

IN-HOSPITAL MORTALITY OF ICU HEART FAILURE PATIENTS, A MIMIC-III DATABASE ANALYSIS

DATA ANALYTICS IN HEALTHCARE AND
CONNECTED CARE

Aline Jacquart, Amine Naqi, Cristian Pinelo, Peter Kögler

Table Of Contents



1	Introduction	3
2	Problem Description	3
3	Querying	3
4	Pre-processing	5
5	Modeling	6
5.1	Upsampling	7
5.2	Feature Selection	8
5.3	Splitting Data	8
5.4	Training and Evaluating the Models	8
5.4.1	SVM Model	8
5.4.2	Random Forest	9
5.4.3	Neural-Networks	9
6	Discussion of the Results	10
7	Distribution of Tasks	11
8	References	12

1 Introduction

The aim of this project is to understand how real-life data from a healthcare system is acquired, stored, processed, visualized and used to provide meaningful insights for clinicians. For this purpose, a study on mortality prediction model of patients in ICU with heart failure disease is conducted. The data used in this study comes from the freely available MIMIC-III database. This database includes information of over forty thousand patients who stayed in the intensive care units of the Beth Israel Deaconess Medical Center between 2001 and 2012 [1].

To achieve the goal of developing an efficient mortality prediction model, the report follows a structured methodology. First, the problem of heart failure is described and the risk factors for mortality are identified. They are then queried using BigQuery, a data warehouse supporting large queries. After selecting and querying the data needed, pre-processing is applied. Using Python, the raw data is filtered and merged in a general data frame. The obtained data frame is afterwards used to feed several classifiers. The models are trained and evaluated thanks to several metrics reporting the performance of the algorithms. The report is concluded with a description of the group organization.

2 Problem Description

Heart failure is “a common complex clinical syndrome of symptoms and signs caused by impairment of the heart’s action as a pump supporting the circulation. It is caused by structural or functional abnormalities of the heart” [2].

Several parameters present in MIMIC-III could be related to a higher mortality in case of heart failure. To determine which ones will be given to the predictive models, the paper [3] is used. In this research, the XGBoost algorithm is used to indicate the contribution of each of the predictors in HF mortality in ICU. The 20 most relevant predictors given by this algorithm are selected for this work, except the urine output because the requirement of measuring in the first 24h is not met for too much data. The chosen parameters with their reference ranges are displayed in Table 1.

For ethical reasons, all the data is anonymized. The patients are identified by a ID number and their date of birth, the date of entrance at the ICU and the date of death are randomly modified. In this way, it is impossible to trace the identity of registered patients.

3 Querying

After agreeing on which parameters to consider to feed the predictive models, the next step is their extraction from the MIMICIII database.

Most of the parameters are a result of laboratory-based measurements. These can be found in the LABEVENTS table. Using the corresponding dictionary table D_LABITEMS to find the specific ITEMIDs for each parameter, the following query was used to obtain the parameters where **PLACEHOLDER** represents the ITEMIDs:

```
SELECT Subj.SUBJECT_ID, VALUE, lab.VALUEUOM, lab.CHARTTIME FROM `physionet-
data.mimiciii_clinical.labevents` lab INNER JOIN (SELECT SUBJECT_ID, ICD9_CODE FROM
`physionet-data.mimiciii_clinical.diagnoses_icd` Subj
```

```
WHERE ICD9_CODE = '4280') Subj ON Subj.SUBJECT_ID = lab.SUBJECT_ID WHERE lab.ITEMID =
PLACEHOLDER
ORDER BY Subj.SUBJECT_ID
```

In this query, LABEVENTS is joined with DIAGNOSES_ICD on SUBJECT_ID to collect only data of patients that are diagnosed with heart failure, indicated by the ICD9_CODE 4280. SUBJECT_ID, VALUE and VALUEUOM are kept to identify the patient later and to obtain an interpretable value.

The parameters *heart rate* and *respiratory rate* can be found in the CHARTEVENTS table. As these two parameters are a result of constantly repeated measurements the amount of data they produce is enormous. Consequently, the previously described query results in too much data to handle. The solution to that problem is using the SQL functions AVG and VARIANCE in connection with GROUP BY, resulting in a query of the following structure where PLACEHOLDER represents ITEMIDs:

```
SELECT * FROM
(SELECT chart.SUBJECT_ID, AVG(DISTINCT CAST(VALUE AS NUMERIC)) AS `Heart Rate`,
VARIANCE(DISTINCT CAST(VALUE AS NUMERIC)) AS `Variance`, VALUEUOM from `physionet-
data.mimiciiii_clinical.charthevents` chart
INNER JOIN `physionet-data.mimiciiii_clinical.diagnoses_icd` d ON chart.SUBJECT_ID =
d.SUBJECT_ID
WHERE chart.ITEMID = PLACEHOLDER and ICD9_CODE = "4280" and REGEXP_CONTAINS(VALUE,
r'^-?[0-9]+\.[0-9]*$')
GROUP BY SUBJECT_ID, VALUEUOM, ITEMID)
```

Like before, only heart failure-diagnosed patient's data is considered. AVG and VARIANCE are functions that must be applied to numerical values. This is guaranteed using SQL's CAST function and a regular expression removing all string entries that do not contain numbers.

Lastly, a table holding all heart failure-diagnosed patients with their SUBJECT_ID and the information whether they expired in hospital or not is extracted. The following query is used:

```
SELECT diag.SUBJECT_ID, pat.DOB, pat.DOD, DISCHTIME, adm.HOSPITAL_EXPIRE_FLAG FROM
`physionet-data.mimiciiii_clinical.patients` pat
INNER JOIN (SELECT SUBJECT_ID, ICD9_CODE FROM `physionet-
data.mimiciiii_clinical.diagnoses_icd` Subj WHERE ICD9_CODE = '4280') diag
ON pat.SUBJECT_ID = diag.SUBJECT_ID
INNER JOIN (SELECT SUBJECT_ID, HOSPITAL_EXPIRE_FLAG, DISCHTIME FROM `physionet-
data.mimiciiii_clinical.admissions`) adm
ON diag.SUBJECT_ID = adm.SUBJECT_ID
```

Besides the HOSPITAL_EXPIRE_FLAG, DOB, DOD and DISCHTIME are extracted allowing the calculation of the patient's age at the time of discharge.

All queries' results are exported to csv files, allowing straight forward subsequent processing in the form of data frames with python. An overview of the parameters is presented in Table 1.

Table 1. Parameters analysed, their reference ranges and the table where they are located via the ITEMID.[4][5]

Parameter	Reference range	ITEMID	Table
Anion gap	4 – 12 mmol/L	50868	LABEVENTS
Lactate	0.5 – 2.2 mmol/L	50813	LABEVENTS
Calcium	2.2 - 2.6 mmol/L	50893	LABEVENTS
Lymphocytes	1.0-4.8 10 ³ /μL	51116	LABEVENTS
White blood cells	4.8-10.8 10 ³ /μL	861	LABEVENTS

Parameter	Reference range	ITEMID	Table
Blood urea nitrogen	6 – 24 mg/dL	51006	LABEVENTS
Mean cell hemoglobin	27-33 pg/cell	51248	LABEVENTS
Creatin kinase	M=40 – 320, F=25 - 200 μmol/L	50910	LABEVENTS
Diastolic blood pressure	<80 mmHg	220180, 220051	LABEVENTS
Red blood cells	M=4.5-6.0, F=4.1-5.1 106/μL	51279	LABEVENTS
PaCO ₂	4.5 - 6.1 kPa	50818	LABEVENTS
Sodium	133 - 146 mmol/L	50983	LABEVENTS
Platelet count	175-450 103/μL	51265	LABEVENTS
Potassium	3.5 - 5.3 mmol/L	50971, 50833	LABEVENTS
N-terminal pro-brain natriuretic peptide (NTproBNP)	<125 pg/mL for <75y <450 pg/mL for >75y	50963	LABEVENTS
Respiratory rate	12 – 18 bpm at rest	618	CHARTEVENTS
Heart rate	60-100bpm at rest	211, 20045	CHARTEVENTS
Chronic Kidney Disease	/	5851, 5852, 5853, 5854, 5855, 5859	DIAGNOSES_ICD
Age	0-89 years, >89 years	-	PATIENTS

4 Pre-processing

After obtaining the raw data from BigQuery it is necessary to filter it and merge it all together into a single dataframe to use in the machine learning models. There are two types of data: binary (for conditions such as chronic kidney disease) and numerical (for all the physiological parameters measured).

Since the amount of raw data is exceptionally large and quite different in the number of entries between patients it is decided to use the mean of all the entries for each patient. To also analyse the separation between these values the standard deviation of each parameter is included into the dataframe and fed to the models. This method of using the mean considers more information for the patients that have larger measurements. One downside is that it may include data not representative of the patient's status if they have a prolonged stay. Another considered option is to take only the data from the first 24h after the patient was admitted to ICU, however, this approach may be missing later information that represents better the patient's status. For this reason, using the mean is selected.

After obtaining the mean and standard deviation of all parameters individually, the data frames are merged. Some consideration considered to further filter the data is to remove all patients with an average heart rate of 30 or less. Since not all the patients have entries for all the parameters, one

set of data is defined by removing the parameters that are missing by more than 20% of patients. Another set is defined by the parameter with the least occurrences, removing all patients that do not have that parameter entry. This results in a dataset that includes all the desired parameters and one with fewer parameters but more accurate data. As can be observed in Figure 1, the parameters that have the least number of occurrences are the NTproBNP and the diastolic blood pressure, being present in less than 50% of the selected population. For this reason, it is decided to remove those two parameters to create the dataset *small data* and the *big data* was created by maintaining these two parameters and removing all patients without entries for them. The latter dataset is used to test the impact of the two mentioned parameters on the accuracy of the machine-learning models.

After defining the two datasets there are still some empty entries remaining. After verifying that no more than 20% of entries are missing per parameter, the empty cells are filled with the average of their respective column. Since this is done for all parameters and those parameters miss less than 20% of their entries, it is assumed to not affect the accuracy of the data in a significant way.

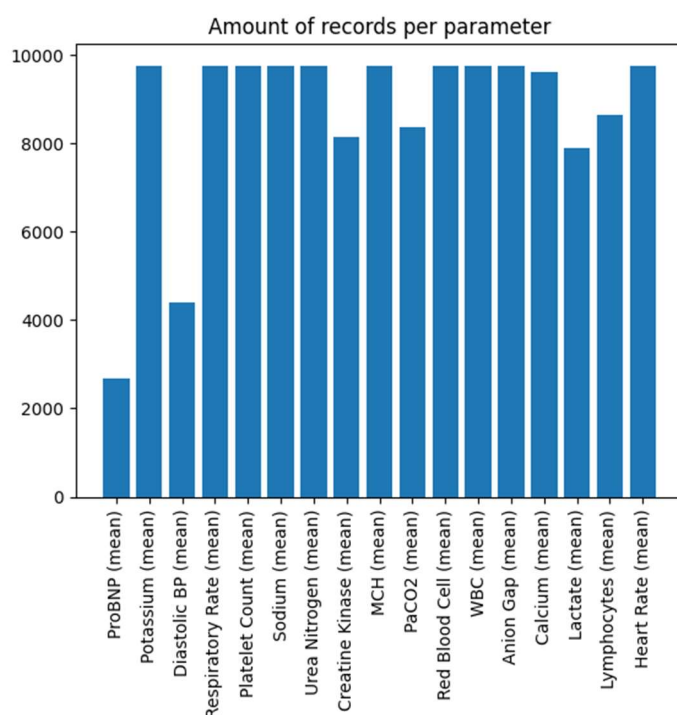


Figure 1. Number of records per parameter of interest. The total number of patients included was 9843.

5 Modeling

Three machine learning models are used for this binary classification problem. Firstly, in SVM, an efficient and relatively simple model is used. As the data has an input dimensionality of 32 and 36 respectively, a random forest and neural network are selected, based on their ability to deal with high dimensional input and general classification potency.

All three models are applied to both datasets. As the model trained on *Small Data* consistently produces superior results, documentation on *Big Data* is emitted. The modeling part follows the steps shown in Figure 2 and is described in more detail in the following.

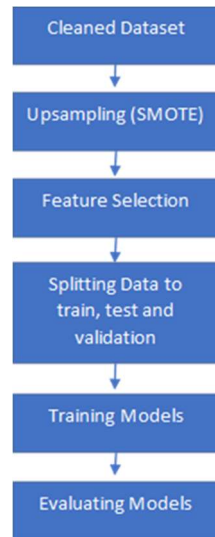


Figure 1. Modeling diagram

5.1 Upsampling

Looking at the preprocessed data, an imbalance between the two classes becomes apparent (Figure 3).

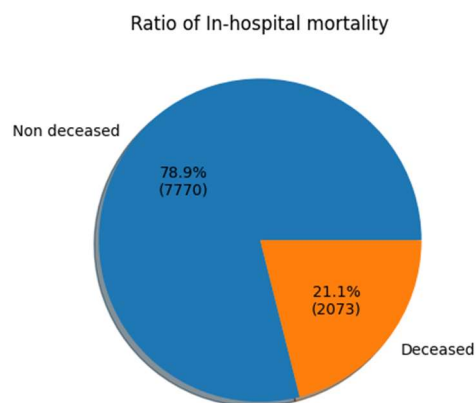


Figure 2. In-hospital mortality ratio before dividing the datasets. The plot shows the class imbalance.

- Small data: 7702 Patients with label 0, and 2044 Patients with label 1.
- Big data: 1432 Patients with label 0, and 347 Patients with label 1.

Since training on this imbalanced data leads to misclassification and bad models, it is decided to upsample it using *SMOTE*.

SMOTE is an oversampling technique that generates synthetic samples from the minority class. It is used to obtain a synthetic class-balanced or class-balanced training set, which is then used to train the classifier. Using this technique, two new datasets are generated (Equal proportions of the binary classification) that are good to be used for training.

5.2 Feature Selection

Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

It shows that feature selection reduced the performance of the models. Moreover, the gain in computational efficiency showed to be negligible. It therefore is decided to refrain from feature selection.

5.3 Splitting Data

Using the *train_test_split* function, we have split our original datasets into training, and testing/validating sets, where the training set consists of 80% of the original data set. As for the 20% remaining, it is decided to split it into two subsets that are used for benchmarking, which are validation and test sets, having both respectively 10% each of the entire dataset.

5.4 Training and Evaluating the Models

For all the models, the scikit-learn pipeline is used in addition to the specific algorithms. This is a valuable tool in machine learning workflows that helps streamline and simplify the process of building and deploying models. In the pipeline, the *StandardScaler* preprocessing technique is used for standardisation. It transforms the data such that each feature has a mean of 0 and a standard deviation of 1.

5.4.1 SVM Model

SVM (Support Vector Machine) is a machine learning algorithm that is used for classification and regression tasks. It works by finding the best possible separation boundary (hyperplane) between different classes of data points in a high-dimensional feature space. SVM aims to maximize the margin between the classes, making it a powerful tool for solving complex classification problems. As the SVM's kernel, RBF is used. The results obtained using SVM are presented in Table 2.

Table 2. Statistical results for the SVM algorithm with the test set.

Accuracy	Precision	Recall
0.827	0.826	0.837

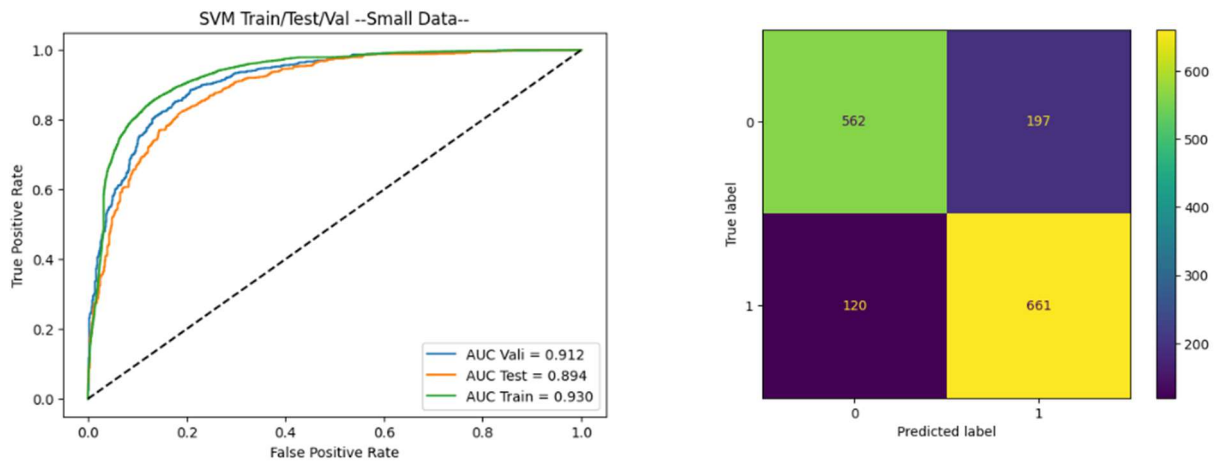


Figure 3. Area under the curve (left) and confusion matrix from validation set (right) for SVM classifier. Results obtained using **small data** values.

5.4.2 Random Forest

Random Forest is a machine learning algorithm that is used for both classification and regression tasks. It operates by constructing an ensemble of decision trees and combining their predictions to make accurate predictions. Each decision tree in the Random Forest is built independently using a subset of the training data and a subset of the features. The algorithm then aggregates the predictions from each tree to produce the final output. Random Forest is known for its ability to handle large datasets, feature selection, and handling non-linear relationships in the data. It is a popular algorithm for various applications due to its robustness and versatility. N=10 is used for the maximum depth of the trees and N=30 for the number of estimators to prevent overfitting. The results obtained using the random forest model are presented in Table 3.

Table 3. Statistical results for the random forest algorithm with the test set.

Accuracy	Precision	Recall
0.880	0.871	0.899

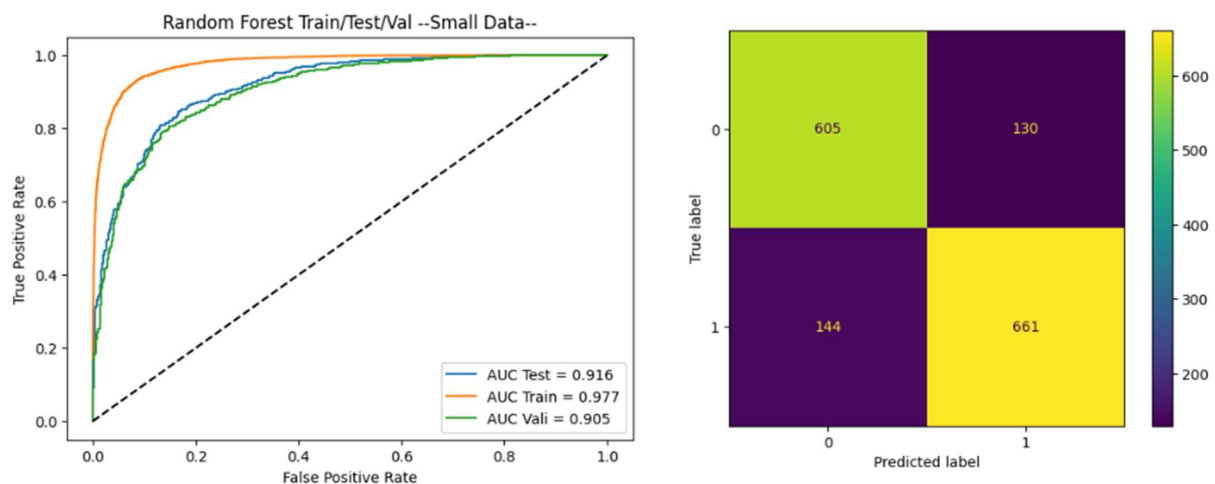


Figure 4. Area under the curve (left) and confusion matrix from validation set (right) for random forest classifier. Results obtained using **small data** values.

5.4.3 Neural-Networks

Neural networks, also known as artificial neural networks (ANNs), are a class of machine learning models inspired by the structure and functioning of the human brain. A neural network consists of

interconnected nodes, called neurons or units, organized into layers. The input layer receives the input data, which then propagates through one or more hidden layers before reaching the output layer. Each neuron receives input signals, applies a transformation (activation function) to those inputs, and produces an output signal. The connections between neurons are represented by weights, which are adjusted during the training process to optimize the network's performance. Two hidden layers with sizes 12 and 8 respectively are used. The training consists of 50 epochs. The results obtained using the neural network are presented in Table 4.

Table 4. Statistical results for the neural network algorithm with the test set.

Dataset	Accuracy	Precision	Recall
Small	0.789	0.810	0.767

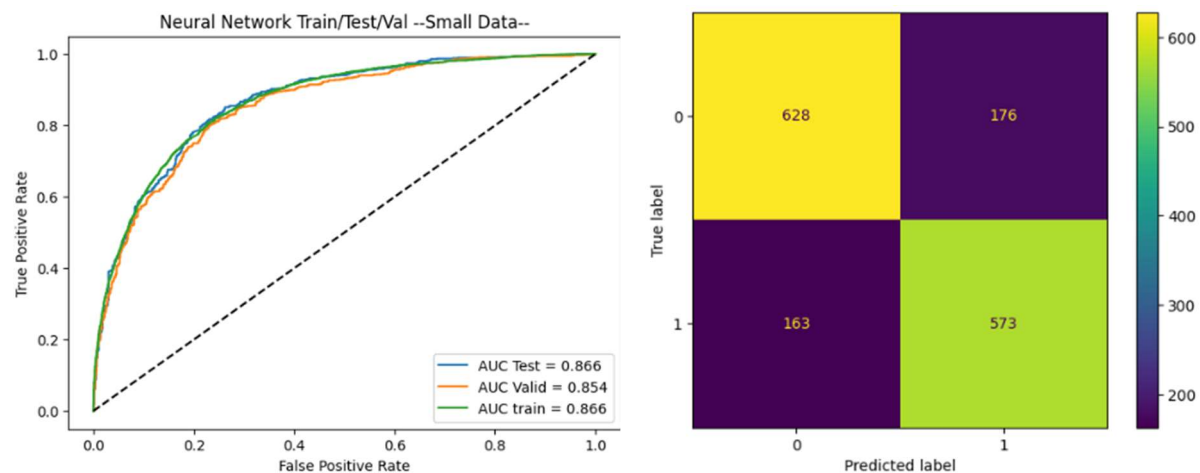


Figure 5. Area under the curve (left) and confusion matrix from validation set (left) for a neural network classifier. Results obtained using **small data** values.

6 Discussion of the Results

For the different machine learning models, the SVM algorithm ended up having an area under the curve of 0.912 for the validation data. This algorithm's predictions had a sensitivity of 84.6% and a specificity of 74% (Figure 4). Comparing the train, validation and test AUC curve it can be observed that they perform similarly well. This indicates the absence of overfitting and good generalization. The random forest algorithm had an area under the curve for the validation of 0.905, sensitivity of 82% and specificity of 82.3% (Figure 5). However, it is apparent that the performance on the train set is superior, with validation and test being close to each other. This indicates that the model is already overfit. It is beyond the scope of this work to perfectly optimize the random forest model. The neural network algorithm had an area under the curve for the validation of 0.854, sensitivity of 77.9% and specificity of 78.1% (Figure 6). Again, train, validation and test curve are very similar, indicating no overfitting.

Random forest proved to be the best model for this dataset following the results that we have obtained. The confusion matrix related to this model shows good classification of the patients throughout all possibilities. When it comes to true positives and true negatives, the model was able to predict around 600 cases in both sections, while only misclassifying 130 patients who should have been classified as negative, and 146 patients who should have been classified as positive.

7 Distribution of Tasks




The decision about what parameters to use was made in meetings with all four group members present. As we took 20 parameters into consideration, the querying of five parameters was assigned to each group member. This task consisted of making available a table in csv format of every parameter.

It was decided that distributing the subsequent tasks among all was the most efficient approach. Aline and Cristian took care of preprocessing the data while Peter and Amine put together the machine learning models. Fundamental decisions like how to treat incomplete datasets and what models to use were discussed with all group members.

The writing of the report was performed simultaneously by all group members. Each member wrote the basis of their task from the practical work and was reviewed by the rest of the group to get feedback and obtain a good report.

8 References

- 
- [1] Johnson, A., Pollard, T., Shen, L. *et al.* MIMIC-III, a freely accessible critical care database. *Sci Data* **3**, 160035 (2016). <https://doi.org/10.1038/sdata.2016.35>
- [2] National Guideline Centre (UK). Chronic Heart Failure in Adults: Diagnosis and Management. London: National Institute for Health and Care Excellence (NICE); 2018 Sep. (NICE Guideline, No. 106.) 2, Introduction. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK536089/>
- [3] Li F, Xin H, Zhang J, Fu M, Zhou J, Lian Z. Prediction model of in-hospital mortality in intensive care unit patients with heart failure: machine learning-based, retrospective analysis of the MIMIC-III database. *BMJ Open*. 2021 Jul 23;11(7):e044779. doi: 10.1136/bmjopen-2020-044779. PMID: 34301649; PMCID: PMC8311359.
- [4] Table of normal values. Lichtman M.A., & Kaushansky K, & Prchal J.T., & Levi M.M., & Burns L.J., & Armitage J.O.(Eds.), (2017). Williams Manual of Hematology, 9e. McGraw Hill. <https://hemonc.mhmedical.com/content.aspx?bookid=1889§ionid=137387407>
- [5] Clinical Biochemistry Reference Ranges Handbook, Eastbourne District General Hospital Issue 14 V2.2 Ratified by: Clinical Biochemistry Team Date ratified: April 2020 Name of author and title: Suzanne Fuggle, Cons Clinical Scientist Date Written: November 2011 Name of responsible committee/individual: Jacqueline Munro, Lead BMS Date issued: April 2020 Issue number: 14 Review date: April 2021. <https://www.esht.nhs.uk/wp-content/uploads/2017/08/Clinical-Biochemistry-reference-ranges-handbook.pdf>

Theory from Data Analytics in Health Care and Connected Care HOC