Handling of missing data for data-driven early warning scores

Veldhorst, Friso J.D.

Award date:
2024

Handling of missing data for data-driven early warning scores.

Bachelor End Project

F.J.D. Veldhorst
1510290

Supervisor
prof.dr.ir. Uzay Kaymak
Second Supervisor
MSc Tom Bakkes

Final version

Eindhoven, May 2024

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Computer Science

Contents

1 Introduction

1.1 Background information on data-driven early warning scores

Hospitalized patients who are outside of the ICU (Intensive Care Unit) and whose condition deteriorates are considerably at a risk of death or at least substantially increased adverse outcome rates [3]. To combat this, Kaiser Permanente Northern California (KPNC) has developed and implemented a predictive data-driven program called the Advance Alert Monitor (AAM). The AAM model is designed to give hospital employees 12 hours of extra time before clinical deterioration [3].

As it stands, Catharina Ziekenhuis Eindhoven (CZE) wants to implement a system inspired by AAM, but considering that the AAM is implemented and used by KPNC, some potential issues arise. First of all, it is likely that the electronic health record (EHR) data that CZE uses is incomplete. If this is the case, this missing data would have to be dealt with in a way that allows for improvement within the AAM-inspired system. It is important to try to find the best way of handling this missing data to gain as much of an improvement as possible.

After finding ways of handling missing data, a way needs to be found to evaluate which method dealing with the missing data is the best. Moreover, the data might also need to be normalized. All this is done to try to find the best way to improve the data, by imputing more data and preprocessing it, which then consequentially would improve the output of the Catharina Ziekenhuis Eindhoven AAM model (CAAM).

1.2 Research questions

Following from the context of subsection 1.1, the research question that we will answer is the following.

To what extent does the input data that comes from the electronic health record of Catharina Ziekenhuis Eindhoven have to be preprocessed, looking specifically at different methods of imputation and normalization, to improve the CAAM model?

To answer this, multiple sub-research questions have been made to divide the question into more clear parts:

1. What do the EHR data and the CZE model look like using exploratory analyses? To get a general idea of the way the data looks, how much data is missing per feature and what kind of features are these?
2. Considering Catharina's electronic health record data is incomplete, what methods are there to make this data complete and how is this done? The main goal of this research is to find the best way to deal with missing data in this data set. To be able to find this, multiple methods of dealing with missing data are tried.
3. How does one evaluate the methods from sub-question 2 to find the best method as a solution? After finding the multiple methods of imputation, it is important to find a way to compare them. Considering these imputation methods can be evaluated in multiple ways, which one is the best, or which combination of evaluation metrics is?
4. Should the data be normalized, taking into consideration both normalizing the data before and after the imputation step and what normalization to apply?

1.3 General methodology to find answers to 1.2

First of all, an exploratory data analysis (EDA) is done of the EHR data used by CZE. This is done to get some preliminary information about the data, ranging from simple things like how much data and what data is missing, the dimensions of the data set, the correlations of the missingness between different features, figuring out the type of randomness of the missing data and what kind of features the features are. Moreover, a look will be taken at the CZE AAM model to figure out the way it works.

For sub-research question 2, we will be looking for multiple methods of dealing with incomplete data to see what works on the EHR data from CZE. Firstly, it is important to note that we only look at imputation methods, as deletion methods (listwise- and pairwise deletion) cause a large loss of data and reduce the statistical power and accuracy [18]. The imputation techniques range from heuristic methods, simple statistical approaches and more advanced techniques such as machine learning [**?**]. The idea behind this sub-question is that it allows for multiple ways of imputing missing data which means these methods can be compared in sub-question 3 to figure out the best method of imputation for this particular data set.

As for research question 3, we want to do a quantitative analysis of all the imputation methods found as a result of sub-question 2 to figure out which one works the best on the AAM-inspired model. As there is not one standardized method of evaluating imputation methods [11], multiple different evaluation metrics are tested. Between those different evaluation metrics, we are hoping to find the best, or at least a better imputation method than the standard imputation method.

Lastly, for sub-research question 4, depending on the methods of imputation used, it might be needed to normalize the data beforehand. One example of this is an imputation method relying on distance metrics. Otherwise, normalization after imputation might also be valid when the absolute value is important for the imputation. This could be useful to look at to improve the pre-processing even more. A summary of the methodology is found in Figure 1.

1.4 Scope of research

To be more specific about this research, it is limited in some ways which will be specified further. Firstly, the source of the data and data collection is the electronic health records data from Catharina Ziekenhuis Eindhoven. Only the data from patients in a ward will be looked at. Moreover, only data from CZE are used. As for the wards in the different departments that are being used, most of them are used except for the children's ward and the pregnancy ward due to differing physiologies. The psychology ward and GGZ will also not be used as they encompass different types of healthcare.

As for types of data, there are 2 types of missing data: not recoverable- and recoverable data. Not recoverable data is intentionally missing data due to the patient being disconnected from a measuring system. Recoverable data is data that is missing by accident due to malfunctioning of the measuring system for example. Both these types of data will be considered within the scope of this research as it isn't possible to differentiate between them both [**?**].

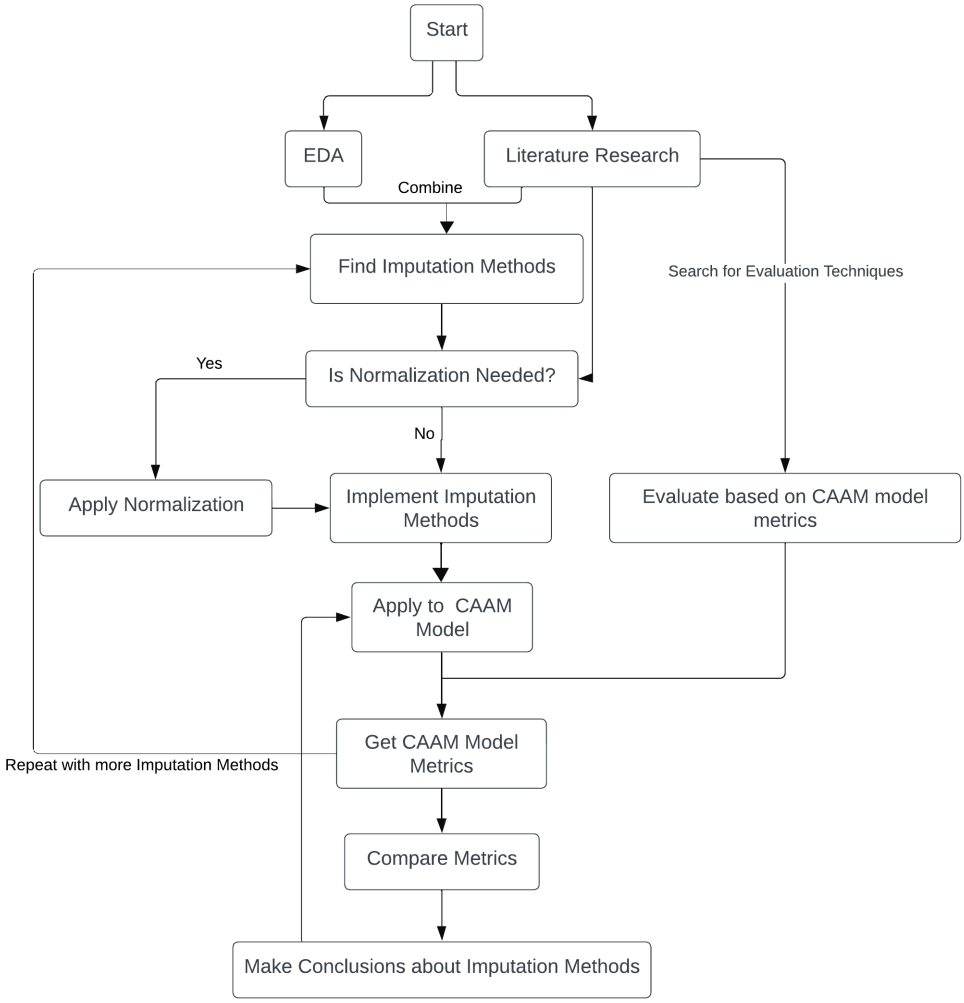


Figure 1: Methodology

2 State of the Art

As there is already a lot of research on this broad subject, it can be hard to find research that specifically can be applied to the type of (missing) data used in this project. Considering this, we tried to summarize the bits of information from the research papers that seemed useful in this case. This ranges from understanding the reasoning behind this research in subsection 2.1 to figuring out what the causes of the missing data are using subsection 2.2. On top of that, a look was taken at the way imputation and normalization were handled in a similar use-case in subsection 2.3. In subsection 2.4, it is discussed how imputation methods could or should be evaluated.

2.1 Escobar - Automated Identification of Adults at Risk for In-Hospital Clinical Deterioration [3]

This paper aims to help with the fact that hospitalized adults whose condition deteriorates while they are in wards have considerable mortality. Usually, they rely on manually calculated scores to identify patients at risk of deterioration.

With the help of a validated model that uses information from electronic health records which allows for automated risk-score calculation, an intervention program was developed involving remote monitoring by nurses to communicate which patients were identified as high risk by the AAM-model to a rapid-response team on location. Consequentially, the outcome - being the mortality within 30 days after an alert - was compared. Hospitalized patients, excluding those in the ICU, whose condition reached the threshold, where the system was operational were compared to the same kind of patients who were at a hospital where the system was not operational, yet their condition would have triggered the system had it been in place.

After implementing this program at 19 hospitals in a staggered fashion, 548,838 non-ICU hospitalizations involving 326,816 patients were identified. 43,949 of those hospitalizations (involving 35,669 patients) were cases where the condition of the patient reached the alert threshold. The mortality rate within 30 days after an alert was lower for the intervention cohort when compared with the cohort where the system was not in place. The adjusted relative risk was 0.84 with a 95% confidence interval and 0.78 to 0.90 had a $P < 0.001$. This means that the risk of mortality is 16% lower for a patient who is in the intervention cohort that produces an alert compared to a patient who is in the comparison cohort that does not produce an alert [10].

This paper makes the following conclusion: the use of an automated predictive model to identify high-risk patients in wards, such as the AAM model, combined with interventions by rapid-response teams based on the model is associated with a decreased mortality rate. This gives reason to believe that implementing the CAAM model could prove to be fruitful.

2.2 Federico - Preprocessing and misclassifying issues in clinical data sets for prediction and intervention [1]

Missing data can be dealt with in 2 ways. Firstly, one could delete all variables corresponding to a given sampling time if at least one of them is missing. Secondly, one could impute values for the missing data in several different ways. However, before working on dealing with the missing data, one has to first identify what the cause of the missing data it is. As Federico mentions in his thesis, there are two causes of missing data.

1. Not recoverable missing data: Data that is intentionally missing (eg. Patients could be disconnected from the ventilator for several hours, and all variables measured by this assisting and

monitoring system would not be recorded during that time).

2. Recoverable missing data: Data that is accidentally missing (eg. accidental disconnection/malfunctioning of sensors, errors in the communication with the server/storing facility, accidental omission of data registration by humans, electricity failures, and unlabeled samples that cannot be associated with any specific experiment.)

Moreover, as Federico mentions, a generally accepted guideline for managing missing data, is that when it represents more than 10% of the total information expected to be present, all records with missing values can be deleted without a significant loss of statistical power in the modeling results based on such dataset. If the missing data is more than 10%, then an imputation strategy can be used since deleting would result in a significant loss of statistical power.

2.3 Kipnis, P. - Development and validation of an electronic medical record-based alert score for detection of inpatient deterioration outside the ICU [5].

In this article, Kipnis discusses the method of making the AAM model. A discrete-time logistic regression model was used to obtain an hourly risk score to predict unplanned transfer to the ICU within a timeframe of 12 hours. This model was based on all hospitalization episodes where the patients were 18+ years old and admitted to a KPNC hospital between 2010 and 2013. Moreover, the patient's initial hospitalization had to occur at a KPNC hospital; patients hospitalized due to childbirth were excluded and the EMR had to have been operational for at least 3 months. The performance of the AAM model was evaluated against two other automated early warning score systems (EWSs) by sensitivity, specificity, negative predictive value, positive predictive value and the area under the receiver operator characteristic curve (AUC).

The data set used had 48,723,248 hourly observations with 378,838 patients who met the inclusion criteria. Predictors that were included are similar to the Catharina dataset, being physiologic data (laboratory tests and vital signs); neurological status; severity of illness and longitudinal comorbidity indices; care directive and health service indicators (ie. Elapsed Length of Stay). The AAM model had a better performance than the other 2 early warning score systems (EWSs): the AUC was 0.82 compared to 0.79 of eCart and 0.76 of NEWS (the other 2 EWSs).

The conclusion was that the AAM score is an example of a score that takes advantage of the multiple data streams within EMRs these days. That said, the big challenge right now is detecting those patients whose data is sparser.

By looking at the way the researchers pre-processed their EHR data, and especially the way they handled missing data, some conclusion might be made for this research. Firstly, this article mentions that within-patient clustering (random) effects can be ignored. Given the large amount of data that was available for Kipnis, they found that within-patient clustering showed minimal effects on their results. Considering the data used for this research is also quite rich, it seems logical that the same assumption can be made for this research - that within-patient clustering (random) effects can be ignored.

Figure 2 shows the transformations that the researchers of the original AAM model used for all of the statistical modeling and machine learning approaches. They fitted each transformed variable into a univariate discrete-time logistic model after which they selected the transformations that gave the best fit to the model. This way of normalization could potentially be used before or after imputation in the CAAM model.

Table 1
Predictors employed for model development.

Predictor	Values or transformation[a]
Laboratory tests[b]	
Anion gap	Linear
Bicarbonate	Quadratic
Glucose	Linear
Hematocrit	Cubic
Lactate	Linear
Log blood urea nitrogen	Linear
Log creatinine	Quadratic
Sodium	Linear
Troponin	Linear
Troponin missing flag	Indicator
Total white blood cell count	Linear
Vital signs[c]	
Latest diastolic blood pressure	Quadratic
Instability[d] of systolic blood pressure	Linear
Latest systolic blood pressure	Cubic
Latest heart rate	Cubic
Log heart rate instability	Quadratic
Log oxygen saturation instability	Linear
Logit[e] latest oxygen saturation	Cubic
Logit worst oxygen saturation	Linear
Log respiratory rate instability	Linear
Log temperature instability	Quadratic
Latest temperature	Quadratic
Latest respiratory rate	Cubic
Worst respiratory Rate	Linear
Latest neurological status	Linear
(Anion gap ÷ serum bicarbonate) × 1000	Linear
Shock index (latest heart rate ÷ latest systolic blood pressure)	Linear
Composite indices[f]	
LAPS2	Cubic
LAPS2 at hospital entry time	Linear
Log COPS2	Linear
Other	
Log transpired length of stay[g]	Linear
Logit age	Quadratic
Sex	Male indicator
Care directive[h]	Full code or not full code
Season	Season 1: months 11, 12, 1, 2 Season 2: months 3, 4, 5, 6 Season 3: months 7, 8, 9, 10
Time of day	Time frame 1: 01:00–07:00 Time frame 2: 07:00–12:00 Time frame 3: all else
Admit category	1 ED[i], SURGICAL 2 NON-ED SURGICAL 3 ED MEDICAL 4 NON-ED, MEDICAL
Hospital	20 hospital indicators
Log (transpired length of stay X LAPS2)	Linear

Figure 2: Kipnis Paper

2.4 H. I. Oberman - Toward a standardized evaluation of imputation methodology [11].

The idea behind imputation is to fill in missing values to complete a data set – after which the complete data goes through some sort of data science pipeline to get to a goal like making a prediction model. More often than not, the goal of creating some model overshadows the imputation that was done to get a full data set. Whilst this is reasonable, it seems that some more attention should also go to the imputation step as this can cause improvements in the completeness of data sets and thus in data models. Recent attention on imputation has shown there is no standardized way of evaluating imputation methods yet. Due to the fact that a 'golden standard' is absent, this paper tries to fill this hole in 3 ways. Firstly, the paper tries to raise concerns with respect to evaluating imputation methodology. Secondly, it provides a suggested course of action when using simulation studies to evaluate imputation techniques. Lastly, it tries to start a discussion for the sake of progress in this field.

This paper is primarily about simulation studies where there is a form of "comparative truth" to assess the inferential validity of imputation methods. This means it does not really apply to this study given that this is a simulation study aimed at comparing the predictive performance of imputation and prediction methods together. In this type of design, where the imputation and prediction method pairs are evaluated on their ability to yield a high accuracy, only the comparative performance is established. In case the inferential validity of the imputations is of interest, trying to get the comparative performance is not recommended. That said, the paper does mention that if there were to be multiple imputation methods under evaluation, it is important that all these imputation methods are applied to the same incomplete data set as this is computationally convenient, minimizes unnecessary variation and makes for fairer comparisons.

When trying to evaluate imputation methods on empirical data, there are a few mistakes that can be made if one wants to get the inferential validity of the imputation methods. Empirical or real-world data contains almost always some missing values which needs to be dealt with before the data can serve as comparative truth. While it may seem intuitive to only draw complete cases from a large data set to yield complete samples, unfortunately, there may be inherent differences in relations between cases with and cases without missing values because of the unknown missing data model. Also, just drawing from complete cases could result in a large loss of data from the data within incomplete cases. Another way to deal with missingness is to impute on the incomplete data set once to create a new data set. Using the new data set, imputation methods could be applied to the incomplete data set and then the results could be compared to the complete data set. This does not work as bias towards the initial imputation method may be introduced. Leaving the missingness as-is and introducing new missing values to impute on also does not work as you would not have a real and unbiased comparative truth.

Oftentimes, diagnostic evaluation of imputation methods is left out of simulation studies. Identifying problems with an imputation method may offer explanations for its underperformance. It might be meaningful to look at the parameters of the method before switching to another one. Next to that, it should be noted that imputation method performance should be measured depending on the specifics of the study, whether the goal is inference or prediction. In the case of inference, the standard errors of the estimates should be correctly calculated, which requires multiple imputation. This method aims not to reproduce the data, but rather obtain a valid inference considering the data is missing. From all of the methods of evaluation, using RMSE is generally not recommended as it does not account for the inherent uncertainty of missing values and may inflate the type I error of statistical inferences.

Lastly, the paper asks simulators to consider the following aspects.
The presence of convergence in the imputation process is the minimum requirement for any imputation method. However, there is some preliminary work that suggests that iterative imputation algorithms could gain inferential validity without reaching convergence [12].
The distributions of the imputations should be checked for anomalies and if they are present, it should be possible to explain them. The plausibility of imputed values should also be evaluated. Whilst not necessary for obtaining valid inference, it might be desired, especially when working with others.

3 Methods

3.1 Exploratory Data Analysis

At the start of any data science project, it is important to first get an understanding of what the data looks like. This is also called an exploratory data analysis (EDA). After making sure the data can be used in a notebook on the Catharina AI environment and certain packages such as pandas, numpy and sklearn were installed, multiple statistics were looked at such as the dimensions of the data and the number of unique patients. Moreover, it felt important to figure out which columns had missing data and also what the absolute- and relative amount of data that was missing per column. Through the use of the Python package Missingno, it is possible to find relations between the missing data of different columns and visualize them effectively as can be seen in Figure 3 and Figure 4. These figures show how closely related the features are. For Figure 4, this means the more red/the higher the number, the more closely related the 2 features and for Figure 3, the closer together they are to the right, the more closely related the features are. It is not the case here, but sometimes one can find interesting patterns within a heatmap or dendrogram which could be reason to apply a different method of imputation to those specific features.

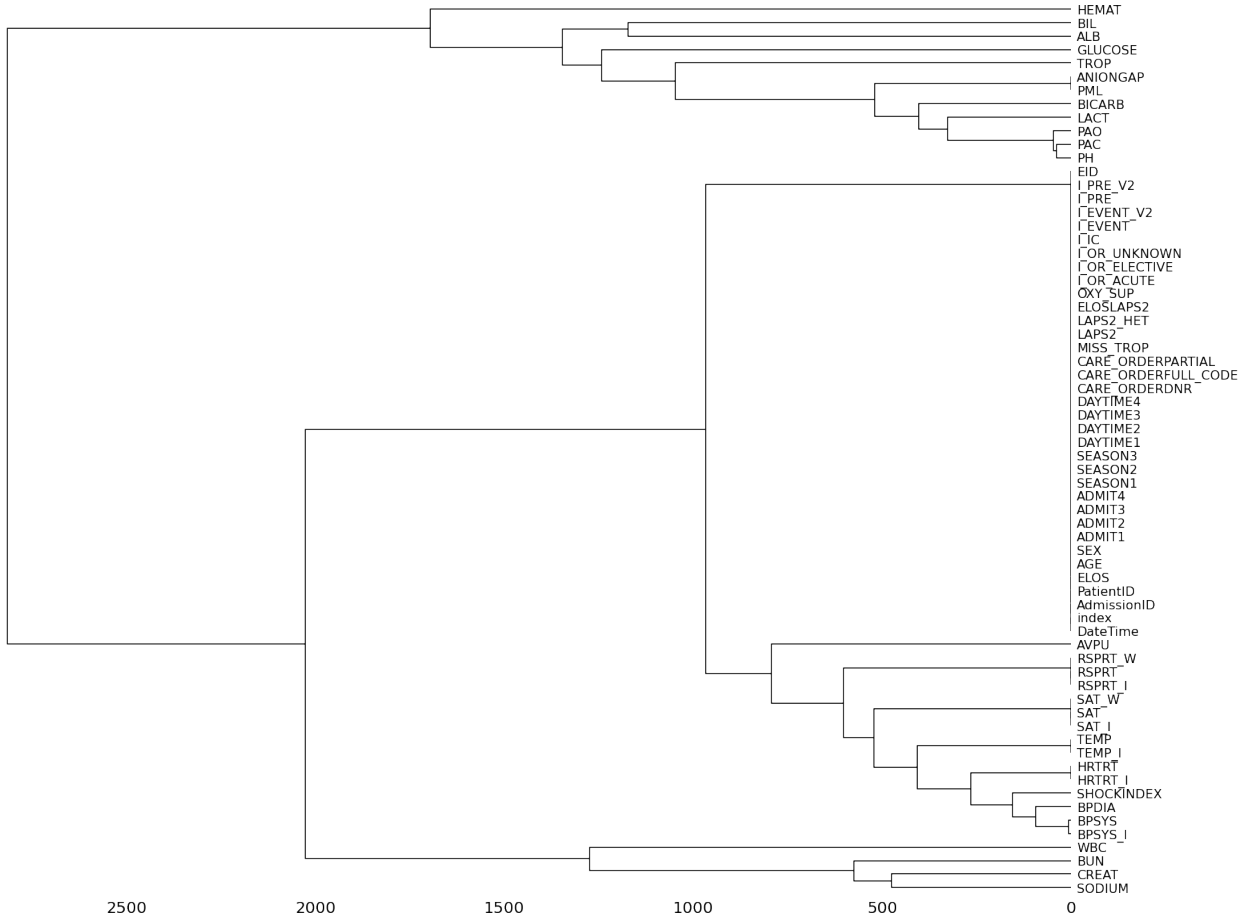


Figure 3: Dendrogram showing relations between variables

Figuring out the way the columns were different was needed as different imputation techniques can only be applied to certain types of data. These different feature types of data can be categorized as numerical, where the values can be measured on a scale, and categorical features, which are values that can be grouped into categories (ie. gender, color and zip code). For numerical, the features can be discrete or continuous; discrete features are countable whereas continuous features are data within some interval. As for categorical features, these can be nominal or ordinal. Data is nominal when the feature has no natural ordering and ordinal when it does have a natural ordering [2].

Apart from looking at all the features, recognizing what kind of missing data there is within the data set is vital. There are three kinds of missing data: Missing at Random (MAR), Missing Not at Random (MNAR) and Missing Completely at Random (MCAR). Imagine a dataset consists of the assessed feature Y and the rest of the features X. In this case, missing data of some variable Y is MCAR when the missing values are independent of both itself (Y) and the other features (X). It is MAR when the missing data of Y is dependent on X and MNAR when the missing data is dependent on Y itself [**?**].

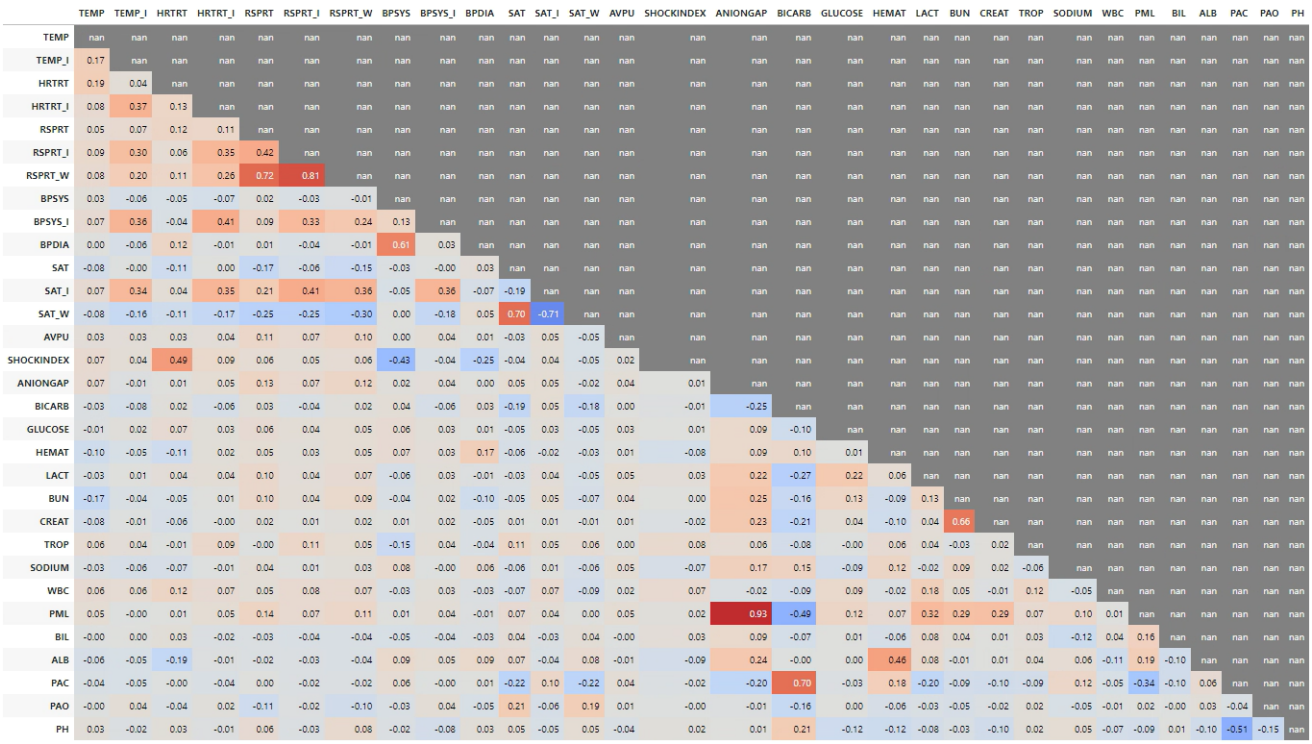


Figure 4: Heatmap of missing value correlations between different variables

3.2 Data imputation methods

As mentioned in subsection 1.3, a few different imputation methods were looked at to compare them. First of all, it is important to at least try simple methods such as mean or median imputation as these do not take much memory, can provide a simple baseline to compare to more complex imputation methods and are simple to do using SimpleImputer from the sklearn package. They fill in the missing data per feature with their respective statistics. These two statistics were calculated from

all present data within the given feature. Now based on the results from the EDA as can be seen in Figure 5, certain other simple methods fit different kinds of data. The table assesses two key aspects: whether the method is unbiased on average and if it calculates the correct standard error. It evaluates unbiasedness concerning mean, regression weight (with the incomplete variable as dependent), and correlation estimates. Each method's assumptions on the missing data mechanism are outlined for unbiased estimation. For example, listwise deletion yields an unbiased mean estimate only under the assumption of Missing Completely At Random (MCAR), but results in an overestimated standard error. To potentially get an even better result, more complex imputation techniques can be tried.

	Unbiased			Standard Error
	Mean	Reg Weight	Correlation	
Listwise	MCAR	MCAR	MCAR	Too large
Pairwise	MCAR	MCAR	MCAR	Complicated
Mean	MCAR	–	–	Too small
Regression	MAR	MAR	–	Too small
Stochastic	MAR	MAR	MAR	Too small
LOCF	–	–	–	Too small
Indicator	–	–	–	Too small

Figure 5: Overview of assumptions made by ad-hoc methods [19]

3.2.1 Multiple Imputation Settings

This method, as proposed by Rubin, fills in missing values by generating plausible numbers derived from distributions of and relationships among observed variables in the data set [14]. It differs from single imputation methods as data is imputed many times with different plausible values estimated from a distribution. This gives a quantification of the uncertainty that is always present in estimating missing values. As such, it provides relatively accurate estimates of quantities or correlations of interest, such as treatment effects, sample means, correlations between two variables and the related variances. Due to this, the chance of false-positive or false-negative conclusions is reduced [7].

Using IterativeImputer from the sklearn package [13], it was possible to use the MICE (Multiple Imputation by Chained Equations) algorithm. It is important to note that this method of imputation is computationally significantly more expensive than the more simple methods mentioned earlier. The running time of IterativeImputer is $\mathcal{O}(knp^3min(n,p))$ where k = max_iter, n = the number of samples and p = number of features. Since p is cubic, it is the most computationally costly. Because of this, n_nearest_features was set to 15, lowering the amount of features used from 56 to 15. max_iter was determined by setting it to the average percentage of missing data [19]. Using Figure 6 the percentages of missing data can be summed to P. Out of the 66 total features, 12 features were dropped and 1 feature was hot-encoded into 3. The total amount of features used for the model is thus $66-12-1+3=56$ which means $\frac{P}{56}=25\%$ of the data is missing, so max_iter was set to 25.

At the start, IterativeImputer fills in all the data by default with the mean. After that, it uses some

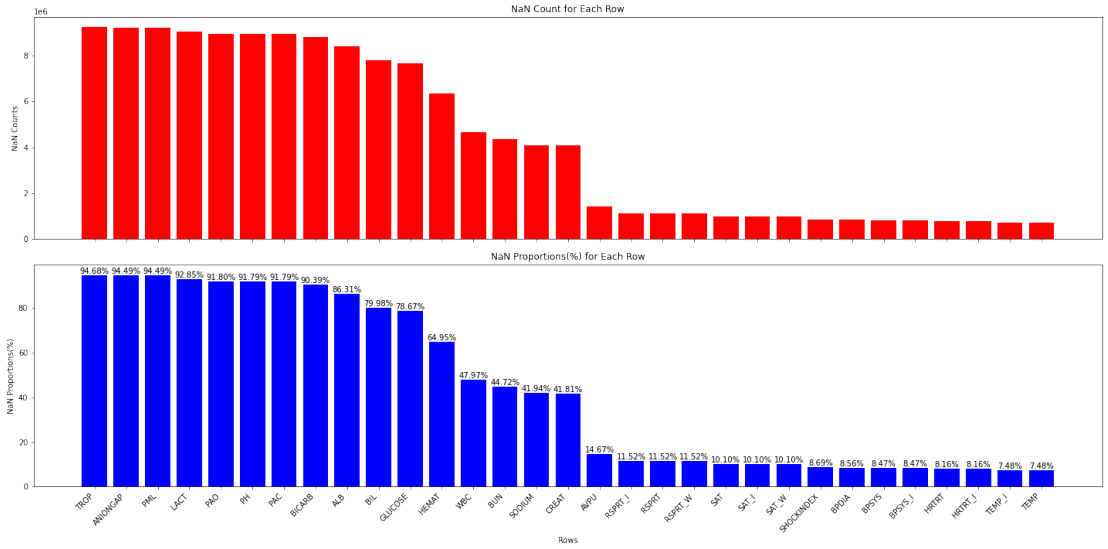


Figure 6: Missing values per feature of the data set

estimator to predict the missing data in column X by using all columns except for the column X itself. This is reiterated untill the algorithm converges. In the case of this dataset, BayesianRidge (which is the default), LinearRegression and RandomForest will be tested as estimators, all gotten from either sklearn.linear_model or sklearn.ensemble. The random state used is 0 for the sake of reproducibility.

3.2.2 Different Estimators Settings for MI

For the BayesianRidge estimator, the default settings from sklearn.linear_model were used.
All three of LinearRegression, RandomForest and KNeighborsRegressor estimators had n_jobs at -1 to use multiple CPUs to decrease the time it took to train and evaluate the model.
For RandomForest, the settings were the following: n_estimators=10, max_depth=10, min_samples_split=10, min_samples_leaf=5, max_samples=0.5 and random_state=42.
Regarding the KNeighborsRegressor, there are multiple different ways of defining the k. Cross-validation is the best way of estimating the optimal k but there is no time for that within this research. Since there is also no way of basing the k on the context and the $\sqrt{N}$ rule would give k = $\sqrt{9.757.973} \simeq 3124$, which is computationally too expensive, the elbow method will be tried using the following set of k's: [5, 50, 500]. Lastly, for the algorithm parameter, there are three possibilities: Brute force, Ball tree and KD Tree. Consider number of samples N and dimensionality D. Brute force grows as $\mathbf{O}(DN)$, Ball tree grows as $\mathbf{O}(D\log(N))$ and KD tree at $\mathbf{O}(D\log(N))$, where for a larger D it almost grows to $\mathbf{O}(DN)$. To be as efficient as possible, the Ball tree was selected considering the high dimensionality of the data set used. Lastly, a normalization was applied using the MinMaxScaler from sklearn.

3.2.3 KNeighbors Regression

One of the most simple and straightforward approaches to ML-based imputation, K-Nearest-Neighbors (KNN) imputation aims to predict missing values based on the nearest neighbors of a given missing value point. At first, we wanted to use KNNImputer, but as it can not use multiple CPUs at the same time, we wanted to switch to another KNN imputation method. This other method was KNeighbors

Regression: it predicts a target by local interpolation using KNN. In order to do this, some parameters need to be determined first and the data has to be normalized. Regarding the parameters, the distance metric that will be used in this research is Euclidean distance as this is used by default. The algorithm used to compute the nearest neighbor distances has to be defined and lastly, the k value, which defines the number of neighbors that will be checked to determine the classification of a given missing value point. Too high of a k value causes high bias and low variance and vice versa. There are multiple ways of finding the appropriate k [6]:

- Cross-Validation: Test a lot of different k values using cross-validation to find the k that gives the best model performance.
- Elbow method: Plot the performance of a few k values against each other and try to interpolate the peak performance and thus the k.
- Domain Knowledge: Consider the context of the problem and base the k on that.
- $\sqrt{N}$ Rule: As a rule of thumb, k can be decided by taking the $\sqrt{N}$, where N is the number of data points.

Lastly, normalization is also important for this method, this will be further dealt with in subsection 3.3. KNeighborsRegressor will be used through IterativeImputer.

3.3 Normalizing the data

As already mentioned in subsubsection 3.2.3, to apply K-nearest-neighbors (KNN) imputation the data first needs to be normalized in a way that allows KNN to work properly. For this particular method of imputation, MinMax normalization is used as it is a bit more accurate for KNN [4]. The KNN imputer that is used uses Euclidean distance. For this imputer to work properly, all features need to be equally important and so fall within the same range of values. MinMax scaling does this by transforming a value A to B by applying the following formula:

$$B = \frac{A - min(A)}{max(A) - min(A)}$$

In research by Kipnis et al. they applied some sort of normalization to all of the continuous columns. They considered cubic splines, log and polynomial transformations for these features and selected the best transformation for each of the features which can be seen in Figure 2. These transformations could also be applied to the data that is used for CAAM. This would best be done after the imputation such that all the data (the imputed data too) is transformed, which could possibly improve the predictions like it did for the AAM model.

Moreover, a Z-score normalization is used in the current CAAM model after the imputation and before training the model, which uses the following formula where X is the value, μ is the mean and σ represents the standard deviation:

$$\text{Z-score} = \frac{X - \mu}{\sigma}$$

3.4 Analysis of imputation methods

Where there is a plethora of ways to evaluate machine learning models, the literature is not as widespread when it comes to the evaluation of imputation methods. As already mentioned in subsection 2.4, most research has been done on making some sort of model and their evaluation, whilst

quickly skimming over the preprocessing. This makes sense as the model is the goal, but if it is possible to get some incremental gain in the model accuracy from optimizing the preprocessing strategy, perhaps it could be suggested to not glance as quickly over it. In order to get the best preprocessing strategy, there needs to be a method to evaluate the preprocessing.

One of these evaluation methods is to first make a baseline of imputation for the CAAM, such as mean imputation. After this is done, other ways of imputation are tried and fitted on the model, attempting to get certain performance metrics of the model and comparing these to the performance metrics of the model using the baseline imputation method. If the model performs better under another method of imputation, it can be concluded that the imputation method is the best for this model. It has to be noted that this way of evaluating the imputation methods means that establishing the inferential validity of those methods is not possible, i.e. these methods are only better or worse when it comes to the CAAM model specifically [11].

Another way of imputation evaluation is by amputating the data. It means that new missing data masks are created for complete data. The point of this is to use some method to impute again on the missing data masks and then look at the imputed data compared to the original data that was amputed. If this is done with multiple imputation methods, one could try to find the method that comes closest to the original data. Whilst this approach may seem intuitive, it is not a great way of evaluation. As van Buuren has noted, it is not useful to evaluate imputation models purely based on their ability to recreate the true data. Suppose that the RMSE (Root Mean Squared Error) of the imputed values is used to find the difference between the imputed values and the 'real' values. This would mean that the model with the lowest RMSE would be the best. This model would repeatedly give the same imputation values, ignoring the uncertainty of missing values. This leads to biased estimates, unreliable statistical conclusions and a potential increase in the number of false positives [19]. This method ignores the inherent uncertainty of missing values that are needed for real-world data [11].

Considering both methods of evaluation, the first method (Comparing imputation methods based on the performance of the CAAM model) seems more suitable for this research. Usually, the evaluation of clinical prediction classifiers involves various measures such as R2 for overall performance, the p-value for calibration and the AUC. However, this model predicts extremely rare outcomes. A classifier aiming to maximize accuracy may achieve that by classifying all observations as non-events. This means model improvements as quantified by the AUC are heavily overshadowed by the large true negative rate [5]. This means that instead of those measures, the CAAM model should be evaluated using the next few metrics: Sensitivity, Specificity, Positive Predictive Value (PPV) and Negative Predictive Value (NPV). Sensitivity gives the proportion of true positives out of all positives whereas specificity is the inverse of sensitivity, giving the proportion of true negatives between all those that are negative. Similarly to the aforementioned metrics, PPV determines the amount of true positives out of all positive findings whereas NPV determines the count of true negatives out of all negatives [17]. Since the AUC is widely used to measure model performance it was calculated both for episodes and for hours, but do take it with a grain of salt considering the aforementioned large true negative rate. Moreover, the Precision-Recall curve is plotted, which shows the relation between Sensitivity and Precision. Lastly, the W:D (work-up detection ratio, 1) is an important metric for clinicians to assess an early warning score. It tells about the number of patients one would need to evaluate to identify a patient with an event [5]. Moreover, the actual number of alerts is also important for those who respond to the alerts.

$$W : D = \frac{1}{PPV} \tag{1}$$

4 Results

4.1 EDA

4.1.1 Data Description

First of all, the dimensions of the data that has been used are 9,757,973 rows by 66 columns. This means the total amount of data has 644,026,218 values of which 134,042,468 values are missing, which is 20.81%. The way this missing data is split as well as which columns include missing data can be seen in Figure 6 and the more specific data can be found in Table 3.

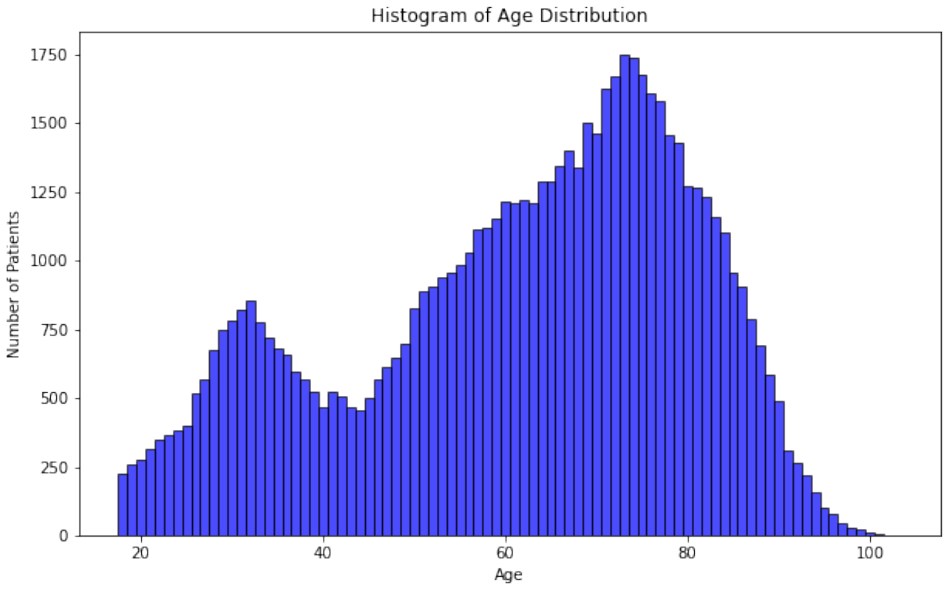


Figure 7: The age distribution within the data set

The roughly 10 million rows are split between 56,609 unique patients with an age distribution which can be seen in Figure 7. The maximum elapsed length of stay for each of these patients can also be seen in Figure 13.

Lastly, it was also interesting to take a look at the correlations of the missing values between different variables. This could allow for finding interesting correlations which could be cause for using a different imputation method. This heatmap can be seen in Figure 4. Using the MissingNo package, it was also possible to create a dendrogram showing these same relations which might be more clear for some. This dendrogram (Figure 3) is supposed to be read such that the closer the variables are connected reading from the right, the more closely their missingness is correlated. As can be seen, Between EID and DateTime, the variables are directly correlated as all these don't have any missing values.

4.1.2 Feature types

Considering that we will only try to make imputations on the missing data, it is only needed to categorize the feature types of the variables which have missing data. This is quite simple since all

columns that have missing data are numerical and continuous. Age is numerical and discrete and some columns are categorical/nominal, but these all have complete data.

4.1.3 Kinds of missing data

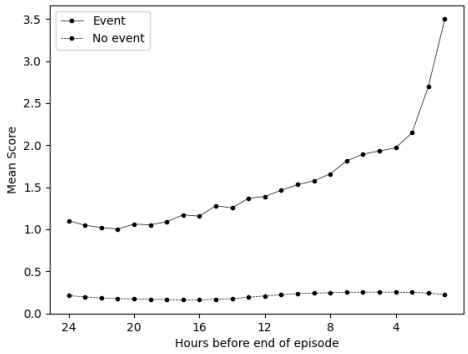
Using Little's test from the package pyampute [16], we could figure out if the data is MCAR [8]. It is a statistical test used to determine whether missing data in a dataset are missing completely at random or if there is a systematic pattern to the missingness. The null hypothesis, which is that the missing data is MCAR, can be rejected With a p-value less than 0.05. This would mean the missing data is either MAR or MNAR. For this data, we got a p-value of 0.0 so it can be assumed that the data is not MCAR, but MAR or MNAR. This means complete-case analysis (a procedure that eliminates all rows with one or more missing values on the analysis variables) should not be done as it can perform quite badly under MAR and some MNAR cases [15]. As Little and Rubin [9] mention, the bias in the estimated mean increases with the difference between means of the observed and missing cases, and with the proportion of the missing data when complete-case analysis is used on non-MCAR data.

4.2 Data imputation results

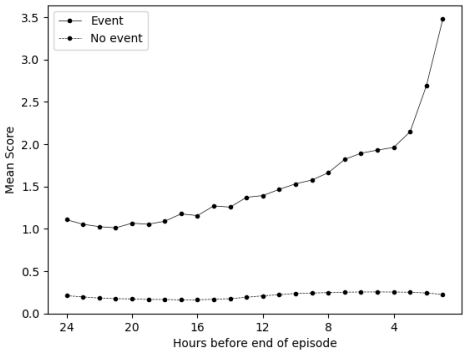
Figure 8 shows the average CAAM score over the 24 hours preceding an event in contrast to the average CAAM score over a random 24-hour window of an episode where no event happened. It shows that event episodes are more likely to have a higher score than those without any event. As can be seen for all imputation methods, the mean CAAM score starts rising rapidly about 8 hours before the event to an average score of 3.5. These graphs do not differ much, but it is proof that the new imputation methods (b to e) drastically change the CAAM score, for better or worse.

Figure 9 and Figure 10 show the sensitivity against 1-specificity of the CAAM model between all the different imputation methods. The AUC gives the probability that, between 2 patients where only 1 patient has experienced the event, the 1 patient gets a higher score than the patient who did not experience an event [5]. For the AUC per episode, mean imputation had the highest, being 81.1%, which means it gives the best general performance at classification. As for the hourly AUCs, the mean, median and multiple imputation (MI) using BayesianRidge shared the highest AUC: 81.1%.

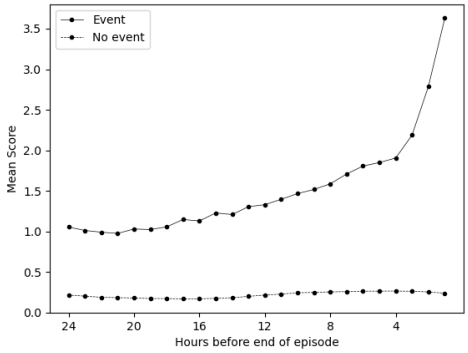
Figure 11 and Figure 12 shows the PPV (precision) vs the sensitivity for for all imputation methods. These graphs are often used to show the results of binary classifiers of outcomes when the outcome variable is very rare [5]. In this case, the rare outcome variable is that an event will happen. The area under the curve (AUC) per episode is the highest for the MI using RandomForest, 14%. That means MI using RandomForest has the lowest false positive and false negative rate. This is quite a large difference when compared to the mean, median and MI using LinearRegression, which have respectively an AUC of 13.5%, 13.4% and 13.3%. As for the hourly Precision Recall, all AUCs are the same with an except for MI by LinearRegression, which has 0.1% less AUC.



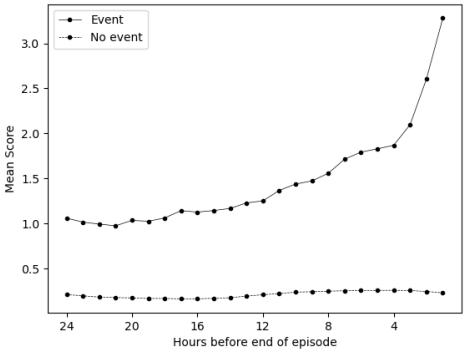
(a) Mean



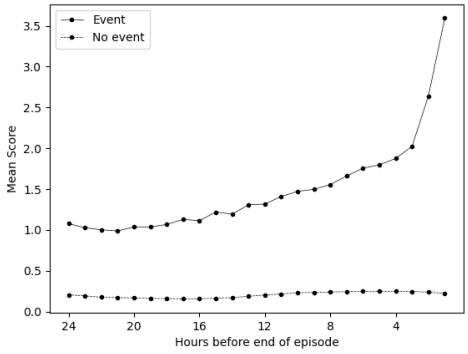
(b) Median



(c) BayesianRidge

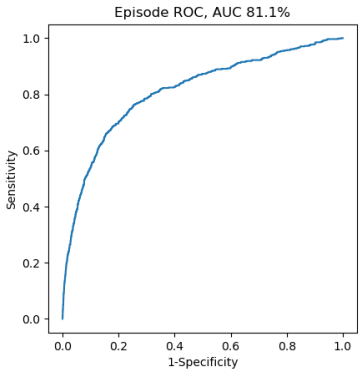


(d) LinearRegression

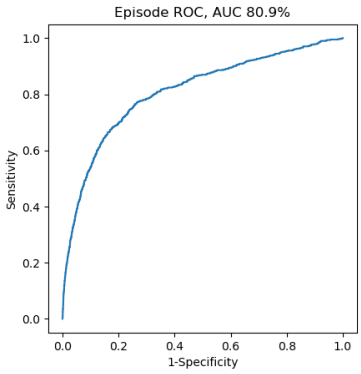


(e) RandomForest

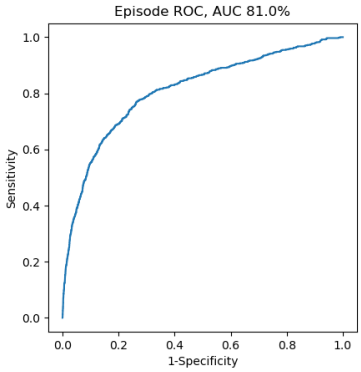
Figure 8: Mean CAAM score plotted over the 24 hours before an event, or a random 24 hours of an episode without an event.



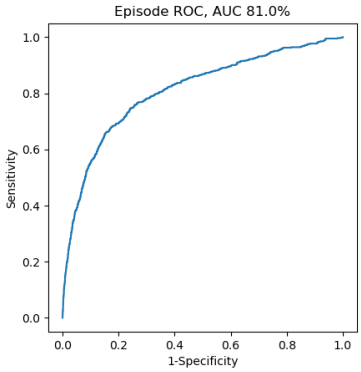
(a) Mean



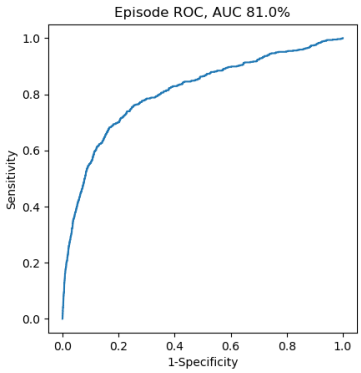
(b) Median



(c) BayesianRidge

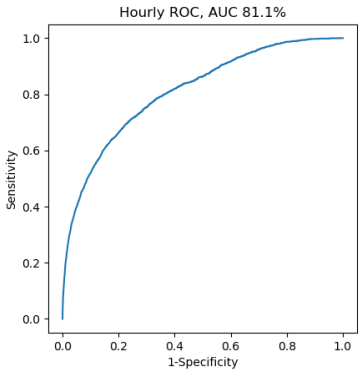


(d) LinearRegression

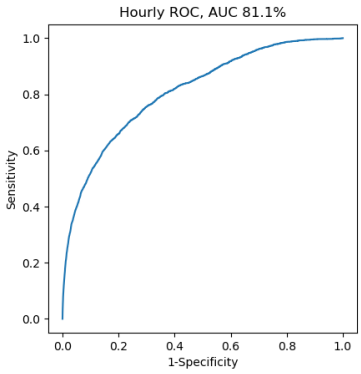


(e) RandomForest

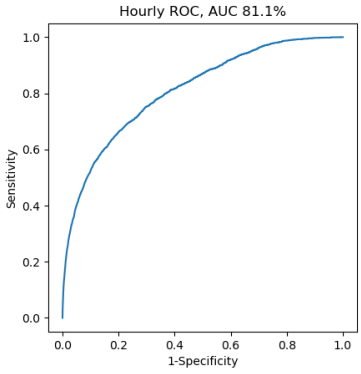
Figure 9: ROC score per each episode per imputation method.



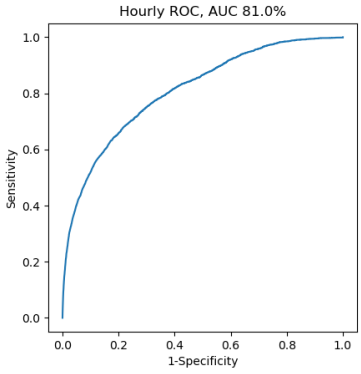
(a) Mean



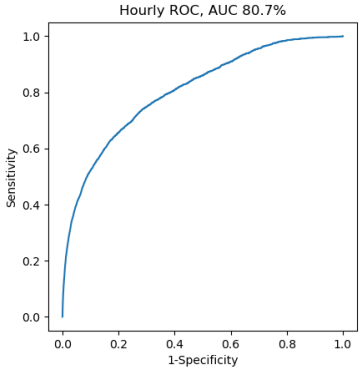
(b) Median



(c) BayesianRidge

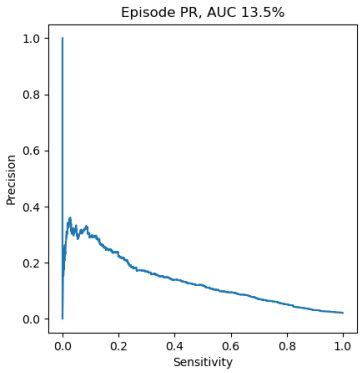


(d) LinearRegression

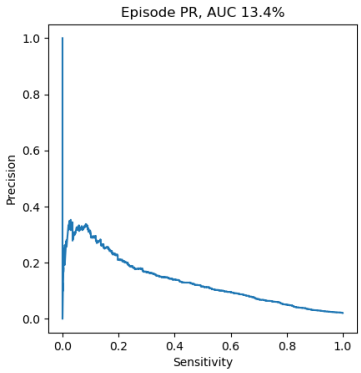


(e) RandomForest

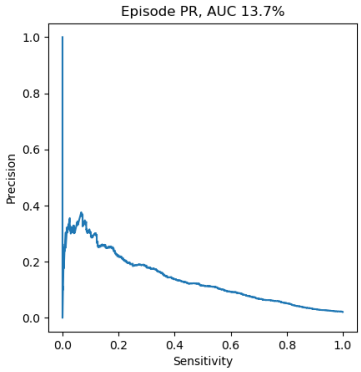
Figure 10: ROC score per hour for each imputation method.



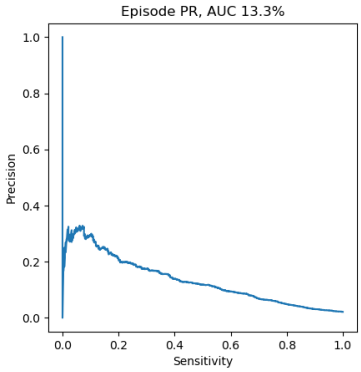
(a) Mean



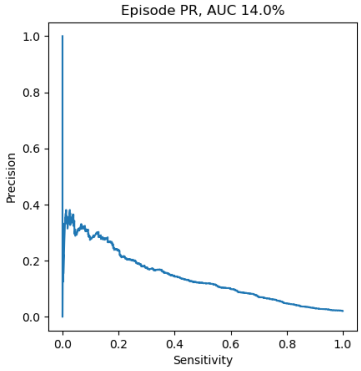
(b) Median



(c) BayesianRidge

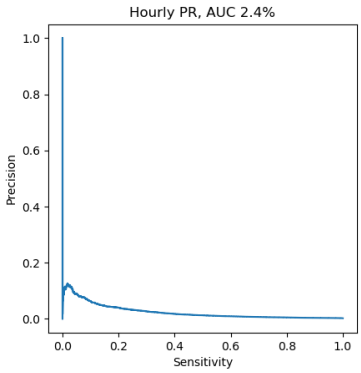


(d) LinearRegression

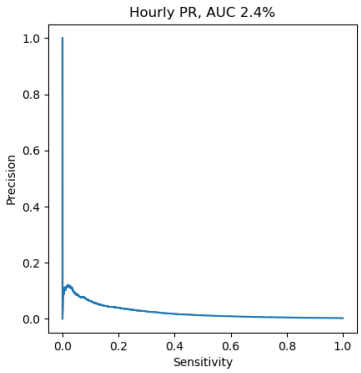


(e) RandomForest

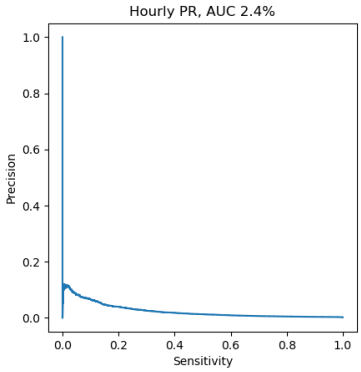
Figure 11: Precision Recall curve per episode for each imputation method.



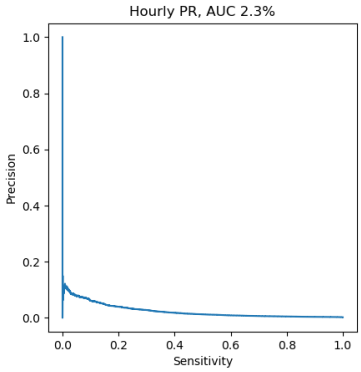
(a) Mean



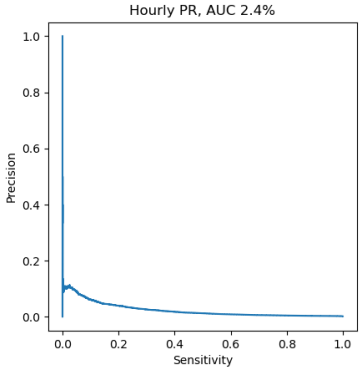
(b) Median



(c) BayesianRidge



(d) LinearRegression



(e) RandomForest

Figure 12: Precision Recall curve per hour for each imputation method.

As seen in Table 1, the PPV and specificity were both the highest for MI using RandomForest with 12.2% and 92.2% respectively. On the other hand, the NPV for all imputation methods is the same, which is 98.8%. This means that 1.2% of the patients who did not have an alert did end up experiencing an outcome. Lastly, MI using BayesianRidge had the highest sensitivity with 50.7%. Further metrics can be seen in Table 2.

MI by KNeighborsRegressor was tried but due to the running time of this algorithm, it was not possible to train it in time, resulting in not getting any performance metrics for this method of imputation. To still try to get a result, we tried to implement it lowering the amount of imputation iterations to 2 and by sampling only 10% of the data. Unfortunately, this still did not yield a result in time. This will be further discussed in section 5.

Imputation Method	Number of Alerts (% of Number of Episodes)	Specificity	Sensitivity	PPV	NPV
Mean Imputer	2650 (9.2%)	91.7%	50.5%	11.7%	98.8%
Median Imputer	2676 (9.3%)	91.6%	50.3%	11.5%	98.8%
Multiple Imputation using BayesianRidge	2736 (9.5%)	91.4%	50.7%	11.4%	98.8%
Multiple Imputation using LinearRegression	2580 (9.0%)	91.9%	49.7%	11.8%	98.8%
Multiple Imputation using RandomForest	2514 (8.7%)	92.2%	49.8%	12.2%	98.8%

Table 1: Most important metrics for all imputation methods.

Performance metric	Mean	Median	BayesianRidge	LinearRegression	RandomForest
Alarms with in 12 hours is	97	96	87	87	93
Hours alarms highest 25% quantile	-75	-75	-82	-76	-81.5
Number of episodes			28804		
Number of alerts	2650 (9.2%)	2676 (9.3%)	2736 (9.5%)	2580 (9.0%)	2514 (8.7%)
% of alerts with events within (PPV):					
12h	3.7%	3.6%	3.2%	3.4%	3.7%
24h	5.2%	5.1%	4.7%	5.0%	5.2%
Entire hospitalization episode	11.7%	11.5%	11.4%	11.8%	12.2%
W:D at:					
12h	27.0	27.8	31.3	29.4	27.0
24h	19.2	19.6	21.3	20.0	19.2
Entire hospitalization episode	8.5	8.7	8.8	8.5	8.2
Number not alerted	26154 (90.8%)	26128 (90.7%)	26068 (90.5%)	26224 (91.0%)	26290 (91.3%)
% of no alerts with no event (NPV):	98.8%	98.8%	98.8%	98.8%	98.8%
Number of events			614 (2.1%)		
Sensitivity in prior:					
12h	15.8%	15.6%	14.2%	14.2%	15.1%
24h	22.6%	22.3%	21.0%	20.8%	21.3%
Entire hospitalization episode	50.5%	50.3%	50.7%	49.7%	49.8%
Number with no event	28190 (97.9%)	28190 (97.9%)	28190 (97.9%)	28190 (97.9%)	28190 (97.9%)
Specificity	91.7%	91.6%	91.4%	91.9%	92.2%

Table 2: Comparison of all imputation methods by evaluation metrics.

5 Discussion

Whilst there are already some interesting results, due to the limited timeframe within this project, we were not satisfied with everything that was achieved. We were only able to get access to the data in week 10 of the project out of the 19 weeks which caused a bottleneck in being able to do research. That said, there are few things that could be researched more to develop a further understanding of the impact of imputation on the CAAM model.

Firstly, some thought did go into picking the parameters of the different imputation methods, but using a technique like GridSearch or Random Search could allow for actual hyperparameter tuning which could have improved the performance of the imputation and thus the CAAM model more. The best K for KNeighborsRegressor could for example have been calculated more carefully using cross-validation. Due to the limited time, the number of iterations for MI was set to 25 based on subsubsection 3.2.1. However, this caused some of the MI imputation methods not to converge: These were LinearRegression, RandomForest and KNeighborsRegressor. Whilst the absence of nonconvergence in an imputation-generating process is vital and the inference resulting from the imputation might be invalid, there is reason to believe that iterative imputation could achieve inferential validity before converging [11]. It has been found that inferential validity is achieved after 5 to 10 iterations, and whilst it never hurts to iterate more, those calculations hardly bring any value [12].

Secondly, although it is not in the scope of this thesis, it might be an idea to try to drop certain features of the data. This could potentially lose valuable information and thus also lower the performance of the CAAM. However, considering so much data is already missing, we feel like it might not lower the performance of the model and potentially could greatly decrease the computational cost. As can be seen in Figure 6, these features are for example TROP, ANIONGAP and PML which all have a bit more than 94% missing data.

Thirdly, as mentioned in subsubsection 3.2.3, instead of doing imputation by K-Nearest-Neighbors, it was decided that multiple imputation using K-Neighbors-Regressor would be used. This is because the KNNImputer package from sklearn does not run parallel and has an asymptotic running time of $\mathcal{O}(n^2)$. This means that the running time is very large for a considerably large data set such as this one. Another way of making this method of imputation quicker is by reducing the number of features as KNN suffers from the curse of dimensionality. Despite all this being done, it was not possible to get results from this method of imputation in time.

Lastly, there is another imputation method that could be tried on this data set specifically. An algorithm that makes a cumulative distribution function (CDF) of each feature and then imputes on the data by random sampling on the CDF could also be interesting to look at given that it would not change the distribution of the already existing data.

6 Conclusion

Figuring out the best imputation method depends a lot on different factors that could be differently important to different population groups. Where running time might matter for someone working on the CAAM model, it is not likely that it affects clinicians. On the other hand, clinicians have to deal with the alerts and so would prefer that there are as few false positives as possible. As for patients, they want to be certain that they do get treated if they happen to experience an outcome, so they would want as few false negatives as possible.

For those building the CAAM model, computational cost may be more important. This is dependent on how often the model is retrained. If this is the case then we suggest staying away from MI using RandomForest and especially KNeighborsRegressor. If computational cost is not as important, then it seems that interests would align more with the other population groups. The AUC's of both the precision-recall (Figures 12 and 11) and ROC (Figures 10 and 9) graphs seem also more important for the developer as these are quick to read and relatively easy to compare. Considering this, we would suggest that either the MI using BayesianRidge or RandomForest is the best.

For clinicians, multiple imputation using RandomForest is the best imputation method as it has the highest specificity, lowest W:D as well as the lowest number of alerts.

As for the patient population group, the NPV is important since the higher it is, the less likely it is that a patient doesn't produce an alert yet does experience an outcome. However, considering this metric is the same for all imputation methods, it doesn't warrant as much attention.

Taking all that into account, we would suggest that MI using RandomForest improves the CAAM model the most, granted that computational cost is not important. In the case that it is, MI using BayesianRidge is a good middle ground as it slightly improves the baseline (median imputation) and has a much lower computational cost than MI with RandomForest. Both these methods don't need Z-score normalization, but it is still used after the imputation for the CAAM model itself.

References

[1] F. Cismondi. *Preprocessing and misclassifying issues in clinical data sets for prediction and intervention*. PhD thesis, 11 2012.

[2] A. Engel. Categorical Variables for Machine Learning Algorithms, 3 2022.

[3] G. J. Escobar, V. X. Liu, A. Schuler, B. Lawson, J. D. Greene, and P. Kipnis. Automated Identification of Adults at Risk for In-Hospital Clinical Deterioration. *New England Journal of Medicine*, 383(20):1951–1960, 11 2020. ISSN 0028-4793. doi: 10.1056/NEJMsa2001090.

[4] H. Henderi. Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *IJIIS: International Journal of Informatics and Information Systems*, 4(1):13–20, 3 2021. ISSN 25797069. doi: 10.47738/ijiis.v4i1.73.

[5] P. Kipnis, B. J. Turk, D. A. Wulf, J. C. LaGuardia, V. Liu, M. M. Churpek, S. Romero-Brufau, and G. J. Escobar. Development and validation of an electronic medical record-based alert score for detection of inpatient deterioration outside the ICU. *Journal of Biomedical Informatics*, 64: 10–19, 12 2016. ISSN 15320464. doi: 10.1016/j.jbi.2016.09.013.

[6] R. Kumar Bohara. How to find the optimal value of K in KNN, 10 2023. URL https://medium.com/@rkbohara097/how-to-find-the-optimal-value-of-k-in-knn-2d5177430f2a.

[7] P. Li, E. A. Stuart, and D. B. Allison. Multiple Imputation. *JAMA*, 314(18):1966, 11 2015. ISSN 0098-7484. doi: 10.1001/jama.2015.15281.

[8] R. J. A. Little. A Test of Missing Completely at Random for Multivariate Data with Missing Values. *Journal of the American Statistical Association*, 83(404):1198–1202, 12 1988. ISSN 0162-1459. doi: 10.1080/01621459.1988.10478722.

[9] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 8 2002. ISBN 9780471183860. doi: 10.1002/9781119013563.

[10] L.-A. McNutt. Estimating the Relative Risk in Cohort Studies and Clinical Trials of Common Outcomes. *American Journal of Epidemiology*, 157(10):940–943, 5 2003. ISSN 00029262. doi: 10.1093/aje/kwg074.

[11] H. I. Oberman and G. Vink. Toward a standardized evaluation of imputation methodology. *Biometrical Journal*, 3 2023. ISSN 0323-3847. doi: 10.1002/bimj.202200107.

[12] H. I. Oberman, S. van Buuren, and G. Vink. Missing the Point: Non-Convergence in Iterative Imputation Algorithms. 2020. URL https://api.semanticscholar.org/CorpusID:260441296.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

[14] D. B. Rubin. Multiple imputation. In *Flexible Imputation of Missing Data*, chapter 2, pages 29–62. Chapman and Hall/CRC, New York, 2nd edition, 7 2018. ISBN 9780429492259.

[15] J. L. Schafer and J. W. Graham. Missing data: our view of the state of the art. *Psychological methods*, 7 2:147–77, 2002. URL https://api.semanticscholar.org/CorpusID:7745507.

[16] R. M. Schouten, D. Zamanzadeh, and P. Singh. pyampute: a Python library for data amputation, 8 2022. URL https://doi.org/10.25080/majora-212e5952-03e.

[17] J. Shreffler and M. R. Huecker. *Diagnostic Testing Accuracy: Sensitivity, Specificity, Predictive Values and Likelihood Ratios*. StatPearls Publishing, Treasure Island (FL), 2023. URL http://europepmc.org/books/NBK557491.

[18] N. Tsikriktsis. A review of techniques for treating missing data in OM survey research. *Journal of Operations Management*, 24(1):53–62, 2005. ISSN 0272-6963. doi: https://doi.org/10.1016/j.jom.2005.03.001. URL https://www.sciencedirect.com/science/article/pii/S027269630500077X.

[19] S. van Buuren. *Flexible Imputation of Missing Data, Second Edition*. Chapman and Hall/CRC, Second edition. Boca Raton, Florida : CRC Press, [2019] , 7 2018. ISBN 9780429492259. doi: 10.1201/9780429492259.

7 Appendix

Variable	Number of missing data	Proportion of missing data (%)
TROP	9238670	94.678167
ANIONGAP	9220748	94.494502
PML	9220748	94.494502
LACT	9060097	92.848146
PAO	8957427	91.795981
PH	8956873	91.790303
PAC	8956516	91.786645
BICARB	8820038	90.388014
ALB	8422360	86.312598
BIL	7804449	79.980227
GLUCOSE	7676651	78.670550
HEMAT	6337988	64.951891
WBC	4681176	47.972832
BUN	4364240	44.724862
SODIUM	4092576	41.940842
CREAT	4079401	41.805824
AVPU	1431527	14.670332
RSPRT_I	1124424	11.523131
RSPRT	1124424	11.523131
RSPRT_W	1124424	11.523131
SAT	985916	10.103697
SAT_I	985916	10.103697
SAT_W	985916	10.103697
SHOCKINDEX	848455	8.694992
BPDIA	835121	8.558345
BPSYS	826850	8.473584
BPSYS_I	826843	8.473512
HRTRT	795980	8.157227
HRTRT_I	795980	8.157227
TEMP_I	730367	7.484823
TEMP	730367	7.484823

Table 3: All variables, their corresponding amount of missing data and their proportion to present data.

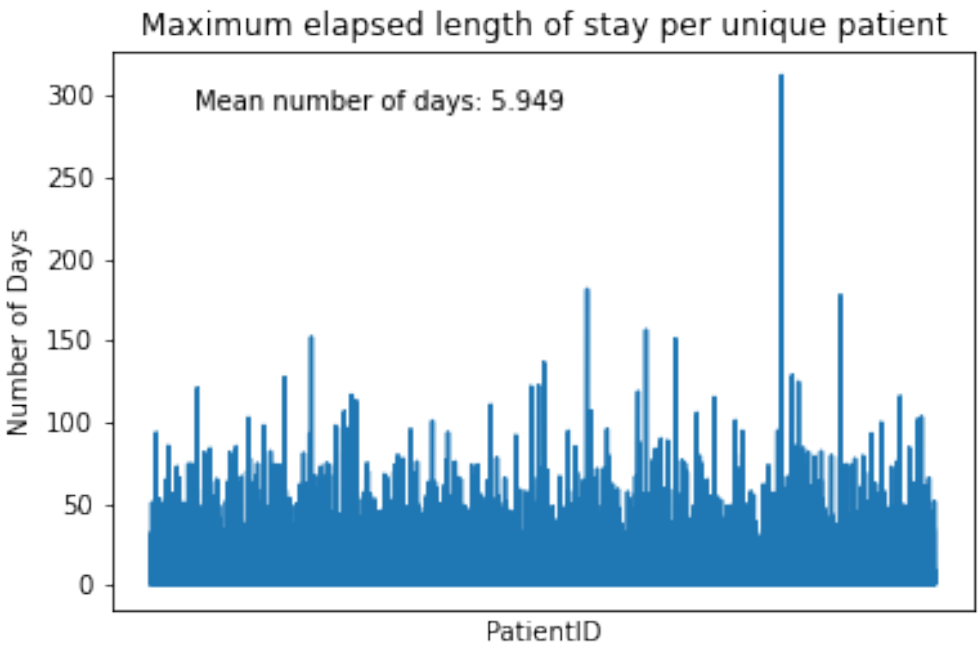


Figure 13: Maximum elapsed length of stay per unique patient