Alicia Qu

Final Report
Patient Survival Prediction

Problem Statement:

The potential client for this project could be healthcare providers or medical researchers who want to analyze patient data to improve patient care and develop new treatments. The client's motivation is to make data-driven decisions and improve patient outcomes. They care about this problem because analyzing patient data can help them understand patient needs, identify patterns, and develop personalized treatments.

So, I build a classification model to predict if the patient will survive or not in the hospital. By analyzing the feature importance of our model, we can help our client to help healthcare providers improve patient outcomes, improve patient care, reduce costs, and optimize resource utilization.

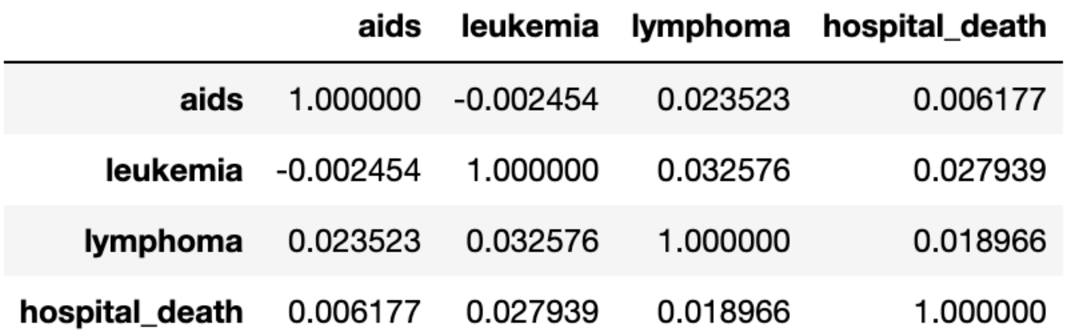
Data Wrangling

The original number of rows for the data is 91713, and it have 85 columns, but there are columns that contains missing values. I dropped some useless columns. Then I investigate categorical data and numerical data separately. I filtered out outliers and set a threshold that if the min value/max value of the columns is 3 times less/larger than the 3-sigma value, I identify it as outlier and dropped it. For missing values, I dropped rows that have more than 20% missing values. I left the rest missing values for now and will fill them later after train/test split to avoid data leakage.
After the data wrangling process, my data set have 84026 rows and 82 columns

Exploratory Data Analysis

Some of the feature have high variance which means they may lead overfitting of models. Models with high variance may fit the training data very well but fail to perform well on new, unseen data due to their inability to generalize beyond the specific training set. So, I may need to normalize the data before build the model.

The variance of 'aids', 'leukemia' and 'lymphoma' are relatively low, they also have a low correlation with our target variable (Table 1). So, we might need to drop these columns. But after investigating these three features, we can see they do have big impact on the deathrate of the patient. So, we'd better to keep them in our model.



	aids	leukemia	lymphoma	hospital_death
aids	1.000000	-0.002454	0.023523	0.006177
leukemia	-0.002454	1.000000	0.032576	0.027939
lymphoma	0.023523	0.032576	1.000000	0.018966
hospital_death	0.006177	0.027939	0.018966	1.000000

Table 1: Correlations between low variance features and target

I calculate the death rate by values of each categorical features and investigate the distribution of each categorical features I find there are two of them share most of their values (Figure 1), they are: apache_3j_bodysystem and apache_2_bodysystem. I only need to keep one of them.

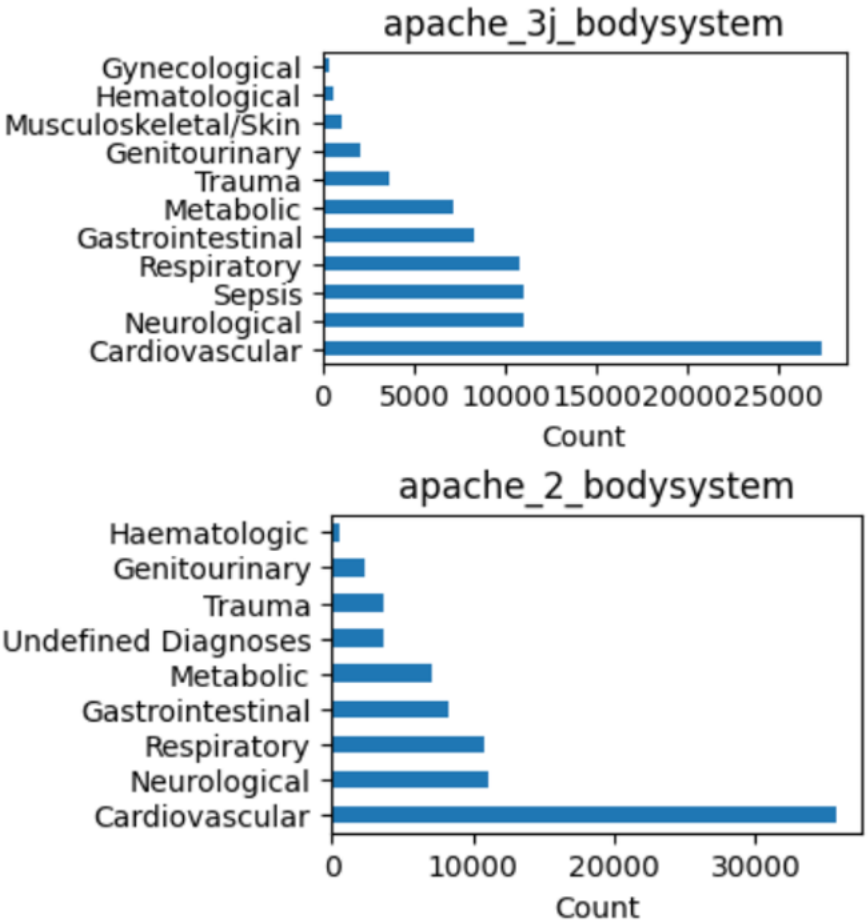


Figure 1 apache_3j_bodysystem and apache_2_bodysystem distribution

According to the death rate table, we find out that value 'Sepsis' of apache_3j_bodysystem have the highest death rate (Table 2), but there is not a separate value as 'Sepsis' in apache_2_bodysystem. So, in this case I will choose to keep apache_3j_bodysystem.

Apache 3j Body system	Death Rate
Cardiovascular	7.39%
Gastrointestinal	6.89%
Genitourinary	5.55%
Gynecological	0.68%
Hematological	7.92%
Musculoskeletal/Skin	4.21%
Neurological	7.54%
Respiratory	10.87%
Sepsis	15.06%
Trauma	6.30%

Table 2 Death Rate for different Apache 3j Body system values

I kept all other features for now, not indicating that all these features are useful. I will do features selection later before modeling.

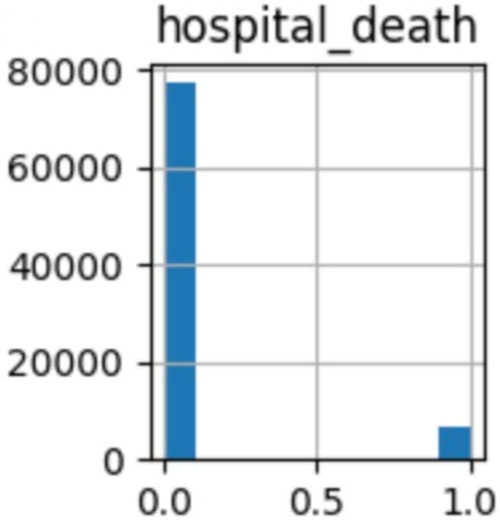


Figure 2 Target variable distribution

Figure 2 shows the distribution of my target variable. This is a super unbalanced data set. The ratio of two categories is larger than 10:1. We need to do resampling before modeling.

Pre-Processing & Modeling

After the EDA I know some of my features are super skewed, so, before modeling, I applied Log Transformations for some of the features.

Also, in the EDA process, I checked the correlations between features and the target, I know some of the features are not that useful, so I need to do feature selection before modeling.

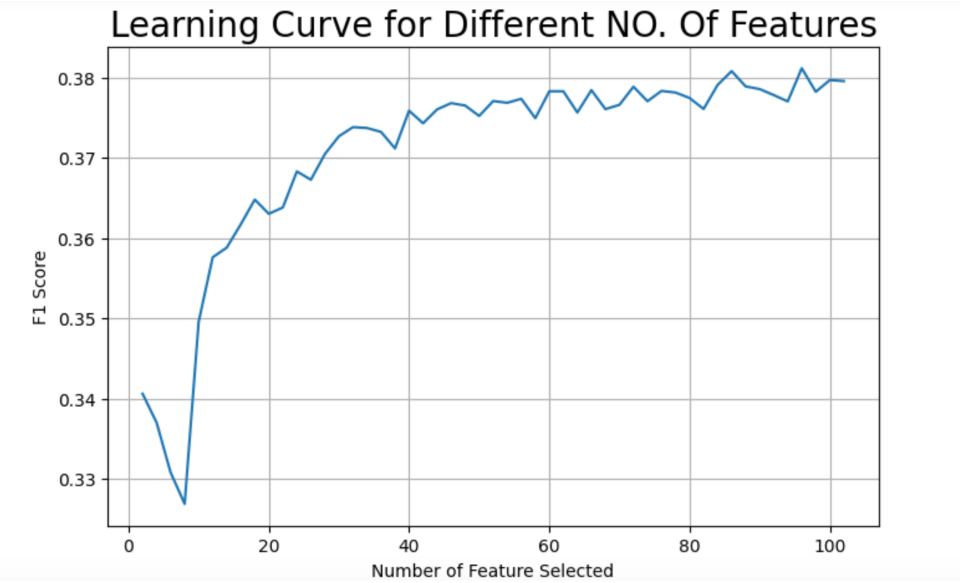


Figure 3

Figure 3 is the learning curve of a Random Forest model while using different number of features. I use f1 score as the metric for this curve. From the graph we can easily tell that there is a sharp rise for the f1 score at the beginning as the feature number increase, but when the feature number reaches around 25, the improve the f1 score began to slow down. For this prediction I will choose 60 features to use in the models.

For this problem, I use 6 classification models:

Logistic Regression, Bernoulli Naïve Bayes, Random Forest, XGBoost, Balanced Random Forest, Balanced Bagging.

From the EDA I know that my data set is a super unbalanced. So, for the first 4 models I add a under sampling step in the pipeline. The Balanced Random Forest and Balanced Bagging are both ensemble classifier that will do resampling automatically.

Model Selecting

The logistic Regression model has the highest test accuracy, but its test accuracy is much higher than its train accuracy, so this model has a underfitting problem. Overall, the Balanced Bagging model perform best in terms of accuracy. Random Forest, XGBoost and Balanced Random Forest all have a relatively decent accuracy. (Figure 4)

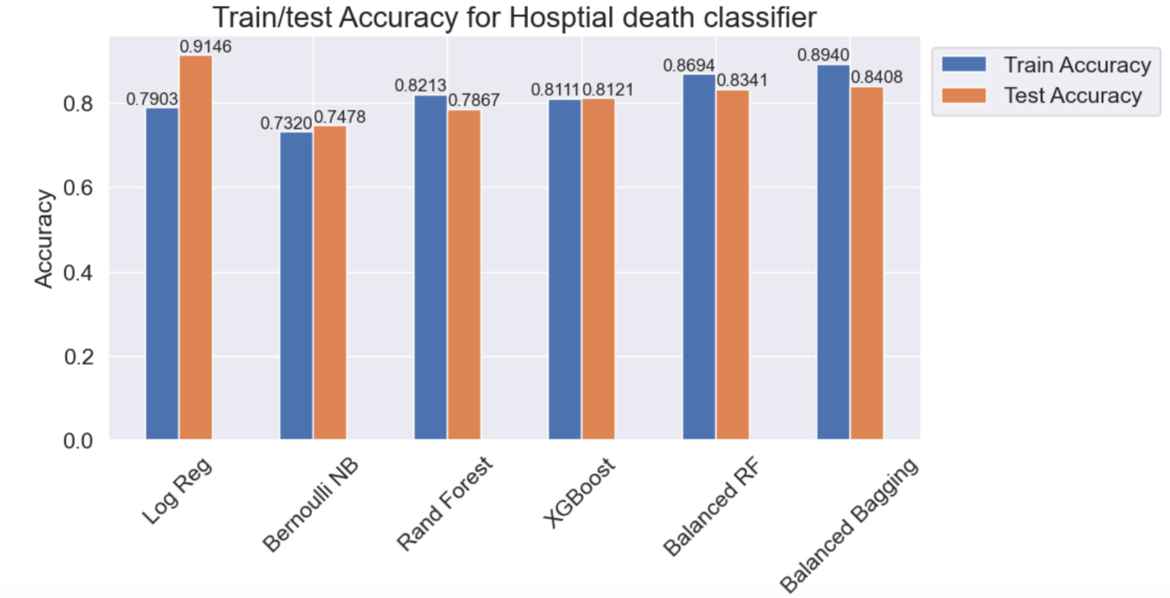


Figure 4 Accuracy

The f1 score for all these 6 models are not very high. Because all these models perform poor on predicting the positive value. Balanced Random Forest has the highest f1 score in these models. (Figure 5)

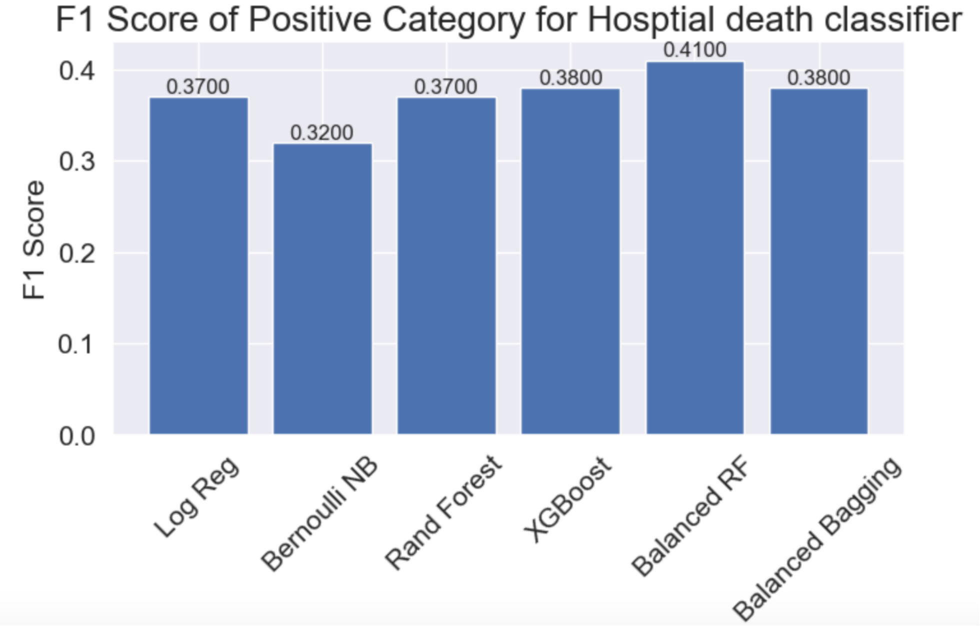


Figure 5

Besides accuracy and f1 scores I also want to investigate the tradeoff between precision and recall for all these models. From the graph we can easily infer that although Balanced Bagging have a relatively higher accuracy, its precision drops fast as its recall increases, so does the Naïve Bayes metrics. Logistic Regression and Balanced Random Forest perform better in terms of the Precision-Recall Curve while Balanced Random Forest and Random Forest yield a better AUC. (Figure 6, Figure 7)

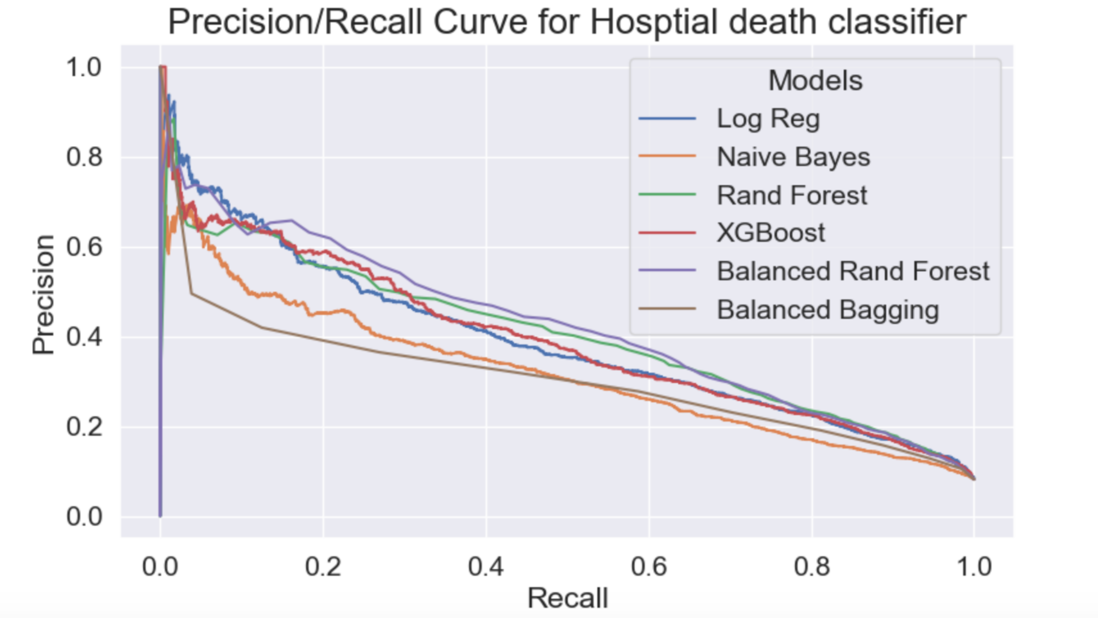


Figure 6

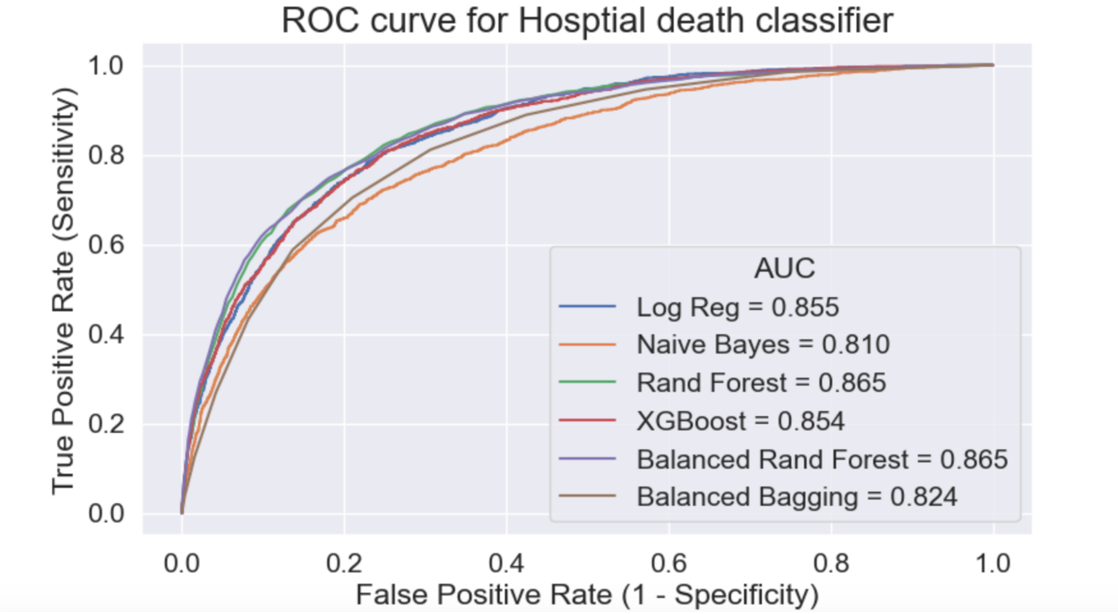


Figure 7

Take the accuracy, confusion matrices, f1 score, precision-recall tradeoff, and AUC score for all these models into consideration, I chose Random Forest, XGBoost and Balanced Random Forest to apply the hyperparameter tuning process.

For this problem, we are trying to target more patient failure as well as reduce the false prediction of patient failure, so precision and recall are both important to this problem, I will choose f1 score as the scoring method for the hyperparameter tuning.

After tuning the hyperparameters, we find our best model: Random Forest. Its train accuracy is 0.9534 and test accuracy is 0.9095, which is much higher than the result before the tuning. Its overall f1 score is around 0.47 and its AUC score is around 0.87. These three metrics for Random Forest all rank the top.

Takeaways

1. My model improves the predictive power on patient death more than 10 times

   Before we have this model, our client has some metrics to predict the patient death. There are two metrics, apache_4a_icu_death_prob and apache_4a_hospital_death_prob

   The APACHE IVa probabilistic prediction of in ICU mortality for the patient which utilizes the APACHE III score and other covariates, including diagnosis. The APACHE IVa probabilistic prediction of in-hospital mortality for the patient which utilizes the APACHE III score and other covariates, including diagnosis.

   I investigated the predictive power of these methods and compare them with our best model's predictive power. From the result, when use the apache_4a_icu_death_prob or the apache_4a_hospital_death_prob to predict the patient death, both have a relatively high accuracy. But when you look at the confusion matrix, you will notice that both methods perform pool on predicting the positive value which means use these methods you can predict the survived patient with a high prediction accuracy, but you can hardly target the patient that will die. On the other hand, our model somehow increases the false negative rate for the patient survived, but it also hugely increases the precision and recall for the patient death prediction.

2. Apache scores are critical metrics

   Our best model has 60 features in total. Apache_4a_hospital_death_prob and apache_4a_icu_death_prob out of all the 60 features contribute the most. But together they explain around 18% of the model. The rest 58 features together explain 82%. (Figure 8)
   So, Apache_4a_hospital_death_prob and apache_4a_icu_death_prob are very important metrics to predict the patient survival. Although by themselves they cannot predict the patient death properly, we do need them in our model.

Besides these two apache scores, there are many other apache scores that also contribute to this model. In the top 25 important features for this model, there are 10 apache score features.

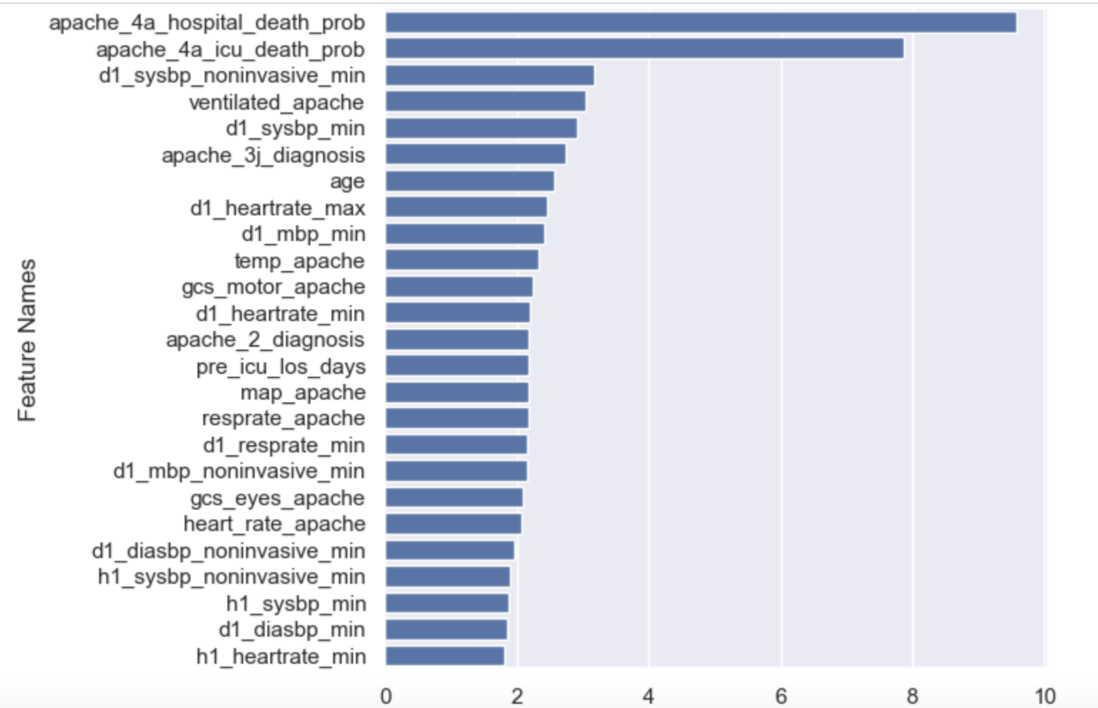


Figure 8 Feature importance (Top 25)

3. Control the patient's highest heart rate and lowest systolic blood pressure during the first 24 hours of their unit stay help the patient survive.

   From the heatmap (Figure 9) of correlation for the top 15 most important features and the target variable, we can see that besides apache_4a_hospital_death_prob and apache_4a_icu_death_prob, ventilated_apache, age and d1_heartrate_max are relatively highly positively correlated with our target. d1_sysbp_noninvasive_min, d1_sysbp_min, d1_mbp_min, temp_apache and gcs_motor_apache are relatively highly negatively correlated with our target feature.

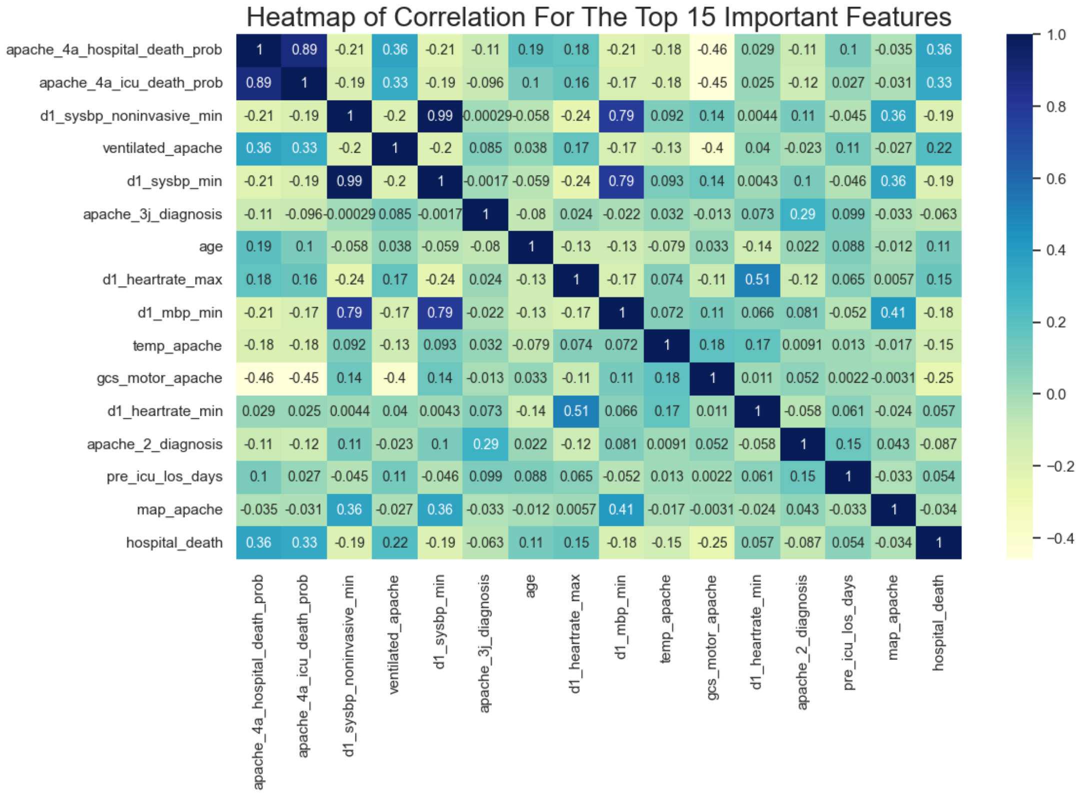


Figure 9

The average d1_sysbp_noninvasive_min is much lower if the patient did not survive in the end. (Figure 10)

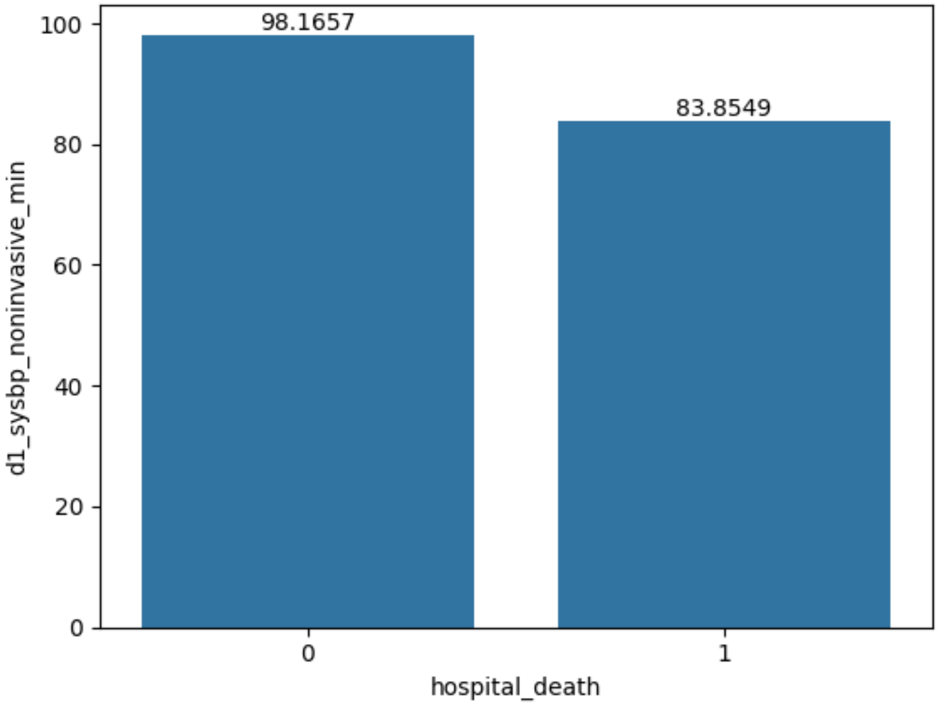


Figure 10 Hospital death vs. average d1_sysbp_noninvasive_min

4. Is use ventilation machine increase the possibility of the patient to die?

   From the heat map (Figure 9) we can tell that ventilated_apache is highly positively correlated to the hospital_death. Does that mean we should not apply ventilation on patients? The answer is no. we can only infer that if a patients' satiation is too bad and must use ventilation machine, it's more likely that this patient cannot survive in the end. So maybe we need to take actions to prevent the patient's situation getting worse to use the ventilation machine.

Future Research

This data set is super unbalanced, the count of value 1 for our target are only around 5k. All my models perform relatively pool on predicting the '1' for our target. So, to improve my mode, I would like the collect more data in the future.

Also, as I mentioned before, I want to know if take actions before the patients have to use ventilation machine will increase their survive rate. Also, is control the patient's highest heart rate and lowest systolic blood pressure during the first 24 hours of their unit stay can really help the patient survive. If it's possible I would like to do some more analysis on the new data set.