Table of Contents

[1 Domain Description 1](#_Toc111668513)

[2 Problem Definition 1](#_Toc111668514)

[3 Literature Review 2](#_Toc111668515)

[4 Dataset Description 3](#_Toc111668516)

[5 Dataset Pre-processing 4](#_Toc111668517)

[6 Experiments 13](#_Toc111668518)

[6.1 Naïve Bayes 13](#_Toc111668519)

[6.2 Decision Tree 14](#_Toc111668520)

[6.3 Random Forest 15](#_Toc111668521)

[7 Analysis & Results 16](#_Toc111668522)

[7.1 Naïve Bayes 16](#_Toc111668523)

[7.2 Decision Tree 18](#_Toc111668524)

[7.3 Random Forest 20](#_Toc111668525)

[8 Conclusions 25](#_Toc111668526)

[References 25](#_Toc111668527)

[Appendix 29](#_Toc111668528)

[R Code 29](#_Toc111668529)

# 1 Domain Description

Many hospitals in the world have proposed various types of medical information systems to give patients the finest services and care. The information provided by big data is shifting the medicine world and has become a crucial component of modern medical research (Yang, et al., 2020).

Modern medical care with advanced technology such as Magnetic Resonance Imagining (MRI), Computed Tomography (CT), and ultrasound scans contribute to the increase of volume and complexity of the data. Data generated in health care grew exponentially from 153 exabytes to 2,314 exabytes between 2013 and 2020 (Stewart, 2020).

This study will apply the cutting-edge possibilities unleashed by machine learning to predict patient survival. This approach is taken, leveraging on the research done by (Jothi, et al., 2015), whose findings show that a greater majority of the data mining experiments conducted in the healthcare sector utilized machine learning techniques. Additionally, while some researchers have noted that algorithms such as Naïve Bayes, Decision Trees, K-nearest neighbours, Logistic Regression, and Support Vector method show better classification and clustering, others, including (Jothi, et al., 2015), argue that a hybrid model is critical in achieving a high accuracy result.

# 2 Problem Definition

In hospitals all around the world, patient mortality has been a significant problem. Many medical practitioners want to understand why patients die. A number of factors inhibit data gathering in the medical domain, including the introduction of new drugs, patients’ response to treatments, geographical restriction in data collection, etc. The sample data used for this research highlights these shortcomings, showing bias toward individuals of a particular race (Nanayakkara, 2018). Another major challenge is data cleaning; this is also highlighted in the sample dataset as several observations in the dataset either had NA values or were missing entirely. Previous researchers have tried to improve patient survival prediction. This research contributes to the effort made towards understanding why patients die in hospitals. Overall, the random forest algorithm produced a better patient survival prediction model with the highest accuracy.

# 3 Literature Review

In Patient Survival Prediction, the observer investigates survival chances, which is known as the time between the start of patient observation and the death of the monitored patient or the end of monitoring (Eldho, 2013). The goal is to forecast life expectancy even when the incident does not result in death by using predictive/explanatory variables from the available data. APACHE II and APACHE III used for predicting hospital mortality among critically ill patients are well-known scoring systems. APACHE II consists of data gathered during the first 24 hours of an ICU admission, which include demographics, vital signs, blood figures and lab test analysis and any history of diseases (William, 2022). The APACHE II score is mainly known as the most important benchmark for patient mortality prediction (Nanayakkara, 2018). Different methods have been successfully applied for patient survival prediction, however previous scholars have noted that in most cases, machine learning algorithms have shown better predictive performance compared to the APACHE II and APACHE III scoring systems (Eldho, 2013).

(Nanayakkara, 2018) used the APACHE II and III score attributes to predict hospital mortality rate in USA and Australia. Random forest and Decision Tree produced an accuracy of 82% and 80% respectively. The Analysis from the exploratory data analysis (EDA) showed that mostly Caucasian and African American patients have the highest mortality rate after cardiac arrest episode. (Eldho, 2013) made use of data mining techniques such as Random Forest, which resulted in a 90% accuracy and Naïve Bayes, an 87.4% accuracy. Other studies by (Ishleen Kaur, 2021) and (Montazeri, 2016) analyzed cancer survival prediction using Random Forest, Naïve Bayes and AdaBoost. Random forest performed better in both scenarios having 76% and 96% accuracy respectively, with Naïve bayes having 70% and 95% accuracy respectively. These statistics is particularly important in this context because these studies highlight the performance of Random Forest over other algorithms like Naïve Bayes and Decision tree. This high performance is credited to the nature of Random forests algorithm which performs well with large data, and trains faster than decision trees. Unlike Random Forest and Decision Tree algorithms, Naïve Bayes tends to perform better with categorical data (Eldho, 2013).

Generally, data mining classifiers work by grouping objects of a data set into multiple classes depending on its characteristics, which is mostly applied on datasets that are unbalanced (Ferdib-Al-Islam, 2021). Machine learning algorithms perform better when the distribution of the target variable in the dataset is balanced (Alexander Liu, 2012). However, in real world problems like fraud or bankruptcy detection or any form of medical classification prediction that mainly contain binary variables, the dataset is mostly imbalanced. (Nanayakkara, 2018) performed oversampling technique to the Patient Survivability dataset to determine if resuscitated cardiac arrest is associated with high mortality. This allowed them to create accurate predictive models using data mining techniques. (Nanayakkara, 2018) also highlighted patterns of differences between the number of Caucasians and African Americans versus Asians and Hispanics. The classifications were imperfect for all races, the scores showed a consistent pattern of overpredicting mortality for African Americans and Caucasians.

Overall, most machine learning algorithms show bias in imbalanced datasets. If we do not account for imbalanced data, the proposed classifier might learn to only predict the majority class, and this should not be acceptable (Alexander Liu, 2012). Multiple papers have studied and proposed how to handle imbalanced data, most have leaned towards oversampling of the data set (Ferdib-Al-Islam, 2021), (Ishleen Kaur, 2021).

# 4 Dataset Description

The dataset was obtained from (Agarwal, 2022). It was created by MIT’s Global Open-Source Severity of Illness Score (GOSSIS) Consortium. This dataset uncovers the factors that determine the chances of survival in intensive care units (ICU). The data was collected over a one-year period from over 130, 000 ICU visits from a range of countries. The size of the sample dataset is 91,714. There are 86 feature variables with 34,778 missing values in the dataset. The target variable is hospital\_death and patients who died are characterized as 1, while the patient that did not die are characterized as 0.

Patients did not die in 91.4% of the observations, whereas in 8.6% of the observations, patients died. The patients that are captured in the dataset are all above the age of 16. The dataset contains 25,971 females with a mean age of 63, and 30,964 males with a mean age of 62. It is vital to highlight a significant drawback of this dataset, which is that a greater percentage of the patients in the dataset are Caucasians (Agarwal, 2022).

A description of the dataset is provided in the link below:

<https://github.com/notrichbish/PatientSurvivalPred/blob/main/Dataset%20Description.docx>

# 5 Dataset Pre-processing

The patient survival prediction dataset is categorized into patients who died and patients who survived. As can be seen from the dataset, 91.4% of the cases, patients survived, while in 8.6% of the cases, patients did die. Therefore, this dataset is imbalanced. Naïve Bayes, Decision Tree, and Random Forest may work with skewed data; nevertheless, training and testing data must reflect an exact distribution of the target variable which is patient survival. In this experiment, we carried out several basic data cleaning processes including removing missing values, identifying outliers, removing redundant columns (Patient ID, ICU ID, Hospital ID, and Encounter ID), replacing blank values with NA values, and removing all NA valued records within the dataset. The three algorithms discussed in this paper were implemented to perform oversampling to the minority class (1), until it reaches 52,044 (Gosain & Sardana, 2017).

A picture containing table

Description automatically generated

Figure 1, Identify Missing Values

While some researchers argue that missing values will negatively impact the accuracy of the machine learning algorithm (Angelov, 2017), others including (Acuña & Rodriguez, 2004) note that missing values ratios over 15% can impact the machine learning model. As shown in Figure 1 above, 69 variables contain missing values with “d1\_potassium\_min” having the highest count of missing values followed by “d1\_potassium\_max”, “h1\_mbp\_noninvasive\_min”, and so on.

Graphical user interface

Description automatically generated with medium confidence

Figure 2, Identify Outliers for every variable

Outliers are data points that are extremely high or extremely low in comparison to the nearest data points and the rest of the neighbouring co-existing values in a dataset (Ghosh & Vogt, 2012). In data pre-processing, outliers are identified through data visualisation, this could be implemented in a boxplot, scatterplot, line plot, and histogram. In this paper, we have identified the outliers by visualising a boxplot of the entire dataset as shown in Figure 2. This we did by identifying the data points that are beyond the boxplot. While previous scholars have justified the removal of outliers from dataset within the medical domain (Hauskrecht, et al., 2012) (Ijaz, et al., 2020), others argue that every data point within a medical dataset is crucial (Laurikkala, et al., 2000) (Cheng & Huang, 2021).

Chart, box and whisker chart

Description automatically generated

Figure 3, Outliers in “apache\_3j\_diagnosis”

Acute Physiology and Chronic Health Evaluation II (APACHE II) invented in 1985 have been used throughout the world by hospitals’ Intensive Care Units (ICU). The APACHE II system uses 12 physiological variables, including the Glasgow Coma Scale (GCS). Numerous experiments support the APACHE II's role in ICU patients. The new, improved APACHE III method was introduced in 1989 which incorporates physiological measures, chronic health measures, and illness classification more effectively and also examines how susceptible the measures are to different measurement biases (Cho & Wang, 1997). Figure 3 shows a boxplot of the “apache\_3j\_diagnosis” variable, with the outliers appearing above the boxplot.

Chart, box and whisker chart

Description automatically generated

Figure 4, Outliers in “Age”, “icu\_type”, “map\_apache”, “d1\_spo2\_max”, “h1\_mbp\_min”, and “aids”

Figure 4 shows the boxplots of 6 sample variable from our dataset, including “Age”, “icu\_type”, “map\_apache”, “d1\_spo2\_max”, “h1\_mbp\_min”, and “aids”. Outliers are identified in “d1\_spo2\_max” and “h1\_mbp\_min” variables.

Table

Description automatically generated

Figure 5, Cleaned missing values

Figure 5 shows a visualisation of the cleaned dataset. This was implemented by replacing the blank values with NA values subsequently removing all the records with NA values.

A picture containing text, receipt

Description automatically generated

Figure 6, Dataset Summary

Many variables in our dataset are of datatype numeric, as shown in Figure 6. The data summary also captures the minimum value, 1st quantile, median, mean, 3rd quantile, and maximum values of each variable in the dataset. All the variables are not normally distributed.

Chart

Description automatically generated

Figure 7, Apache II Body System based on Gender and Ethnicity

The APACHE II Body System variable is visualised based on the gender and ethnicity of the individuals in the dataset as shown in Figure 7. A near equal proportion of male and female in the dataset are diagnosed with most of the diseases. Majority of the ethnicities that are diagnosed with most of the diseases are Caucasians. This makes sense as most of the observations within the dataset belong to individuals who are Caucasians.

Chart, bar chart

Description automatically generated

Figure 8, APACHE III Body System based on Gender and Ethnicity

The APACHE III Body System variable is visualised based on the gender and ethnicity of the individuals in the dataset as shown in Figure 8. A near equal proportion of male and female in the dataset are diagnosed with all the diseases except Gynaecological where only females are diagnosed with it. Majority of the ethnicities that are diagnosed with all the diseases are Caucasians. This makes sense as most of the individuals within the dataset are Caucasians.

Graphical user interface, chart

Description automatically generated

Figure 9, Hospital Death based on Ethnicity and Gender

The target variable, hospital death, is visualised based on ethnicity and gender of the individuals in the dataset as shown in Figure 9. Most of the patients did not die, however, amongst those who died Caucasians are the majority. In terms of gender, an equal proportion of male and females died and did not die.

A picture containing graphical user interface

Description automatically generated

Figure 10, AIDS and Lymphoma based on Ethnicity

The variables AIDS and Lymphoma are visualised based on ethnicity as shown in Figure 10. None of the patients are diagnosed with AIDS while a small proportion of Caucasians are diagnosed with Lymphoma.

Graphical user interface

Description automatically generated with medium confidence

Figure 11, Leukaemia and Immunosuppression based on Ethnicity

The variables leukaemia and immunosuppression are visualised based on ethnicity as shown in Figure 11. A small proportion of Caucasians are diagnosed with Leukaemia and Immunosuppression, followed by African Americans.

Graphical user interface

Description automatically generated

Figure 12, Hepatic Failure and Diabetes Mellitus based on Ethnicity

The variables hepatic failure and diabetes mellitus are visualised based on ethnicity as shown in Figure 12. Majority of the patients that are diagnosed with Diabetes Mellitus are Caucasians followed by African American. While a small proportion of Caucasians are diagnosed with Hepatic Failure.

Chart

Description automatically generated

Figure 13, Cirrhosis based on Ethnicity

The variable cirrhosis is visualised based on ethnicity as shown in Figure 13. A small proportion of Caucasians are diagnosed with Cirrhosis.

Chart, pie chart

Description automatically generated

Figure 14, Distribution of gender and ethnicity (Pie Charts)

The distribution of gender and ethnicity are shown in Figure 14. The number of male patients are slightly more than the female patients. 80% of the ethnicity within the dataset are Caucasians followed by African American and other ethnicities.

Chart, scatter chart

Description automatically generated

Figure 15, Variables with the highest correlation

The variables with the most significant relationship are shown in Figure 15. The highest correlation exists between “d1\_diasbp\_noninvasive\_max” and “d1\_diasbp\_max” with 0. 9985 followed by the correlation between “d1\_diasbp\_noninvasive\_min” and “d1\_diasbp\_min” with 0.9984.

Engineering drawing

Description automatically generated

Figure 16, Variables with the lowest correlation

The variables with the least significant relationship are shown in Figure 16. The lowest correlation exists between “gcs\_verbal\_apache” and “h1\_mbp\_noninvasive\_max” with -0.0009792 followed by the correlation between “gcs\_verbal\_apache” and “h1\_mbp\_max” with -0.001454.

# 6 Experiments

In this study, the machine learning algorithms that are implemented are based on the literature review. Naïve Bayes, Random Forest, and Decision Tree were reported by the reviewed papers as the high predictive classifiers.

## 6.1 Naïve Bayes

Naïve Bayesian classifiers are fast-supervised classifiers which are capable of handling large-scale classification and prediction tasks. This classifier operates on two main assumptions, including that the numeric attribute values are normally distributed within each class and that the impact of feature values on a specific class is distinct of the values of the other attributes (Leung, 2007). The classifier predicts that a sample X corresponds to a group with the most probability.

Many enhancements to the fundamental naïve Bayes algorithm have been suggested to help offset its principal weakness, which is the presumption that variables in a dataset are independent of each other. (Mark, 2006) presents a straightforward filter approach for modifying attribute weights to increase performance while maintaining model quality, while (Lee, 2007) argues that incorporating unlabeled training data enhances the performance of the models.

Naïve Bayes classifier has been widely used for prediction and classification within the medical domain (K.Vembandasamy, et al., 2015), (P.L.Geenena, et al., 2011), and (Parthiban, et al., 2011).

In this study, naïve bayes classifier was implemented on the sample dataset, after oversampling.

## 6.2 Decision Tree

Decision tree is a supervised machine learning algorithm that utilizes the concept of divide and conquer to examine huge databases for patterns and characteristics to classify and predict a certain class variable (Myles, et al., 2004). The classifier operates by breaking down the dataset into smaller subsets while simultaneously constructing a decision tree to show the predictions based on the set of feature-based splits. To ensure the best split in a decision tree, the algorithm uses entropy which determines the homogeneity of the sample (Quinlan, 1986).

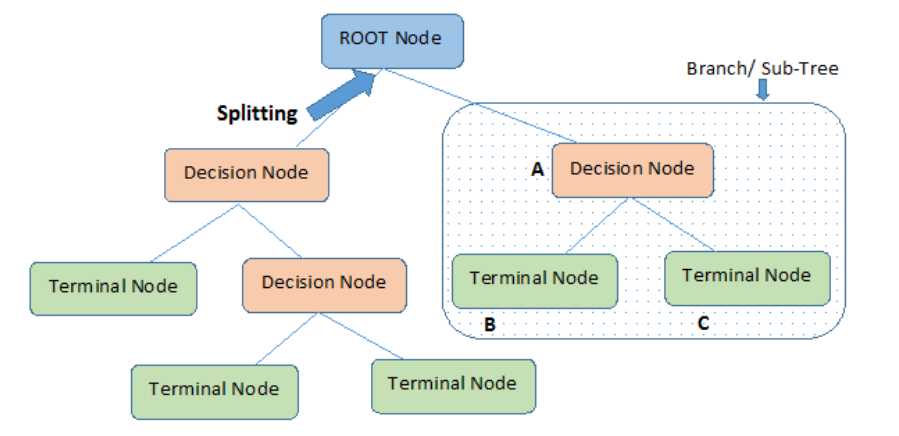


Figure 17, Sample of Decision Tree (Saini, 2021)

As seen in Figure 17, a decision tree starts with a root node and ends with a terminal node. The root node represents the start of decision tree where the splitting initializes, the decision node is the node produced after splitting the root nodes, and terminal/leaf node is the node where no further splitting is not possible (Saini, 2021).

Decision tree is a sensitive algorithm that could overfit the data during model training, as a slight change can alter the tree structure which could lead to a misclassification. To prevent this, pruning is used to improve the performance of the tree by removing the nodes which are not important (Krueger, 2021) (Patil, et al., 2010).

Some researchers have utilized decision tree to classify and predict within the healthcare domain (Mishra, et al., 2020), (D.Lavanya & Rani, 2011), and (Mirza, et al., 2018).

In this study, decision tree was implemented on the dataset to predict patient survivability upon oversampling.

## 6.3 Random Forest

Random Forest is a well-known supervised machine learning technique that utilizes multiple classification and regression tree that works as an ensemble (Biau & Scornet, 2016). Random forest is a collection of decision trees. Decision trees are very sensitive during the training stage, a small change can alter the tree structure significantly. Thus, random forest uses bootstrap aggregation (bagging) to randomly sample the dataset with replacement which creates different classifier trees as seen in Figure 18 (Yiu, 2019). Each decision tree will produce a class prediction and the model prediction will be chosen based on the class with the most votes (R, 2021).

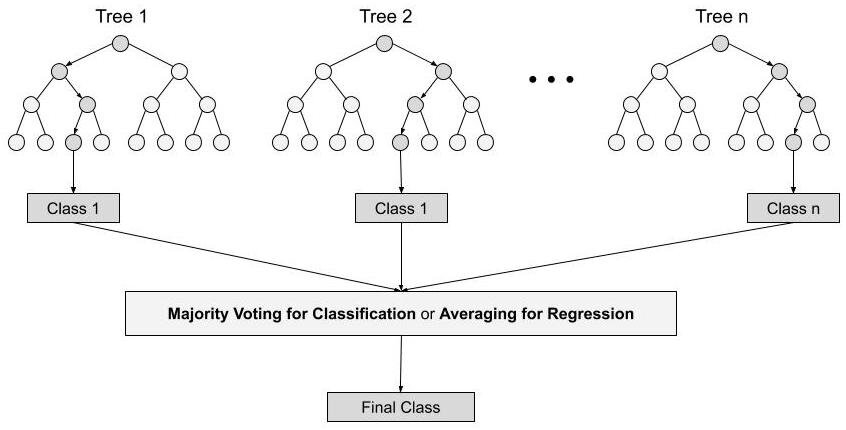


Figure 18, Random Forest Architecture (R, 2021)

Previous scholars have proposed many improvements to the Random Forest algorithm. (Liu, et al., 2017) proposes a hybrid algorithm between ReliefF algorithm for dimension trifunctionality paired with random forest for modelling and analysis of complex diseases, while (Xu, et al., 2020) propose a combination of SMOTE oversampling and Edited Nearest Neighbors (ENN) with random forest algorithm to classify medical datasets.

The use of random forest algorithm for prediction and classification within the medical domain was performed by many researchers including, (Gupta, et al., 2018), (Dai, et al., 2018), and (Kabiraj, et al., 2020).

In this paper, the random forest algorithm is applied to the patient survival prediction dataset.

# 7 Analysis & Results

## 7.1 Naïve Bayes

With the dataset split into training and test data in a 70:30 ratio, the algorithm produced a prediction accuracy of 70.8%.

Figure 19, shows the importance rating of all the variables in the dataset that were utilized by the naïve bayes algorithm during the training process. “gc\_verbal\_apache” is seen to be the variable with highest importance.

Chart, scatter chart

Description automatically generated

Figure 19, Naïve Bayes Variable Importance

Additionally, we validated the model by checking the error rate on the test sample data which produced an error rate of 0.2483022.

Furthermore, we compared the correlation coefficients of all the variables in the dataset, identifying the variables with the strongest association as shown in Table 1,. All the variables in column 1 were eliminated from the dataset. A naïve bayes model was implemented on the updated dataset which produced a prediction accuracy of 72%.

Table 1, Correlation between variables

|  |  |
| --- | --- |
| Perfectly Associated Variables | |
| Column 1 (Variables) | Column 2 (Variables) |
| elective\_surgery | apache\_post\_operative |
| d1\_diasbp\_noninvasive\_max | d1\_diasbp\_max |
| d1\_sysbp\_max | d1\_sysbp\_noninvasive\_max |
| d1\_sysbp\_noninvasive\_min | d1\_sysbp\_min |
| d1\_diasbp\_noninvasive\_min | h1\_mbp\_max |
| h1\_mbp\_noninvasive\_max | h1\_mbp\_min |

Overall, naïve bayes performs better with categorical data. Perhaps, the prediction accuracy from our experiment was affected by the numerical attributes in our dataset.

## 7.2 Decision Tree

With the data divided into training and test data with 70:30 ratio, the algorithm produced an accuracy of 72.9%. Additionally, pruning was performed to the trained decision tree model, the pruned decision tree model produced an accuracy of 72.9% with the same tree structure.

Diagram

Description automatically generated

Figure 20, Decision Tree on Patient Survival Prediction dataset

Figure 20 shows the decision tree model built on the training dataset, the root node begins the splitting from the “ventilated\_apache” attributes into “d1\_sysbp\_min” and “apache\_3j\_diagnosis” and ends with a total of 6 leaf nodes.

Table 2, Decision Tree Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | **Actual: 0** | **Actual: 1** |
| **Predicted: 0** | 10943 | 3781 |
| **Predicted: 1** | 4684 | 11833 |

Table 3: Statistical values of the Decision Tree algorithm

|  |  |
| --- | --- |
| **Accuracy** | 0.729 |
| **Error Rate** | 0.271 |
| **Sensitivity** | 0.7003 |
| **Specificity** | 0.7578 |
| **Positive predictive value** | 0.7432 |
| **Negative predictive value** | 0.7164 |
| **Prevalence** | 0.5002 |
| **Detection rate** | 0.3503 |

K-fold cross validation is performed for the decision tree model to ensure that the model is not overfitting the dataset. The K-fold cross validation is applied to the whole dataset.

Text, letter

Description automatically generated

Figure 21, Decision Tree K-fold Cross Validation result

Figure 21 shows the result of the K-fold cross validation of decision tree algorithm. The complexity parameter of 0.003782458 have the best accuracy result of 91.91%. Compared to the base decision tree algorithm, the accuracy of the decision tree with K-fold validation (91.91%) is better than the base decision tree algorithm (72.9%).

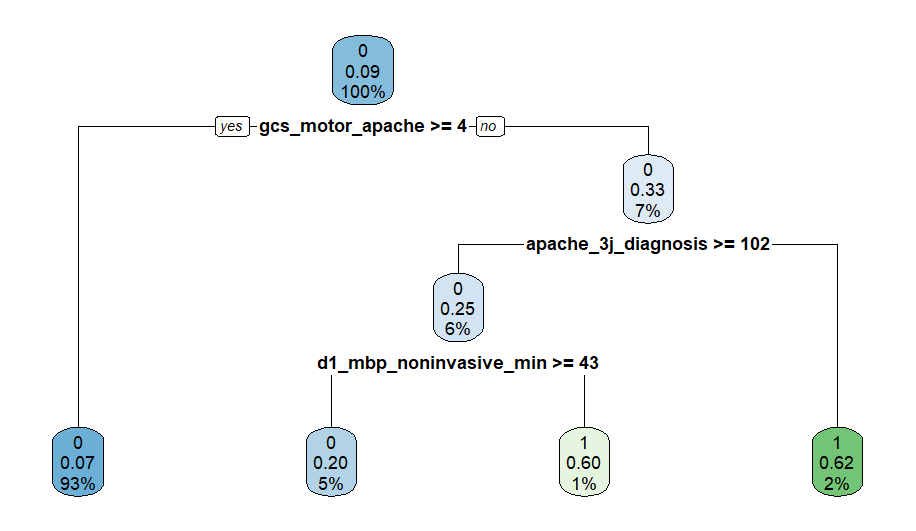


Figure 22, Decision Tree with K-fold validation tree structure

Figure 22 shows the decision tree with K-fold cross validation on the whole dataset, the root node begins the splitting from the “gcs\_motor\_apache” attributes into “apache\_3j\_diagnosis” and “d1\_mbp\_noninvasive\_min” and ends with a total of 4 leaf nodes. The tree depth is 4 levels.

Decision tree works effectively with non-linear data (CFI Team, 2022). While a major drawback is that the algorithm is very sensitive during training as a slight change in the dataset can alter the tree structure significantly. To prevent this, pruning and Cross Validation is used to ensure that the model does not overfit the model (Krueger, 2021).

## 7.3 Random Forest

With the dataset oversampled and divided into a 70:30 train-test dataset, the algorithm produced a prediction of 98% accuracy. The K-fold cross validation in random forest prevents overfitting of the model and applying hyperparameter on each iteration. Thus, a K-fold cross validation model of random forest is applied to the whole dataset to obtain the best “mtry” value, use it to train the training dataset, and predict using the test dataset.

Text, letter

Description automatically generated

Figure 23, Result for K-fold Cross Validation of Random Forest

Figure 23 shows the random forest with K-fold cross validation on the entire dataset. The best “mtry” value achieved was 55 mtry with an accuracy of 92.58%. This best “mtry” value will be implemented to the training dataset to obtain a better result.

Graphical user interface, text, application

Description automatically generated

Figure 24, Random Forest error plot with K-fold Cross Validation

Figure 24 shows the error based on the number of trees generated, the black line in the middle shows the average error for both target predictions. The error reached a convergence around 55 trees with approximately 0.1 error.

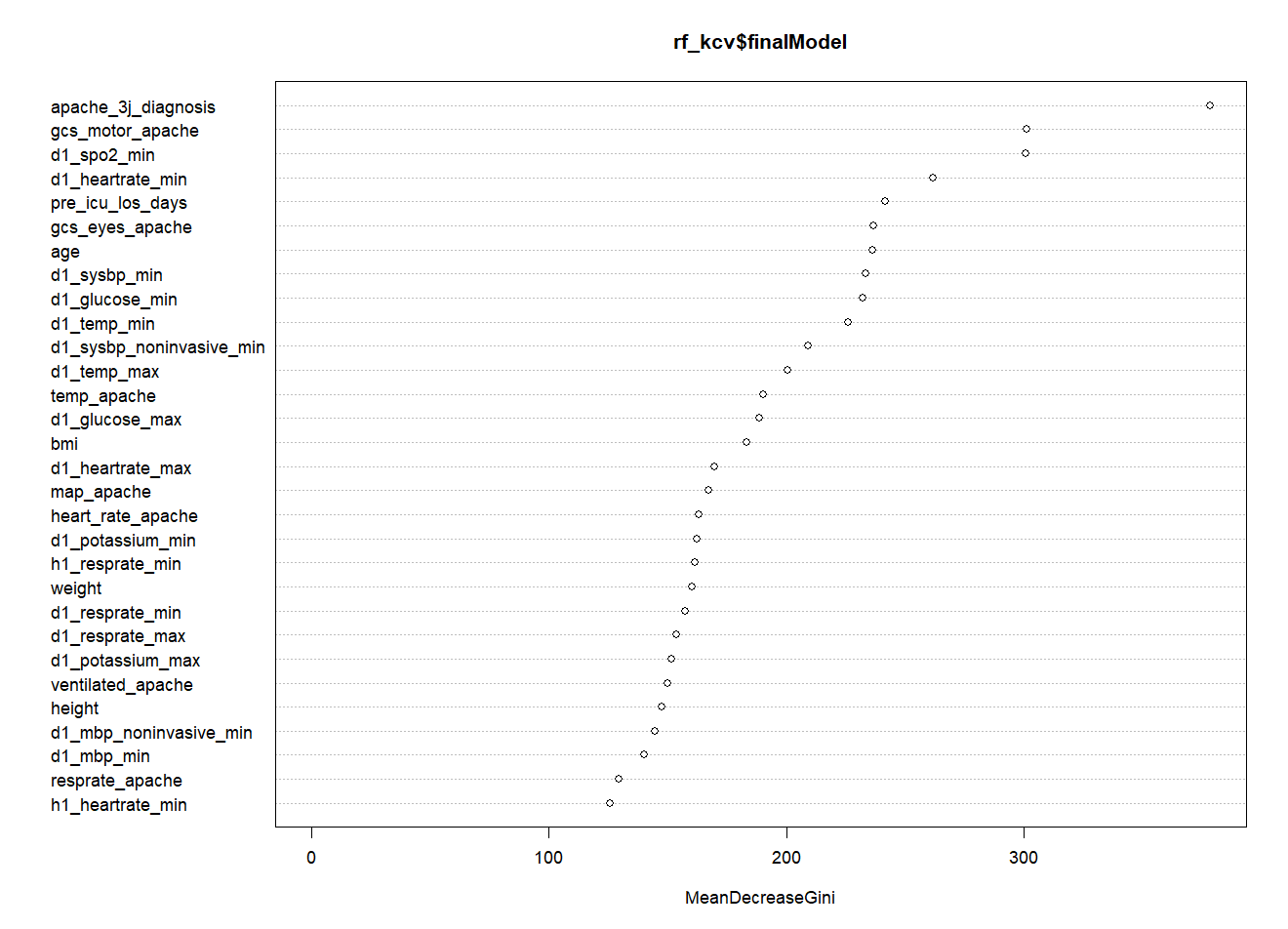


Figure 25, Variable Importance based on Mean Decrease Gini in Random Forest KCV

(Han, et al., 2016) highlights that variable selection is crucial for prediction especially in high dimensional datasets. Thus, an evaluation on the variable importance based on the mean decrease Gini coefficient is showed in Figure 25, this shows how each variable impacts to the uniformity of the random forest's nodes and leaves (Martinez-Taboada & Redondo, 2020). The variable “apache\_3j\_diagnosis” has the highest importance in the random forest model.

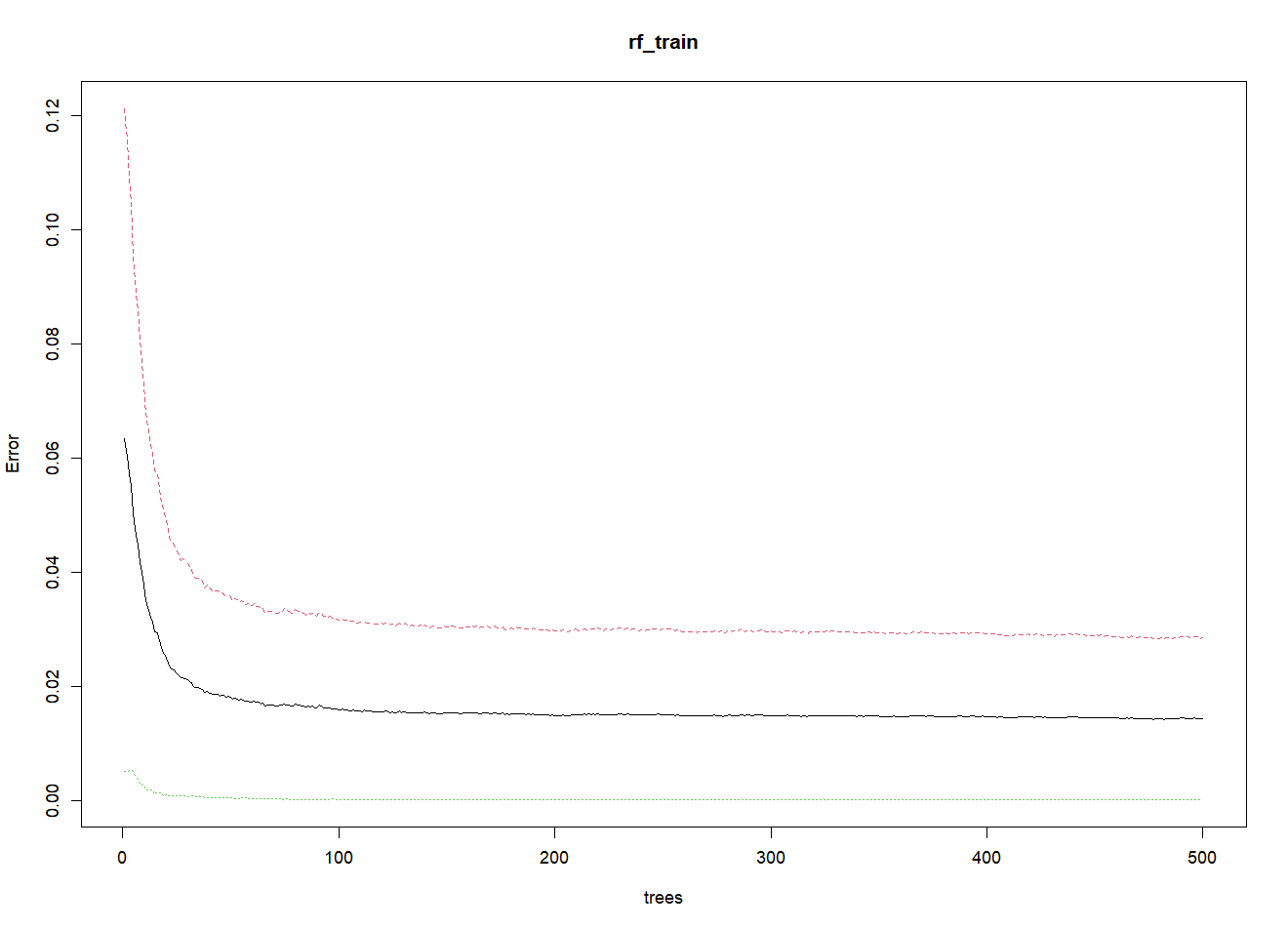


Figure 26, Error plot on Random Forest model

Figure 26 shows the error based on the number of trees generated, the black line in the middle shows the average error for both target predictions. The error reached a convergence around 100 trees with approximately 0.02 error.

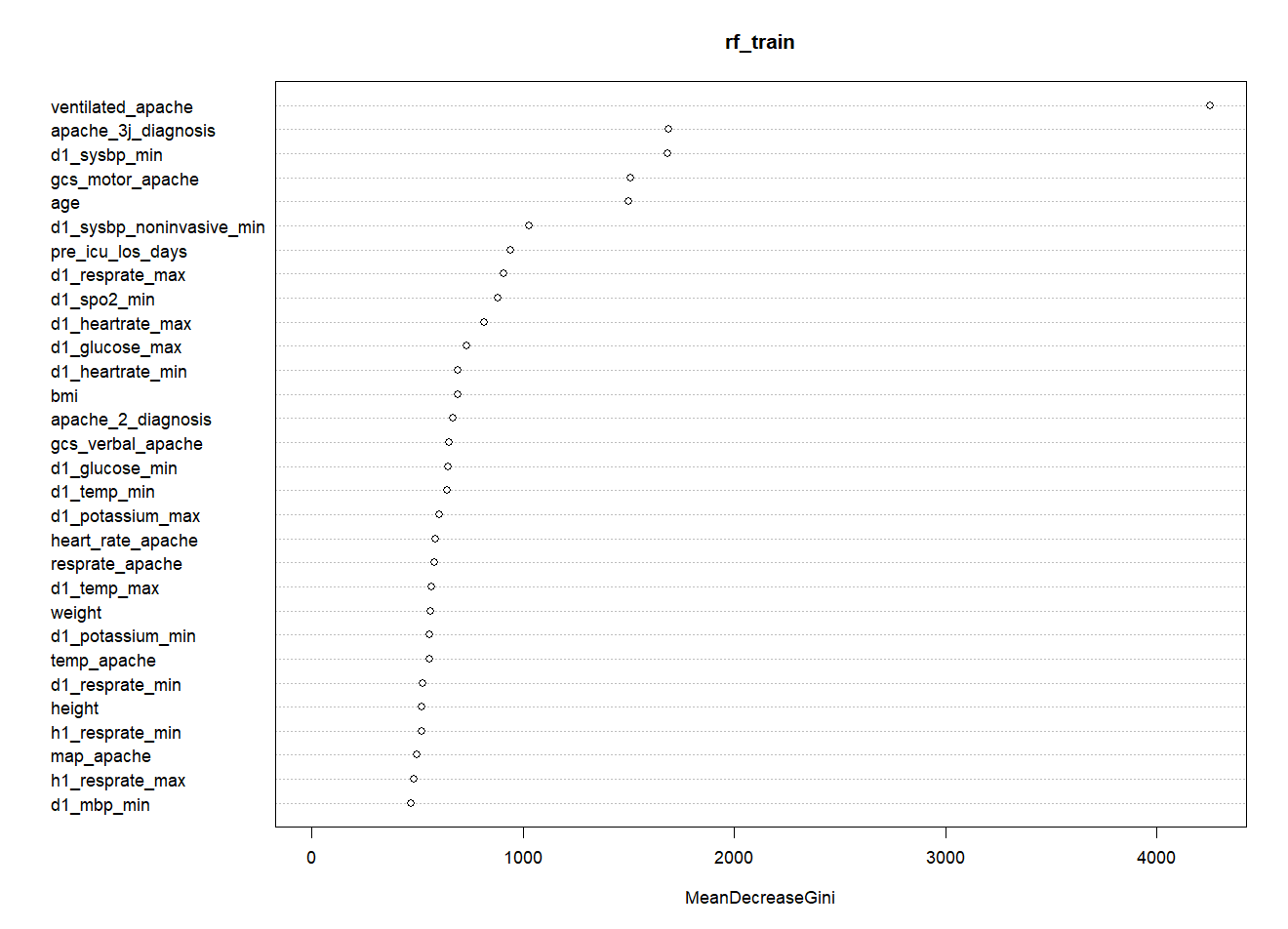


Figure 27, Variable Importance based on Mean Decrease Gini in Random Forest model

Figure 27 shows the variable importance based on Mean Decrease Gini in Random Forest model where the variable “ventilated\_apache” has the highest importance. Compared to the random forest in K-fold cross validation, the variable with the highest importance is “apache\_3j\_diagnosis”. Therefore, it is important to note that there is a change in the variable importance.

The following Table 4 and Table 5 shows the result of the random forest model using the best “mtry” value.

Table 4, Random Forest Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | **Actual: 0** | **Actual: 1** |
| **Predicted: 0** | 1516 | 4 |
| **Predicted: 1** | 471 | 15610 |

Table 5: Statistical values of the Random Forest algorithm

|  |  |
| --- | --- |
| **Accuracy** | 0.9848 |
| **Error Rate** | 0.0152 |
| **Sensitivity** | 0.9699 |
| **Specificity** | 0.9997 |
| **Positive predictive value** | 0.9997 |
| **Negative predictive value** | 0.9707 |
| **Prevalence** | 0.5002 |
| **Detection rate** | 0.4851 |

The high performance of the random forest algorithm can be attributed to the large dataset used in this study. Typically, the best results from random forest algorithms come from a large collection of very uncorrelated models (trees) working together as a group (Yiu, 2019). A major drawback of the algorithm is the increased training time, and this can also be attributed to the large size of the dataset. Additionally, this algorithm is ineffective for real-time predictions (Donges, 2021).

# 8 Conclusions

This study highlights the importance of removing redundant variables from a dataset before implementing machine learning algorithms.

The performance of the random forest algorithm was compared with that of decision tree and naïve bayes. Overall, the results showed that the random forest is more efficient in the prediction of patient survival. The experimental result from this study also agrees with those of previous scholars.

Future research may wish to implement feature selection before model training to enable the machine learning algorithm to use the most critical features in the dataset. This information could also be of benefit to the hospital as the doctors and medical practitioners could know the critical parameters to be cautious of while tending to the patients under their care.

# References

Acuña, E. & Rodriguez, C., 2004. The Treatment of Missing Values and its Effect on Classifier Accuracy. *Classification, Clustering, and Data Mining Applications,* p. 639–647.

Agarwal, M., 2022. *Patient Survival Prediction.* [Online]   
Available at: https://www.kaggle.com/datasets/mitishaagarwal/patient  
[Accessed 28 July 2022].

Alexander Liu, J. G., 2012. Generative Oversampling for Mining Imbalanced Datasets. *IEEE,* pp. 678-683.

Angelov, B., 2017. *Working with Missing Data in Machine Learning.* [Online]   
Available at: https://towardsdatascience.com/working-with-missing-data-in-machine-learning-9c0a430df4ce  
[Accessed 8 August 2022].

Biau, G. & Scornet, E., 2016. A random forest guided tour. *Test,* pp. 197-227.

Cheng, C.-H. & Huang, S.-F., 2021. A novel clustering-based purity and distance imputation for handling medical data with missing values. *Application of soft computing,* p. 11781–11801.

Cho, D.-Y. & Wang, Y.-C., 1997. Comparison of the APACHE III, APACHE II and Glasgow Coma Scale in acute head injury for prediction of mortality and functional outcome. *Intensive Care Med,* pp. 77-84.

D.Lavanya & Rani, D. K., 2011. Performance Evaluation of Decision Tree Classifiers on Medical Datasets. *International Journal of Computer Applications,* 26(4), pp. 1-4.

Dai, B., Chen, R.-C., Zhu, S.-Z. & Zhang, W.-W., 2018. *Using Random Forest Algortihm for Breast Cancer Diagnosis.* Taichung, IEEE.

Donges, N., 2021. *Random Forest Algorithm: A Complete Guide.* [Online]   
Available at: https://builtin.com/data-science/random-forest-algorithm  
[Accessed 17 August 2022].

Eldho, M. D., 2013. Risk prediction of hospital mortality for adult patients admitted to Australian and New Zealand intensive care units: Development and validation of the Australian and New Zealand Risk of Death model. *Journal of Critical Care,* pp. 935-941.

Ferdib-Al-Islam, L. A. M. M. I., 2021. *Hepatocellular Carcinoma Patient’s Survival Prediction Using Oversampling and Machine Learning Techniques.* DHAKA, Bangladesh, IEEE, pp. 445-450.

Ghosh, D. & Vogt, A., 2012. Outliers: An Evaluation of Methodologies. *Section on Survey Research Methods,* pp. 3455-3460.

Gosain, A. & Sardana, S., 2017. *Handling Class Imbalance Problem using Oversampling Techniques: A Review.* New Delhi, IEEE.

Gupta, V. K., Gupta, A., Kumar, D. & Sardana, A., 2018. Prediction of COVID-19 Confirmed, Death, and Cured Cases in India Using Random Forest Model. *BIG DATA MINING AND ANALYTICS,* 4(2), pp. 116-123.

Han, H., Guo, X. & Yu, H., 2016. *Variable Selection Using Mean Decrease Accuracy And Mean Decrease Gini Based on Random Forest.* Beijing, IEEE.

Hauskrecht, M. et al., 2012. Outlier detection for patient monitoring and alerting. *Journal of Biomedical Informatics,* Volume 46, pp. 47-55.

Ijaz, M. F., Attique, M. & Son, Y., 2020. Data-Driven Cervical Cancer Prediction Model with Outlier Detection and Over-Sampling Methods. *Sensors,* Volume 20, p. 2809.

Ishleen Kaur, T. A., 2021. An Integrated  Approach for Cancer Survival Prediction Using Data Mining Techniques. *Computational Intelligence and Neuroscience,* pp. 21-45.

Jothi, N., Rashid, N. A. & Husain, W., 2015. Data Mining in Healthcare – A Review. *Procedia Computer Science,* Volume 72, pp. 306-313.

K.Vembandasamy, R.Sasipriya & E.Deepa, 2015. Heart diseases detection using Naive Bayes algorithm. *International Journal of Innovative Science, Engineering & Technology,* 2(9), pp. 441-444.

Kabiraj, S. et al., 2020. *Breast Cancer Risk Prediction using XGBoost and Random Forest Algorithm.* Khulna, IEEE.

Krueger, E., 2021. *Build Better Decision Trees with Pruning.* [Online]   
Available at: https://towardsdatascience.com/build-better-decision-trees-with-pruning-8f467e73b107  
[Accessed 11 August 2022].

Laurikkala, J., Juhola, M. & Kentala, E., 2000. *Informal identification of outliers in medical data.* Berlin, 5th International Workshop on Intelligent Data Analysis in Medicine and Pharmacology.

Lee, C.-H., 2007. Improving classification performance using unlabeled data: Naive Bayesian case. *Knowledge Based Systems,* 20(3), pp. 230-224.

Leung, K. M., 2007. Naive Bayesian Classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering,* pp. 123-156.

Liu, M., Xu, X., Tao, Y. & Wang, X., 2017. *An Improved Random Forest Method Base on RELIEFF for Medical Diagnosis.* Qingdao, IEEE.

Mark, H., 2006. *A decision tree-based attribute weighting filter for naive bayes.* Hamilton, International Conference on Innovative Techniques and Applications of Artificial Intelligence.

Martinez-Taboada, F. & Redondo, J. I., 2020. The SIESTA (SEAAV Integrated evaluation sedation tool for anaesthesia) project: Initial development of a multifactorial sedation assessment tool for dogs. *Plos One.*

Mirza, S., Mittal, D. S. & Zaman, D. M., 2018. Decision Support Predictive model for prognosis of diabetes using SMOTE and Decision tree. *International Journal of Applied Engineering Research,* 13(11), pp. 9277-9282.

Mishra, S. et al., 2020. Performance Evaluation of a Proposed Machine Learning Model for Chronic Disease Datasets Using an Integrated Attribute Evaluator and an Improved Decision Tree Classifier. *Applied Sciences,* 10(22), pp. 1-35.

Montazeri, M., 2016. Machine learning models in breast cancer survival prediction. *Technology and Health Care,* pp. 31-42.

Myles, A. J. et al., 2004. An introduction to decision tree modeling. *Journal of Chemometrics,* 18(6), pp. 275-285.

Nanayakkara, F. S., 2018. Characterising risk of in-hospital mortality following cardiac arrest using machine learning. *Europe Pmc,* pp. 15-67.

P.L.Geenena, Gaaga, L. d., W.L.A.Loeffenb & A.R.W.Elbers, 2011. Constructing naive Bayesian classifiers for veterinary medicine: A case study in the clinical diagnosis of classical swine fever. *Research in Veterinary Science,* 91(1), pp. 64-70.

Parthiban, G., Rajesh, A. & S.K.Srivatsa, 2011. Diagnosis of Heart Disease for Diabetic Patients using Naive Bayes Method. *International Journal of Computer Applications,* 24(3), pp. 7-11.

Patil, D. D., Wadhai, V. & Gokhale, J., 2010. Evaluation of Decision Tree Pruning Algorithms for Complexity and Classification Accuracy. *International Journal of Computer Applications,* 11(2), pp. 23-30.

Quinlan, J., 1986. Induction of Decision Trees. *Machine Learning,* pp. 81-106.

R, S. E., 2021. *Understanding Random Forest.* [Online]   
Available at: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/  
[Accessed 10 August 2022].

Saini, A., 2021. *Decision Tree Algorithm – A Complete Guide.* [Online]   
Available at: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/  
[Accessed 11 August 2022].

Stewart, C., 2020. *Total amount of global healthcare data generated in 2013 and a projection for 2020.* [Online]   
Available at: https://www.statista.com/statistics/1037970/global-healthcare-data-volume/#:~:text=The%20amount%20of%20global%20healthcare,new%20data%20generated%20in%202020.  
[Accessed 30 July 2022].

William, K., 2022. *APACHE II Score.* [Online]   
Available at: https://www.mdcalc.com/calc/1868/apache-ii-score#creator-insights

Xu, Z., Shen, D., Nie, T. & Kou, Y., 2020. A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data. *Journal of Biomedical Informatics,* Volume 107.

Yang, J. et al., 2020. Brief introduction of medical database and data mining technology in big data era. *Journal of Evidence-Based Medicine,* 13(1), pp. 57-69.

Yiu, T., 2019. *Understanding Random Forest.* [Online]   
Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2  
[Accessed 9 August 2022].

# Appendix

## R Code

|  |
| --- |
| #Import Library  library**(**smotefamily**)**  library**(**ROSE**)**  library**(**rpart**)**  library**(**rpart.plot**)**  library**(**randomForest**)**  library**(**ggplot2**)**  library**(**naniar**)**  library**(**reshape**)**  library**(**corrplot**)**  library**(**dplyr**)**  library**(**caret**)**  library**(**rattle**)**  library**(**ElemStatLearn**)**  library**(**klaR**)**  library**(**ggcorrplot**)**  set.seed**(**123**)**  #Import the data set  df **<-** read.csv**(**"C:\\Users\\richa\\Downloads\\Data Mining Dataset\\dataset.csv",stringsAsFactors **=** **TRUE)**  df\_clean **=** df  #removing redundant columns  df\_clean**$**patient\_id **<-** df\_clean**$**hospital\_id**<-** df\_clean**$**icu\_id **<-** df\_clean**$**encounter\_id **<-** df\_clean**$**apache\_4a\_hospital\_death\_prob **<-** df\_clean**$**apache\_4a\_icu\_death\_prob **<-** **NULL**  #Exploratory Data Analysis  #Identify Missing and NA Values in each columns  colSums**(**is.na**(**df\_clean**))**  colSums**(**df\_clean **==** ""**)**  #visualise missing values  vis\_miss**(**df, warn\_large\_data **=** **FALSE)**  gg\_miss\_var**(**df\_clean**)** **+** labs**(**y **=** "Missing Values"**)**  #Identify Outliers  boxplot**(**df\_clean**)$**out  boxplot**(**df\_clean**$**apache\_3j\_diagnosis, xlab **=** "apache\_3j\_diagnosis"**)**  var **=** df\_clean**[**,c**(**"age", "icu\_type", "map\_apache", "d1\_spo2\_max", "h1\_mbp\_min", "aids"**)]**  boxplot**(**var**)**  #Replace the blank values into NA values  df\_clean**[**df\_clean **==** ""**]** **<-** **NA**  #Remove all NA values  df\_clean **<-** na.omit**(**df\_clean**)**  #Check for NA and blank Values after removing  gg\_miss\_var**(**df\_clean**)** **+** labs**(**y **=** "Missing Values"**)**  colSums**(**is.na**(**df\_clean**))**  colSums**(**df\_clean **==** ""**)**  #Check for duplicates  table**(**duplicated**(**df\_clean**))**  summary**(**df\_clean**)**  str**(**df\_clean**)**  #Count mean for gender  male **=** df\_clean**$**age**[**df\_clean**$**gender **==** "M"**]**  mean**(**male**)**  female **=** df\_clean**$**age**[**df\_clean**$**gender **==** "F"**]**  mean**(**female**)**  ##Data Visualization  #Plot the target variable "hospital\_death" in histogram  hist**(**df\_clean**$**hospital\_death, main **=** "Hospital Death Histogram", xlab **=** "Hospital Death"**)**  #Plot "Apache 3j Bodysystem" based on Gender  ggplot**(**df\_clean, aes**(**y **=** apache\_3j\_bodysystem**))** **+**  geom\_bar**(**aes**(**fill **=** gender**)**, position **=** position\_stack**(**reverse **=** **TRUE))** **+**  ggtitle**(**"Apache 3j Bodysystem based on Gender"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Apache 2 Bodysystem" based on Gender  ggplot**(**df\_clean, aes**(**y **=** apache\_2\_bodysystem**))** **+**  geom\_bar**(**aes**(**fill **=** gender**)**, position **=** position\_stack**(**reverse **=** **TRUE))** **+**  ggtitle**(**"Apache 2 Bodysystem based on Gender"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Apache 2 Bodysystem" based on Ethnicity  ggplot**(**df\_clean, aes**(**y **=** apache\_2\_bodysystem**))** **+**  geom\_bar**(**aes**(**fill **=** ethnicity**)**, position **=** position\_stack**(**reverse **=** **TRUE))** **+**  ggtitle**(**"Apache 2 Bodysystem based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Apache 3j Bodysystem" based on Ethnicity  ggplot**(**df\_clean, aes**(**y **=** apache\_3j\_bodysystem**))** **+**  geom\_bar**(**aes**(**fill **=** ethnicity**)**, position **=** position\_stack**(**reverse **=** **TRUE))** **+**  ggtitle**(**"Apache 3j Bodysystem based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "hospital\_death" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** hospital\_death**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**))** **+**  ggtitle**(**"Hospital Death based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "hospital\_death" based on Gender  ggplot**(**df\_clean, aes**(**x **=** hospital\_death**))** **+** geom\_bar**(**aes**(**fill **=** gender**))** **+**  ggtitle**(**"Hospital Death based on Gender"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Lymphoma" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** lymphoma**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**))** **+**  ggtitle**(**"Lymphoma based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "AIDS" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** aids**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**))** **+**  ggtitle**(**"AIDS based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Leukemia" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** leukemia**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**))** **+**  ggtitle**(**"Leukemia based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Immunosuppression" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** immunosuppression**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**)** **)+**  ggtitle**(**"Immunosuppression based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Cirrhosis" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** cirrhosis**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**)** **)+**  ggtitle**(**"Cirrhosis based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Diabetes Mellitus" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** diabetes\_mellitus**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**)** **)+**  ggtitle**(**"Diabetes Mellitus based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Hepatic Failure" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** hepatic\_failure**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**)** **)+**  ggtitle**(**"Hepatic Failure based on Ethnicity"**)** **+** theme**(**plot.title **=** element\_text**(**hjust **=** 0.5**))**  #Plot "Gender" based on Ethnicity  ggplot**(**df\_clean, aes**(**x **=** gender**))** **+** geom\_bar**(**aes**(**fill **=** ethnicity**))**  #Plot pie chart for Gender  gender **=** df\_clean %>% count**(**gender**)** %>% mutate**(**Freq **=** n**/**56935**)**  pie **=** ggplot**(**gender, aes**(**x**=**"", y**=**Freq, fill**=**gender**))** **+** geom\_bar**(**stat**=**"identity", width**=**1**)**  pie **=** pie **+** coord\_polar**(**"y", start**=**0**)** **+** geom\_text**(**aes**(**label **=** paste0**(**round**(**Freq**\***100**)**, "%"**))**, position **=** position\_stack**(**vjust **=** 0.5**))**  pie **=** pie **+** scale\_fill\_manual**(**values**=**c**(**"#55DDE0", "#33658A"**))**  pie **=** pie **+** labs**(**x **=** **NULL**, y **=** **NULL**, fill **=** **NULL**, title **=** "Gender"**)**  pie **=** pie **+** theme\_classic**()** **+** theme**(**axis.line **=** element\_blank**()**,  axis.text **=** element\_blank**()**,  axis.ticks **=** element\_blank**()**,  plot.title **=** element\_text**(**hjust **=** 0.5, color **=** "#666666"**))**  #Plot pie chart for Ethnicity  ethnicity **=** df\_clean %>% count**(**ethnicity**)** %>% mutate**(**Freq **=** n**/**56935**)**  pie\_eth **=** ggplot**(**ethnicity, aes**(**x**=**"", y**=**Freq, fill**=**ethnicity**))** **+** geom\_bar**(**stat**=**"identity", width**=**1**)**  pie\_eth **=** pie\_eth **+** coord\_polar**(**"y", start**=**0**)** **+** geom\_text**(**aes**(**label **=** paste0**(**round**(**Freq**\***100**)**, "%"**))**, position **=** position\_stack**(**vjust **=** 0.5**))**  pie\_eth **=** pie\_eth **+** scale\_fill\_manual**(**values**=**c**(**"#55DDE0", "#33658A", "#2F4858", "#F6AE2D", "#F26419", "#999999"**))**  pie\_eth **=** pie\_eth **+** labs**(**x **=** **NULL**, y **=** **NULL**, fill **=** **NULL**, title **=** "Ethnicity"**)**  pie\_eth **=** pie\_eth **+** theme\_classic**()** **+** theme**(**axis.line **=** element\_blank**()**,  axis.text **=** element\_blank**()**,  axis.ticks **=** element\_blank**()**,  plot.title **=** element\_text**(**hjust **=** 0.5, color **=** "#666666"**))**  #Correlation analysis  #Produce the correlation value for each variable within the dataset  corr **=** model.matrix**(~**0**+**., data**=**df\_clean**)** %>%  cor**(**use**=**"pairwise.complete.obs"**)**  #plot the correlation between "d1\_diasbp\_noninvasive\_min" and "d1\_diasbp\_min"  x **=** df\_clean**$**d1\_diasbp\_noninvasive\_min  y **=** df\_clean**$**d1\_diasbp\_min  plot**(**x,y, xlab **=** "d1\_diasbp\_noninvasive\_min", ylab **=** "d1\_diasbp\_min", main **=** "Correlation"**)**  abline**(**lm**(**y**~**x**))**  cor.test**(**x,y, "two.sided", "pearson"**)**  #plot the correlation between "d1\_diasbp\_noninvasive\_max" and "d1\_diasbp\_max"  x **=** df\_clean**$**d1\_diasbp\_noninvasive\_max  y **=** df\_clean**$**d1\_diasbp\_max  plot**(**x,y, xlab **=** "d1\_diasbp\_noninvasive\_max", ylab **=** "d1\_diasbp\_max", main **=** "Correlation"**)**  abline**(**lm**(**y**~**x**))**  cor.test**(**x,y, "two.sided", "pearson"**)**  #plot the correlation between "gcs\_verbal\_apache" and "h1\_mbp\_max"  x **=** df\_clean**$**gcs\_verbal\_apache  y **=** df\_clean**$**h1\_mbp\_max  plot**(**x,y, xlab **=** "gcs\_verbal\_apache", ylab **=** "h1\_mbp\_max", main **=** "Lowest Correlation"**)**  abline**(**lm**(**y**~**x**))**  cor.test**(**x,y, "two.sided", "pearson"**)**  #plot the correlation between "h1\_mbp\_noninvasive\_max" and "gcs\_verbal\_apache"  x **=** df\_clean**$**h1\_mbp\_noninvasive\_max  y **=** df\_clean**$**gcs\_verbal\_apache  plot**(**x,y, xlab **=** "gcs\_verbal\_apache", ylab **=** "h1\_mbp\_noninvasive\_max", main **=** "Lowest Correlation"**)**  abline**(**lm**(**y**~**x**))**  cor.test**(**x,y, "two.sided", "pearson"**)**  #Oversampling the data set  df\_clean\_over **=** ovun.sample**(**hospital\_death**~**., data**=**df\_clean, method**=**"over", N**=**length**(**df\_clean**[**df\_clean**$**hospital\_death**==**0,**]$**hospital\_death**)\***2**)$**data  table**(**df\_clean\_over**$**hospital\_death**)**  #Split into train and test data sets  set.seed**(**123**)**  ind **=** sample**(**2, nrow**(**df\_clean\_over**)**, replace **=** **TRUE**, prob **=** c**(**0.70,0.30**))**  train **=** df\_clean\_over**[**ind**==**1,**]**  test **=** df\_clean\_over**[**ind**==**2,**]**  #shows that the target variable class is balanced due to oversampling  table**(**train**$**hospital\_death**)**  #Convert target variable into factor type  train**$**hospital\_death **=** as.factor**(**train**$**hospital\_death**)**  test**$**hospital\_death **=** as.factor**(**test**$**hospital\_death**)**  ##Naive Bayes Classifier  X\_train **=** train**[**,**-**78**]**  y\_train **=** train**$**hospital\_death  X\_test **=** test**[**,**-**78**]**  y\_test **=** test**$**hospital\_death  #train the naive bayes classifier and cross validate  nb\_model **=** train**(**X\_train, y\_train, method **=** 'nb',  trControl **=** trainControl**(**method **=** 'cv', number **=** 20**))**  #Predict the naive bayes model based on test dataset  nb\_predict **=** predict**(**nb\_model**$**finalModel, X\_test**)$**class  confusionMatrix**(**nb\_predict, test**$**hospital\_death**)**  #Hyperparameter Tuning for Naive Bayes  #Remove the highly correlated variable  nb\_train **=** train  nb\_test **=** test  nb\_train**$**elective\_surgery**<-** nb\_train**$**d1\_diasbp\_noninvasive\_max**<-**nb\_train**$**d1\_sysbp\_noninvasive\_max**<-**nb\_train**$**d1\_sysbp\_noninvasive\_min**<-**nb\_train**$**d1\_diasbp\_noninvasive\_min**<-**nb\_train**$**h1\_mbp\_max**<-**nb\_train**$**h1\_mbp\_noninvasive\_min**<-NULL**  nb\_test**$**elective\_surgery**<-** nb\_test**$**d1\_diasbp\_noninvasive\_max**<-**nb\_test**$**d1\_sysbp\_noninvasive\_max**<-**nb\_test**$**d1\_sysbp\_noninvasive\_min**<-**nb\_test**$**d1\_diasbp\_noninvasive\_min**<-**nb\_test**$**h1\_mbp\_max**<-**nb\_test**$**h1\_mbp\_noninvasive\_min**<-NULL**  #Split into train and test  X\_train\_hyp **=** nb\_train**[**,**-**71**]**  X\_test\_hyp **=** nb\_test**[**,**-**71**]**  y\_train\_hyp **=** nb\_train**$**hospital\_death  y\_test\_hyp **=** nb\_test**$**hospital\_death  #Run the naive bayes again with new train and test dataset after removing highly correlated variables  nb\_model\_hyp **=** train**(**X\_train\_hyp, y\_train\_hyp, method **=** 'nb',  trControl **=** trainControl**(**method **=** 'cv', number **=** 20**))**  #Predict the hyperparamtered naive bayes model  nb\_predict\_hyp **=** predict**(**nb\_model\_hyp**$**finalModel, X\_test**)$**class  #Evaluate the hyperparameter Naive Bayes  confusionMatrix**(**nb\_predict\_hyp, nb\_test**$**hospital\_death**)**  #Plot Variable Importance on the Naive Bayes  X **<-** varImp**(**nb\_model\_hyp**)**  plot**(**X**)**  ###Decision Tree Classifier  #Decision Tree with rpart  fit **<-** rpart**(**hospital\_death**~**., data **=** train, method **=** 'class'**)**  rpart.plot**(**fit, extra **=** 106**)**  rpart.plot**(**fit**)**  #Predict the Decision Tree Classifier  predictions **=** predict**(**fit, test, type **=** "class"**)**  confusionMatrix**(**predictions, test**$**hospital\_death**)**  #plot the complexity parameter  printcp**(**fit**)**  plotcp**(**fit**)**  #Decision Tree pruning  DT\_rpart **=** prune**(**fit, cp **=** fit**$**cptable**[**which.min**(**fit**$**cptable**[**,"xerror"**])**,"CP"**])**  DT\_rpart  #plot the pruned decision tree  rpart.plot**(**DT\_rpart**)**  #Predict pruned decision tree  DT\_pred **=** predict**(**DT\_rpart, test, type **=** "class"**)**  confusionMatrix**(**DT\_pred, test**$**hospital\_death**)**  #Train the Decision Tree model with K-fold CV  dt\_kcv\_model **=** train**(**X\_train, y\_train, method **=** 'rpart',  trControl **=** trainControl**(**method **=** 'cv', number **=** 10**))**  x\_dt **=** df\_clean**[**,**-**78**]**  y\_dt **=** as.factor**(**df\_clean**$**hospital\_death**)**  #K-Fold CV Decision tree for the whole data set  df\_clean**$**hospital\_death **=** as.factor**(**df\_clean**$**hospital\_death**)**  df\_kcv\_model **=** train**(**hospital\_death**~**., data **=** df\_clean, method **=** 'rpart',  trControl **=** trainControl**(**method **=** 'cv', number **=** 10**))**  #Plot the Decision Tree Model K-CV  rpart.plot**(**df\_kcv\_model**$**finalModel**)**  ###Random Forest Classifier  #Train the Decision Tree model with K-fold Cross Validation (KCV)  rf\_kcv **=** train**(**hospital\_death**~**., data **=** df\_clean, method **=** 'rf',  trControl **=** trainControl**(**method **=** 'cv', number **=** 10**))**  #Plot the random forest with KCV  plot**(**rf\_kcv**$**finalModel, main **=** "Error Plot in Random Forest with K-Fold Cross Validation"**)**  #Plot Variable Importance based on Mean Decrease Gini  varImp**(**rf\_kcv**$**finalModel**)**  varImpPlot**(**rf\_kcv**$**finalModel**)**  #Train Random Forest model with Train data set and best mtry based on RF KCV  rf\_train **=** randomForest**(**hospital\_death**~**., data **=** train, mtry **=** 55**)**  #plot the random forest  plot**(**rf\_train**)**  #Plot the variable importance  varImpPlot**(**rf\_train**)**  #Predict the Random Forest model  rf\_pred **=** predict**(**rf\_train, test, type **=** "class"**)**  confusionMatrix**(**rf\_pred, test**$**hospital\_death**)** |