AS1101: Advance Statistics

Statistical Analysis of In-hospital Mortality of ICU Patients with Heart Failure

PREPARED BY

Priya Soni (2020BTechCSE059)

Department of Computer Science Engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University Jaipur

16 December, 2022

CERTIFICATE

This is to certify that the project work entitled "**Statistical Analysis of In-hospital Mortality of ICU Patients with Heart Failure**" submitted by **Priya Soni (2020BTechCSE059) and Ritisha Mathur (2020BTechCSE065)** towards the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science and Engineering** of JK Lakshmipat University, Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted.

Dr. Jaya Gupta
Assistant Professor-Mathematics
Institute of Enginnering and Technology
Date of Submission: 16 December, 2022

ACKNOWLEDGEMENT

We have completed this project under the guidance and supervision of **Dr. Jaya Gupta,** Assistant Professor, JK Lakshmipat University. We will be failed in our duty if we do not acknowledge the esteemed scholarly guidance, assistance, feedback, and knowledge I have received from them towards the fruitful and timely completion of this work.

We express our deepest thanks to **Dr Dheeraj Sanghi**, Vice Chancellor, JK Lakshmipat University, and **Dr Sanjay Goel**, Director, Institute of Engineering and Technology, JK Lakshmipat University for their constant support, encouragement, and guidance.

We also acknowledge with a deep sense of reverence, our gratitude towards our parents for their direct or indirect support during the entire course of this project.

Thanking You

Sincerely Yours,

Priya Soni (2020BTechCSE059)

Ritisha Mathur (2020BTechCSE065)

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Health problems and their study had always intrigued the curiosity of data scientists, due to increasing competitive demand for accurate information in the health industry. The collection and retrieval of data through proper channels can help provide improved quality healthcare to users. From healthcare institutes to doctors and researchers to health insurance providers, everyone relies on factual data collection and its accurate analysis to make well-informed decisions about patients' health status. Diseases and their survival impact can be predicted at the earliest stage with the help of data science in healthcare. As a rapidly evolving area, data analytics can become the right solution to detect, manage and predict diseases which threaten life and can cause high economic cost. This report seeks to establish a statistical, graphical, and predictive analysis of an available dataset related to in-hospital mortality for intensive care units (ICU) – admitted HF patients.

CHAPTER 1: INTRODUCTION

In- hospital mortality rates quite well indicate the quality of healthcare provided by institutions and doctors. Variation in mortality rates should not be ignored, as they might tell us about the unavoidable changes in healthcare, but it cannot be the only criteria to judge the quality of healthcare. Application of data science helps us predict the symptoms of disease at a very early stage. A predictive analytical model uses previous data, finds patterns and similarities in the data, and generates accurate predictions. Such a model correlates and associate every feature or data point, symptoms and biological tests to diseases and survival output. This enables us to identify the disease's stage, extent and thus implement appropriate treatment measures.

Predictive analytics in the healthcare industry can be helpful in analyzing and monitoring the demand for pharmaceutical logistics, predicting any near or future crisis of patients' health. Understanding the data using machine learning algorithms can solve various problems like predicting stroke patterns, chances of survival of a heart failure patient with other symptoms. ML Algorithms helps to combine variables like lab test values, socioeconomic background, already diagnosed diseases and other individual information to generate results of patients' health conditions.

The aim of this study is to analyze the data of heart failure patients admitted in intensive care units(ICU) of a hospital, compare in-hospital mortality rates dependency of the patient of various possible comorbidities.

About Data

We have a data of 1177 patients with Heart Failure(HF) admitted in a hospital. It includes demographic characteristics like age, sex, comorbidities(diabetes, COPD, drug information), vital signs recorded on ICU admission (respiratory rate, heart rate, blood pressure(systolic and diastolic), temperature. Lab test including anion gap, blood urine nitrogen, chloride, calcium, potassium, sodium, MCH, MCHC, MCV, hemoglobin, platelet, RDW, WBC, RBC. The primary outcome was in-hospital mortality, which tells about the survival status at the time of discharge.

CHAPTER 2: LITERATURE SURVEY

Lots of research has been done and published discussing the affective implementation of data science and its tools in health care industry.

S.S. Alaoui [1] tried to establish statistical and predictive analysis of a dataset related to chronic kidney disease (CKD) by employing the widely used statistical model including multiclass logistic regression, decision forest, neural network to estimate disease risk prediction. They took into consideration many factors like sex, RBCs, anaemia, albumin, blood glucose, blood urea to analyse CKD dataset to generate 100% accurate based on machine learning algorithm.

D Han [2] tried to establish a nomogram that predicts the in-hospital death of patients with CHF in the intensive care unit (ICU). They analysed the in-hospital mortality rate to be 12.4%. They used multivariate analysis to determine independent risk-factors like age, sec, dopamine, intubation, heart rate, blood pressure (systolic and diastolic), blood urea nitrogen and many more. They used Decision curve analysis to assess the clinical usefulness of the model.

CHAPTER 3: OBJECTIVES

1 Data analysis using correlation to assess degree of association between comorbidities (Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) and age.

2 To estimate the extent of association of Atrial Fibrillation in the age group (>60) i.e., elderly patients (exposed group).

3 To determine the dependency of the diseases (Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) on gender for the patients with admitted in hospital.

4 To determine the dependency of the diseases (Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) on age for the patients admitted in hospital and which age group has maximum disease incidence.

5 To predict if a patient has renal failure based on the significant risk factors using feature scaling.

6

a. To predict Creatinine levels in females based on renal failure output.
b. To predict Creatinine levels in males based on renal failure output

CHAPTER 4: TOOLS USED

1. Correlation Coefficient

A statistical indicator of the strength of a linear link between two variables is the correlation coefficient. Its values may be between -1 and 1. Values in one series rise as those in the other drop, and vice versa, according to a correlation coefficient of 1, which denotes a complete negative or inverse connection. A value of 1 indicates a direct and flawlessly positive link. No linear relationship exists when the correlation coefficient is 0.

Correlation Coefficient formula:

$$\rho_{xy} = \frac{\mathrm{Cov}(x, y)}{\sigma_x \sigma_y}$$

where:

ρ_{xy} = Pearson product-moment correlation coefficient

$\mathrm{Cov}(x, y)$ = covariance of variables x and y

σ_x = standard deviation of x

σ_y = standard deviation of y

2. CHI-Squared Test

chi-squared test is a statistical analysis based on observations of a random collection of variables. Its symbol is 2. Usually, it involves comparing two sets of statistical data. The chi-square test, which takes the null hypothesis as a given and treats it as such, is used to gauge the likelihood of the observations being made.

The chi-squared test is used to determine whether the observed value and expected value differ in any way.

$$x_c^2 = \frac{\Sigma\,(O_i\; - E_i)^2}{E_i}$$

Where, O_i is the observed value and E_i is the expected value.

$$\mathrm{Ei} = \frac{(\mathrm{Rowtotal})(\mathrm{ColumnTotal})}{N}$$

3. Relative Risk Ratio

A risk ratio (RR), often known as a relative risk, contrasts the risk of a health event (such as an illness, injury, risk factor, or death) among two groups. The risk (incidence percentage, attack rate) in group 1 is divided by the risk (incidence proportion, attack rate) in group 2, and the result is the calculated risk. Typically, the two groups are separated by demographic characteristics like sex (for example, males versus girls) or exposure to a possible risk factor (e.g., did or did not eat potato salad). Frequently, the comparison group is referred to as the unexposed group and the primary interest group as the exposed group.

The formula for risk ratio (RR) is:

$$\frac{\textit{Risk of disease (incidence proportion, attack rate) in group of primary interest}}{\textit{Risk of disease (incidence proportion, attack rate) in comparison group}}$$

A risk ratio of 1.0 shows that the two groups have the same level of danger. A risk ratio greater than 1.0 denotes a higher risk for the numerator group, which is often the exposed group. If the risk ratio is less than 1, the exposed group is at a lower risk, suggesting that exposure might be acting as a preventative measure for disease.

4. Logistic Regression

Using a given set of independent factors, logistic regression is used to predict the categorical dependent variable. Classification issues are solved using logistic regression. In Logistic Regression, we discover the S-curve that allows us to categorise the sample data. For accuracy estimation, the maximum likelihood estimation method is applied.

Logistic regression's equation can be written as:

$$\mathbf{log}\left[\frac{\mathbf{y}}{\mathbf{1-y}}\right] = \mathbf{b_0} + \mathbf{b_1 x_1} + \mathbf{b_2 x_2} + \mathbf{b_3 x_3} + \cdots + \mathbf{b_n x_n}$$

Logistic regression functions F(x) = $\frac{1}{(1+e^{-x})}$

5. Linear Regression

Using a specific set of independent variables, linear regression is utilised to forecast the continuous dependent variable. Finding the best fit line in linear regression allows us to anticipate the outcome with ease. The relationship between the dependent variable and the independent variable in linear regression must be linear. For estimating accuracy, the least squares estimate approach is applied.

The regression line is written as: **y= a0+a1x+ ε,**

Here, a0 and a1 are coefficients and ε is the error.

CHAPTER 5: METHODOLOGY

We have used the following methodological framework, which includes the most important set of steps followed to generate the model to accomplish our objectives.

Figure 1: Methodological Framework

1) Data Collection

We have collected data about prediction of in-hospital mortality for intensive care units (ICU) admitted patients with heart failure from "Kaggle". This data was created in 2015 and it has 1177 entries as number of patients admitted. Table1 shows the names of 50 attributes and their types, whether they are numerical(continuous) or categorical.

Table 1: Data Attributes description

Name Of Attributes	Type	Nominal Values
ID	int64	
outcome	categorical	1: Survives; 0: Not Survives
age	int64	
gender	int64	1: Male; 2:Female
BMI	float64	
Hypertensive	categorical	1: affected; 0: not affected
Atrialfibrillation	categorical	1: affected; 0: not affected
CHD	categorical	1: affected; 0: not affected
Diabetes	categorical	1: affected; 0: not affected
Deficiencyanemias	categorical	1: affected; 0: not affected
Depression	categorical	1: affected; 0: not affected
Hyperlipemia	categorical	1: affected; 0: not affected
Renal_failure	categorical	1: affected; 0: not affected
COPD	categorical	1: affected; 0: not affected
heart rate	float64	(normal, abnormal)

Systolic blood pressure	float64	
Diastolic blood pressure	float64	
Respiratory rate	float64	
temperature	float64	
SP O2	float64	(normal, abnormal)
Urine output	float64	
hematocrit	float64	
RBC	float64	(normal, abnormal)
MCH	float64	
MCHC	float64	
MCV	float64	
RDW	float64	
Leucocyte	float64	
Platelets	float64	
Neutrophils	float64	
Basophils	float64	
Lymphocyte	float64	
PT	float64	
INR	float64	
NT-proBNP	float64	(normal, abnormal)
Creatine kinase	float64	
Creatinine	float64	(normal, abnormal)
Urea nitrogen	float64	(normal, abnormal)
glucose	float64	
Blood potassium	float64	
Blood sodium	float64	
Blood calcium	float64	

Chloride	float64	
Anion gap	float64	(normal, abnormal)
Magnesium ion	float64	
PH	float64	
Bicarbonate	float64	
Lactic acid	float64	
PCO2	float64	
NT	float64	

Table 2: Dataframe Statistics

	count	mean	std	min	25%	50%	75%	max
ID	1177.0	150778.120646	29034.669513	100213.000000	125603.000000	151901.000000	176048.000000	199952.000000
outcome	1177.0	0.135089	0.341964	0.000000	0.000000	0.000000	0.000000	1.000000
age	1177.0	74.055225	13.434061	19.000000	65.000000	77.000000	85.000000	99.000000
gendera	1177.0	1.525064	0.499584	1.000000	1.000000	2.000000	2.000000	2.000000
BMI	962.0	30.188278	9.325997	13.346801	24.326461	28.312474	33.633509	104.970366
hypertensive	1177.0	0.717927	0.450200	0.000000	0.000000	1.000000	1.000000	1.000000
atrialfibrillation	1177.0	0.451147	0.497819	0.000000	0.000000	0.000000	1.000000	1.000000
CHD	1177.0	0.085811	0.280204	0.000000	0.000000	0.000000	0.000000	1.000000
diabetes	1177.0	0.421410	0.493995	0.000000	0.000000	0.000000	1.000000	1.000000
deficiencyanemias	1177.0	0.338997	0.473570	0.000000	0.000000	0.000000	1.000000	1.000000
depression	1177.0	0.118946	0.323863	0.000000	0.000000	0.000000	0.000000	1.000000
Hyperlipemia	1177.0	0.379779	0.485538	0.000000	0.000000	0.000000	1.000000	1.000000
Renal_failure	1177.0	0.365336	0.481729	0.000000	0.000000	0.000000	1.000000	1.000000
COPD	1177.0	0.075616	0.264495	0.000000	0.000000	0.000000	0.000000	1.000000
heart rate	1177.0	84.575848	15.929917	36.000000	72.540541	83.782609	95.608696	135.708333
Systolic blood pressure	1177.0	117.995035	17.249067	75.000000	105.500000	116.400000	128.485714	203.000000
Diastolic blood pressure	1177.0	59.534497	10.611747	24.736842	52.288136	58.642857	65.409091	107.000000
Respiratory rate	1177.0	20.801511	3.980800	11.137931	17.960000	20.454545	23.365854	40.900000
temperature	1177.0	36.677286	0.602630	33.250000	36.287037	36.661616	37.015873	39.132478
SP O2	1177.0	96.272900	2.285265	75.916667	95.000000	96.416667	97.888889	100.000000
Urine output	1177.0	1899.276512	1252.737309	0.000000	998.000000	1715.000000	2475.000000	8820.000000
hematocrit	1177.0	31.914014	5.202102	20.311111	28.160000	30.800000	35.012500	55.425000
RBC	1177.0	3.575010	0.626835	2.030000	3.120000	3.490000	3.900000	6.575000
MCH	1177.0	29.539939	2.619054	18.125000	28.250000	29.750000	31.240000	40.314286
MCHC	1177.0	32.864327	1.402302	27.825000	32.011111	32.985714	33.825000	37.011111
MCV	1177.0	89.903812	6.532629	62.600000	86.250000	90.000000	93.857143	116.714286
RDW	1177.0	15.952129	2.131643	12.088889	14.460000	15.506250	16.937500	29.050000
Leucocyte	1177.0	10.712948	5.229402	0.100000	7.440000	9.680000	12.740000	64.750000

Platelets	1177.0	241.504323	113.120623	9.571429	168.909091	222.666667	304.250000	1028.200000
Neutrophils	1177.0	80.113544	10.429365	5.000000	76.450000	80.500000	86.600000	98.000000
Basophils	1177.0	0.405569	0.410620	0.100000	0.200000	0.400000	0.405569	8.800000
Lymphocyte	1177.0	12.233024	8.083096	0.966667	7.100000	11.633333	14.700000	83.500000
PT	1177.0	17.481057	7.323904	10.100000	13.183333	14.700000	18.675000	71.271429
INR	1177.0	1.625465	0.826916	0.871429	1.142857	1.300000	1.714286	8.342857
NT-proBNP	1177.0	11014.130912	13148.664625	50.000000	2251.000000	5840.000000	14968.000000	118928.000000
Creatine kinase	1177.0	246.778456	1376.444776	8.000000	51.000000	110.000000	246.778456	42987.500000
Creatinine	1177.0	1.642846	1.279651	0.266667	0.940000	1.287500	1.900000	15.527273
Urea nitrogen	1177.0	36.298423	21.851545	5.357143	20.833333	30.666667	45.250000	161.750000
glucose	1177.0	148.796531	51.098648	66.666667	114.000000	137.375000	169.000000	414.100000
Blood potassium	1177.0	4.176646	0.414836	3.000000	3.900000	4.115385	4.400000	6.566667
Blood sodium	1177.0	138.890016	4.151347	114.666667	136.666667	139.250000	141.600000	154.736842
Blood calcium	1176.0	8.500894	0.572263	6.700000	8.148864	8.500000	8.869063	10.950000
Chloride	1177.0	102.283835	5.339733	80.266667	99.000000	102.500000	105.571429	122.526316
Anion gap	1177.0	13.925094	2.652732	6.636364	12.250000	13.666667	15.416667	25.500000
Magnesium ion	1177.0	2.120169	0.251532	1.400000	1.955556	2.092308	2.241667	4.072727
PH	1177.0	7.378532	0.058367	7.090000	7.350000	7.378532	7.410000	7.580000
Bicarbonate	1177.0	26.911766	5.167512	12.857143	23.454545	26.500000	29.875000	47.666667
Lactic acid	1177.0	1.853426	0.882849	0.500000	1.300000	1.833333	2.000000	8.333333
PCO2	1177.0	45.535382	11.008284	18.750000	39.000000	45.535382	47.272727	98.600000
NT	1177.0	1101.413091	1314.866463	5.000000	225.100000	584.000000	1496.800000	11892.800000

2) Data Cleaning

It is an important part of statistical data analysis. We have numeric classification for categorical data, so no transformation is required. Our data has a lot of null/missing values occurrence. So, we have replaced null values with the mean of the respective attribute to handle missing values.

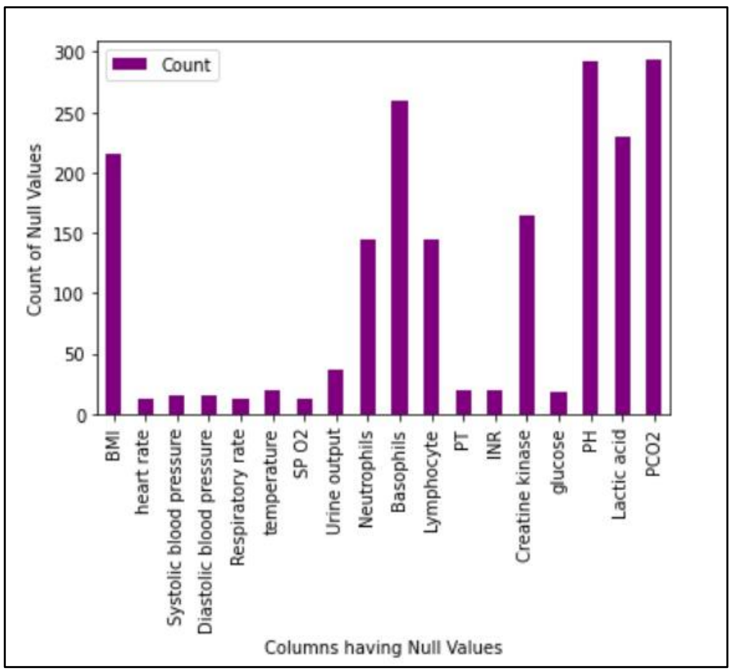


Figure 2: Columns having Null Values

3) Data Exploration

It is understanding the characteristics of the data and the behaviour of other variables towards the target variable, which is Survival/Output in our data.

Figure 3: Graphical representation of Diabetic patients' survival output

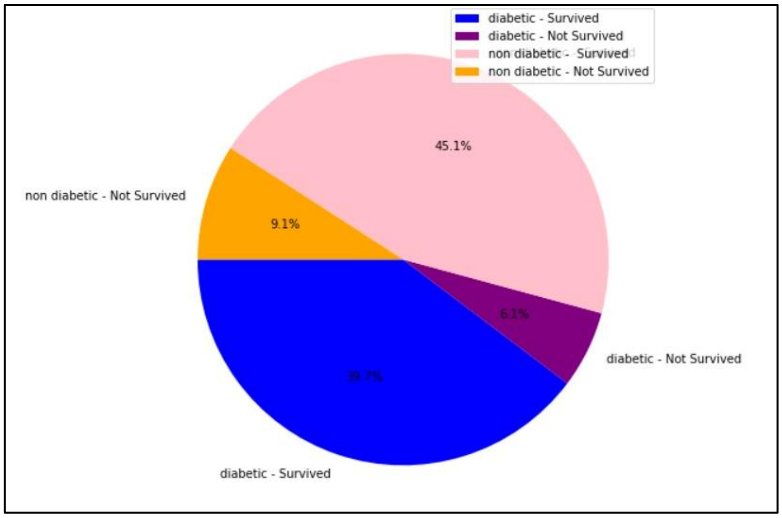


Figure 4: Graphical representation of Hyperlipemia patients' survival output

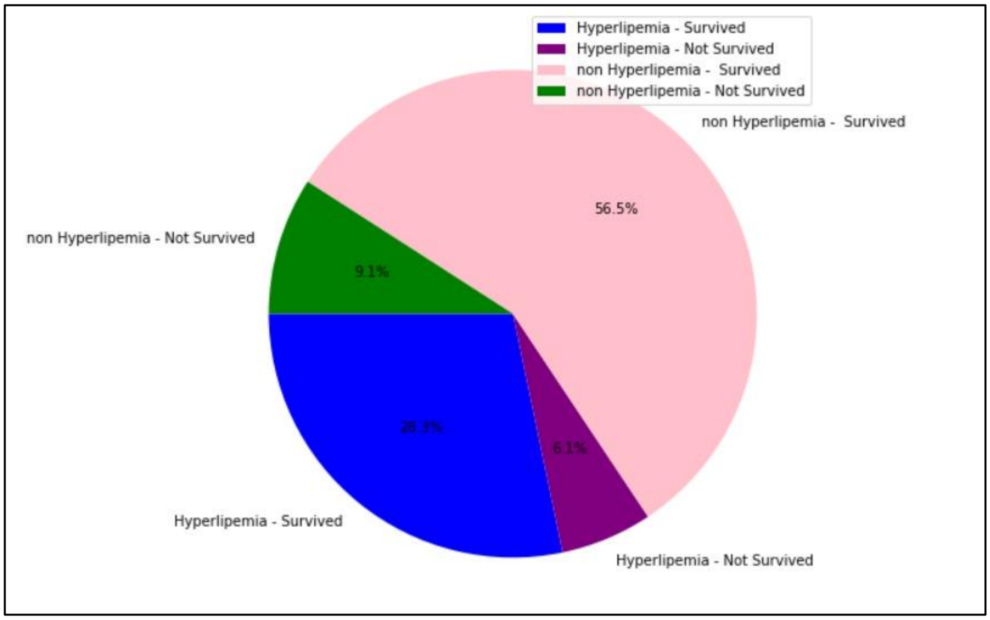


Figure 5: Graphical representation of Hypertensive patients' survival output

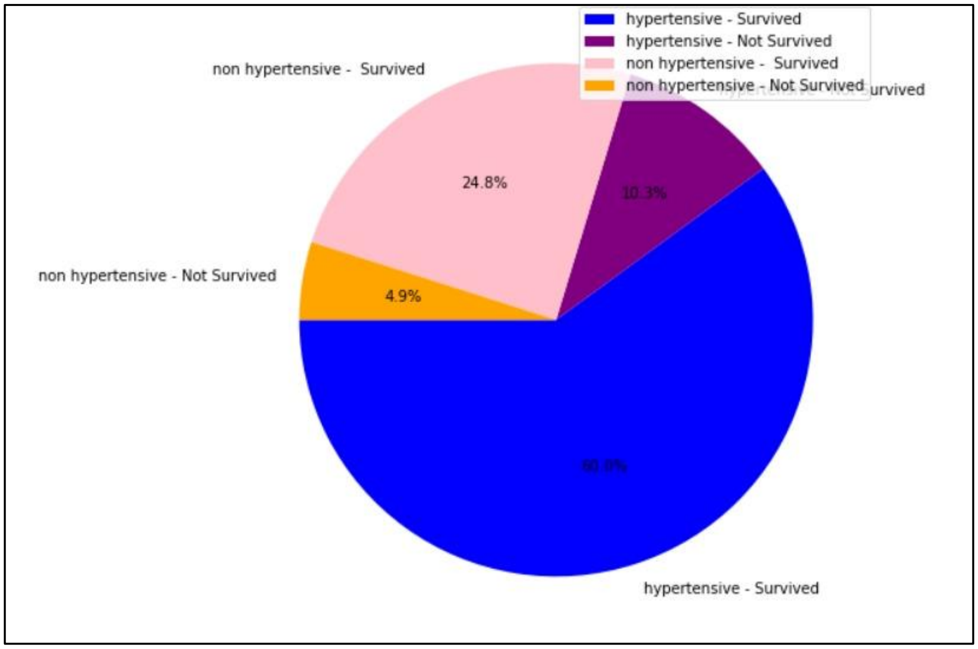


Figure 6: Graphical representation of Atrial Fibrillation patients' survival output

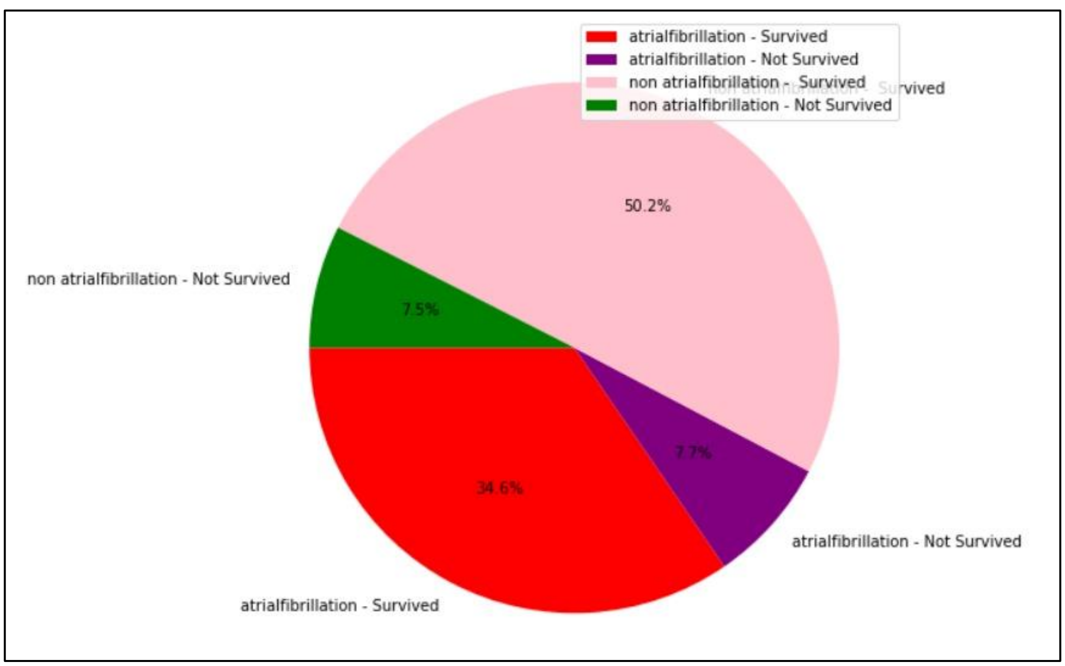
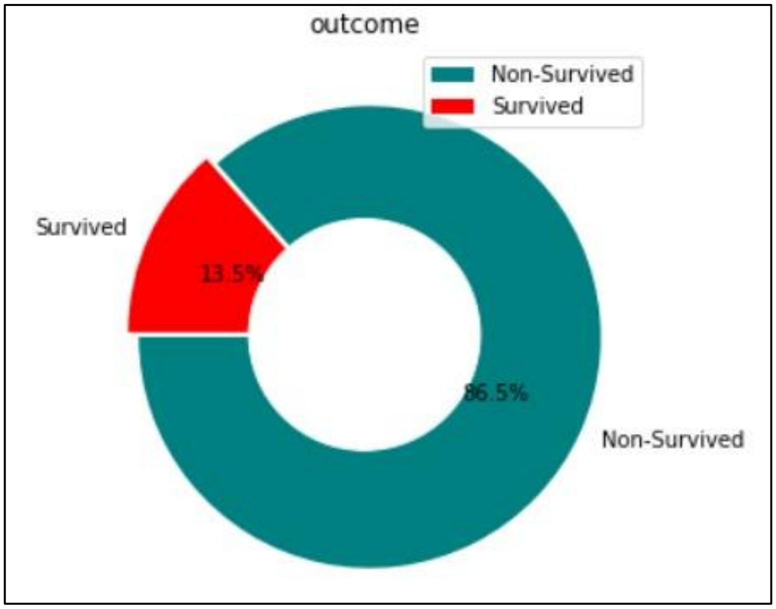


Figure 7: Graphical representation of survival



4) **Feature Selection**

To get better understanding of the relationship between variable/features and outcome class of patients' survival which will impact the model accuracy.

Figure 8: Correlation Coefficient values

	ID	outcome	age	gendera	BMI	hypertensive	atrialfibrillation	CHD	diabetes	deficiencyanemias	depression
ID	1.000000	0.040259	-0.026546	-0.030853	0.046694	-0.001704	-0.014781	0.035383	0.016410	-0.027295	0.029687
outcome	0.040259	1.000000	0.064270	-0.022324	-0.062086	-0.072635	0.101238	-0.014590	-0.050359	-0.099244	-0.060752
age	-0.026546	0.064270	1.000000	0.081705	-0.384185	0.177060	0.291003	0.037594	-0.089103	0.015099	-0.094543
gendera	-0.030853	-0.022324	0.081705	1.000000	0.024556	0.008776	-0.036957	-0.079159	-0.035943	0.080868	0.081415
BMI	0.046694	-0.062086	-0.384185	0.024556	1.000000	-0.032086	-0.118993	-0.063444	0.155664	-0.020931	0.024645
hypertensive	-0.001704	-0.072635	0.177060	0.008776	-0.032086	1.000000	0.006757	0.010040	0.129649	-0.005795	-0.043798
atrialfibrillation	-0.014781	0.101238	0.291003	-0.036957	-0.118993	0.006757	1.000000	-0.003449	-0.013032	-0.097414	-0.058864
CHD	0.035383	-0.014590	0.037594	-0.079159	-0.063444	0.010040	-0.003449	1.000000	0.008831	0.043327	0.046724
diabetes	0.016410	-0.050359	-0.089103	-0.035943	0.155664	0.129649	-0.013032	0.008831	1.000000	0.061274	0.005329
deficiencyanemias	-0.027295	-0.099244	0.015099	0.080868	-0.020931	-0.005795	-0.097414	0.043327	0.061274	1.000000	0.063983
depression	0.029687	-0.060752	-0.094543	0.081415	0.024645	-0.043798	-0.058864	0.046724	0.005329	0.063983	1.000000
Hyperlipemia	-0.020006	-0.053185	0.114893	-0.037522	-0.017770	0.225965	0.050439	0.047766	0.133406	0.027618	0.042347
Renal_failure	-0.046248	-0.108856	0.112246	-0.098146	-0.042829	0.193266	0.046120	0.025835	0.188646	0.149957	0.004649
COPD	-0.006602	-0.047223	-0.004048	-0.069055	0.013233	0.015029	-0.046189	0.004162	-0.074879	-0.028314	-0.005820
heart rate	0.017142	0.129293	-0.209241	-0.013628	-0.013943	-0.128110	-0.007047	-0.016652	-0.134587	-0.043540	0.055244
Systolic blood pressure	0.048500	-0.132362	-0.028960	0.084345	0.106668	0.142344	-0.118018	-0.084818	0.129911	0.045528	0.016516
Diastolic blood pressure	0.051116	-0.087077	-0.343134	-0.133641	0.152993	-0.022783	-0.072102	-0.005160	-0.054841	-0.107377	0.088710
Respiratory rate	-0.020959	0.116603	-0.044003	-0.042068	-0.044051	-0.055710	-0.029758	0.007503	-0.094196	-0.034283	-0.009542
temperature	-0.015350	-0.092496	-0.211713	-0.012786	0.088705	0.016209	-0.156367	-0.061600	0.025467	0.009350	0.039880
SP O2	0.032610	-0.070938	0.057754	0.024066	-0.177589	0.061880	0.058226	0.056612	0.068126	0.082916	0.026070
Urine output	0.038764	-0.171332	-0.249722	-0.138575	0.281175	-0.034639	-0.153664	0.012823	0.053307	-0.035093	0.025662
hematocrit	-0.000650	-0.016786	-0.019583	-0.114740	0.133204	-0.028032	0.022021	0.003097	-0.067017	-0.362072	0.007739
RBC	0.006631	-0.024182	-0.053557	-0.096151	0.165467	-0.008640	0.016521	0.000317	-0.040542	-0.315511	0.000883

Analysing correlation coefficient values of different disease with features helped us select the risk factor.

Like in our data, Atrial fibrillation has moderate positive correlation with age (0.29) so we have analysed the risk of having atrial fibrillation in elderly patients'.

Renal failure had significant positive correlation with Creatinine, Urea nitrogen, anion gap and NTproBNP so we have tried analysing the abnormalities in Creatinine values of patients having renal failure.

CHAPTER 6: OBJECTIVE ANALYSIS

OBJECTIVE 1 – Data analysis using correlation to assess degree of association between comorbidities (Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) and age.

- **Method Used –** Correlation Coefficients
- **Python Code -**

```
In [177]: s=[]
d=new_df[['age','hypertensive','atrialfibrillation','deficiencyanemias',
          'CHD','COPD','diabetes','Hyperlipemia','depression','Renal_failure','outcome']]
for col in d.columns:
    c=d['age'].corr(d[col])
    s.append(c)
dep=pd.DataFrame({'Columns': d.columns, 'Correlation Coefficents': s})
dep
```

Figure 9 Python Code for Correlation Coefficients between Comorbidities and age.

- **Results Using Python Programming -**

	Columns	Correlation Coefficents
0	age	1.000000
1	hypertensive	0.177060
2	atrialfibrillation	0.291003
3	deficiencyanemias	0.015099
4	CHD	0.037594
5	COPD	-0.004048
6	diabetes	-0.089103
7	Hyperlipemia	0.114893
8	depression	-0.094543
9	Renal_failure	0.112246
10	outcome	0.064270

Figure 10 Correlation Coefficients between Comorbidities and age.

- **Analysis :** From the above result we can see that age have moderate positive correlation with atrial fibrillation(0.29), and weak positive correlation with hypertensive(0.17) , Renal failure(0.11) , Hyperlipemia(0.11)

OBJECTIVE 2 - To estimate the extent of association of Atrial Fibrillation in the age group (>60) i.e., elderly patients (exposed group).

- **Relative Risk Analysis:** Risk ratio (RR) is used to compare the risk of having Atrial Fibrillation among elderly patients' age: >60) with the risk among the remaining patients (age: <60)
- The group of primary interest here is age (>60) so it is labelled as exposed group, and the other comparison group, age (<60) is labelled the unexposed group.

Age Group (years)	(Having Atrial Fibrillation) 1	(Not Having Atrial Fibrillation) 0	Total
61-100 (Exposed)	496	490	986
19-60 (Unexposed)	35	156	191
Total	531	646	1177

Table 3: showing count of patients who have atrial fibrillation and not have atrial fibrillation on the basis of age

- Risk of Atrial Fibrillation among age (61-100) = 496/986 = 0.5030 = 50.30%
- Risk of Atrial Fibrillation among other age group (<61) = 35/191 = 0.1832 = 18.32%
- Risk Ratio = o.5030/0.1832 = 2.729
- **Analysis -** Risk ratio is greater than 1.0, indicating a increased risk for the exposed (>60) age patients. Patients with age (>60) were more than thrice (approximately as, 2.729) as likely to develop Atrial Fibrillation as were patients in other age group (<60)**.**

Our research shows, that Atrial fibrillation is one of the most common diseases in elderly patients (>61 age) and its prevalence increases with age. The primary factors that contribute to the high risk of AF in the elderly, are coronary artery disease, aging heart and systemic diseases like hypertension.

OBJECTIVE 3 - To determine the dependency of the diseases(Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) on gender for the patients with admitted in hospital.

- **Null Hypothesis (H_0) -** The diseases are independent on sex.
- **Alternate Hypothesis (H_1) –** The diseases are dependent on sex.
- **Method Used– CHI- SQUARED TEST (χ^2)**, for determining the dependencies of categorical varibales namely diseases (Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) and Sex (Males and females).
- **Methodology-**
    - **Forming the matrix showing the count of females and males having above mentioned diseases.**

Disease	Male	Female	Total
Hypertension	399	446	845
Atrial fibrillation	263	268	531
CHD	61	40	101
Diabetes	246	250	496
Anaemia	167	232	399
Renal Failure	232	198	430
COPD	53	36	89
Total	1421	1470	2891

Table 4: Table 4: Data analysis using chi squared test of independence among different sex for the patients admitted in hospital.

- **Test Statistics**

$$\chi^2 = \sum\sum\frac{(O_i - E_j)^2}{E_j}$$

Where E_{ij} is the expected value of the i^{th}row and j^{th} column
O_{ij} is the observed value of the i^{th} row and j^{th} column

$$E_{ij} = \frac{(Rowtotal)(columntotal)}{N}$$

Where: N is the grand total

- **<u>Python Programme-</u>**

Figure 11 Programmed code for Chi- Squared test.

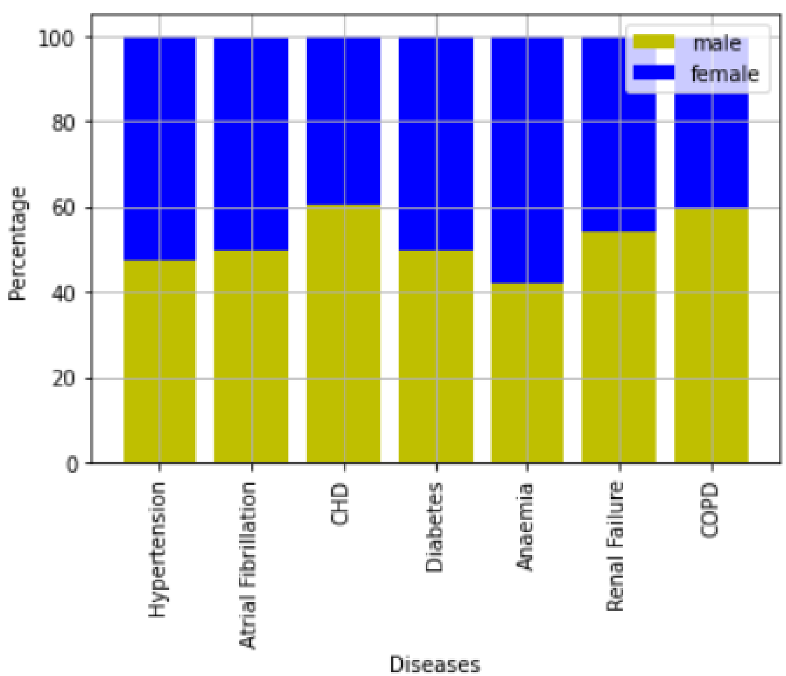
```
In [228]: from scipy.stats import chi2_contingency

          # defining the table
          data = [[399,446], [263,268],[61,40],
                  [246,250],[167,232],[232,198],
                  [53,36]]
          stat, p, dof, expected = chi2_contingency(data)

          # interpret p-value
          alpha = 0.05
          print("p value is " + str(p))
          if p <= alpha:
              print('Dependent (reject H0)')
          else:
              print('Independent (H0 holds true)')

          p value is 0.0008808133077103179
          Dependent (reject H0)
```

Figure 12: Percentage Distribution of diseases by Sex



- **Analysis –** P value is less than 0.05 which implies we should reject the null hypothesis which inferred as diseases are dependent on sex, so that sex wise comparison of known diseases could be done to prepare an efficient model.
- Both the genders have almost equal percentage distribution w.r.t to all the diseases.

OBJECTIVE 4- To determine the dependency of the diseases (Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) on age for the patients admitted in hospital and which age group has maximum disease incidence.

- **Null Hypothesis (H_0) -** The diseases are independent of age.
- **Alternate Hypothesis (H_1) –** The diseases are dependent of age.
- **Method Used – CHI- SQUARED TEST(χ^2)**, for determining the dependencies of categorical variables namely diseases (Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) and age group.
- **Methodology**
    - **Forming the matrix showing the count of patients in respective age group having above mentioned diseases.**

Table 5: 9 Data analysis using chi squared test of independence among different age group for the patients admitted in hospital

Disease	19-39	40-60	61-81	82-102	Total
Hypertension	6	107	401	331	845
Atrial fibrillation	0	35	250	266	551
CHD	0	5	63	33	101
Diabetes	2	93	254	166	515
Depression	3	31	68	38	140
Anaemia	8	63	178	150	399
Renal failure	4	48	208	170	430
COPD	0	11	53	25	89
Total	23	393	1475	1179	3070

- **Python Code-**

In [263]:
```python
from scipy.stats import chi2_contingency

# defining the table
data = [[6,107,401,331], [0,35,250,266],
        [0,5,63,33], [2,93,254,166],
        [3,31,68,38], [8,63,178,150],
        [4,48,208,170],[0,11,53,25]]
stat, p, dof, expected = chi2_contingency(data)

# interpret p-value
alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
```

```
p value is 1.264617037354978e-12
Dependent (reject H0)
```

```
Count of Patients having comborbidities and belongs to the age group of 19-39 are :  12
Count of Patients having comborbidities and belongs to the age group of 40-60 are :  157
Count of Patients having comborbidities and belongs to the age group of 61-81 are :  533
Count of Patients having comborbidities and belongs to the age group of 82-102 are :  419
```

- **Analysis-** **'**P' value is less than 0.05 which implies we should reject the null hypothesis which inferred for the diseases to be dependent on age.
- patients of age lying in the range (61<= age <=81) has most comorbidities.

OBJECTIVE 5- To predict if a patient has Renal Failure using feature scaling

- **Feature Scaling-** checking if there exists a significant dependence of renal failure with other diseases and laboratory test.

- **Correlation Coefficients Using Python-**

Figure 13: python code for correlation coefficients.

```
]:  s=[]
    for col in new_df.columns:
        c=new_df['Renal_failure'].corr(new_df[col])
        s.append(c)
    s
    dep=pd.DataFrame({'Columns': new_df.columns, 'Correlation Coefficents': s})
    dep
```

Out of all features in data, the significant correlation of Renal failure exits with-

Figure 14: Python Code for showing significant correlation coefficients.

```
s=[]
d=new_df[['hypertensive','diabetes','Anion gap'
          ,'Urea nitrogen','Creatinine','Hyperlipemia','NT-proBNP','Renal_failure']]
for col in d.columns:
    c=d['Renal_failure'].corr(d[col])
    s.append(c)
dep=pd.DataFrame({'Columns': d.columns, 'Correlation Coefficents': s})
dep
```

Figure 15: Lab test variables having significant correlation with Renal Failure.

	Columns	Correlation Coefficents
0	hypertensive	0.193266
1	diabetes	0.188646
2	Anion gap	0.247680
3	Urea nitrogen	0.424517
4	Creatinine	0.450427
5	Hyperlipemia	0.097050
6	NT-proBNP	0.254034
7	Renal_failure	1.000000

- **Logistic Regression to predict Renal Failure**

To predict if a patient with HF admitted in the hospital has Renal Failure or not we are selecting those risk factors from the dataset which shows significant correlation coefficients with Renal failure.

Most influential risk factors are hypertension, Diabetes, Anion gap, Urea Nitrogen, creatinine, Hyperlipemia, NT-pro BNP.

We train the supervised logistic regression model using these risk factors.

Input Variables - Diabetes, Anion gap, Urea Nitrogen, creatinine, Hyperlipemia, NT-pro BNP

Output Variable - Renal Failure

Regression Model - logistic regression

Splitting the dataset such that training data is used to train the supervised regression model which contains 80% of dataset and out of all training data, 75% data is used for training and remaining percent of data is used for validation, Validation data is used for the evaluation of the regression model to have a better accuracy, testing data which consist of 20% of dataset

Figure 17: Python Code for splitting data into training and testing data

```
training_data, testing_data=train_test_split(new_df, train_size=0.8, test_size=0.2, random_state=42, shuffle=True)
testing_data
X_test=testing_data[['hypertensive','diabetes','Anion gap',
                        'Hyperlipemia','Creatinine','NT-proBNP']]
Y_test=testing_data['Renal_failure']
training_data
```

Figure 16: Python Code for splitting data into training and testing data

```
training_data, validation_data=train_test_split(training_data, train_size=0.75, test_size=0.25, random_state=42, shuffle=True)
training_data
X_train=training_data[['heart rate','Respiratory rate','Urea nitrogen','Leucocyte',
                       'RDW', 'Anion gap','Lactic acid','Blood potassium',
                       'atrialfibrillation','PT','NT-proBNP']]
Y_train=training_data['outcome']

validation_data

X_validation=validation_data[['heart rate','Respiratory rate','Urea nitrogen','Leucocyte',
                       'RDW', 'Anion gap','Lactic acid','Blood potassium',
                       'atrialfibrillation','PT','NT-proBNP']]
Y_validation=validation_data['outcome']
```

- **Fitting a Logistic regression model**

Figure 18: Logistic Regression model with 5 iterations

```
# Logistic Regression with validation data at 5  number of iterations

model=LogisticRegression(max_iter=5,solver='liblinear')
model.fit(X_train,Y_train)
b=model.predict(X_valid)
c=accuracy_score(b,Y_valid)
print("Accuracy score of training model = ",c) # accuracy is 61.58%

Accuracy score of training model =  0.615819209039548
```

Figure 19: Logistic Regression model with 30 iterations

```
model=LogisticRegression(max_iter=30,solver='liblinear')
model.fit(X_train,Y_train)
b=model.predict(X_valid)
c=accuracy_score(b,Y_valid)
print("Accuracy score of training model = ",c) # accuracy is 80.79%

Accuracy score of training model =  0.807909604519774
```

From the above accuracy results for fitting the logistic regression model with different iterations, we can see that accuracy of validation data is increasing with the increase in number of iterations in the training dataset but out of several iterations, the best accuracy is 80.79% with 30 number of iterations.

- **Testing the Regression Model**

Figure 20: Logistic Regression Model on test data with 50 iterations

```
: model=LogisticRegression(max_iter=50,solver='liblinear')
  model.fit(X_train,Y_train)
  b=model.predict(X_test)
  c=accuracy_score(b,Y_test)
  print("Accuracy score of training model = ",c) # accuracy is 76.69%

  Accuracy score of training model =  0.7669491525423728
```

The model is trained with significant accuracy of 80.79%, so we can use this model to predict the renal failure, providing testing data to the trained supervised regression model to predict the renal failure and this model predicted the outcome of renal failure with the accuracy of 76.69%.

- **Model Evaluation**

Classification Matrices

True Positives = 126,

True Negatives = 54,

False Positives = 21

False Negatives = 35

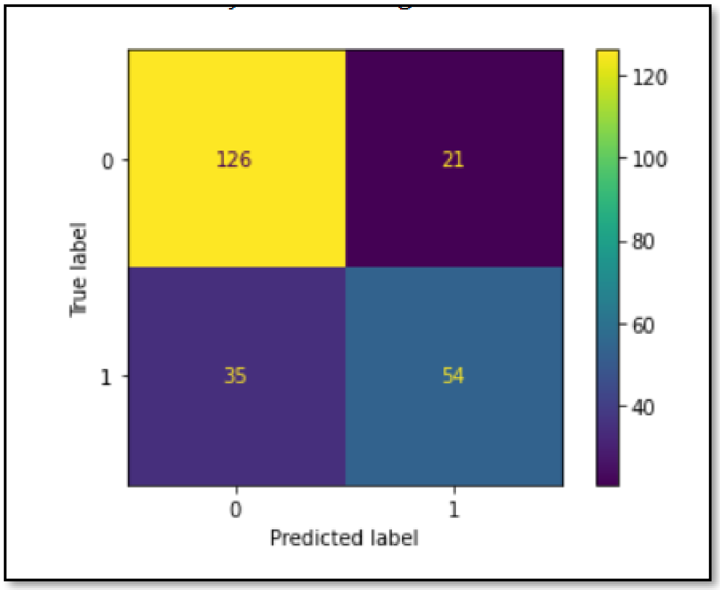


Figure 21 Confusion Matrix

Accuracy: The accuracy on testing data is: 0.72

Precision: The precision on testing data is: 0.61

Recall: The recall on testing data is: 0.32

Misclassification Error: 0.237

The accuracy of predicting renal failure is 72% with 23.7% error.

OBJECTIVE 6(a)- Implementing Linear Regression to predict Creatinine level in females based on Renal failure output.

- **Using Simple Regression Model to predict the Creatinine level on the basis of Renal Failure output**

  To predict the Creatinine of female patients admitted in hospital we are selecting renal failure from the dataset which shows significant correlation coefficients with Renal failure with creatinine. We train the supervised linear regression model, to predict Creatinine value based on renal failure.

  Independent Variable/ Categorical Variable: Renal Failure (1: Having renal failure, 0: Not having renal failure)
  Dependent Variable / Continuous Variable: Creatinine
  Regression Model Used: Linear Regression

- **Training Of Regression Model Using Python**

Figure 22: Python Code for splitting the data into training and testing dataset.

```
new_df_1=new_df.query('gendera==2')
training_data, testing_data=train_test_split(new_df_1, train_size=0.8, test_size=0.2, random_state=42, shuffle=True)
testing_data
X_test=testing_data[['Renal_failure','hypertensive']]
Y_test=testing_data['Creatinine']

training_data, validation_data=train_test_split(training_data, train_size=0.75, test_size=0.25, random_state=42, shuffle=True)
training_data
X_train=training_data[['Renal_failure','hypertensive']]
Y_train=training_data['Creatinine']

validation_data

X_validation=validation_data[['Renal_failure','hypertensive']]
Y_validation=validation_data['Creatinine']
```

Figure 23 Python Code for fitting the model on test data

```
model=LinearRegression()
model.fit(X_train[['Renal_failure']],Y_train)
b=model.predict(X_test[['Renal_failure']])
b
s=model.coef_
c=model.intercept_
print("Coefficient is = " ,s)
print("Intercept is  = " ,c)
mse=mean_squared_error(b,Y_test)
rmse=np.sqrt(mse)
print("Root Mean Square valuue is = ",rmse)
pd.DataFrame({'Actual ': Y_test, 'Predicted': b})
```

Figure 24: Output of RMSE

```
Coefficient is =  [1.3392]
Intercept is  =  1.1563995659023436
Root Mean Square valuue is =  0.7332002659496762
```

We have trained the linear regression model and we get the best fitted regression line equation be :

y= 1.339 *x + 1.1560 + 0.7332(Root mean square error)

with the help of regression line equation ($y^{\wedge}$=1.339*x +1.1560) we can predict the Creatinine level of females

	Actual	Predicted
84	0.844444	1.1564
1106	0.800000	1.1564
151	0.880000	1.1564
605	1.400000	2.4956
200	0.612500	1.1564
...	...	...
10	1.820000	1.1564
186	1.228571	1.1564
208	0.385714	1.1564
303	1.378571	1.1564
359	0.400000	1.1564

124 rows × 2 columns

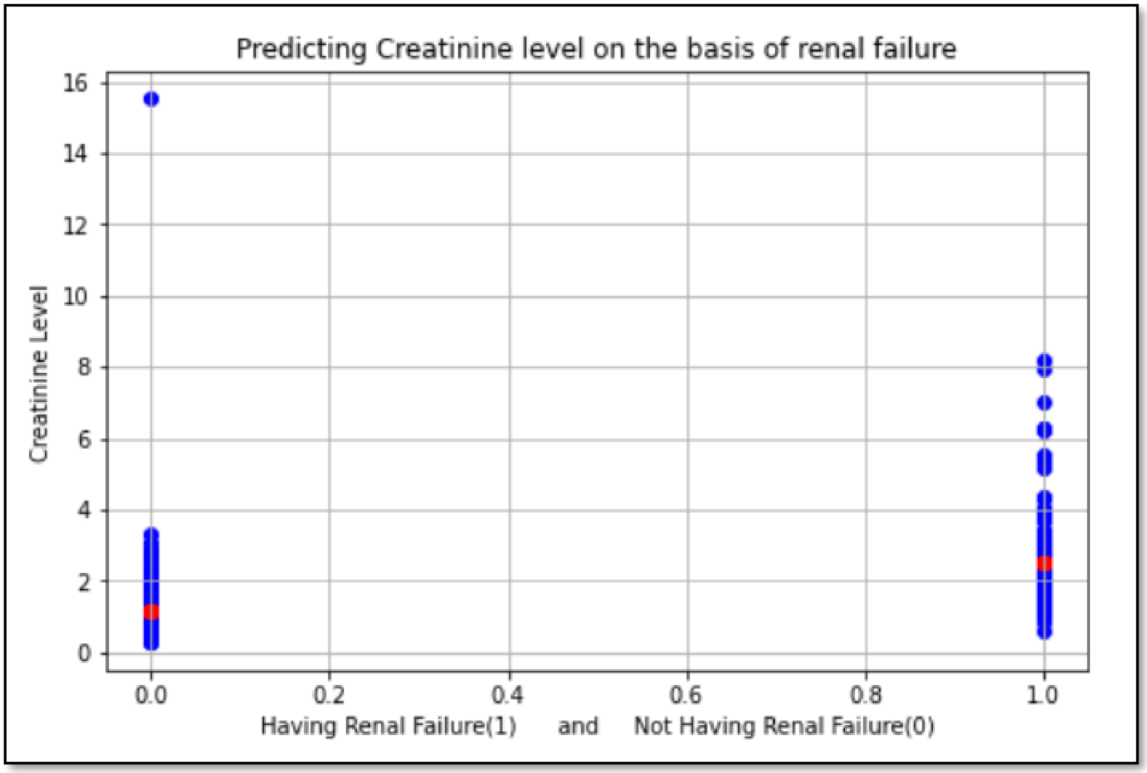
Figure 25 Predictions of Creatinine level of females.

- **Graphical Analysis**

Figure 26: Python code for graph plotting

```
plt.figure(figsize=(8,5))
plt.scatter(X_train['Renal_failure'],Y_train,color='b')
plt.scatter(X_test['Renal_failure'],b,color='r')
plt.grid()
plt.xlabel("Having Renal Failure(1)      and     Not Having Renal Failure(0)")
plt.ylabel("Creatinine Level")
plt.title("Predicting Creatinine level on the basis of renal failure")
```

. Figure 27 Graph showing Creatinine in females who have renal failure and who don't have renal failure.



From the graph we can infer that creatinine in females who have renal failure is 2.4956 mg/dl , Creatinine in females who didn't have renal failure is 1.564 mg/dl. **While the normal creatinine in females is 1.2 mg/dl.**

OBJECTIVE 6(b)- Implementing Linear Regression to predict Creatinine in males based on renal failure output

- **Using Simple Regression Model to predict the Creatinine level based on Renal Failure output**

  To predict the Creatinine of male patients admitted in hospital we are selecting renal failure from the dataset which shows significant correlation coefficients with Renal failure with creatinine. We train the supervised linear regression model, to predict Creatinine value based on renal failure.

  Independent Variable/ Categorical Variable: Renal Failure (1: Having renal failure, 0: Not having renal failure)
  Dependent Variable / Continuous Variable: Creatinine
  Regression Model Used: Linear Regression

- **Training Of Regression Model Using Python**

Figure 28: Python Code for splitting the data into training and testing dataset.

```
new_df_1=new_df.query('gendera==1')
training_data, testing_data=train_test_split(new_df_1, train_size=0.8, test_size=0.2, random_state=42, shuffle=True)
testing_data
X_test=testing_data[['Renal_failure','hypertensive']]
Y_test=testing_data['Creatinine']

training_data, validation_data=train_test_split(training_data, train_size=0.75, test_size=0.25, random_state=42, shuffle=True)
training_data
X_train=training_data[['Renal_failure','hypertensive']]
Y_train=training_data['Creatinine']

validation_data

X_validation=validation_data[['Renal_failure','hypertensive']]
Y_validation=validation_data['Creatinine']
```

Figure 29 Python Code for fitting the model on test data

```
model=LinearRegression()
model.fit(X_train[['Renal_failure']],Y_train)
b=model.predict(X_test[['Renal_failure']])
b
s=model.coef_
c=model.intercept_
print("Coefficient is = " ,s)
print("Intercept is  = " ,c)
mse=mean_squared_error(b,Y_test)
rmse=np.sqrt(mse)
print("Root Mean Square valuue is = ",rmse)
pd.DataFrame({'Actual ': Y_test, 'Predicted': b})
```

Figure 30: Output of RMSE

```
Coefficient is =  [0.87137131]
Intercept is  =  1.3294593567688677
Root Mean Square valuue is =  1.5238857162448864
```

We have trained the linear regression model and we get the best fitted regression line equation be: **y= 0.871 *x + 1.329 + 1.524(Root mean square error)**

with the help of regression line equation (**y^=0.871 *x + 1.329**) we can predict the Creatinine level of males

	Actual	Predicted
326	0.971429	1.329459
954	12.837500	2.200831
120	1.185714	2.200831
497	2.336364	2.200831
155	1.271429	1.329459
...	...	...
40	4.133333	2.200831
772	1.061538	1.329459
58	2.566667	2.200831
792	1.807143	2.200831
771	4.360000	2.200831

112 rows × 2 columns

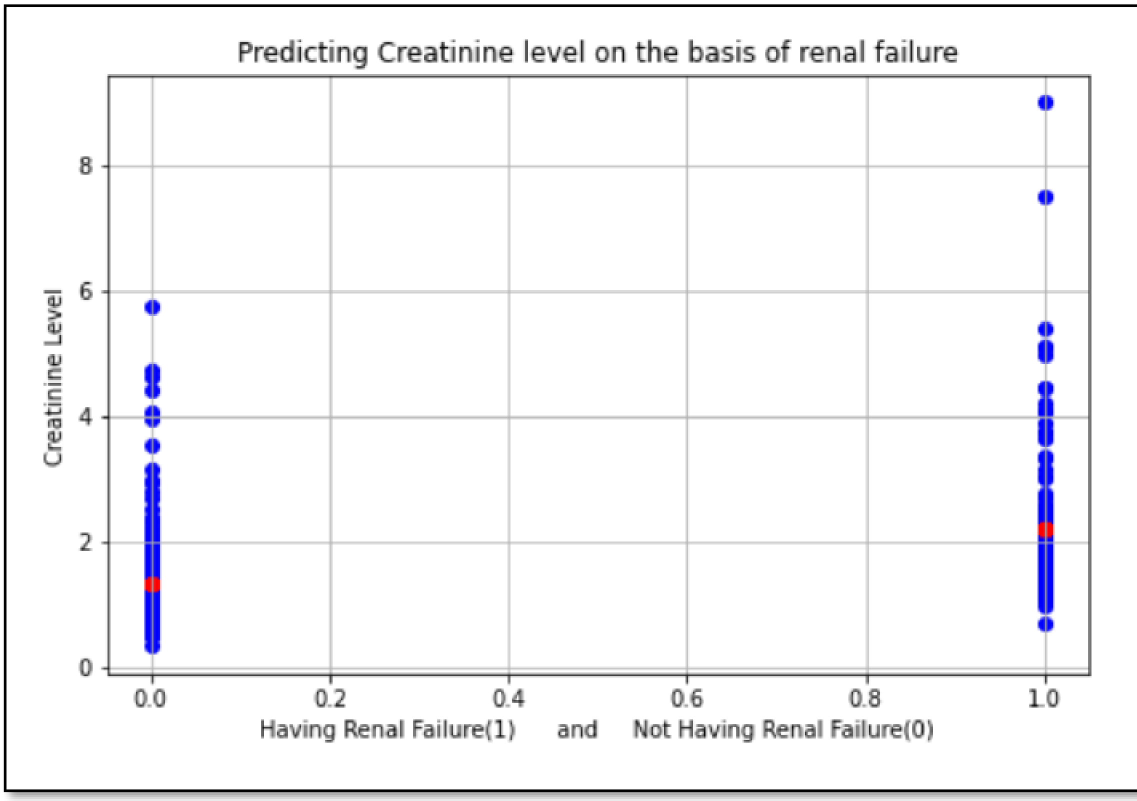
Figure 31 Predictions of Creatinine level of females.

- **Graphical Analysis**

Figure 32: Python code for graph plotting

```
plt.figure(figsize=(8,5))
plt.scatter(X_train['Renal_failure'],Y_train,color='b')
plt.scatter(X_test['Renal_failure'],b,color='r')
plt.grid()
plt.xlabel("Having Renal Failure(1)      and     Not Having Renal Failure(0)")
plt.ylabel("Creatinine Level")
plt.title("Predicting Creatinine level on the basis of renal failure")
```

. Figure 33 Graph showing Creatinine in females who have renal failure and who don't have renal failure.



From the graph we can infer that creatinine in males who have renal failure is 2.2mg/dl , Creatinine in males who didn't have renal failure is 1.32 mg/dl. **While the normal creatinine in females is 1.4 mg/dl.**

CHAPTER 7: RESULTS & DISCUSSIONS

1. We have assessed the degree of association between comorbidities (Hypertension, Ischemic Heart Disease, Atrial fibrillation, Diabetes, Depression, Anaemia, Hyperlipidaemia, Chronic Kidney Disease and Chronic Obstructive Pulmonary Disease) and age using correlation coefficient and the results are as such: moderate positive correlation with atrial fibrillation (0.29), and weak positive correlation with hypertensive (0.17), Renal failure (0.11), Hyperlipemia (0.11).

2. While estimating the extent of association of Atrial Fibrillation in the age group (>60) i.e., elderly patients (exposed group), we have found that Risk ratio is greater than 1.0, indicating a increased risk for the exposed (>60) age patients. Patients with age (>60) were more than thrice (approximately as, 2.729) as likely to develop Atrial Fibrillation as were patients in other age group (<60)**.**

3. While determining the dependency of diseases on age we have analysed those patients of age lying in the range (61<= age <=81) has most comorbidities.

4. While predicting if a patient has renal failure based on the significant risk factors i.e., Creatinine, accuracy obtained is about 72%.

CHAPTER 8: CONCLUSION

To manage the high rates of in-hospital mortality of heart failure patients affected with several other comorbidities and abnormal lab test values, we tried to analyse the dataset using important statistical techniques including machine learning algorithms and graphical analysis in order to generate an effective and accurate model. The results obtained act as a key for gaining insights from the respective dataset, forecasting the survival status of a new admitted patient, and this analysis can be helpful in offering safety and preventive measures for the healthcare processes towards HF patients in ICU.

CHAPTER 9: REFERENCES

[1]. Agarwal, R. (2012) "Multiple comparisons, interaction effects, and statistical inference: Lessons from chronic kidney disease progression among blacks," *Kidney International*, 81(6), pp. 516–519. Available at: https://doi.org/10.1038/ki.2011.442.

[2]. Han, D. *et al.* (2022) *Early prediction of in-hospital mortality in patients with congestive heart failure in Intensive Care Unit: A retrospective observational cohort study*, *BMJ Open*. British Medical Journal Publishing Group. Available at: https://bmjopen.bmj.com/content/12/7/e059761 (Accessed: December 17, 2022).

[3]. Ifraz, G.M. *et al.* (2021) *Comparative analysis for prediction of kidney disease using intelligent machine learning methods*, *Computational and Mathematical Methods in Medicine*. Hindawi. Available at: https://www.hindawi.com/journals/cmmm/2021/6141470/ (Accessed: December 17, 2022).

[4]. Oviyashri (2022) *Chronic_kidney_disease_prediction*, *Kaggle*. Kaggle. Available at: https://www.kaggle.com/code/oviyashri/chronic-kidney-disease-prediction (Accessed: December 17, 2022).

[5]. panelB.IsmailaManjulaAnilbPersonEnvelope, A.links open overlay *et al.* (2014) *Regression methods for analyzing the risk factors for a life style disease among the young population of India*, *Indian Heart Journal*. Elsevier. Available at: https://www.sciencedirect.com/science/article/pii/S0019483214002028 (Accessed: December 17, 2022).

[6]. *Principles of Epidemiology* (2012) *Centers for Disease Control and Prevention*. Centers for Disease Control and Prevention. Available at: https://www.cdc.gov/csels/dsepd/ss1978/lesson3/section5.html#:~:text=A%20risk%20ratio%20(RR)%2C,attack%20rate)%20in%20group%202. (Accessed: December 17, 2022).

[7]. Ragan, A. (2018) *Taking the confusion out of confusion matrices*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/taking-the-confusion-out-of-confusion-matrices-c1ce054b3d3e (Accessed: December 17, 2022).

[8]. Shahane, S. (2021) *In hospital mortality prediction*, *Kaggle*. Available at: https://www.kaggle.com/datasets/saurabhshahane/in-hospital-mortality-prediction (Accessed: December 17, 2022).

[9]. *Solve Xerox Mortality Prediction Challenge* (no date) *HackerRank*. Available at: https://www.hackerrank.com/contests/xerox-research-innovation-challenge-2015/challenges/xerox-predict-mortality (Accessed: December 17, 2022).