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

Robert Edwards

2416963E

MASTER THESIS

Biostatistics

Contents

1 Introduction

1.1 Aim of the Thesis

- paragraph about importance of in-sample vs. out-of-sample prediction accuracy
- Cross validation
- Over fitting / under fitting
- Paragraph about Missing data

1.2 The Clinical Study

1.3 Study Population & Data Description

1.4 The Statistical Challenege

2 Methodology

2.1 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 2018)**. 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. 2018)**.

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

2.2 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.

2.2.1 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:

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

2.2.2 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. 2017)**, even when the classes are not normally distributed.

2.2.3 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.

2.2.4 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).

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

2.3 Accuracy Metrics

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

2.3.1 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?)

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

Table 1: Confusion matrix for two classes.

		Observed	
		N	Y
Predicted	N	a	b
	Y	c	d

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

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.

2.3.2 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.

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

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

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

3 Statistical Methods for the Analysis

Describe the methods step-by-step for the analysis

3.1 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.

3.2 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.

3.3 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.

3.4 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].

3.4.1 Fully Conditional Specification

3.4.2 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.

3.5 Ensemble Multiple Imputation

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

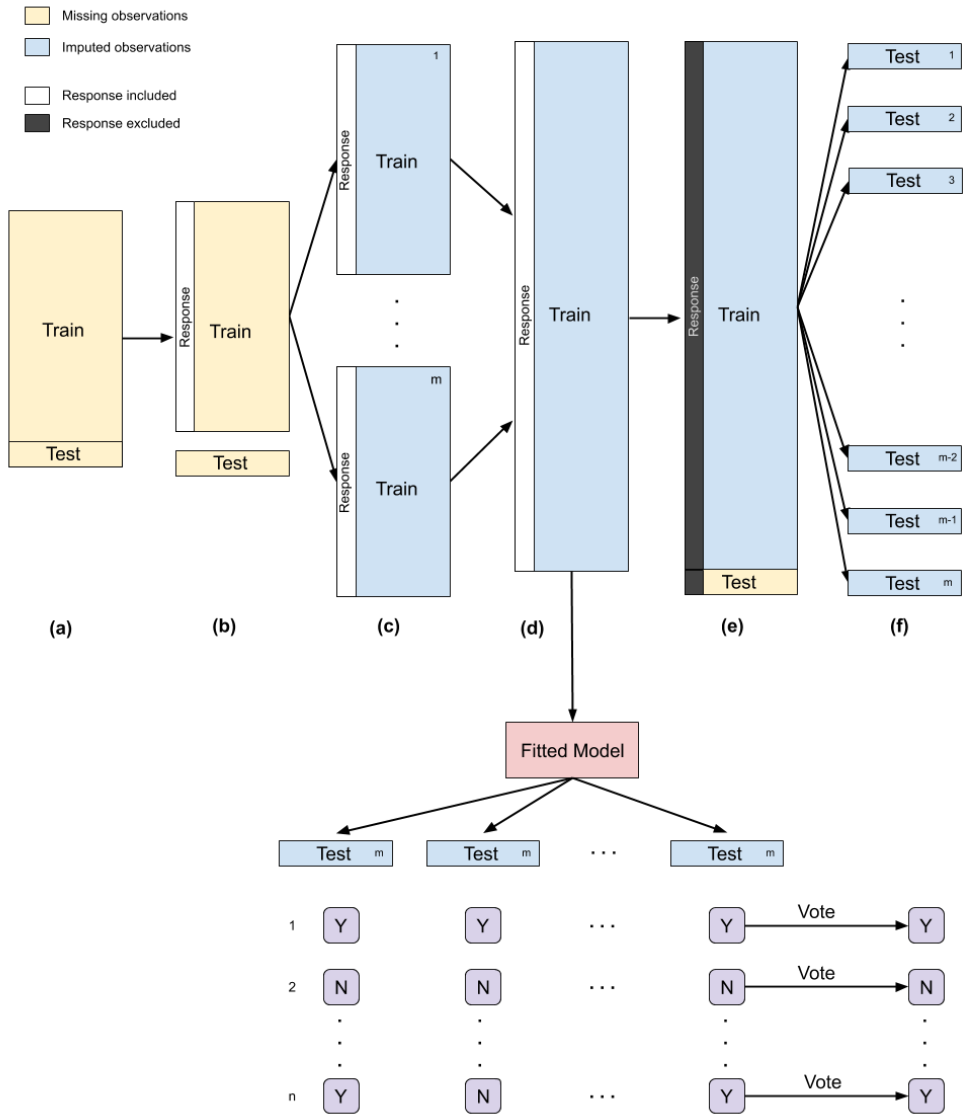

Figure 1: Outline of the algorithm used to pool predictions from multiple imputation. (a) Step 1. (b) Step 2. (c) Step 3. (d) Step 4. (e) Step 5. (f) Step 6.

3.5.1 Number of Imputations

- Rubin's Rule
- Sample size calculator

$$\left(\hat{p} - 1.96\sqrt{\frac{0.25}{n}}, \quad \hat{p} + 1.96\sqrt{\frac{0.25}{n}}\right)$$

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.

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."

- Rubin's Rules allow the pooling of parameter estimates in GLMs but...
- To my knowledge, there has been insufficient work on estimating the required number of imputations for pooling posterior probabilities in classification problems.

- Cite the Dutch Master Thesis
- Adapt Rubin's Rules - arguing that

3.6 Voting

3.6.1 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, 1994).

(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.

4 Results

4.1 Exploratory Data Analysis

- describe the data:
- 3 Categorical variables
- 30 continuous variables
- Violin plots in appendix

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.

4.2 Missing Data Patterns

Before imputation, and indeed multiple imputation, it is important to inspect the missingness patterns in the data and check assumptions. Figure 2 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. We see that **all variables are useful except XXX**. 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).

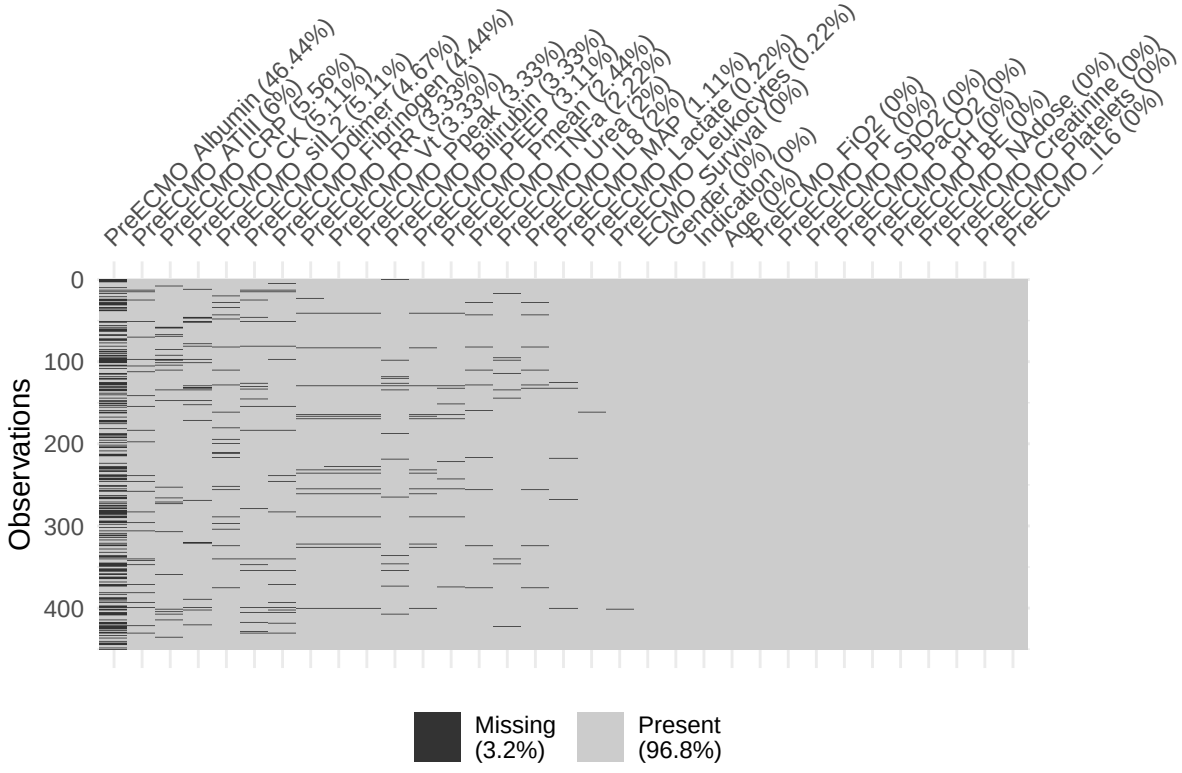

Figure 2: Visual representation of missing observations in the ARDS dataset.

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: 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 7: Mean imputation accuracy metrics. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=11, respectively.

	Sensitivity	Specificity	Accuracy	Kappa
Logit	0.259	0.871	0.723	0.147
LDA	0.222	0.859	0.705	0.091
QDA	0.111	0.906	0.714	0.021
KNN	0.259	0.835	0.696	0.102
RF	0.074	0.976	0.759	0.071

Table 8: Predictively mean-matching imputation accuracy metrics. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=13, respectively.

	Sensitivity	Specificity	Accuracy	Kappa
Logit	0.259	0.882	0.732	0.162
LDA	0.259	0.882	0.732	0.162
QDA	0.111	0.906	0.714	0.021
KNN	0.259	0.859	0.714	0.131
RF	0.074	0.953	0.741	0.037

Table 9: 99 Predictively mean-matching imputation accuracy metrics. The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=13, respectively.

	Sensitivity	Specificity	Accuracy	Kappa
Logit	0.259	0.894	0.741	0.178
LDA	0.296	0.882	0.741	0.202
QDA	0.111	0.906	0.714	0.021
KNN	0.222	0.871	0.714	0.106
RF	0.148	0.941	0.750	0.116

5 Discussion

5.1 Model Performance

Logistic Regression
For complete-case analysis, mean imputation, and predictive mean-mathcing, 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.

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

6 Conclusion

7 Appendices

7.1 Additional Material

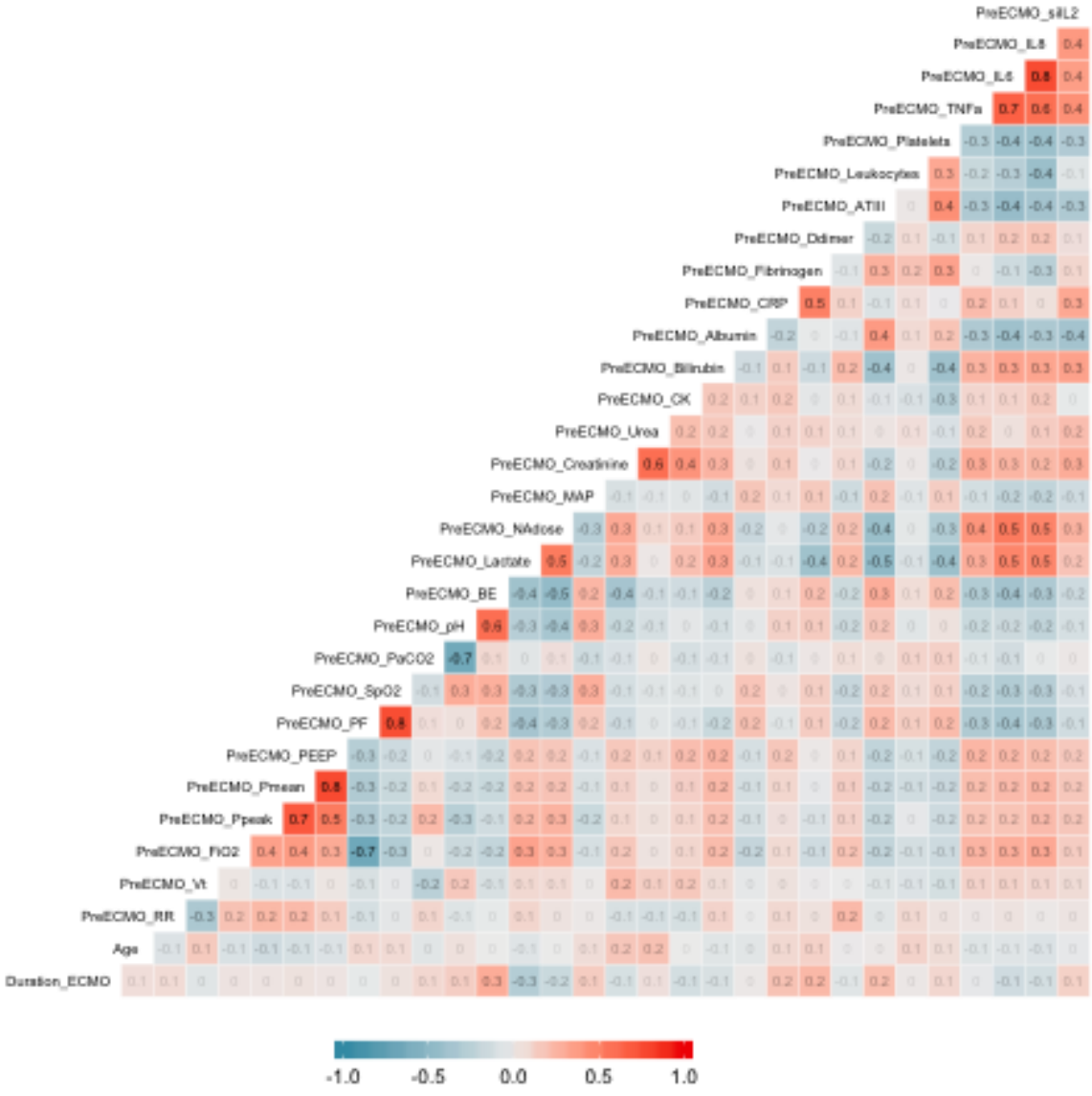

Figure 3: Heatmap of standardized and transformed variables.

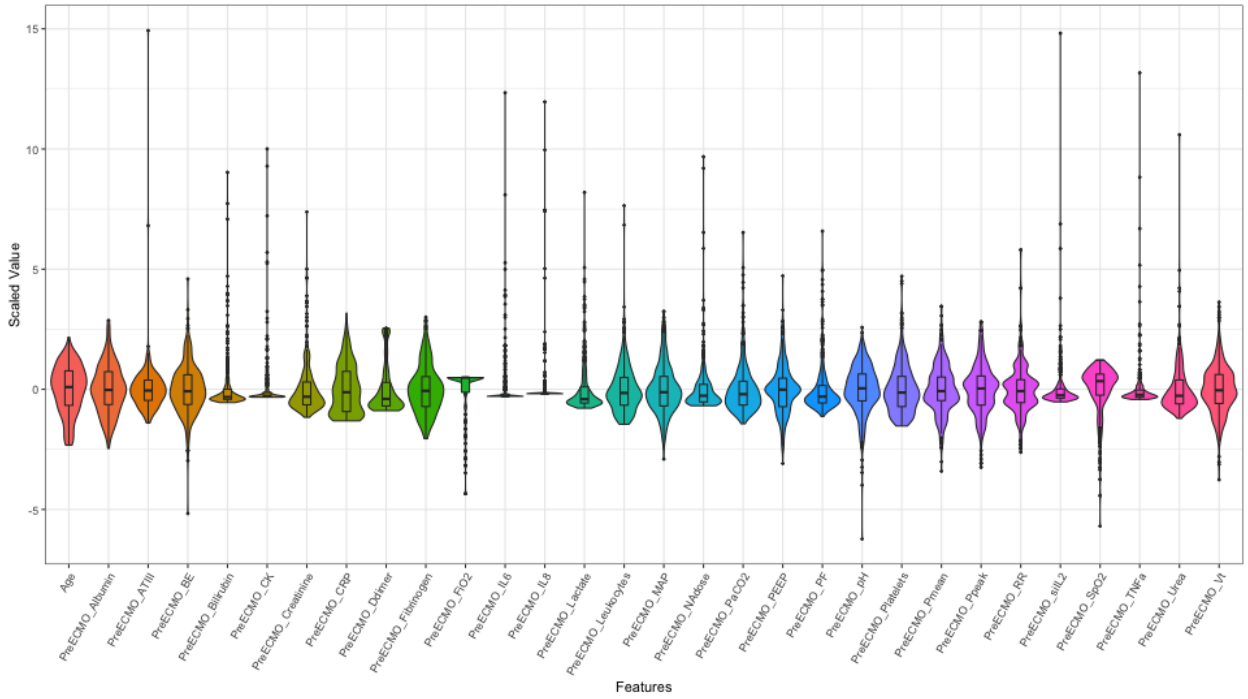

Figure 4: Violin plot of standardized variables.

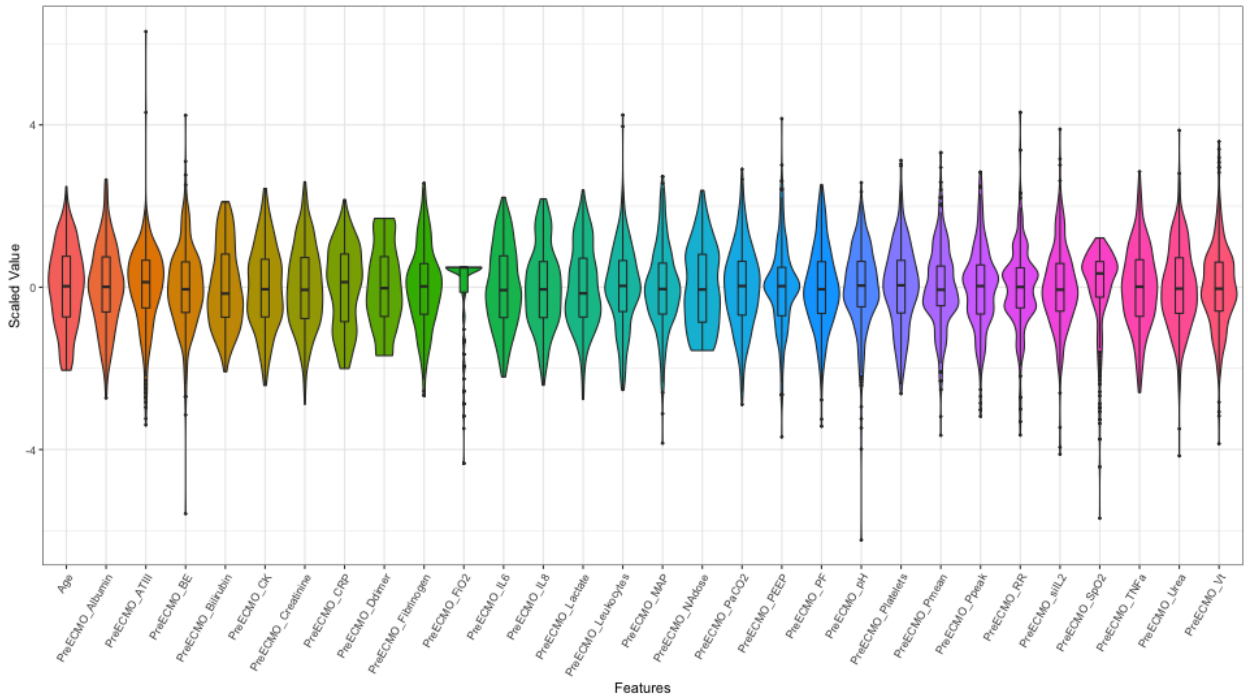

Figure 5: Violin plot of standardized and transformed variables.

Table 10: Averaged Cohen's Kappa for each model fitted in cross-validation. The tuned hyperparameters 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

7.1.1 Kappa Values for Model Selection

7.2 R Code

Figure showing flow of code

References

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.