Mortality Prediction Models using MIMIC-III

Xiaoyan Liu
Siqi Mo
Qiuhao Jin
Yuhua Cai

November 21, 2020

1 Introduction

An intensive care unit (ICU) provides intensive treatment medicine for patients with severe and life-threatening illness and injuries, or those directly transferred from emergency department. ICUs have higher staff-to-patient ratio then normal wards to provide intensive care and comprehensive monitoring to severe patients, and hence generate a massive amount of electronic healthcare records (EHR) which are useful to predict patients' disease status and the amount of healthcare needed. The Medical Information Mart for Intensive Care III (MIMIC-III) is a freely-accessible ICU database comprising de-identified EHR of over 60,000 ICU stays for around 40,000 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. The database consists of rich information about patients' demographic characteristics, such as gender, age, ethnicity, admission type, and various in-hospital measurements, laboratory tests, procedures and medication of ICU patients over the time. The database provides data from two electronic healthcare record systems, namely the CareVue (from 2001 to 2008) and MetaVision (from 2008 to 2012), which collect and store data differently.

Over the past few decades, several ICU scoring systems have been developed for mortality prediction using rule- based method or data mining approach. Some standard scoring systems include Acute Physiology And Chronic Health Evaluation (APACHE), Simplified Acute Physiology Score (SAPS) and Sepsis-related organ Failure Assessment Score (SOFA). APACHE is a severity scoring systems designed to provide a morbidity score for a patient. A predicted mortality can be derived from this score. SAPS9 was designed to predict morbidity for a particular patient by comparing the outcome with other patients or a group of patients by comparing the outcome with another group of patients. SOFA provides a daily score to track a person's status during an ICU stay to determine the extent of a person's organ function or rate of failure.

With the advancements in parallel and distributed computing and machine learning, better mortality prediction models trained on larger input features have been developed using machine learning approach. Desautels et al. introduced InSight, a machine learning classification system that uses multivariable combinations of easily obtained patient data (vitals, peripheral capillary oxygen saturation, Glasgow Coma Score, and age), to predict sepsis using MIMIC-III. Ghassemi et al.12 applied Latent Dirichlet Allocation to free-text hospital notes to make mortality prediction using MIMIC-II.Hrayr et al. applied Recurrent Neural Network and provided four clinical prediction benchmarks using MIMIC-III. Pirracchio et al. applied ensemble methods to improve mortality prediction in the ICU. Awad et al. provided early hospital mortality prediction of ICU patients, i.e. first 6 hours since ICU admissions, using an ensemble approach.

2 Objective and Database

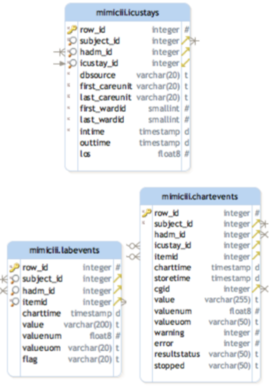
The goal is to build a model framework to predict in-hospital mortality. The model would be useful to promptly identify high-risk patients who might be dead within hours or days since ICU admission, so that resources can be efficiently allocated during the early stage of ICU st.

3 Dataset Description

We referred to the instruction form MIMIC official website to set up our database in PostgreSQL. We firstly downloaded the whole dataset from the website. And then we referred the mimiciii schema and build a set of empty tables. After that we import the CSV data files into the empty tables. To interface with Python, we use psycopg2 to connect to database.

4 Feature Extraction

In this project, to predict the mortality in the early stage of ICU stay, we are using three severity scores: SAPS II, APACHE II, and SOFA scores. For APACHE II [1], it is calculated from a patient's age and 12 routine physiological measurements. And measurements are measured during the first 24 hours after admission, and utilized in addition to information about previous health status (recent surgery, history of severe organ insufficiency, immunocompromised state) and baseline demographics such as age. Another key thing is that APACHE II only consider the most deranged physiology value physiology, which means we need to extract the minimum and maximum value with 24 hours.

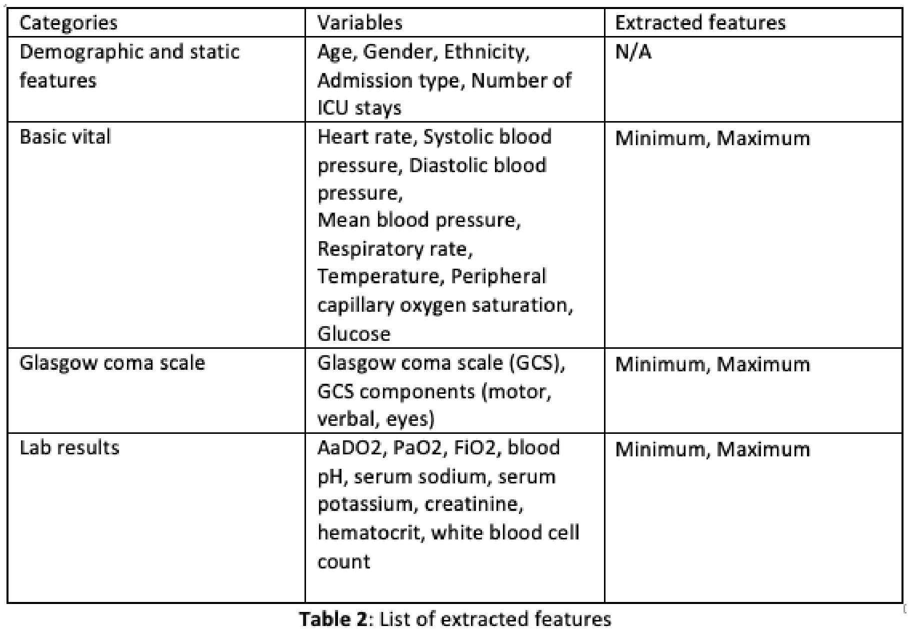


mimiciii.icustays		
row_id	integer	#
subject_id	integer	
hadm_id	integer	
icustay_id	integer	
dbsource	varchar(20)	t
first_careunit	varchar(20)	t
last_careunit	varchar(20)	t
first_wardid	smallint	#
last_wardid	smallint	#
intime	timestamp	d
outtime	timestamp	d
los	float8	#

mimiciii.labevents		
row_id	integer	#
subject_id	integer	
hadm_id	integer	
itemid	integer	
charttime	timestamp	d
value	varchar(200)	t
valuenum	float8	#
valueuom	varchar(20)	t
flag	varchar(20)	t

mimiciii.chartevents		
row_id	integer	#
subject_id	integer	
hadm_id	integer	
icustay_id	integer	
itemid	integer	
charttime	timestamp	d
storetime	timestamp	d
cgid	integer	
value	varchar(255)	t
valuenum	float8	#
valueuom	varchar(50)	t
warning	integer	#
error	integer	#
resultstatus	varchar(50)	t
stopped	varchar(50)	t

In order to extract the target variables, we first need a deep understanding on the data schema. According our observation, we found out that the main ICU admission data are included in table "icustays". And the variable required to calculated these 3 score are distributed in "labevents" table and "chartevents" table. And these two tables connect to "icustay" through a key called subjectID. Table 2 provides the list of extracted features used in the study.



Categories	Variables	Extracted features
Demographic and static features	Age, Gender, Ethnicity, Admission type, Number of ICU stays	N/A
Basic vital	Heart rate, Systolic blood pressure, Diastolic blood pressure, Mean blood pressure, Respiratory rate, Temperature, Peripheral capillary oxygen saturation, Glucose	Minimum, Maximum
Glasgow coma scale	Glasgow coma scale (GCS), GCS components (motor, verbal, eyes)	Minimum, Maximum
Lab results	AaDO2, PaO2, FiO2, blood pH, serum sodium, serum potassium, creatinine, hematocrit, white blood cell count	Minimum, Maximum

Table 2: List of extracted features

5 Predictive Variables

Next, a series of determining variables are collected when predicting survival time after discharge from hospital admission. The collected variables are extracted from queries to the database and in some cases to the preprocessing of these. The distribution in classes attends only to organizational criteria.

5.1 Demographic Information

Five variables of a demographic nature are collected: age, sex, marital status, religion and ethnicity. These variables are extracted directly from the ADMISSIONS table, except for age, which is calculated from the time difference between the date of birth, stored in the PATIENTS table, and the date of hospital admission, from the ADMISSIONS table.

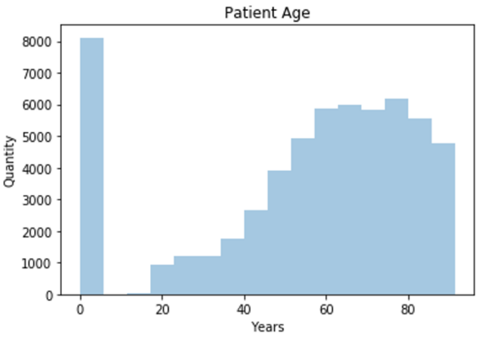
5.1.1 Age

The age of patients older than 91 years is displaced in time in order to protect their identity and make it difficult to identify you, in compliance with US privacy law, HIPPA. This Thus, we found elderly patients older than 300 years. Using a preprocessing function we substitute the age of these patients for 91 years. Later we will discard these records from the set of data that will be used to train the neural network, as it is considered unreliable and prone to inducing errors. So Likewise, neonates will also be discarded, as they present a very different medical behavior from that of the adult population.

SELECT hadm$_id$

$EXTRACT(epochFROM(admittimedob))/(3600 * 24 * 365)$
$ASage$
$FROMadmissionsto$
$INNERJOINpatientsp$
$ONa.subject_i d = p.subject_i d$

It is statistically distributed as follows:

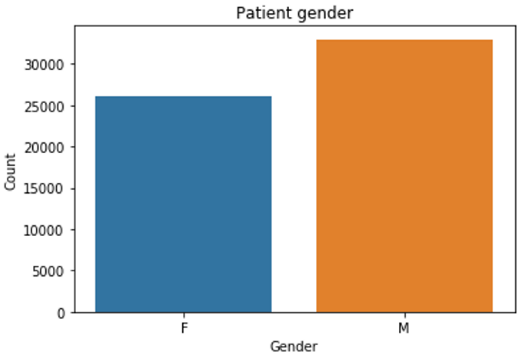


5.1.2 Sex

We extract this variable for each hospital admission directly from the database, without any type of prerequisite.processed, using the following simple query.

SELECT hadm$_i d, gender$
$FROMadmissionsto$
$INNERJOINpatientsp$
$ONa.subject_i d = p.subject_i d$

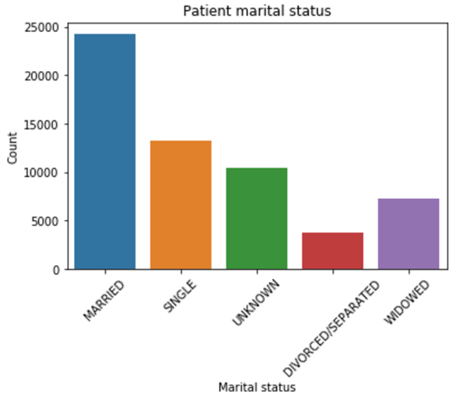
It is distributed as follows



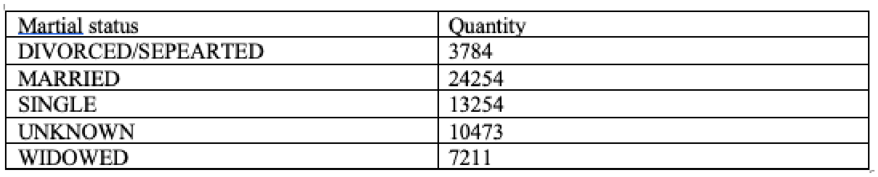
5.1.3 Marital status

We obtain it through the following query, analogous to the patient's sex.

SELECT hadm$_i d, marital_s tatus$
$FROM admissions to$
$INNER JOIN patients p$
$ON a.subject_i d = p.subject_i d$



We observe clearly unbalanced and insignificant classes that it is convenient to treat. For the preprocessing of this variable, we unify those groups with similar characteristics. Specifically, they come together DIVORCED and SEPARATED groups into one, and LIFE PARTNER is included within MARRIED. To do this grouping takes into account the life habits and social-demographic factors that can characterize each group. After making this grouping, we come to the following classes:



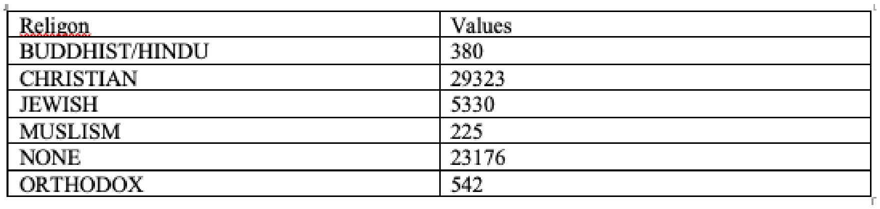
Martial status	Quantity
DIVORCED/SEPEARTED	3784
MARRIED	24254
SINGLE	13254
UNKNOWN	10473
WIDOWED	7211

5.1.4 Religion

We carry out a procedure analogous to that carried out in the variable MARITAL STATUS, taking into account the same considerations when unifying groups. We extract the variable from the database with the following query:

SELECT hadm$_i d, religion$
$FROM admissions to$
$INNER JOIN patients p$
$ON a.subject_i d = p.subject_i d$

In the same way as with the marital status, we obtain unbalanced and insignificant groups. Specific, we obtain 20 groups, 14 of which have less than a thousand records, out of a total of 59,000. After grouping them According to similar cultural characteristics, we arrive at the following groups:



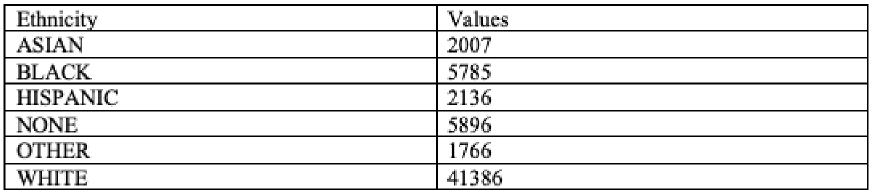
Religon	Values
BUDDHIST/HINDU	380
CHRISTIAN	29323
JEWISH	5330
MUSLISM	225
NONE	23176
ORTHODOX	542

5.1.5 Ethnicity

Carrying out the same process as for the previous variables, we extract and unify the ethnicity of the patient to every hospital admission.

SELECT hadm$_i d, ethnicity$
$FROM admissionsto$
$INNERJOIN patientsp$
$ON a.subject_i d = p.subject_i d$

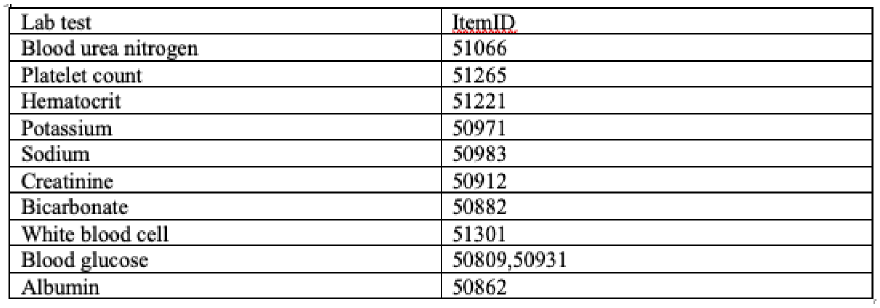
41 different ethnic origins are collected, some very similar to each other. For example, a distinction is made from Hispanics depending on the country, giving rise to numerous categories with less than ten entries. The same happens with patients of Asian and Caucasian origin. There are also records of patients with native North American origin present (72 records) or Caribbean native (9 records). These insignificant records are grouped under the OTHER category. The result of preprocessing the variable is as follows:



Ethnicity	Values
ASIAN	2007
BLACK	5785
HISPANIC	2136
NONE	5896
OTHER	1766
WHITE	41386

5.2 Laboratory tests

Ten common laboratory tests, performed routinely after a patient's admission, are drawn to use them as predictor variables. For each of them, we obtain their mean value and their standard deviation, giving rise to a total of twenty variables. Each of the test results is stored in the table LABEVENTS using an identifier, ITEMID. To obtain the average and standard deviation of each of these variables, for example for sodium in blood, The list of ITEMIDs for laboratory tests is as follows:

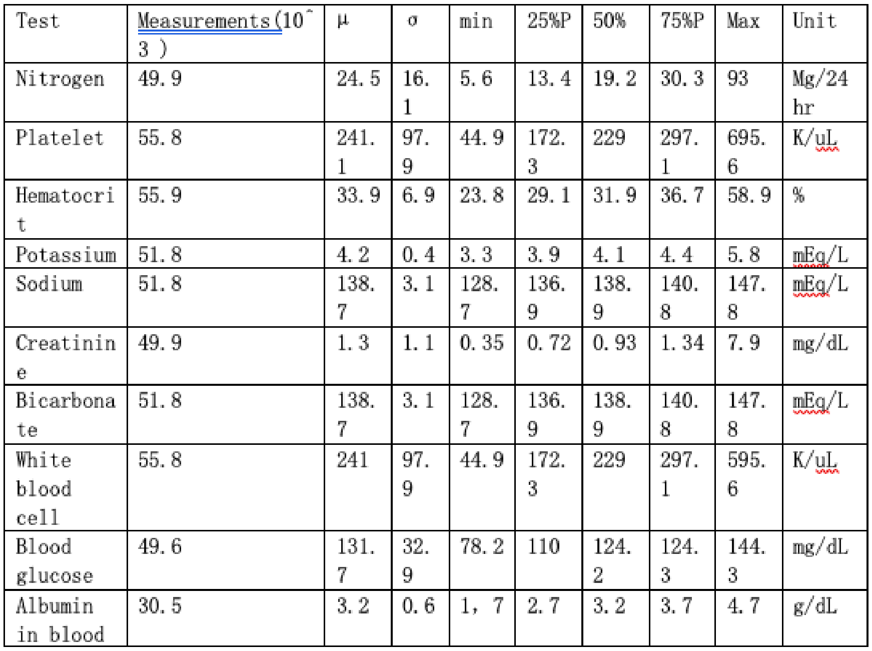


Lab test	ItemID
Blood urea nitrogen	51066
Platelet count	51265
Hematocrit	51221
Potassium	50971
Sodium	50983
Creatinine	50912
Bicarbonate	50882
White blood cell	51301
Blood glucose	50809,50931
Albumin	50862

we make the following query:

SELECT hadm$_i d$,
$avg(valuenum)ASAVG_SODIUM$,
$stddev(valuenum)ASSTD_SODIUM$,
$FROMlabevents$
$WHEREitemid = 50983$
$GROUPBYhadm_i d$

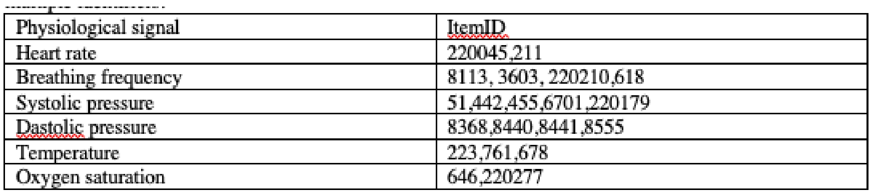
This function runs in a loop for all lab tests. As for preprocessing, we discard those values below the 1 percentile and above the 99 percentile, considering them aberrant errors or measurement failures, in addition to being insignificant. It is done by a function created for it. After extracting the variables and treating them, we obtain the following result:



Test	Measurements(10ˆ 3)	μ	σ	min	25%P	50%	75%P	Max	Unit
Nitrogen	49.9	24.5	16.1	5.6	13.4	19.2	30.3	93	Mg/24 hr
Platelet	55.8	241.1	97.9	44.9	172.3	229	297.1	695.6	K/uL
Hematocrit	55.9	33.9	6.9	23.8	29.1	31.9	36.7	58.9	%
Potassium	51.8	4.2	0.4	3.3	3.9	4.1	4.4	5.8	mEq/L
Sodium	51.8	138.7	3.1	128.7	136.9	138.9	140.8	147.8	mEq/L
Creatinine	49.9	1.3	1.1	0.35	0.72	0.93	1.34	7.9	mg/dL
Bicarbonate	51.8	138.7	3.1	128.7	136.9	138.9	140.8	147.8	mEq/L
White blood cell	55.8	241	97.9	44.9	172.3	229	297.1	595.6	K/uL
Blood glucose	49.6	131.7	32.9	78.2	110	124.2	124.3	144.3	mg/dL
Albumin in blood	30.5	3.2	0.6	1，7	2.7	3.2	3.7	4.7	g/dL

5.3 Physiological Signals

In the same way that we obtain the results of laboratory tests, we extract from the database the mean and standard deviation of six physiological signals to be used as predictor variables. The measurements have been taken with two different monitoring systems, Philips CareVue and Metavision. So The database itself distinguishes between measures taken automatically and measures expressly taken by the medical personnel, among other factors. That is why the same measure has multiple identifiers.



Physiological signal	ItemID
Heart rate	220045,211
Breathing frequency	8113, 3603, 220210,618
Systolic pressure	51,442,455,6701,220179
Dastolic pressure	8368,8440,8441,8555
Temperature	223,761,678
Oxygen saturation	646,220277

Physiological signals are recorded in the CHARTEVENTS table. We use the following query, very similar to the one used to obtain the results of laboratory tests. For example, to extract the desired values in the case of oxygen saturation we would use the following query.

SELECT hadm$_i$$d, avg(valuenum)ASAVG_SPO2, stddev(valuenum)ASSTD_SPO2,$
$FROMcharteventsWHEREitemidIN(646, 220277)GROUPBYhadm_id$

We apply the same processing used previously, that is, we discard those values below the 1th per-centile and those above 99th.

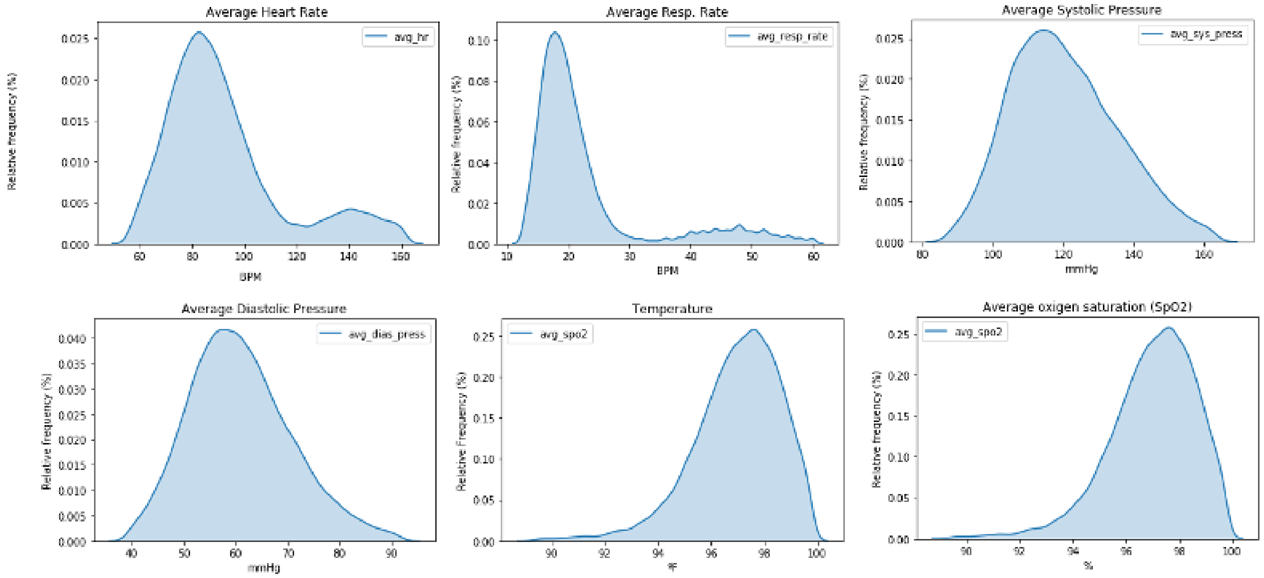
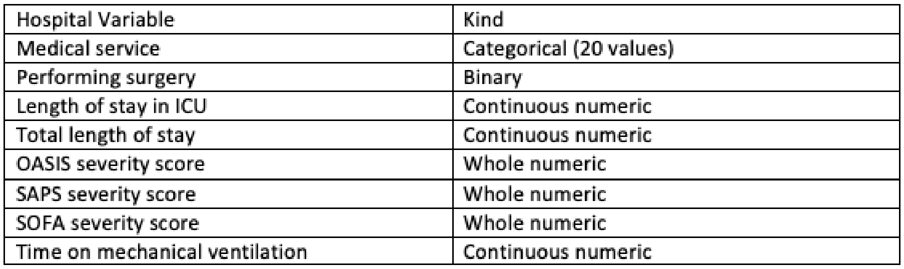


Figure: Probability distribution of the physiological signals extracted

5.4 Hospital information

Eight variables related to each hospital stay were extracted:



Hospital Variable	Kind
Medical service	Categorical (20 values)
Performing surgery	Binary
Length of stay in ICU	Continuous numeric
Total length of stay	Continuous numeric
OASIS severity score	Whole numeric
SAPS severity score	Whole numeric
SOFA severity score	Whole numeric
Time on mechanical ventilation	Continuous numeric

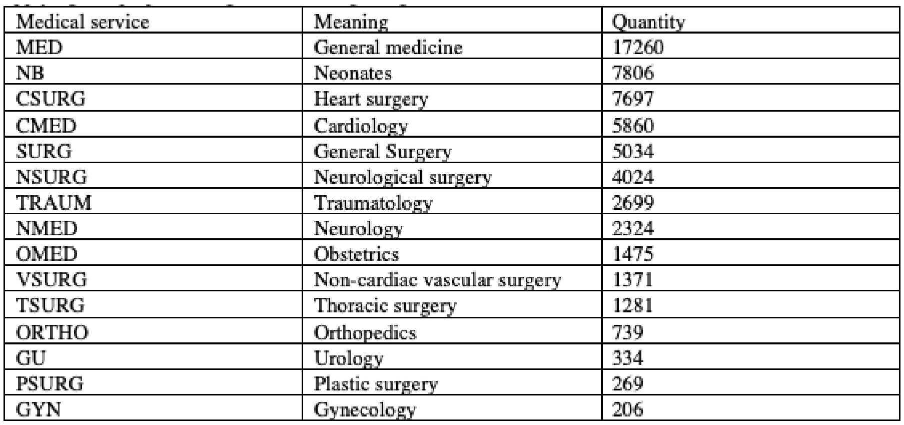
5.4.1 Medical service

This is a categorical variable that indicates the most relevant medical service for which it is cared for the patient in the hospital stay. Because on numerous occasions a patient remains in more than one service during your stay, a preprocessing function is required to extract the service from the largest importance based on a criterion.

Specifically, a function has been designed that uses the following priority to extract a single service for each hospital stay.

- Specialized surgery services
- General surgery service
- Specialized service
- General medicine service

In this way, a patient admitted to the general medicine service and later transferred to the cardiac surgery, will appear as a patient treated under the cardiac surgery service only, for example. This allows reducing the dimensionality of the variable and obtaining the most relevant information. After applying this preprocessing, the following categories and counts are obtained.

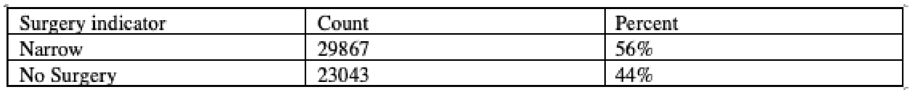


Medical service	Meaning	Quantity
MED	General medicine	17260
NB	Neonates	7806
CSURG	Heart surgery	7697
CMED	Cardiology	5860
SURG	General Surgery	5034
NSURG	Neurological surgery	4024
TRAUM	Traumatology	2699
NMED	Neurology	2324
OMED	Obstetrics	1475
VSURG	Non-cardiac vascular surgery	1371
TSURG	Thoracic surgery	1281
ORTHO	Orthopedics	739
GU	Urology	334
PSURG	Plastic surgery	269
GYN	Gynecology	206

5.4.2 Performing surgery

To detect whether surgical interventions have been performed on a patient during their hospital stay we will use the surgery indicators, (Surgery Flags), provided by the HCUP, Health- care Cost and Utilization Project, an initiative funded by the US government through the 'Agency for Healthcare Research and Quality '(AHRQ) dedicated to the management and analysis of medical data. This entity provides tools to identify interventions and surgical events using ICD-9 codes of procedure or CPT (Current Procedural Terminology) codes, both present in the MIMIC-III database It allows the classification of procedures into three groups:

- NARROW: Invasive therapeutic surgical procedures requiring incision, excision, manipulation or suturing of tissue that penetrates or passes through the skin, typically performed in the operating room and under local anesthesia or general or sedation.
- BROAD: Surgical procedures that cannot be classified as those included in the NAR- indicator. ROW, but they are performed under surgical conditions. This group includes surgical procedures of different agnostic, such as endoscopic or percutaneous procedures, or those performed through natural orifices. These are less invasive interventions.
- NEITHER: Procedures not registered as NARROW or BROAD, that is, non-surgical procedures.



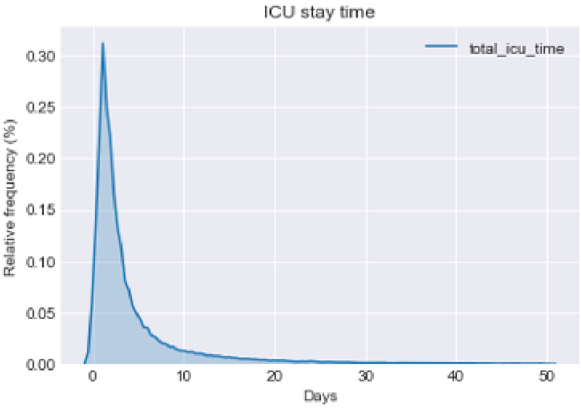
Surgery indicator	Count	Percent
Narrow	29867	56%
No Surgery	23043	44%

5.4.3 Length of stay in ICU

From the database it is possible to directly extract the length of stay in ICU in days for each patient in the same hospital admission. This information is in the ICUSTAYS table and we obtain it through the following query.

SELECT hadm$_id$,
$sum(los)AStotal_icu_time$
$FROMicustays$
$GROUPBYhadm_id$

It is necessary to use the aggregate function of addition in the consultation because on certain occasions a patient is admitted in the ICU, he is transferred to another section and later returns to the ICU, with which different durations are recorded for the same stay. In this way, we obtain a continuous variable, to which we do not apply preprocessing.



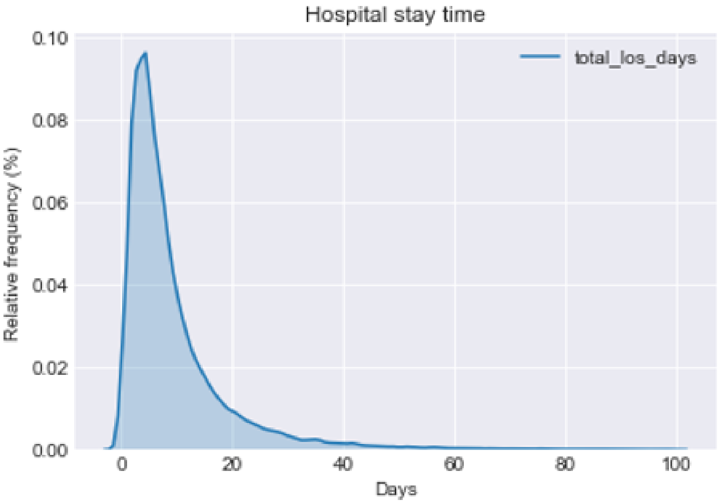
Probability distribution of ICU stay

5.4.4 Duration of hospital stay

We obtain the length of hospital stay, including the duration in ICU, such as the difference between admission and discharge time. For this we use the EXTRACT function and epoch, typical of PostgreSQL.

SELECT hadm$_id$,
$EXTRACT(epochFROM(dischtime - admittime))/(3600 * 24)AStotal_{l}os_{d}ays$
$FROMadmissions$

This variable is also measured in days and does not require preprocessing. It is distributed as follows:



Probability distribution of the entire hospital stay

5.4.5 Severity indicators

Different indicators of severity have been developed in order to predict hospital mortality from of the information of the patients, in particular of the measures taken during the first 24 hours of their admission.

However, they have certain limitations, for example by relying on subjective measures taken by staff medical or employing linear relationships that do not adapt to reality. We will use as predictor variables the indicators SOFA, SAPS and OASIS, which we will obtain through the next query:

SELECT o.hadm$_i d$,

$AVG(o.oasis)ASoasis_avg,$

$AVG(so.sofa)ASsofa_avg,$

$AVG(sa.saps)assaps_avg$

$FROMoasisor$

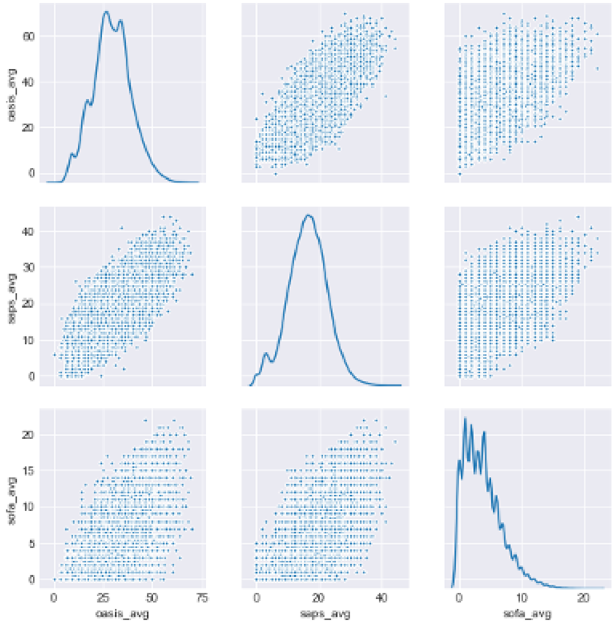
$INNERJOINsofaso$

$ONo.hadm_id = so.hadm_id$

$INNERJOINsapssa$

$ONsa.hadm_id = so.hadm_id$

$GROUPBYo.hadm_id$



Pairplot of severity scores

5.4.6 Time in mechanical ventilation

Using again a materialized view available in the repository code of MIMIC-III, we obtain the time that each patient spends on mechanical ventilation during their stay hospitable. We use the following query on the VENTDURATIONS view.

SELECT hadm$_i d, SUM(duration_h ours) AStotal_m ech_v ent_t ime$
$FROMventdurationsv$
$INNERJOINicustaysi$
$ONv.icustay_i d = i.icustay_i d$
$GROUPBYhadm_i d$

Sometimes a patient spends time connected to the mechanical ventilator, is disconnected, and reconnected later. That is why the materialized view sometimes records different entries for the same stay in ICU, so it is convenient to calculate the sum of durations for a stay.

6 Prediction Modeling

The ultimate goal is to build a model framework to predict in-hospital mortality. For the first try, a binary classifier was trained using the extracted features listed in Table 2 to predict in-hospital mortality. 49,632 ICU stays of our study population were split into 80 percent of training set and 20 percent of test set. Random forest classifiers were trained on the training set to predict the in-hospital mortality label (around 12 percent were dead) using 24-hour ICU stay data. Hyperparameter tuning was performed using grid search on 5-fold cross-validation (CV) of the training set. The model performance of the best classifier resulted from the grid search were then compared and evaluated on the test set.

Secondly, we trained a Logistic Regression Model with the same test-train split, hyperparameter tuning and CV techniques. Experimental results in the later sections show that both of machine learning models provide us a base to identify high-risk patients who might be dead within 24-hour since ICU admission in the early stage of ICU stay.

Finally, we used neural network. A simple, basic, fully connected neural network with four layers, and four neurons per layer is implemented using Keras. After this initial success, the network should be tuned to increase its performance in order to make better predictions. Predictive power is determined by several factors, mostly network parameter tuning. Tuning is performed in an iterative manner, by trying out several configurations until we find an optimal one that maximizes accuracy. The following parameters were tuned:

- Number of hidden layers
- Epoch
- Batch Size
- Optimization algorithm

Three main tuning strategies can be used to accomplish this task. The most time consuming, Grid Search, involves trying out all possible combinations of parameters. This is often unfeasible in complex scenarios, as the combinations may easily exceed 10k, which given a training time of about 5 mins would mean that we need to wait for months to find the optimal parameter configuration. An alternative to Grid Search is Random Search, which tests a predefined number of random configurations, maybe 50 or 100, and picks the best one.

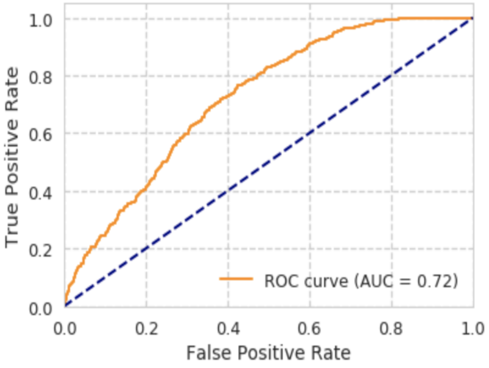
After 50 iterations, the optimal parameters are gifted to us: 6 hidden layers, 8 neurons per layer, a batch size of 50, epoch of 25 and Adam algorithm as optimization function.

7 Evaluation

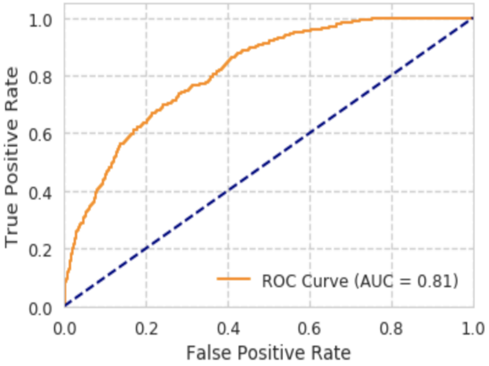
The dataset was split into 80 percent of training set and 20 percent of test set. Hyperparameter tuning was done on 5-fold CV of the training set and the final evaluation of model performance was done on the test set. Multiple machine learning models in this project were evaluated and compared using the Area under the Receiver Operating Characteristic curve (AUROC) on the test set. The ROC curve is the true positive rate against the false positive rate at various threshold settings. AUROC provides a single measure of the diagnostic ability of a binary classifier as its discrimination threshold is varied. We have used AUROC to compare the model performance of the binary classifiers.

8 Experimental Results

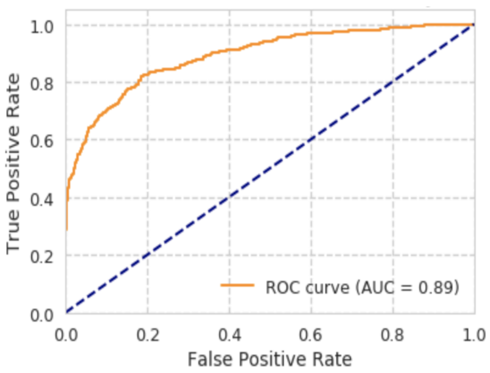
ROC curve for Logistic Regression Model:



ROC curve for Random Forest:



ROC curve for Neural Network:



9 Discussions

Even though the main objective has been achieved, earning my bachelor's degree, some improvements could be made to boost the model's performance and achieve better prediction capabilities. For example, more features could be added, and Long Short-Term Memory Network (LSTM) might can be deployed to handle time series data instead of using aggregation of features

In addition, results obtained are great, but we are not sure it is conclusive enough to be used in medical practice Even though the mathematical reasoning behind many algorithms used in AI tools is clear, its complexity has made it difficult to comprehend it under the hood. If we are willing to put our lives in the hands of AI, we must make sure we fully understand how this tool works, as well as creating a well thought out legal regulation around its use.