Predicting Death or Clinical Deterioration from the Emergency Department with Supervised and Unsupervised Learning

Authors: Zach Sletten, Allen Chezick, Hunter Belous
SIADS 696 - Milestone II

Introduction

The emergency department is a cognitively challenging work place. There are often distractions and interruptions in the midst of critical decisions. Clinicians often make decisions with limited information and with limited time to spend with the patients or review data. Having an early warning system that would aid clinicians in identifying patients at risk for death or deterioration could prove helpful. We aim to develop such an aid. We hope that developing a machine learning model that recognizes clues within the electronic medical record that predict death or deterioration could save lives, reduce errors and offload cognitive burden for clinicians. This team has a background in science and healthcare related work. Additionally, one of the team members is a physician in the emergency department.

We used the multimodal ED benchmark (MDS-ED) data set which was derived from the MIMIC-IV dataset. The data set is very large, containing 129,095 rows, 2,408 columns and represents 71,098 patients and 121,195 ED visits. It focuses on data obtained in the first 1.5 hours of the ED visit. It captures data such as biometrics, demographics, vital signs, laboratory values, demographic information, diagnosis, along with outcomes such as clinical deterioration and death. The size of the dataset creates an opportunity to explore the potential patterns that might exist so that key clinical outcomes may be predicted. This project is designed to incorporate supervised and unsupervised learning techniques to be able to extract structures and predictive values so that such predictions can be made given the large scale dataset.

Utilizing the cleaned and updated dataset, the aim for the unsupervised portion of the was to identify naturally occurring subgroups of patients with shared characteristics using KMeans. Based on the large number of features the second goal was to explore using dimensionality reduction using principle component analysis (PCA). We observed no improvements in prediction modeling when using principal components.

Similar to the unsupervised component, the supervised machine learning component of this project utilized a cleaned data set with new engineered features and targets. Various models and targets were explored using pycaret initially which provided excellent insights into how predictive our models could be along with how well models generally performed on various targets based on a variety of metrics. Ultimately, we selected logistic regression, random forest classifier and SVM to further evaluate based on preliminary results and based on the desire to have a variety of distinctly different methods for further tuning. We focused on predicting early death (<28 days) or need for organ support (deterioration). We chose AUPRC as our target metric as we wanted a metric that balanced precision and recall. We found our random forest regressor did best, with significant lift provided. Our project was novel in its narrow focus on demographics, vitals, biometrics, vitals and labs in predicting short term outcomes.

Related Work

Upon initial investigation, a few examples of related work helped identify existing gaps that we wish to cover with this project. A few examples include but are not limited to:

Establishment of ICU Mortality Risk Prediction Models with Machine Learning Algorithm Using MIMIC-IV Database (https://pmc.ncbi.nlm.nih.gov/articles/PMC9139972/):

This study applied machine learning models (XGBoost, logistic regression, SVM, decision tree) on ICU patient data drawn from MIMIC-IV, just like this project. The goal of the project was to predict in-hospital mortality using subscores of APACHE III and LODS as inputs. Our study improves on this, by working with a broader cohort and utilizing features that utilize deltas and integrate both supervised prediction and unsupervised clustering rather than just one outcome modeling. Additionally our study focuses on information available in the ED and combines deterioration into the outcome.

Machine learning models to predict 30-day mortality for critical patients with myocardial infarction: a retrospective analysis from MIMIC-IV database (https://pubmed.ncbi.nlm.nih.gov/39371393/):

This study extracted myocardial infarction patients from the MIMIC-IV dataset and applied ML principles (XGBoost, Random Forest, etc) to predict 30 day mortality outcomes. Our study improves on this, by working with a broader cohort and is not merely a predictive study. We implement both unsupervised and supervised principals to better understand the infrastructure behind the MIMIC-IV dataset and draw predictions on mortality/deterioration outcomes through the use of the supervised component.

Unsupervised Machine Learning with Cluster Analysis in Patients Discharged after an Acute Coronary Syndrome: Insights from a 23,270-Patient Study (https://pubmed.ncbi.nlm.nih.gov/36870114/):

This study used clustering on patient features post-acute coronary syndrome to identify subgroups with different risk profiles. Our study expands beyond a single disease cohort and works with a more general patient population. The clustering done in our study is also paired with a supervised component.

Data Sources

We used the multimodal ED benchmark (MDS-ED) data set which was derived from the MIMIC-IV dataset (https://physionet.org/content/multimodal-emergency-benchmark/1.0.0/). The data set is very large, containing 129,095 rows, 2,408 columns and represents 71,098 patients and 121,195 ED visits. It focuses on data obtained in the first 1.5 hours of the ED visit. The data was collected from 2008 to 2019 for patients seen at Beth Israel Deaconess Medical Center (BIDMC) in Boston, MA. It's a mixed data set with tabular data, categorical/numeric data, waveform data, diagnosis, ecg waveform data. Many one hot encode features. Some of the important data included 7 columns with demographics, 3 columns with biometrics (weight/height/bmi), 55 columns with vitals obtained in the first 1.5 hours (heart rate, blood pressures), laboratory values (405 columns), 1428 diagnostic prediction targets, 15 death/deterioration prediction targets.

Feature Engineering

This file was very large and cumbersome, and we were unable to open the entire file in the collaborative vocareum environment as the kernel would crash. We had to first load the first 10 rows and identify columns we wanted to remove. We the below changes to prepare the file

a) Non-informative identifiers such as subject_id, stay_id, row_id, general_file_name, and diagnoses fields were removed to shift focus to more relevant data and remove chances for leakage
b) Key continuous fields including temperatures, weights, time-to-event metrics, etc were rounded to two decimal places for uniformity.
c) There were several redundant or non contributory lab features. Only lab measurements ending in _first or _change were retained to reduce redundancy while preserving temporal information. As the majority of patients only had one lab draw and lab value, values such as std, mean etc were non helpful.
d) Sentinel values (-999) were replaced with NaN. Columns were dropped if more than 90% of the values were missing. Rows were dropped if over 60% of their entries were missing.
e) A new binary feature, admit, was created from general_ed_hadm_id, representing whether the patient was admitted to the hospital (1) or discharged (0).
f) Several clinically meaningful target variables were derived from general_mortality_days being:
    i) mortality_any, mortality_28d, mortality_365d, and mortality_gt365d (short-, medium-, and long-term mortality).
    ii) Mortality_category (a categorical determination based on the above criteria ie short term death, medium term etc)
    iii) clinical_deterioration_any (a binary indicator if any deterioration metric was flagged)

iv) Death_or_deterioration_any (a composite outcome combining mortality and deterioration.
v) Organ_support (a composite of need for intubation, pressors, ionotropes, cardiac arrest)
vi) Support_mortality_combo (permutations of need for organ support and the mortality categories resulting in 8 distinct categories)
vii) Shortterm_death_or_deterioration (composite of patients who died in 28 days or less or needed organ support)
viii) Using general_90min, two other new features were created:
ix) month_time (representing the month of presentation)
x) tod_time (hour of the day)

g) Redundant metadata columns were dropped and the final dataset was saved as df_best.pkl. This pkl file includes the completed feature-engineeried dataset that was fed into the supervised and unsupervised components.

h) The final dataset (df_best.pkl) contained approximately 160 columns across seven categories: demographics (7), biometrics (3), vitals (~30), lab measurements (~60), temporal encounter features (2), binary admission and outcome targets (10), and supporting metadata for model separation. Only lab and vital variables ending in _first or _change were retained. Reference the Appendix for a full feature list.

For visual clarity, Figure 1 depicts a high level process flow diagram laying these steps out accordingly:

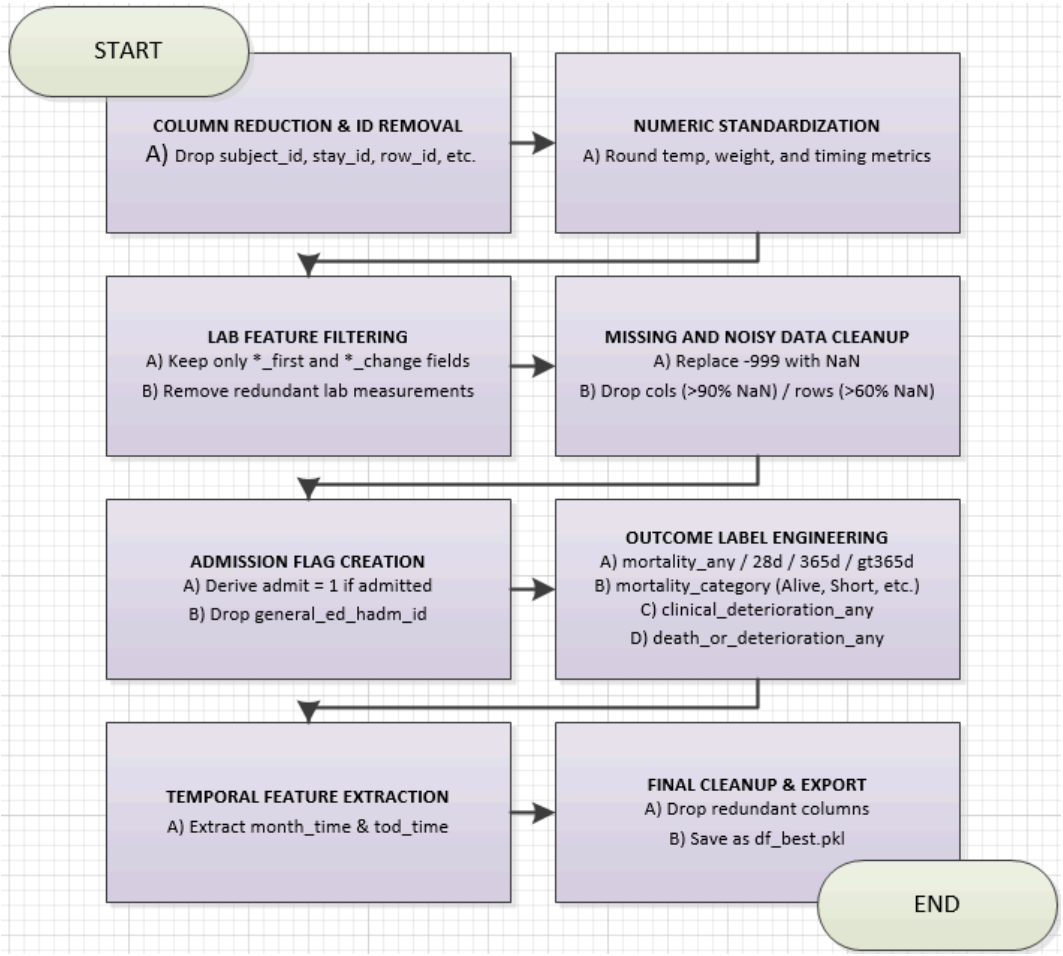


Figure 1

Part A: Supervised Learning

METHODS DESCRIPTION:

We fed the cleaned and engineered version of the df which had been stored as a pkl file into the supervised machine learning pipeline. This was the same pkl file that was used in the unsupervised learning notebook. We applied a support function titled "make_model_ready" that removed target features which would not be used and kept only the target feature of interest to avoid leakage. This function provided the ability to quickly evaluate different targets of interest. Additionally we used the results of our unsupervised learning model to provide a reduced feature space for comparison.

We initially began exploring the relationship between different machine learning models and different target features using pycaret. Pycaret allowed us to rapidly compare multiple targets, methods and metrics simultaneously. Generally what we were seeing was that tree based models were performing better on the majority of metrics. We also discovered that pycaret does not offer the ability to classify multi class features for logistic regression type models. Generally we noticed strong performance from many models on a broad combination of metrics.

We ultimately decided to focus on developing an early warning system that would predict short term mortality (<28 days) or need for organ support (ex pressors, intubation, ionotropes, ecmo, cardiac arrest). We felt this would offer the most clinically useful information to an emergency physician who has limited data and is making decisions that focus mostly on short term death and deterioration. Although it would be interesting to have a model to target features such as mortality category (ex alive, short term death, medium term death, long term death) or a combination of deterioration and mortality categories, we decided targeting short term deterioration or death was the most useful in the ED setting. We found that roughly 7% of cases involved short term death or need for organ support. Obviously representing a class imbalance. We used SMOTE to synthetically generate more of the less represented class (target) and found that it significantly improved recall and performance on logistic regression. Below is a view of our pycaret evaluation using SMOTE with short term death or need for organ support as a target. Figure 2 represents the outputs which include a comparison of the different models based on distinct performance metrics along with ranking of feature importance.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.9414	0.8915	0.4086	0.6462	0.5004	0.4710	0.4851	52.6930
lightgbm	Light Gradient Boosting Machine	0.9407	0.8971	0.3934	0.6436	0.4879	0.4584	0.4743	5.2320
et	Extra Trees Classifier	0.9406	0.8964	0.3912	0.6424	0.4859	0.4564	0.4724	16.8090
dummy	Dummy Classifier	0.9282	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.9560
gbc	Gradient Boosting Classifier	0.9277	0.8729	0.4111	0.4971	0.4497	0.4114	0.4137	132.7780
ada	Ada Boost Classifier	0.9085	0.8511	0.4713	0.3880	0.4254	0.3762	0.3784	25.5340
dt	Decision Tree Classifier	0.8969	0.6938	0.4567	0.3392	0.3890	0.3341	0.3387	8.3350
qda	Quadratic Discriminant Analysis	0.8613	0.7961	0.5222	0.2643	0.3509	0.2825	0.3032	2.3960
nb	Naive Bayes	0.8485	0.8088	0.5547	0.2502	0.3447	0.2727	0.3006	1.2050
knn	K Neighbors Classifier	0.7995	0.8019	0.6786	0.2156	0.3272	0.2449	0.3009	8.8990
lr	Logistic Regression	0.7959	0.8542	0.7512	0.2247	0.3459	0.2645	0.3328	3.5530
ridge	Ridge Classifier	0.7935	0.8500	0.7415	0.2209	0.3403	0.2582	0.3253	1.2070
lda	Linear Discriminant Analysis	0.7934	0.8505	0.7425	0.2210	0.3405	0.2584	0.3258	2.4750
svm	SVM - Linear Kernel	0.7908	0.8501	0.7495	0.2200	0.3401	0.2576	0.3265	2.4180

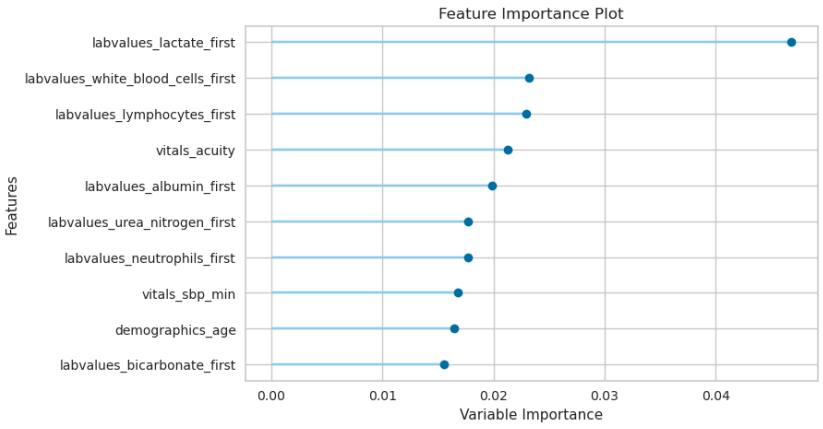


Figure 2

Based on the initial glimpse into model performance we decided to compare three distinct methods including logistic regression, SVM and random forest classifier. These models were showing strong performance in the initial pycaret models. They each offer different strengths and work in different ways. Logistic regression offers a fast, straightforward method that works well with tabular data like the data present in this data set. It is easy to understand and interpret the results. It is great for capturing linear relationships and provides probabilities. These probabilities can be clinically useful. Coefficients for a given feature represent an odds ratio as well. Penalization can be used for

feature selection, especially sparse features that may be present in large data sets like MIMIC IV. SVM works to find a 'plane' that maximizes the margins between classes. It works well with high dimensional data, and unlike logistic regression can capture non linear relationships which we thought might be present in our data. Additionally SVM does well with sparse data sets. Random forest regression does well with class imbalance and handles nonlinearities and interactions (ex vitals and labs). Random forest does well without scaling, which is great for mixed data like the data in this data set. Additionally it can be easy to visualize which features are most influential using forest models, showing what specific decision points led to branching.

For logistic regression as mentioned above, we did find that within pycaret, SMOTE improved recall and helped address class imbalance. When creating our logistic regression model, we made sure to address imbalance by defining class_weight="balanced" in the parameters. We also defined the regularization technique as elastic, we allowed for up to 5000 iteration to allow for convergence on this high dimensional data set. We used the saga solver which allowed for an elastic regularization technique and can work well on large sparse data sets. We used a simple median based imputation technique, or most frequent in case of any categorical data. Using 5-fold stratified cross validation we tuned the regularization strength using C, and offered 3 distinct values (0.1, 1.0, 10). This number defines how strong the regularization pressure is on the model, with lower C providing stronger regularization and reducing more features. In our elastic technique for regularization we also tuned whether a more L1 or L2 technique would be utilized using the L1 ratio. Numbers closer to 1 favor L1 (lasso) which is the technique that pushes some features to 0.

With our random forest classifier we ensured the class imbalance was addressed in an initial static parameter similar to how we handled logistic regression. Using 5-fold stratified cross validation we tuned the number of trees between 300 or 600 and the number of branches was between none and 12. Having between 300 and 600 trees should reduce variance as a general starting value. More trees decrease model variance but increase computational cost. Increasing branch depth improves the complexity of the model but risks overfitting. The minimal samples per leaf are tuned between 1,2 and 5, where small values are more likely to learn noise and cause overfitting. The max_features parameter was set to "sqrt" to limit the number of features that can be selected and prevent over reliance on one or two features, improving generalization.

With our SVM we again ensured to address class imbalance initially. We handled imputations similar to how it was handled with logistic regression. We also added a scaler, to scale features which is particularly important with SVM. The kernel is defined as rbf to allow for discovery of lon linear patterns. We added probability=False to reduce computational cost. Using 3 fold cross validation (3 instead of 5 to reduce computation cost) we tuned C with values between 1 and 10 which defines how wide the margins need to be, with smaller C being wider margins (less overfitting). We also tuned gamma with values between scale, and 0.1 which defines the influence any given point can have on the boundaries, larger numbers (larger gamma) means decision points limited to more local area allowing for more curved boundaries but more overfitting.

SUPERVISED EVALUATION:

We ultimately decided to use AUPRC as our evaluation metric because it balances precision with recall, thus balancing the importance of being sensitive to bad outcomes with the risk of activating too many false alarms. We imagine this model being used to provide an alert to the provider about patients that are at high risk for short term death or deterioration, too many false alarms will cause alert fatigue. AUPRC metric works better for class imbalanced data sets, as AUROC being deceivingly high in those circumstances. Figure 3 shows the comparison in performance between our three models. Of note the baseline prevalence of short term death or need for organ support is .07.

	Family	CV_AUPRC_mean	CV_AUPRC_sd
0	RandomForest	0.608	0.010
1	LogisticRegression	0.414	0.015
2	SVM-RBF	0.414	0.003

Figure 3

After further analysis on our best random forest classification model we were able to review the most important features from the model. In Figure 4 We see the first lactate was the most important with over 5% of forest splits gaining useful information from this feature.

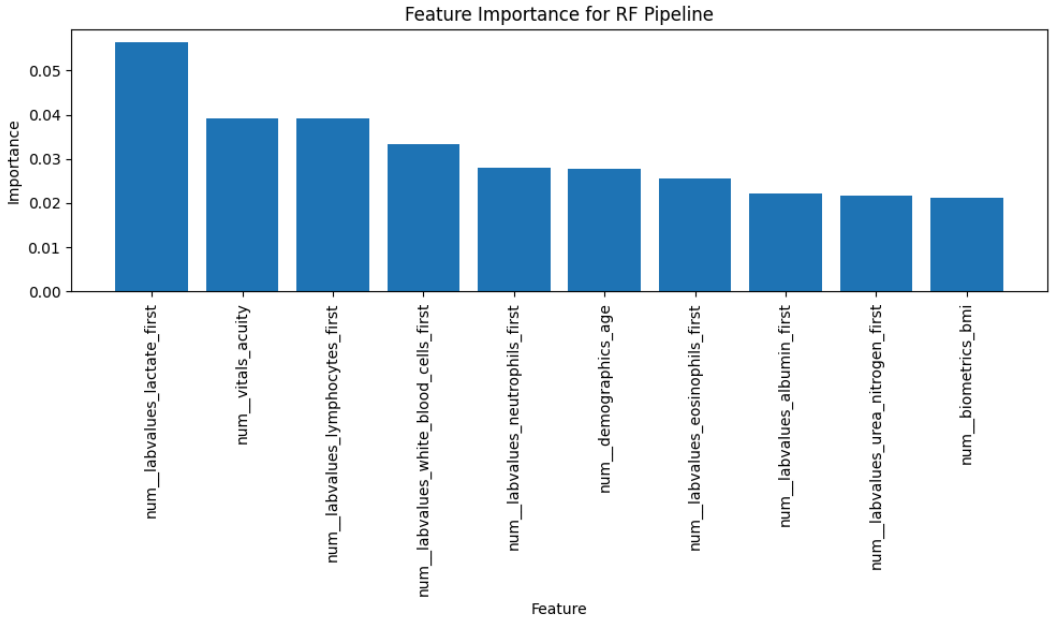


Figure 4

We further performed a permutation test to analyze the contribution of each feature based on the impact of removing the feature. In Figure 5 and 6, we compare the top 10 most useful features and our top ten most unuseful features. We see the first lactate lab was the most helpful, while systolic blood pressure max (vitals_sbp_max) was our least helpful. The values show the mean change in accuracy if the feature is removed. Negative values indicate the feature is unhelpful noise.

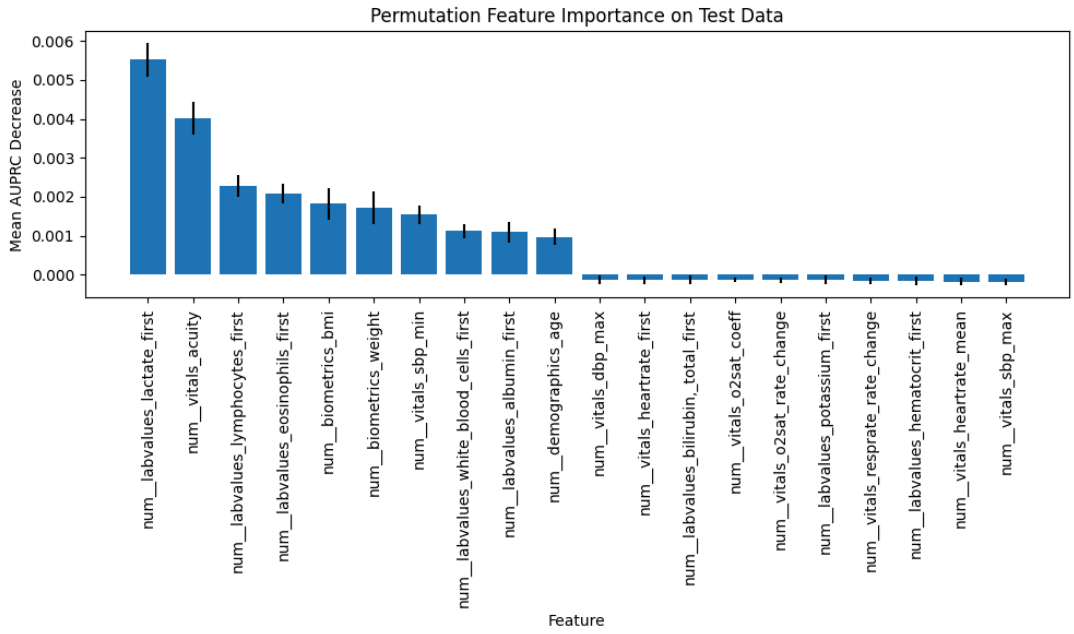


Figure 5

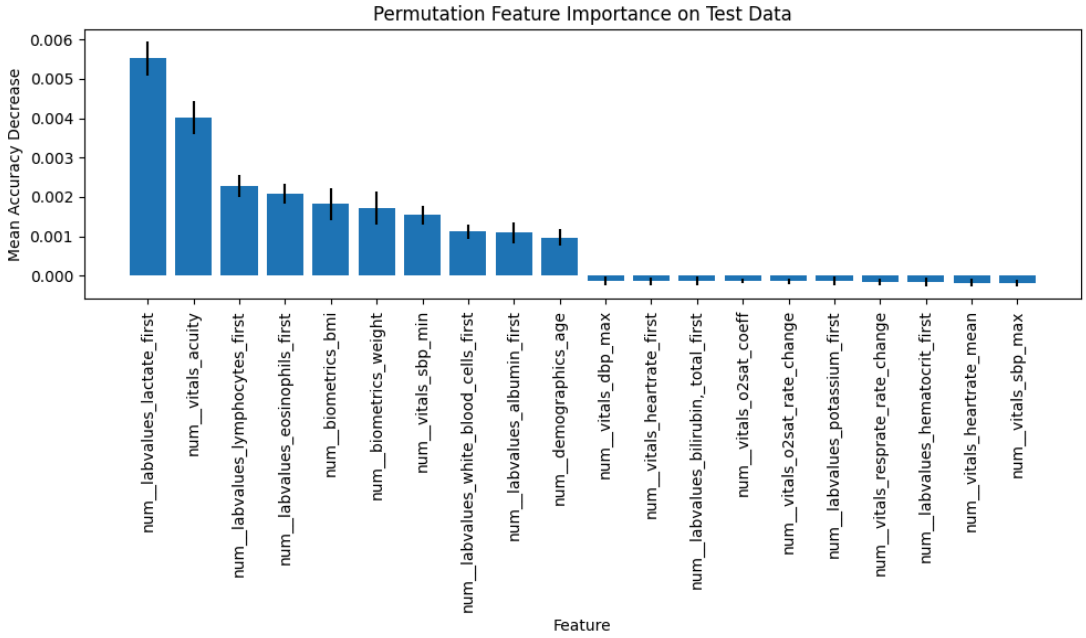


Figure 6

We also used our initial unsupervised learning output which reduced the data frame to three dimensions. This resulted in a significant decrease in performance on AUPRC, declining from 0.61 to 0.34. Figure 7 illustrates the impacts of the 3 principle components on AUPRC. This implies the diverse combination of features is important in helping to make predictions. This suggests there aren't many redundant features or high collinearity.

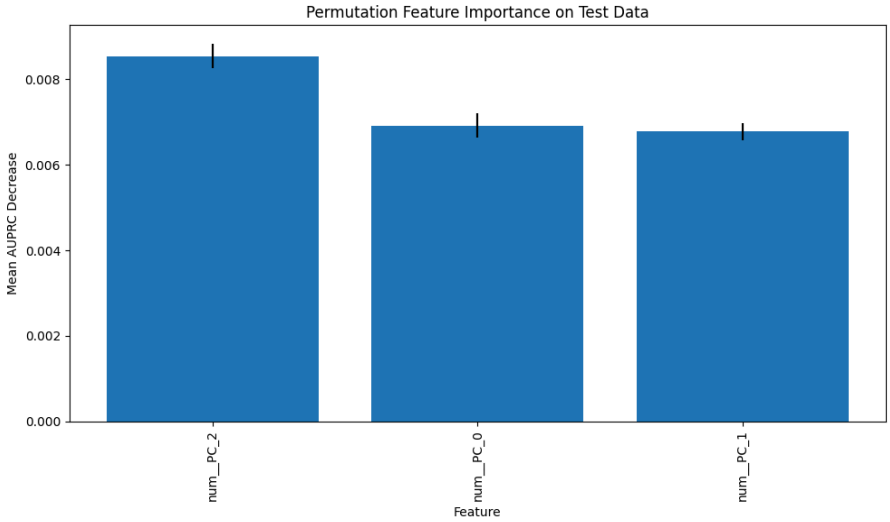


Figure 7

We decided to perform an ablation analysis on our model by rerunning it with each of the top 10 features removed to assess performance of the model when each feature was removed. The below dataframe shows the new AUPRC when the feature has been removed from the dataframe. Of note, the baseline AUPRC with all features is 0.6137 which is significantly higher than the baseline prevalence of 0.07 for the target outcome of death or deterioration. The model provides significant lift of ~9 times (about 9 X better at predicting positive from random guess). As shown in Figure 8 the model doesn't appear overly reliant on any one feature.

	feature	AUPRC
0	labvalues_lactate_first	0.608184
1	vitals_acuity	0.604703
2	labvalues_lymphocytes_first	0.611880
3	labvalues_eosinophils_first	0.612319
4	biometrics_bmi	0.610060
5	biometrics_weight	0.608505
6	vitals_sbp_min	0.612202
7	labvalues_white_blood_cells_first	0.610985
8	labvalues_albumin_first	0.610959
9	demographics_age	0.609839

Figure 8

We also performed a sensitivity analysis to evaluate how sensitive the model is to tuning, and specifically looked at the minimum sample per leaf to see how much that affected our AUPRC. We found the model is slightly sensitive to this tuning parameter, generally stable across different values as shown in Figure 9.

	min_samples_leaf	AUPRC
1	2	0.613740
2	5	0.604702
0	1	0.603668
3	10	0.583016

Figure 9

We took a closer look at the tradeoff between precision and recall within our model, compared to the baseline prevalence of the target outcome. In the below graph we can see a clear trade off between recall (never missing a short term 'bad outcome') and precision (every prediction of a short term 'bad outcome' is correct). Additionally, we can appreciate that our model is performing well when comparing the AUPRC to the baseline prevalence(see Figure 10).

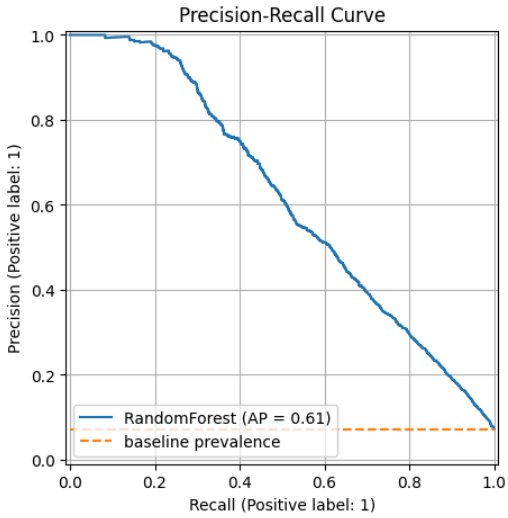


Figure 10

In our analysis we also saw a significant tradeoff with speed, dataframe size/complexity and performance. When reducing the dataframe to 3 principle components the model trained much quicker, but at a significant cost to performance (AUPRC from 0.61 to 0.34).

A failure analysis was performed to get insights into our model. We identified 79 false positives and 1173 false negatives. 29% of the false positives eventually went on to die outside the 28 days window or require ICU admission, suggesting there was a signal of a 'bad outcome', just not short term.. We further evaluated the top 3 false positives and top 3 false negatives with the highest and lowest probabilities respectively (most confidently wrong) along with a comparison of values for the 10 ten most predictive features within those cases. All of our false negatives were missing lactate levels, likely speaking to a potential over reliance on that feature. Generally speaking, there were many missing values on the most confident false positives and negatives, which potentially could be improved with a more sophisticated imputation strategy (ie iterative). The false negatives also had younger patients, which could speak to an overreliance on that feature as well. We attempted to apply SHAP analysis to the dataframe but due to the size and complexity, the run time exceeded 12 hours and we aborted.

Part B: Unsupervised Learning

METHODS DESCRIPTION:

Our team started with the unsupervised portion of the project, using a similar initial workflow as described above for supervised learning. For the unsupervised learning portion of the project we had two main goals: the first was to explore dimension reduction and our second was to identify hidden patient subgroups and compare the subgroups to potential prediction learning targets. Potential targets included various deterioration and mortality columns, along with over 1000 ICD-10 diagnoses codes. In total there were over 1400 potential targets for a ML model to predict.

The thought was to see if using PCA and then Kmeans would create groups which would overlap with potential targets for supervised machine learning.

Since there were so many features, decreasing potential columns to train on by completing principle component analysis (PCA) was explored. Prior to executing PCA we created a function which removes all target columns. This is important since leaving any target column for PCA would be considered leakage. We also used variance threshold to remove features which had low variance and thus would be computationally wasteful for PCA. RobustScaler was used to scale the remaining columns.

When applying KMeans we implemented the elbow method.As the number of actual clusters and centroids is increased, the within cluster sum of squares (WCSS) will decrease (because of overlap among the centroid group members, more groups = less distance between any one group which is a lower WCSS). The elbow method plots the WCSS against the value of K. When increasing the number of centroids as defined by k, the intra-cluster distance (WCSS) decreases but at some point will reach a plateau which will appear as an elbow, and represents a point of diminishing returns.

UNSUPERVISED EVALUATION:
The main parameter we experimented with for PCA was n_components and n_init for K-means. Since our data set had many sparse columns and many potential features we initially passed a fixed n=30 components for this parameter. This ensures 30 components or a 30 dimensional feature space. When we moved onto the next part of our unsupervised section, KMeans, we noticed good silhouette scores; with most runs between 0.683 and 0.898 After these results we tried passing n=0.95 for n_components; indicating we we wanted to reduce to a feature space that ensured 95% of variance was explained, rather than defining a static feature space. Ultimately this adjustment resulted in the creation of 3 principal components. Applying these changes resulted in excellent silhouette scores between 0.830 and 0.930. Figure 11 shows the silhouette score of the K means sweep with n=30 and n=0.95 respectively.

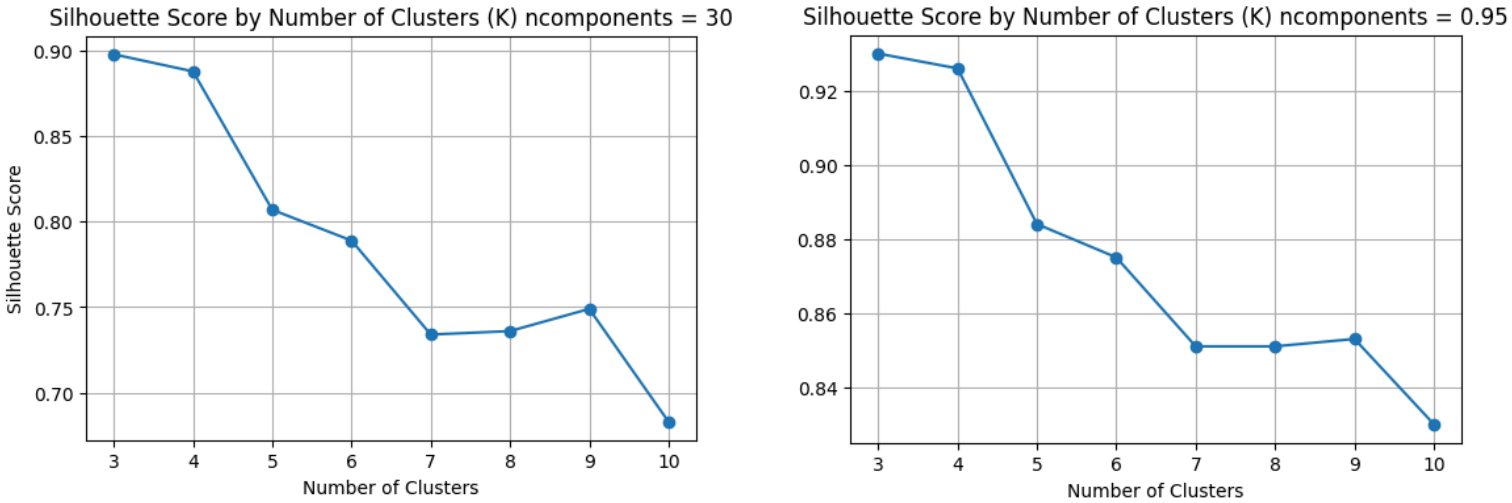


Figure 11

Given our goals of finding hidden patterns among patients and dimensional reduction, we decided to stay with n = 0.95 rather than trying to find a specific number of components. N_init , is the number of starting initializations

(centroid seeds) tried by Kmeans. We initially kept n_init = auto for KMeans and then tried n_init =10. The silhouette max score was equal to n_init =auto.

Silhouette score was used as the primary evaluation metrics to assess how tightly grouped the data points were within each cluster relative to the nearest clusters that neighbor. The optimal model achieved a silhouette score of 0.93 at k=3 with n_components set at 0.95.

The table below shows the predictive performance of our target outcome when the models were trained only on the principal components rather than the original feature rich data. The performance is comparatively much worse. These results indicate that valuable signal is lost when removing the other features; biomedical data is complex and when features are numerous and diverse using PCA may come at a performance cost. Overall, the random-forest models still performed the best with the SVM and logistic regression models suffering nearly equally in terms of performance reduction.

Family	CV_AUPRC_mean_PCA	CV_AUPRC_sd_PCA	CV_AUPRC_mean	CV_AUPRC_sd
RandomForest	0.341	0.052	0.608	0.010
LogisticRegression	0.168	0.008	0.414	0.015
SVM-RBF	0.154	0.006	0.414	0.003

A 2-D PCA Cluster plot (see Appendix, Attachment 1) was created to display each patient as a point, colored by cluster, The plot shows visible group separation along the first two principal components with some overlap. This shows that Kmeans identified subregions of higher patient density in the reduced feature space.

In addition to this, a Death/Deterioration overlay plot (see Appendix, Attachment 2) displayed the binary outcome variable (1=yes, 0=no) across the same two principal component axes PC1 and PC2. Areas of higher outcome density mapped to specific clusters, showing partial enrichment of deterioration within those subgroups.

Lastly, A 3-D PCA scatterplot (see Appendix, Attachment 3) displayed the three cluster structure across top principal components, indicating that the spatial alignment of the 2D output maintains integrity when projected into another axis.

Overall applying the selected unsupervised machine learning techniques led to significant declination in performance of our model in predicting the target outcome. The changes improved the speed of training our supervised model but the decline in performance was too costly to justify its use. Some natural clustering did appear within the reduced feature space however. The below tables compare the performance of predicting the target outcome with the principal components vs the original features.

Models with PCA

Model	Accuracy	AUC	Recall	Prec.	F1
Random Forest Classifier	0.7419	0.7282	0.7419	0.6815	0.6818
Logistic Regression	0.7325	0.0000	0.7325	0.5767	0.6238
SVM - Linear Kernel	0.7317	0.0000	0.7317	0.5597	0.6220

Models without PCA

Random Forest Classifier	0.9414	0.8915	0.4086	0.6462	0.5004
Logistic Regression	0.7959	0.8542	0.7512	0.2247	0.3459
SVM - Linear Kernel	0.7908	0.8501	0.7495	0.2200	0.3401

Discussion

Supervised Learning:

In performing the supervised learning portion, we found that traditional algorithms (mostly logistic regression and random forest classifiers) can achieve meaningful discrimination even on high dimensional data that is very sparse. In using the SMOTE technique in order to correct the class imbalance identified, logistic regression substantially improved recall - showing that careful data balancing is actually more impactful than raw model complexity. The random forest technique maintained robust performance without scaling and highlighted the importance of interpretable feature structures within mixed clinical data. Generally, we were surprised with how well our model performed in predicting the target after adjusting for class imbalance with SMOTE.

We were able to identify the most influential features such as first lactate level in predicting our target outcome. We were also able to learn that our random forest model is stable to parameter tuning with only small changes in performance with adjusting minimum leaf sample. We learned that our model performed significantly better than chance with a strong AUPRC. Surprising findings included how significant lymphocyte count was in contributing to AUPRC and how unhelpful systolic blood pressure was.

We felt AUPRC was a more appropriate metric for the development of an early warning system that would be used in the ED, especially when considering the underlying prevalence of the target outcome (rare). The AUPRC metric proved much more informative than the AUROC for the skewed dataset. This reinforced the importance of using precision/recall based evaluation for clinical imbalances in this context. Both false positives and false negatives are important to reduce; false positives could lead to too many alarms for staff, while false negatives would lead to a delay, or miss, in a critical medical intervention. All three modes performed relatively competitively, showing some level of convergence on predictions when decent feature engineering target formation is conducted.

Initially, we anticipated that the nonlinear approaches such as SVM and random forests would outperform logistic regression due to their ability to capture complex nonlinear interactions given a high dimensional data like the MIMIC-IV dataset. However, the results indicated that logistic regression when regularized and class balanced adjusted via SMOTE performed nearly as well indicating that the highly predictive signals (first lactate, vital sign instability) were linear in their relationships to short-term outcomes. This was surprising, and suggests that linear probabilistic models may work for early warning triage.

When conducting this portion, several challenges came up. The most significant challenges were mostly computational and some statistical. The first of these were high runtime on large scale models (especially SVM). This was mitigated by converting to a 3 fold cross validations and reducing the parameter tuning to only two options along with setting probability to false.

Another challenge faced was class imbalance, as previously mentioned. Initially, only about 7% of the data observed a positive class. We applied SMOTE (Synthetic Minority Oversampling Technique) to balance the imbalanced dataset that created new, synthetic data points for minority classes. As a result, minority class recall was improved.

With more time we would like to refine our imputation strategy, due to computational expense and time restraints we opted for more simple strategies such as mean and median. Iterative approaches would be more elegant. We would like to apply our models to a variety of targets, to include more long term and more composite outcomes. Additionally, we would like to incorporate the ECG tracings as a predictive feature. Incorporating the Great lakes computer GPU would likely have been helpful as running our models took considerable time which limited our ability to experiment with and tweak our models.

Unsupervised Learning:
In performing the unsupervised learning portion, we intended to uncover hidden structure within the cleaned MIMIC-IV .pkl file without using outcome labels. We utilized PCA to reduce clinical and demographic variables to a smaller set of components that are interpretable. In performing this, we found approximately 96%of the datasets' variance could be captured by three principal components. This supported the idea that much of the information in the vitals and labs provided was highly correlated - implying that a lower dimensional representation may preserve most of the clinical variation.

Following this, K-means clustering was applied to the PCA-transformed data. A K-means sweep from k=3 to k=10 was conducted and the best k (k=3, silhouette score = 0.93). This indicates a modest, but real separation among patient subgroups. These clusters were merged back with the original dataset to assess enrichment of deterioration related outcomes. This showed that some clusters exhibited noticeably higher proportions of deterioration events.

We expected K-means to isolate extremely distinct subtypes based on the cleaned pkl, but the clusters were much more overlapping than anticipated when visualized in both two and three dimensional spaces. This is a likely reflection of the inherent complexity and noise of real world health record data. In this context, physiological data blend instead of clear boundaries forming. This also showed how normalization and correlation structure can cause certain vitals or labs to dominate the component axes.

The largest challenge of completing the unsupervised portion of this project was attributed to scaling, ensuring no data leakage from features, and computational costs. Running PCA across over 100,000 points and 100+ features required significant preprocessing in order to ensure some type of convergence.

This was addressed by performing median imputation for missing numeric values, specifically in unsupervised.ipynb under (SimpleImputer(strategy=”median”)). Variance thresholding was also performed to remove near constant features under (VarianceThreshold(threshold=1e-5)). Lastly, standardizing numeric magnitudes via RobustScaler was utilized in order to reduce outlier influence. These were implemented in the pipeline prior to PCA to stabilize the transformation and improve silhouette consistency across the K-means sweep. Preprocessing proved critical for this step, as our PCA runs typically produced unstable or uninformative components dominated by single columns/features or noise.

With more time and resources, future work could include both experimenting with alternative clustering methods including Gaussian Mixture Models in order to capture irregular/probabilistic cluster shapes and the use of nonlinear dimensionality reduction methods such as t-SNE or UMAP to uncover manifolds. We also could have experimented with running supervised models with more principle components. It would be interesting to explore how including all features not used to compute principal components along with the principal components impacts model performance.

Ethical Considerations
Given the sensitive nature of the MIMIC-IV ED-MDS dataset from a demographic standpoint, both the unsupervised and supervised portions required careful consideration of the ethical implications using such data. The

MIMIC-IV dataset contains detailed patient records that reflect health and also demographic and socioeconomic variation. In our feature engineering steps kept this in mind in order to not enhance risk of encoding or amplifying bias if handled without care.

In the supervised learning portion (part A), class imbalance introduced a related ethical concern - a model that primarily predicts the majority "no-event" class would systematically fail to identify at-risk patients. This would disproportionately harm smaller or underrepresented subgroups whose health trajectories differ from the majority population. Through the use of SMOTE, we were able to balance class weighting to allow for minority outcome cases (those who actually deteriorated) were treated with equal importance in training the model. This improved recall and reduced the risk of false reassurance for ill individuals.

Our team prioritized transparency and responsible feature use at every stage. Given time constraints, we chose to remove most demographic features entirely rather than risk partial or inconsistent handling. This helped our models remain clinically focused and ethically neutral.

Statement of Work

Predicting Clinical Deterioration Utilizing Supervised and Unsupervised Learning was conducted by Hunter Belous, Allen Chezick, and Zach Sletten. The division of work was as follows:

Zach Sletten led data preprocessing and feature engineering, developed most of the cleaning pipeline, constructed mortality and deterioration targets, implemented supervised learning models, and authored the data sources and supervised learning section of the report along with components of the introduction and discussion.

Allen Chezick helped to develop the cleaning pipeline, performed unsupervised learning analysis, conducted PCA dimensionality reduction, fed principle components into, and edited supervised machine learning, completed K-Means clustering, optimized cluster selection through silhouette scoring, prepared enrichment summaries linking clusters to outcomes, and authored the unsupervised learning section of the report.

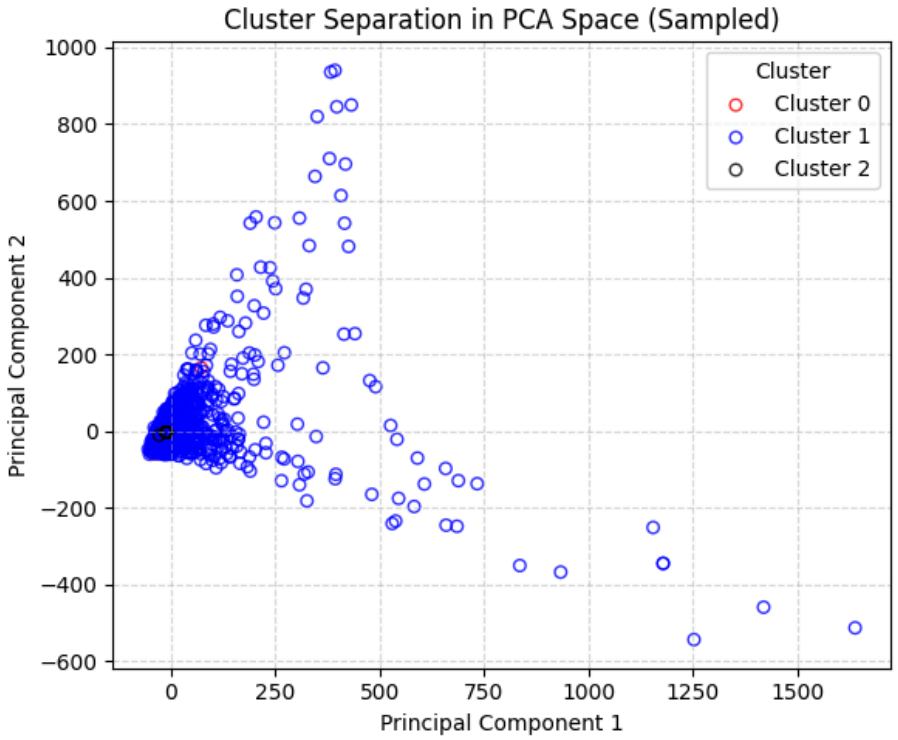
Hunter Belous coordinated final project integration, created code for visualizations, and reporting, standardized and refined notebooks for submission, ensured consistent data and file structure, contributed to the interpretation of the unsupervised outputs, authored the introduction, feature engineering, related work, discussion, and ethical considerations. Also finalized and submitted the applicable pipelines to GitHub.

References

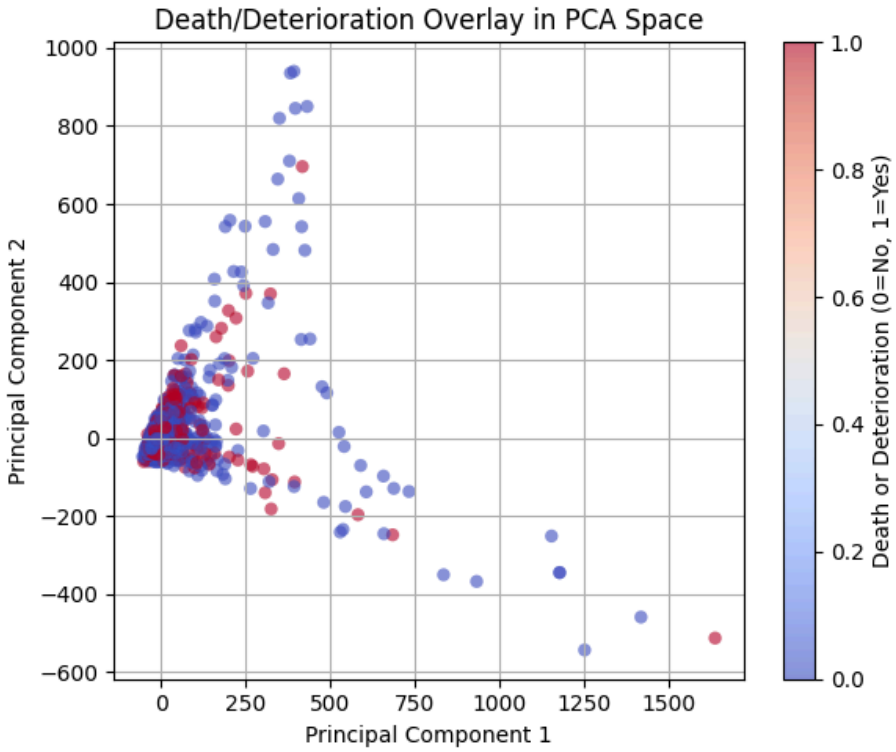
1. *Establishment of ICU mortality risk prediction models with machine learning algorithm using MIMIC-IV database*. (n.d.). PMC Home. https://pmc.ncbi.nlm.nih.gov/articles/PMC9139972/
2. *Machine learning models to predict 30-day mortality for critical patients with myocardial infarction: A retrospective analysis from MIMIC-IV database*. (n.d.). PubMed.
3. *MIMIC-IV-Ext-MDS-ED: Multimodal decision support in the emergency department - a benchmark dataset for diagnoses and deterioration prediction in emergency medicine*. (n.d.). PhysioNet. https://physionet.org/content/multimodal-emergency-benchmark/1.0.0/
4. *Unsupervised machine learning with cluster analysis in patients discharged after an acute coronary syndrome: Insights from a 23,270-Patient study*. (n.d.). PubMed.

Appendix

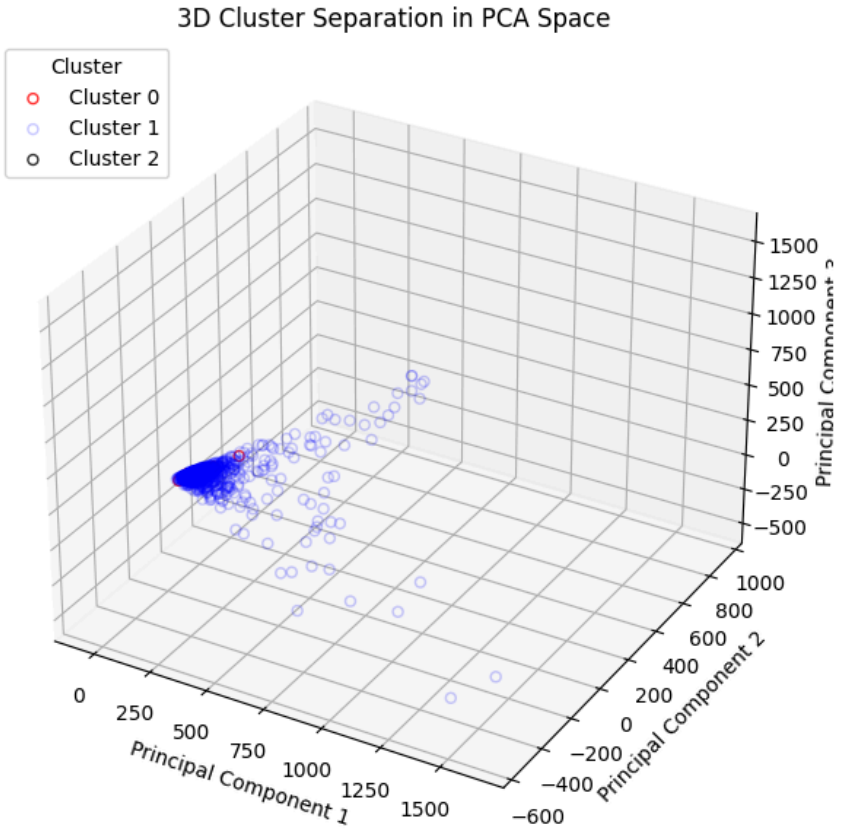
Attachment 1:



Attachment 2:



Attachment 3:



Final Features:

Columns in df_best.pkl:
['general_ed_diag_ed', 'general_ed_diag_hosp', 'demographics_gender', 'demographics_age', 'general_race', 'general_mortality_hours', 'general_mortality_days', 'demographics_ethnicity_asian', 'demographics_ethnicity_black/african', 'demographics_ethnicity_hispanic/latino', 'demographics_ethnicity_other', 'demographics_ethnicity_white', 'biometrics_bmi', 'biometrics_weight', 'biometrics_height', 'vitals_temperature_mean', 'vitals_temperature_median', 'vitals_temperature_min', 'vitals_temperature_max', 'vitals_temperature_std', 'vitals_temperature_first', 'vitals_temperature_last', 'vitals_temperature_rate_change', 'vitals_temperature_coeff', 'vitals_heartrate_mean', 'vitals_heartrate_median', 'vitals_heartrate_min', 'vitals_heartrate_max', 'vitals_heartrate_std', 'vitals_heartrate_first', 'vitals_heartrate_last', 'vitals_heartrate_rate_change', 'vitals_heartrate_coeff', 'vitals_resprate_mean', 'vitals_resprate_median', 'vitals_resprate_min', 'vitals_resprate_max', 'vitals_resprate_std', 'vitals_resprate_first', 'vitals_resprate_last', 'vitals_resprate_rate_change', 'vitals_resprate_coeff', 'vitals_o2sat_mean', 'vitals_o2sat_median', 'vitals_o2sat_min', 'vitals_o2sat_max', 'vitals_o2sat_std', 'vitals_o2sat_first', 'vitals_o2sat_last', 'vitals_o2sat_rate_change', 'vitals_o2sat_coeff', 'vitals_sbp_mean', 'vitals_sbp_median', 'vitals_sbp_min', 'vitals_sbp_max', 'vitals_sbp_std', 'vitals_sbp_first', 'vitals_sbp_last', 'vitals_sbp_rate_change', 'vitals_sbp_coeff', 'vitals_dbp_mean', 'vitals_dbp_median', 'vitals_dbp_min', 'vitals_dbp_max', 'vitals_dbp_std', 'vitals_dbp_first', 'vitals_dbp_last', 'vitals_dbp_rate_change', 'vitals_dbp_coeff', 'vitals_acuity', 'labvalues_absolute_basophil_count_first',

'labvalues_absolute_eosinophil_count_first', 'labvalues_absolute_lymphocyte_count_first', 'labvalues_alanine_aminotransferase_(alt)_first', 'labvalues_albumin_first', 'labvalues_alkaline_phosphatase_first', 'labvalues_asparate_aminotransferase_(ast)_first', 'labvalues_basophils_first', 'labvalues_bicarbonate_first', 'labvalues_bilirubin,_total_first', 'labvalues_calcium,_total_first', 'labvalues_chloride_first', 'labvalues_creatinine_first', 'labvalues_eosinophils_first', 'labvalues_glucose_first', 'labvalues_hematocrit_first', 'labvalues_hemoglobin_first', 'labvalues_inr(pt)_first', 'labvalues_lactate_first', 'labvalues_lymphocytes_first', 'labvalues_magnesium_first', 'labvalues_neutrophils_first', 'labvalues_pt_first', 'labvalues_ptt_first', 'labvalues_phosphate_first', 'labvalues_platelet_count_first', 'labvalues_potassium_first', 'labvalues_rdw_first', 'labvalues_red_blood_cells_first', 'labvalues_sodium_first', 'labvalues_troponin_t_first', 'labvalues_urea_nitrogen_first', 'labvalues_white_blood_cells_first', 'labvalues_ph_first', 'deterioration_severe_hypoxemia', 'deterioration_ecmo', 'deterioration_vasopressors', 'deterioration_inotropes', 'deterioration_mechanical_ventilation', 'deterioration_cardiac_arrest', 'deterioration_icu_24h', 'deterioration_icu_stay', 'deterioration_mortality_1d', 'deterioration_mortality_7d', 'deterioration_mortality_28d', 'deterioration_mortality_90d', 'deterioration_mortality_180d', 'deterioration_mortality_365d', 'deterioration_mortality_stay', 'admit', 'mortality_any', 'mortality_28d', 'mortality_365d', 'mortality_gt365d', 'mortality_category', 'clinical_deterioration_any', 'death_or_deterioration_any', 'month_time', 'tod_time', 'organ_support', 'mortality_class', 'support_mortality_combo', 'shortterm_death_or_deterioration', 'support_mortality_combo_id']