PATIENT SURVIVAL PREDICTION

PROJECT REPORT

Group 46

Rohith Kanakagiri - (469)975-2669

Anurag Palanki - (617)870-1921

kanakagiri.r@northeastern.edu

palanki.a@northeastern.edu

Percentage of Effort Contributed by Student 1: 50%

Percentage of Effort Contributed by Student 2: 50%

Signature of student 1: Anurag Palanki

Signature of student 2: Rohith Kanakagiri

Submission Date: April 25, 2022

INDEX

Problem Setting:

There are various factors that affect the chances of a patient surviving when admitted to the hospital for treatment. The extent to which hospitals vary in survival outcomes and the degree to which this variation is explained to patients and hospital factors is unknown. The use of a model to factor in all the predictors to evaluate the survival of a patient would be useful.

Problem Description:

The goal of this project is to select the best classification model and filter the most important factors that go into the model to predict the in-hospital mortality among admitted patients using past data. We plan to see the effect other ailments, such as diabetes & bp, have on the classification model and gather information on any other factors that may influence the outcome.

Data Source:

Patient survival prediction dataset is acquired from Kaggle:

https://www.kaggle.com/mitishaagarwal/patient

Dataset Description:

In this dataset, there are 84 factors, given for 91,713 patients, which are involved when a patient is hospitalized. The file contains factors like :

Age: age of patient at admission

Bmi : body mass index of patient

Ethnicity: determines which culture does person belongs to

Gender: gender of patient

Height: height of patient at the time of admission

Weight : weight of the person

The minimum and maximum systolic and diastolic blood pressure measured using invasive and non-invasively, the minimum and maximum heart rate, respiratory rate and blood oxygen saturation rate, potassium, glucose all measured during the first 1 hour after admission and also over the period of 24 hrs after admission.

One of the most common systems is the 2nd version of the Acute Physiologic Assessment and Chronic Health Evaluation (APACHE) score. It generates a point score ranging from 0 to 71 based on 12 Physiologic variables, age and underlying health conditions.

The APACHE scores for the patient for various factors has been given like the Apache 2j, 3 diagnosis scores, Apache heart, incubated, map, temperature, GCS verbal, motor, and eyes scores are also included in the dataset.

Finally, the final dataset contain categorical variables aids, diabetes, cirrhosis, hepatic failure, leukemia, lymphoma, solid tumor carcinoma, and weather the patient was given an immunosuppressants in the last 6 months.

Data Pre-Processing:

Patient survival prediction consists of 85 columns with 91713 rows. Column "Unnamed:83" is removed as it consists only of null values. We can see a few redundant variables like 'encounter_id', 'patient_id', 'hospital_id', 'ice_admit_source','icu_id', 'icu_stay_type' and 'icu_type' which are not constructive in determining the final outcome of whether a patient will survive or not, hence we decided to drop them. Out of the remaining 77 variables, 55 are numerical and 22 are categorical variables. All the numerical variables contain some degree of null values but only BMI, HEIGHT and

WEIGHT do not require domain knowledge to manipulate. We replaced missing bmi, height and weight data points with the mean of the other patients whose gender and age match with the patients whose data is missing. We then dropped rows with more than 8 null values in it since imputation of too many variables is not desirable. After doing all the above operation we ended up with 77 columns with 85996 records.

Data Exploration:

For performing univariate and bivariate analysis, we considered mainly the numerical variables and a few categorical variable like the death of a patient, the patients gender and the APACHE 3j body-system score. The other variables were not considered since they are mostly focusing on auxiliary conditions of the patient. When viewed individually they do not make a huge difference but together on a whole make a difference.

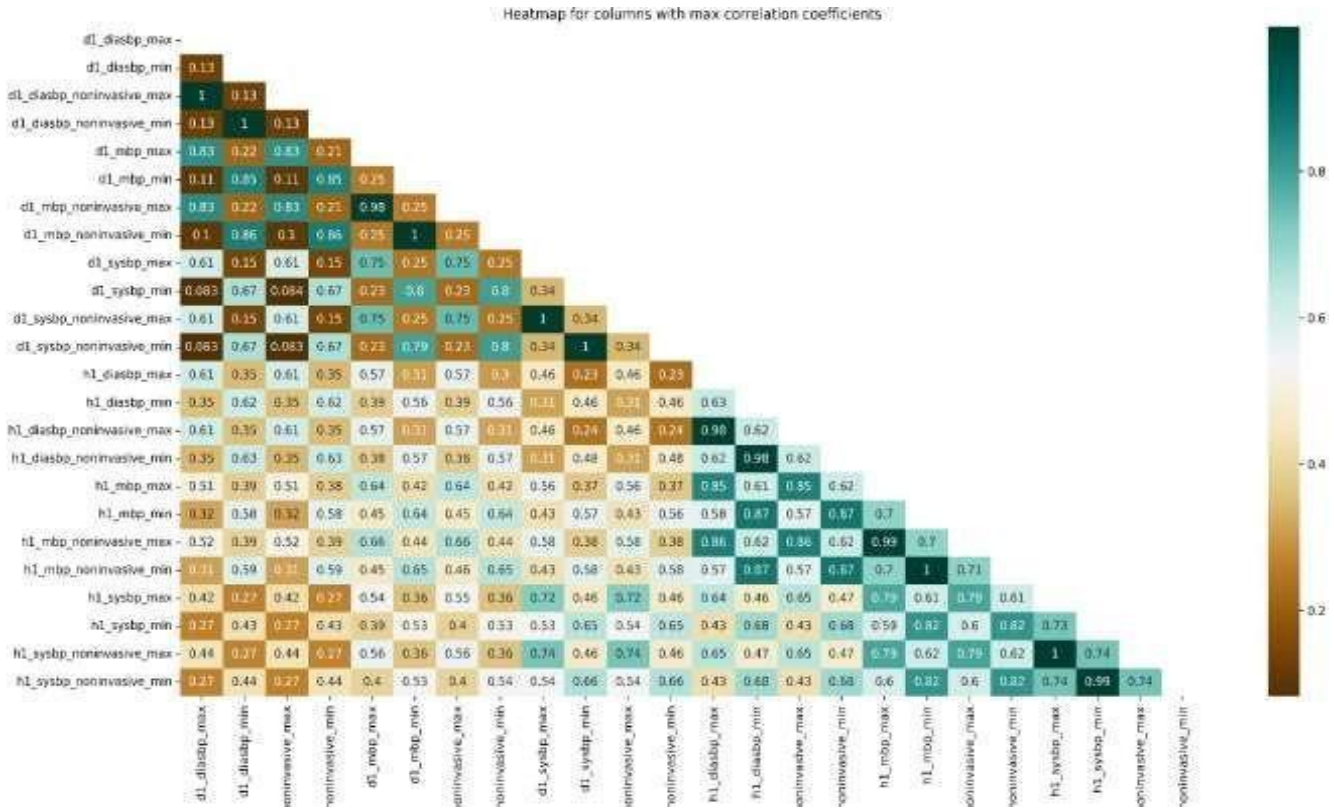


Fig.1 Correlation matrix of Numerical Variables with more than 0.8 correlation to another variable

As expected, the blood pressure columns are the columns with the most correlation between them since there are multiple ways to measure a person's blood pressure but ultimately, they all yield similar results. All of the columns in the heatmap with more than 0.8 correlation coefficient belong to a bp measurement. We may select a widely accepted measurement for our analysis and exclude the rest.

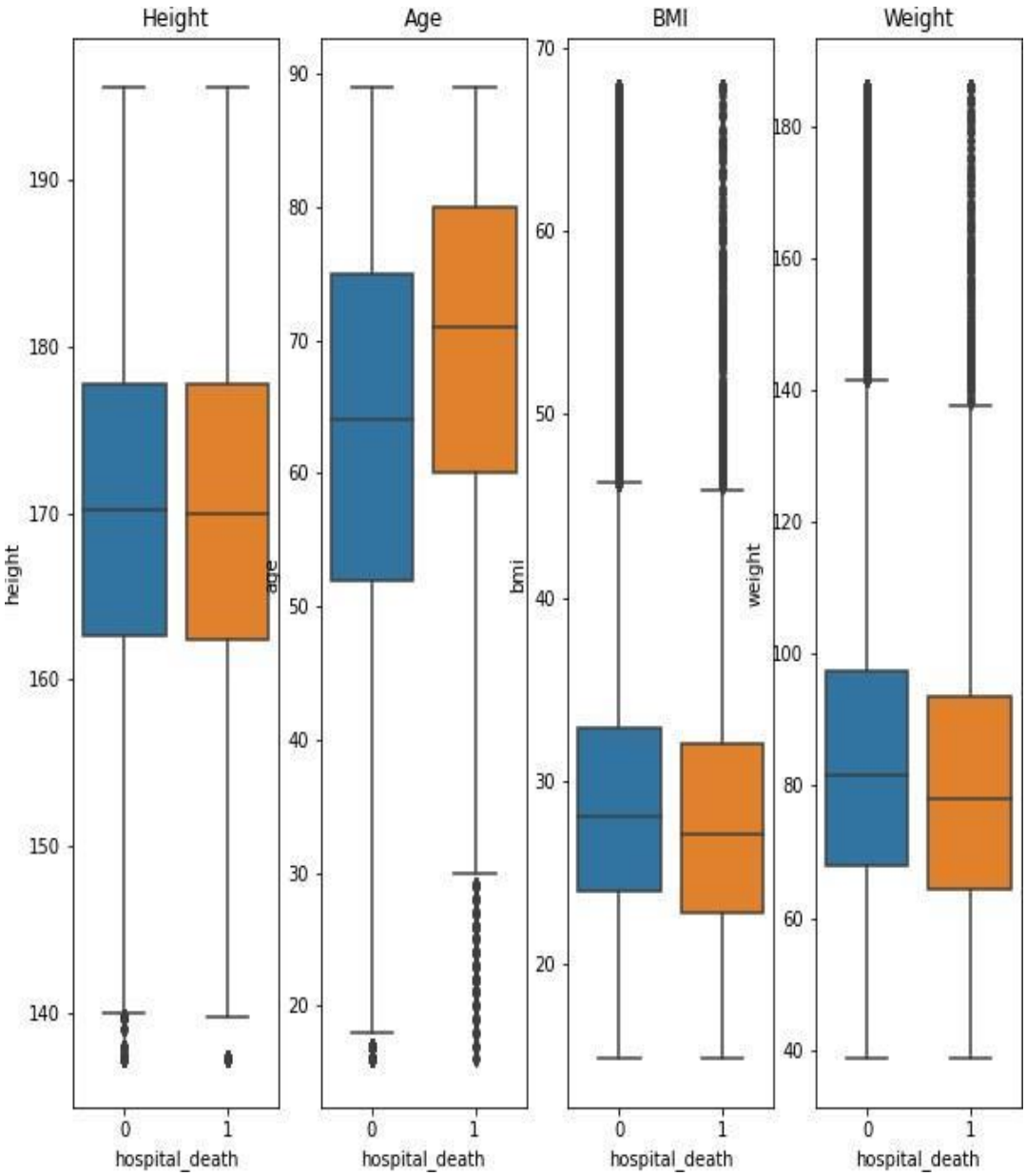


Fig.2 Box Plot of height, weight , age and bmi of survivors and non survivors

Next, we considered the most well-known factors like age, height, weight and bmi on the effect it has a patient's health and ultimately the survival outcome.

The height of patients in case of both survivors and non survivors is about same so the effect of the height of a person doesn't have an impact on the survival outcome. With respect to age of a patient, as expected the older a patient is the more the chances of the patient not surviving when struck with any ailments.

Surprisingly, when it comes to weight and body mass index, the opposite of what is expected is seen, the median weight and bmi of non survivors is lower than the survivors.

Considering the gender distribution for non survivors, blue represents males and red represents female in the plot.

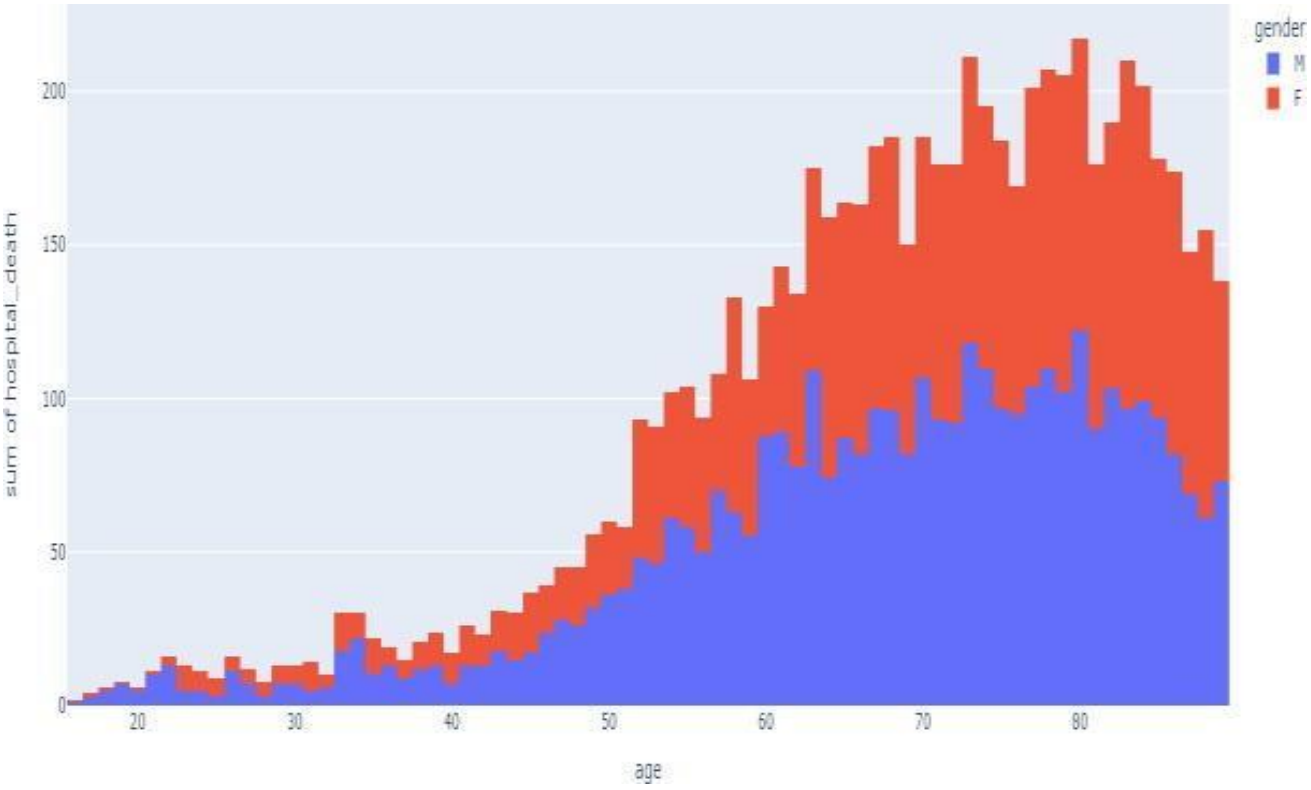


Fig.3 Gender Distribution plot with age

Although the distribution of males and females in the data is uniform (about a 54:46) split, the above density plot shows that females admitted to the icu are higher and they had also succumbed to their aliments more frequently than men at all ages and especially more so at a age of more than 52.

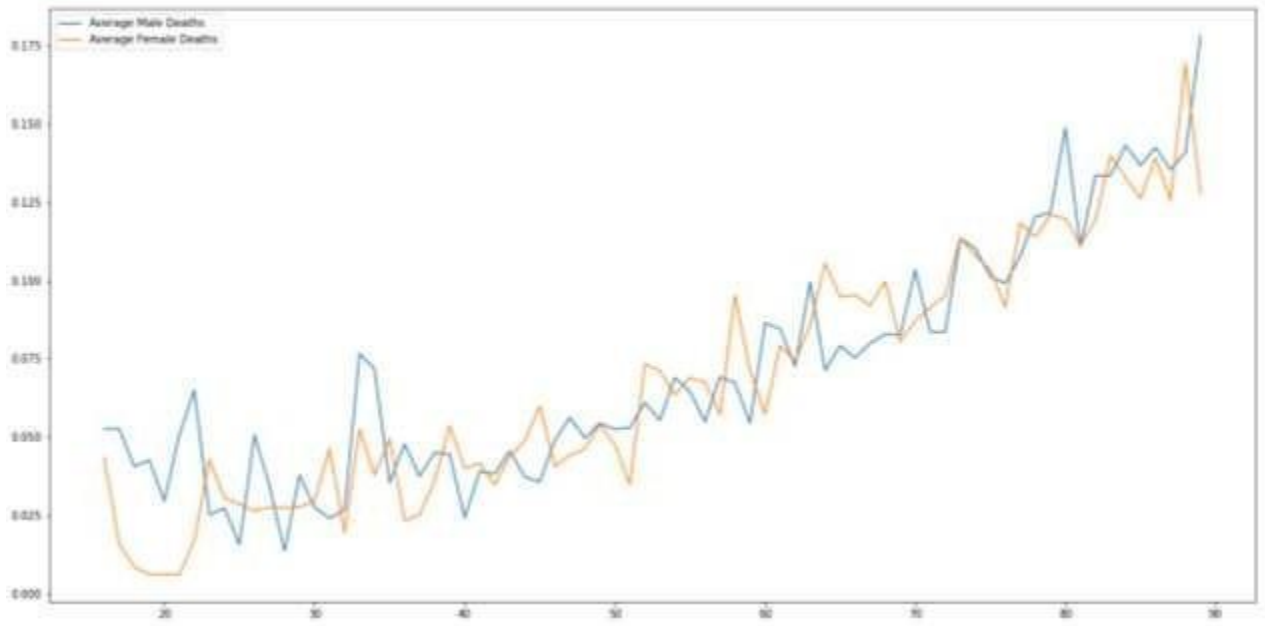


Fig.4 Avg females and males death with Age

The density is more the average number of deaths per age group for males and females is similar.

The last factor considered is the APACHE 3J Bodysystem which depicts which type of problem the patient is admitted with. It consisted of 11 distinct types and we can see that depending on the type of problem area the distribution of patient varies.

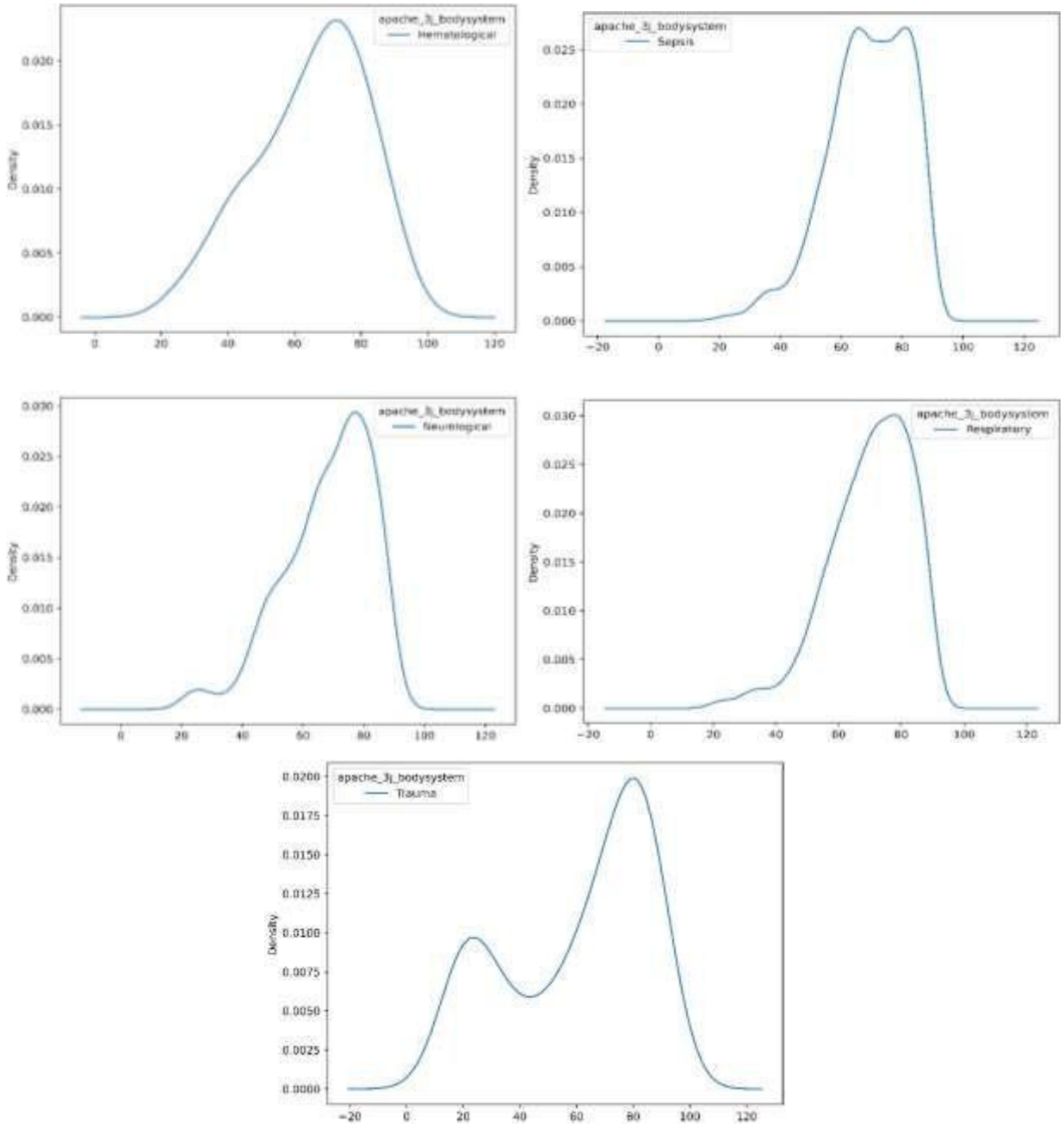


Fig.5 Line density plots for APACHE 3j Body system.

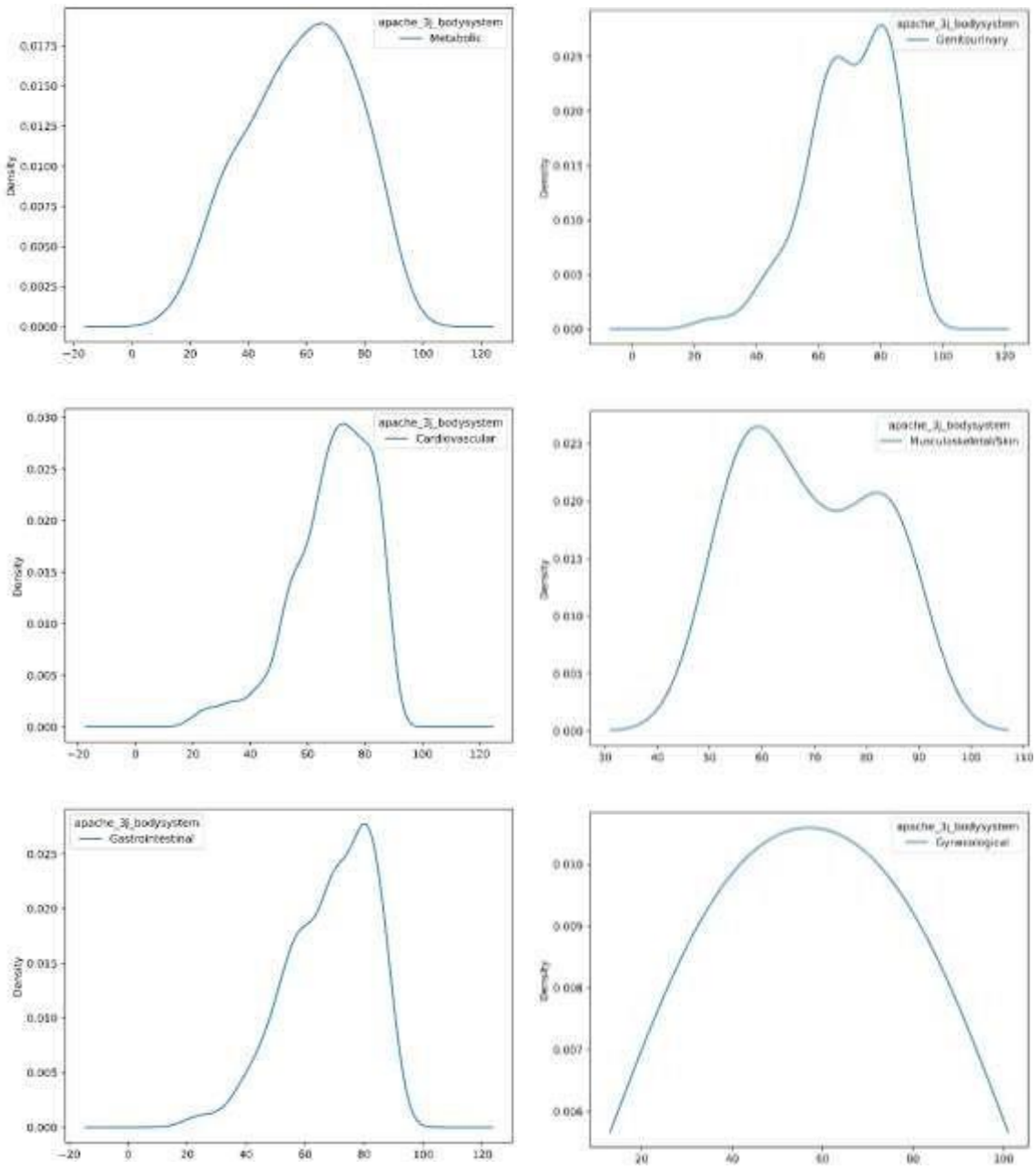


Fig.5. condt. Line density plots for APACHE 3j Body system.

The most anomalies are found in the case of musculoskeleton/skin, gynecological, metabolic and trama cases of apache 3j body system.

Data Mining Tasks:

For implementing data mining models, we first need to remove the null values. But medical conditions cannot be easily replaced with the mean or mode of the data so we imputed the missing values using KNN Imputer after running KNN Classifier to determine the best k value to use for the imputer. Next, to implement distance-based models, we removed the columns that are highly correlated with each other linearly. We ended up dropping 23 columns to do this and implement regression models, not removing them would've caused collinearity problems.

We split both the feature and outcome data into testing data and training data in a 1:3 split (25% testing data and 75% training data). The split data has 68784 rows in the training set and 22929 rows in the testing set. After splitting the data, we encoded the categorical variables (Ethnicity, Gender, Apache_3j_bodysystem, Apache_2_bodysystem) which contained text data while the other variables were already in numerical form. Encoding the data brought the total number of columns to 79 from 54. After encoding we normalized the training and test data based on the mean and standard deviation of the training data. Since the data only has about 8.6% of patients who don't survive, we decided to oversample the minority class in the training set to balance the data. To oversample the data we used Synthetic minority oversampling technique over-sampler, which oversamples the data while creating some variety in the data unlike random oversampler which reproduces the same data increasing the chances of overfitting. After oversampling the minority class the total number of rows in the data ended up being 117878, which is a very large number. Due to this reason, we avoided using distance- based models like KNN classifier, which would take very long to run.

Data Mining Models:

We chose to implement Logistic Regression Classifier, Gaussian Naïve Bayes Classifier, Gradient Boosting Classifier, Neural Network. We chose these models since they work very well with large

datasets and are widely used for classification problems. The features include all the variables except the hospital death column which is the target variable y.

1. **Logistic regression Classifier**:

   It's a form of statistical software that analyzes the association between a dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation. This type of analysis can help you anticipate the likelihood of an event or a choice occurring.

   Implementation:

   The base logistic regression model was executed on data which was scaled, oversampled and encoded. Solver 'lbfgs', parameters 'l2', class weight = 'balanced' and max iteration = 1000, were used as the ideal parameters for training the model.
   The resulting model showed an accuracy of 92% but a specificity of 0.19 and sensitivity of 0.985, Clearly the model is biased towards the majority class. 91.4% of the data is in the same majority class so an accuracy of 92% is not a good result.

2. **Naïve Bayes Classifier:**

   Naive Bayes is a straightforward technique for creating classifiers, which are models that give class labels to problem cases represented as vectors of feature values, with the class labels selected from a limited set. For training such classifiers, there is no one algorithm, but rather a variety of algorithms based on the same principle: all naive Bayes classifiers assume that the value of one feature is independent of the value of any other feature, given the class variable.

   Implementation:

   The Gaussian Naive Bayes classifier model was executed, resulting in an accuracy of 80%

3. **Gradient Boosting Classifier:**

Gradient boosting is a machine learning approach that may be used for a variety of applications, including regression and classification. It returns a prediction model in the form of an ensemble of weak prediction models, most often decision trees. The resultant approach is called gradient-boosted trees when a decision tree is the weak learner; it generally beats random forest. A gradient-boosted trees model is constructed in the same stage-wise manner as other boosting approaches, but it differs in that it allows optimization of any differentiable loss function.

Implementation:

The Gradient Boosting classifier model was executed with criterion as squared mean and the minimum number of samples per leaf as 10000 (to avoid overfitting) , resulting in an accuracy of 82 %.

4. **Neural Networks:**

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Implementation:

The Neural networks model was executed, resulting in an accuracy of 84%. Activator ‘relu’ and solver ‘adam’ were used as the ideal parameters for training data.

Performance evaluation:

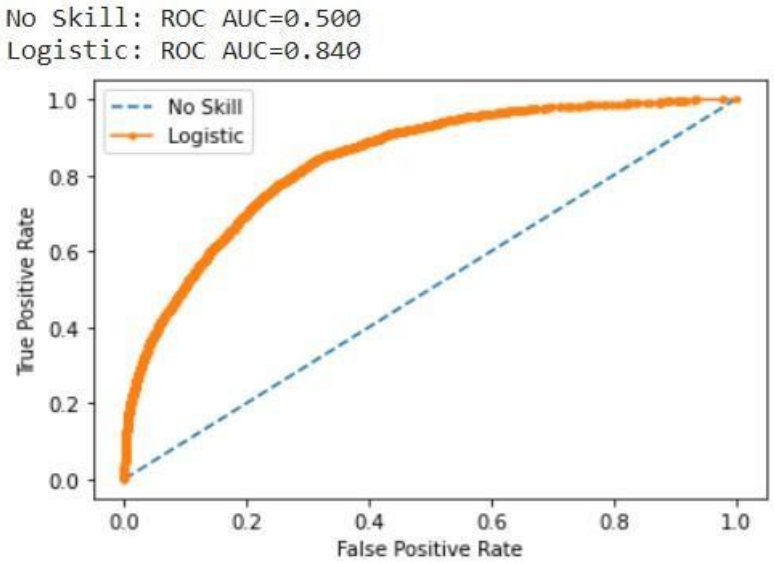
1. Logistic Regression:

Grid Search was done to find the parameter that gave the best result on the validation set. It was found that for 'solver'= 'lbfgs', 'max_iter'= 1000, and 'class_weight' = 'balanced' gave the best result. Using oversampled data the accuracy obtained was 92% but the specificity for the model trained on the oversampled data was found to be 19% while the sensitivity was 98.5%. The sensitivity shows the accuracy of the model in predicting the chance of a patient surviving and depicts a high true positive value but the low specificity shows that the model is not able to predict the likelihood of a patient succumbing to their ailments in the hospital. The Area under the curve for logistic regression is not high with only a value of 0.84 so it is clearly not that great at predicting the outcome. The f1 score is also similarly not high for class -1 which is the case where the patient doesn't survive. Logistic regression assigns each row a probability of belonging to a class and classifies data by applying a default threshold of 0.5. We altered the threshold to check and improve the outcome of the model. We used the f1 score and the point where the sensitivity and specificity of the model were equal to tune the model resulting in an accuracy of 81.11% and 75.6% respectively. Considering the imbalanced nature of the dataset this outcome is better than the previous result as both sensitivity and specificity are balanced.

Confusion Matrix	Predicted (Will Survive)	Predicted (won't survive)
Actual (Survived)	17324	3626
Actual (Didn't Survive)	691	1288

	precision	recall	f1-score	support
0	0.93	0.98	0.96	20950
1	0.53	0.28	0.36	1979
accuracy			0.92	22929

ROC AUC Curve for Logistic Regression



2. **<u>Naïve Bayes Classifier:</u>**

The default parameters worked best for the Gaussian Naïve Bayes Classifier in our case. Using encoded non-oversampled data, the accuracy obtained was 85%, while lower than the accuracy obtained by logistic regression, the specificity for the model drastically increased to 57% while the sensitivity dropped to 88%. The sensitivity shows the accuracy of the model in predicting the chance of a patient surviving and depicts a high true positive value but the relatively low specificity shows that the model is not able to predict the likelihood of a patient succumbing to their ailments in the hospital. The Area under the curve for Naïve Bayes Classifier is not higher

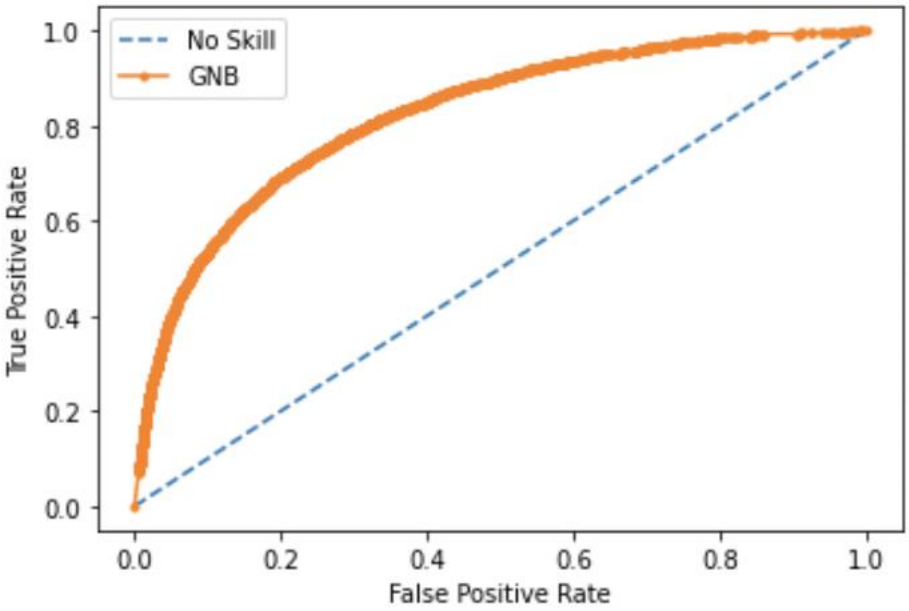
than that of logistic regression with only a value of 0.824. The f1 score is 0.4 for class -1 which is the case where the patient doesn't survive.

Confusion Matrix	Predicted (Will Survive)	Predicted (won't survive)
Actual (Survived)	18434	2516
Actual (Didn't Survive)	850	1129

	precision	recall	f1-score	support
0	0.96	0.88	0.92	20950
1	0.31	0.57	0.40	1979
accuracy			0.85	22929

ROC AUC Curve for Naïve Bayes Classifier

```
No Skill: ROC AUC=0.500
GNB: ROC AUC=0.824
```



3. **<u>Gradient Boosting:</u>**

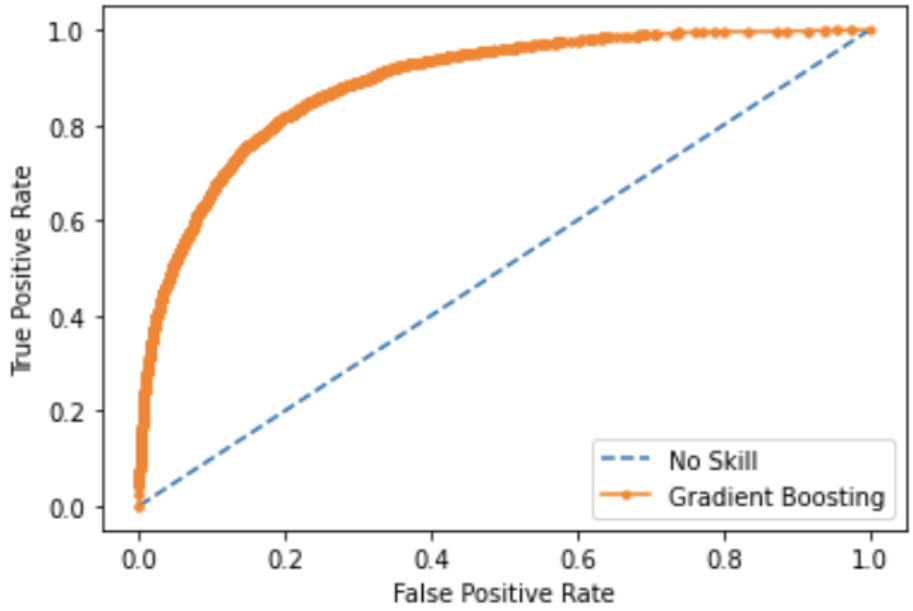
For implementing Radom Forest with gradient boosting the best result was obtained by limiting the minimum samples per leaf to 10000 (which is about 10% of the oversampled data that was input into the model). Using oversampled data, the accuracy obtained was 81%. The specificity for the model is 79% while the sensitivity was found to be 81%. The sensitivity and specificity are nearly equal in the case of random forest model and they are but close to 80%. This is better than the other models in which the majority class was more accurately predicted while the minority class was not. The Area under the curve for is 0.89 which is very close to 1 and it is the highest so far. The f1 score is 0.43 for class -1 which is the case where the patient does not survive and the f1 score for the surviving class is 0.89.

Confusion Matrix	Predicted (Will Survive)	Predicted (won't survive)
Actual (Survived)	17103	3847
Actual (Didn't Survive)	402	1577

	precision	recall	f1-score	support
0	0.98	0.82	0.89	20950
1	0.29	0.80	0.43	1979
accuracy			0.81	22929

ROC AUC Curve for Gradient Boosting (Radom Forest)

```
No Skill: ROC AUC=0.500
Gradient Boosting: ROC AUC=0.890
```



4. **Neural Network:**

The grid search cross validation was used to find the best parameters, which were found to be 'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (2), 'learning_rate': 'adaptive', and 'solver': 'adam'. Using oversampled data, and running the neural network using the above mentioned parameters the accuracy obtained was 82%. The specificity for the model is 76% while the sensitivity was found to be 82.3%. The sensitivity and specificity are again nearly equal like in the case of gradient boosting model. The f1 score is 0.42 for class -1 which is the case where the patient does not survive and the f1 score for the surviving class is 0.89, similar to gradient boosted model.

Confusion Matrix	Predicted (Will Survive)	Predicted (won't survive)
Actual (Survived)	17234	3716
Actual (Didn't Survive)	476	1503

	precision	recall	f1-score	support
0	0.97	0.82	0.89	20950
1	0.29	0.76	0.42	1979
accuracy			0.82	22929

Project Results

After evaluating and tuning the 4 models, the results show that the Gradient Boosting model worked best for the given data. The reason we selected this as the best model for the data is because the model has a higher accuracy in predicting the patients who do not survive without classifying all or a most of the data into the majority class thus improving the accuracy but compromising on the predictive power of the model. Other models showed a higher overall accuracy but took a hit in predicting whether a patient would survive.

Model/Metrics	**Accuracy**	**Sensitivity**	**Specificity**	**F1(0)**	**F1(1)**
Logistic regression	81	82	75	96	37
Naïve Bayes	85	88	57	92	40

Gradient Boosting	81	81	79	89	43
Neural Network	82	82	76	89	42

While the accuracy of the logistic regression model was initially higher at 92%, 91.6% of the data belong to a single class (0 which is the case of a patient surviving). So even classifying all the cases as 0 would lead to a 91.6% accuracy so an accuracy of 92% is not appealing in this case. The gradient boosting classifier on the other hand predicts 80% of the cases accurately. Although it also isn't better than the 91.6% obtained in a majority classification case the sensitivity and specificity of the outcome are balanced at 0.81 and 0.79 respectively and unlike the balanced logistic regression model which has a sensitivity and specificity of 75.6%. Class 0's score was compromised in return for a greater accuracy in obtaining a more accurate Class 1 prediction. In conclusion, gradient boosting classifier was found to be best in predicting the survival of a patient.

Impact of Project Outcomes

The goal of the project was to implement a model which would help predict the outcome of a patient surviving after having been to the ICU. Numerous parameters (80) were taken into consideration to build a model to help achieve this and it was observed that most patients make it in the end but about 8.6% of the patients do not. The column hospital_death in the data signifies the survival of the patient. Being able to predict the mortality of a patient is very useful in assessing the severity of the illness and helps allocate resources and time to lessen the chances of succumbing to the ailments.

It was observed that the gradient boosting model based on regression trees worked best in doing this task due its relatively high f1 score and accuracy in both predicting the survivors and non survivors without any bias towards one.