Heart Failure Prediction Using the MIMIC III ICU database.

J.D. Schonhoft

I – Project Overview

Goal:

For myself to learn more about heart failure with the MIMIC Intensive Care Unit patient records database. Also this database is stored using SQL, which I have not used before, so it provides a good opportunity to learn.

Background:

Heart failure is a condition where the heart does not pump blood efficiently and can be separated into two broad categories: HF with preserved ejection fraction (HFpEF) where the heart walls thicken and muscles are not able to relax properly and HF with reduced ejection fraction (HFrEF) where the heart walls become weak and the heart is not able to eject blood properly. In many respects it is a 'catch-all' diagnosis, where there are many underlying causes that lead to the condition including high blood pressure, congenital heart diseases, stroke, lung conditions, and hereditary loss of function diseases. Many of these causes have yet to be discovered or fully understood.

The Intensive Care Unit or ICU is where patients requiring constant monitoring and intervention are seen. Nearly 5 % of ICU admissions are linked to HF and it is one of the leading causes of mortality. Further the vast majority of the cost is associated with inpatient care in units such as the ICU and importantly nearly 2% of the total health care cost is associated with HF and ICU admissions. Therefore, predicting outcomes of HF paitents after ICU admission is a very important problem.

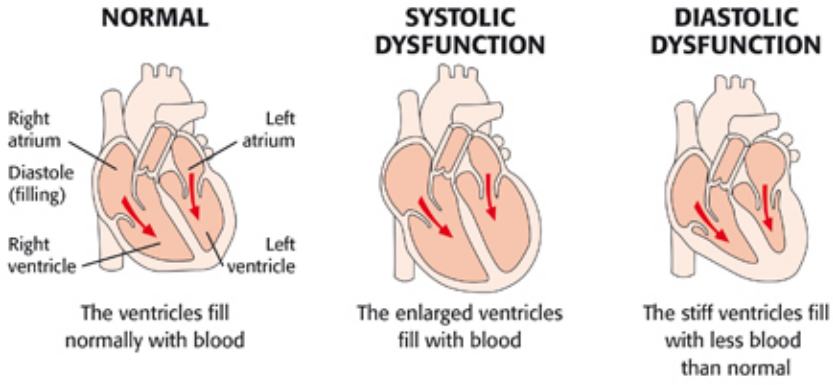


Figure 1: Anatomical illustration of systolic and diastolic Heart Failure (HF). Heart Failure is a condition where the heart cannot pump blood efficiently. Systolic HF or HF with reduced ejection fraction (HFrEF) is where the ventricle walls weaken and the heart cannot pump out a large enough volumne. Diastolic HF or HF with preserved ejection (HFpEF) is where the heart walls become stiff and are not able relax properly. Illustration was taken from: http://www.biomerieux-diagnostics.com/heart-failure

HF is typically diagnosed in the following way. First prior clinical history, symptoms and resting ECG are used and if deemed abnormal plasma Natriuretic Petides are measured (proBNP etc.) and finally echocardiography is used if appropriate.

There have been several models published that utilize machine learning methods to predict outcomes such as mortality or rehospitalization. A quick search in the pubmed database includes about 10-20 references (a dearth given the prevalence of Heart Failure and the associated cost burden). Most of these studies use smaller databases (100-200 patients), therefore using the MIMIC database (10,000 + HF patients records) should provide a good research opportunity. There are a few studies, however, that use larger patient databases that are reasonably successful. I describe one of these below that I will use as my benchmark.

Problem Statement:

I want to see if I can predict mortality after each patients ICU visit for heart failure patients with available ICU records. In other words, whether the patient remains alive or dead after being released from the hospital. The lab records are primarily numeric and contain measurements from different clinical tests. An example of this would be blood oxygen content or blood hemoglobin levels. In total, there are over a 700 different measurements for these patients, so an important aspect is to define important features of the data for which to train ML algorithms on.

Predicting mortality is a classification problem ideal for supervised learning. However, as a side, it would also be interesting to look for different clusters within heart failure patients to understand if particular groups of people are more susceptible or the like and also to pick which tests of the 1000 are most predictive.

Metrics:

This is a binary classification problem so I will use accuracy and the F_1 score which is a balance of precision and recall. Using F_1 is important because the target is not perfectly balanced. Precision, recall and F_1 are defined as the following in equations 1-3:

[eq 1] $$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

[eq 2] $$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

[eq 3] $$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

II – Analysis

Exploratory Analysis:

I am using the MIMIC III ICU database (https://mimic.physionet.org). This database is a database of of over 60000+ hospital records of patients who were admitted to the critical care unit. In total 10414 of these patients have a heart failure diagnosis. Data is stored in a SQL relational database, is about 80GB in size, and contains ~10+ tables for each ICU visit. The patient or subject ID number connects or maps the majority of the information between tables.

Some useful tables are:

1) ADMISSIONS – table of dates, subject ID, and type of ICU admission
2) DIAGNOSES_ICD – mapping between diagnosis and subject ID
3) D_ICD_DIAGNOSES – list of all possible diagnostic codes and their titles
4) LABEVENTS – List of all clinical tests, their numeric value and subject ID
5) D_LABEVENTS – List of all lab tests and their titles and mapping to their codes
6) PATIENTS – gender and other patient information

The 'diagnoses_icd9' table maps patient ids to their diagnosis (icd9 code). For Heart Failure and the different sub-categories, the code begins with 428. After retrieving a list of patient id's who have a diagnosis of heart failure using this feature, I then extracted other information and mapped this information to the patient id (lab tests, etc.).

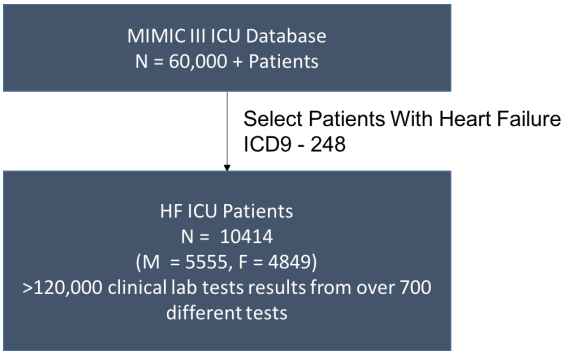


Figure 2: Selection of HF Patient Data from MIMIC III

Below is an example of the pandas dataframe created from an SQL query containing information such as HF patient gender and whether or not they have died based on social security data.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import psycopg2
%matplotlib inline

# Connect to local copy of the MIMIC III database
sqluser = 'postgres'
dbname = 'mimic'
schema_name = 'mimiciii'

# Connect to local postgres version of mimic
con = psycopg2.connect(dbname=dbname, user=sqluser)
cur = con.cursor()
cur.execute('SET search_path to ' + schema_name)

# selecting all patient ids with heart failure diagnosis using ICD9code. There are about 10,000 in total.
data = pd.read_sql_query("SELECT * FROM diagnoses_icd WHERE icd9_code LIKE '%428%'",con)
HF_subject_id = data.subject_id.unique()

#Next creating dataframe of patient information with id
patient_list = ', '.join([str(i) for i in HF_subject_id])
query= 'SELECT * FROM patients WHERE subject_id IN (%s)' % patient_list

# creating pandas data frame with patient demographics. Note that dates are shifted to that patients cannot be identif
HF_patients = pd.read_sql_query(query, con)
HF_patients.head()
```

Out[1]:

	row_id	subject_id	gender	dob	dod	dod_hosp	dod_ssn	expire_flag
0	2	3	M	2025-04-11	2102-06-14	NaT	2102-06-14	1
1	8	9	M	2108-01-26	2149-11-14	2149-11-14	2149-11-14	1
2	18	21	M	2047-04-04	2135-02-08	2135-02-08	2135-02-08	1
3	23	26	M	2054-05-04	2128-02-25	NaT	2128-02-25	1
4	26	30	M	1872-10-14 00:00:00	NaT	NaT	NaT	0

Exploratory Visualization:

The first question to ask is how many ICU admissions each of these patients had. Most (~7000) had only one ICU admission, whereas the other 3000 were readmitted at least one time. Of the 10414 HF patients, 6115 are no longer living.

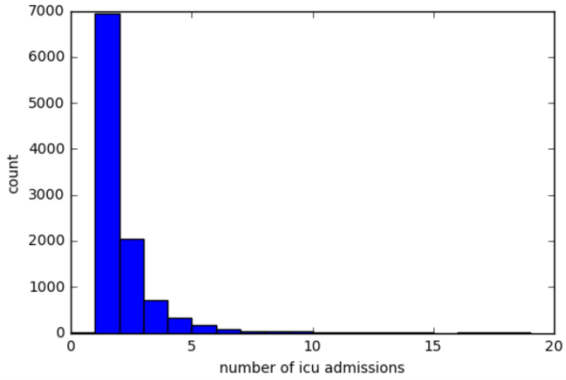


Figure 3: Histogram of number of ICU admissions for HF patients.

Next, do ICU readmissions correlate with mortality? The answer is yes to some degree. Focusing on the data points where there are significant number of patients (ICU admissions 1-5) there is a clear uptick in mortality rate. This makes sense of course, because the sicker a person is the more likely they are to die and to be readmitted to the ICU.

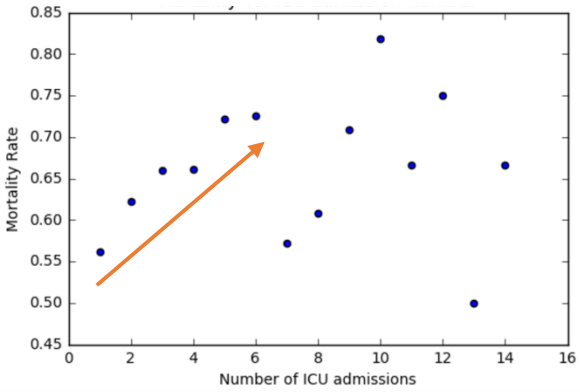


Figure 4: Mortality Rate vs. ICU admission number.

Next, what is the typical amount of time each patient spends in the ICU?

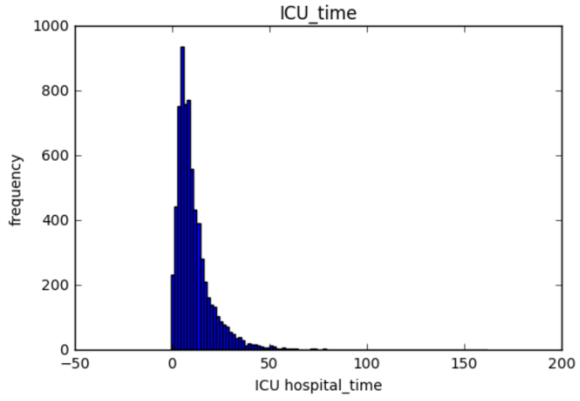


Figure 5: Histogram of days spent in the ICU for HF patients

There are many questions that can be asked with this data, however my focus here is to use clinical lab tests to predict mortality. To retrieve this data there are two tables of interest. The first table (D_LABITEMS) has a list of all possible lab tests. The second table(LABEVENTS) has a list of the numeric values, date, and patient ID for every available clinical measurement. In the HF patient subset, there are about 120,000 available measurements.

Here a are a few examples of the types clinical lab measurements

Out[6]:

	row_id	itemid	label	fluid	category	loinc_code
0	1	50800	SPECIMEN TYPE	BLOOD	BLOOD GAS	None
1	2	50801	Alveolar-arterial Gradient	Blood	Blood Gas	19991-9
2	3	50802	Base Excess	Blood	Blood Gas	11555-0
3	4	50803	Calculated Bicarbonate, Whole Blood	Blood	Blood Gas	1959-6
4	5	50804	Calculated Total CO2	Blood	Blood Gas	34728-6
5	6	50805	Carboxyhemoglobin	Blood	Blood Gas	20563-3
6	7	50806	Chloride, Whole Blood	Blood	Blood Gas	2069-3

Next, I picked one lab test and retrieved all measurements for the HF patients. I picked the haptoglobin test (**Fig. 6**), which is a blood protein associated with inflammation just to see if I could retrieve the data. There seems to be a slight difference in distribution between patients who are alive and dead. In the Ipython notebook for data cleaning, I ran this code on all available lab tests and produced histograms for each test so that I could scroll through visually.

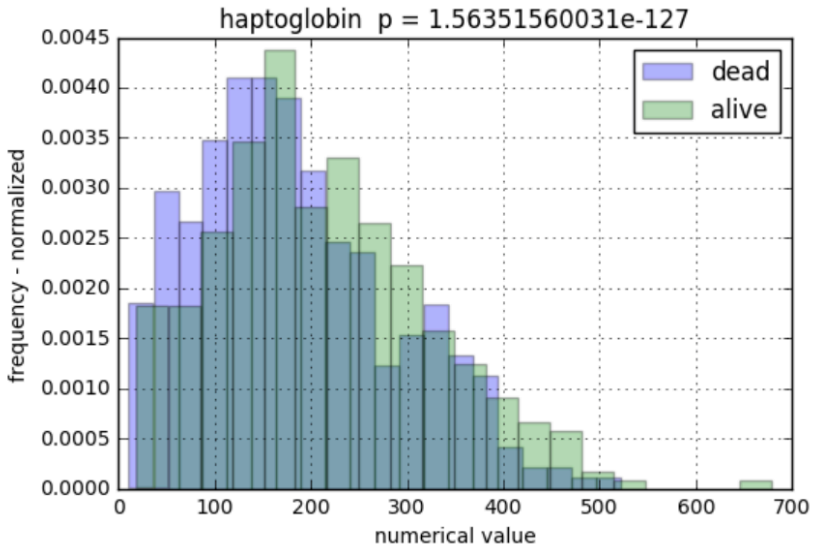


Figure 6: Normalized distribution of haptoglobin concentrations of HF patients. The p-value was calculated using a Mann-Whitney non-parametric t-test, illustrating that the means between dead and alive patients are significantly different.

Algorithms and Techniques:

The idea is to predict mortality based on simple numeric lab test values. Therefore, supervised learning methods are ideal for this. I chose to use a random forest first because I can easily understand the results and pin down the important features. I also tested the utility of other supervised learning algorithms such as a SVM, KNN, a multilayer perceptron, AdaBoost and Naïve Bayes to compare performance.

A ***very*** brief summary of each of these algorithms that I will include in my ML pipeline is as follows:

1) **Random Forest:** An ensemble method (ensemble = combination of other ML algorithms) that splits the data interatively using decision tree like boundaries. With each split the algorithm minimizes the information entropy between the new groups.
2) **Naïve Bayes:** Applies bayes theorem to the multidimensional dataset to come up with probabilities for each class.
3) **K Nearest Neighbor:** Finds clusters in the data by dropping a random point or points and then minimizing the distance to the nearby data points. Typically euclidean distance is calculated at each iteration, the point is moved and eventually the distance is minimized.
4) **AdaBoost:** Takes many weak classifiers and over time iteratively strengthens them to make one good classifier.

5) **Multilayer Perceptron:** A network of linear classifiers consisting of weights and biases. The weights and biases get strengthened if the classifier is good at separating the data.
6) **Support Vector Machine:** Draws hyperplanes with soft margins separating the data in high dimensional space. Can be made non-linear by using the kernel trick.

Benchmark:

There is an excellent review (*1*) that summarizes in high detail all of the work that has been done on this and similar problems in heart failure outcomes prediction. I choose one study to compare my work to. In Taslimitehrani *et al.* (*2*) they used an SVM with input data from comorbidities, drugs taken, demographics and a few lab tests. With an SVM they achieved a 75% accuracy and with inclusion of other non-numeric feature by logistic regression were able to achieve accuracies above 80%. I will use this study to compare my results as I feel it is the most complete and has the most data points.

Table 5
Precision, recall, and accuracy of different classifiers.

Measure	Model	SVM	Log Reg.	CPXR(Log)
Precision	1 year	0.66	0.752	0.82
	2 years	0.42	0.703	0.78
	5 years	0.2	0.513	0.721
Recall	1 year	0.7	0.66	0.782
	2 years	0.68	0.643	0.76
	5 years	0.5	0.506	0.615
Accuracy	1 year	0.75	0.89	0.914
	2 years	0.55	0.758	0.83
	5 years	0.66	0.717	0.809

Figure 7. Results from Taslimitehrani *et al.* (*2*) whose SVM and logistic regression models I will use as a benchmark.

III – Methodology

Data Preprocessing:

Using an SQL query I extracted all available lab tests for the HF patient cohort. I simplified the dataset by only using patients that had one hospital ICU admission. This brought down the number of HF patients to 6954. Also many tests had multiple values for each test as it was measured many times during the course of the ICU admission. Therefore, I took the mean of all of these values in each individual test category for each individual patient. This resulted in a dataframe of 345 total clinical tests with 6954 patients. Note that many of these tests are non-numeric or had only been given to a handful of patients. This is why there are only 345 from the possible 700.

Next I plotted histograms of each test to compare the differences between living and patinets who had died. I stored the number of patients that had a particular test done and then used a t-test (Mann-Whitney) to sort tests where the mean was significantly different. I realize that a t-

test or the like is unlikely to sort the importance of each test, but this type of thing is typical in medicine and thought it would be a good comparison. Not surprisingly many of these tests have very low p-values due to the large number of data points.

Out[93]:

	itemid	label	N	P-value
37	50852	% Hemoglobin A1c	1927	2.893341e-242
83	50924	Ferritin	1904	3.854427e-126
369	51493	RBC	1759	9.394976e-113
377	51516	WBC	1745	3.614403e-99
69	50904	Cholesterol, HDL	1975	1.294155e-93
359	51476	Epithelial Cells	1772	1.557898e-92
97	50952	Iron	2026	6.521778e-90
133	50998	Transferrin	1958	4.794797e-88
98	50953	Iron Binding Capacity, Total	1961	3.320995e-87
68	50903	Cholesterol Ratio (Total/HDL)	1968	1.267591e-80
134	51000	Triglycerides	2319	2.294252e-68
73	50908	CK-MB Index	1755	3.056776e-59

Figure 8. Lab tests sorted by P-value between alive and dead patients with only a single ICU admission

Then to prepare data for ML algorithms, I removed the following:

- Tests where there were less than 3000 patients who had that test
- Tests with non-numeric values

Finally I performed the following manipulations:

- I imputed missing data by replacing all missing values with the **mean** of particular test
- I normalized all data between 0 and 1.

Implementation of Supervised Machine Learning Algorithms:

I first implemented a Random Forest (RF) algorithm in sklearn. I chose a RF because I have a large number of features most of which are probably useless and I felt a RF would allow me to understand the results and the contribution of each feature to the prediction. With other ML methods, this is not so straightforward or fast on my laptop computer.

In implementing the initial model I first split the data using sklearn (66% for training and 33% for testing). In the first iteration of the RF, I found that at best I could predict correctly 60-70% of the time. Using the output from the decision tree. I picked the most important features from the lab test data.

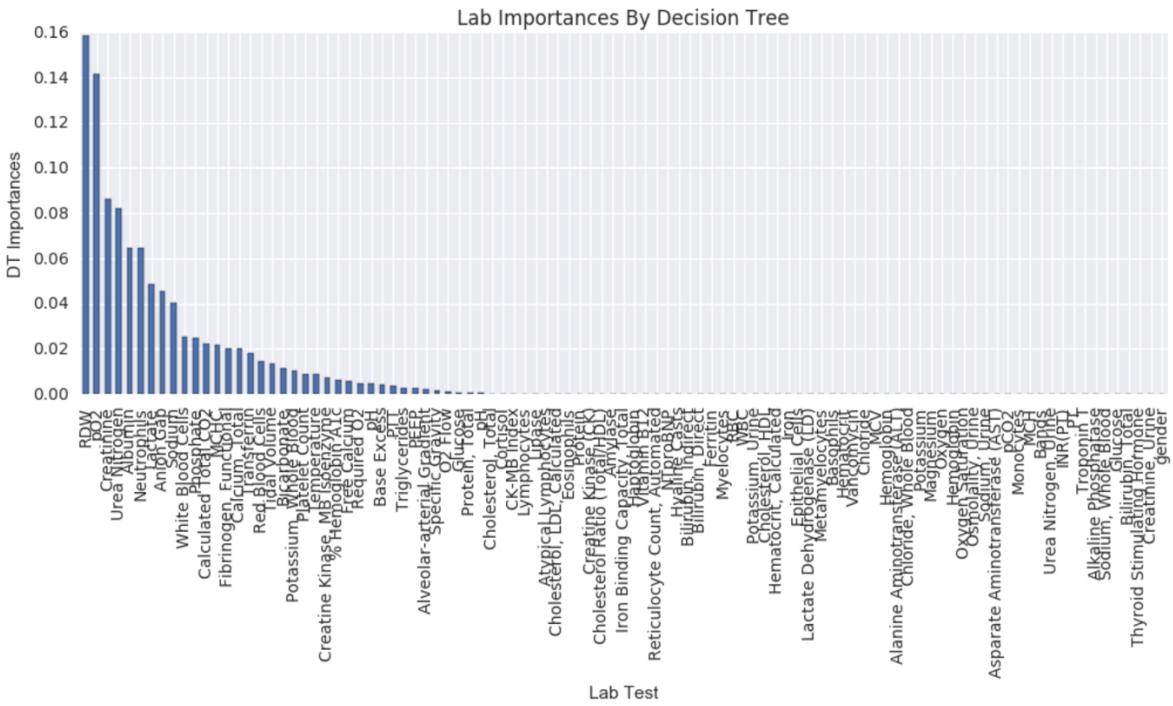


Figure 9: Contribution of the top most important features to the Random Forest model prediction.

Using the top 10 features, I tried several other ML supervised learning algorithms with out of the box parameters. Notably despite fine tuning some of the parameters (for RF and SVM) they all performed very similarly. For example, the SVM and RF both performed equally well with an accuracy of just over 70%. Here are all of the ones that I tried and the random forest actually seems to work the best. The parameters were the defaults in this case.

	Random Forest	SVM	Naïve Bayes	AdaBoost	KNN	NN Perceptron
Precision	0.69	0.77	0.59	0.80	0.69	0.70
Recall	0.74	0.70	0.75	0.69	0.67	0.75
F1-score	0.72	0.74	0.66	0.74	0.68	0.72
Accuracy	0.69	0.69	0.66	0.68	0.63	0.70

Table 1: Comparison of ML algorithms for predicting ICU mortality. Default parameters were used as defined in the sklearn documentation.

IV Results

Refinement of the Random Forest Model.

In my final model I used sklearn's Gridsearch function to explore parameter space and to pick optimal values for the Random Forest classifier. I varied the max_depth (number of nodes in the tree), max_features (number of features to use in the model) and n_estimators (the number of decision trees). With this I was able to achieve a modest improvement in the metrics chosen (F1 = 0.72 – 0.76). The reason why optimization has little performance benefit is explained in the conclusion section below.

Random Forest Model		
Metrics	**Before Optimization**	**After Optimization**
Precision	0.69	0.82
Recall	0.74	0.71
F1-score	0.72	0.76
Accuracy	0.69	0.71
Parameters	**Before Optimization**	**After Optimization**
max_depth	None	3
n_estimators	10	10
max_features	None	10

Model Evaluation and Validation

To control for overfitting, I compared the test set accuracy to the training set accuracy as a function of DT depth. With this experiment I concluded that a depth of 3 or 4 and that using a maximum of 10-20 features was optimal (also see above).

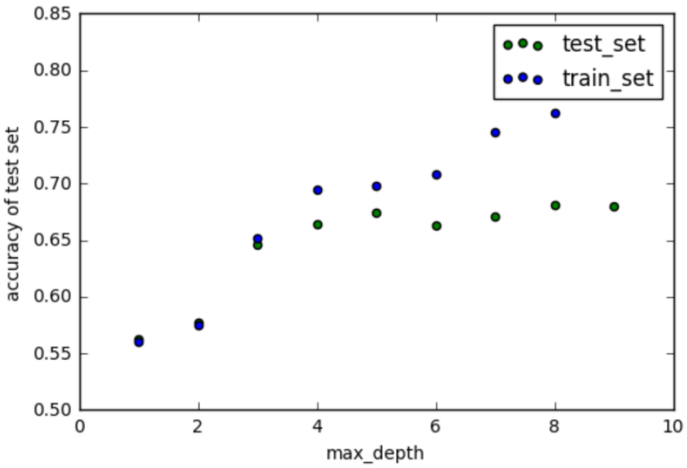


Figure 10: Decision tree accuracy in prediction of mortality in the MIMIC III database.

Finally, I performed cross-validation where I retrained the model on different portions of the data and very little difference was observed. Again the best I could do was a minor incremental improvement to achieve accuracies of just over 70%.

Clustering identifies HF and kidney failure and/or sepsis as highly predictive of mortality.

Heart Failure is a broad class of heart disease and contains many underlying causes that are exacerbated by comorbities. Therefore, I next sought to cluster the features that I had deemed important using the above decision tree using both a PCA and t-SNE. The main reason for using both of these is that I wanted to compare for my own sake in learning. The goal of this is to find clusters of HF patients that are easier or harder to predict mortality. These types of natural clusters are also indicative of the actual mechanism behind heart failure. For example, heart failure with diabetes mellitus versus heart failure with sepsis likely have different outcomes on average. A major question I want to answer is whether or not these types of clusters can be seen when analyzing only the lab records. This is important because these measurements are very simple and almost always available, which may not be the case for more sensitive (and expensive) tests like an ECG.

First I used a PCA to reduce data from the top 10 features to 2 dimensions. When plotting these two dimensions, the dead vs. alive, the patients separate out to some degree. I'll come back to this below.

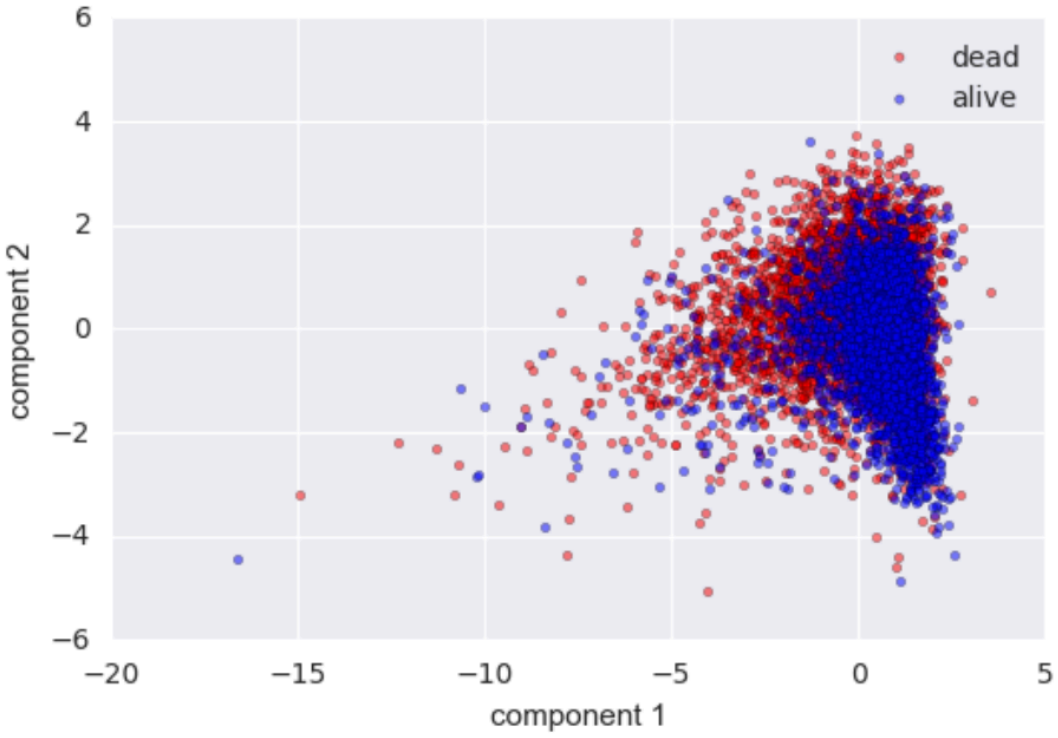


Figure 12: PCA dimensionality reduction of top lab tests identified in the decision tree model.

Secondly, I used t-SNE to reduce the dimensionality of the data. A similar trend was observed where some of the data partially separated while other parts of the data remained overlapping. Overall I found the separation to be better by t-SNE so I went forward with this. Importantly, this trend was independent of the parameters of t-SNE (perplexity, epsilon).

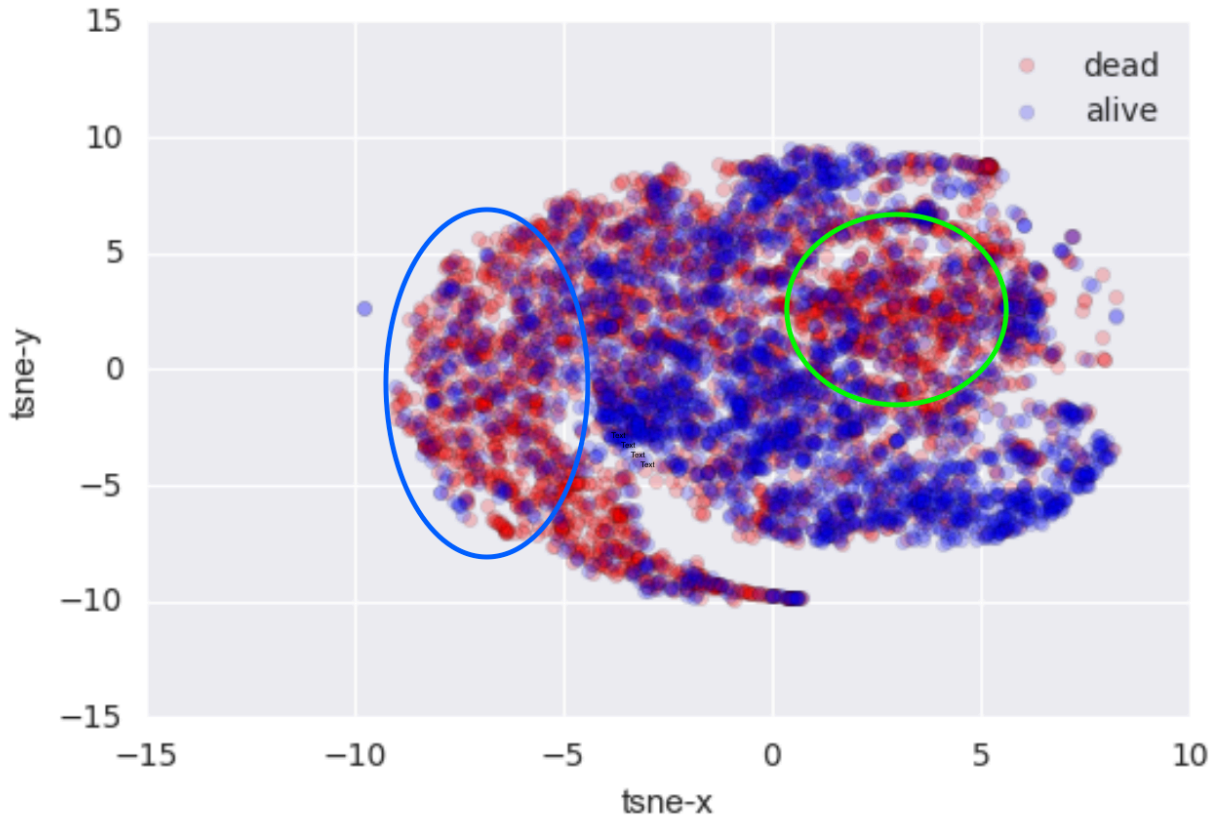


Figure 13: t-SNE dimensionality reduction of HF patient lab clinical tests. Visually, some sub clusters are observed.

I noticed that in the t-SNE plot, in the upper right quadrant there is a higher density of dead patients versus alive and similarly in the bottom/middle left. I isolated these patients within the indicated circles above and plotted the top associated comorbidities. First, however, it is useful to look at comorbidities or co-diagnoses for the entire HF patient set (CHF NOS is Heart Failure or ICD9 code 248).

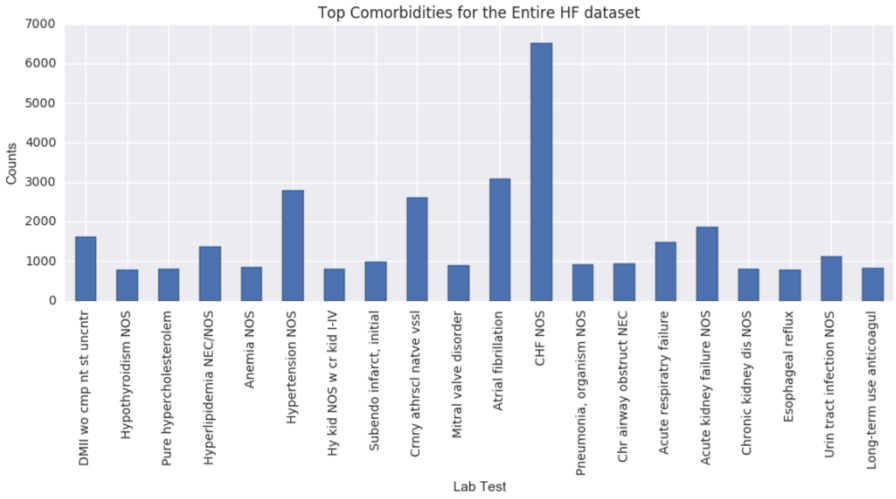


Figure 14: Comorbidities in the HF patient dataset

First in sum for all HF patients, Diabetes Mellitus and kidney dysfunction is one of the top comorbidities. Now in just subsetting the left red cluster in the t-SNE plot (blue circle), there is now a greater proportion of patients with kidney related disorders (kidney failure, chronic kidney disease). While the difference is subtle, this fits with the predictive lab tests. The most predictive measures are all associated with renal dysfunction (Albumin, Urea nitrogen etc.). In the right cluster, sepsis is over represented. Therefore, it seems that I can separate my data into 3 broad groups: HF with renal dysfunction, HF with sepsis, and HF with other comorbidities. The latter two are easiest to predict mortality. Interestingly, while most but not all HF patients in these clusters have kidney problems or sepsis, it may suggest that several of these patients have important undiagnosed conditions.

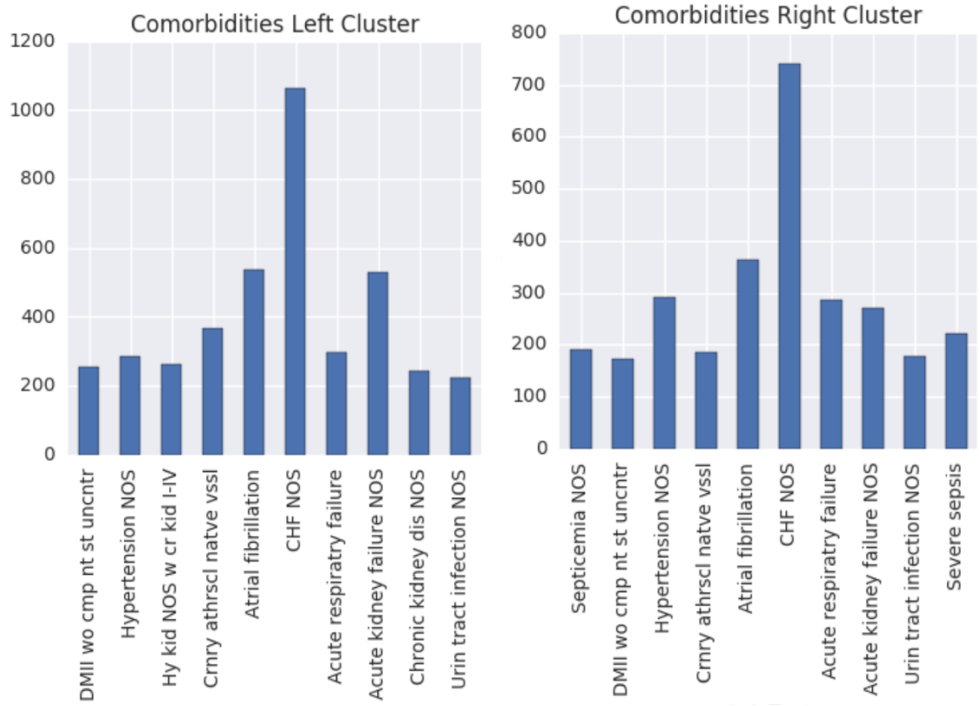


Figure 15: Top comorbidities associated with left (blue circle, Fig. 13) and right (green circle, Fig. 13) t-SNE cluster.

Justification

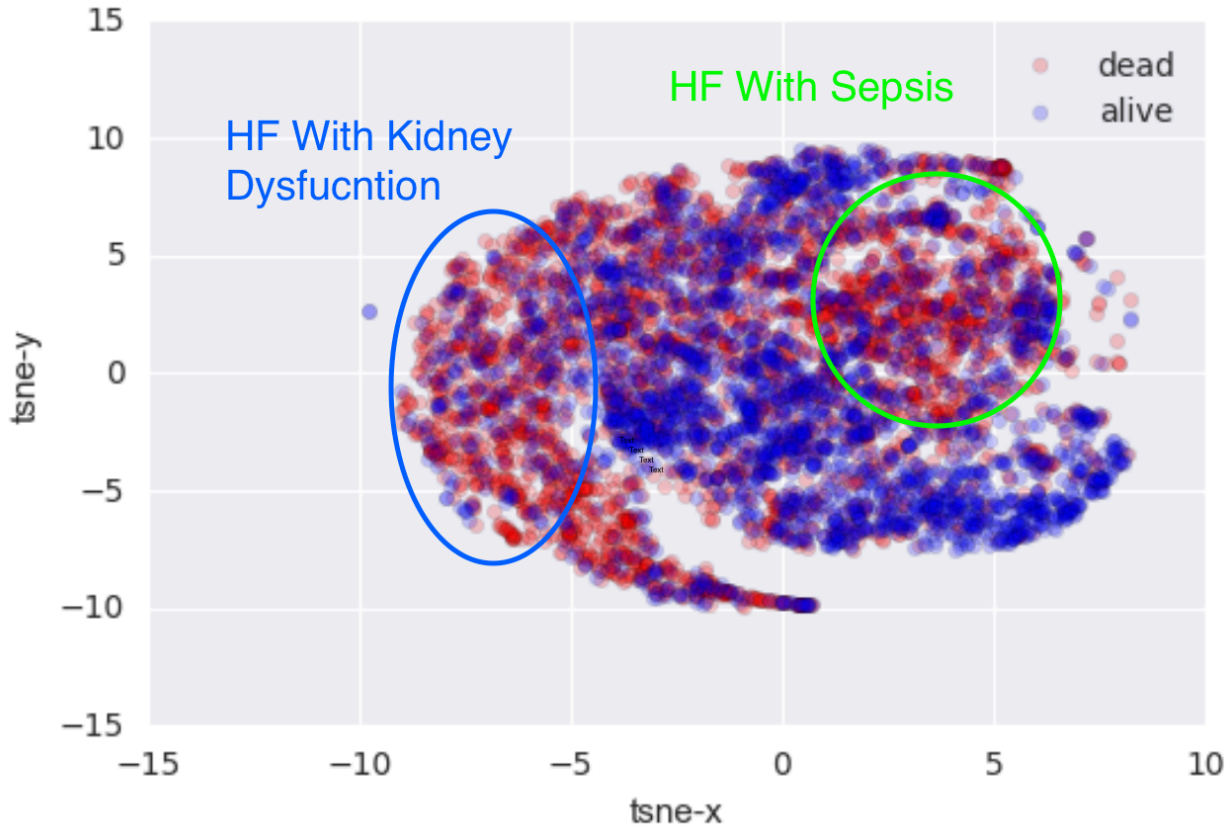
From this analysis, I realized that with just a few lab tests my model performs about as well as more complex published models that incorporate other factors such as demographics, age *etc*. and many other features. Compared to the benchmark model described above in **Fig. 7**, my random forest models with just 5-10 lab test features averaged over the course of the persons ICU stay can predict with 70% accuracy whether that person will die after being released from the hospital. This is compared with ~70-90% accuracy for published models.

Upon analysis of clusters that emerge from the lab tests features, renal failure is a major predictor of mortality. This makes sense, because kidney failure correlates with changes in the most predictive features in my random forest model. These features are Urea nitrogen, Albumin concentration, bicarbonate, anion gap – to name a few. The other ~30% of patients are harder to predict and these patients have a lower prevalence of renal dysfunction and sepsis. In the benchmark models it appears that using comorbidities, drugs taken, demographics etc. can provide the extra bump needed to achieve more predictive power (80-90% range).

V – Conclusion

Free form visualization

I chose the t-SNE representation of the data as it is straightforward to understand by eye and generally represents how well my analysis and other models have done. From this I classify HF patients into two broad categories, a subset that are easy to predict (e.g. with kidney dysfunction or other severe comorbidities like sepsis) and another subset where prediction is very challenging from lab tests alone (e.g. the overlapping data points).



Summary of the Final Model:

For this problem, I found the Random Forest supervised learning classifier to be best. This algorithm had the advantage of being fast and also being able to determine the features that are important for prediction. This latter aspect is critical, because there are a large number of features that are useless. Finally, after making the classifier with just a few (~10) lab tests I was able to predict with about 70-75% accuracy whether patients died or remained alive. Next after picking the best features, I used unsupervised learning to find clusters of HF patients that had unique comorbidities. For this problem t-SNE turned out to be the most useful and identified HF with Sepsis and HF with kidney dysfunction as the most predictive of mortality.

Reflection

This was a good exercise for me and as a result I learned a lot about Heart Failure, how to retrieve data using SQL, and in applying supervised and unsupervised learning algorithms. Overall, I realized one very important thing, which is that prediction is only as good as the underlying data. Seems obvious, yet despite more complex models existing in large part only a few features (Albumin, Urea Nitrogen) are needed to make a solid prediction compared to the benchmark – the other parameters simply fine tune the model to only a modest degree.

Another aspect that I realized was that medical record data is very irregular and sparse. Meaning that the time between lab tests varies, very few tests are given to all patients in the records and most of the common tests used, only generally measure the health of the individual. Such as blood oxygen content. Very few of these tests by themselves are truly predictive and are based on 50+ year old research.

Improvement:

There are many improvements that I could implement in the model but would be beyond the scope and purpose of this project. I will focus mostly on the use of the lab tests, because this is the most unbiased measure. One improvement would be to experiment with trends in the lab data. Many of these measurements were taken over time during the ICU visit and I simply used the mean (also I tried the median – made little difference). It's possible that changes in the lab test data over time would be useful in prediction e.g. from a regression or the like. Another aspect would be to focus on particular groups of patients. For example, systolic versus diastolic dysfunction. This might be possible, but would require extracting meaning from the patient clinical notes tables, which is a significantly harder problem and possibly a project in it's own right. I was disappointed in the fact that in this data set many of the sub ICD9 codes are not present. This often is the case that a diagnosis of heart failure is made without any further classification in the records. I could imagine the use of bag of words or similar algorithms to extract this type of meaning and possibly a network model.

Finally, I think this project made me realize many of the limitations of using medical records data and that is that most of the tests are based on knowledge from 50 years ago and are applied almost at random. E.g. measuring hemoglobin levels or p02 – which isn't to say that these aspects are not important. However, very rarely are more modern methods like proteomics, genomics included in these records in a standardized way. I imagine the future will be much different. In my ideal world a simple blood test at any point in time would simultaneously measure changes in tens of thousands of metabolites, proteins or the like. This technology exists but has yet to be refined enough to be in the main stream. I could imagine that coupling this type of measure to hospital records over time would be transformative and would allow a higher resolution clustering and analysis to be made. We are not there yet, however, I believe we are at the beginning of this phase in medicine and considering that only 10 years ago, these types of databases did not exist, things are moving fast.

References

1. E. E. Tripoliti, T. G. Papadopoulos, G. S. Karanasiou, K. K. Naka, D. I. Fotiadis, Heart Failure: Diagnosis, Severity Estimation and Prediction of Adverse Events Through Machine Learning Techniques. *Comput Struct Biotechnol J* **15**, 26-47 (2017).
2. V. Taslimitehrani, G. Dong, N. L. Pereira, M. Panahiazar, J. Pathak, Developing EHR-driven heart failure risk prediction models using CPXR(Log) with the probabilistic loss function. *J Biomed Inform* **60**, 260-269 (2016).