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

Robert Edwards

2416963E

MASTERS THESIS

Biostatistics

Acknowledgements

To my peers, alone we sink but together we swim.

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 Google, couldin'a donnit wit' out ya.

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 but little can be rememdied post-hoc. Missing values in the data complicate analysis even further and are often handled either by dropping missing observations or filling in the missing value by the mean. Both methods can be valid if certain assumptions hold, but useful information is either lost to the analysis or the sample become biased **(Citation)**.

the natural distribution of the data is effected.

1.1 Study Population & Data Description

This paper investigates predicting survival of patients diagnosed with Acute Respiratory Distress Syndrom (ARDS) after (ECMO) treatment. ARDS is a _____ that affects _____ people world wide. It has a moretality of _____. Current treatments... ECMO is a treatment used in ____ that is thought to help patients with ARDS.

The dataset is composed of 450 observations on patients with Acute Respiratory Distress Syndrome who underwent ECMO treatment. The response variable, `ECMO_Survival`, is a binary categorical variable for survival indication with levels "Y" and "N". 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` 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 both allows retention of observations in the analysis as well as accounts for the uncertainty of the imputed value. The advantages come at the cost of complexity and increased computation time. Multiple datasets must be imputed and results somehow pooled.

2 Methodology

2.1 PreProcessing

Before analysis the data are standardized 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. The data are transformed using the Yeo-Johnson transformation (Yeo and Johnson, 2000). 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 (Hastie et al., 2009). The data is randomly split into K folds, where the K^{th} fold is taken as the validation set and the the remaining $K-1$ folds are used for training the model. The procedure is then repeated K times and the prediction error averaged. K-fold cross validation is most useful on sparse datasets as it allows more observations to be used in training the model. The choice of K can effect the variability of the prediction error; if $K = 1$, the model will overfit the data and prediction error will be highly variable and if $K = n$ (the number of observation in the dataset), the model is fit with no validation set for training parameters. Typical values used are $K = 5$ & 10 (Hastie et al., 2009) **Breiman and Spector (1992) Kohavi (1995)**.

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

2.3 Models

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

2.3.1 Logistic Regression

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

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

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

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

+ **State how I implemented Logistic Regression**

2.3.2 LDA and QDA

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

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

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

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

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

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

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

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

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

where

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

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

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

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

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

- **State how I implemented LDA and QDA**

2.3.3 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a commonly used non-parametric classification method. To predict the class of a new observation, a distance matrix is constructed between all

observations and the K nearest labelled observations to the new observation are considered. The new observation is then assigned the class label that the majority of its neighbors share. In case of only two classes, ties in class assignments are avoided by using odd values of K. In the event of a tie, a class can be chosen at random. Various distance metrics may be used but it is common to use Euclidean distance to determine the closest training points, though it is advisable to scale variables so that one direction does not dominate the classification **(Citation)**.

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.

- **State how I implemented KNN**

2.3.4 Random Forests

Random forests (Brieman, 2001) are one of the most successful general-purpose modern algorithms (Biau and Scornet, 2016). They are an ensemble learning method that can be applied to a wide range of tasks, namely classification and regression. A random forest is created by building multiple decision trees, where randomness is introduced during the construction of each tree. Predictions are made by classifying a new observation to the mode of the multiple decisions tree classifications. Random forests often make accurate and robust predictions, even for very high-dimensional problems (Biau, 2012). See **(Appendix X)** for an explanation of the random forests algorithm.

- **State why random forests are good predictors**
- **State how I implemented Random Forests**

2.4 Accuracy Metrics

2.4.1 Accuracy, Sensitivity, and Specificity

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. 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 binary classification, sensitivity and specificity provide more insight into classification performance.

For the two class confusion matrix in Table 1 accuracy metrics used in the analysis 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 proportions.

2.4.2 Cohen's Kappa

Kappa or Cohen's Kappa (Cohen, 1960) 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. Cohen's 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.

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

where

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

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

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.

- **State how I Kappa and other metrics are used**

2.5 Missing Data

Missing data is a common problem that must be dealt with in machine learning, statistics, and medicine. Understanding the missing mechanism for the missing observations is important in the analysis. (RUBIN, 1976) defined three types of missing data mechanisms: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). The data are said to be missing completely at random (MCAR) if the probability of being missing is the same for all cases. This implies the causes of the missing data are unrelated to the data itself. While MCAR is conveinient because it allows many complexities that arise because data are missing to be ignored, it is typically an unrealistic assumption (van Buuren, 2012). The data is said to be MAR if the probability of being missing is the same only within groups defined by the observed data. MAR is a more general and more realistic assumption than MCAR. If neither MCAR nor MAR applies, then the probability of being missing depends on an unknown mechanism and said to be MNAR. Most simple approaches to dealing with missing data are only valid under MCAR assumption. Modern methods to dealing with missing data begin from the MAR assumption.

2.6 Imputation Methods

2.6.1 Complete Case Analysis

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

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

2.6.2 Mean Imputation

Another common method for handling missing data is mean imputation; the missing value is replaced by the mean of the 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 (Little and Rubin, 2014). Van Buuren suggests mean

imputation should only be used only when there are few missing values, and should be generally avoided (van Buuren, 2012). Mean imputation is considered in this paper because although it is often a poor method of choice for imputing missing values, it is commonly done in medical datasets **(Citation)**.

- **State how I implemented mean imputation is done**

2.6.3 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 2.

2.6.4 Fully Conditional Specification

2.6.5 Predictive Mean Matching

Predictive Mean Matching (PMM) is a semi-parametric imputation approach to imputing missing values. 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). 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.

- **Equations for Predictive Mean Matching**

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 (Belanche et al., 2014) and Cox regression **(Zavrakidis 2017)** from multiply imputed datasets. The method is to concatenate the m imputed datasets and fit a classifier, and optimize, to the resulting set; this accounts for the variability of the parameter estimates as well as the variability of the training observations in relation to the imputed values (2014). The second proceedure fits separate classifiers to each imputed data set and get the pooled (i.e. avereaged) performance of the m classifiers. Results from both studies either show similar results between appraoches **(Zavrakidis 2017)** or slightly better performance with the first approach (2014). For simplicity and the sake of computational costs, this paper, only considers the first approach as outlined in Figure 1.

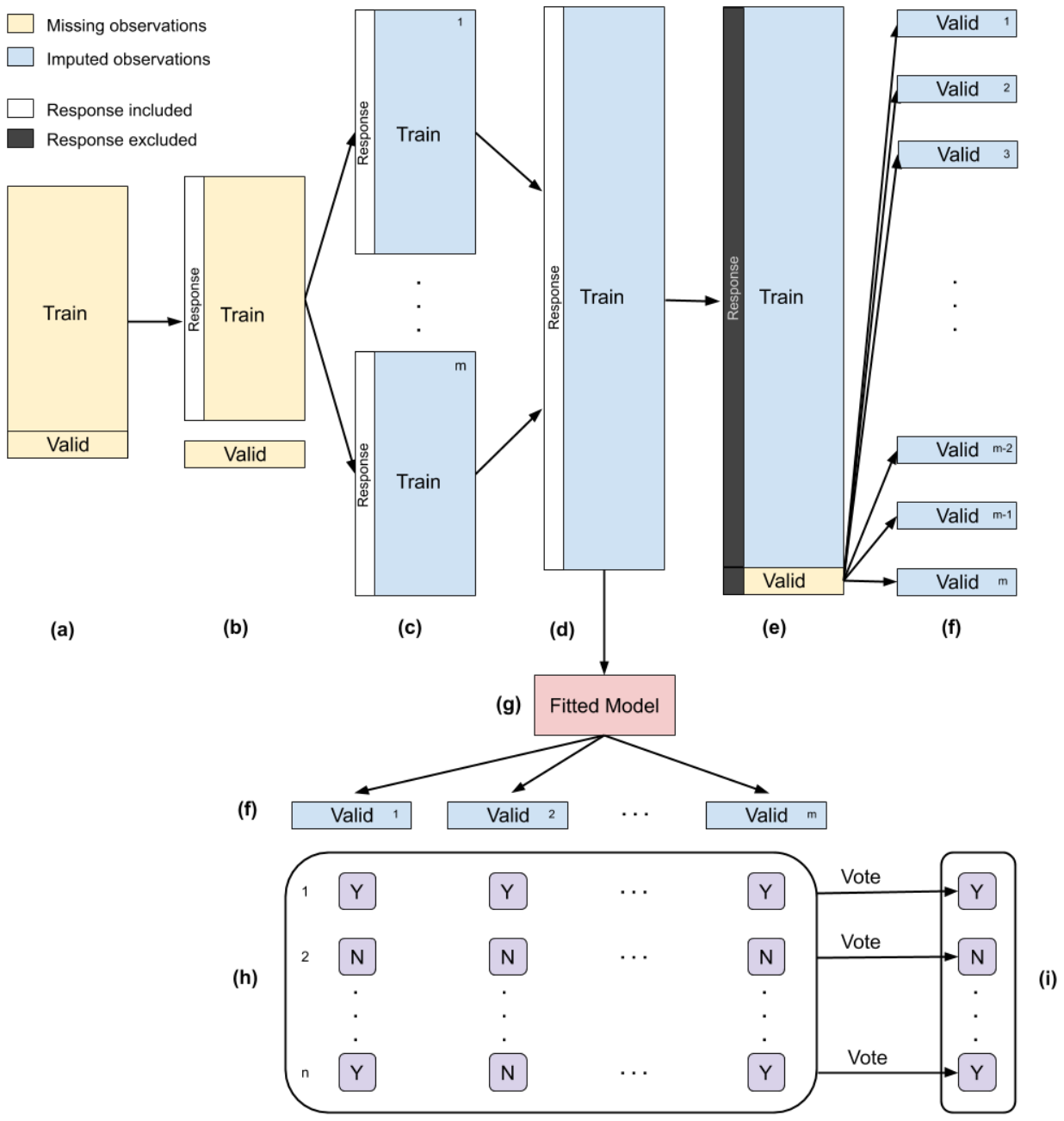

Figure 1: Outline of Kth step in the ensemble algorithm used to combine MI in cross-validation. (a) The Kth fold is taken as the test set and the remaining K-1 folds are taken as the training set. (b) The test set is separated from the analysis. (c) The training set is imputed m times with the response included. (d) The m iputed datasets are 'stacked' to form one training set. (e) The test set 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 test sets. (i) The m predictions are pooled by a majority vote.

The following steps describe 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 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 performance

"Rubin's Rules" (RUBIN, 1976) 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 (van Buuren, 2012).

The training set is multiply imputed with PMM for $m = 9$ and $m = 99$ and the predictions pooled by majority vote. There has been sufficient exploration into pooling of posterior probabilities resulting from classification problems **(Citation 1) (Citation 2)**. *Additionally, not all statistical methods considered produce posterior probabilities and the comparison of pooled models from multiple imputation is an area ripe for more analysis.* Indeed, others have pooled predictions from various machine learning methods by taking the majority vote **(Zavrakidis) (Citation 2)**, and comparing prediction performance. The combination 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 (Lam and Suen, 1995).

- **Put this Cross-Validation??**

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 Principal 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 (Kemp, 2003); problems include falsely narrow confidence intervals for effects and predicted values (Altman and Andersen, 1989) and multiple hypothesis testing inflating risks of capitalising on chance features of the data (Altman, 1991), such as noise covariates gaining entry into the model when the number of candidate variables is large (Derksen and Keselman, 1992). 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 (Guyon et al., 2002). Both logistic regression with LASSO regularization (Tibshirani, 1996) and the analogous Sparse Discriminant Analysis (Clemmensen et al., 2011) are embedded feature selection methods that are dependent on the classification method.

Principal Component Analysis (PCA) (F.R.S, 1901) is a feature extraction method that is independent of the classification method. The training set are orthogonally transformed into new uncorrelated variables called principal components that are linear combinations of the original variables. Feature extraction is accomplished by selecting the k largest principal 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 (Song et al., 2010). Let C_i be the contribution of a given variable on the principal 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 principal 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

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

The C_j are sorted in descending order where C_1 contributes the most variation to the extracted principal 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, if all variables contributed equally, is selected as important for the analysis.

- **How are the top $C_j's$ chosen?**

The number of principal components retained, k, is based on the proportion of variance retained of the p principal components, where the variance threshold is chosen to be 80%.

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

The number of principal components retained is based on the proportion of variance. If the contribution of the p variables were uniform, the expected value would be $\dfrac{1}{\mathrm{p}} = 0.03$.

For a given component, an observation with a contribution larger than this cutoff could be considered as important in contributing to the component.

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 2.

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.

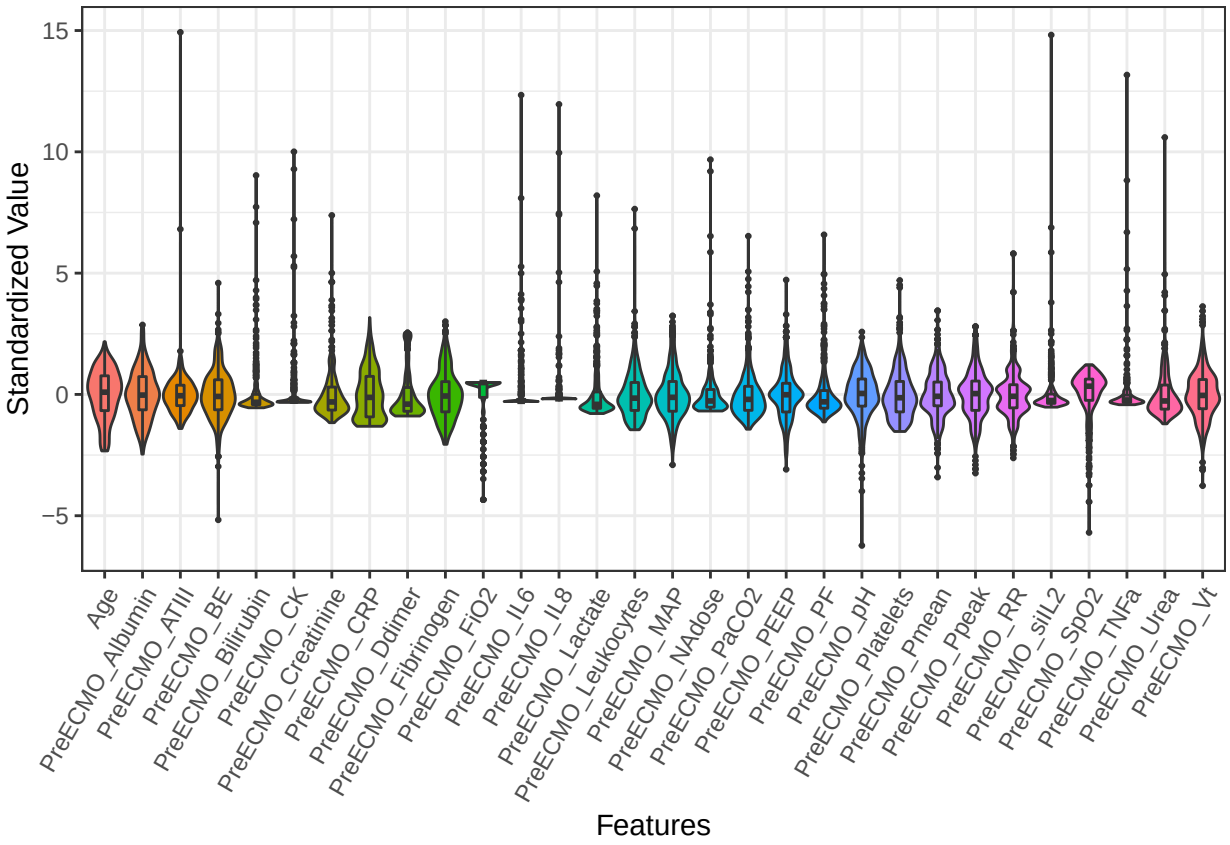

Figure 2: Violin plot of continuous variables.

Many of the standardized continuous variables in Figure 2 are highly skewed with a number of outliers. This can affect the performance of some classification methods that assume a distributional form for the data.

The heatmap in Figure 3 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.

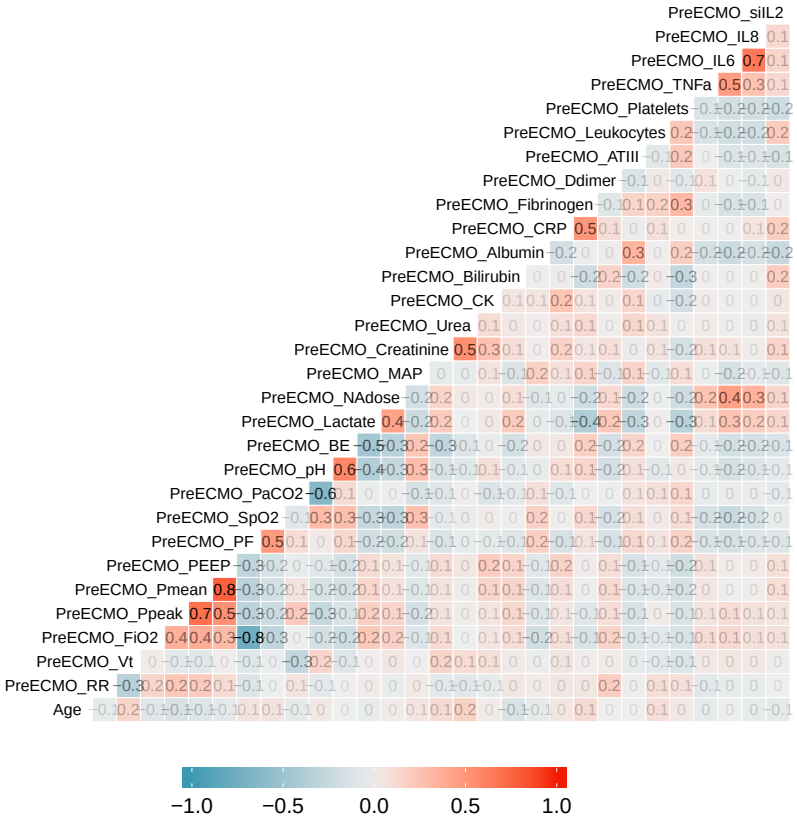

Figure 3: Heatmap of standardized and transformed variables.

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 4 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. **Without more information, this analysis assumes the data is MCAR.**

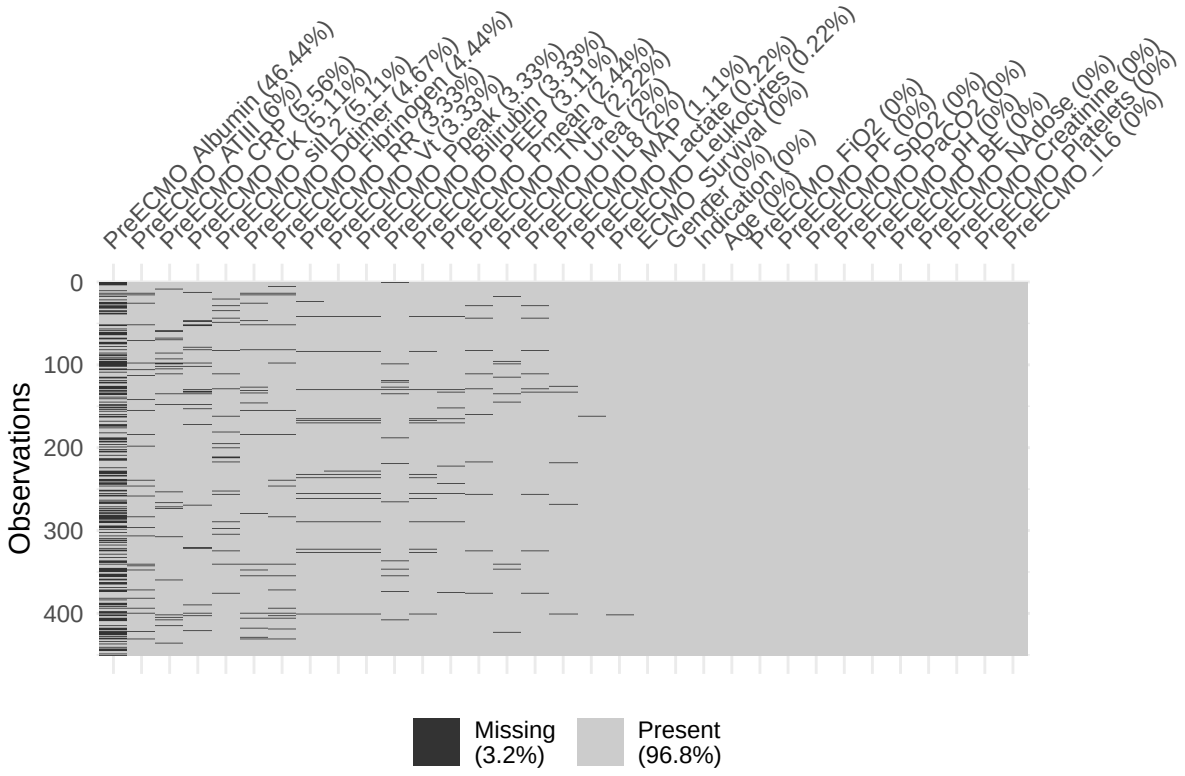

Figure 4: Visual representation of missing observations in the ARDS dataset.

From Figure 4, `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%. **An analysis into how difficult variables are to impute and how helpful variables are when imputing is beyond the scope of this paper but is provided in Appendix C.**

Table 3 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 opposit manner, by dividing the number of pairs $(Y_j,\ Y_k)$ with Y_j observed and Y_k missing, by the total number of complete cells. For a completely observed variable, outflux will have a value of 1 and 0 if completely missing. Outflux gives an indication of how useful the variable will be for imputing other variables in the dataset, while influx is an indicator for how easily the variable can be imputed. Table 3 shows that all variables will be useful during impuation except `PreECMO_Albumin`. A high outflux variable might turn out to be useless for the imputation procedure if it is unrelated to the incomplete variables, while the usefulness of a highly predictive variables is severely

limited by a low outflux value (2012).

Table 3: 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

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 split into 75% training and 25% test with class proportions preserved. The five classification models were trained in 10-fold cross-validation using the ensemble imputation approach. Table 4 shows the averaged Kappa from each analysis in 10-fold cross-validation. In complete case analysis and mean imputaion, LDA is the highest performer. While for predictive mean-matching with $m = 9$ and $m = 99$ logistic regression has the highest averaged Kappa.

Table 4: 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
Complete Case	0.139	0.205	0.038	0.053	0.035
Mean	0.191	0.220	0.040	0.136	0.085
MI (m=9)	0.179	0.124	0.106	0.088	0.136
MI (m=99)	0.185	0.158	0.037	0.127	0.177

4.1.1 Validation on Test Set

Using the parameters values learned in cross-validation, models were fit on the full training set and validated against the test set. 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

At least 16 principal components are needed to explain 80% of the variance in the imputed training data and at least 15 principal components for the complete case analysis. The red dashed lines in Figure 5 indicate the expected average contribution of each variable

Table 5: Pooled performance results of trained models validated on test set.

		Sensitivity	Specificity	Accuracy	Kappa
Complete Case	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
Mean	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

to the selected principal components if each variable contributed equally to each principal component.

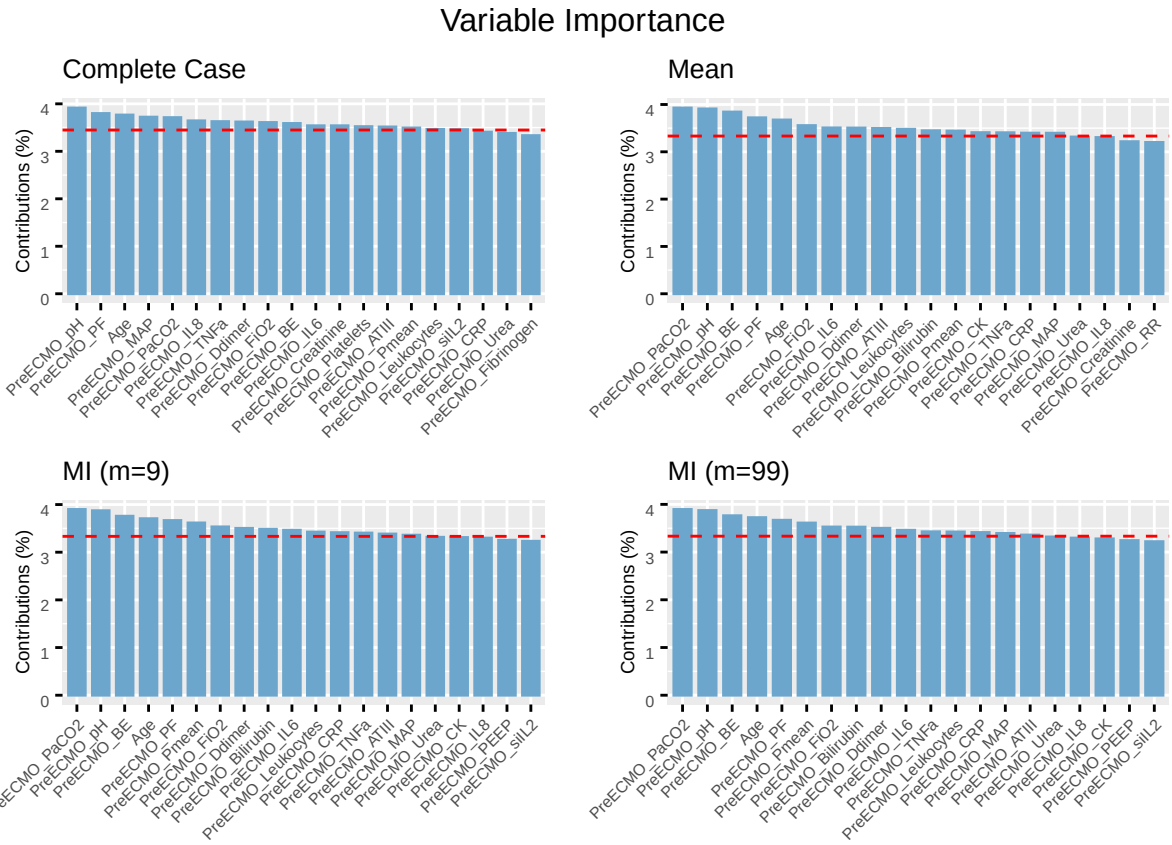

Figure 5: Contribution of variables to the principal components whose cumulative sum explains >80% of the variation in the data.

5 Discussion

Model performance on the imputed datasets were generally better than in complete case analysis. Non-parametric methods, KN 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 performs 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 (Efron, 1975), but LDA is not robust to outliers (Hastie et al., 2009) and Figure 2 shows a number of outliers in almost every variable. Logistic regression is robust to outliers and makes less assumptions than LDA (Hastie et al., 2009) 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 eeven more difficult.

A surprising result is that KNN performed the best in the complete case analysis, $\kappa = 0.161$ and also performed better than the best performing model on MI9 (Logit with $\kappa = 0.153$). On the imputed datasets KNN performed relatively poorly but similarly to LDA. 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.

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 to a 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.

Feature Importance:

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 $K_\text{o} - 1$ 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 $K_\text{o} = 10$ 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 (Hastie et al., 2009). The drawback to nested cross-validation is that the time complexity scales from $O(K_\text{i})$ to $O(K_\text{o}K_\text{i})$. Indeed, the full time complexity for m imputations and a grid search over p parameters would then be $O(K_\text{o}K_\text{i}mp)$.

Variable selection via PCA is independent of the classification method and allows important variables to be identified outside of the classification analysis. Method dependent methods may better select variables **(Citation)**. Regularized logistic regression can select variables through use of the LASSO (Tibshirani, 1996). LASSO methods have also been developed for LDA and QDA Sparse Discriminant Analysis (Clemmensen et al., 2011) and DALASS (Trendafilov and Jolliffe, 2007). Random forests naturally select important variables by _____ **(Citation)**.

- Predict with selected features
- Pool posterior probabilities of predictions

5.2 Conclusion

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

Appendices

A. Additional Exploratory Data Analysis

B. Algorithms

5.2.1 Random Forests Algorithm

The random forests algorithm depicted is adapted from (Hastie et al., 2009).

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

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

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

Algorithm 1: Random Forest Classifier

5.2.2 MICE Algorithm

The MICE algorithm is adapted from (van Buuren, 2012).

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

Algorithm 2: Multiple Imputation via Chained Equations

5.2.3 Majority Vote

(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.

C. Additional Missing Data Diagnostics

5.2.4 Visual Insepction of Imputations

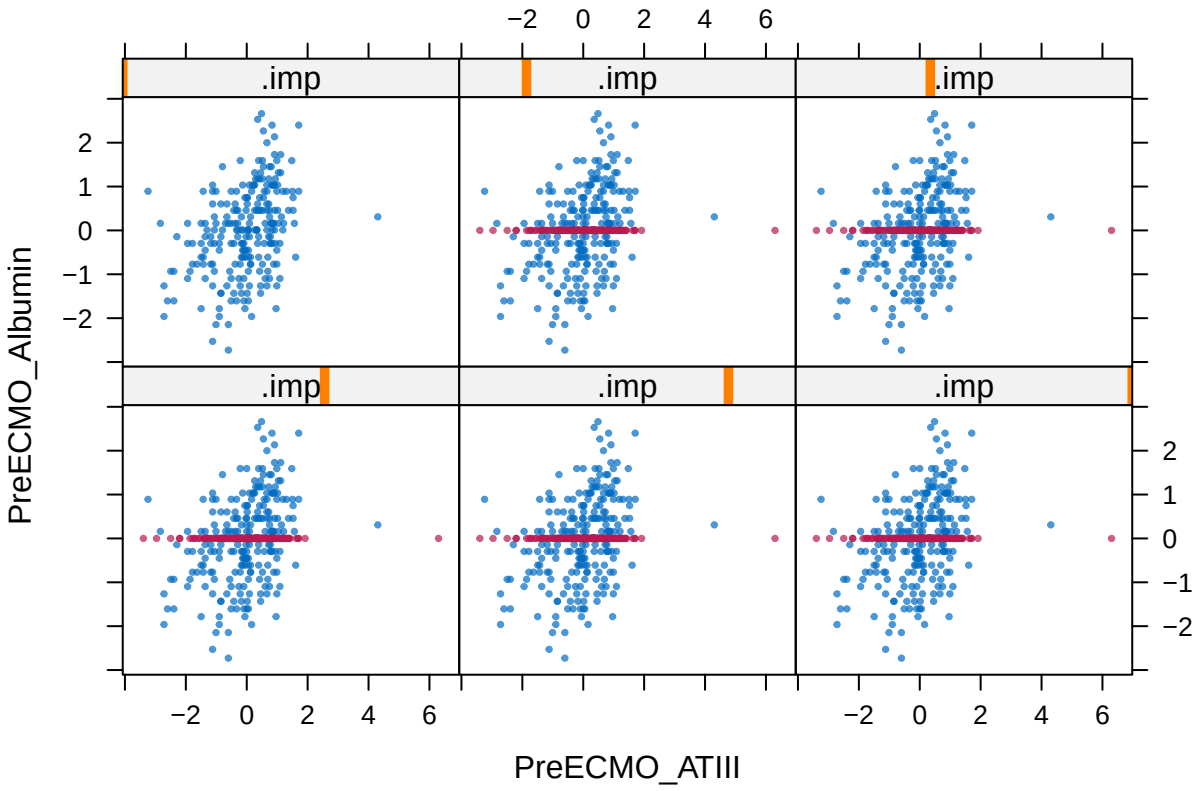

Figure C1: Scatterplot of each imputed dataset

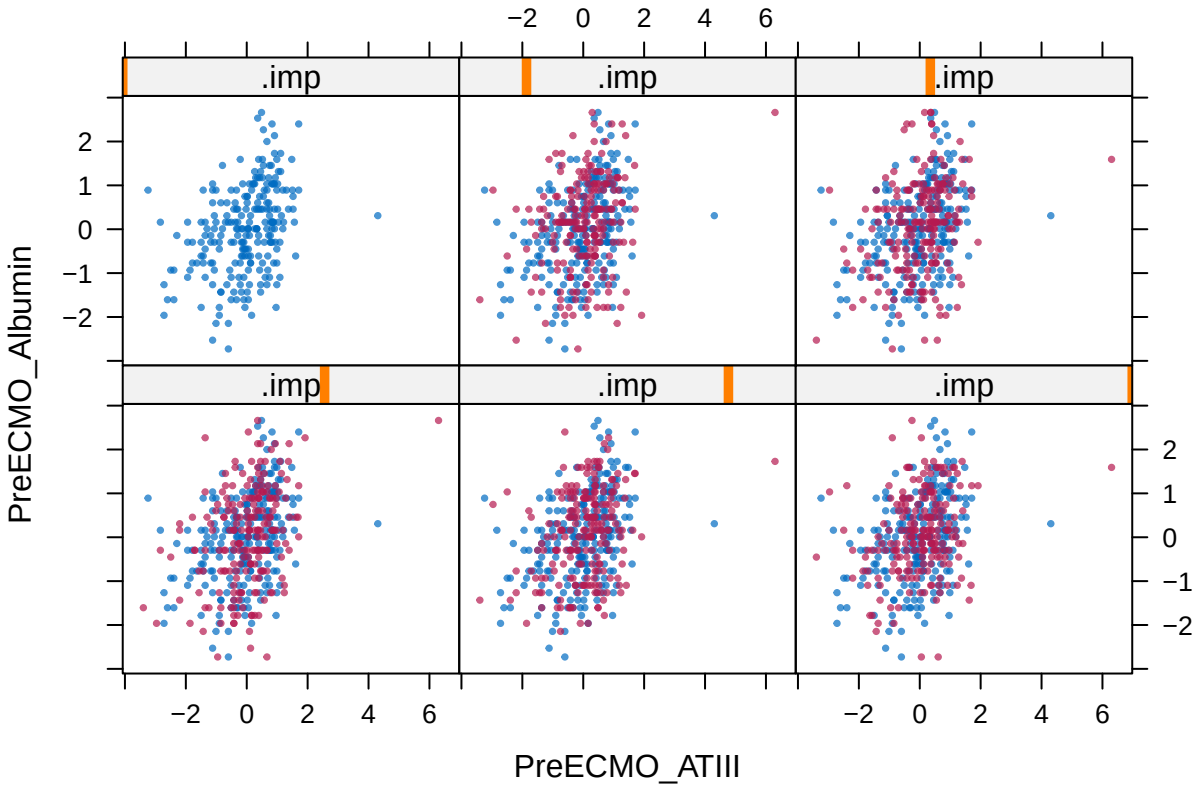

Figure C2: Scatterplot of each imputed dataset

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

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

5.2.5 Convergence Monitoring

- **Plot of convergence**

D. Feature Selection

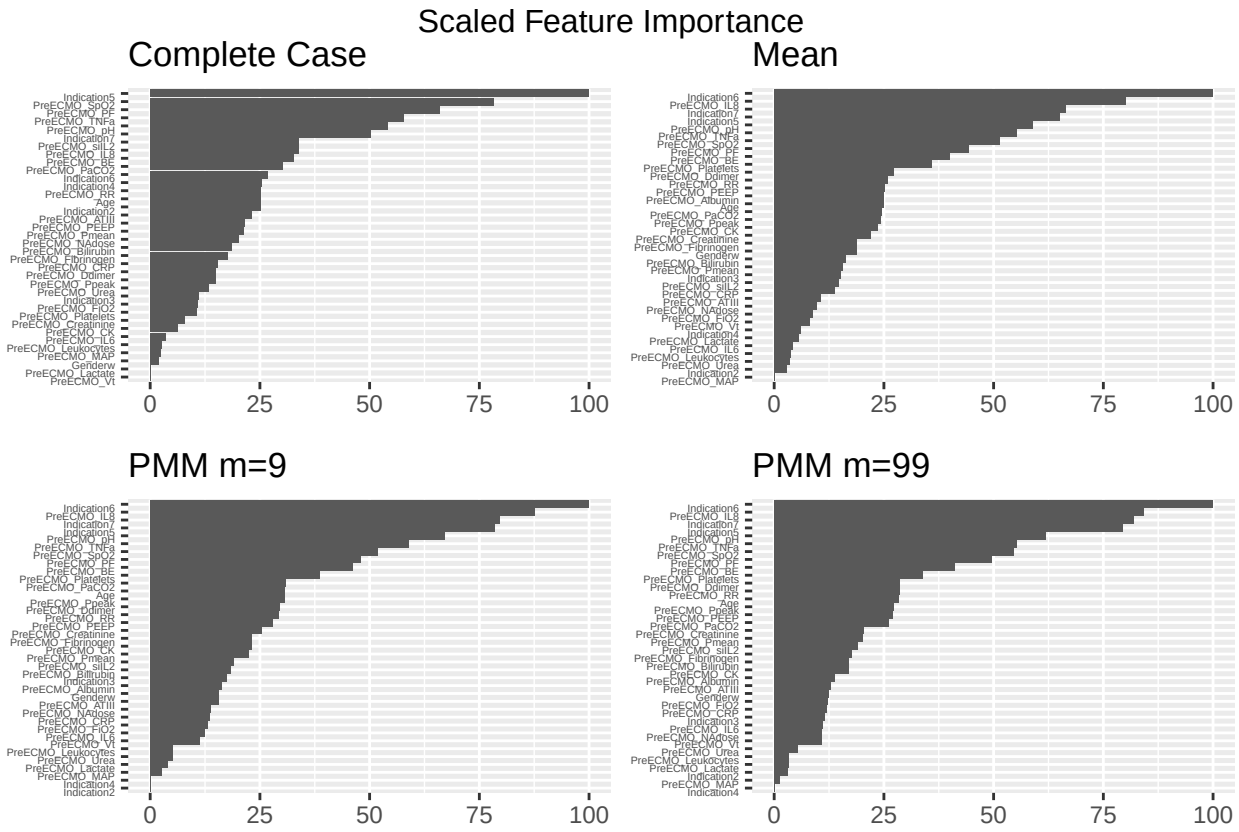

Figure D1: Ordered feature importance from Logit model

E. 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`.

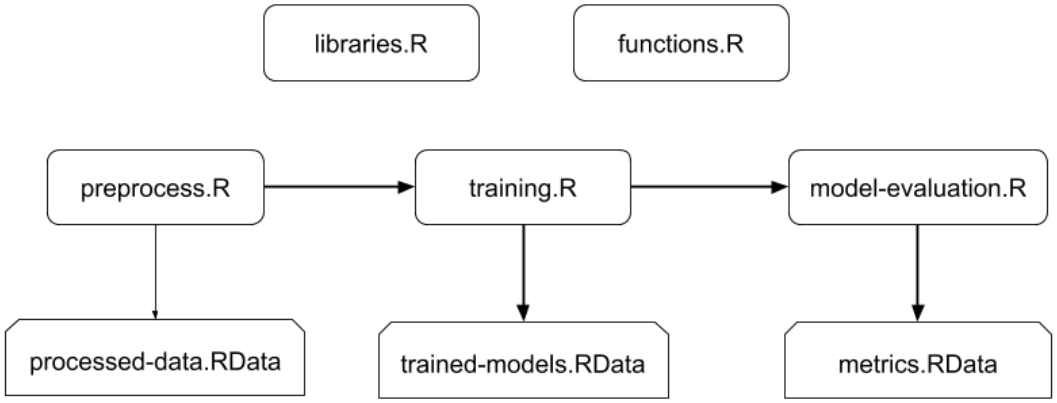

Figure E1: Flowchart of code structure.

F. OLD PLOTS & FIGURES

Table E1: 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 E2: Mean imputation accuracy metrics (m=1). The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=11, respectively.

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

Table E3: MICE via predictive mean matching accuracy metrics (m=9). The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=5 and mtry=13, respectively.

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

Table E4: MICE via predictive mean matching accuracy metrics (m=99). The tuned hyperparameters for K-Nearest Neighbors and Random Forests are K=13 and mtry=15, respectively.

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

References

Altman, D. (1991). *Practical Statistics for Medical Research*, volume Chapter 12. Chapman & Hall: London.

Altman, D. G. and Andersen, P. K. (1989). Bootstrap investigation of the stability of a cox regression model. *Statistics in Medicine*, 8(7):771–783.

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

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

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

Cover, T. M. (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.

Derksen, S. and Keselman, H. J. (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.

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

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

F.R.S, K. P. (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.

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

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer.

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.

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

Little, R. J. A. and Rubin, D. B. (2014). Bayes and Multiple Imputation. In *Statistical Analysis with Missing Data*, pages 200–220. John Wiley & Sons, Ltd.

RUBIN, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Song, F., Guo, Z., and Mei, D. (2010). Feature Selection Using Principal Component Analysis. In *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, volume 1, pages 27–30.

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

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

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

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

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