# Predictive Analysis of in-hospital Mortality of ICU Patient's

**Akshat Karambe, Manish Kumar Lomada, Sumedh Kulkarni, Prajwal Parlawar**
***Dept of Industrial Engineering, Northeastern University, Boston, MA***

## Abstract

This project is about developing predictive analysis for In-Hospital ICU mortality rate depending on patient specific data. By observing dynamic changes in vital signs may aid in early identification of patients with elevated risk as well as those whose status maybe stable or improving. We propose that this approach would be more useful to give accurate results about the condition of the patient and can help us in improving the treatment process required. The data used in this paper consist of 5 general descriptors and 36 time series variables which are calculated from the 48 hours stay in an ICU of every patient of the total 4000 recorded from the MIMIC II database. Our main goal is about obtaining the highest accuracy comparing the sensitivity values of different methodologies like Support Vector Machine, Logistic Regression and Linear Discriminant Analysis.

## 1. Introduction:

There are many techniques such as acuity scores of SAPS, APACHE and SOFA which are used to compare the efficiency in medication, type of surgery, hospital type and many more which can affect the Mortality of the Intensive Care Unit (ICU) Patient's. When we compare the mortality rate of the population treated in such conditions, it comes out to be different for various factors. However, we are making use of observations including time series of vital signs after the ICU admission during the 48 hours stay of that patient. Our hypothesis is that using this time series data we can predict the accurate number of deaths which can be used for early detection of disease and can help in effective medication technique for both the surviving and improving patients.

## 2. Background Related Work:

The objective of this paper is to determine how the parameters other than SAPS, APACHE and SOFA are going to affect the mortality of a patient in ICU. we are using in-hospital death as the out-come variable to be predicted in the challenge. This was done in order to ensure that the risk estimates accurately reflected individual patient risks, rather than simply the risk for the entire population of patients. Given that the data sets were created from a diverse population with a wide variety of life-threatening conditions, with frequent missing and occasionally incorrectly recorded observations, idiosyncrasies of care administration, and highly unbalanced class sizes, we expected this challenge to be difficult. We removed the columns and rows having high NA's and replaced the left over missing data using the means of the respective parameters to overcome this problem. Since the parameters vary for patients who are alive and who are deceased, we bifurcated the data into two lists. Now we replaced the missing data with their means of parameters depending on the outcome of the patient's mortality rate. We also removed the generic parameters which don't affect the outcome of the model

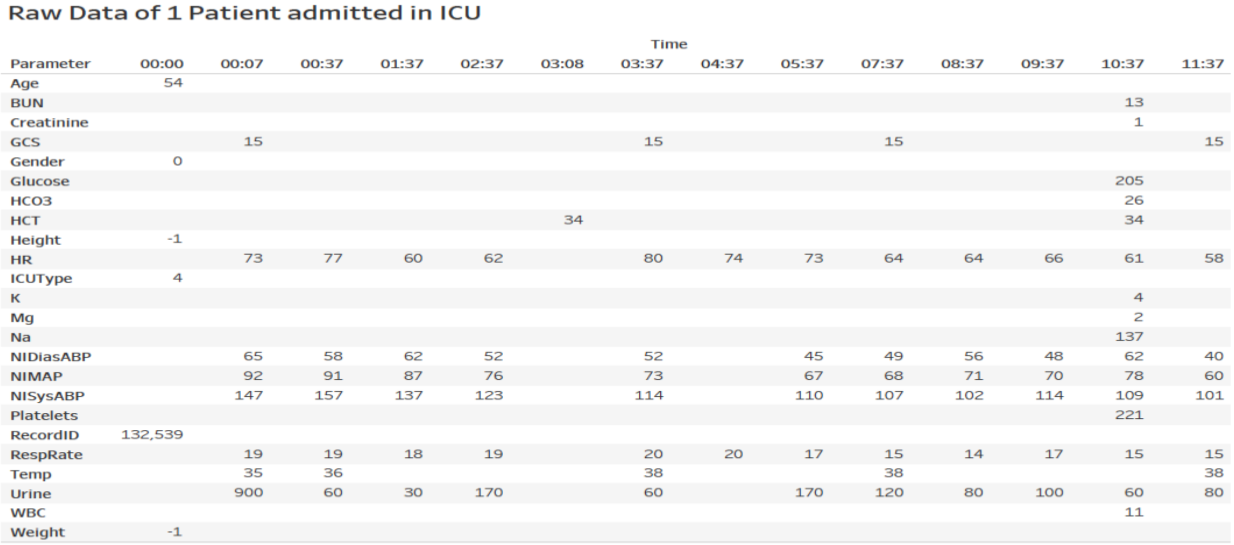
Now after cleaning the data and we had to decide the vital parameters contributing to the outcome of patient's mortality. Using co-variances as our criteria, we removed the parameters which are highly dependent on each other and brought down the dimensionality of the data. The plots of the co-variances can be seen below. The dark blue represents the parameters which have high co-variance, which are significantly reduced after data processing. After dividing the data into training and testing we noticed there's an imbalance in the data of patient's mortality.

To avoid overfitting, we decided to use a combination of over sampling and under sampling to make the data more balanced to make a better estimate and increase the efficiency of the model. This is further explained in the paper. Cross validation has been a viable method to make optimal analysis of the data. We created a five-fold cross validation and made it repeat three to get the best possible output. Logistic Regression, Support Vector Machining and Linear Discriminant Analysis are few of the models we used for classification and regression analysis. By running the model on the three techniques, we estimated the accuracy, specificity and sensitivity of the training data. This has been explained in detail as we go further down in the paper.

## 3. Problem Definition:

### 3.1. Data Description

Data consist of 4000 subjects whose age at ICU admission is more than 16 and whose stay is at least 48 hours in the ICU. We are dividing this dataset into 70% for training and 30% for testing. For each subject there are 41 testing parameters out of which 5 are generic (Age, Gender, Height, Weight, ICU Type). These 41 parameters were recorded at least once during the first 48 hours after the admission into ICU. The 36 non-generic time series variables may have multiple observations associated with a time-stamp indicating the elapse time of the observation in hours and minutes since ICU admission. There were 4000 files of patients having multiple inputs, we used python to convert rows to columns and took mean of every value that is repeating for every patient separately to convert it into a single csv file.

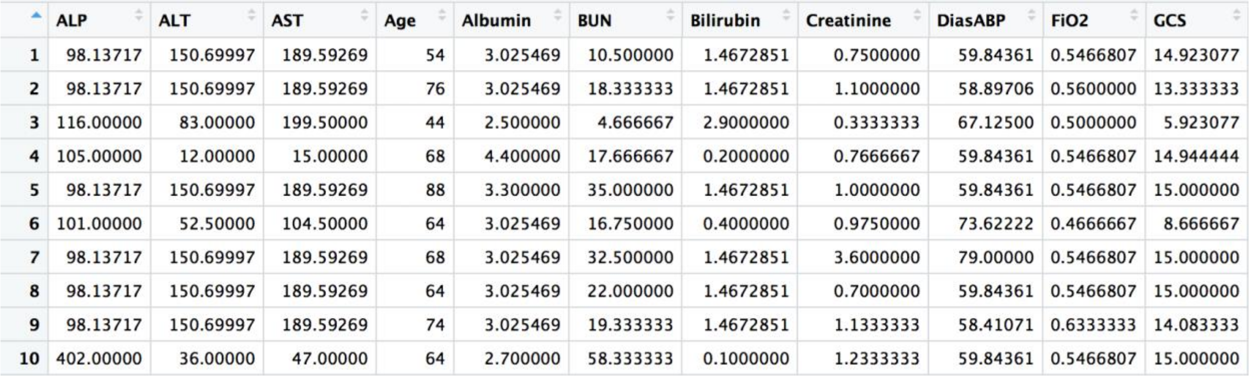


Raw Data of 1 Patient admitted in ICU

| | Time | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Parameter | 00:00 | 00:07 | 00:37 | 01:37 | 02:37 | 03:08 | 03:37 | 04:37 | 05:37 | 07:37 | 08:37 | 09:37 | 10:37 | 11:37 |
| Age | 54 | | | | | | | | | | | | | |
| BUN | | | | | | | | | | | | | 13 | |
| Creatinine | | | | | | | | | | | | | 1 | |
| GCS | | 15 | | | | | 15 | | | 15 | | | | 15 |
| Gender | 0 | | | | | | | | | | | | | |
| Glucose | | | | | | | | | | | | | 205 | |
| HCO3 | | | | | | | | | | | | | 26 | |
| HCT | | | | | | 34 | | | | | | | 34 | |
| Height | -1 | | | | | | | | | | | | | |
| HR | | 73 | 77 | 60 | 62 | | 80 | 74 | 73 | 64 | 64 | 66 | 61 | 58 |
| ICUType | 4 | | | | | | | | | | | | | |
| K | | | | | | | | | | | | | 4 | |
| Mg | | | | | | | | | | | | | 2 | |
| Na | | | | | | | | | | | | | 137 | |
| NIDiasABP | | 65 | 58 | 62 | 52 | | 52 | | 45 | 49 | 56 | 48 | 62 | 40 |
| NIMAP | | 92 | 91 | 87 | 76 | | 73 | | 67 | 68 | 71 | 70 | 78 | 60 |
| NISysABP | | 147 | 157 | 137 | 123 | | 114 | | 110 | 107 | 102 | 114 | 109 | 101 |
| Platelets | | | | | | | | | | | | | 221 | |
| RecordID | 132,539 | | | | | | | | | | | | | |
| RespRate | | 19 | 19 | 18 | 19 | | 20 | 20 | 17 | 15 | 14 | 17 | 15 | 15 |
| Temp | | 35 | 36 | | | | 38 | | | 38 | | | | 38 |
| Urine | | 900 | 60 | 30 | 170 | | 60 | | 170 | 120 | 80 | 100 | 60 | 80 |
| WBC | | | | | | | | | | | | | 11 | |
| Weight | -1 | | | | | | | | | | | | | |

**Figure 3.1: Pre-processed data**

Data Cleaning was started with elimination of generic columns which wouldn't be contributing towards the mortality rate of that patient which are Gender, Height, RecordID, ICUType. We also eliminated columns having missing data of more than 60% in a single column which are Cholesterol, RespRate, TroponinI, TroponinT followed by missing values of more than 68% per row for every patient which deleted another 20 rows. We used dplyr function to separate out survived and dead patients which followed with taking column wise mean of survived and dead patients separately so the final data we got was of 3980 Patients and 28 parameters contributing towards the mortality of that patient.



| | ALP | ALT | AST | Age | Albumin | BUN | Bilirubin | Creatinine | DiasABP | FiO2 | GCS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 98.13717 | 150.69997 | 189.59269 | 54 | 3.025469 | 10.500000 | 1.4672851 | 0.7500000 | 59.84361 | 0.5466807 | 14.923077 |
| 2 | 98.13717 | 150.69997 | 189.59269 | 76 | 3.025469 | 18.333333 | 1.4672851 | 1.1000000 | 58.89706 | 0.5600000 | 13.333333 |
| 3 | 116.00000 | 83.00000 | 199.50000 | 44 | 2.500000 | 4.666667 | 2.9000000 | 0.3333333 | 67.12500 | 0.5000000 | 5.923077 |
| 4 | 105.00000 | 12.00000 | 15.00000 | 68 | 4.400000 | 17.666667 | 0.2000000 | 0.7666667 | 59.84361 | 0.5466807 | 14.944444 |
| 5 | 98.13717 | 150.69997 | 189.59269 | 88 | 3.300000 | 35.000000 | 1.4672851 | 1.0000000 | 59.84361 | 0.5466807 | 15.000000 |
| 6 | 101.00000 | 52.50000 | 104.50000 | 64 | 3.025469 | 16.750000 | 0.4000000 | 0.9750000 | 73.62222 | 0.4666667 | 8.666667 |
| 7 | 98.13717 | 150.69997 | 189.59269 | 68 | 3.025469 | 32.500000 | 1.4672851 | 3.6000000 | 79.00000 | 0.5466807 | 15.000000 |
| 8 | 98.13717 | 150.69997 | 189.59269 | 64 | 3.025469 | 22.000000 | 1.4672851 | 0.7000000 | 59.84361 | 0.5466807 | 15.000000 |
| 9 | 98.13717 | 150.69997 | 189.59269 | 74 | 3.025469 | 19.333333 | 1.4672851 | 1.1333333 | 58.41071 | 0.6333333 | 14.083333 |
| 10 | 402.00000 | 36.00000 | 47.00000 | 64 | 2.700000 | 58.333333 | 0.1000000 | 1.2333333 | 59.84361 | 0.5466807 | 15.000000 |

**Figure 3.1: Processed Data**

## 3.2. Problem Statement:

Acuity scores, such as APACHE, SAPS, MPM, and SOFA, are widely used to account for population differences in studies aiming to compare how medications, care guidelines, surgery, and other interventions impact mortality in Intensive Care Unit (ICU) patients. Our Project is focusing more on the other vital parameters which contribute to the mortality of the patient in an ICU. Using Support Vector Machining, Logistic Regression and Linear Discriminant Analysis, we are able to validate by performing classification and regression analysis, giving statistical significance to the model

## 4. Methods:

The above flowchart explains the steps followed right from the scratch data to applying different models to obtain particular accuracy, sensitivity and specificity and used all the models to validate the training and testing data.

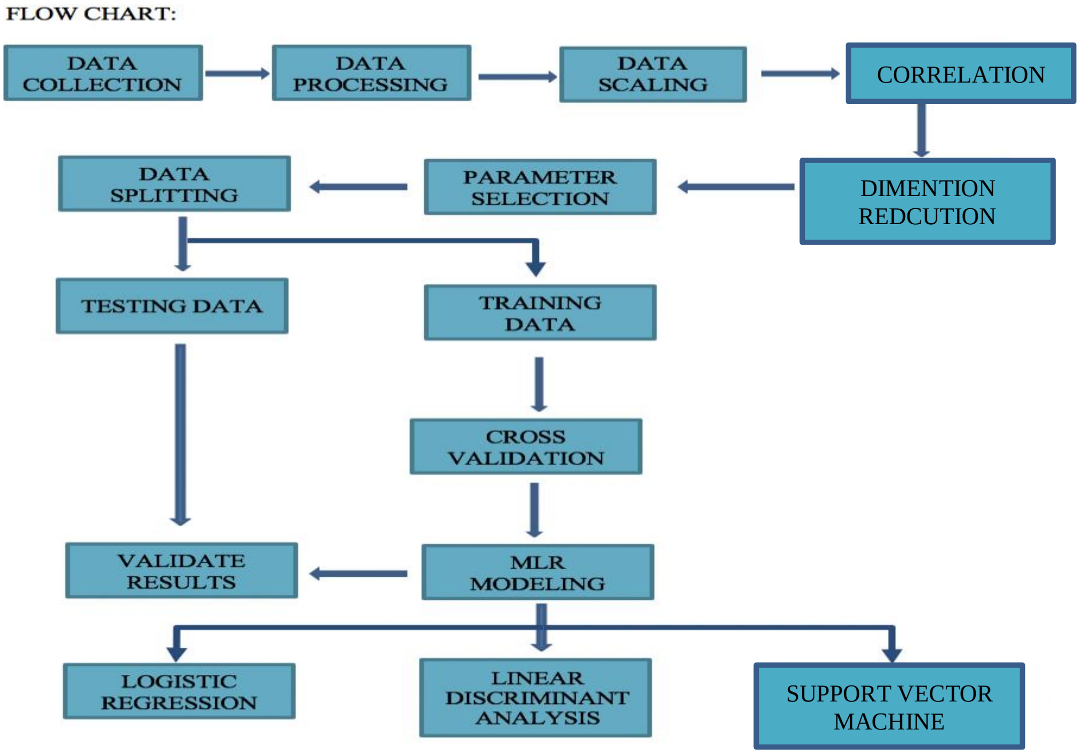


**Figure 4.1. Methodology for performing different models**

### 4.1. Covariance and Correlation:

Covariance and Correlation basically describes how to different variables are related, they are positive if the move in the same direction and negative if the move in the opposite direction. Covariance has two outcomes as 0 or 1 whereas correlation as a third type of -1 too which makes it different, where 0 acts as neutral. We have applied covariance and correlation functions just to check the relations between every parameters and the percentage contribution of all the PCs in order to choose PCA components.

After plotting the covariances, we found out that we are getting around 95% of variance till the 27th Variable. We added a filter at 60% and we removed all the variable which are correlated more than 0.7 where the dimensions reduced from 33 to 28 and we then proceeded with data partitioning, balancing , running five-fold cross validation and applying different models to predict and compare the accuracy of those models.

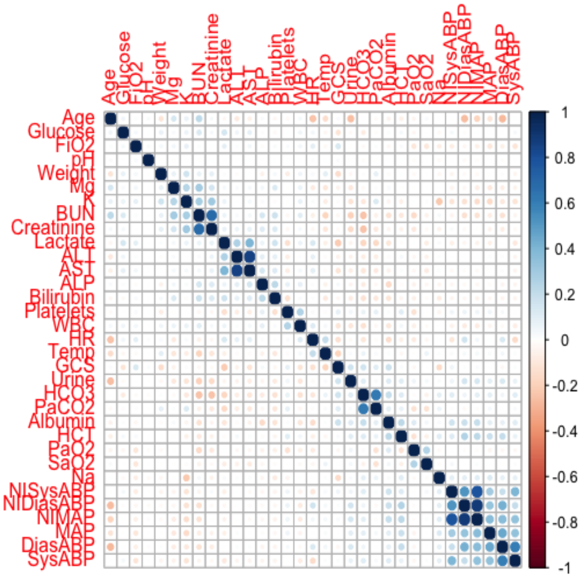


Figure 4.2.1 Covariances of Parameters

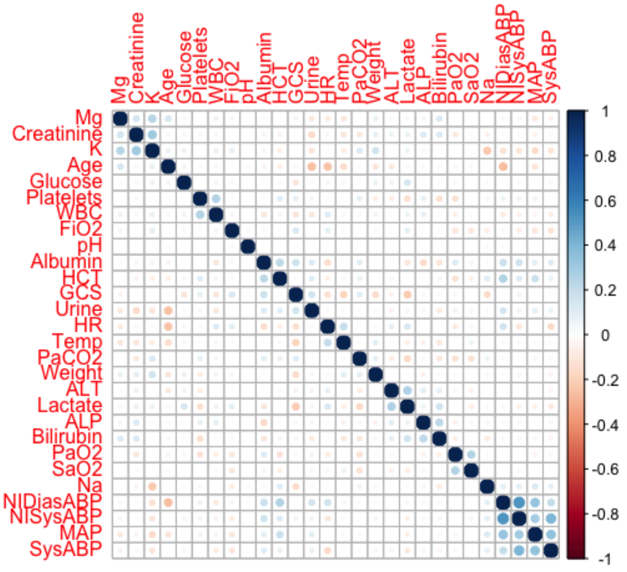


Figure4.2.2 : Applied Correlation Filter

### 4.2. Five-Fold Cross Validation:

We were running processed data without cross validation but the problem we faced was that accuracy was variating. We then ran five-fold cross validation on the whole data, but we got 100% accuracy which made us think the model was wrong. Then we split the data into 70% of training and 30% of testing and we performed cross validation on training and then predicted the model on the testing dataset.

In 5-fold cross-validation, the original sample is randomly partitioned into 5 equal size subsamples. Of the 5 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 4 subsamples are used as training data. The cross-validation process is then repeated 5 times (the folds), with each of the 5 subsamples used exactly once as the validation data. The 5 results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

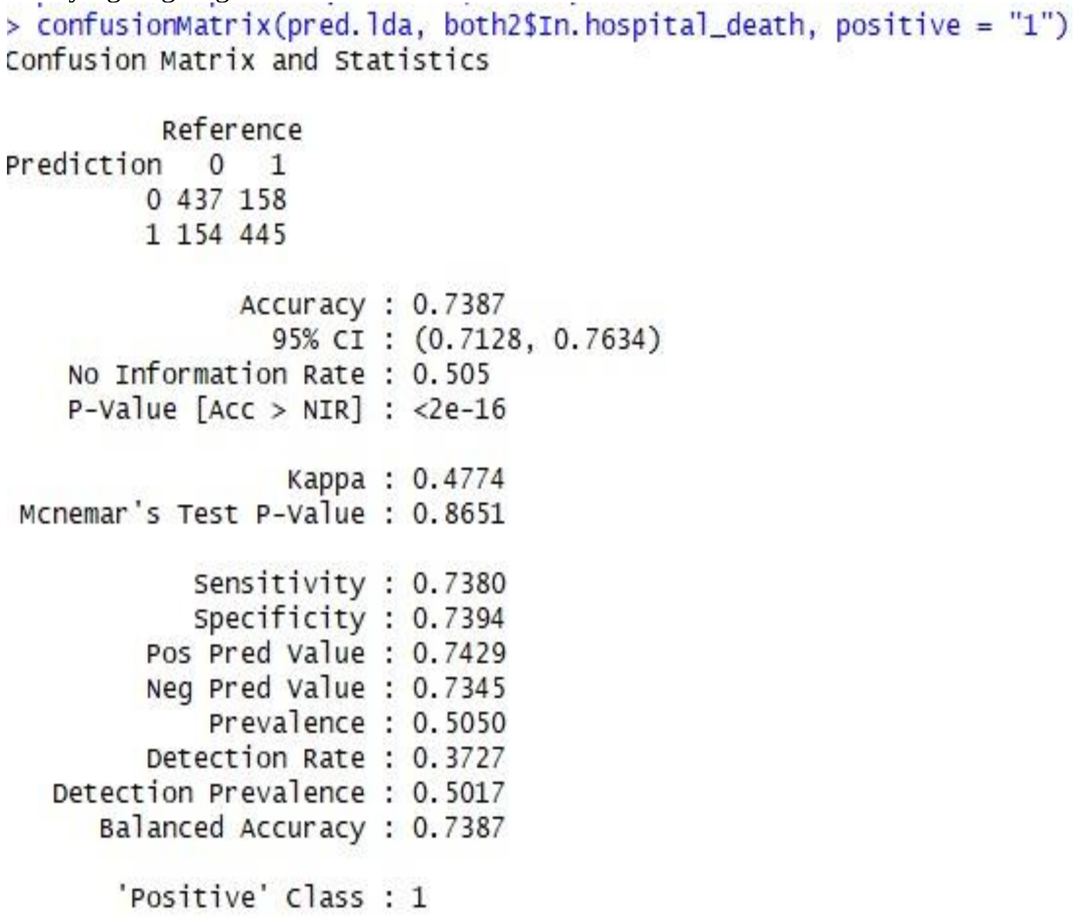
By performing the 5-fold cross validation, we could come up with better prediction of data which is proved by the results we have obtained in the three models (Logistic Regression, Support Vector Machining and Linear Discriminant Analysis)

# 5. Results

## 5.1. Linear Discriminant Analysis :

**Linear discriminant analysis** (**LDA**) or **discriminant function analysis** is a generalization of **Fisher's linear discriminant**, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification. LDA is closely related to analysis of variance (ANOVA) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. However, ANOVA uses categorical independent variables and a continuous dependent variable, whereas discriminant analysis has continuous independent variables and a categorical dependent variable (*i.e.* the class label).[3] Logistic regression and binary classification model are more similar to LDA than ANOVA is, as they also explain a categorical variable by the values of continuous independent variables. These other methods are preferable in applications where it is not reasonable to assume that the independent variables are normally distributed, which is a fundamental assumption of the LDA method.

Here from the confusion matrix we can see that 437 out of 591 patients were accurately predicted as survived and 445 out of 603 patients were accurately predicted as died. The accuracy is coming just less than 73%. Also, we can see that the sensitivity is less than the specificity which we don't want. Since, we are not getting higher sensitivity than the specificity we are not going to settle for this classification model and we are trying Log Regression.



```
> confusionMatrix(pred.lda, both2$In.hospital_death, positive = "1")
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 437 158
         1 154 445

               Accuracy : 0.7387
                 95% CI : (0.7128, 0.7634)
    No Information Rate : 0.505
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.4774
 Mcnemar's Test P-Value : 0.8651

            Sensitivity : 0.7380
            Specificity : 0.7394
         Pos Pred Value : 0.7429
         Neg Pred Value : 0.7345
             Prevalence : 0.5050
         Detection Rate : 0.3727
   Detection Prevalence : 0.5017
      Balanced Accuracy : 0.7387

       'Positive' Class : 1
```

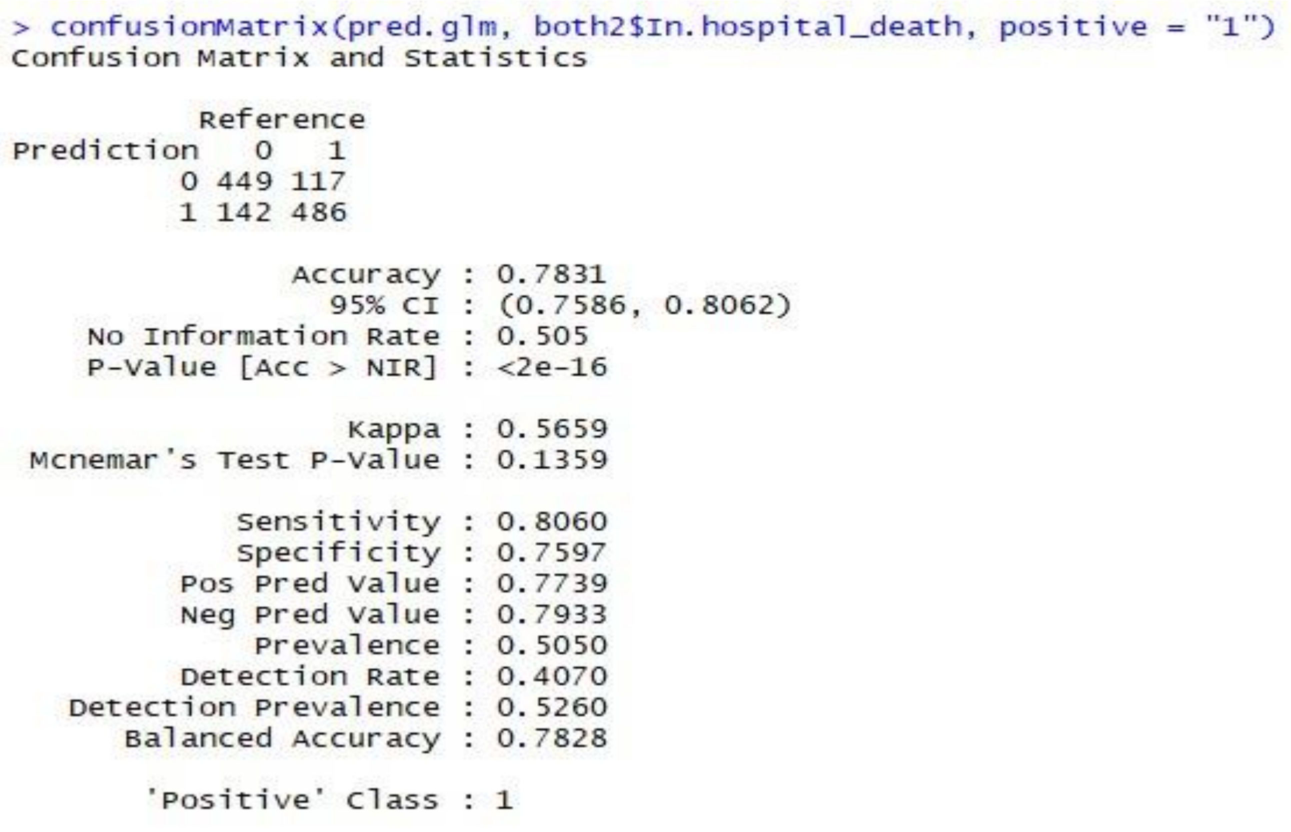
**Figure 5.2.1: Confusion Matrix of LDA Model**

### 5.2. Logistic Regression

Logistic Regression is a model where it deals with categorical variables which are dependent onto each other. The results from this model can be binomial or multinomial but for our dataset prediction model we need a binomial result i.e.  0- survived and 1 -dead, which is used to estimate the probability of a binary model based on the total 28 predictors. The model is a direct probability model and not a classifier.
The logistic function is defined as:

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

Where t can take any real value of the parameters and the outcome can take value between 0 or 1. Here from the confusion matrix we can see that 449 patients were accurately predicted as survived and 486 patients were accurately predicted as died. The accuracy is coming just more than 78% which is better than the linear discriminant analysis. Also, we can see that the sensitivity is more than the specificity which we wanted. So, between LDA and Log Regression we have better accuracy as well as better sensitivity for Log Regression.



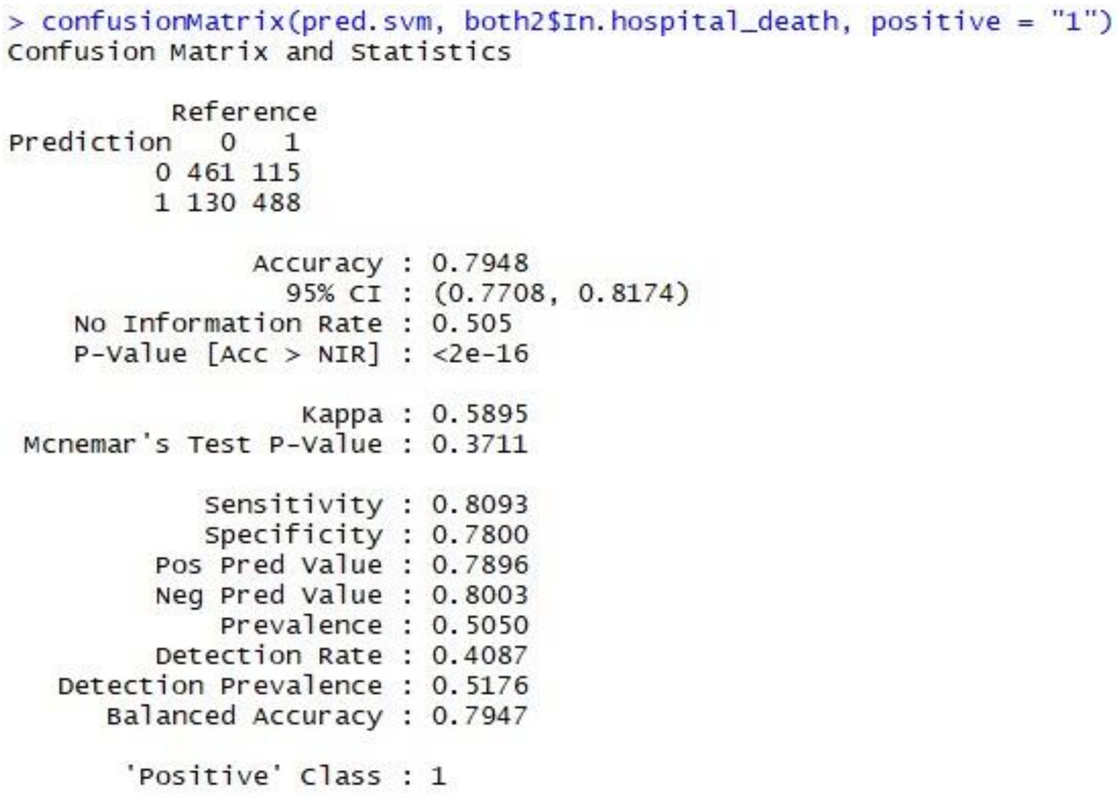
```
> confusionMatrix(pred.glm, both2$In.hospital_death, positive = "1")
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 449 117
         1 142 486

               Accuracy : 0.7831
                 95% CI : (0.7586, 0.8062)
    No Information Rate : 0.505
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.5659
 Mcnemar's Test P-Value : 0.1359

            Sensitivity : 0.8060
            Specificity : 0.7597
         Pos Pred Value : 0.7739
         Neg Pred Value : 0.7933
             Prevalence : 0.5050
         Detection Rate : 0.4070
   Detection Prevalence : 0.5260
      Balanced Accuracy : 0.7828

       'Positive' Class : 1
```

**Figure 5.1.1: Confusion Matrix of Logistic Regression Model**

### 5.3. Support Vector Machine

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Here from the confusion matrix we can see that 461 patients were accurately predicted as survived and 488 patients were accurately predicted as died. The accuracy is coming just more than 79% which is better than the las model ie. Log Regression. Also, we can see that the sensitivity and specificity are better than the Log Regression which we wanted. So, between LDA, Log Regression and Support Vector Machine we have better accuracy as well as better sensitivity for Support Vector Machine.

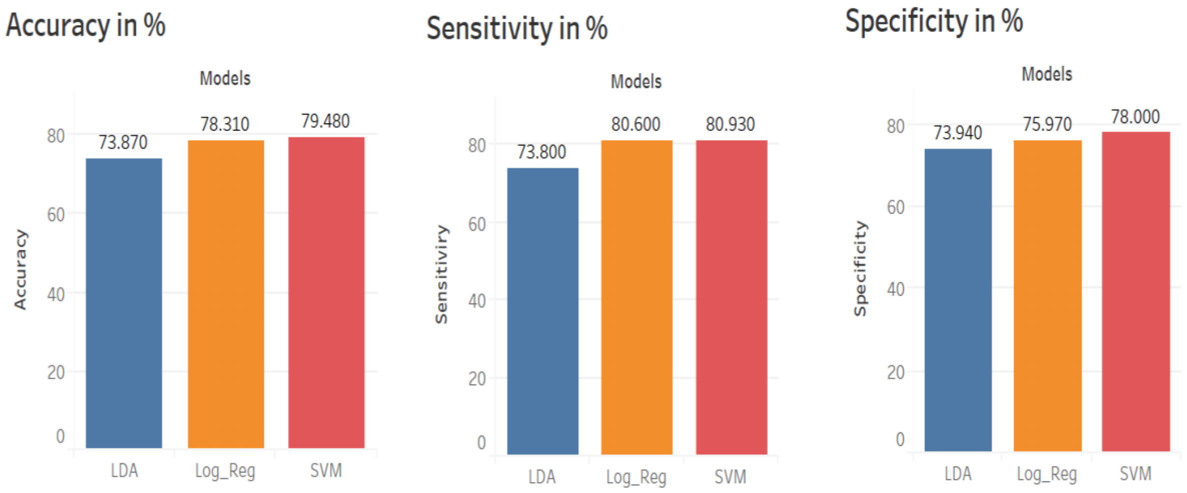


```
> confusionMatrix(pred.svm, both2$In.hospital_death, positive = "1")
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 461 115
         1 130 488

               Accuracy : 0.7948
                 95% CI : (0.7708, 0.8174)
    No Information Rate : 0.505
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.5895
 Mcnemar's Test P-Value : 0.3711

            Sensitivity : 0.8093
            Specificity : 0.7800
         Pos Pred Value : 0.7896
         Neg Pred Value : 0.8003
             Prevalence : 0.5050
         Detection Rate : 0.4087
   Detection Prevalence : 0.5176
      Balanced Accuracy : 0.7947

       'Positive' Class : 1
```

**Figure 5.3.1: Confusion Matrix of Support Vector Machine**

## 6. Conclusion:

Our model captures the direct effect of vital health parameters on the output. We do not investigate more on other parameters which would require additional data. However, we may miss the effect of parameters such as SAPS, APACHE and SOFA which further alter the prediction and outcome of the patients.



Although all the three models gave similar results. we came to a conclusion that Support Vector Machine is the most apt model to predict the patient's mortality which has a high sensitivity compared to the other models

\

## References

1. Predicting In-Hospital Mortality of ICU Patients: The PhysioNet/Computing in Cardiology Paper
2. SaeedM,VillarroelM,ReisnerAT,CliffordG,LehmanLW, Moody G, Heldt T, Kyaw TH, Moody B, Mark RG. Multipa- rameter Intelligent Monitoring in Intensive Care II (MIMIC- II): A public-access intensive care unit database. Critical Care Medicine 2011;39(5):10.
3. Hosmer D, Lemeshow S. Applied Logistic Regression. Wi- ley, 2000.
4. Linear Discriminant Analysis in R: An Introduction: https://www.r-bloggers.com/linear-discriminant-analysis-in-r-an-introduction
5. Logistic Regression: http://r-statistics.co/Logistic-Regression-With-R.html
6. Support Vector Machine: https://stackoverflow.com/questions/34529119/how-to-implement-support-vector-machine-in-r
7. Definition’s: https://www.wikipedia.org