Mortality Analysis across ICU patients between 2001 to 2012

Aiyu Liu[1], Akhil Jalla[1], Jasmeet Khalsa[1], Ellis Pridgeon[1], and Jon Pont[2]

[1]Department of Computer Science, University of Bristol
[2]Department of Engineering Mathematics, University of Bristol

***Abstract*— Using the MIMIC-III dataset, useful features are extracted with the aim of predicting mortality for admitted patients. Due to the complex and unstructured nature of the MIMIC-III dataset, extensive preprocessing is required. Multiple sampling methods are considered to mitigate the effects of the class imbalance present in the dataset. Throughout the project, the security and confidentiality of the patients is protected. Additionally, text analysis and natural language processing techniques are applied to each patients doctor's note. Finally, machine learning models are applied to the processed data and are shown to give acceptable estimates as to a patient's mortality.**

I. Introduction

Predicting in-hospital mortality for patients in Intensive Care Units (ICUs) is crucial for determining the value and severity of treatments and diagnoses. A patient's medical condition is continuously monitored, making the ICU a data rich environment.

The APACHE system is one that makes predictions regarding patient mortality based on ICU data [1]. Initially, APACHE relied solely on rules derived from domain experts but has since been updated to become far more data-driven. APACHE's development has given rise to other predictive systems. One such example of a predictive system is the Simplified Acute Physiology Score which has utilised statistical modelling techniques [2].

Despite many extensions and modifications to SAPS, it still suffers from overestimation of in hospital mortality prediction [3]. It was found that both SAPS-I and SAPS-II both were unreliable in their predictions (from 28000 admissions to 10 different ICUs in Italy), and that SAPS-III has a greater number of overestimations than its previous version [4].

Data mining, machine learning and other statistical methods have been applied extensively in recent years to the medical field. The expansion can be attributed to advances in machine learning as well as data archiving. The aforementioned APACHE and SAPS systems have been limited by technical and practical considerations [5]. However, the increase in electronic data collection as part of routine medical practise has opened the door to many more studies.

During this investigation, a range of models are considered for the goal of prediction including Logistic Regression and Decision Trees [6][7]. The methodologies of these studies varied in their approaches, but all showed better performance against a baseline. A logical progression would be to combine models in an ensemble approach. The cleaning technique in our scenario involved the selection and omittance of various features in the chosen dataset, such as HOSPITAL_WARD and LENGTH OF STAY. With such a variability in selection, it is important to pursue all possible relations within the data which could correspond assist to predict mortality. In doing so, various data science techniques are applied to give valid justification where suited. Using this approach a reliable evaluation over the different models implemented. An efficient evaluation of a model that performs successfully in achieving the aim could prove advantageous for hospital administrators, physicians and government bodies alike.

This project aims to provide a pipeline taking in the MIMIC-III dataset and returning prediction of mortality scores alongside an evaluation of the outputs. All the steps that will be taken to achieve this are shown in Figure 1 which provides a high-level visualisation of the steps taken to achieve this goal.

II. Dataset

The two most well renowned ICU datasets are the MIMIC-III (Medical Information Mart for Intensive Care III) dataset and the Challenge-2012 dataset [8], [9]. The Challenge-2012 dataset is a smaller, cleaned and publically available version of the MIMIC-III dataset. The MIMIC-III dataset is far larger and requires basic medical and ethical training for access to be granted.

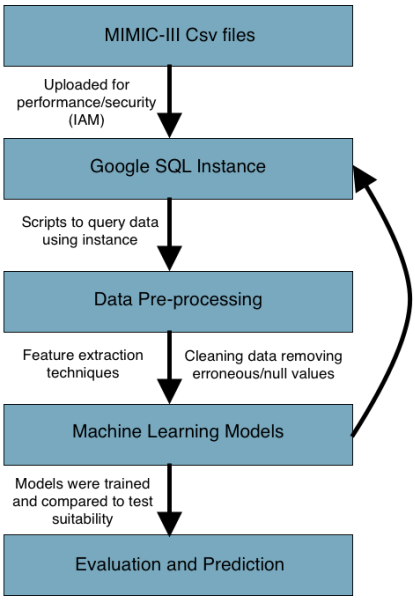


Fig. 1: Data Analysis Pipeline for working with the MIMIC-III Data

1) Challenge Dataset: The Challenge Dataset was extracted from the MIMIC-III dataset consisting of 12,000 subjects, whose age at the time of ICU admission was 16 or over. The data itself is pre-processed comprising of the last 48 hours of a specific patient's stay. In the challenge dataset, the dataset is randomly divided into three groups each having 4,000 patients. The first group acts as training set, the second acts as an open test set and the third set is the hidden set. This final set is commonly used as a further test set for various modelling and visualisation projects.

2) MIMIC-III : The MIMIC-III dataset is a set of relational databases containing data corresponding to patients admitted within any intensive care units at Beth Israel Deaconess Medical Center. The data is widely accessible to researchers internationally under a data use agreement. The data lists de-identified health related data, which contains information such as demographics, procedures and medications used amongst many other fields. Prior to requesting access to the MIMIC-III dataset, a CITI Data or Specimens Only Research course has to be completed. Upon completion data can be retrieved from a secure file server utilising authorised credentials. [8][10].

In contrast to the challenge dataset, the MIMIC-III dataset poses numerous challenges in that it contains many null and erroneous values. The amount of data available for each patient varies greatly due to the variations in condition and length of stay within the ICU. Furthermore, the challenge dataset contains training and test sets, whereas these sets had to be composed by ourselves.

A. Data Volume

Various assumptions are necessary to clean, justify and process the set used.

The challenge set stores their relations under one table whereas the MIMIC-III is stored upon 25 different tables in a '.csv' format. The table CHARTEVENTS contains records for 58,000 ICU patient admissions between the years 2001-2012 which comprise over 40GB of raw data. The size of the dataset adds complexity to the process of querying the respective tables. A Google Cloud SQL instance is used here to host and query the data.

The goal for this initial analysis was to figure out relations in data across the tables and record key features which could be helpful in reaching our aim.

B. Data Quality

Unlike the cleaned Challenge set, the MIMIC-III dataset has many rows where the data is not present, for example in ADMISSIONS, the columns LANGUAGE, RELIGION and MARITAL_STATUS do not contain data for all the patients and therefore are inconsistent. Some of the important tables such as DRGCODES, ICUSTAYS which will be used for feature extraction also contain inconsistent data, such as in table fields DRG_SEVERITY, DRG_MORTALITY and LOS. These anomalies are handled in section III-B.

C. Data Heterogeneity

The challenge set has data for a subset of patients wherein the data is distributed in a single table and is for the final 48 hours of the patient history in the hospital.

In case of MIMIC-III dataset, the data is distributed over 25 tables where each table is linked to one or more tables (Figure 13 in Appendix B). The data is not restricted to the final 48 hours of the patient in the hospital but contains the full data from the time of admittance to time of discharge or death.

D. Ethics

Various tables within the MIMIC-III dataset contain information for each patient regarding protected characteristics such as RELIGION and MARITAL_STATUS amongst other personal information such as LANGUAGE. Here, these features are considered so to ascertain their relevance in predicting mortality. Although it is unexpected that `LANGUAGE` would correlate to

mortality, another characteristic 'Age' would intuitively correlate with mortality. Any implementation of this work would have to further take into consideration the use of these features.

III. Data Pre-Processing

A. Data Security

To protect the sensitive information contained in the MIMIC-III dataset, the data is stored and encrypted on the server side, before it is written to the disk. As well as providing a secure environment in the cloud, a Google SQL instance further allows easy access for data cleaning and pre-processing. Since these are cloud based services, a Google Cloud proxy client was required in order to access the datasets. Google Cloud Identity and Access Management (IAM) provides access rules that are applied onto the aforementioned cloud services such that only authorised users have permission to read or write to the dataset.

B. Cleaning

Gathering model predictions from the MIMIC-III data outright is not feasible. Application to this dataset would incorporate erroneous values including missing and invalid data. Firstly to highlight errors in the data, one must consider to what extent of MIMIC-III data will actually be utilised.

1) Key Fields and Tables: In terms of data contribution, the CHARTEVENTS table provides the majority of patient data with 330,712,483 different rows. On reflection of querying the data and evaluating, an extremely large portion of ITEM values turned out to be insignificant when it came to the patients admission. These ITEM values are identifiers for a single measurement type. Thus when it came to creating the pre-processed clean table, CHARTEVENTS had to be discarded, due to the lack in unique (per-table) information to contribute to mortality prediction as well as redundancy. For example, there are many ITEM identifiers corresponding to bed rotation. 'Bed Rotation': 6282, 'Bed rotation': 6330, 'bed rotation': 6333, 'Bed Rotation.': 7443, 'Bed rotation.': 7582 and even a misspelled 'bed roation': 7907. Many other items are also not intuitively related to mortality, identifying all of these and removing redundancy would require an extensive amount of time and resources.

The ADMISSIONS table in particular provided invaluable information regarding patient stay, where SUBJECT_ID acts as the primary key allowing queries to be made across the distributed table structure. A range of potentially useful features across this structure were included, these features are analysed in Feature Selection (section III-C) to discover their associated importance in achieving the aim.

After gathering this subset, assumptions had to be made about normalising the data to infer reliable results. Many patients had inconsistent number of stays in the ICU units. This study utilised a single admission, the utilisation of multiple will require an understanding of readmission patterns beyond the scope of this study. Hence, only the last admission could be considered. International Statistical Classification of Diseases provides a clinical diagnostic identifier ICD9_CODE that describes the patients current condition in the care unit. These codes incorporate thousands of specific conditions and thus would be unfair to only consider them individually. The reason behind this is motivated in Feature Selection (section III-C), as often specific conditions would appear less and thus would not contribute to determining an important feature. Instead, a dictionary was created to categorise related conditions in the format given[1]. These categories allow consideration of particular system sub-sets (e.g. Respiratory/Cardiovascular), and their effect into the output of mortality.

Null Values are replaced with zeros where applicable, for example in flags corresponding to death within the hospital and billing costs related to condition severity. In other areas assumptions had to be made, such as with the length of stay, where the null values are replaced with a mean value in order not to discount this information. The reason for inclusion in this instance is because such a small minority of admission values have null length of stay. Following this, further normalisation techniques are investigated below.

C. Feature Selection

There are over 100 features present in the MIMIC-III dataset out of which one can expect few to be related to mortality. In order to identify and attain features, 5 tables were combined, these being PATIENTS, ADMISSIONS, DRGCODES, ICUSTAYS, DIAGNOSES_ICD. Few of the features were dropped for further analysis of the dataset by using non-statistical means such as DIAGNOSIS. DIAGNOSIS is not selected for further analysis as it only has plain text data and multiple fields has similar data with minimal difference. Further it is observed that ICD9_CODE was

[1]ICD9 Code categories: https://icd.codes/icd9cm

more consistent than DIAGNOSIS. Some of the other features dropped with this process include:

- ROW_ID - the index identifier of the each table in the MIMIC-III dataset
- DBSOURCE - indicates whether the patient's information is present in the new database or the old one
- HAS_CHART_EVENT_DATA - a binary flag signifying if the patient's chart events were recorded or not

Additionally, INTIME and OUTTIME are dropped as our work is not based on time series prediction.

Statistical analysis is performed to extract features which are highly correlated to mortality. Univariate Feature Selection is implemented to examine each feature individually to determine the strength of the relationship of the feature with the output variable, EXPIRE_FLAG. Based on this method, a scoring matrix for each of the features showing its correlation to EXPIRE_FLAG is constructed. For the model presented here, 13 features are selected which show the best correlation with our output variable. This helped to remove some of the features which could be expected not to be related to mortality such as ETHNICITY, MARITAL_STATUS, RELIGION, INSURANCE and so on. Table 1 shows the scoring matrix selected for a cleaner dataset. These features are further utilised by machine learning models for predicting mortality.

Feature Name	Score
ICD9_CODE	16691.72
DRG_CODE	6621.18
ADMISSION_TYPE	1879.94
LOS	1634.69
DRG_MORTALITY	564.69
LAST_CAREUNIT	400.69
FIRST_CAREUNIT	358.61
DRG_SEVERITY	231.34
FIRST_WARDID	229.30
LAST_WARDID	215.34
LANGUAGE	202.09
ICD9_CAT	169.29
ADMISSION_LOCATION	156.21

TABLE I: Relevance score of 13 best feature used for the cleaned dataset

D. Normalisation

Since each feature has a completely different range of values, normalisation is required. This is common for many machine learning estimators: they might behave badly if the individual features do not approximate to distributed data. The Standard Scalar method is chosen for this task, adjusting the distribution such that each data point is shifted by the mean and standard deviation of the respective feature – resulting in a mean of 0 and standard deviation of 1 [11]. The Standard Scalar is used instead of the Min-Max Scaling method which computes the difference between the min and max values as a normalisation factor to the underlying data. The former is chosen, due to the increased performance in suppressing the effect of outliers [11].

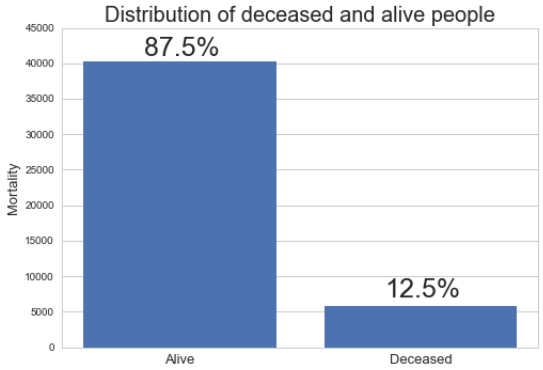


Fig. 2: Percentage of alive and deceased people

E. Sampling

As Figure 2 shows, the dataset is largely imbalanced with over 41000 alive records and under 6000 deceased records. Training the model will require an even ratio between deceased to alive patients. An imbalance of these sizes will cause significant bias weighted in the direction of the larger subset [12], which in this case proved to be the alive subset as seen in Figure 2. In order to mitigate this, a range of undersampling and oversampling techniques are explored.

Before any sampling method is utilised, the data is first split, with 80% reserved for the training set and the remaining 20% for the test. The splitting of data will be done in a stratified fashion for variance reduction[13].

1) Undersampling - Random undersampling of majority class: Figure 2 shows a clear class imbalance in the data. A simple technique to class-balance the training set is to randomly undersample from the alive class. The test set maintains the class imbalance and will test the classifiers ability to distinguish between the classes on a dataset representative of the real-world data. The main disadvantage to this method is that data about the majority class is discarded [14].

2) Undersampling - NearMiss [15]: NearMiss performs undersampling of points in the majority class

by calculating the distance to other points in the same class. The points from the majority class are retained whose mean distance to the k-nearest points in the minority class is lowest. In our case, k is arbitrarily set to 5. The rest of the points are then removed. NearMiss aims to undersample more cleverly than random undersampling by only selecting points which are most relevant in the majority class and assuming they represent the class as a whole. Once again, by undersampling data from within the majority class is disregarded.

3) Undersampling - K-Means Clustering [16]: K-means clustering enables the grouping together of similar data points. Each cluster will have a centroid, an n-dimensional point in space that is minimally distant from all points in the cluster. The centroids is assumed to represent the cluster, thus can replace numerous data points, reducing the dimensionality. The 30,232 alive training samples are clustered into 4,309 clusters, to match the number of deceased samples, and the centroids of these clusters are saved as artificial data points. The resulting dataset contains 4,309 synthetic alive data points and 4,309 genuine data points. Again, data is discarded and a new issue arises in the form of using only synthetic data for one class. However, the synthetic data should efficiently summarise the training data and enable effective training [17].

4) Oversampling - Random oversampling of the minority class: This method samples points from the minority (deceased) class randomly without replacement. In essence, it supplements the training data with multiple copies of some minority class instances. The issue however, is that this method is susceptible to overfitting due to the repeat occurrences of many data points in the final dataset [14].

5) Oversampling - SMOTE: Synthetic Minority Over-sampling Technique(SMOTE) creates a balance by increasing the data points for the deceased subset.

In order to create a synthetic instance, it randomly selects one of the k-nearest neighbours of each minority instance and then calculates linear interpolations to produce a new minority instance in the neighbourhood [18]. The value of k was selected to be 5. The negative of this approach however, is that the model is trained upon synthetic instances of the minority class, where assumptions have been made by SMOTE on the relations of near points. As such, it may be possible that the model is fitted to synthetic data, which may not necessarily be representative of real world data [19].

The overall sampling method that is chosen relies on a few factors. These include, recall, F1 score, accuracy and whether it influences our model to over/underestimate mortality.

F. Natural Language Processing

Every admission into the ICU contains a doctor's note describing the patient's previous medical history, current condition, prescribed medication and details of any health conditions amongst other things. The notes are contained in the table NOTE_EVENTS, where the notes are merged with other features using SUBJECT_ID as the foreign key. To extract relevant information from these notes, numerous processing steps are required. Efficient vectorisation of the notes is necessary to enable their use in any further models. All non-text characters and symbols must be removed and the remaining text split into words for use in a natural language model.

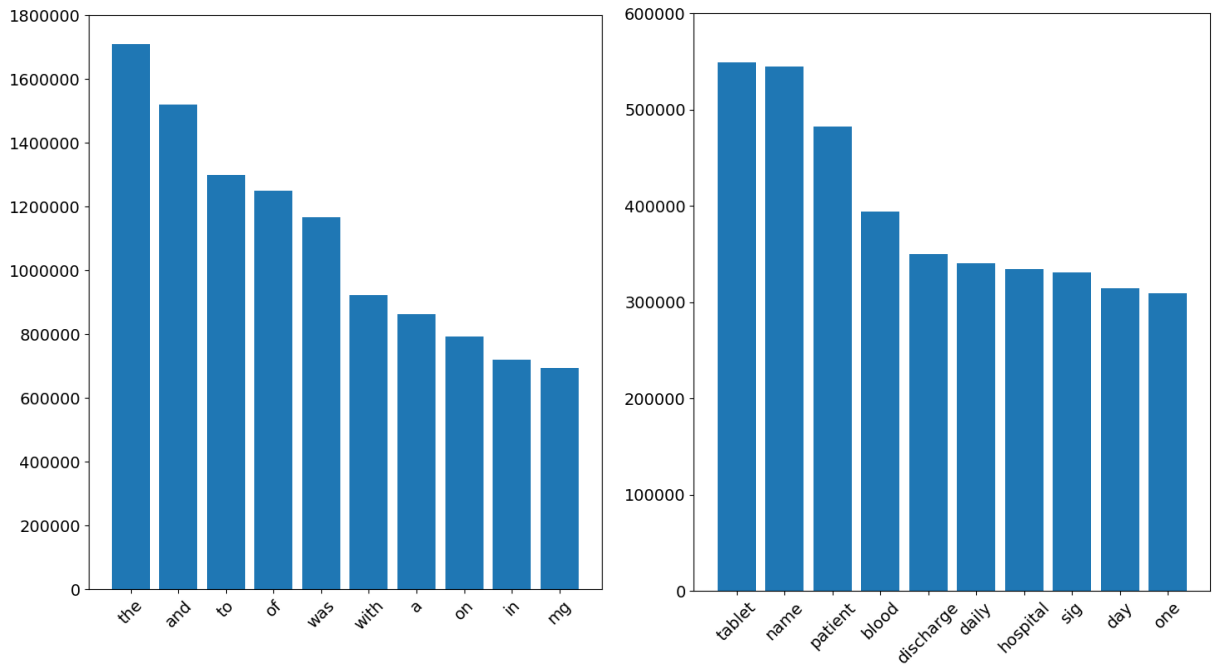
Notes for each ICU stay are extracted NOTE_EVENTS. A handful of admissions do not contain any doctor's notes and are not included in the text processing. Each note is cleaned removing all numbers, symbols and special characters. The text is parsed to tokenise all words, using whitespace as the separator. Finally, a vocabulary is constructed made up of each unique word contained in the cleaned notes.

The vocabulary defined on cleaned notes contains 126,000 unique words, not far from the 171,476 words in the English language[2]. The size of the vocabulary suggests that it could be refined.

To reduce the vocabulary size, all one and two letter words are removed and so are the English 'stop-words' as these words carry little semantic meaning. Figure 3 shows the most common words used before and after the initial reductions were applied. The remaining corpus is fed through a spell-checker designed to eliminate the abundance of errors that appear in the notes.

The spellchecker builds a new vocabulary from the old via an iterative process. Any word contained in the English Dictionary is added to the new vocabulary. For each remaining word, every possible word an edit distance of 1 away from it is collected. From the possible words, the proportional use frequency of each word in the English language is considered so that the most common word is selected. Possible edit actions are delete, transpose, replace and insert for each letter. If no English word exists at an edit distance of 1, then

[2]Oxford dictionary: `en.oxforddictionaries.com/explore/how-many-words-are-there-in-the-english-language`



(a) Before word processing.

(b) After word processing.

Fig. 3: Top ten occurring words in the doctor's notes vocabulary.

words with an edit distance of 2 are considered. Further edit distances are not considered.

The spellchecker returns a vocabulary of 71,000 words, a significant improvement on the original vocabulary, but still too large for purpose to vectorise the notes efficiently. A more intelligent approach is needed to efficiently vectorise the notes for use as a feature or basis for clustering.

1) The word2vec Model [20]: The word2vec algorithm takes advantage of the similarity between words to enable more powerful text analysis. The word2vec algorithm is a two-layered neural network that is trained to map words to a high-dimensional vector space where words used in similar contexts are close to each other. Beyond capturing the vector space location of words, the word2vec algorithm can also be used to produce mappings for entire sentences or paragraphs. What these features represent is largely unknown but their definition enables the distance between notes to be calculated in euclidean space.

A vector representation of the doctor's notes is created following this method by taking gensim's word2vec model [21] and training it on all of the notes present in the randomly undersampled subset. The vector space representation of these notes are stored. This vector encoding of the notes enables them to be used as features for machine learning methods.

G. Models

To understand our choice of models, the definition of the problem needs to be understood. The time series data attained from MIMIC-III can be considered a multivariate stochastic process. Each measurement, or field (length of stay, ICD-9 code etc.) can be considered as a sequence of random variables at given times. For the sake of maintaining simplicity, many of our decisions have been to turn text and numeric data into categorical data. This allows us to make some inference from the data, without it being too sparse.

The goal here is a classification task where models are utilised to approximate a mapping function from our input variables to the discrete set of outcomes: deceased or alive.

1) Logistic regression [22]: Unlike linear regression logistic regression is capable of dealing with outliers using a sigmoid function. A logistic function is a standard sigmoid function which takes any real value between 0 and 1. This function is defined :

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

It is assumed that t is a linear combination of multiple variables such that is can be expressed as follows:

$$t = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_M x_N$$

If substituted in the logistic function it can be interpreted as the probability of the dependent variables being equal to a success or failure case.

2) Naive Bayes [23]: This is a probabilistic classifier inspired by Bayes thoerem where it is assumed that attributes (X) are conditionally independent.

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times ... \times P(x_n|C_i)$$

The classification works by attaining the maximum posterior, maximal to $P(C_i|X)$. The assumption mentioned above, reduces computational cost greatly, where it only counts the class distribution.

Naive Bayes can be scalable to large datasets and takes linear time, as opposed to other classifiers which use expensive iterative approximation.

3) Random forest [24]: A decision tree builds classification in the form of a tree structure. It utilises a mutually exclusive and exhaustive if-then rule in doing so. These rules are learned sequentially and one at a time from the training data. Once learned, the tuples covered by the rule are then removed and this process is continued until a termination condition is met.

Random forest is a supervised learning algorithm and in its simplest form builds multiple decision trees and merges them together to get a more accurate and stable prediction.

The training algorithm for random forests applies the general technique of bagging (or sometimes feature-bagging). Given a training set with output labels, it repeatedly selects a randomly sampled subset of input features with replacement of the training set and fits trees to these. This process occurs B times. For a classification with p features typically a subset of size $\sqrt{p}$ will be chosen.

```
For b = 1 to B:
    Sample with replacement,
    training samples as a random
    subset of input features
    along with the output labels

    Call these Xb, Yb

    Train classification tree fb on Xb and Yb
```

If one or more features are strong predictors for the output label, these features will be selected in many of the trees, causing their output to become correlated. An unseen input sample x', after training, can have its class predicted by taking the majority vote from individual trees.

The bagging procedure decreases the variance of model without increasing the bias and as such leads to better model performance. A single tree may be susceptible to noise with respect to classification, but by taking a majority vote it is less likely.

H. Optimisations

A few optimisation methods were considered in order to attain optimal classification performance.

1) Voting ensemble [25]: A voting ensemble combines the predictions from multiple machine learning models. Having pre-defined the aforementioned models, a voting classifier can then be applied to wrap those models and then average the predictions of the sub-models when making new predictions for unseen data.

Explanations seen for improvements achieved by voting classifiers is based on the separation of the expected error into a variance and bias term. The former measures error due to fluctuations in generating a single hypothesis, where by averaging over many of these the variance and thus the expected error can be reduced. The latter term measures the error that remains, even with the number of individually trained hypotheses tending towards infinity.

2) Gridsearch [26]: Hyperparameter optimisation is choosing optimal hyperparameters for a learning algorithm. This type of parameter affects the learning process of a machine learning model. Some parameter combinations will allow the model to perform more optimally than others.

Grid search is the traditional way of performing this type of optimisation, which is an exhaustive search through a specified subset of the hyper-parameter space of a learning algorithm. It's performance is typically guided by cross-validation of the training set. Grid search finds near optimal parameter combinations within given ranges [27].

In context of our problem, values between 1 and 100 were tested for the hyperparamter C corresponding to logistic regression. This relates to the inverse regularisation strength, where smaller values have stronger regularisation. Values between 20 and 200 for the number of trees in the random forest were applied as well as 5 fold cross-validation to measure the performance during training.

The entire methodology of the analysis over the MIMIC-III dataset can be seen in figure 1.

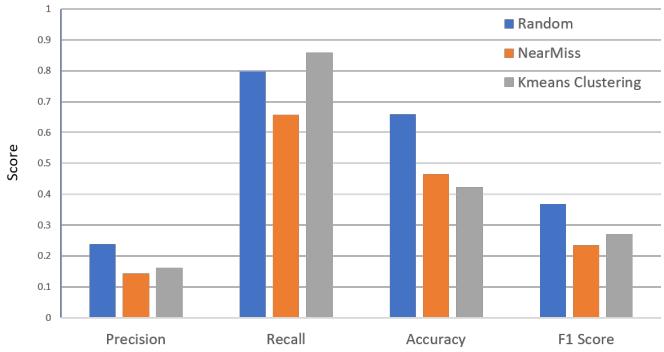


Fig. 4: Undersampling scores for each model.

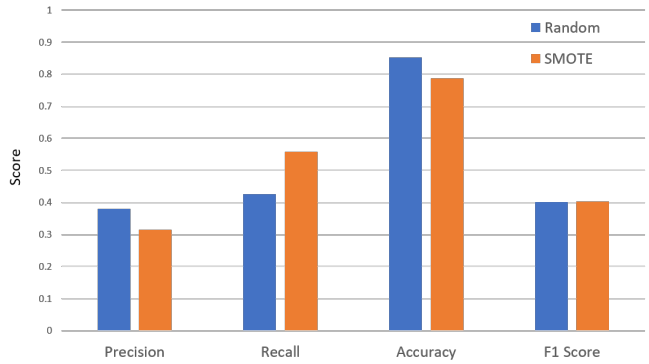


Fig. 5: Oversampling scores for each model.

IV. Models and Experiments

A. Sampling method

The oversampling methods produce a balanced training set containing 30232 of each class, with a proportionally representative test set. This contains 10078 members of the alive class and 1436 of the deceased.

Figures 4 and 5 show the precision, recall, accuracy and F1 score respectively. These scores are generated based on the sampling method. From these, the best two F1 scores correspond to the oversampling methods. Although SMOTE seems to detect a greater number of actual deceased patients (as shown in Figure 12), it also seems to overestimate the number of deceased significantly more than random oversampling seen in Figure 11. Conversely, the random oversampling method seems to overestimate the number surviving patients more than SMOTE. Yet with this method, the model managed to predict the number of survivals quite well.

In the context of being a stakeholder in mortality prediction, it could be assumed that overestimation of mortality could have negative consequences. From the sampling methods explored, this leads us to select random oversampling. Here the minority class of deceased patients is oversampled to be equal to the number of alive patients in the training set. Although such replication could lead to overfitting of the minority class (deceased patients). From the confusion matrix it seems that this was not the case when validated against the test set. With a larger dataset, perhaps there will be more variability, where the proportion of oversampling will be reduced in order to meet the majority class. As such, one could hope for more accurate predictive results.

Voting Classifier	
Recall	0.4262
Precision	0.3804
Accuracy	0.8518
F1	0.4016

TABLE II: Table showing evaluation metrics for Voting Classifier

B. Model evaluation

The model that is utilised is the Voting Classifier, which combines Random Forest, Logistic Regression as well as Naive Bayes.

The evaluation of the model relates to certain metrics. These include accuracy, recall and precision. An F1 score is then generated to balance recall and precision. A value close to 1 is ideal[28].

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

The results are shown in table II. The true positive rate (TPR) can also be plotted against the false positive rate (FPR) to generate a Receiver Operating Characteristics (ROC) curve; this being a probability curve where the area underneath represents the measure of separability [29]. This curve is presented in Figure 6 with the area underneath the curve (AUC) labelled. The value of AUC is 0.8039 (to 4 decimal places) which indicates that the model has a credible measure of separability. A value close to 0.5, indicates that a model is incapable of distinguishing between classes.

The vectors created in section III-F using NLP can be used as features for training the aforementioned machine learning models. Figure 7 shows that the feature vectors provide no information regarding mortality of the patients. The model is completely unable to separate the classes.

C. Discussion

To investigate the data further to evaluate its importance and integrity a 'Mortality Rate' is calculated

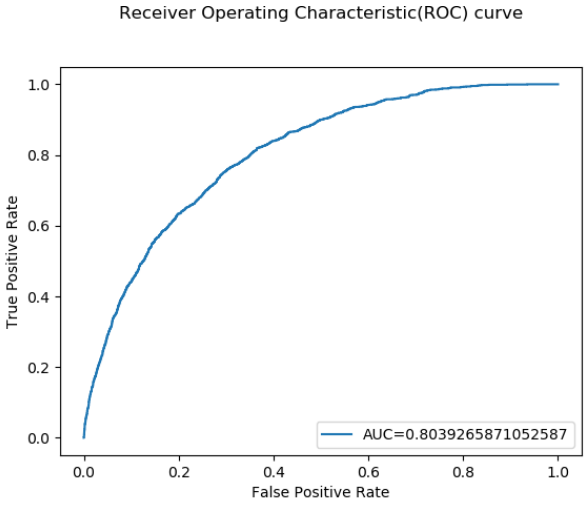


Fig. 6: ROC curve for Voting Classifier

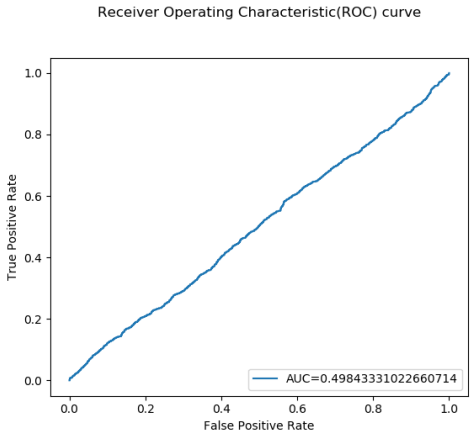


Fig. 7: Doctor's note model ROC curve.

based on the probability value given via the model. This allowed us to explore trends and potential patterns from features determined previously from uni-variate statistical analysis allowing the visualisation of their relation to mortality. Figure 8 has no distinct variations across the majority of ICD9-Categories, suggesting no specific category of disease or condition gives a large weightage towards mortality. Due to high diversity within these categories, resultant predictions exhibit high fluctuations as seen by the size of the error bars. However, significant troughs are seen from categories 0,14 and 15 corresponding to external causes of injury, congenital anomalies and perinatal period conditions respectively.

Secondly, in Figure 9 Mortality Rate is plotted against the different admission types to produce a similar bar chart visualisation. The trend in this data shows that the Emergency tag correlates the most with

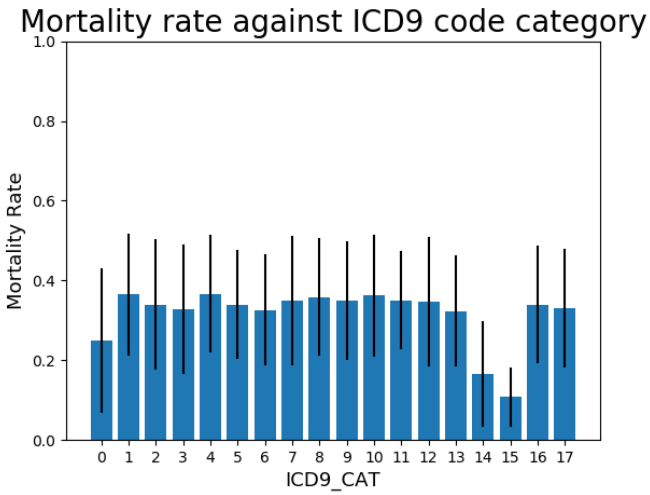


Fig. 8: Mortality plotted against the categories made of ICD9-Codes. Table III displays the correspondence from the dictionary values displayed and their actual category

	ICD9 categories
0	External causes of injury
1	Infectious and parasitic diseases
2	Neoplasms
3	Endocrine and metabolic diseases, and immunity disorders
4	Blood and related organs diseases
5	Mental disorders
6	Nervous system and sense organ diseases
7	Circulatory system diseases
8	Respiratory system diseases
9	Digestive system diseases
10	Genitourinary system diseases
11	Pregnancy, childbirth and puerperium complications
12	Skin and subcutanous tissue diseases
13	Musculoskeletal system and connective tissue diseases
14	Congenital anomalies
15	Perinatal period conditions
16	Systems, signs and ill-defined conditions
17	Injury and poisoning

TABLE III: Table showing evaluation metrics for Voting Classifier

the mortality with a value of 0.39. However, an accurate comparison would require normalising the data based on the size of data each ICU type has.

The hospital also issue mortality scores allowing for higher billing costs when a diagnosis is more severe, and vice versa. The mortality values: 0 to 4 correspond to these. The 0 value assumes there is no need for higher billing costs.

As expected a correlation can be seen with the label issued to our calculated rate. The 0 score indicates a distribution of probability (mortality rate) for mortality that did not incur large billing costs.

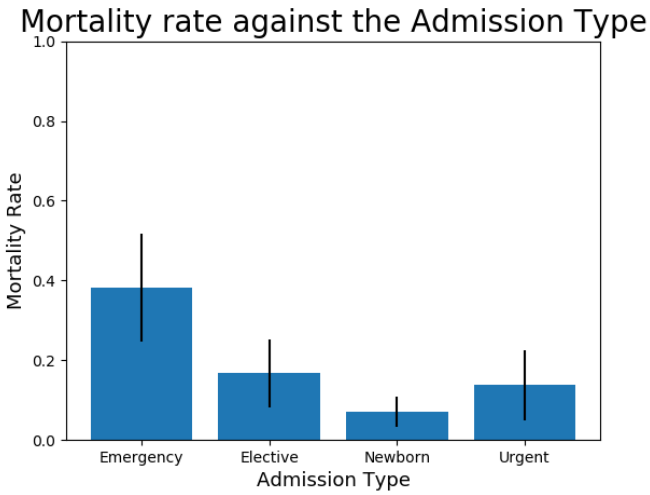


Fig. 9: Mortality plotted against the various admission types recorded

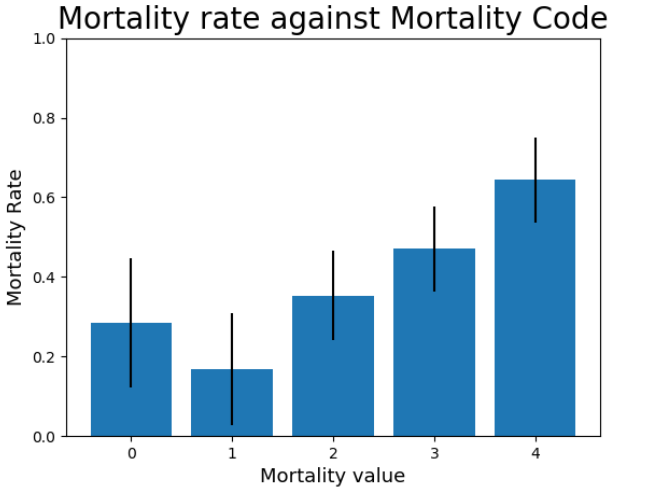


Fig. 10: Mortality plotted against the Mortality Code [0-4]

To suggest the benefit of combining doctor's notes to our current model, it was investigated separately. However, the NLP model's inability to distinguish between the classes, with specific reference to Figure 7, suggests that no meaningful information was pulled from the notes. It can be inferred that the language used in doctor's notes does not seem to be indicative of mortality as initially thought.

D. Issues

Some of the issues faced during the course of analysis were:

- The data imbalance between alive and deceased. More data points in the minority class would aid in accurate prediction.
- CHARTEVENTS could prove more useful, however there was no domain expertise to relay meaningful information from the table. Including all events to learn correlations to mortality would incur the curse of dimensionality.
- This study utilised a single admission, the utilisation of multiple will require an understanding of readmission patterns. This is beyond the scope, hence only the last admission is considered.

V. Conclusions

It is concluded that the features extracted from the data provided enables some prediction of the mortality of patients with a reasonable level of accuracy. The MIMIC-III dataset is complex and unstructured, but large enough that sufficient information can be retrieved. The model trained can separate a patient into classes with 80.4% degree of separability, a credible score. However, given the use case, the 20% rate of false positives and false negatives has serious negative consequences. There are instances where patients in the test set are given a high mortality score but go on to leave the ICU with their health. The model is not perfect or complete in its predictions and should not be treated as such. Were it to be used in a professional setting, it should only be used to provide a further level of insight to stakeholders, such as doctors, to help inform their decisions.

VI. Further Work

Ethical features of the dataset were all removed apart from LANGUAGE, largely removing any possibility of discrimination. Although it is not clear why LANGUAGE was a relevant feature, it would be worth investigating why it is considered important and what the effects of removing it from the model would be such that a completely non-discriminatory model could be provided to stakeholders.

The vocabulary produced by the spellchecker could be improved by trying to only identify relevant medical terms. Rather than comparing present words to the Oxford Dictionary it would be beneficial to compare them to a dictionary which only includes medical terms involving conditions, medications and other medical keywords. The hope would be that the vocabulary produced would be smaller and more relevant to the health conditions of the patients, thus providing more relevant features for the model.

The vectorisation of the doctors' notes enables the clustering of the notes based on content. K-Means

clustering is often used to achieve this but has not been pursued here. It would be of interest to assess patients who get clustered together to identify what common features they possess and if this could be useful in identifying mortality for future patients who would be assigned to their cluster.

A domain expert is required to enable the utilisation of the CHARTEVENTS table, which appears to contain a lot of relevant information hidden amongst a significant amount of noise and irrelevant information. Detailed data for each patient is available and would no doubt assist in the prediction of mortality if it could be identified.

REFERENCES

[1] W A Knaus, J E Zimmerman, D P Wagner, E A Draper, and D E Lawrence. APACHE-acute physiology and chronic health evaluation: a physiologically based classification system. *Critical care medicine*, 9(8):591–7, aug 1981.

[2] J R Le Gall, P Loirat, A Alperovitch, P Glaser, C Granthil, D Mathieu, P Mercier, R Thomas, and D Villers. A simplified acute physiology score for ICU patients. *Critical care medicine*, 12(11):975–7, nov 1984.

[3] Antonio Paulo Nassar Jr, Amilcar Oshiro Mocelin, André Luiz Baptiston Nunes, Fabio Poianas Giannini, Leonardo Brauer, Fabio Moreira Andrade, and Carlos Augusto Dias. Caution when using prognostic models: a prospective comparison of 3 recent prognostic models. *Journal of critical care*, 27(4):423–e1, 2012.

[4] Daniele Poole, Carlotta Rossi, Nicola Latronico, Giancarlo Rossi, Stefano Finazzi, Guido Bertolini, et al. Comparison between saps ii and saps 3 in predicting hospital mortality in a cohort of 103 italian icus. is new always better? *Intensive care medicine*, 38(8):1280–1288, 2012.

[5] Alistair EW Johnson and Roger G Mark. Real-time mortality prediction in the intensive care unit. In *AMIA Annual Symposium Proceedings*, volume 2017, page 994. American Medical Informatics Association, 2017.

[6] Thomas Desautels, Ritankar Das, Jacob Calvert, Monica Trivedi, Charlotte Summers, David J Wales, and Ari Ercole. Prediction of early unplanned intensive care unit readmission in a uk tertiary care hospital: a cross-sectional machine learning approach. *BMJ open*, 7(9):e017199, 2017.

[7] MIT Critical Data. *Secondary Analysis of Electronic Health Records*. Springer International Publishing, 2016.

[8] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[9] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pages 245–248. IEEE, 2012.

[10] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[11] 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 E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pages 1–8. Citeseer, 2003.

[13] Mansour Keramat and Richard Kielbasa. A study of stratified sampling in variance reduction techniques for parametric yield estimation. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(5):575–583, 1998.

[14] Ajinkya More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, 2016.

[15] Show-Jane Yen and Yue-Shi Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In *Intelligent Control and Automation*, pages 731–740. Springer, 2006.

[16] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[17] Show-Jane Yen and Yue-Shi Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727, 2009.

[18] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[19] Russel Pears, Jacqui Finlay, and Andy M Connor. Synthetic minority over-sampling technique (smote) for predicting software build outcomes. *arXiv preprint arXiv:1407.2330*, 2014.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[21] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[22] Strother H Walker and David B Duncan. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967.

[23] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[24] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[25] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[26] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. 2003.

[27] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. Svm parameter optimization using grid search and genetic algorithm to improve classification performance. *Telkomnika*, 14(4):1502, 2016.

[28] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*, pages 345–359. Springer, 2005.

[29] MJ Goddard and I Hinberg. Receiver operator characteristic (roc) curves and non-normal data: an empirical study. *Statistics in medicine*, 9(3):325–337, 1990.

[30] Laboratory for Computational Physiology-MIT. Schemaspy analysis of mimic.mimiciii, 2017.

APPENDIX A: CONFUSION MATRICES

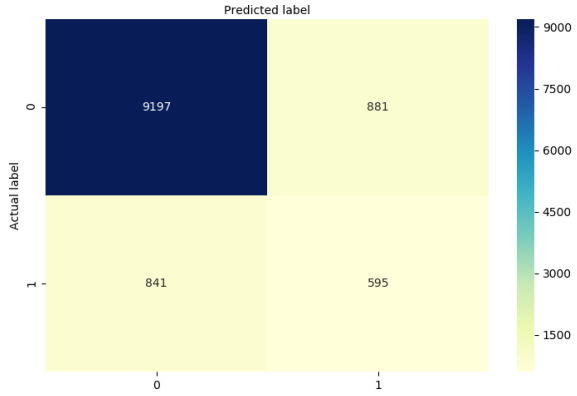


Fig. 11: Confusion matrix for random oversampling

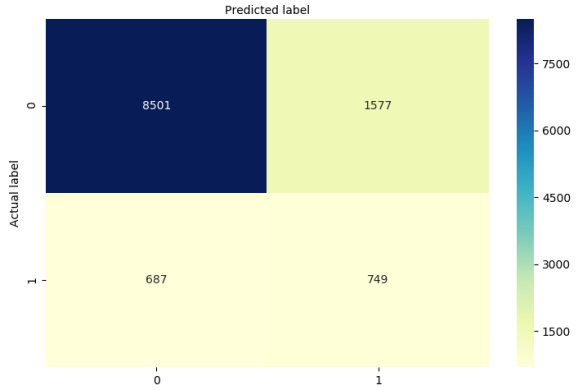


Fig. 12: Confusion matrix for SMOTE

APPENDIX B: MIMIC HIERARCHY

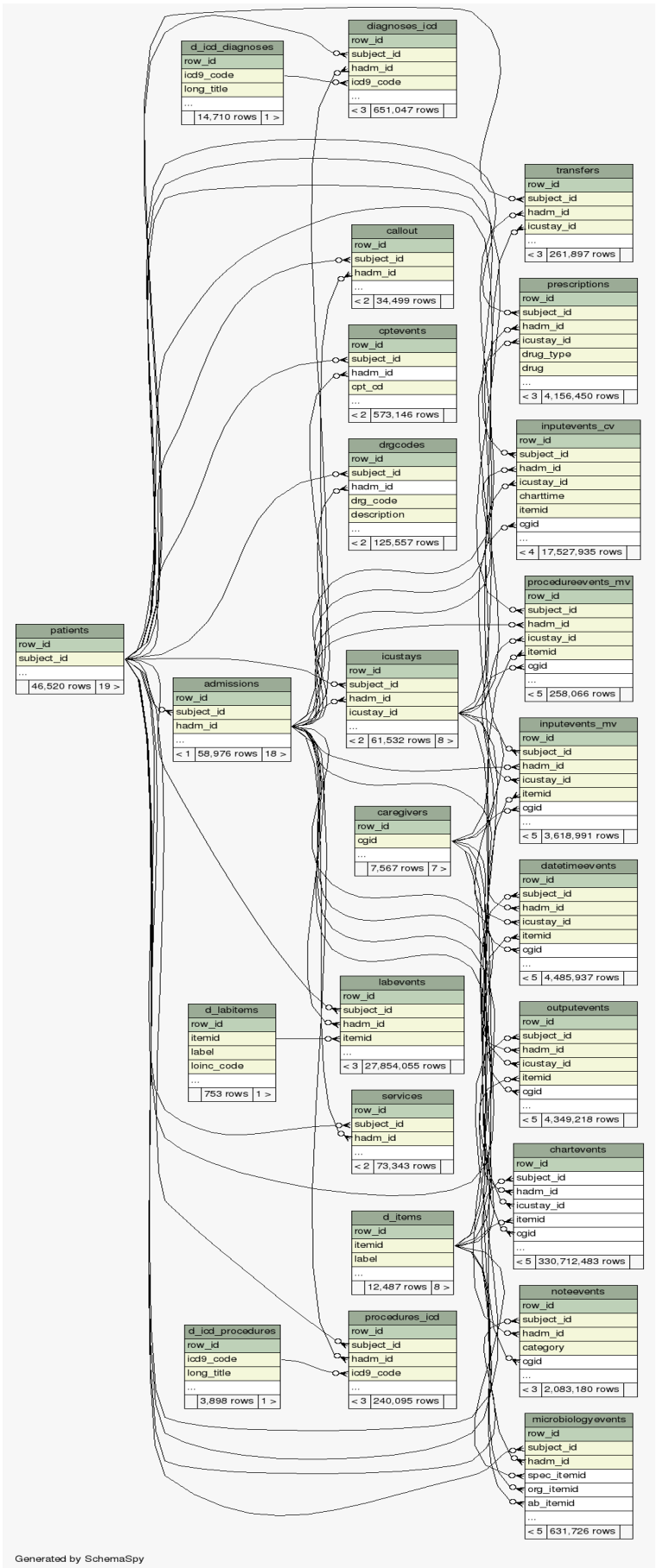


Fig. 13: Hierarchy of the relational databases in the MIMIC-III dataset which each describe various information about a patients stay in the ICU ward. This diagram has been extracted from LCP-MIT constructed using SchemaSpy[30].