NED UNIVERSITY

OF ENGINEERING SCIENCE AND TECHNOLOGY

COMPARATIVE ANALYSIS OF LOGISTIC REGRESSION AND RANDOM FOREST

Dissertation

Submitted to:

Miss Sana Fatima

By:

Ahmed Gala[1], Mohammad Zain Ul Abedin[2], Syed Arsalan Naseeruddin[3], Khawar Khan[4], Mohammad Zubair[5]

[1, 2, 3, 4, 5] Department of Software Engineering, NED University of Engineering Science and Technology

Abstract

Rapid technological development has given rise to Machine Learning concepts that assist in solving complex real-world problems. There are several Machine Learning algorithms, each with its advantages and disadvantages. However, the goal of the study is to compare, test, analyze and evaluate two algorithms: Logistic Regression and Random Forest, using a dataset of patients in the Intensive Care Units (ICUs). The research aims at finding which one of these two algorithms is effective in predicting the patient's survival in ICU. For this, a Patient Survival Prediction Dataset is used to train the algorithms and then the results are analyzed. The study then ascertains the performance of the algorithms through a set of evaluation metrics (precision, accuracy, recall, confusion matrix, recall, specificity, and receiver operating characteristics curves) and derives results and statistics from it. By the data findings, it became evident that Logistic Regression is better than Random Forest for classifying the outcome of this experiment. The results illustrated that Logistic Regression leads in almost all the evaluation metrics used while it also provides more accurate predictions as Logistic Regression showed 0.6% greater accuracy and 0.9% higher precision than Random Forest for the prediction of patient survival in ICU.

Keywords: *Machine Learning, Logistic Regression, Random Forest, Patient Survival Prediction*

TABLE OF CONTENTS

TITLE **PAGE**

LIST OF ABBREVIATIONS

AUC	:	Area Under the Curve
BMI	:	Body Mass Index
FN	:	False Negative
FP	:	False Positive
FPR	:	False Positive Rate
ICU	:	Intensive Care Unit
IG	:	Information Gain
LR	:	Logistic Regression:
ML	:	Machine Learning
PSP	:	Patient Survival Prediction
RF	:	Random Forest
ROC	:	Receiver Operating Characteristics
SVMs	:	Support Vector Machines
TN	:	True Negative
TP	:	True Positive
TPR	:	True Positive Rate

1. INTRODUCTION:

1.1. Background of the Study

In this era of technological advancement, data scientists have been training computers through advanced techniques thereby developing intelligence in computers. Such technological advancements also resulted in several Machine Learning algorithms. These Machine Learning algorithms are an integral part of this modern development that helps solve real-world complex problems in these dynamic times.

Machine Learning (ML), often referred to as predictive analytics or predictive modelling, is a branch of Artificial Intelligence. It uses data and algorithms to help computers emulate the human learning process. Arthur Samuel, a pioneer of artificial intelligence research, devised this term defining it as a “computer’s ability to learn without being explicitly programmed.”

It uses algorithms that examine input data to predict outcomes that lie in the desired range of values. Hence, through the data fed into these algorithms, they are trained to make predictions and optimize their operational performance enabling the world to produce numerous sophisticated and revolutionary products such as self-driving cars.

1.2. Problem Statement and Usefulness of this Study

Patients who are brought into Intensive Care Units (ICUs) are often unresponsive and their medical records may take days to transfer which makes it difficult to extract information regarding the state of the patient. Therefore, quick determination of a patient’s condition such as a chronic condition is crucial to help in clinical decisions.

That is why this study is useful in developing a potential system which can assist a healthcare professional to predict a patient’s survival through one of these two Machine Learning algorithms: Logistic Regression and Random Forest. The prediction of the patient’s survival in the ICU will benefit in improving the patient’s survival outcome by making appropriate medical decisions.

Hence, the paper will address this by identifying the best one of these two classification algorithms for this purpose by performing their comparative analysis using a particular dataset.

1.3. Objectives of the Study

The goal of this study is to extensively discuss two ML algorithms: Logistic Regression and Random Forest. This study will describe each of these algorithms in detail and will also outline the working of these algorithms. Subsequently, the study will perform an extensive comparative analysis of both algorithms.

Comparing algorithms and stating one of them as a better one is a complex task and an open problem. Even the most experienced data scientists can tell which algorithm will perform best only when they experiment with other algorithms on a given dataset. Therefore, to collate the two algorithms, this research paper evaluates their performance through following benchmarks: confusion matrix, accuracy, precision, recall, specificity and the ROC curves.

An appropriate input dataset is used to quantify their performance. Additionally, a consistent method of evaluation of results is selected to compare the output values and derive a conclusion from it. Moreover, the paper compares the algorithms based on their advantages and disadvantages as well. Hence, the research concludes which of these two algorithms is better in this problem context or for this particular input dataset.

In a nutshell, the objective of the paper is also to assess which one of these two Machine Learning algorithms effectively evaluates and gives better results of whether the patient will survive in the ICU or not based on the input dataset.

1.4. Limitations of the Study

Firstly, the original dataset was downsized with respect to the attributes, so this study uses 8 attributes of the dataset which are the most significant for the results. Downsizing was done because of the time constraint as the training algorithm on a large dataset requires ample time and because of some missing values and outliers in the dataset.

Moreover, research focuses on confusion matrix, accuracy, precision, recall, specificity, and the ROC curves metrics for performance evaluation. Both algorithms were implemented in Python version 3.8.16.

1.5. Structure of the Study

The report is organized into five main headings as listed below:

- **Introduction** – It describes the background of the study, the problem statement, how the study is useful, its benefits, the objectives of the study, and its limitations.
- **Literature Review** – It explains Logistic Regression and Random Forest algorithms in detail along with their working, their types (if exist), and their applications.
- **Comparative Analysis** – It elaborately compares the two algorithms. This section contains an input dataset description, research framework, explanation of the experiment, performance evaluation through metrics, results and statistics, analysis of results, comparison of advantages and disadvantages of both algorithms, and interference of the results obtained.
- **Conclusion** – It describes the conclusion and future work for subsequent researchers.
- **References**

2. LITERATURE REVIEW

2.1 LOGISTIC REGRESSION:

2.1.1 Definition

In Supervised Machine learning, regression is the method used for prediction when the target variable is continuous. There are many situations in which Machine Learning models have to provide a discrete outcome of an event on basis of probability, whether a certain outcome is expected to happen or not. Therefore, it was necessary to develop an algorithm which gives us the likelihood of an event's occurrence based on an analysis of previous related outcomes. This led to the development of a classification algorithm known as Logistic Regression. Logistic Regression is used in Supervised Machine Learning to model a relationship between a categorical dependent variable and one or more independent variables to predict the output of that dependent variable **[1]**. It is also known as a Binary classification algorithm as it returns a dichotomous output (0 or 1) for any given number of input variables by utilizing the concept of probability and statistics. As a member of the generalized linear model family, it also shares three key characteristics:

- The inputs given to the model are linear in nature.
- The output (response) variable relates to the linear inputs through a linking function.
- From the three exponential distributions: Binary distribution, Poisson distribution and Gaussian distribution, the output variable of Logistic Regression is from the Binary exponential distribution family.

For dichotomous response variable, Binary distribution is applied whereas when the output variable is the representation of number of outcomes, Poisson distribution is the choice. However, for continuous dependent variable, Gaussian distribution is utilized.

Amongst the three, a basic Logistic Regression model belongs to the Binary distribution family of linear models as it only gives the probability of the occurrence of an event. Keeping this in view, a few assumptions need to be fulfilled for this algorithm to give precise and accurate results. These are explained as follows:

- The response variable must be binary in nature.
- The independent variables and the linking function should be linearly correlated.
- There should be no multicollinearity among the independent variables, which is the interrelationship existing between two independent variables thereby reducing the accuracy of the results of the regression model.
- There should be no extreme outliers (abnormal data values) in the dataset.
- The number of variables which have a little or negligible effect on the response variable should not be included in the model.
- The dataset should be large to obtain better and more accurate results.

2.1.2 Types of Logistic Regression

The implementation of Logistic Regression on a certain event depends on what outcome is to be expected. Therefore, there are three types of Logistic Regression algorithms which are applied to model relationships between dependent and independent variables:

1) **Binomial Logistic Regression:** A type of Logistic Regression in which the response variable has two possible outcomes, 0 or 1. This type of Logistic Regression suggests whether a certain output is true or false based on probability and statistical analysis.

2) **Multinomial Logistic Regression:**
It is an extension of Binomial Regression in which the response variable has more than one predicted outcome and in which the order of the output obtained is not taken into consideration is called multinomial regression.

3) **Ordinal Logistic Regression:**
It is a type of Logistic Regression where the target variable (response variable) has two or more ordered outcomes. It is also an extension of Binomial Logistic Regression.

2.1.3 Working of Logistic Regression

The basic model of Logistic Regression is a modified version of the Linear Regression model, whose equation is given as **[1]**:

$$y' = \beta_0 + \beta_1 x + \epsilon$$

Where *x* is the input feature of the model. If N number of inputs is to be given, we may use the matrix notation as shown below:

$$y = \left(\begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix}_{(1,n)} X \right) + b$$

For Binomial Logistic Regression, there are only two variables, the response variable and the independent variable, that is why the equation of Logistic Regression can be derived from the first equation. The outcome probability of the new equation must be between 0 and 1 which implies that the outcome has to be a positive number and should always be less than one. It should be noted that the exponent (***e***) raised to the power of any negative number gives a positive result and any number divided by a number greater than it returns a number smaller than one. Therefore, by implementing these conditions, the equation can be written as **[2]**,

$$P(Y=1|X)=\frac{\mathrm{e}^{(\beta_o+\beta_1 x)}}{\mathrm{e}^{(\beta_o+\beta_1 x)}+1}$$

Where *β_0 and β_1* are the regression coefficients which can be calculated by using the maximum likelihood estimation and *x* is the independent variable with which the response variable is being compared. The expression *P (Y = 1 | X)* is read as "the probability that Y is equal to 1 for a given value of X". This enables the outcome (Y) to be between 0 and 1. Note that here there are only two terms involving regression constants and one independent variable *'x'* which is the indication of the Binomial nature of Logistic Regression.

Upon simplification, the above equation becomes,

$$\ln(\frac{p}{1-p})=\beta_0+\beta_1 x$$

Where the right-hand side of the equation represents the linear combination of the independent variables, and the left-hand side is known as the ***log-odds*** or ***odds ratio*** or the ***logit function*** which is the linking function between the response variable and the independent variable.

Further simplification gives the final form of the equation which is **[1]**,

$$P(x)=\frac{1}{1+\ e^{-y\prime}}$$

Here, ***P(x)*** is a special logistic function called a **sigmoid function** which limits the probability output of the model between 0 and 1. The following graph demonstrates the sigmoid function:

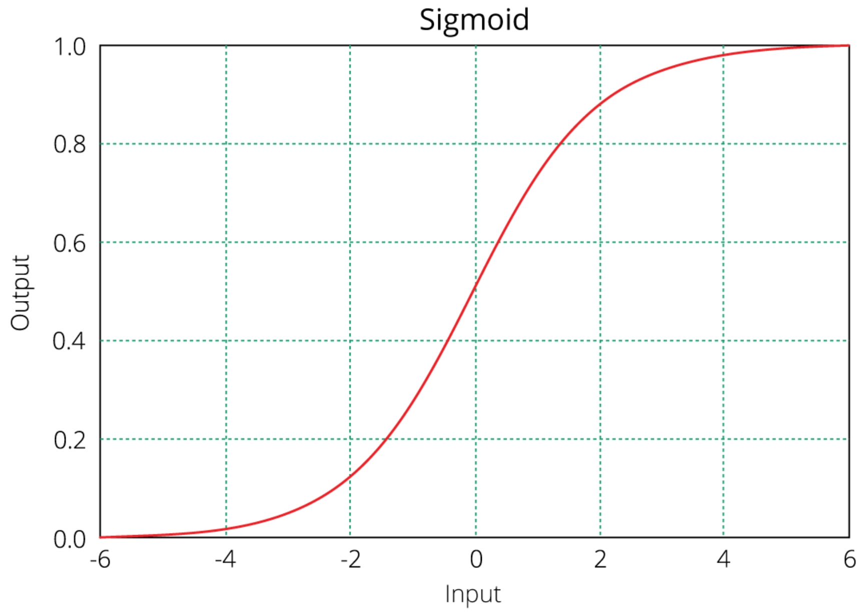


Figure 2.1 Sigmoid Graph

This indicates that for every unit change in the input value, the odds ratio is multiplied by the ***e to the power of β.*** Therefore, it can be concluded that the odds of generating a specific outcome are significantly high if the log-odds is more than one. On the other hand, it is less likely that a certain outcome can be expected provided that the log-odds turns out to be less than one.

Hence, Logistic Regression is an extremely useful algorithm to train models involving categorical dependent variables to identify the response when certain input variables are provided to it. By using the ***logit function,*** Logistic Regression accurately predicts the likelihood of an event by mapping the response variable to either 0 or 1.

2.1.4 Applications:

There are numerous applications of Logistic Regression. Some of these applications are discussed below.

2.1.4.1 Fraud Detection

Logistic Regression is used to identify fraud in credit card transactions by training models to identify fake transactions and alert the supportive organizations. Khare and Sait **[3]** examined and analyzed the presentation of Decision Tree, Random Forest, SVM and Logistic Regression classifier algorithms where the models were applied to the raw and pre-handled information. The results of the analysis have indicated that Logistic Regression has an accuracy of 97.7%

2.1.4.2 Diagnosis of Diabetes Mellitus

Logistic Regression is also useful in identifying patients suffering from Diabetes Mellitus by developing a model which uses the data of the patient's diet history and tells us whether he is suffering from Diabetes or not. A research study has been conducted by Panday **[4]** in which he investigated and assessed the presentation of decision tree and Random Forest classifiers with a dataset of patients having Diabetes Mellitus and implemented Random Forest and Logistic Regression. The data findings demonstrated that Logistic Regression results were 81.17% accurate. This indicates the significance of Logistic Regression in the field of medical sciences.

2.2 RANDOM FOREST:

2.2.1 Definition:

Random Forest is an ensemble-based learning algorithm consisting of different decision trees **[5]**. Ensemble means to make a stronger prediction by using the results collected by combining different models instead of using a single model.

Random forest is a type of homogeneous ensemble which combines the results of all the decision trees and this process is known as aggregation. The decision trees uses top-down approach that makes different decisions to decrease entropy and in making the predictions **[5]**. Random Forest is an example of the bagging method as it trains multiple models on different subsets of the training data known as "bagging" and rather than using all features to train each tree it selects a random subset of features for each tree known as "the random subspace".

Decision trees are very common in supervised learning, but it has some drawbacks such as overfitting and biases. On the other hand, Random Forest tends to deliver more accurate results and better performances by minimizing bias.

2.2.2 Working:

Random Forest consists of several decision trees and each of them works separately to create an output or make a prediction based on certain conditions. On every iteration, Random Forest will randomly take a subset of m_{try} number of features which is less than the total number of features available **[13]** after which the decision tree (whose working will be described later) forms a prediction.

The common values of m_{try} for Random Forest is the total number of features divided by three for regression and it is square root times the sum of classification features. A decision tree predicts by going from the root node to the leaf node through the decision path **[13]**.

For a single decision tree, some split features are defined which are used to decrease entropy at each node. Here, Entropy refers to the degree of randomness of the data or impurity of a dataset. So, n number of trees will produce n results out of which some will be identical while others will be different. Random forest will then form the prediction by averaging the result of n number of trees.

Hence, the chances of error in Random Forest are very less as compared to decision trees **[6]**. The formula for calculating the entropy in decision trees is given below.

$$\text{Entropy} = -p\log_2(p) - q\log_2(q)$$

Where p is the proportion of positive examples in the dataset and q is the proportion of negative examples in the dataset. Samples belonging to the same class result in minimum entropy **[7]** while the classes of datasets which have low entropy are more stable **[7]**.

Another term which is used for defining split features is information gain or entropy gain. It returns the expected decrease in entropy while partitioning the dataset according to that attribute or selected split feature. The formula for information gain is as follows:

$$IG = Entropy(Sd) - \sum_{v \in Values} (|Sv|/|Sd|)Entropy(Sv)$$

Where Sd refers to all the predictions that can be made on the values of that particular attribute and values means all the possible values of the feature. The leaf nodes always possess the information gain value as zero **[8]**.

Random forest working is illustrated in fig.1

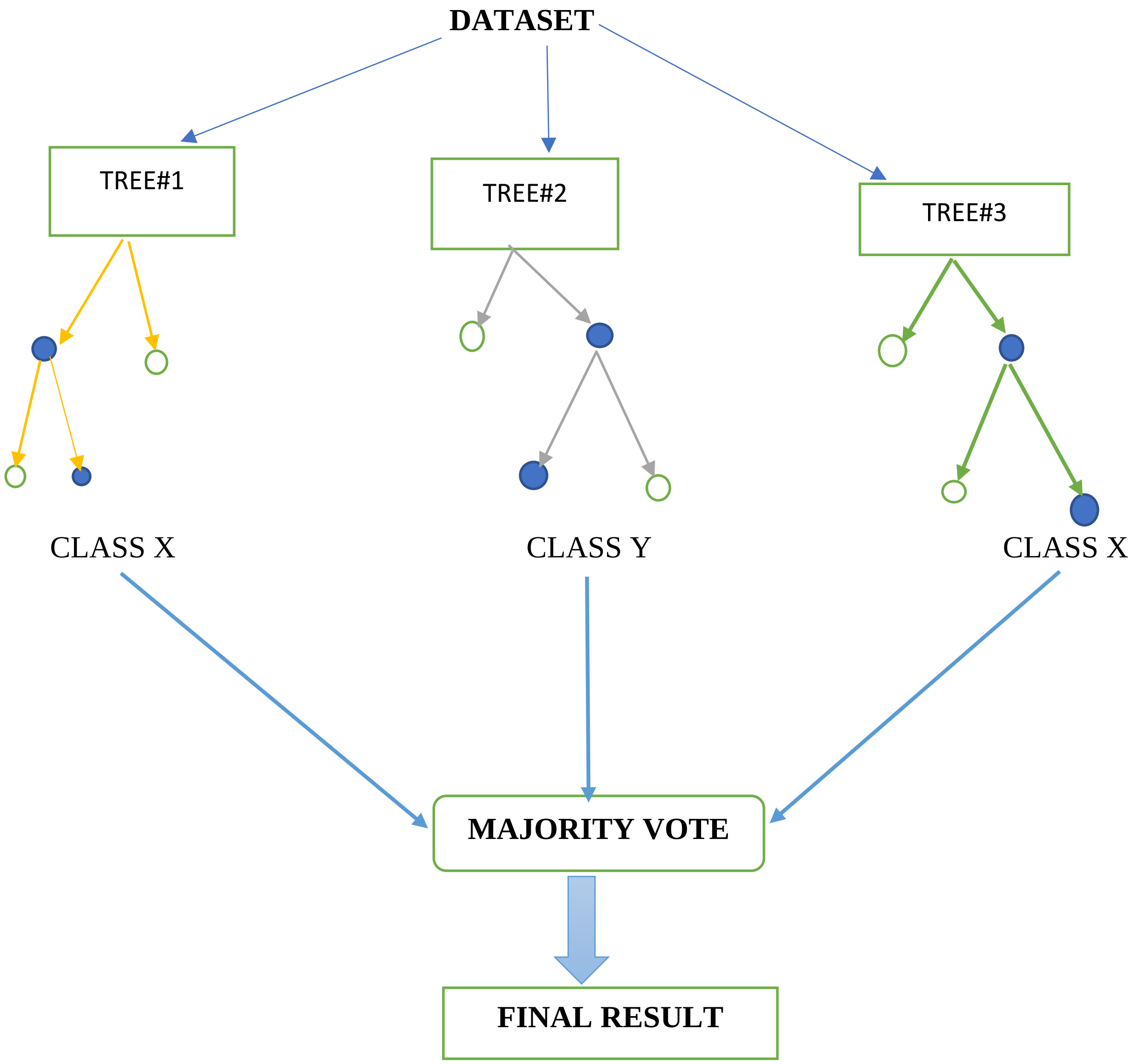


Figure 2.2 Random Forest

2.2.3 Applications:

2.2.3.1 Railway Crossing Crashes

In highway rail-grade crossings, authorities depend on prediction techniques for checking crossing safety before using any added resources. The paper **[8]** suggests that both crash and non-crash predictions were improved when we use Random Forest. Random Forest is also used for allocating the budget for novel resources in crossing safety as it can reduce the false alarm rates quite effectively for such data which is unbalanced.

2.2.3.2 Water Resources Identification

In the area of water resources, random forest usage exponentially increased since its properties in the area were discovered. Unlike the models used before in water resources, Random Forest can make an accurate prediction from calculated results. Only a few pieces of research suggest the disadvantages of Random Forest. In one-third of the related works in water resources, Random Forest improves inference of other approaches **[9]**.

3. COMPARATIVE ANALYSIS

This section deals with the successful implementation and comparative analysis of Logistic Regression and Random Forest for Patient Survival Prediction to predict whether the patient will survive or not.

3.1 Methodology:

When applying any Machine Learning algorithm to a dataset, the first task is to identify the nature of the data. After that, a suitable algorithm is chosen for that dataset and finally, a model is constructed that outputs predictions.

Hence, the research follows this graphical workflow:

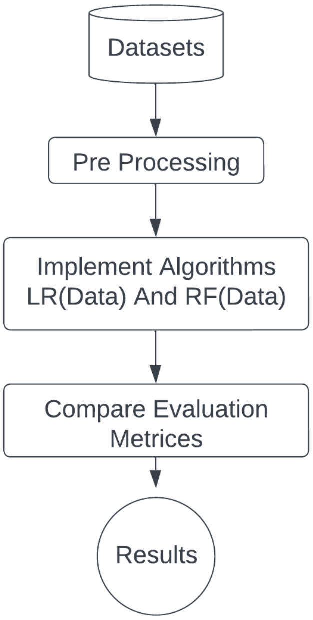


Figure 3.1

3.1.1 Tools:

In this research, Python version 3.8.16 is used for simulation using Google Collaboratory cloud-based service on a hosted system of Intel Xeon 2.20GHz processor with 13 GB RAM operating on Linux OS. The following libraries are used:

- **Pandas:** This Python library was used in research for data manipulation such as selection, filtration, reshaping, aggregating, splitting, and selecting data.
- **Matplotlib:** It was used for data visualization, such as learning curves and confusion matrices.
- **NumPy (Numerical Python):** The study makes use of it for handling arrays and numerical data as the input dataset is in the form of 2D array.

- **Scikit-learn:** It was used for the formation of Logistic Regression and Random Forest models and for evaluation of models and data analysis features.

3.2 Sources of Data:

The source of data is secondary as it was downloaded from Kaggle.com and the dataset is of Patient Survival Prediction (PSP) **[12]** which contains 91700 patients' records spanning over 186 columns. This paper uses only 8 attribute variables suitable for the research since the study is not meant for any large-scale benchmarks. After cleaning data and removing missing values, the categorical data was converted into binary type. The final dimensions of the dataset used in the study are [91700 rows * 8 columns].

At this stage, preprocessing is successfully completed which includes the removal of entries with missing values, duplication, and loading errors.

3.3 Data Structure:

The table below mentions the main data structures used in the study:

S. No	Attributes	Attribute Type
1	D1 lactate maximum	Numerical
2	D1 lactate minimum	Numerical
3	APACHE – 4A Hospital Death Probability	Numerical
4	APACHE – 4A ICU Death Probability	Numerical
5	Age	Numerical
6	Weight	Numerical
7	BMI	Numerical
8	Gender	Binary
9	Outcome	Binary

Figure 3.2

3.4 Data Sample:

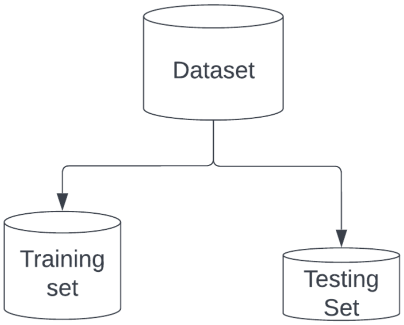
Following is the data sample used in the study:

D1 lactate max	D1 lactate min	APACHE 4A Hospital Death Probability	APACHE 4A ICU Death Probability	Age	Weight	BMI	Gender	Outcome
1.3	1	0.1	0.05	68	73.9	22.73	0	0
3.5	3.5	0.47	0.29	77	70.2	27.42	1	1
2.1	2.3	0	0	25	95.3	31.95	1	1
1.5	2.7	0.04	0.03	81	61.7	22.64	1	0

Figure 3.3

3.5 Data Splitting:

After Dataset Pre-processing, the dataset was divided into two subsets called training and testing set in ratios of 80% and 20% respectively. The division was not ordered but random and this is called ***random sampling***. Random sampling is done for an unbiased evaluation and to increase the predictive capabilities of the model.



3.6 Performance Evaluation

In this study, performance evaluation was carried out to assess the performance of the two classification algorithms on the given dataset. These results were important in identifying which algorithm was more efficient and in which scenario as well. There could be two possible results, the patient will be able to survive or not. Therefore, a confusion matrix was used to determine how well the algorithms analyzed the data and gave correct predictions. Using the results of the confusion matrix, five parameters were evaluated for the algorithms named, ***precision, accuracy, recall, specificity, ROC (receiver operating characteristic)*** and ***Area under the Curve***.

3.6.1 Confusion Matrix

The confusion matrix uses tabular representation to depict the results of model's analysis. It classifies the outcomes into four distinct categories which include the following fields **[4]**:

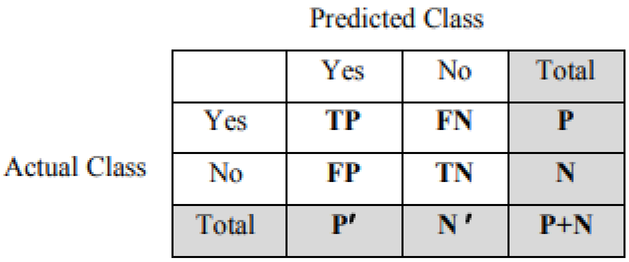
- **True Positives (TP):** These are defined as the positive values which are predicted correctly by the model, i.e., the actual value and the predicted values both are affirmative.

- **True Negatives (TN):** These are defined as the negative values which are correctly predicted by the model and that the actual and predicted values both are negative.

- **False Positives (FP):** These are defined as the positive values which are predicted by the model, but they are negative. These are the incorrect values predicted by the model. It is also known as a **Type *I* error.**

- **False Negatives (FN):** These are defined as positive values that the model labels as negative. It is also known as a **Type *II* error.**

Given below is a general configuration of the confusion matrix used in the study **[4]**:



		Predicted Class		
		Yes	No	Total
Actual Class	Yes	**TP**	**FN**	**P**
	No	**FP**	**TN**	**N**
	Total	**P′**	**N ′**	**P+N**

Figure 3.4

3.6.2 Accuracy:

Accuracy is the ratio of the correctly predicted values to the total number of observations predicted from the test dataset. This value indicates to what extent the model has correctly labelled a certain value out of all the classes.

Mathematically it can be written as **[10]**:

$$Accuracy = \frac{TP + TN}{\sum(TP + FP + TN + FN)}$$

3.6.3 Precision:

Precision is the ratio of the correctly predicted values to all the positively predicted values in the dataset. Precision helps in estimating how correctly a model has predicted the outcome of the event provided in the dataset. **[11]**;

Mathematically it can be written as;

$$Precision = \frac{TP}{TP + FP}$$

3.6.4 Recall:

Recall provides the ratio of the positively predicted values to the actual positive values and falsely predicted positive values. This demonstrates how many values the model has predicted as positive are also actually positive. It is also called ***sensitivity*** *or* ***True Positive Rate.***

Mathematically it can be written as **[4]**;

$$Recall = \frac{TP}{TP + FN}$$

3.6.5 Specificity:

It is the ratio of the values predicted correctly as negative to the total number of actual negative values and the falsely predicted negative values. This shows how well the model has classified the data correctly as negative.

Mathematically it can be written as **[10]**;

$$Specificity = \frac{TN}{TN + FP}$$

3.6.6 ROC Curve:

It is a curve which shows the performance of the classifier at all levels of the classification threshold. It is plotted in a 2D plane with two parameters [**10**]

- **True Positive Rate (TPR):** It is another term for

- **False Positive Rate (FPR):** It **c**omputes the ratio of false positives and the total number of negatives.

$$False\ Positive\ Rate = \frac{FP}{TN + FP}$$

3.6.7 AUC:

It stands for Area Under the Curve which is a term derived from Integral Calculus. It measures the area under the entire ROC curve and outputs a combined measure of performance across all classification thresholds.

AUC from Study [**10**]

$$AUC = \frac{1}{2} * (Recall + Specificity)$$

AUC in the context of integral calculus

$$\int_0^1 ROC\ \ dt$$

3.7 Results and Statistics:

This part deals with the result and statistical data gathered during the research.

3.7.1 Tabular Analysis:

3.7.1.1 The Confusion Matrix

The Confusion Matrices for Logistic Regression and Random Forest were found to be of order 2x2 for binary classification.

Predicted Class

		Survived	Deceased
Actual Class	Survived	3072	29
	Deceased	224	61

Logistic Regression

Predicted Class

		Survived	Deceased
Actual Class	Survived	3039	55
	Deceased	219	73

Random Forest

From the above matrices the four parameters which can be derived from the confusion matrix are given below:

Parameters	**Logistic Regression**	**Random Forest**
True Positive	3072	3039
True Negative	61	73
False Positive	29	55

False Negative	224	219

Fig 3.5

3.7.1.2 Elementary Metrics

The four elements of confusion matrices are important for the evaluation of elementary metrics and can be calculated through formulae discussed in section 3.2

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	AUC (%)
Logistic Regression	92.5	99.1	93.2	67.8	80.5
Random Forest	91.9	98.2	93.3	57.0	75.1

Fig 3.6

3.7.2 Graphical Analysis:

3.7.2.1 Elementary Metrics plot

The following chart shows a cluster plot of the comparative results of the evaluation metrics of both algorithms.

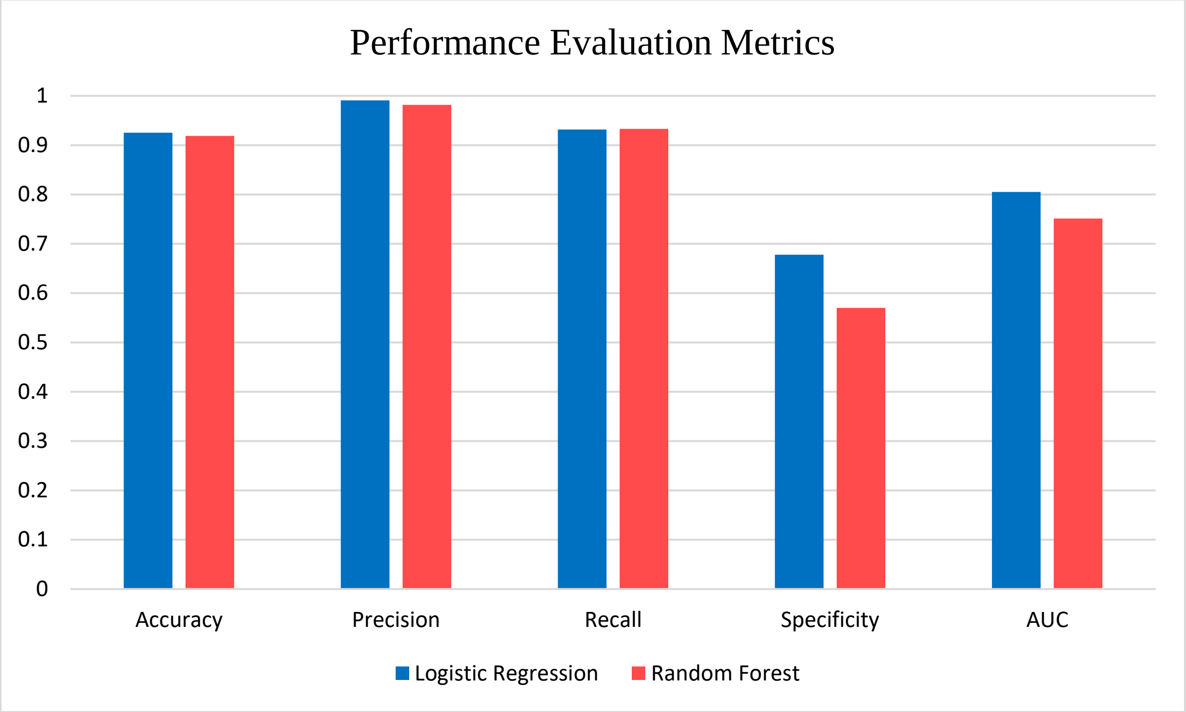


Fig 3.7

3.7.2.2 ROC Curves

The area under the curve is called AUC. The diagonal line represents a boundary. If the curve is to be formed under the diagonal line the outcome is said to be worse than random prediction however this is not certainly the case here.

Logistic Regression

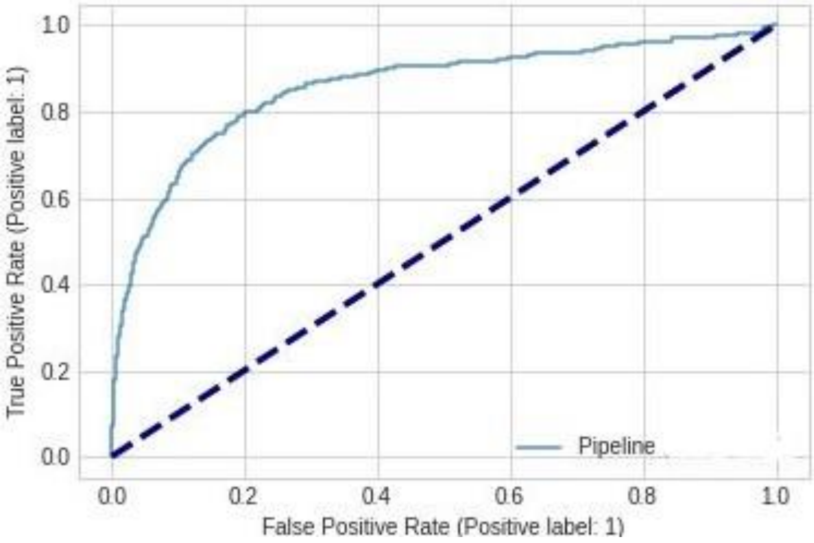


Fig 3.8

Random Forest

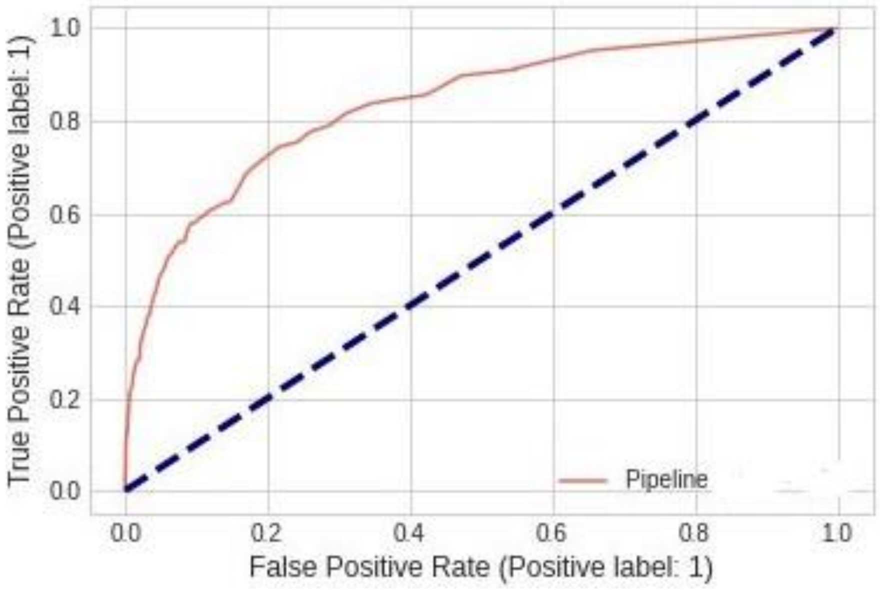


Fig 3.9

3.7.3 Result Analysis

This section highlights the main advantages and disadvantages between the working and prerequisites of both algorithms, leading towards the further analysis of the experimental results obtained from section 3.3.

3.7.3.1 Advantages Comparison

LOGISTIC REGRESSION	RANDOM FOREST
Logistic Regression gives optimal results when the input features are linearly separable.	Random Forest is an ensemble model which utilizes state-of-art techniques for separating non-linear data.
Logistic Regression is easier to implement and train because of its simplicity and efficiency.	It is a complex computational model capable of predicting outcomes of very complex data.

Logistic Regression performs well when the dataset is low-dimensional i.e., the number of input features is less as compared to the observations taken.	Random Forest supersedes Logistic Regression in the domain of high dimensional data analysis.
It is less prone to over-fitting, as low-dimensional data is provided to the algorithm.	Random Forest tends to over-fit only when the dimensions of the data shrink.
Logistic Regression can be efficiently extended to map multiple independent variables to a single response variable.	Random Forest is a binary as well as multi-class classifier capable of solving a wide array of problems.
Since Logistic Regression relies on Stochastic Gradient Descent, the data can be easily and efficiently updated with minimal occurrence of errors in the output.	Random Forest uses hyperparameter tuning for the optimization of ensemble decision trees.

3.7.3.2 Disadvantages Comparison

LOGISTIC REGRESSION	RANDOM FOREST
Logistic Regression struggles to provide accurate results in non-linear problems because of its high dependency on linear data.	Random Forest cannot extrapolate the linear trend for linear problems.
Logistic Regression is less efficient if there exist complex relationships between the datasets, resulting in inaccurate outcomes.	Random Forest does not give any noticeable performance for simple datasets.
Logistic Regression underperforms when the input dataset is high-dimensional, thereby causing over-fitting in the model. This leads to inaccurate results with high errors and deviations.	It requires a large dataset for proper working otherwise it gives performance leaks (i.e., Random Forest does not perform up to the intended mark).

It is highly sensitive to outliers, which are abnormal values in the dataset arising due to measurement errors, sampling problems or invalid data entries.	Random Forest being an ensemble model minimizes error but at the same time poses significant computational overhead.
The input features require a lot of filtering to remove redundant data, collinearity and outliers, which increases the complexity of the process.	The output of Random Forest is an aggregate of multiple decision trees hence it does not require rigorous filtering, but it can induce noise in the outcome.
Logistic Regression can only be used to predict discrete outcomes as it uses binary distribution to predict the outcomes. This makes it inefficient when the input data is continuous.	Random Forest determines output on basis of Majority Voting if data is multi-collinear then it gets biased in favour of small groups.

3.7.3.3 Inference

In the experiment conducted, most of the input features selected for the analysis were linearly separable which reduced the redundant amount of data present in the dataset, despite having a few multicollinear independent variables as well. Furthermore, the dataset was low-dimensional i.e., the number of input features was less than the number of observations taken due to which there should have been minimal chances of over-fitting in the models **[2]**.

The results of the comparative analysis of both algorithms are shown in Figures 3.1, 3.2, and 3.3 while the parameters chosen for the comparison were the confusion matrix, accuracy, precision, recall, specificity and the ROC curves. The results make it evident that Logistic Regression was the best-suited algorithm for classifying the outcome of this experiment.

Figure 3.1 shows that the predictions of Logistic Regression were more accurate than Random Forest among which 3072 were correctly labelled as true and 224 were correctly labelled as false. On the other hand, Random Forest correctly labelled 3039 inputs as true and 219 as false which were less accurate predictions as compared to Logistic Regression.

Figure 3.2 implies that Logistic Regression gave an *accuracy* of 92.5% whereas Random Forest was 91.9% accurate. The *precision* result obtained from Logistic Regression was 99.1% and that of Random Forest was 98.2%. There was a slight difference of 0.1% in the recall observation of the two algorithms, with the Logistic Regression at 93.2% and Random Forest at 93.3%. In terms of *Specificity* and *AUC*, Logistic Regression had better performance than Random Forest, with the difference in specificity being 10.8% and that of AUC being 5.4%.

When all the evaluation metrics are taken into consideration, Logistic Regression had better performances as compared to Random Forest. This may be due to the dataset being linearly separable**[2]** which made it easy for Logistic Regression to classify the outcomes. Moreover, Logistic Regression also performed well because of the low-dimensional nature of the data.

Yet there was still errors and deviation in the results of Logistic Regression which may be due to the existence of slight multicollinearity between the input features. On the other hand, Random Forest did not perform well because the size of the dataset was too small which made it difficult for the algorithm to correctly identify the results, as any outlier or error would have also been considered as a correct value by the algorithm, leading towards inaccurate results.

In addition to that, the usage of a low-dimensional dataset did not work in favour of Random Forest as it led towards over-fitting in the model. Nevertheless, the overall performance of both algorithms was satisfactory with high accuracy and precision despite a few errors due to the data conditions.

4. CONCLUSION

The main goal of this research study was to compare the performance of two Machine Learning algorithms, Logistic Regression and Random Forest, by implementing them on a Patient Survival Prediction Dataset and analyzing the results obtained from them. The data findings illustrated that among the two algorithms, Logistic Regression gave better performance than Random Forest because of the low-dimensional dataset and linear separation between the independent variables and the response variable, but it was less likely to provide accurate results if the data was multicollinear.

On the contrary, Random Forest would have performed better with multicollinear datasets as it uses decision trees to evaluate and assess the outcomes of the experiment. This algorithm also performed well but due to the small size of the input features, it was more prone to over-fitting which led towards errors in the results.

The research also discussed and contrasted the advantages and disadvantages of Logistic Regression and Random Forest. It is decided by the nature of the dataset which algorithm should be implemented. If it is a linear dataset, Logistic Regression would be the suitable choice; otherwise, for a non-linear dataset, Random Forest should be the chosen technique.

Logistic Regression remains an obvious choice when the primary goal is to predict the outcome of a linearly separable dataset. Overall, this study was limited to a small dataset and the optimal performance of the algorithms cannot be assessed from it. Therefore, further research is required to evaluate these algorithms on larger datasets and in different pre-conditions as well.

5. REFERENCES

[1] A. Mishra, and C. Ghorpade, "Credit card fraud detection on the skewed data using various classification and ensemble techniques," *IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2018. **[Online]**. **Available:** https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Credit+card+fraud+detection+on+the+skewed+data+using+various+classification+and+ensemble+techniques&btnG= **[Accessed** Dec. 29, 2022**]**.

[2] R. Couronne, P. Probst, and A. L. Boulesteix, "Random forest versus logistic regression: a large-scale benchmark experiment," *BMC bioinformatics*, vol. 270, July 17, 2018.**[Online]. Available:** https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Random+forest+versus+logistic+regression%3A+a+large-scale+benchmark+experiment%2C%E2%80%9D+&btnG= **[Accessed** Dec. 29, 2022**]**.

[3] N. Khare, S. Y. Sait, "Credit card fraud detection using machine learning models and collating machine learning models," *Int J Pure Appl Math*, vol. 118, 2018.**[Online]. Available:** https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Credit+card+fraud+detection+using+machine+learning+models+and+collating+machine+learning+models&btnG= **[Accessed** Dec. 29, 2022**]**.

[4] P. Madhu, "comparative analysis of random forest and logistic regression for diagnosis of diabetes mellitus.", 2019.**[Online].**

Available: https://elibrary.tucl.edu.np/handle/123456789/10243 **[Accessed** Dec. 29, 2022**]**.

[5] K.Kirasich, T.Smith, B.Sadler. "Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets," SMU Data Science Review, Vol:1, No.3,2018. **[Online]. Available:** https://scholar.smu.edu/datasciencereview/vol1/iss3/9?utm_source=scholar.smu.edu%2Fdatasciencereview%2Fvol1%2Fiss3%2F9&utm_medium=PDF&utm_campaign=PDFCoverPages **[Accessed** Dec. 29, 2022**]**.

[6] M.Schonlau, R.Y.Zou. The random forest algorithm for statistical learning, Vol.20, no.1, Mar. 24, 2020.**[Online]. Available:** https://doi.org/10.1177/1536867X20909688 **[Accessed** Dec. 29, 2022**]**.

[7] Z.Wang, C.Cao, Y.Zhu. "Entropy and Confidence-Based Undersampling Boosting Random Forests for Imbalanced Problems**,"** *IEEE Transactions on Neura Networks and Learning Systems*, vol.31, no.12, pp.5178-5191. Jan. 24, 2020. **[Online]. Available:** https://ieeexplore.ieee.org/abstract/document/8968753?casa_token=Fi1P-

FarwkwAAAAA:s6KZsXU_RdZKfkvDxRxwjeX2B3TWSmcxodwDFpzmXn11bHqyEo8VHw2OHHUUlcE93-VGBPXgyjo **[Accessed** Dec. 29, 2022**]**.

[8] X.Zhou, P.Lu, Z.Zheng, D.Tolliver, A.Keramati. "Accident Prediction Accuracy Assessment for Highway-Rail Grade Crossings Using Random Forest Algorithm Compared with Decision Tree," *Reliability Engineering & System Safety*, vol.200. 2020. **[Online]. Available:** https://doi.org/10.1016/j.ress.2020.106931 **[Accessed** Dec. 29, 2022**]**.

[9] H.Tyralis, G.Papacharalampous, A.Langousis. "A Brief Review of Random Forests for Water Scientists and Practitioners and Their Recent History in Water Resources," *Water*, vol. 11, no.9, Apr. 30, 2019. **[Online]. Available:** https://doi.org/10.3390/w11050910 **[Accessed** Dec. 29, 2022**]**.

[10] N. F. Hordri, S. S. Yuhaniz, N. F. M. Azmi, and S. M. Shamsuddin, "Handling Class Imbalance in Credit Card Fraud using Resampling Methods," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 11, 2018. **[Online]. Available:** https://www.researchgate.net/profile/Nur-Hordri/publication/329418254_Handling_Class_Imbalance_in_Credit_Card_Fraud_using_Resampling_Methods/links/5c0f22244585157ac1b918df/Handling-Class-Imbalance-in-Credit-Card-Fraud-using-Resampling-Methods.pdf **[Accessed** Dec. 29, 2022**]**.

[11] U. Uzma, S. Kanjilal, and N. Yadav, "Comparative Analysis of Stress among Undergraduate Students Using Logistic Regression and Random Forest Techniques," *NEU Journal for Artificial Intelligence and Internet of Things*, no. 1, Jan. 14, 2022. **[Online]. Available:** https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Comparative+Analysis+of+Stress+among+Undergraduate+Students+Using+Logistic+Regression+and+Random+Forest+Techniques&btnG= **[Accessed** Dec. 29, 2022**]**.

[12] https://www.kaggle.com/code/sadiaanzum/patient-survival-prediction/notebook

[13] X.Zhao, Y.Wu, D.L.Lee, W.Cui. "iForest: Interpreting Random Forests via Visual Analytics," *IEEE Transactions on Visualisation and Computer Graphics*, vol. 25, no.1, pp.407-416. Sept, 5, 2018. **[Online]. Available:**

https://ieeexplore.ieee.org/abstract/document/8454906?casa_token=z0chcQbWpAoAAAAA:XcsvfZ5tuUX4XDvcnaVzjo2eMUJoeAyopVWXVqKKDG2RO5sbdQuAdAMNbVdQ91U-xGSLeZhtUFIs **[Accessed** Dec. 29, 2022**]**.

6. AUTHOR'S BIOGRAPHY

Ahmed Gala (SE-21077)

Ahmed Gala is currently pursuing a four-year bachelor's degree of Software Engineering at NED University of Engineering and Technology, Karachi. He completed his O-Levels from Osman Public School System (2019) and A-levels from Highbrow College (2021) with 4 A*. He is in second year of his bachelor's degree with 3.98 CGPA. He is an intermediate C++ and Python developer with interest in Machine Learning and Data Science.

Mohammad Zain Ul Abedin (SE-21081)

Muhammad Zain completed his matriculation from Metropolitan Academy (2019) and his Intermediate in Pre-Engineering from Delhi Govt. College (2021) securing fourth position in Board of Intermediate Education. He is currently enrolled in a bachelor's programme in Software Engineering at NEDUET. He is an intermediate python Developer. He wishes to advance his career in the field of Machine Learning, Operating Systems and Database Management.

Syed Arsalan (SE-21084)

Syed Arsalan is currently enrolled in Bachelor of Engineering in Software Engineering at NED University of Engineering and Technology, Karachi. He secured third position in the Intermediate Examinations held in 2021. He is currently in the Second Year of his bachelor's with a CGPA of 3.88. He is proficient in C++ programming language and has elementary proficiency in Python. He has interests in Full-Stack Web Development and Machine Learning and is acquiring skills to establish a career in these fields of Software Engineering.

Khawar Khan (SE-21093)

Khawar Khan completed his matriculation from Al-Maarif High School and then completed his intermediate Pre-Engineering from Adamjee Government Science College, securing the 12th position on the board of intermediate education, Karachi. He is currently pursuing his bachelor's degree in Software Engineering from NEDUET, Karachi. His areas of interest are Data Science, Cloud Computing, and web development.

Mohammad Zubair (SE-21094)

Muhammad Zubair completed his matriculation from Bahria College EAB-1 Majeed and then completed his intermediate Pre-Engineering from Bahria College, Karsaz, securing seventh position in board of intermediate education Karachi. He is currently pursuing his bachelor's degree in Software Engineering from NEDUET, Karachi. His areas of interest are Cloud Computing, Blockchain and web development.

APPENDIX

SOURCE CODE

A.1 Importing Libraries

```
import pandas as pd                      #a dataframe library

import numpy as np

import matplotlib.pyplot as plt               #to plot data
```

For support

```
from sklearn.pipeline import Pipeline          #to automate multiple transformation steps

from sklearn.metrics import  *                 #For Evaluating performance metrics

from sklearn.preprocessing import MinMaxScaler,StandardScaler

# to transform the values of a dataset on a similar scale
```

A.2 For splitting of dataset

```
from sklearn.model_selection import train_test_split
```

```
t_x = dataset[x]

t_y = dataset['hospital_death']

train_x,test_x,train_y,test_y = train_test_split(t_x,t_y , train_size=0.95)
```

A.3 Dataset imputing with mean

```
From pandas import read_csv
import numpy

dataset = pd.read_csv(r'/content/drive/MyDrive/Dataset.csv')
features_with_nan = [features for features in dataset.columns if dataset[features].isnull().sum()>=1]
c_features_na = [features for features in dataset.columns if dataset[features].isnull().sum()>=1 and dataset[features].dtype=='O']
def replace_cat_features(dataset,c_features_na):
    data = dataset.copy()
    data[features] = data[features].fillna('Missing')
    return data
n_features_na = [features for features in dataset.columns if dataset[features].isnull().sum()>=1 and dataset[features].dtype!='O']

for features in n_features_na:
    median_values = dataset[features].median()
    dataset[features] = dataset[features].fillna(median_values)
dataset = replace_cat_features(dataset,c_features_na)
dataset['gender'].fillna(1)
dataset.replace([np.inf, -np.inf], np.nan)
dataset.dropna(inplace=True)
```

A.4 For Data Visualization

```
plt.style.use('seaborn-whitegrid')

ax = plt.gca()

rfc_disp = RocCurveDisplay.from_estimator(a, test_x, test_y, ax=ax, alpha=0.8)


plt.plot([0, 1], [0, 1], color='navy', lw=3, linestyle='--')

plt.show()
```

A.5 For Creating Logistic Regression

```
From sklearn.linear_model import LogisticRegression

from sklearn.model_selection import train_test_split


t_x = dataset[x]

t_y = dataset['hospital_death']

train_x,test_x,train_y,test_y = train_test_split(t_x,t_y , train_size=0.95)

model = Pipeline([('scalar',MinMaxScaler()),

                  ('LR', LogisticRegression())])


a=model.fit(train_x,train_y)

pred = model.predict(test_x)
```

A.6 For Creating Random Forest

```
from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier


t_x = dataset[x]

t_y = dataset['hospital_death']

train_x,test_x,train_y,test_y = train_test_split(t_x,t_y , train_size=0.95)

model = Pipeline([('scalar',MinMaxScaler()),

                  ('RF', RandomForestClassifier())])


a=model.fit(train_x,train_y)

pred = model.predict(test_x)
```

A.7 For Evaluation of Performance Metrics

```
From sklearn.metrics import *

a=model.fit(train_x,train_y)

pred = model.predict(test_x)

print(pred)

acc = accuracy_score(pred,test_y)

print(acc)

conf_mat = confusion_matrix(test_y, pred)

print(conf_mat)
```

```
rfc_disp = RocCurveDisplay.from_estimator(a, test_x, test_y, ax=ax, alpha=0.8)
```