance of a method. Non-survival was best predicted by logistic regression in MI99 with a prediction accuracy of 0.333.

Logistic regression performed consistently well in predicting non-survival and performed well for imputation methods except for complete case analysis. LDA also performed rather consistently for each imputed dataset. The consistent performance of LDA and logistic regression is not surprising given that they are similar methods, however logistic regression outperforms LDA in each analysis. LDA can perform better than logistic regression when the covariates are normally distributed (48), but LDA is not robust to outliers (18) and Figure A1 shows a number of outliers in almost every variable. Logistic regression is robust to outliers and makes less assumptions than LDA (18) allowing it to generalize better.

There were 136 less observations for complete case analysis than for the other experiments. Performance metrics have moderate variance due to the non-survival class in the test set only having 27 observations. Predicting one or two more observations as non-survival can have moderately large effects on Kappa. The relatively low number of observations compounded by the imbalance in the response classes make prediction difficult. Low predictive power of the variables make this problem even more difficult.

A surprising result is that KNN performed the best on CC, $\kappa = 0.161$ and also performed better than logistic regression on MI9 ($\kappa = 0.153$). RF performed poorly on CC ($\kappa = 0.022$) but was the best performer on SI1 ($\kappa = 0.197$) and second best on MI9 and MI99 ($\kappa = 0.116$ and $\kappa = 0.144$). The inverse performance of KNN and RF may be surprising but Tang et al. show (49) that random forests can perform poorly on datasets that work well for nearest-neighbor search problems without sufficient subsampling, due to a lack of diversity. On the imputed datasets KNN performed relatively poorly but similarly to LDA. It should be noted that KNN consistently was able to predict non-survival better than other methods, but at the cost of lower accuracy in predicting survival. QDA consistently performed the worst, no better than random chance based on the class likelihoods ($\kappa \approx 0$), suggesting that the class distributions do not support a quadratic decision boundary.

Feature Importance

The selected variables via PCA were the same for SI1, MI9, and MI99 with some variations in the order of importance. There were four selected variables in CC that were not selected in the imputed datasets: `PreECMO_IL8`, `PreECMO_Creatinine`, `PreECMO_Platelets`, and `PreECMO_SiL2`.

5.1 Improvements

One way to increase predictive performance is to include more observations in the analysis. Obtaining new data to include in the analysis could prove expensive or difficult. Instead, some observations from the test set could be retained for training the model in a nested cross-validation approach. The analyses done in this paper would constitute one iteration of the K_o outer cross-validation iterations where a new test set is selected by stratified randomly sampling, models are trained on the $K_\mathrm{o} - 1$ via an inner K_i-fold cross-validation. Since the data was originally split into 25% test and 75% train, If K_o is chosen to be >4, more observations can be retained in the training set. If $K_\mathrm{o} = 10$ were chosen, the prediction models would be trained on 67 more observations. The outer cross-validtion would then give the expected test prediction since it averages over different training sets (18). The drawback to nested cross-validation is that the time complexity scales from $O(K_\mathrm{i})$ to $O(K_\mathrm{o}K_\mathrm{i})$. Indeed, the full time complexity for m imputations and a grid search over p parameters would then be $O(pK_\mathrm{o}K_\mathrm{i}m)$.

The selected variables considered important in the survival prediction of ARDS patients after ECMO treatment should be verified. A simple way this can be done is by using only the variables shown as important in Figure 3 and conducting the ensemble imputation. Realistically, this can only be done for CC and SI1. For MI9 and MI99, the important variables should be analysed for each imputed dataset in the ensemble imputation algorithm. However, this adds another layer of complexity and embedded variable selection methods would be a more suitable option. Regularized logistic regression could be used to select variables through use of the LASSO (43). LASSO methods have also been developed for LDA and QDA: Sparse Discriminant Analysis (44) and DALASS (50). Random forests also naturally select important variables by accumulating the improvement in the split-criterion over all the trees in the forest for each variable (18).

Weighted predictions can be obtained by pooling posterior probabilities of the predictions using the arithmetic mean. Kittler et al. (35) show that classifier combination strategies based on the sum (*i.e* arithmetic mean) are less sensitive to errors than strategies based on the product (*i.e* geometric mean) but perform just as well as a majority vote. When combining predictions using the arithmetic mean, more certain predictions (near 0 or 1) weight the prediction more than uncertain predictions (near 0.5). Retaining posterior probabilities would allow a choice in posterior probability cutoffs for predictions and allow a performance analysis using ROC.

5.2 Conclusion

The CC has some distinctly different properties than the imputed datasets; the best classification method was KNN, which performed noteably worse in SI1, MI9, and MI99. The variables selected as important for the predictions also differed in CC compared to the imputed datasets. Given the curious results for CC, complete case analysis may not be an appropriate method for handling the missing values on the ARDS dataset. A more appropriate method would be mean imputation or multiple imputation. ECMO treatment survival can be predicted slightly better than random chance, $\kappa = 0.274$, using logistic regression trained on MI99. Future expected prediction accuracy for survival and non-survival of ECMO treatment are 0.906 and 0.333, respectively. Variables deemed important for survival prediction are: `PreECMO_PaCO2`, `PreECMO_pH`, `PreECMO_Be`, `Age`, `PreECMO_PF`, `PreECMO_PMean`, `PreECMO_FiO2`, `PreECMO_Bilirubin`, `PreECMO_Ddimer`, `PreECMO_IL6`, `PreECMO_TNFa`, `PreECMO_Leukocytes`, `PreECMO_CRP`, `PreECMO_MAP`, and `PreECMO_ATIII`.

Appendices

A. Additional Exploratory Data Analysis

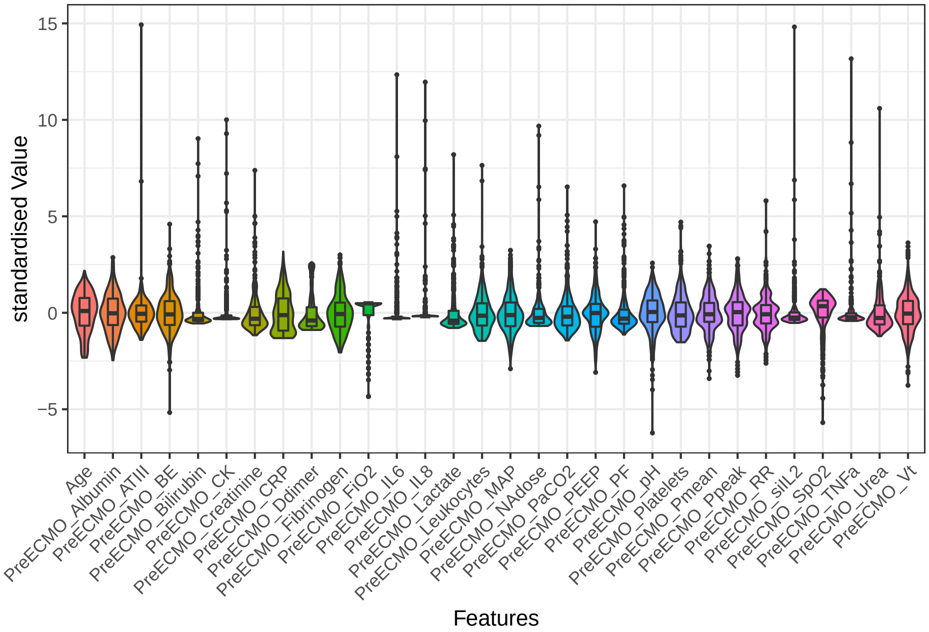

Figure A1: Violin plot of standardised continuous variables.

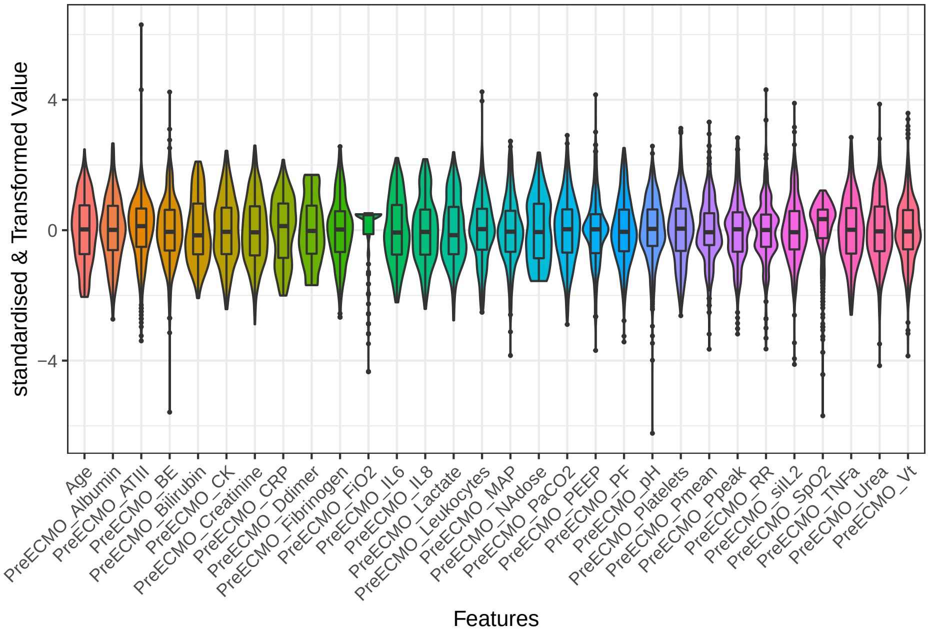

Figure A2: Violin plot of standardised and transformed continuous variables.

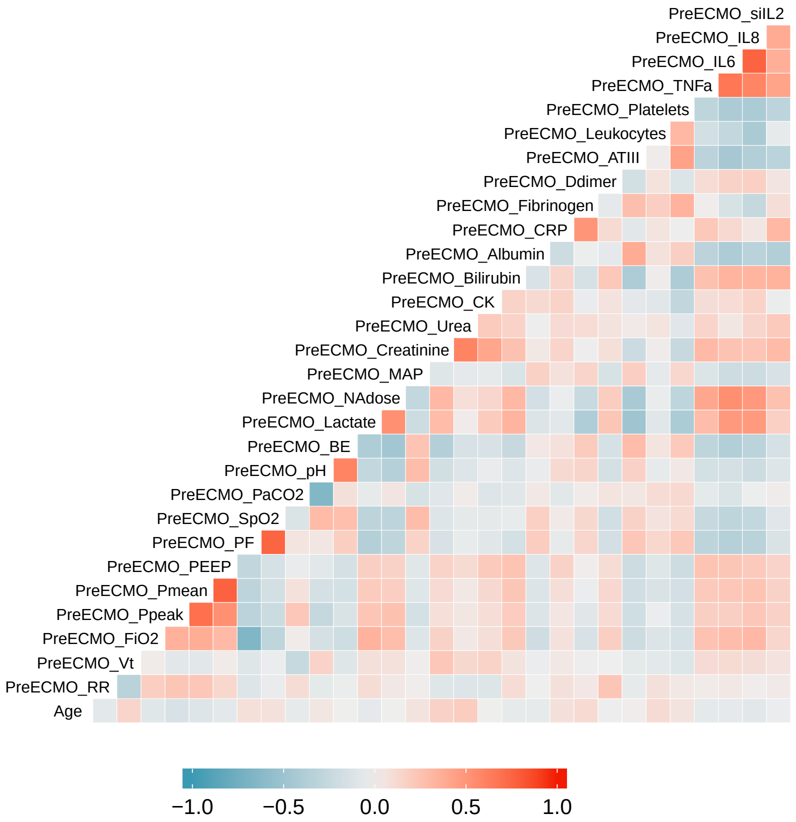

Figure A3: Heatmap of standardised and transformed variables.

Table A1 provides some measures about variable dependence in the dataset. The first column 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 opposite 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 A1 shows that all variables will be useful during impuation with the exception of `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 (1).

Table A1: 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

B. Algorithms

5.2.1 Random Forests Algorithm

The random forests algorithm depicted is adapted from (18).

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

5.2.2 MICE Algorithm

The MICE algorithm is adapted from (1).

```
Specify an imputation model $P(Y_j^{\text{mis}}|Y_j^{\text{obs}}, Y_{-j}, R)$ for variable $Y_j$ with $j = 1, ..., p$
For each $j$, fill in starting imputation $Y_j^0$ by random draws from $Y_j^{\text{obs}}$
Repeat for $t = 1, ..., T$ :
    Repeat for $j = 1, ..., p$ :
        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 $Y_j$
        Draw $\phi_j^t \sim P(\phi_j^t|Y_j^{\text{obs}}, Y_{-j}^t, R)$.
        Draw imputations from $Y_j^t \sim P(Y_j^{\text{mis}}|Y_j^{\text{obs}}, Y_{-j}^t, R, \phi_j^t)$.
    End repeat $j$.
End repeat $t$.
```

Algorithm 3: Multiple Imputation via Chained Equations

Using Equation 9 imputations are drawn under the normal linear model using non-informative priors for each parameter. The required inputs are:

- y_{obs}, the $n_1 \times 1$ vector of observed data in the target variable y;
- X_{obs}, the $n_1 \times q$ matrix of predictors of rows with observed data in y;
- X_{mis}, the $n_0 \times q$ matrix of predictors of rows with missing data in y.

```
Calculate the cross-product matrix $S = X'_{\text{obs}} X_{\text{obs}}$.
Calculate $V = (S + \text{diag}(S)\kappa)^{-1}$, with some small $\kappa$.
Calculate regression weights $\hat{\beta} = V X'_{\text{obs}} y_{\text{obs}}$.
Draw a random variable $\dot{g} \sim \chi_\nu^2$ with $\nu = n_1 - q$.
Calculate $\dot{\sigma}^2 = (y_{\text{obs}} - X_{\text{obs}}\hat{\beta})'(y_{\text{obs}} - X_{\text{obs}}\hat{\beta})/\hat{g}$.
Draw $q$ independent $N(0, 1)$ variate in vector $\dot{z}_1$.
Calculate $V^{1/2}$ by Cholesky decomposition.
Calculate $\dot{\beta} = \hat{\beta} + \dot{\sigma}\dot{z}_1 V^{1/2}$.
Calculate $\dot{\eta}(i, j) = |X_{\text{obs},[i]}\dot{\hat{\beta}} - X_{\text{mis},[j]}\dot{\beta}|$ with $i = 1, ..., n_1$ and $j = 1, ..., n_0$;
Construct $n_0$ sets $Z_j$, each containing $d$ candidate donors, from $Y_{\text{obs}}$ such that $\sum_d \dot{\eta}(i, j)$ is minimum for all $j = 1, ..., n_0$. Break ties randomly;
Draw one donor $i_j$ from $Z_j$ randomly for $j = 1, ..., n_0$
Calculate imputations $\dot{y}_j - y_{i_j}$ for $j = 1, ..., n_0$.
```

Algorithm 4: Imputation of y by predictive mean matching

Where κ is a ridge parameter used to avoid problems with singular matrices. The *mice* package uses $\kappa = 1e - 05$ as default.

C. Additional Missing Data Diagnostics

5.2.3 Visual Insepction of Imputations

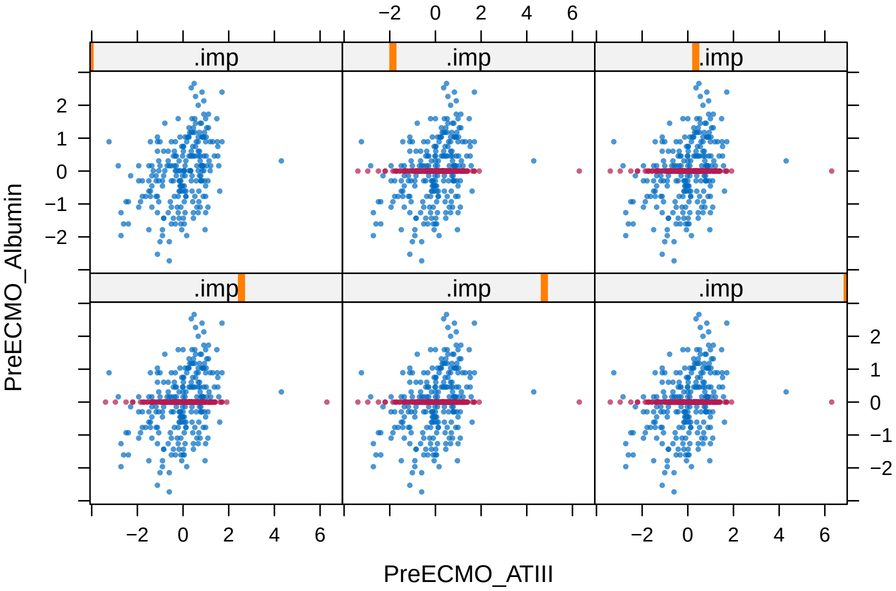

Figure C1: Scatterplot of 9 mean imputed datasets.

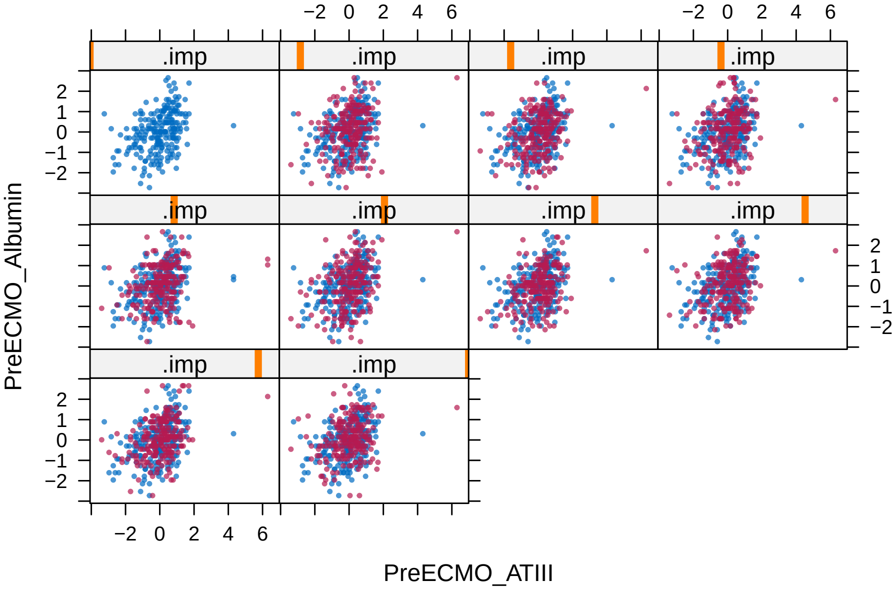

Figure C2: Scatterplot of the $m = 9$ imputed datasets for MI9. Can be extended to show similar results for MI99.

5.2.4 Convergence Monitoring

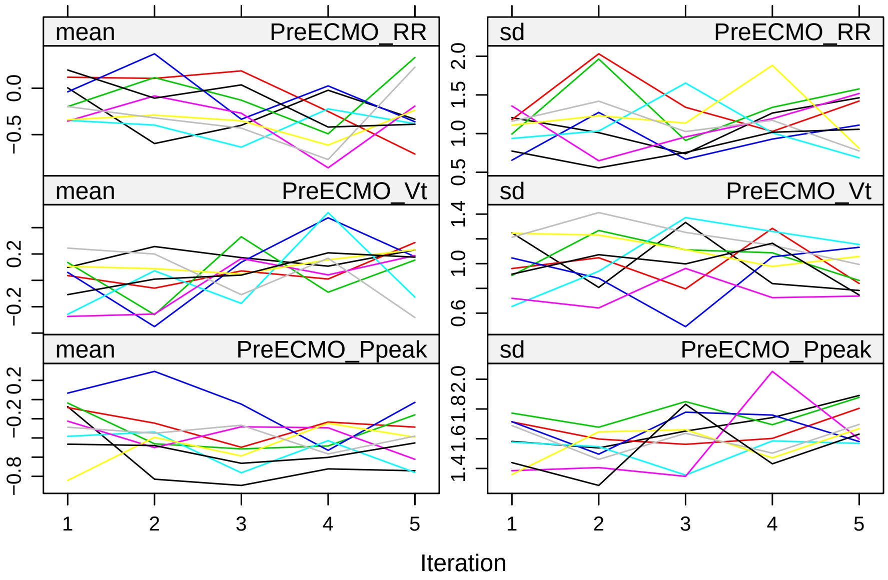

Figure C3: Mean and standard deviation of the synthetic values plotted against iteration number for SI1. The m streams should intermingle with one another in convergence, without showing any definite trends. Convergence occurs when the variance between different sequences is no larger than the variance within each individual sequence.

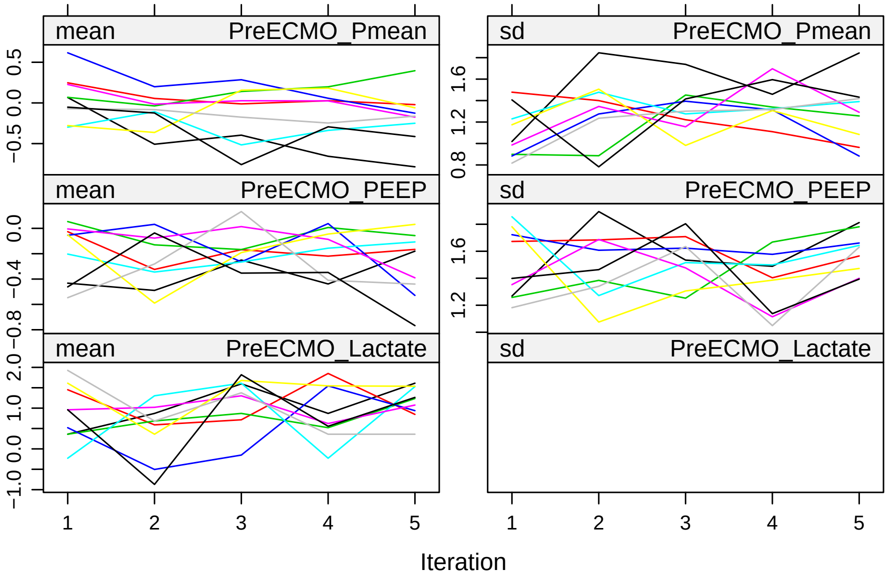

Figure C4: Mean and standard deviation of the synthetic values plotted against iteration number for SI1. The m streams should intermingle with one another in convergence, without showing any definite trends. Convergence occurs when the variance between different sequences is no larger than the variance within each individual sequence.

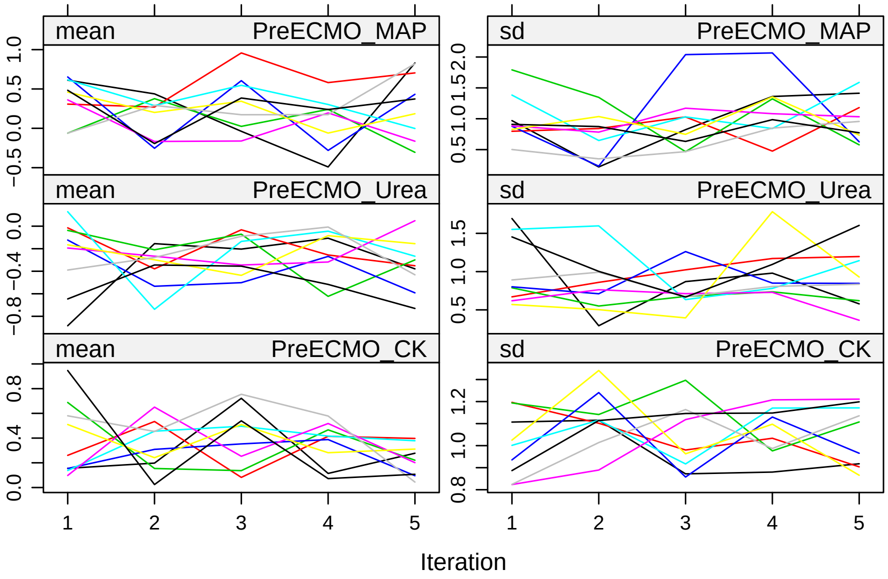

Figure C5: Mean and standard deviation of the synthetic values plotted against iteration number for SI1. The m streams should intermingle with one another in convergence, without showing any definite trends. Convergence occurs when the variance between different sequences is no larger than the variance within each individual sequence.

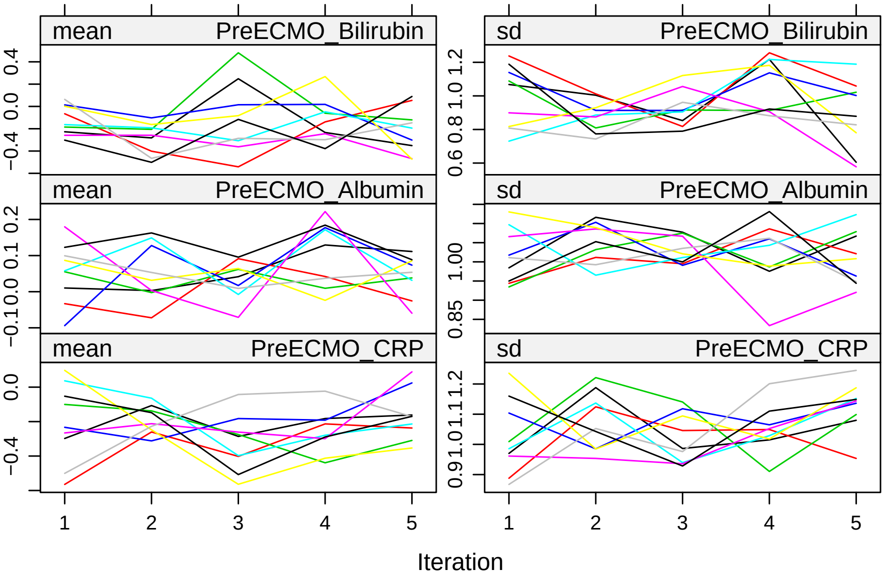

Figure C6: Mean and standard deviation of the synthetic values plotted against iteration number for SI1. The m streams should intermingle with one another in convergence, without showing any definite trends. Convergence occurs when the variance between different sequences is no larger than the variance within each individual sequence.

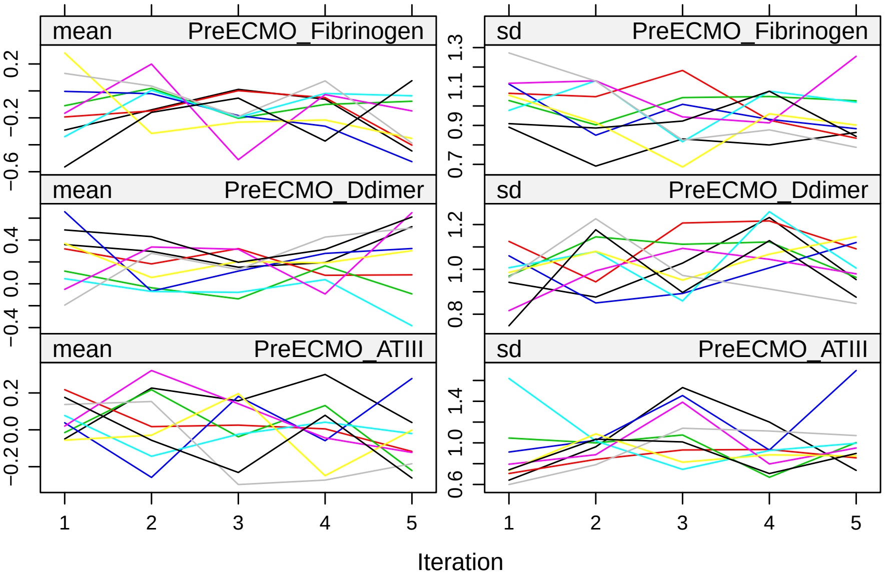

Figure C7: Mean and standard deviation of the synthetic values plotted against iteration number for SI1. The m streams should intermingle with one another in convergence, without showing any definite trends. Convergence occurs when the variance between different sequences is no larger than the variance within each individual sequence.

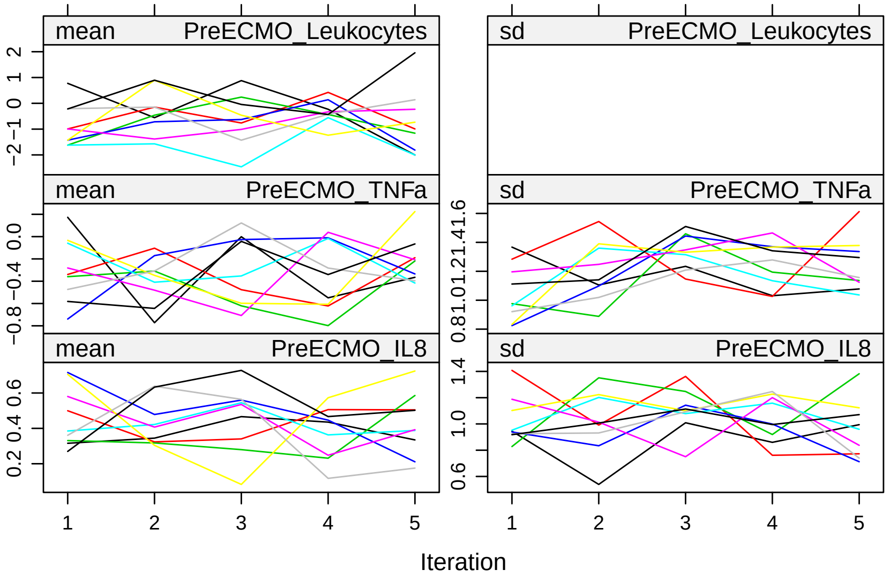

Figure C8: Mean and standard deviation of the synthetic values plotted against iteration number for SI1. The m streams should intermingle with one another in convergence, without showing any definite trends. Convergence occurs when the variance between different sequences is no larger than the variance within each individual sequence.

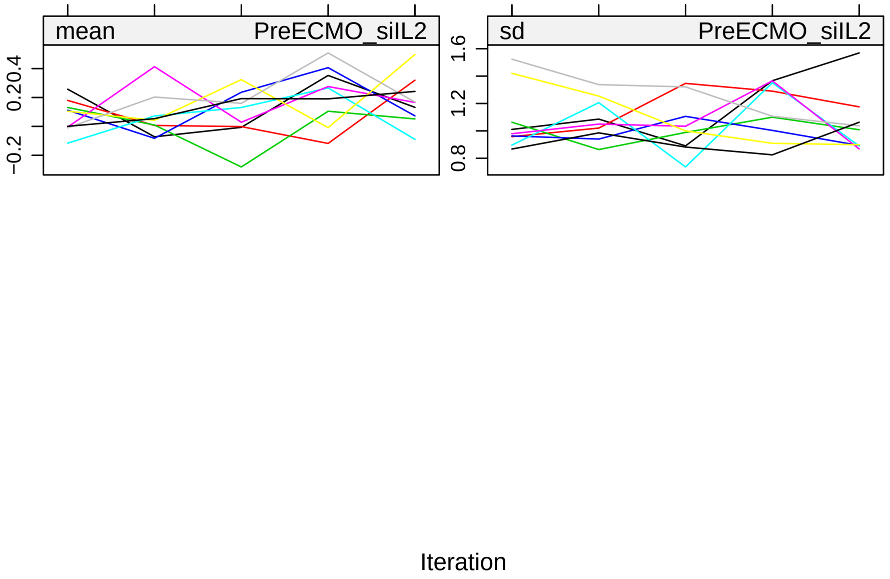

Figure C9: Mean and standard deviation of the synthetic values plotted against iteration number for SI1. The m streams should intermingle with one another in convergence, without showing any definite trends. Convergence occurs when the variance between different sequences is no larger than the variance within each individual sequence.

D. Code Structure

The code organization is described in Figure E1. `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`.

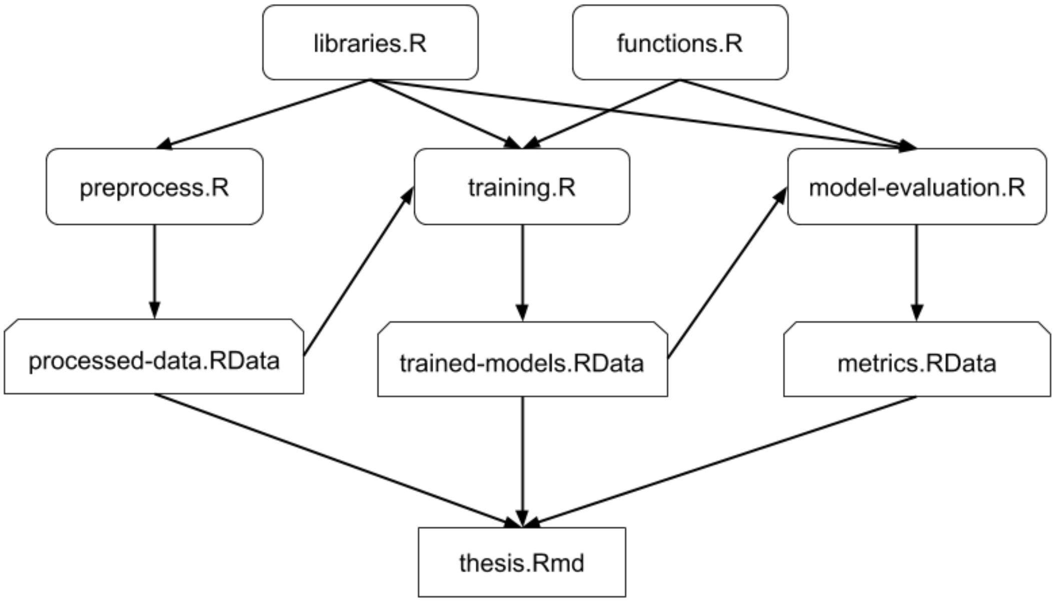

Figure E1: Flowchart of code structure.

References

1. Buuren S van (2012) *Flexible Imputation of Missing Data* (Chapman & Hall, London). Second Edition.

2. Schafer JL, Graham JW (2002) Missing data: Our view of the state of the art. *Psychological Methods* 7(2):147–177.

3. Powney M, Williamson P, Kirkham J, Kolamunnage-Dona R (2014) A review of the handling of missing longitudinal outcome data in clinical trials. *Trials* 15(1):237.

4. Karahalios A, Baglietto L, Carlin JB, English DR, Simpson JA (2012) A review of the reporting and handling of missing data in cohort studies with repeated assessment of exposure measures. *BMC medical research methodology* 12:96–96.

5. Wood AM, White IR, Thompson SG (2004) Are missing outcome data adequately handled? A review of published randomized controlled trials in major medical journals. *Clinical Trials* 1(4):368–376.

6. Mackinnon A (2010) The use and reporting of multiple imputation in medical research – a review. *Journal of Internal Medicine* 268(6):586–593.

7. Hayati Rezvan P, Lee KJ, Simpson JA (2015) The rise of multiple imputation: A review of the reporting and implementation of the method in medical research. *BMC Medical Research Methodology* 15(1):30.

8. Bellani G, et al. (2016) Epidemiology, Patterns of Care, and Mortality for Patients With Acute Respiratory Distress Syndrome in Intensive Care Units in 50 CountriesTrends in Acute Respiratory Distress Syndrome in 50 CountriesTrends in Acute Respiratory Distress Syndrome in 50 Countries. *JAMA* 315(8):788–800.

9. Rubenfeld GD, Herridge MS (2007) Epidemiology and Outcomes of Acute Lung Injury. *Chest* 131(2):554–562.

10. Fan E, Brodie D, Slutsky AS (2018) Acute Respiratory Distress Syndrome: Advances in Diagnosis and TreatmentAcute Respiratory Distress SyndromeAcute Respiratory Distress Syndrome. *JAMA* 319(7):698–710.

11. Paolone S (2017) Extracorporeal Membrane Oxygenation (ECMO) for Lung Injury in Severe Acute Respiratory Distress Syndrome (ARDS): Review of the Literature. *Clinical Nursing Research* 26(6):747–762.

12. Park PK, Napolitano LM, Bartlett RH (2011) Extracorporeal Membrane Oxygenation in Adult Acute Respiratory Distress Syndrome. *Critical Care Clinics* 27(3):627–646.

13. Wallace DJ, Milbrandt EB, Boujoukos A (2010) Ave, CESAR, morituri te salutant! (Hail, CESAR, those who are about to die salute you!). *Critical Care* 14(2):308.

14. Sahetya SK, Brower RG, Stephens RS (2018) Survival of Patients With Severe Acute Respiratory Distress Syndrome Treated Without Extracorporeal Membrane Oxygenation.

American Journal of Critical Care 27(3):220–227.

15. Calfee CS (2018) Acute respiratory distress syndrome subphenotypes and differential response to simvastatin: Secondary analysis of a randomised controlled trial. *The Lancet Respiratory Medicine* 6(9):691–698.

16. Sinha P, et al. (2018) Latent class analysis of ARDS subphenotypes: A secondary analysis of the statins for acutely injured lungs from sepsis (SAILS) study. *Intensive Care Medicine* 44(11):1859–1869.

17. Yeo I, Johnson RA (2000) A new family of power transformations to improve normality or symmetry. *Biometrika* 87(4):954–959.

18. Hastie T, Tibshirani R, Friedman J (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer) Available at: https://books.google.co.uk/books?id=eBSgoAEACAAJ.

19. Breiman L, Spector P (1992) Submodel Selection and Evaluation in Regression. The X-Random Case. *International Statistical Review / Revue Internationale de Statistique* 60(3):291–319.

20. Kohavi R (1995) A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95. (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA), pp 1137–1143.

21. Wing MKC from J, et al. (2019) Caret: Classification and Regression Training. Available at: https://CRAN.R-project.org/package=caret [Accessed August 29, 2019].

22. FISHER RA (1936) THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. *Annals of Eugenics* 7(2):179–188.

23. Cover TM (1965) Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers* EC-14(3):326–334.

24. Breiman L (2001) Random Forests. *Machine Learning* 45(1):5–32.

25. Biau G, Scornet E (2016) A random forest guided tour. *TEST* 25(2):197–227.

26. Biau G (2012) Analysis of a Random Forests Model. *J Mach Learn Res* 13:1063–1095.

27. Cohen J (1960) A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20(1):37–46.

28. RUBIN DB (1976) Inference and missing data. *Biometrika* 63(3):581–592.

29. Little RJA, Rubin DB (2014) Bayes and Multiple Imputation. *Statistical Analysis with Missing Data* (John Wiley & Sons, Ltd), pp 200–220.

30. Rubin DB (1986) Statistical Matching Using File Concatenation With Adjusted Weights

and Multiple Imputations. *Journal of Business & Economic Statistics* 4(1):87–94.

31. Little RJA (1988) Missing-Data Adjustments in Large Surveys. *Journal of Business & Economic Statistics* 6(3):287–296.

32. team) VA(M, team) MR-R(E (2019) Micemd: Multiple Imputation by Chained Equations with Multilevel Data. Available at: https://CRAN.R-project.org/package=micemd [Accessed August 29, 2019].

33. Belanche LA, Kobayashi V, Aluja T (2014) Handling missing values in kernel methods with application to microbiology data. *Neurocomputing* 141:110–116.

34. Zavrakidis I (2017) Combining Multiple Imputation with cross-validation for calibration and assessment of Cox prognostic survival models.

35. Kittler J, Hater M, Duin RPW (1996) Combining classifiers. *Proceedings of 13th International Conference on Pattern Recognition*, pp 897–901 vol.2.

36. James G (1998) *Majority Vote Classifiers: Theory and Applications*.

37. Lam L, Suen CY (1995) Optimal combinations of pattern classifiers. *Pattern Recognition Letters* 16(9):945–954.

38. Kemp F (2003) Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences. *Journal of the Royal Statistical Society: Series D (The Statistician)* 52(4):691–691.

39. Altman DG, Andersen PK (1989) Bootstrap investigation of the stability of a cox regression model. *Statistics in Medicine* 8(7):771–783.

40. Altman D (1991) *Practical Statistics for Medical Research* (Chapman & Hall: London).

41. Derksen S, Keselman HJ (1992) Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology* 45(2):265–282.

42. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning* 46(1):389–422.

43. Tibshirani R (1996) Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1):267–288.

44. Clemmensen L, Hastie T, Witten D, Ersbøll B (2011) Sparse Discriminant Analysis. *Technometrics* 53(4):406–413.

45. F.R.S KP (1901) LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2(11):559–572.

46. Song F, Guo Z, Mei D (2010) Feature Selection Using Principal Component Analysis. *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, pp 27–30.

47. Husson F, Josse J, Le S, Mazet J (2019) FactoMineR: Multivariate Exploratory Data Analysis and Data Mining. Available at: https://CRAN.R-project.org/package=FactoMineR

[Accessed August 29, 2019].

48. Efron B (1975) The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis. *Journal of the American Statistical Association* 70(352):892–898.

49. Tang C, Garreau D, Luxburg U von (2018) When do random forests fail?

50. Trendafilov NT, Jolliffe IT (2007) DALASS: Variable selection in discriminant analysis via the LASSO. *Computational Statistics & Data Analysis* 51(8):3718–3736.