PREDICTION OF LIVER CIRRHOSIS AND KIDNEY DISEASE

A PROJECT REPORT

Submitted by

MOHANA PRIYA S R [REGISTER NO:211420104164]

INDHUMATHI V [REGISTER NO: 211420104101]

DHEEKSHA J [REGISTER NO: 211420104063]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **" PREDICTION OF LIVER CIRRHOSIS AND KIDNEY DISEASE "** is the bonafide work of " **MOHANA PRIYA S R[REGISTER NO:211420104164] , INDHUMATHI V[211420104101],DHEEKSHA J[REGISTER NO:211420104063] "** who carried out the project work under my supervision.

Signature of the HOD with date

Dr L.JABASHEELA M.E., Ph.D.,
PROFESSOR AND HEAD,
Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai – 123.

Signature of the Supervisor with date

Mrs.P. VIJAYALAKSHMI M.Tech.,
SUPERVISOR
ASSISTANT PROFESSOR(Grade-1)
Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai – 123.

Certified that the above candidates were examined in the End Semester Project Viva-Voice Examination held on ....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **MOHANAPRIYA S R(211420104164), INDHUMATHI V(211420104101), DHEEKSHA J(211420104063**) hereby declare that this projectreport titled **" PREDICTION OF LIVER CIRRHOSIS AND KIDNEY DISEASE"** , under the guidance of **Mrs.P.VIJAYALAKSHMI., M.Tech.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

Dheeksha J

Indhumathi V

Mohana priya S R

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

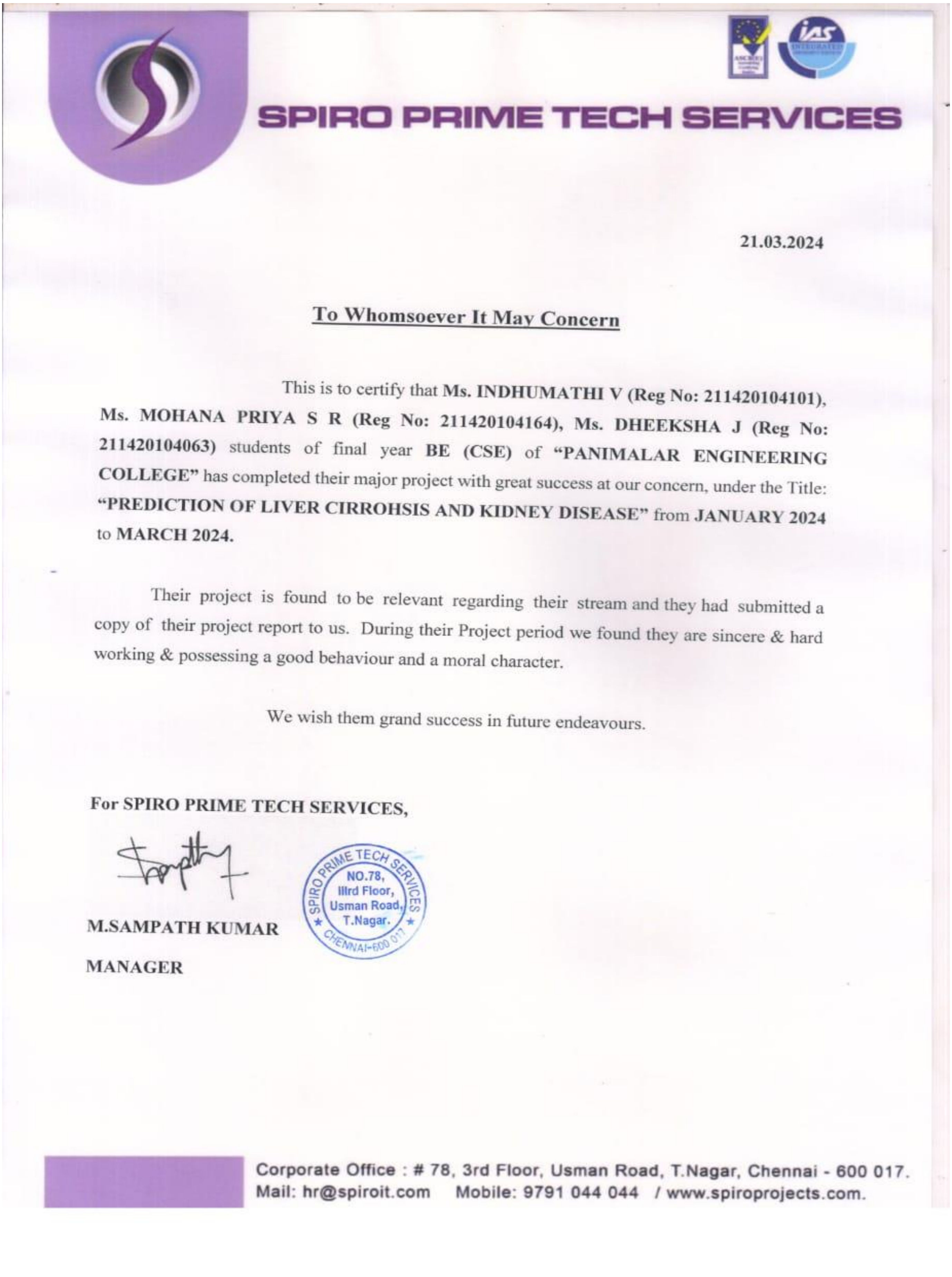
We express our heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Project Coordinator,Dr. KAVITHA SUBRAMANI,M.E.,Ph.D.,** and **Project Guide, Mrs P.VIJAYALAKSHMI.,M.Tech.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

Dheeksha J

Indhumathi V

Mohana Priya S R



SPIRO PRIME TECH SERVICES

21.03.2024

To Whomsoever It May Concern

This is to certify that **Ms. INDHUMATHI V (Reg No: 211420104101), Ms. MOHANA PRIYA S R (Reg No: 211420104164), Ms. DHEEKSHA J (Reg No: 211420104063)** students of final year **BE (CSE)** of **"PANIMALAR ENGINEERING COLLEGE"** has completed their major project with great success at our concern, under the Title: **"PREDICTION OF LIVER CIRROHSIS AND KIDNEY DISEASE"** from **JANUARY 2024** to **MARCH 2024.**

Their project is found to be relevant regarding their stream and they had submitted a copy of their project report to us. During their Project period we found they are sincere & hard working & possessing a good behaviour and a moral character.

We wish them grand success in future endeavours.

For SPIRO PRIME TECH SERVICES,

SPIRO PRIME TECH SERVICES
NO.78,
IIIrd Floor,
Usman Road,
T.Nagar.
CHENNAI-600 017

M.SAMPATH KUMAR

MANAGER

Corporate Office : # 78, 3rd Floor, Usman Road, T.Nagar, Chennai - 600 017.
Mail: hr@spiroit.com Mobile: 9791 044 044 / www.spiroprojects.com.

ABSTRACT

Liver Cirrhosis and kidney disease are both critical health conditions that can greatly benefit from the application of machine learning techniques for early detection and improved patient care. Cirrhosis is characterized by the gradual deterioration of liver function, while kidney disease can lead to impaired renal function.Leveraging supervised machine learning models on comprehensive clinical and laboratory datasets can help identify key biomarkers and patterns associated with these diseases. A dataset comprising clinical and laboratory features of patients with and without Cirrhosis and kidney disease is collected. By analyzing patient data, these models can assist healthcare professionals in diagnosing these conditions earlier and customizing treatment plans for better patient outcomes.The main objective is to predict whether a person has cirrhosis and kidney disease in the early stage . Further research is warranted to validate these findings on larger and more diverse datasets and to integrate the models into clinical practice. The accuracy of Gradient Boosting Algorithm and Random Forest Algorithm are 99 and 99.1 respectively.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The liver is located in the upper part of the gastrointestinal tract of the human body, and its weight in men ranges between 1400–1800 g and in women between 1200–1400 g. It performs important functions related to digestion, metabolism, releasing toxins, immunization and nutrient storage. That is why some liver diseases can even lead to death.Liver diseases are categorized based on their aetiology and effect on the liver. The aetiology may include infection, injury, exposure to drugs or toxic substances, a process, or a genetic abnormality (such as hemochromatosis). The above causes can lead to hepatitis, cirrhosis, and stones that can increase in size and cause blockages, fatty infiltration and, in rare cases, liver cancer. Genetic abnormalities can also interfere with vital functions of the liver and lead to the deposition and concentration of harmful components, such as iron or copper.

Cirrhosis is also one of the most serious liver diseases. This disease causes healthy tissue to be replaced by scar tissue. Thus, the liver is permanently injured and cannot function properly. The main causes of cirrhosis of the liver include alcoholism, non-alcoholic fatty liver disease, chronic hepatitis C, and chronic hepatitis B

Nowadays, Kidney disease is a rapidly growing disease, and millions of people die due to lack of timely affordable treatment. Chronic kidney disease patients mostly belong to low-class and middle-class income-generating countries . In 2013, about one million people died due to chronic kidney disease A lot of work has been done for the early diagnosis of chronic kidney disease so that the disease could be treated at an early. Chronic kidney disease is a common type of kidney disease that occurs when both kidneys are damaged.

With the efficient use of these algorithms, the death rate can be minimized due to early-stage diagnosis and patients can be treated timely. Along with maintaining the clinical symptoms, chronic kidney disease patients should include physical activities in daily life. They should exercise, drink water, and avoid junk food.

1.2 PROBLEM DEFINITION

The proposed research work develops a system,where liver cirrhosis and kidney disease is predicted from the user's given input .We can predict the disease using predictive model by using machine learning technique .The scope of this project is to investigate a liver dataset and kidney dataset to find the cirrhosis and kidney disease using machine learning algorithms.By comparing various algorithm the one with more accuracy is used and by using metrics(accuracy, precision , recall, etc..) we can validate the model accuracy. The main aim is to implement an effective machine learning algorithm to predict and diagnose the occurrence of liver failure and kidney diseases in humans in order to eliminate the use of manual methods of analysis .

CHAPTER 2
LITERATURE SURVEY

CHAPTER 2

LITERATURE REVIEW

A literature review is a collection of written works with the purpose of reviewing the important aspects of the state of the art regarding and/or methodological approaches to a specific subject. It discusses information that has been published in a specific subject area and occasionally information that has been published within a specific time frame. These are secondary sources. Its main objective is to update the reader on the most recent research on a given subject. It also serves as a foundation for other objectives, including potential future study in the field, comes before a research proposal, or it could just be a straightforward summary of the available sources. It often follows an organized format and incorporates both synthesis and summary.

A review is a rearranging or rearranging of the information, whereas a summary is a recap of the most significant details from the source. It may offer a fresh perspective on antiquated information, synthesize contemporary and historical perspectives, or chart the intellectual development of the discipline, encompassing significant discussions. The literature review may assess the sources and recommend to the reader which ones are most applicable or useful based on the circumstances. Trends in loan default have long been investigated from a socioeconomic perspective. The majority of economics surveys rely on empirical modeling of these intricate networks to forecast a given person's likelihood of defaulting on a loan. It is currently noticing a trend in the application of machine learning for these kinds of activities.

Review of Literature Survey

Title : The Diagnosis of Chronic Liver Disease using Machine Learning Techniques

Author: Golmei Shaheamlung, Harshpreet Kaur

Year : 26-March-2021

In the 21st-century, the issue of liver disease has been increasing all over the world. As per the latest survey report, liver disease death toll has been rise approximately 2 million per year worldwide. The overall percentage of death by liver disease is 3.5% worldwide. Chronic Liver disease is also considered to be one of the deadly diseases, so early detection and treatment can recover the disease easily. The hidden knowledge of liver disease is recognized and extracted using a historical liver disease database. The complex queries are responded to diagnose liver disease The proposed model improved by applying a combination of three classifiers, Logistic regression, Random forest, and KNN algorithm. The python is employed for the implementation of the suggested model and the result proved regarding accuracy that is achieved 77.58 percent.

Title : Prediction and Analysis of Liver Disorder Diseases by using Data Mining Technique: Survey

Author: Shambel Kefelegn, Pooja Kamat

Year 2018

Liver disorder diseases one of the major diseases in the world, Liver is one of the huge solid organ in the human body; and is also considered a gland because, among its many functions, it makes and secretes bile. The liver theatres vital role in many physical functions from protein manufacture and blood clotting to fat, sugar and iron metabolism. Liver disorder diseases are any trouble of liver purpose that reason for sickness.

The study of paper to predicting and analysing liver disorder diseases to produce better performance accuracy by comparing various data mining classification algorithm and the performance of the accuracy is measured by confusion matrices. Decision Tree considered for performance evaluation in liver disorder diseases prediction Future work we can use the Hybrid approach to get better performance accuracy for liver disorder diseases prediction with their suitable data sets.

Title: Liver disease prediction by using different decision tree techniques

Author: Nazmun Nahar and Ferdous Ara

Year 2018

Early prediction of liver disease is very important to save human life and take proper steps to control the disease. Decision Tree algorithms have been successfully applied in various fields especially in medical science. This research work explores the early prediction of liver disease using various decision tree techniques. The liver disease dataset which is select for this study is consisting of attributes like total bilirubin, direct bilirubin, age, gender, total proteins, albumin and globulin ratio. The main purpose of this work is to calculate the performance of various decision tree techniques and compare their performance. The study employed some decision tree algorithm such as J48, LMT, Random Forest, Random tree, REPTree, Decision Stump and Hoeffding Tree to predict the liver disease at an earlier stage. These algorithm gives various result based on Accuracy, Mean Absolute Error, Precision, Recall, Kappa statistics and Runtime. These techniques were evaluated and their performance was compared. From the analysis. The results of this study will encourage us to continue developing other advanced decision trees such as CART

Title: A Comparative Study On Liver Disease Prediction Using Supervised Machine Learning Algorithms

Author: A.K.M Sazzadur Rahman, F. M. Javed Mehedi Shamrat, Zarrin Tasnim, Joy Roy, Syed Akhter Hossain

Year : 11, NOVEMBER 2019

Chronic Liver Disease is the leading cause of global death that impacts the massive quantity of humans around the world. This disease is caused by an assortment of elements that harm the liver. For example, obesity, an undiagnosed hepatitis infection, alcohol misuse. Which is responsible for abnormal nerve function, coughing up or vomiting blood, kidney failure, liver failure, jaundice, liver encephalopathy and there are many more. This disease diagnosis is very costly and complicated. Therefore, the goal of this work is to evaluate the performance of different Machine Learning algorithms in order to reduce the high cost of chronic liver disease diagnosis by prediction. In this work, we used six algorithms Logistic Regression. We just explored some popular supervised machine learning algorithms, more algorithms can be picked to assemble an increasingly precise model of liver disease prediction and performance can be progressively improved. Additionally, this work likewise ready to assume a significant role in health care research and just as restorative focuses to anticipate liver infection.

Title: Performance Evolution of Different Machine Learning Algorithms for Prediction of Liver Diseas

Author: Muktevi Srivenkatesh

Year : December 2019

Liver malady is an overall medical issue that is related with different inconveniences and high mortality. It is of basic significance that illness be recognized before such huge numbers of these lives can be spared. The phases of liver ailment are a significant viewpoint for focused treatment. It is a terribly troublesome undertaking for therapeutic analysts to foresee the disease inside the beginning times on account of sensitive manifestations. Generally the side effects become evident once it's past the point of no return. To beat this issue, we have liver infection forecast. Liver sickness might be distinguished with incalculable order systems, and these have been classified the utilization forecast of a number highlights and classifier blends. As end, the use of information digging systems for prescient examination is significant in the wellbeing field since it enables us to confront ailments prior and accordingly spare individuals' lives through the expectation of fix. In this work, we utilized a few learning calculation K-Nearest Neighbour, Support Vector Machines, Logistic Regression, Navi Bayes, Random Forest to foresee patients with constant liver disappointment infection, and patients who are not experiencing this illness. Re-enactment results demonstrated that Logisticregression classifier demonstrated its exhibition in foreseeing with best outcomes regarding precision and least execution time.

CHAPTER 3
THEORETICAL BACKGROUND

CHAPTER 3
THEORETICAL BACKGROUND

3.1 IMPLEMENTATION ENVIRONMENT

3.1.1. Software Requirements

Operating System	: Windows
Tool	: Anaconda with Jupyter Notebook

3.1.2. Hardware Requirements

Processor	: Pentium IV/III
Hard disk	: minimum 80 GB
RAM	: minimum 2 GB

Anaconda

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Anaconda distribution comes with more than 1,400 packages as well as the Conda package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packages can be individuallyinstalled from the Anaconda repository with the conda install command or using the pip install command that is is installed with Anaconda.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- Spyder
- PyCharm
- VSCode
- Glueviz
- Orange 3 App
- RStudio
- Anaconda Prompt (Windows only)
- Anaconda PowerShell (Windows only)

Jupyter Notebook and VSCode

Anaconda navigator website acts as “meta” documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities. The *Jupyter Notebook App* is a server-client application that allows editing and running notebook documents via a web browser.

In addition to displaying/editing/running notebook documents, the *Jupyter Notebook App* has a “Dashboard” (Notebook Dashboard), a “control panel” showing local files and allowing to open notebook documents or shutting down their kernels.

Installation: The easiest way to install the Jupyter Notebook App is installing a scientific python distribution which also includes scientific python packages. The most common distribution is called Anaconda

Running the Jupyter Notebook

Launching Jupyter Notebook App: The Jupyter Notebook App can be launched by clicking on the Jupyter Notebook icon installed by Anaconda in the start menu (Windows) or by typing in a terminal (cmd on Windows): “jupyter notebook” This will launch a new browser window (or a new tab) showing the Notebook Dashboard, a sort of control panel that allows (among other things) to select which notebook to open.

When started, the Jupyter Notebook App can access only files within its start-up folder (including any sub-folder. Modifications to the notebooks are automatically saved every few minutes. To avoid modifying the original notebook, make a copy of the notebook document (menu file -> make a copy…) and save the modifications on the copy.

Anaconda Distribution works with Visual Studio Code (VS Code), Microsoft’s lightweight and fast open-source code editor.

VS Code is free for both private and commercial use, runs on Windows, macOS, and Linux, and includes support for linting, debugging, task running, version control and Git integration, IntelliSense code completion, and conda environments.VS Code is openly extensible and many extensions are available.

In Anaconda Navigator version 1.7 or higher, use the launch VS Code. When you launch VS Code from Navigator, VS Code is configured to use the Python interpreter in the currently selected environment.

3.2 EXISTING SYSTEM

A dual-keyless-attention (DuKA) model that enables interpretable predictions of organ failure using electronic health record (EHR) data.Three modalities of medical data from EHR, namely diagnosis, procedure, and medications, are selected to predict three types of vital organ failures: heart failure, respiratory failure, and kidney failure. DuKA utilizes pre- trained embeddings of medical codes and combines them using a modality- wise attention module and a medical conceptwise attention module to enhance interpretation. Three organ failure tasks are addressed using two datasets to verify the effectiveness of DuKA. Moreover, one significant advantage offered by DuKA is that it allows clinicians to trace the contribution of variables from different sources of input (diagnosis/procedure/medication), which is meaningful in clinical practice. The modality-level attention scores offer valuable guidance to clinicians, encouraging them to prioritize diagnosis information when dealing with organ failure patients. The construction of DuKA takes two important factors into account. Firstly, DuKA is designed to fuse pre-trained medical code/concept embeddings originating from different modalities, which are trained separately. Secondly, DuKA aims to maintain a simple model structure while maximizing interpretability. Hence, instead of employing the multihead module, embed the keyless attention mechanism into DuKA. Overall, the proposed DuKA model addresses the challenges specific to modeling tabular EHR data.

DEMERITS

- Performance of BERT algorithm is poor .
- The accuracy range is from 80 to 90 percent
- They did not implement the deployment process.
- More complex process to train the data.

3.3 SYSTEM ARCHITECTURE

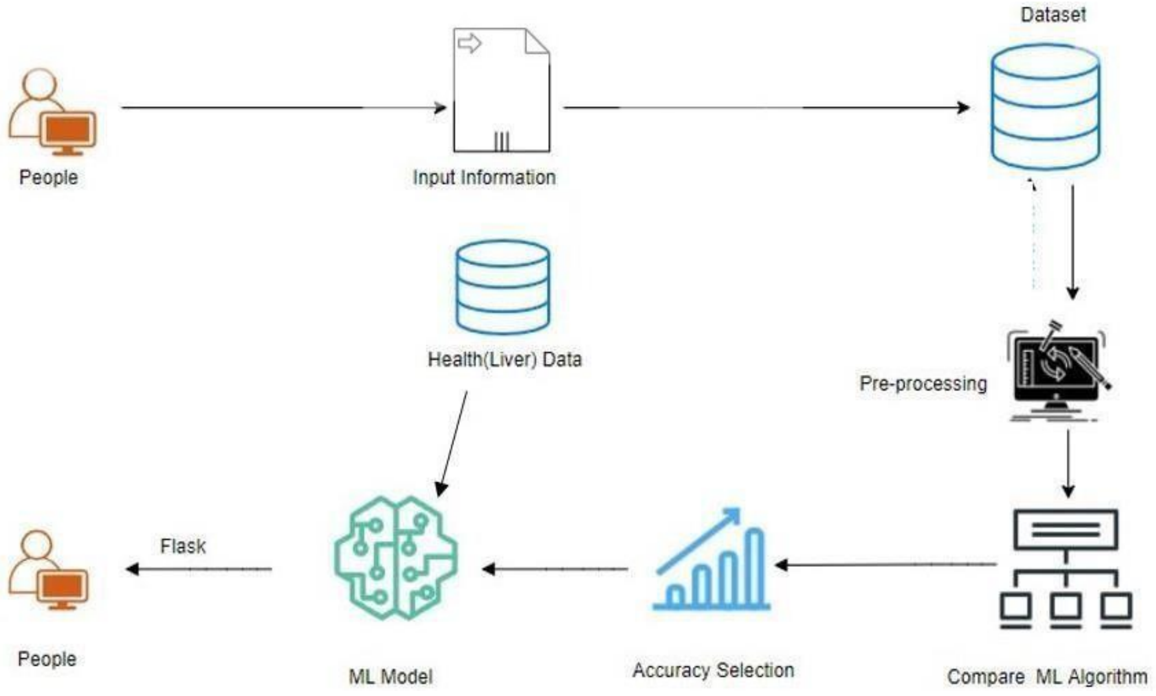


Figure 3.3.1 Architecture diagram for Prediction of liver cirrhosis and kidney disease

This displays the whole design of our working model including the components and the states occuring during the execution of the process.It displays the very initial process of people giving data input followed by the preprocessing and accuracy selection and prediction.If the user selects button submit then the user’s data will be submitted and those data is used for prediction of liver cirrhosis and kidney disease .

3.4 PROPOSED METHODOLOGY

We proposed a system to develop the project using machine learning algorithm. Recently, Machine learning and Artificial intelligence has plays a big role in various industries for their improvement and development. So we tried to implement machine learning algorithm to diagnosis the Cirrhosis and kidney disease disease. We collected the previous record of patient who had the Cirrhosis and kidney disease disease and who does had the disease and those who had symptoms. By collection of those peoples information our machine is tried to identifies the pattern of the datasets by various performing calculations. After identifies the pattern using various machine learning algorithm the model can able to predict the instance based on previous information.The user can login using their credentials and they can add their datas in our database and they can check whether they have chance of having liver cirrhosis and kidney disease by providing essential details .So that the user can predict liver cirrhosis and kidney disease in the early stage itself.

ADVANTAGES

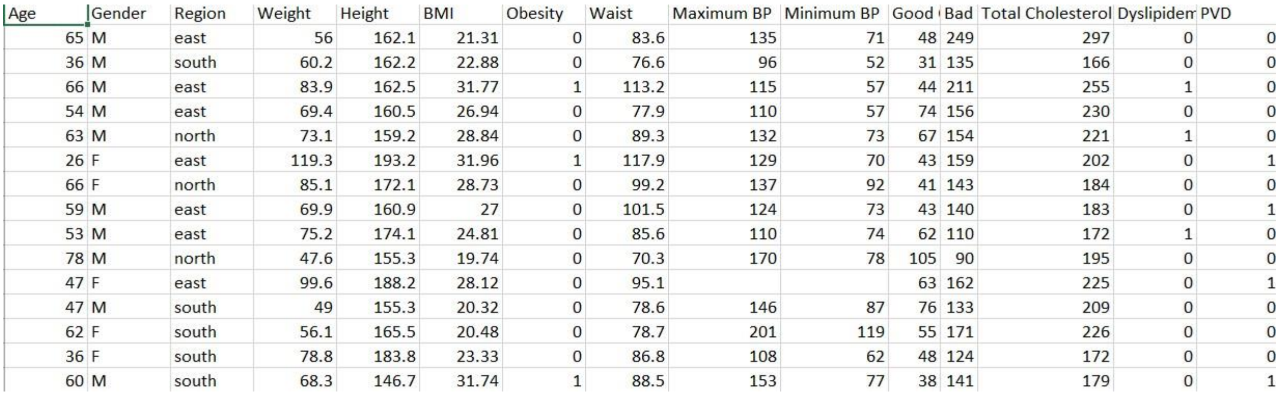
- We build a production level application for deployment purpose.
- We build an advance machine learning techniques to build a predictive model.
- We compared more than a two architecture to getting better accuracy level.
- We train the structured data for machine learning model.

3.4.1 DATASET DESCRIPTION

This project uses 2 different dataset to predict liver cirrhosis and kidney disease respectively.

1) Cirrhosis Dataset

This dataset consists of 8786 records of patients combining those who had cirrhosis and those who are not having cirrhosis and the dataset attributes include age, gender, region, Weight, height, BMI, Obesity, Waist, Maximum and Minimum Blood pressure, Good Cholestrol, bad Cholestrol, Total Cholestrol, Dyslipidemia,PVD(Peripheral Vascular Disease), Physical Activity, Education,Unmarried, Income, Source of care, Poor Vision, Alcohol Consumption, Hypertension, Family Hypertension, Diabetes ,Family Diabetes, Hepatitis ,Family Hepatitis, Chronic Fatigue,ALF(Acute Liver Failure).



Age	Gender	Region	Weight	Height	BMI	Obesity	Waist	Maximum BP	Minimum BP	Good	Bad	Total Cholesterol	Dyslipidem	PVD
65	M	east	56	162.1	21.31	0	83.6	135	71	48	249	297	0	0
36	M	south	60.2	162.2	22.88	0	76.6	96	52	31	135	166	0	0
66	M	east	83.9	162.5	31.77	1	113.2	115	57	44	211	255	1	0
54	M	east	69.4	160.5	26.94	0	77.9	110	57	74	156	230	0	0
63	M	north	73.1	159.2	28.84	0	89.3	132	73	67	154	221	1	0
26	F	east	119.3	193.2	31.96	1	117.9	129	70	43	159	202	0	1
66	F	north	85.1	172.1	28.73	0	99.2	137	92	41	143	184	0	0
59	M	east	69.9	160.9	27	0	101.5	124	73	43	140	183	0	1
53	M	east	75.2	174.1	24.81	0	85.6	110	74	62	110	172	1	0
78	M	north	47.6	155.3	19.74	0	70.3	170	78	105	90	195	0	0
47	F	east	99.6	188.2	28.12	0	95.1			63	162	225	0	1
47	M	south	49	155.3	20.32	0	78.6	146	87	76	133	209	0	0
62	F	south	56.1	165.5	20.48	0	78.7	201	119	55	171	226	0	0
36	F	south	78.8	183.8	23.33	0	86.8	108	62	48	124	172	0	0
60	M	south	68.3	146.7	31.74	1	88.5	153	77	38	141	179	0	1

Figure 3.4.1 Sample of the Dataset

2) Kidney Disease Dataset

This dataset consists of 401 records of patients including those who had kidney disease and those who are not having and the dataset attributes includes BP(Blood Pressure), SG(Specific gravity), Al(Albumin level), Su(Salicycluric acid), Rbc(Hematuriadiasease), Bu(Blood urea nitrogen), Sc, Sod(Sodium), Pot(potassium), Hemo(haemoglobin), Wbcc(White blood cell count), Rbcc(Red blood cell count), Htn(Hypertensive Nephropathy), class(kidney disease).

3.4.2 MODULE DESIGN

3.4.2.1 Flow diagram

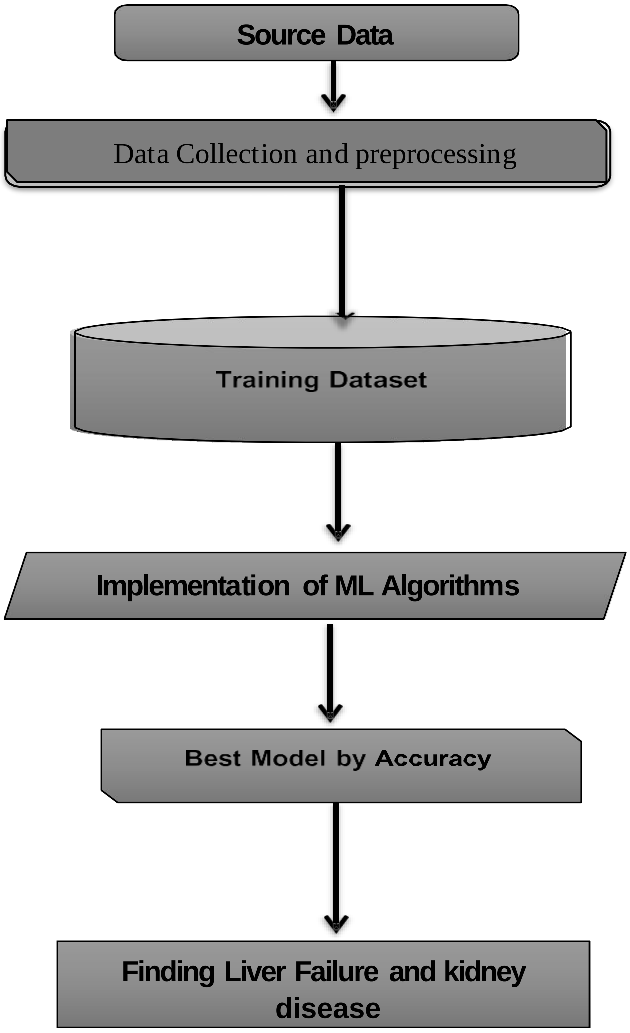

Source Data
Data Collection and preprocessing
Training Dataset
Implementation of ML Algorithms
Best Model by Accuracy
Finding Liver Failure and kidney disease

Figure 3.4.2 Working flow of the Prediction of cirrhosis and kidney disease

A flow diagram, also known as a flowchart, is a graphical representation of a process or system, illustrating the steps involved and the sequence in which they occur. Creating a flow diagram for predicting liver cirrhosis and kidney disease involves outlining the steps and decision points involved in the prediction process.So it starts from gathering data from patients and preprocessing of those data . By implementing machine learning algorithms which has high accuracy helps to predict liver failure and kidney disease.

3.4.2.2 UML diagrams

Use case diagram

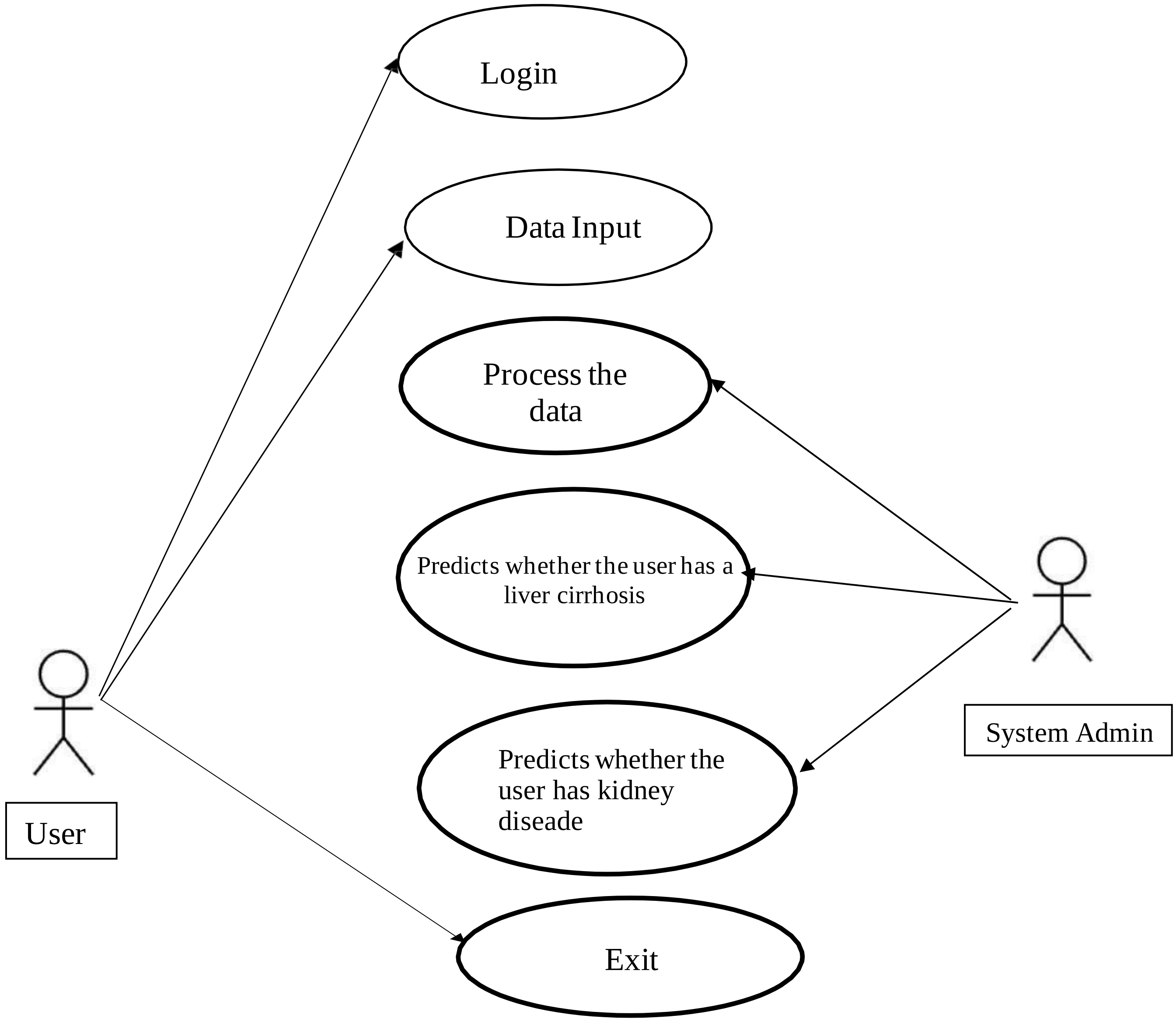

Login
Data Input
Process the data
Predicts whether the user has a liver cirrhosis
Predicts whether the user has kidney diseade
Exit
User
System Admin

Figure 3.4.3 Use Case diagram for prediction of liver cirrhosis and kidney disease

The use case diagram refers to activities done by the system and the users and the coressponding use cases .The user have to login using their credentials and they have to provide necessary input details based on the details the system wil predict whether the user has a liver cirrhosis and kidney disease or not. .

Class Diagram

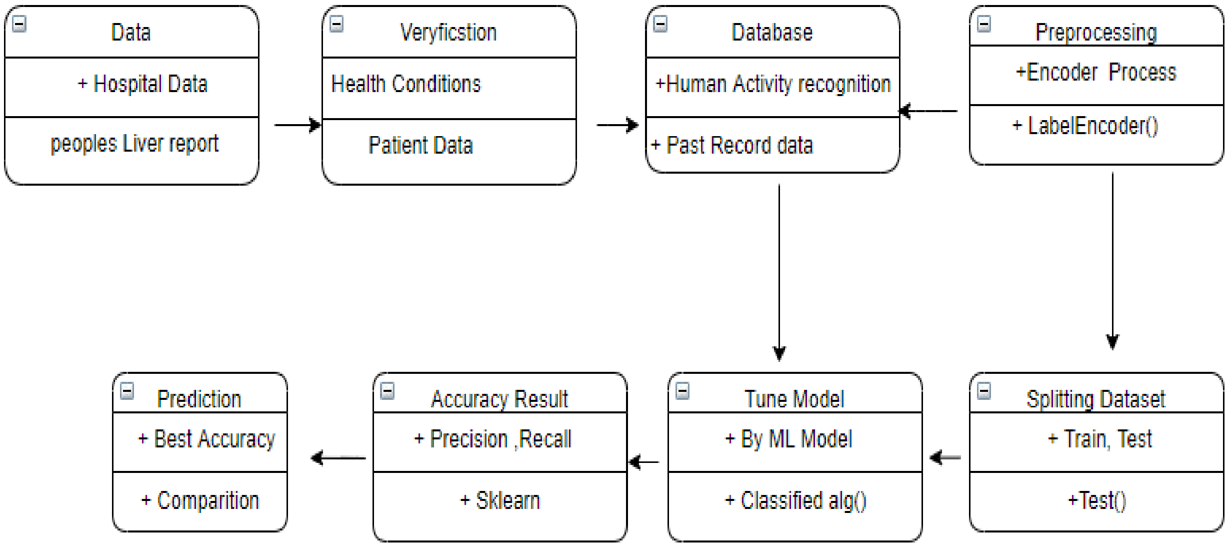


Figure 3.4.4 Class diagram for prediction of liver cirrhosis and kidney disease

Class diagram illustrates structure and relationships between different classes .It represents the static view of a system, showcasing classes, attributes, methods, and their associations. In the context of predicting liver cirrhosis and kidney disease,Data class has attributes of hospital data such as age,gender,BMI,..,Preprocessing consist of method LabelEncoder and the Splitting Dataset consists of train and Test methods ,Tune Model has Classified algorithm method ,accuracy result consist of Precision,recall and Sklearn method and Prediction class consists of comparison of accuracy method.

Activity Diagram

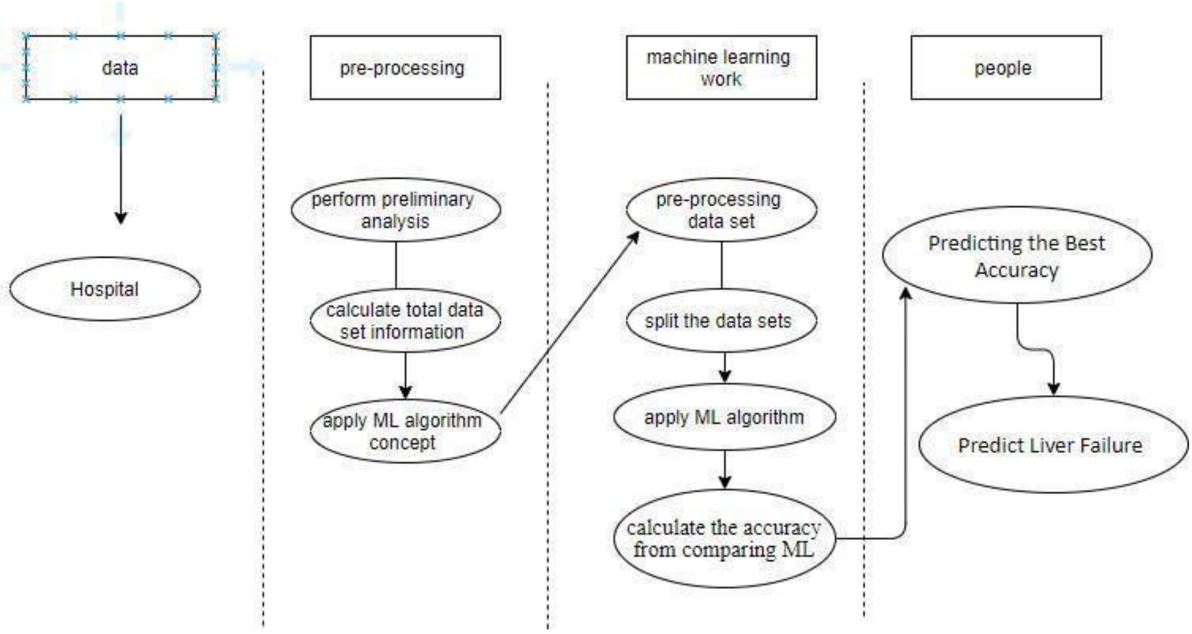

data
pre-processing
machine learning work
people
Hospital
perform preliminary analysis
calculate total data set information
apply ML algorithm concept
pre-processing data set
split the data sets
apply ML algorithm
calculate the accuracy from comparing ML
Predicting the Best Accuracy
Predict Liver Failure

Figure 3.4.5 Activity diagram for Prediction of Liver Cirrhosis and Kidney disease

An activity diagram is a UML (Unified Modeling Language) diagram that represents the flow of activities in a system or a business process. It provides a visual representation of the sequential and parallel activities within a system. In the context of predicting liver cirrhosis and kidney disease,the above diagram shows the flow of activities from giving input to the prediction of liver and kiney disease.

CHAPTER 4
SYSTEM IMPLEMENTATION

CHAPTER 4

SYSTEM IMPLEMENTATION

Modules

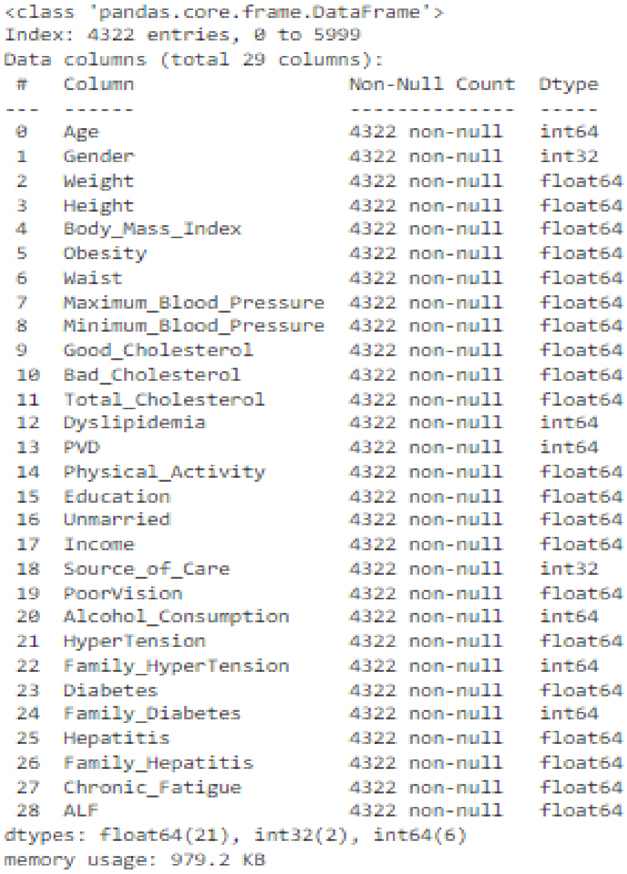
- Data Pre-processing
- Data Analysis of Visualization
- Implementing GRB Algorithm
- Implementing Random Forest Classifier
- Deployment Using Django

4.1 Data Pre-processing

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset.To finding the missing value, duplicate value and description of data type whether it is float variable or integer. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand data and its properties; this knowledge will help to choose which algorithm to use to build your model.A number of different data cleaning tasks using Python’s Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It’s important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it’s important to understand the sources of missing data. Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.



```
<class 'pandas.core.frame.DataFrame'>
Index: 4322 entries, 0 to 5999
Data columns (total 29 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      4322 non-null   int64  
 1   Gender                   4322 non-null   int32  
 2   Weight                   4322 non-null   float64
 3   Height                   4322 non-null   float64
 4   Body_Mass_Index          4322 non-null   float64
 5   Obesity                  4322 non-null   float64
 6   Waist                    4322 non-null   float64
 7   Maximum_Blood_Pressure   4322 non-null   float64
 8   Minimum_Blood_Pressure   4322 non-null   float64
 9   Good_Cholesterol         4322 non-null   float64
 10  Bad_Cholesterol          4322 non-null   float64
 11  Total_Cholesterol        4322 non-null   float64
 12  Dyslipidemia             4322 non-null   int64  
 13  PVD                      4322 non-null   int64  
 14  Physical_Activity        4322 non-null   float64
 15  Education                4322 non-null   float64
 16  Unmarried                4322 non-null   float64
 17  Income                   4322 non-null   float64
 18  Source_of_Care           4322 non-null   int32  
 19  PoorVision               4322 non-null   float64
 20  Alcohol_Consumption      4322 non-null   int64  
 21  HyperTension             4322 non-null   float64
 22  Family_HyperTension      4322 non-null   int64  
 23  Diabetes                 4322 non-null   float64
 24  Family_Diabetes          4322 non-null   int64  
 25  Hepatitis                4322 non-null   float64
 26  Family_Hepatitis         4322 non-null   float64
 27  Chronic_Fatigue          4322 non-null   float64
 28  ALF                      4322 non-null   float64
dtypes: float64(21), int32(2), int64(6)
memory usage: 979.2 KB
```

Figure 4.1.1 Information of the dataset after preprocessing

4.2 Data analysis of Visualization

Data visualization is an important skill in applied statistics and machine learning. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

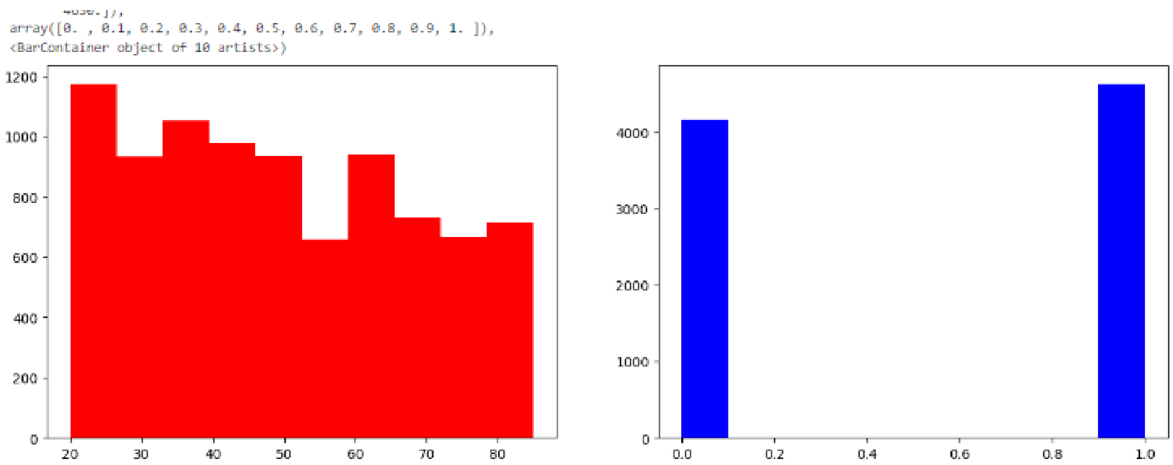
Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

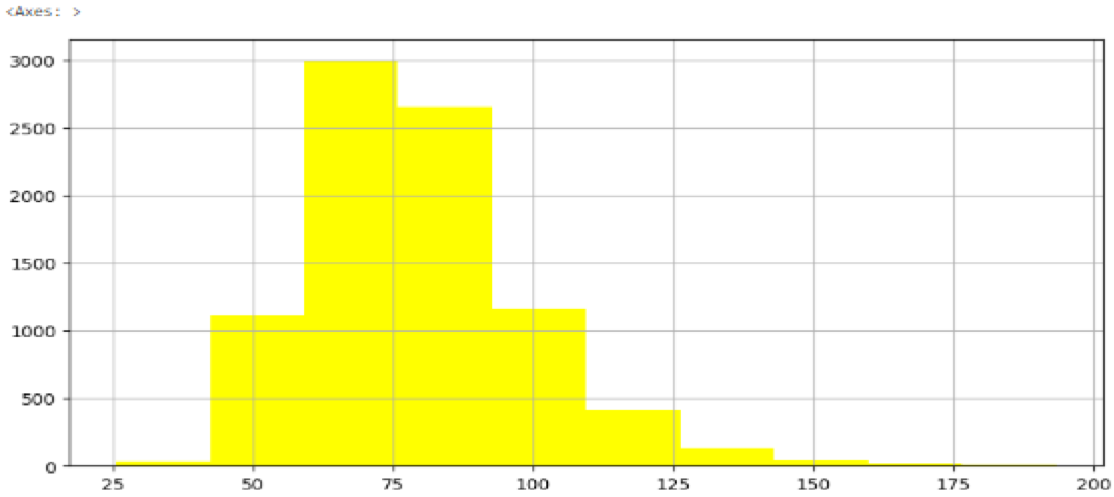
True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.



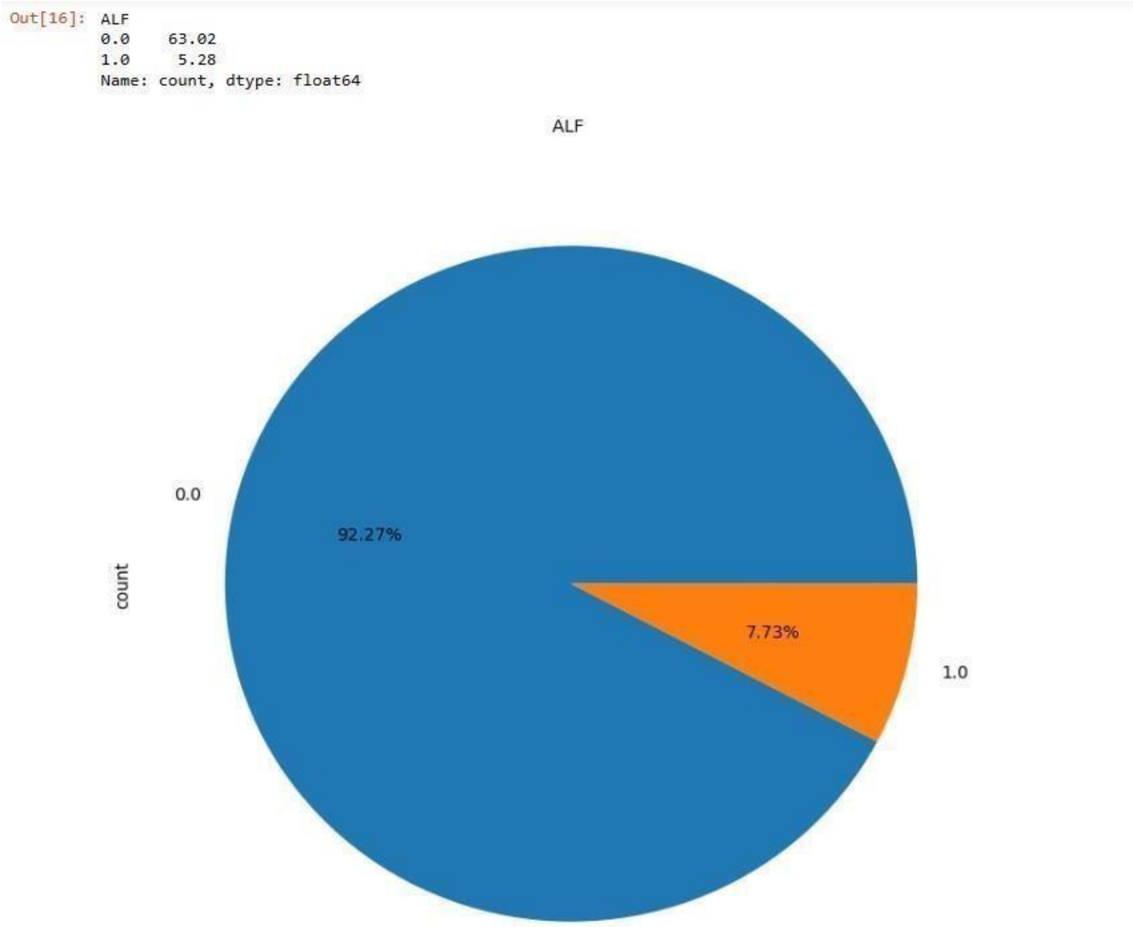
```
       4636.]),
array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
<BarContainer object of 10 artists>)
```

Figure 4.2.1 Histogram plot for age and gender



```
<Axes: >
```

Figure 4.2.2 Histogram plot for Weight



```
Out[16]: ALF
         0.0    63.02
         1.0     5.28
         Name: count, dtype: float64
```

Figure 4.2.3 Pie-chart for target column by its counts

Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data.When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

- Random Forest
- GRBoosting

The K-fold cross validation procedure is used to evaluate each algorithm, importantly configured with the same random seed to ensure that the same splits to the training data are performed and that each algorithm is evaluated in precisely the same way. Before that comparing algorithm, Building a Machine Learning Model using install Scikit-Learn libraries. In this library package have to done preprocessing, linear model with logistic regression method, cross validating by KFold method, ensemble with random forest method and tree with decision tree classifier. Additionally, splitting the train set and test set. To predicting the result by comparing accuracy.

4.3 Algorithm and techniques

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

Used Python Packages

sklearn

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like train_test_split, DecisionTreeClassifier or Logistic Regression and accuracy_score.

NumPy

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

GRBoosting Classifier Algorithm

Machine learning is one of the most popular technologies to build predictive models for various complex regression and classification tasks. **Gradient Boosting Machine** (GBM) is considered one of the most powerful boosting algorithms.

Although, there are so many algorithms used in machine learning, boosting algorithms has become mainstream in the machine learning community across the world. Boosting technique follows the concept of ensemble learning, and hence it combines multiple simple models (weak learners or base estimators) to generate the final output. GBM is also used as an ensemble method in machine learning which converts the weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error or cross-entropy of the previous model using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current ensemble and then trains a new weak model to minimize this gradient. The predictions of the new model are then added to the ensemble, and the process is repeated until a stopping criterion is met.

1. Ensemble Learning

- Gradient Boosting is an ensemble learning technique where multiple weak learners (typically decision trees) are combined to create a strong learner.

2. Weak Learners (Decision Trees)

- The base learners, often decision trees, are called weak learners because they perform slightly better than random chance.

3. Boosting Concept

- Gradient Boosting builds trees sequentially, each one correcting errors made by the previous trees.

- At each stage, a new tree is trained on the residuals (the differences between the actual values and the predictions made by the existing ensemble).

4. Objective Function

- The algorithm minimizes a predefined loss function, which measures the difference between the predicted values and the actual values.
- Common loss functions include mean squared error for regression problems and deviance (logistic loss) for classification problems.

5. Gradient Descent

- The "Gradient" in Gradient Boosting refers to the gradient descent optimization used to minimize the loss function.
- At each iteration, the algorithm calculates the negative gradient of the loss function with respect to the current ensemble's predictions.

6. Learning Rate

- A learning rate parameter controls the step size during the gradient descent optimization.
- A lower learning rate makes the algorithm more robust, but it requires more iterations. A higher learning rate speeds up convergence but may lead to overshooting.

7. Trees and Weak Learners

- Trees are typically shallow, and each new tree addresses the errors of the combined ensemble.
- Shallow trees reduce overfitting and contribute to the model's interpretability.

8. Regularization

- Gradient Boosting includes regularization techniques, such as tree pruning and feature subsampling, to prevent overfitting.

9. Prediction

- The final prediction is the sum of the predictions from all the weak learners, each multiplied by its associated learning rate.

10. Libraries Implementation

- Popular libraries that implement Gradient Boosting include Scikit-learn (with GradientBoostingClassifier for classification tasks) and XGBoost, LightGBM, and CatBoost, which are specialized libraries optimized for performance.

Advantages

- Good predictive performance.
- Handles both numerical and categorical data.
- Can capture complex relationships in the data.

Challenges

- May require tuning of hyperparameters.
- Prone to overfitting if hyperparameters are not properly set.

Gradient Boosting has proven to be highly effective in various machine learning tasks and is widely used in practice. Its ability to build strong models by sequentially improving weak learners makes it a valuable tool in the data scientist's toolbox.

GradientBoosting is used to predict kidney disease and the confusion matrix score of GRADIENT BOOSTING CLASSIFIER: [[50 0] [1 49]].

Random forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.
- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.

- It can also maintain accuracy when a large proportion of data is missing.

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps :

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision.

Implementation in Scikit-learn

For each decision tree, Scikit-learn calculates a nodes importance using Gini Importance, assuming only two child nodes (binary tree):

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \text{ --------------(1)}$$

- ni_j= the importance of node j
- w_j = weighted number of samples reaching node j
- C_j= the impurity value of node j
- left(j) = child node from left split on node j

- right(j) = child node from right split on node j

The importance for each feature on a decision tree is then calculated as:

$$f\ \mathbf{i_i = \sum_{j\ :\ node\ j\ splits\ on\ feature\ i} ni_j \ / \sum_{k\ €\ all\ nodes} ni_k}\text{-----------(2)}$$

- fi_i= the importance of feature i
- ni_j= the importance of node j

These can then be normalized to a value between 0 and 1 by dividing by the sum of all feature importance values.

The final feature importance, at the Random Forest level, is it's average over all the trees. The sum of the feature's importance value on each trees is calculated and divided by the total number of trees:

$$\mathbf{RF\ f\ i_i = \sum_{j\ €\ all\ trees} norm\ f\ i_{ij} \ / \ T}\text{-------(3)}$$

- $Rffi_i$= the importance of feature i calculated from all trees in the Random Forest model
- $Normfi_{ij}$= the normalized feature importance for i in tree j
- T = total number of trees

It is used for predicting liver cirrhosis and the confusion matrix score is:
[[794 13] [0 807]].

4.4 Deployment

In this module the trained deep learning model is converted into hierarchical dataformat file (.h5 file) which is then deployed in our django framework for providing better user interface and predicting the output whether the given image is CKD / Not CKD.The name of the package is used to resolve resources from inside the package orthe folder the module is contained in depending on if the packageparameter resolves to an actual python package or a standard module (just a .py file).

DJANGO:

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.Django was designed to help developers take applications from concept to completion as quickly as possible.Django takes security seriously and helps developers avoid many common security mistakes.Some of the busiest sites on the web leverage Django's ability to quickly and flexibly scale.With Django, you can take web applications from concept to launch in a matter of hours. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Fully loaded and Reassuringly secure

Django includes dozens of extras you can use to handle common web development tasks. Django takes care of user authentication, content administration, site maps, RSS feeds, and many more tasks — right out of the box.Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords.

CHAPTER 5
RESULTS & DISCUSSIONS

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Performance Analysis

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

True Positive Rate (TPR) = TP / (TP + FN)

False Positive rate (FPR) = FP / (FP + TN)

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. The accuracy calculated by using the above formula for both randomforest and gradient boost classifier algorithm is shown below

```
THE CROSS VALIDATION TEST RESULT OF ACCURACY :
[99.31846344 99.19454771 99.19454771 99.19454771 99.38042131]
THE ACCURACY SCORE OF RANDOM FOREST CLASSIFIER IS : 99.19454770755885

THE CROSS VALIDATION TEST RESULT OF ACCURACY :
[ 99. 100. 100.  99. 100.]
THE ACCURACY SCORE OF GRADIENT BOOSTING CLASSIFIER IS : 99.0
```

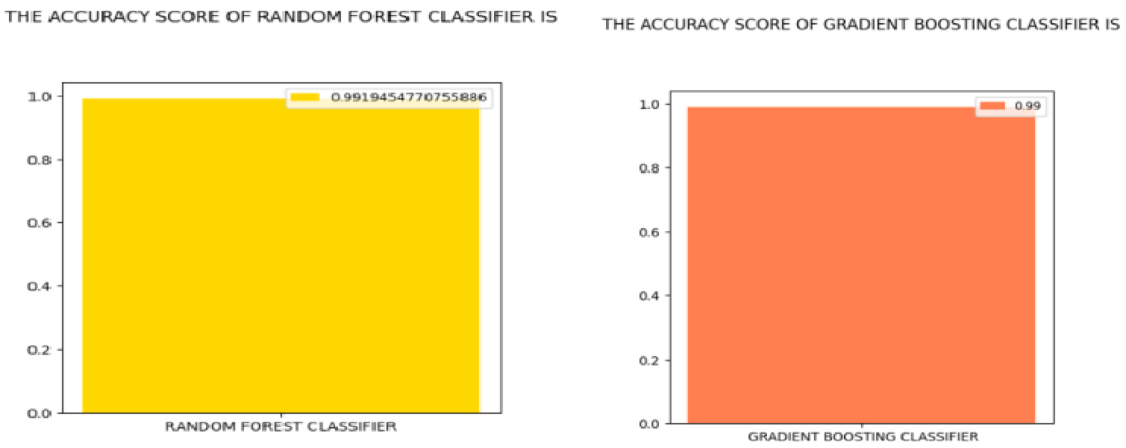


Figure 5.1.1 Accuracy score of Random forest and Gradient Boosting classifier algorithm

Precision: The proportion of positive predictions that are actually correct. (When the model predicts default: how often is correct?)

Precision = TP / (TP + FP)

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = TP / (TP + FN)

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

F- Measure = 2TP / (2TP + FP + FN)

F1-Score Formula

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

CHAPTER 6
CONCLUSION & FUTURE WORK

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In conclusion, the application of advanced machine learning techniques, specifically Random Forest and Gradient Boosting, has proven to be instrumental in predicting the onset of liver cirrhosis and kidney disease. Through the analysis of relevant medical data, these models demonstrated a high level of accuracy and reliability in identifying key indicators and patterns associated with the progression of these conditions.The Random Forest algorithm, with its ensemble approach and ability to handle complex datasets, showcased robust predictive performance. Similarly, Gradient Boosting, through its iterative optimization process, exhibited superior predictive power by capturing subtle relationships within the data and improving overall model precision.The successful implementation of these machine learning techniques not only enhances our understanding of the factors contributing to liver cirrhosis and kidney disease but also holds significant promise for early detection and intervention. This, in turn, could lead to improved patient outcomes and more efficient healthcare resource allocation.

The subtle nature of liver disease's symptoms makes diagnosis especially difficult. Nearly 38,170 fatalities from chronic liver disease were reported in the United States in 2014 out of a total of 2,626,418 deaths. The significance of computer prediction will only increase. In order to increase prediction capacity, this project uses two potential machine learning algorithms. The molecular biology method is frequently impacted by age, ethnicity, and food. The chemical method is a more reliable way to make predictions. Molecular biology research, however, has the potential to save lives by helping to unlock the mysteries of human anatomy. The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score is will be find out. This project can help to find the Liver Failure stage based on the patient health.

6.2 FUTURE WORK

Continued research and refinement of these models could pave the way for personalized and proactive healthcare strategies, ultimately contributing to better patient care and outcomes in the realm of hepatic and renal health.Future works include automating this process by showing the prediction result in web application or desktop application. We can deploy this model in any cloud based system. Incorporating multi-omics data (genomics, transcriptomics, proteomics, metabolomics) into prediction models can provide a more comprehensive understanding of the molecular mechanisms underlying liver cirrhosis and kidney disease. Developing wearable devices and remote monitoring technologies for continuous assessment of liver and kidney function could revolutionize disease prediction and management. These devices could track relevant biomarkers, physiological parameters, and lifestyle factors, providing timely feedback and early warning signs of disease progression. Integrating prediction models into clinical decision support systems (CDSS) can assist healthcare providers in risk stratification, treatment planning, and patient management.

APPENDICES
A1. SDG GOAL

The United Nations created 17 world development goals called the Sustainable Development Goals(SDG). They were created in 2015 with the aim of "peace and prosperity for people and the planet, now and into the future”. The application of machine learning techniques for the prediction of liver cirrhosis and kidney disease aligns with several United Nations Sustainable Development Goals (SDGs). Here are some relevant SDGs that can be associated with this healthcare initiative

SDG 3: Good Health and Well-being

It specifically aims to reduce premature mortality from non-communicable diseases. Early prediction and intervention for liver cirrhosis and kidney disease contribute to achieving this target by improving overall health outcomes and reducing the burden of chronic diseases.

SDG 9: Industry, Innovation, and Infrastructure

Leveraging advanced machine learning algorithms demonstrates a commitment to innovation in healthcare. The development and application of predictive models contribute to building resilient infrastructure and promoting sustainable industrialization.

SDG 10: Reduced Inequalities

Implementing predictive models for disease prediction can help address health inequalities. By ensuring that these models are validated on diverse populations and are accessible to all, regardless of socio-economic factors, the initiative supports the goal of reducing inequalities in health outcomes.

SDG 17: Partnerships for the Goals:

Collaborating with healthcare professionals, researchers, and technology experts in the development and validation of predictive models aligns with the spirit of SDG 17. Building effective partnerships is essential for the successful implementation of innovative solutions in the healthcare sector.

A2. SOURCE CODE

Module 1: Data Preprocessing and Data Cleaning

```
# Import Required libraries
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
# Load CSV file
df = pd.read_csv('CIRRHOSIS.csv')
del df['Region']
df.head()
# Total no of rows and columns in the datasets
df.shape
# Total no of elements present in the dataset
df.size
# List of columns in the dataset
df.columns
# Convert categorical values into integers
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

var = ['Gender','Source of Care']

for i in var:
    df[i] = le.fit_transform(df[i]).astype(int)
# display last five rows in the table
df.tail()
# Rename columns indes without space
df.rename(columns={"Body Mass Index": "Body_Mass_Index",
                   "Maximum Blood Pressure":"Maximum_Blood_Pressure",
                   "Minimum Blood Pressure":"Minimum_Blood_Pressure",
                   "Good Cholesterol":"Good_Cholesterol",
                   "Bad Cholesterol":"Bad_Cholesterol",
                   "Total Cholesterol":"Total_Cholesterol",
                   "Physical Activity":"Physical_Activity",
                 "Source of Care":"Source_of_Care",
                 "Alcohol Consumption":"Alcohol_Consumption",
                 "Family Diabetes":"Family_Diabetes",
                 "Family Hepatitis":"Family_Hepatitis",
                 "Chronic Fatigue":"Chronic_Fatigue",
                "Family  HyperTension":"Family_HyperTension"}, inplace=True)

                 # Checking for the possiblility of Missing values
df.isnull()
# Remove NaN values in the datasets
df = df.dropna()
# Unique values in the Target columns
df['ALF'].unique()
# five point summary of the datasets
df.describe()
# Findout the Relationship between independent and dependent variables
df.corr()
# Information about the dataset
df.info()
# categorical comparison between two columns
pd.crosstab(df["Hepatitis"], df["HyperTension"])
# categorical comparison between grouped columns
df.groupby(["Education","Dyslipidemia"]).groups
# target columns unique counts
df["ALF"].value_counts()
# categorical columns description
pd.Categorical(df["Height"]).describe()
# Check for Duplicated values
df.duplicated()
# total count of duplicated values
sum(df.duplicated())
```

Module 2: Data Visualisation and Data analysis

```
# Import Required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# Load data set file
df = pd.read_csv('CIRRHOSIS.csv')
del df['Region']
df.head()

# Columns List
df.columns
# Convert categorical values to integers
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

var = ['Gender','Source of Care']

for i in var:
   df[i] = le.fit_transform(df[i]).astype(int)
# Rename columns index
df.rename(columns={"Body Mass Index": "Body_Mass_Index",
               "Maximum Blood Pressure":"Maximum_Blood_Pressure",
               "Minimum Blood Pressure":"Minimum_Blood_Pressure",
               "Good Cholesterol":"Good_Cholesterol",
               "Bad Cholesterol":"Bad_Cholesterol",
               "Total Cholesterol":"Total_Cholesterol",
               "Physical Activity":"Physical_Activity",
               "Source of Care":"Source_of_Care",
               "Alcohol Consumption":"Alcohol_Consumption",
               "Family Diabetes":"Family_Diabetes",
               "Family Hepatitis":"Family_Hepatitis",
               "Chronic Fatigue":"Chronic_Fatigue",
               "Family HyperTension":"Family_HyperTension"}, inplace=True)
# Target columns unique value counts
plt.figure(figsize=(12,7))
sns.countplot(x='ALF',data=df)
# Histogram plot for two columns
plt.figure(figsize=(15,5))

plt.subplot(1,2,1)
plt.hist(df['Age'],color='red')
plt.subplot(1,2,2)
plt.hist(df['Gender'],color='blue')

# Histogram graph for whole datasets
df.hist(figsize=(15,55),layout=(15,4), color='green')
plt.show()


# Histogram plot for weight
df['Weight'].hist(figsize=(10,5),color='yellow')
# Line plot for Height columns
sns.lineplot(df['Height'], color='brown') # scatter, plot, triplot, stackplot
# Alcohol consumption Representation
sns.violinplot(df['Alcohol_Consumption'], color='purple')
# Density of faimly huper tension relations
df['Family_HyperTension'].plot(kind='density')
# Displot for hepatitis analysis
sns.displot(df['Hepatitis'], color='purple')
# Displot of chronic_Fatigue
sns.displot(df['Chronic_Fatigue'], color='coral') # residplot, scatterplot
# plot heatmap to check correlation between columns
fig, ax = plt.subplots(figsize=(20,15))
sns.heatmap(df.corr(),annot = True, fmt='0.2%',cmap = 'autumn',ax=ax)
# Plot the Target column by its counts
def plot(df, variable):
   dataframe_pie = df[variable].value_counts()
   ax = dataframe_pie.plot.pie(figsize=(9,9), autopct='%1.2f%%', fontsize = 10)
   ax.set_title(variable  + ' \n', fontsize = 10)
   return np.round(dataframe_pie/df.shape[0]*100,2)
plot(df, 'ALF')
```

Module 3: Random Forest Classifier Algorithm

```
# Import Required libaries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
# Load the datasets with required columns
df = pd.read_csv('CIRRHOSIS.csv')
del df['Region']
df.head()
# Counts of columns titles
df.columns

# Drop missing values
df=df.dropna()
# Convert Categories into integers
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

var = ['Gender','Source of Care']

for i in var:
    df[i] = le.fit_transform(df[i]).astype(int)
# Rename columns with out spaces
df.rename(columns={"Body Mass Index": "Body_Mass_Index",
                   "Maximum Blood Pressure":"Maximum_Blood_Pressure",
                   "Minimum Blood Pressure":"Minimum_Blood_Pressure",
                   "Good Cholesterol":"Good_Cholesterol",
                   "Bad Cholesterol":"Bad_Cholesterol",
                   "Total Cholesterol":"Total_Cholesterol",
                   "Physical Activity":"Physical_Activity",
                   "Source of Care":"Source_of_Care",
                   "Alcohol Consumption":"Alcohol_Consumption",
                   "Family Diabetes":"Family_Diabetes",
                   "Family Hepatitis":"Family_Hepatitis",
                   "Chronic Fatigue":"Chronic_Fatigue",
                   "Family HyperTension":"Family_HyperTension"}, inplace=True)
df.columns
# List the last five rows in the table
df.tail()
# Seperate dependent and independent columns in the table
```

```
x1 = df.drop(labels='ALF', axis=1)
y1 = df.loc[:,'ALF']
# check for imbalance and its treatment
import imblearn
from imblearn.over_sampling import RandomOverSampler
from collections import Counter

ros =RandomOverSampler(random_state=42)
x,y=ros.fit_resample(x1,y1)
print("OUR DATASET COUNT         : ", Counter(y1))
print("OVER SAMPLING DATA COUNT  : ", Counter(y))


# for model selection split the dataset based on requirement
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20,
random_state=42, stratify=y)
print("NUMBER OF TRAIN DATASET    : ", len(x_train))
print("NUMBER OF TEST DATASET      : ", len(x_test))
print("TOTAL NUMBER OF DATA    : ", len(x_train)+len(x_test))
# Display the datsets after seperation
print("NUMBER OF TRAIN DATASET    : ", len(y_train))
print("NUMBER OF TEST DATASET      : ", len(y_test))
print("TOTAL NUMBER OF DATASET    : ", len(y_train)+len(y_test))
# Import Random forest classifiers
from sklearn.ensemble import RandomForestClassifier
# Fit the datsets
RFC = RandomForestClassifier(random_state=42)
RFC.fit(x_train,y_train)
# Predict the test vales by fit
predicted = RFC.predict(x_test)
# predict Confusion matrix based on prediction
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,predicted)
print('THE CONFUSION MATRIX SCORE OF RANDOM FOREST
CLASSIFIER:\n\n\n',cm)
# CaLculate accuracy based on validation
from sklearn.model_selection import cross_val_score
```

```
accuracy = cross_val_score(RFC, x, y, scoring='accuracy')
print('THE CROSS VALIDATION TEST RESULT OF ACCURACY :\n\n\n',
accuracy*100)
# Accuracy of the model
from sklearn.metrics import accuracy_score
a = accuracy_score(y_test,predicted)

print("THE ACCURACY SCORE OF RANDOM FOREST CLASSIFIER IS
:",a*100)
# Loss occred while model performance
from sklearn.metrics import hamming_loss
hl = hamming_loss(y_test,predicted)
print("THE HAMMING LOSS OF RANDOM FOREST CLASSIFIER IS
:",hl*100)



# Precion based on model performance
from sklearn.metrics import precision_score
P = precision_score(y_test,predicted)
print("THE PRECISION SCORE OF RANDOM FOREST CLASSIFIER IS
:",P*100)
# Recall based on the performances of the model
from sklearn.metrics import recall_score
R = recall_score(y_test,predicted)
print("THE RECALL SCORE OF RANDOM FOREST CLASSIFIER IS
:",R*100)
# combine score score of precision and recall
from sklearn.metrics import f1_score
f1 = f1_score(y_test,predicted)
print("THE PRECISION SCORE OF RANDOM FOREST CLASSIFIER IS
:",f1*100)
# Confusion matrix Plot
def plot_confusion_matrix(cm, title='THE CONFUSION MATRIX SCORE OF
RANDOM FOREST CLASSIFIER\n\n', cmap=plt.cm.Blues):
    target_names=['']
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(target_names))
    plt.xticks(tick_marks, target_names, rotation=45)
    plt.yticks(tick_marks, target_names)
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

cm=confusion_matrix(y_test, predicted)
print('THE CONFUSION MATRIX SCORE OF RANDOM FOREST
CLASSIFIER:\n\n')
print(cm)

sns.heatmap(cm/np.sum(cm), annot=True, cmap = 'Blues', annot_kws={"size":
16},fmt='.2%')
plt.show()


# Model accuracy based on target result
def graph():
    import matplotlib.pyplot as plt
    data=[a]
    alg="RANDOM FOREST CLASSIFIER"
    plt.figure(figsize=(5,5))
    b=plt.bar(alg,data,color=("gold"))
    plt.title("THE ACCURACY SCORE OF RANDOM FOREST CLASSIFIER
IS\n\n\n")
    plt.legend(b,data,fontsize=9)
graph()
# Save as Model File
import joblib
joblib.dump(RFC, 'CIRRHOSIS.pkl')
```

Module 4: Gradient Boosting Classifier Algorithm

```
# Import required libaries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
# Read csv file
df = pd.read_csv('KIDNEY.csv')
df.head()
# Drop Missing values
df=df.dropna()
# Display last five rows in the table
df.tail()
# split the dataset into dependent and independent variables
x1 = df.drop(labels='Class', axis=1)
y1 = df.loc[:,'Class']
# Check for imbalanced dataset
import imblearn
from imblearn.over_sampling import RandomOverSampler
from collections import Counter
ros =RandomOverSampler(random_state=42)
x,y=ros.fit_resample(x1,y1)
print("OUR DATASET COUNT         : ", Counter(y1))
print("OVER SAMPLING DATA COUNT  : ", Counter(y))
# Specific grouping for testing and training dataset
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20,
random_state=42, stratify=y)
print("NUMBER OF TRAIN DATASET    : ", len(x_train))
print("NUMBER OF TEST DATASET     : ", len(x_test))
print("TOTAL NUMBER OF DATASET    : ", len(x_train)+len(x_test))
# Display the dataset after seperation
print("NUMBER OF TRAIN DATASET    : ", len(y_train))
print("NUMBER OF TEST DATASET     : ", len(y_test))
print("TOTAL NUMBER OF DATASET    : ", len(y_train)+len(y_test))
# import gradient boosting classifier
from sklearn.ensemble import GradientBoostingClassifier
# Fit the datasets for prediction
GRB = GradientBoostingClassifier(random_state=42)
GRB.fit(x_train,y_train)
# Prediction for the datasets
```

```
predicted = GRB.predict(x_test)
# Predict confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,predicted)
print('THE CONFUSION MATRIX SCORE OF GRADIENT BOOSTING
CLASSIFIER:\n\n\n',cm)
# make valdation for accuracy
from sklearn.model_selection import cross_val_score
accuracy = cross_val_score(GRB, x, y, scoring='accuracy')
print('THE CROSS VALIDATION TEST RESULT OF ACCURACY :\n\n\n',
accuracy*100)
# Accuracy score for model prediction
from sklearn.metrics import accuracy_score
a = accuracy_score(y_test,predicted)
print("THE ACCURACY SCORE OF GRADIENT BOOSTING CLASSIFIER
IS :",a*100)
# Loss based on model prediction
from sklearn.metrics import hamming_loss
hl = hamming_loss(y_test,predicted)
print("THE HAMMING LOSS OF GRADIENT BOOSTING CLASSIFIER IS
:",hl*100)
# Prcision based on model result
from sklearn.metrics import precision_score
P = precision_score(y_test,predicted)
print("THE PRECISION SCORE OF GRADIENT BOOSTING CLASSIFIER IS
:",P*100)
# Recall score for model prediction
from sklearn.metrics import recall_score
R = recall_score(y_test,predicted)
print("THE RECALL SCORE OF GRADIENT BOOSTING CLASSIFIER IS
:",R*100)
from sklearn.metrics import f1_score
f1 = f1_score(y_test,predicted)
print("THE PRECISION SCORE OF GRADIENT BOOSTING CLASSIFIER IS
:",f1*100)
def plot_confusion_matrix(cm, title='THE CONFUSION MATRIX SCORE OF
GRADIENT BOOSTING CLASSIFIER\n\n', cmap=plt.cm.Blues):
    target_names=['']
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(target_names))
    plt.xticks(tick_marks, target_names, rotation=45)
    plt.yticks(tick_marks, target_names)
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

cm=confusion_matrix(y_test, predicted)
print('THE CONFUSION MATRIX SCORE OF GRADIENT BOOSTING
CLASSIFIER:\n\n')
print(cm)

sns.heatmap(cm/np.sum(cm), annot=True, cmap = 'Blues', annot_kws={"size":
16},fmt='.2%')
plt.show()

def graph():
    import matplotlib.pyplot as plt
    data=[a]
    alg=" GRADIENT BOOSTING CLASSIFIER"
    plt.figure(figsize=(5,5))
    b=plt.bar(alg,data,color=("coral"))
    plt.title("THE ACCURACY SCORE OF GRADIENT BOOSTING
CLASSIFIER IS\n\n\n")
    plt.legend(b,data,fontsize=9)
graph()
import joblib
joblib.dump(GRB, 'KIDNEY.pkl')
```

View.py

```
from django.shortcuts import render, redirect
from . models import UserPersonalModel
from . forms import UserPersonalForm, UserRegisterForm
from django.contrib.auth import authenticate, login,logout
from django.contrib import messages
import numpy as np
import joblib

def Landing_1(request):
    return render(request, '1_Landing.html')

def Register_2(request):
    form = UserRegisterForm()
    if request.method == 'POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            user = form.cleaned_data.get('username') # Get the username from the form
            form.save()
            messages.success(request, f'Account was successfully created. {user}') # Use an f-
string to concatenate the message
            return redirect('Login_3')

    context = {'form': form,'messages':messages}
    return render(request, '2_Register.html', context)

def Login_3(request):
    if request.method =='POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(username=username, password=password)

        if user is not None:
            login(request, user)
            return redirect('Home_4')
        else:
            messages.info(request, 'Username OR Password incorrect')
    context = {}
    return render(request,'3_Login.html', context)

def Home_4(request):
    return render(request, '4_Home.html')
```

```
def Teamates_5(request):
    return render(request,'5_Teamates.html')

def Domain_Result_6(request):
    return render(request,'6_Domain_Result.html')

def Problem_Statement_7(request):
    return render(request,'7_Problem_Statement.html')

def Per_Info_8(request):
    if request.method == 'POST':
        fieldss = ['firstname','lastname','age','address','phone','city','state','country']
        form = UserPersonalForm(request.POST)
        if form.is_valid():
            print('Saving data in Form')
            form.save()
        return render(request, '4_Home.html', {'form':form})

    else:
        print('Else working')
        form = UserPersonalForm(request.POST)
        return render(request, '8_Per_Info.html', {'form':form})

Model = joblib.load('C:/Users/SPIRO25/Desktop/PROJECT/project/ITPML22 -
CIRRHOSIS/deploy/PROJECT/APP/CIRRHOSIS.pkl')
Model1 = joblib.load('C:/Users/SPIRO25/Desktop/PROJECT/project/ITPML22 -
CIRRHOSIS/deploy/PROJECT/APP/KIDNEY.pkl')
def Deploy_9(request):
    #CIRRHOSIIS
    if request.method == "POST":
        int_features = [x for x in request.POST.values()]
        int_features  = int_features[1:]
        print(int_features)
        final_features = [np.array(int_features, dtype=object)]
        print(final_features)
        prediction = Model.predict(final_features)
        print(prediction)
        output = prediction[0]
        print(f'output{output}')
        if output == 0:
            return render(request, '9_Deploy.html', {"prediction_text":"THE
DISEASE MIGHT NOT INFECT IN THIS CONDITION"})
        elif output == 1:
            return render(request, '9_Deploy.html', {"prediction_text":"THE CIRRHOSIS
DISEASE MIGHT INFECT IN THIS CONDITION"})
    else:
        return render(request, '9_Deploy.html')

def Per_Database_10(request):
    models = UserPersonalModel.objects.all()
    return render(request, '10_Per_Database.html', {'models':models})

def deploy(request):
    if request.method == "POST":
        int_features = [x for x in request.POST.values()]
        int_features  = int_features[1:]
        print(int_features)
final_features = [np.array(int_features, dtype=object)]
        print(final_features)
        prediction = Model1.predict(final_features)
        print(prediction)
        output = prediction[0]
        print(f'output{output}')
        if output == 0:
            return render(request, 'deploy.html', {"prediction_text":"THE KIDNEY DISEASE
MIGHT NOT AFFECT IN THIS CONDITION"})
        elif output == 1:
            return render(request, 'deploy.html', {"prediction_text":"THE KIDNEY DISEASE
MIGHT  AFFECT IN THS CONDITION"})
    else:
        return render(request, 'deploy.html')

def Logout(request):
    logout(request)
    return redirect('3_Login')
```

A3. SAMPLE SCREENSHOTS

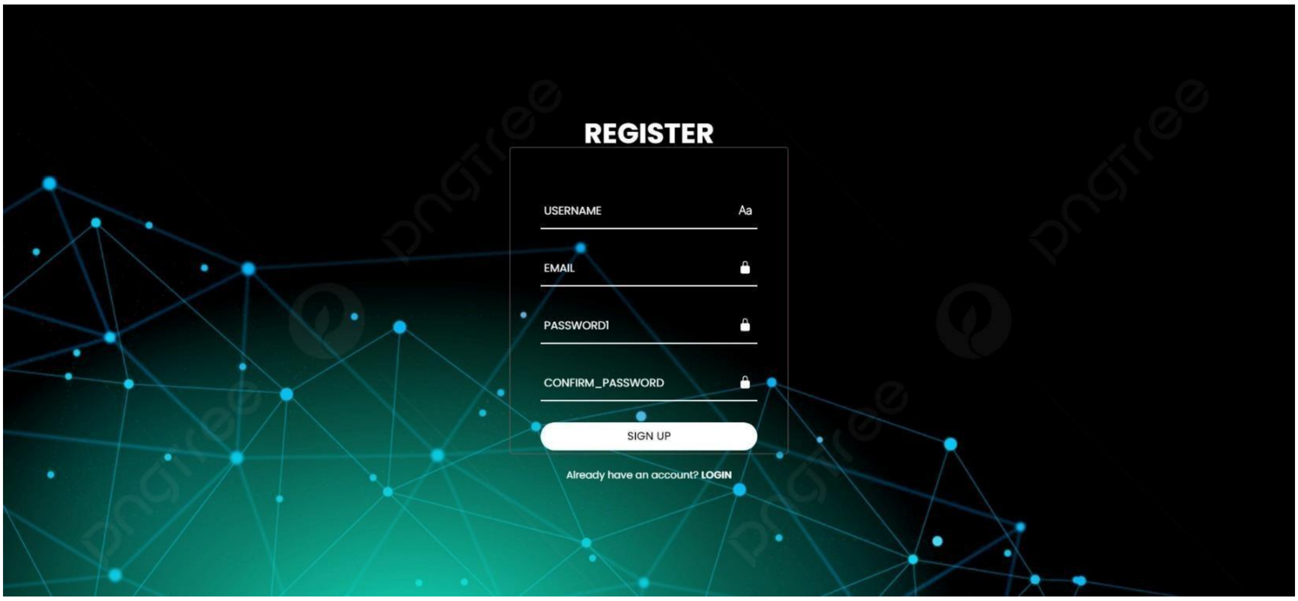
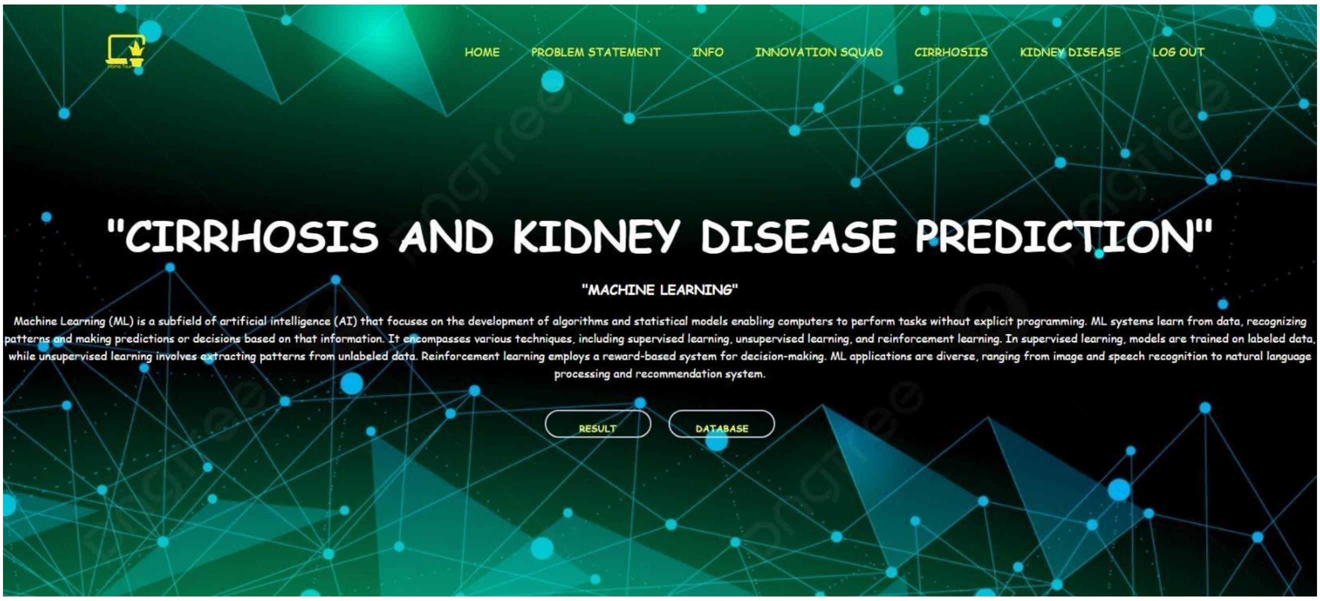


Figure A3.1 Sign up and Register Page



"CIRRHOSIS AND KIDNEY DISEASE PREDICTION"

"MACHINE LEARNING"

Machine Learning (ML) is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models enabling computers to perform tasks without explicit programming. ML systems learn from data, recognizing patterns and making predictions or decisions based on that information. It encompasses various techniques, including supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, models are trained on labeled data, while unsupervised learning involves extracting patterns from unlabeled data. Reinforcement learning employs a reward-based system for decision-making. ML applications are diverse, ranging from image and speech recognition to natural language processing and recommendation system.

Figure A3.2 Home Page

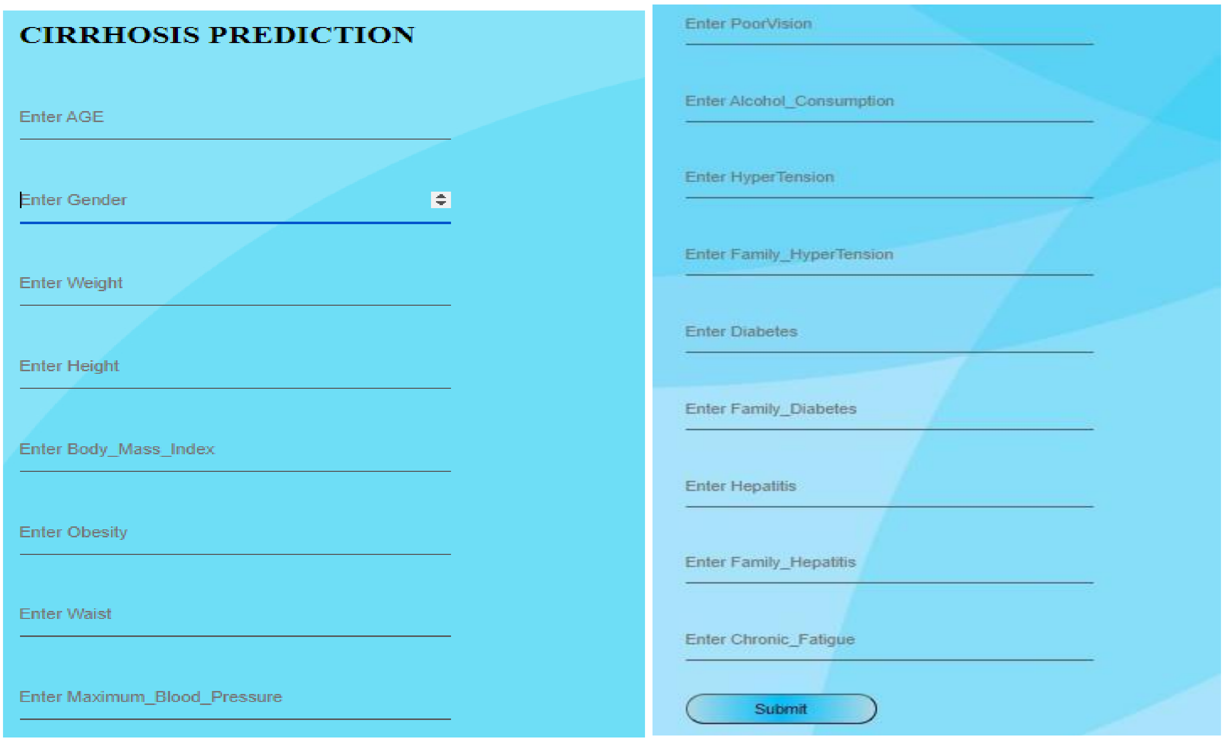


Figure A3.3 Liver Cirrhosis Prediction Page

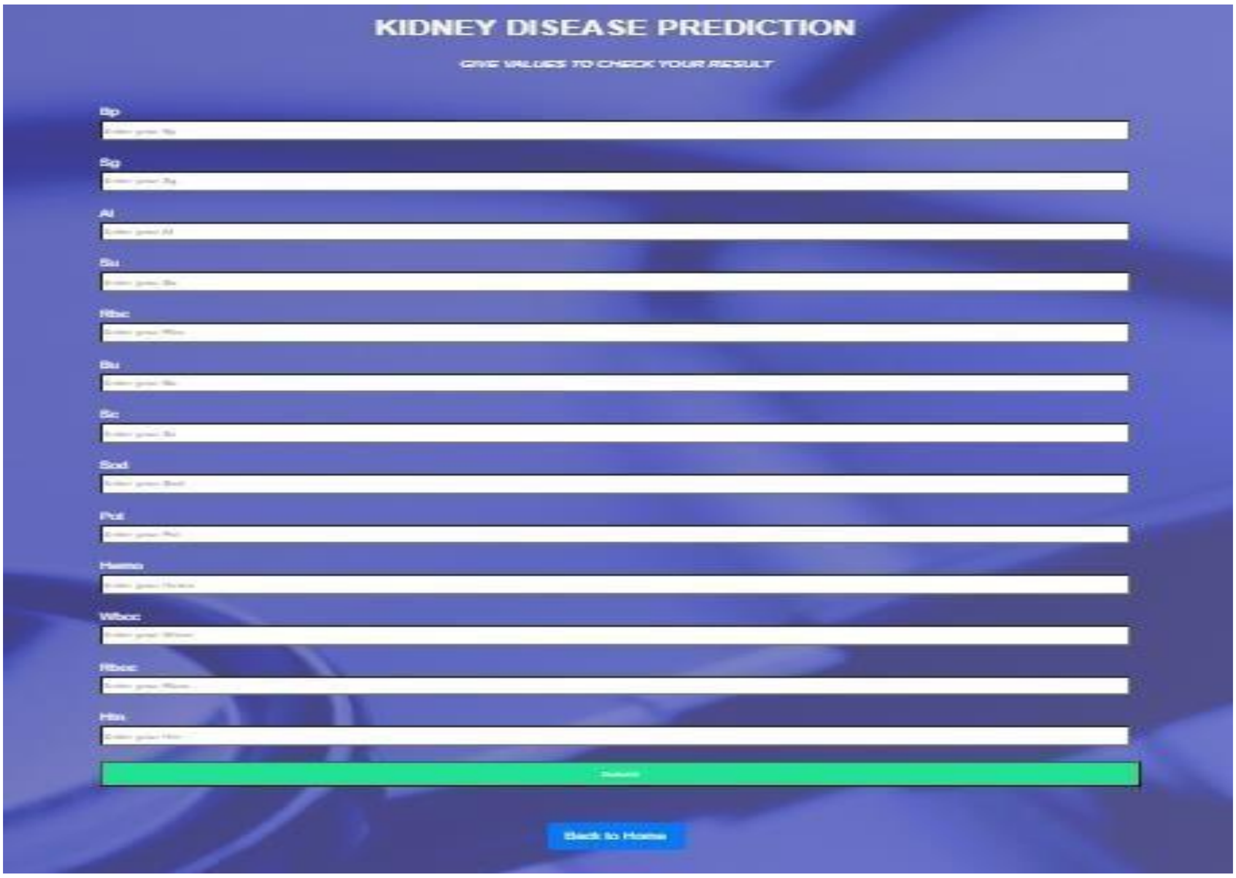


Figure A3.4 Kidney Disease Prediction Page

finalpaper report

by Plagiarism Checking

Submission date: 21-Mar-2024 04:20AM (UTC-0400)

Submission ID: 2326652792

File name: finalpaper.pdf (490.95K)

Word count: 4482

Character count: 25354

PREDICTIVE MODELING OF LIVER AND KIDNEY DISEASE: A COMPARATIVE ANALYSIS OF RANDOM FOREST AND GRADIENT BOOSTING ALGORITHM

Dheeksha J
Department of Computer Science and Engineering
Panimalar Engineering College
Chennai
dheekshareena@gmail.com

Mohana Priya S R
Department of Computer Science and Engineering
Panimalar Engineering College
Chennai
srmohanapriya11@gmail.com

Vijayalakshmi P
Assistant Professor
Department of Computer Science and Engineering
Panimalar Engineering College
Chennai
pasramviji@gmail.com

Indhumathi V
Department of Computer Science and Engineering Panimalar Engineering College
Chennai
indhumathivenkatesan@gmail.com

***Abstract*--Liver Cirrhosis and kidney disease are both critical health conditions that can greatly benefit from the application of machine learning techniques for early detection and improved patient care. Cirrhosis is characterized by the gradual deterioration of liver function, while kidney disease can lead to impaired renal function. Leveraging supervised machine learning models on comprehensive clinical and laboratory datasets can help identify key biomarkers and patterns associated with these diseases. A dataset comprising clinical and laboratory features of patients with and without Cirrhosis and kidney disease is collected. By analyzing patient data, these models can assist healthcare professionals in diagnosing these conditions earlier and customizing treatment plans for better patient outcomes. The main objective is to predict whether a person has cirrhosis and kidney disease in the early stage . Further research is warranted to validate these findings on larger and more diverse datasets and to integrate the models into clinical practice.**

Keywords— Liver cirrhosis,Machine learning,Kidney disease.

I.INTRODUCTION

The human liver lies in the upper section of the gastrointestinal tract. Its weight varies between 1400 and 1800 g in men and between 1200 and 1400 g in women. It carries out critical tasks for immunity, digestion, metabolism, excretion of pollutants, and nutrient storage. For this reason, certain liver illnesses may even be fatal. Based on their cause and impact on the liver, illnesses of the liver are classified. The cause could be a process, a hereditary disorder (such hemochromatosis), an infection, an injury, exposure to medications or other harmful substances, or another factor. The aforementioned conditions can result in cirrhosis, hepatitis, and kidney stones, which can become larger and obstruct blood vessels, infiltrate fat, and, in extreme circumstances, induce liver cancer. Additionally, Genetic disorders can also cause the accumulation and concentration of toxic substances, including iron or copper, as well as impair the liver's essential activities. [illegible] dangerous [illegible] conditions [illegible] replaces [illegible], non-alcoholic [illegible] chronic hepatitis C, and chronic hepatitis B are the primary causes of liver cirrhosis.

Millions of people die from kidney disease recently as a result of a lack of timely and reasonably priced treatment, which is a rapidly expanding illness. Patients with chronic renal disease are primarily from middle-class and lower-class nations that generate income. About a million people passed away from chronic renal disease in 2013. Many efforts have been made to identify chronic renal disease early on and begin treatment as soon as possible. When both kidneys are damaged, a frequent kind of kidney illness called chronic renal disease develops.

By using these algorithms effectively, patients can receive prompt treatment and the death rate can be reduced by early detection. Patients with chronic renal disease should maintain their clinical symptoms and engage in physical activity on a regular basis. They ought to stay away from junk food, exercise, and drink water.

II. LITERATURE SURVEY

In the 21st-century, the issue of liver disease has been increasing all over the world. As per the latest survey report, liver disease death toll has been rise approximately 2 million per year worldwide. The overall percentage of death by liver disease is 3.5% worldwide. Chronic Liver disease is also considered to be one of the deadly diseases, so early detection and treatment can recover the disease easily. The proposed model improved by applying a combination of three classifiers, Logistic regression, Random forest, and KNN algorithm and the result proved regarding accuracy that is achieved 77.58 percent.[2].The liver theatres vital role in many physical functions from protein manufacture and blood clotting to fat, sugar and iron metabolism.The study of paper to predicting and analysing liver disorder diseases to produce better performance accuracy by comparing various data mining classification algorithm and the performance of the accuracy is measured by confusion matrices [3]. Nazmun Nahar and Ferdous Ara research work explores the early prediction of liver disease using various decision tree techniques. The liver disease dataset which is select for this study is consisting of attributes like total bilirubin, direct bilirubin, age, gender, total proteins, albumin and globulin ratio [4]. Chronic Liver Disease is the leading cause of global death that impacts the massive quantity of humans around the world. This disease is caused by an assortment of elements that harm the liver. For example, obesity, an undiagnosed hepatitis infection, alcohol misuse. Which is responsible for abnormal nerve function, coughing up or vomiting blood, kidney failure, liver failure, jaundice, liver encephalopathy and there are many more. A comparative study on Liver Disease Prediction uses machine learning algorithm like SVM,KNN,Naïve Bayes and the performance of these techniques were estimated and it gives precision of around 53%[5]. Sumedh Sontakke, Jay Lohokare, Reshul Dani ,in their research work uses Back propagation and SVM algorithm to diagnose liver diseases and found that the accuracy of both algorithm is only 73%.[6].A supervised learning algorithm is used , and one has to manually retrain the model every time and the model utilizes large datasets collected by healthcare industries to automate the diagnosis of liver diseases[7].

Machine Learning Hybrid Model for the Prediction of Chronic Kidney Disease consists of GaussianNaive Bayes, gradient boosting, and decision tree to predict chronic kidney disease.[8].Liver malady is an overall medical issue that is related with different inconveniences and high mortality. It is of basic significance that illness be recognized before such huge numbers of these lives can be spared. In the work of Muktevi Srivenkatesh, utilized a few learning calculation K-Nearest Neighbour, Support Vector Machines, Logistic Regression, Navi Bayes, Random Forest to foresee patients with constant liver disappointment infection, and patients who are not experiencing this illness.[9]. [10] made two attempts to use the Cleveland Clinical Foundation's Coronary Heart Disease Dataset to identify heart disease. One method involves employing a solitary data mining algorithm on the heart disease dataset and comparing its accuracy to the baseline. Additionally, in the second strategy, they experimented with bagging algorithms and hybrid algorithms like the J4.8 decision tree. It was suggested that the accuracy of data mining algorithms might be improved by applying two voting techniques: varied data discretization levels and reduced error. With their methods, they were able to reach an accuracy of up to 84.1%, and they stated that more study is being done. Additionally, the University of California, Irvine (UCI) machine learning repository's heart disease dataset was examined in [11]. Using an ensemble of classifiers, the authors' adaptive boosting approach was able to achieve the best accuracy of 96% following a series of trials on the Hungarian Institute of Cardiology (HIC) dataset. The UCI repository and Shaheed Mohtarma Benazir Bhutto Medical University are two real-world datasets that the researchers in [12] used to diagnose a variety of thyroid illnesses, including euthyroid, hypothyroid, hyperthyroid, subclinical hypothyroid, and subclinical hyperthyroid. There are two classifiers employed, such as binary and multi-class SVM. The system's accuracy, using 10-k fold cross-validation, is 95.7%. In addition to heart disease and thyroid issues, research has been conducted on a number of other illnesses, including diabetes, liver problems, dengue, hepatitis, and breast cancer.Using machine learning algorithms, Chieh-Chen Wu et al. [13] have predicted liver illness, which could help doctors diagnose individuals who are not at high risk. The comparison of the classification algorithms according to their performance factors is the goal of Joel Jacob1 et al. [14]. Python has been used to create a graphical user interface that will assist the medical community in diagnosing liver illness in patients. Physicians and other healthcare professionals can easily use the GUI as a screening tool for liver disease. The following is a summary of a related survey of machine learning approaches that have been applied to different liver disease datasets by V.V. Ramalingam, A. Pandian, and R. Ragavendran[15]. Ferdous Ara and Nazmun Nahar [16] have computed and compared the effectiveness of several decision tree approaches. Decision tree approaches such as J48, LMT, Random Forest, Random Tree, REPTree, Decision Stump, and Hoeffding Tree have been employed by them. The investigation demonstrates that, in comparison to other approaches, Decision Stump offers the highest accuracy. Using classification algorithms, Dr. S. Vijayarani, S. Dhayanand [17] has predicted liver disorders. They have worked with support vector machines (SVM) and naïve bayes. The classification accuracy and execution time of these classifier algorithms are the performance variables that are used for comparison. The results of the experiment indicate that the Support Vector Machine (SVM) is a more accurate classifier for predicting liver disorders.

III. PROPOSED METHODOLOGY

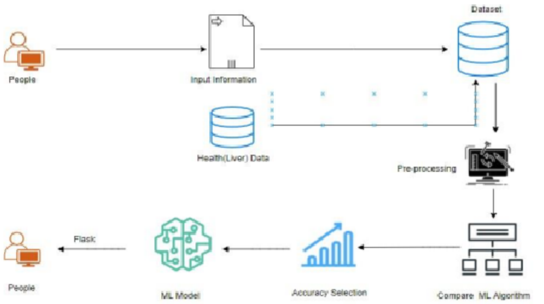


Figure 1. System architecture of proposed Prediction model

Figure 1 depicts the application's flow. We proposed a system to develop the project using machine learning algorithm. Artificial intelligence and machine learning have recently become major factors in the growth and development of many different sectors. Thus, we attempted to apply machine learning algorithms for the identification of renal disease and cirrhosis. We gathered the prior medical records of patients with kidney disease and cirrhosis, including those who were symptomatic as well as those who were not. By gathering their personal data, our system attempts to use a variety of computations to find patterns in the datasets. Once the pattern has been identified by a variety of machine learning algorithms, the model can use the historical data to forecast the occurrence.

ADVANTAGES:

- We build a production level application for deployment purpose.

- We build an advance machine learning techniques to build a predictive model.
- We compared more than a two architecture to getting better accuracy level.
- We train the structured data for machine learning model.

DATASETS

Here we are using 2 different dataset to predict two diseases.

1)Cirrhosis Dataset

This dataset consists of 8786 records of patients combining those who had cirrhosis and those who are not having cirrhosis and the dataset attributes include age, gender, region, Weight, height, BMI, Obesity, Waist, Maximum and Minimum Blood pressure, Good Cholestrol, bad Cholestrol, Total Cholestrol, Dyslipidemia,PVD, Physical Activity, Education,Unmarried, Income, Source of care, Poor Vision, Alcohol Consumption, Hypertension, Family Hypertension, Diabetes ,Family Diabetes, Hepatitis ,Family Hepatitis, Chronic Fatigue,ALF.

2)Kidney Disease Dataset

This dataset consists of 401 records of patients including those who had kidney disease and those who are not having and the dataset attributes includes BP, SG, AI, Su, Rbc, Bu, Sod, Pot, Hemo, Wbcc, Rbcc, Htn, class.

Table 1.Kidney disease dataset attributes and attribute information

Sno	Attribute	Attribute information
1.	BP	Blood Pressure
2.	SG	Specific Gravity
3.	Al	Albumin level
4.	Su	Salicycluric acid
5.	Rbc	Hematuria disease
6.	Bu	Blood urea nitrogen
7.	Sod	Sodium level
8.	Pot	Potassium level
9.	Hemo	Haemoglobin
10.	Wbcc	White blood cell count
11.	Rbcc	Red Blood cell count
12.	Htn	Hypertensive Nephropathy
13.	Class	Kidney disease/not

SYSTEM IMPLEMENTATION

1.Data Pre-processing

The machine learning (ML) model's error rate, which is as near to the actual error rate of the dataset as possible, is obtained through validation approaches.Identifying duplicate values, missing values, and the data type—integer or float—must be done. A time-consuming to-do list may result from data collection, analysis, and the process of addressing the content, quality, and structure of the data. Understanding data and its characteristics is helpful throughout the data identification phase since it will help you decide which algorithm to employ to develop a model. Several distinct data cleaning jobs utilizing Python's Pandas library; in particular, it focuses on missing values, which is perhaps the largest data cleaning work, and it can clean data more quickly. It would rather spend more time investigating and modeling than cleaning data. A few of these sources are merely careless errors. In some cases, missing data may have deeper cause. It's critical to comprehend these various kinds of missing data from a statistical perspective. The nature of the missing data will determine how to handle detecting and filling in the gaps, as well as how to handle basic imputation and more complex statistical methods. Prior to integrating into code, it's crucial to understand the sources of missing data. Here are some typical reasons why data is missing:

• The user neglected to complete a field.

• When manually moving data from a legacy database, some was lost.

• A programming error occurred.

• Due to their concerns regarding the use or interpretation of the results, users choose not to fill out a particular field.

Process of Data Validation, Cleaning,and Preparation

Importing the specified dataset while importing the library packages. Analyzing the variable identification by data type and shape, assessing duplicate and missing values, etc. Data cleaning procedures and methods differ depending on the type of dataset. Finding and eliminating mistakes and abnormalities is the main objective of data cleaning, which aims to improve the value of data for analytics and decision-making.

2.Data analysis and Visualization

In applied statistics and machine learning, data visualization is a critical competency. An essential set of tools for developing a qualitative understanding is provided by data visualization. This can be useful for discovering trends, faulty data, outliers, and much more while examining and getting to know a dataset. Plots and charts that are more visceral and stakeholders than measures of association or significance can express and illustrate important relationships with the use of data visualizations, provided the user has some topic knowledge. When data is presented visually, like with charts and graphs, it can sometimes make sense. In applied statistics as well as applied machine learning, the ability to visualize data samples and other types of information rapidly is crucial. It will reveal the various plot types that you should be aware of when using Python to visualize data and show you how to apply them to enhance your understanding of your own data.

The changes we make to our data prior to supplying it to the algorithm are referred to as pre-processing. Information One method for transforming the raw data into a clean data set is preprocessing. Put differently, anytime data is obtained in raw format from several sources, it is not suitable for analysis. The data must be entered correctly for the machine learning process to yield improved outcomes from the applied model. Certain machine learning models, like the Random Forest algorithm, require data in a certain format.Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Thus, null values from the initial raw data collection must be controlled in order to run the random forest method. Additionally, the format of the data collection should be such that multiple Machine Learning and Deep Learning algorithms can be run on the supplied dataset.

False Positives (FP): When the actual class is no and predicted class is yes. E.g. if actual class reports this passenger did not survive but predicted class reports you that this passenger will survive.

False Negatives (FN): When the actual class is yes but predicted class says no. E.g. if actual class value says that this passenger survived and predicted class reports you that passenger will die.

True Positives (TP): These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value tells that this passenger survived and predicted class indicates you the same thing.

True Negatives (TN): These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class indicates this passenger did not survive and predicted class also says the same thing.

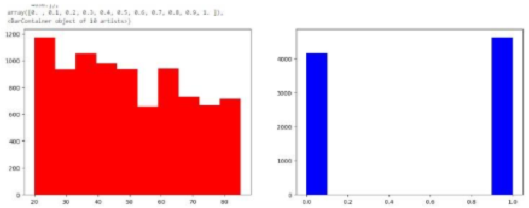


Figure 1.Histogram Plot for Age and Gender

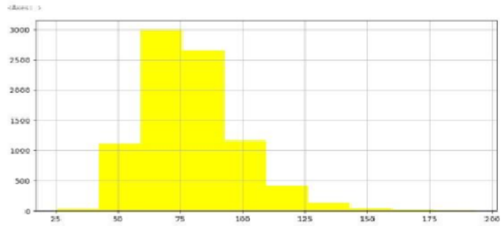


Figure 2 Histogram plot for weight

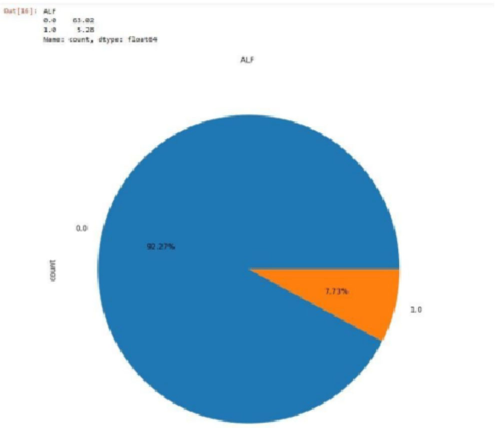


Figure 3 Pie-chart for target column by its count.

3. Algorithm and techniques

In machine learning and statistics, classification is a supervised learning approach where a computer program learns from the data it is fed and applies that knowledge to categorize new observations. This data collection can be multi-class as well as bi-class (e.g., indicating if the message is spam or not, or if the recipient is male or female). Speech recognition, handwriting recognition, biometric identification, document classification, and other areas are a few instances of classification challenges. Algorithms in supervised learning gain knowledge from labeled data. The algorithm first analyzes the data to identify which label to apply to fresh data, then associates the patterns with the unlabeled data.

Used Python Packages are

- sklearn
- NumPy
- Pandas
- Matplotlib

3.1 Gradient Boosting Classifier:

One of the most widely used technologies for creating predictive models for a variety of challenging regression and classification tasks is machine learning. One of the most potent boosting techniques is the gradient boosting machine (GBM). Because the boosting technique is based on the idea of ensemble learning, it generates the final result by combining several simple models, often known as base estimators or weak learners. In machine learning, GBM is also utilized as an ensemble approach to help turn weak learners into strong learners. The topic of "GBM in Machine Learning" will cover a variety of topics, including gradient machine learning algorithms, machine learning boosting techniques, the background and operation of GBM, terms used in GBM, etc. However, before you begin, familiarize yourself with the machine learning boosting principle and its different variants. [illegible] from t[illegible]. The residuals, or r[illegible] a new tree at each stage.

Benefits

• Strong prediction ability;
• Capable of handling both category and numerical data;
• Able to grasp intricate relationships within the data.

Challenges

• Hyperfitting may occur if hyperparameters are not properly specified.
• Hyperparameter adjustment may be necessary.

Gradient Boosting is a popular machine learning technique that has shown to be quite successful in a variety of tasks. Its capacity to gradually improve weak learners into strong models makes it an important tool in the toolbox of data scientists.Gradient Boosting algorithm is used for kidney disease prediction and the confusion matrix score of GRADIENT BOOSTING CLASSIFIER is found out to be [[50 0] [1 49]].

3.2 Random forest Algorithm

One well-known machine learning algorithm that is a part of the supervised learning approach is Random Forest. It can be applied to ML issues involving both classification and regression. Its foundation is the idea of ensemble learning, which is the process of merging several classifiers to solve a challenging issue and enhance the model's functionality.

According to its name, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Rather than depending on a single decision tree, the random forest forecasts the outcome based on the majority vote of projections from each tree. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Random Forest Assumptions

Certain decision trees might anticipate the right output while others might not because the random forest combines numerous trees to estimate the dataset's class. However, when combined, every tree forecasts the right result. Consequently, the following two presumptions lead to an improved Random forest classifier:

o The dataset's feature variable needs to contain some real values in order for the classifier to forecast precise results as opposed to conjectured ones.

o There must be extremely little association between any tree's predictions.

o Compared to other algorithms, it requires less training time.

o It operates efficiently even for big datasets, predicting output with high accuracy.

o It can also retain accuracy when a large proportion of data is missing.

The two main phases of execution are the creation of the random forest from the combination of N decision trees and the prediction of each tree generated in the first phase.

The stages listed below can be used to explain the working process.

Step 1: From the training set, choose K data points at random.

Step 2:Create the decision trees linked to the chosen data points (subsets) in step two.

Step 3: Select the number N for the decision trees you wish to construct.

Step 4: Carry out Steps 1 and 2.

Step 5: Locate each decision tree's predictions for the new data points, then allocate them to the group receiving the majority of votes.

Thus Random forest algorithm is used for liver failure and the confusion matrix score is found out to be[[794 13] [0 807]].

4. Deployment

A high-level Python web framework called Django promotes efficient development and simple, straightforward design. It handles a lot of the bothersome aspects of web development and was built by seasoned developers, allowing you to concentrate on developing your app instead of having to start from scratch. It is open source and free.Django was designed to help developers take applications from concept to completion as quickly as possible. Django takes security seriously and helps developers avoid many common security mistakes. Dozens of add-ons come with Django that you can use for managing standard web development tasks. Right out of the box, Django handles a host of functions including user authentication, content management, site mapping, RSS feeds, and much more. Because Django takes security seriously, it assists developers in avoiding several typical security errors, including clickjacking, SQL injection, cross-site scripting, and cross-site request forgery. Managing user accounts and passwords is made safe by its user authentication method. incredibly adaptable Django has been used by businesses, groups, and governments to create a wide range of applications, including social networks, scientific computing platforms, and content management systems.

IV. RESULT AND DISCUSSIONS

In order to predict a value, the logistic regression technique also employs a linear equation with independent predictors. Anywhere from negative infinity to positive infinity can be the expected value.. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

True Positive Rate (TPR) = TP / (TP + FN)
False Positive rate (FPR) = FP / (FP + TN)

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and nondefaulters.

A.Accuracy calculation

Accuracy = (TP + TN) / (TP + TN + FP + FN)

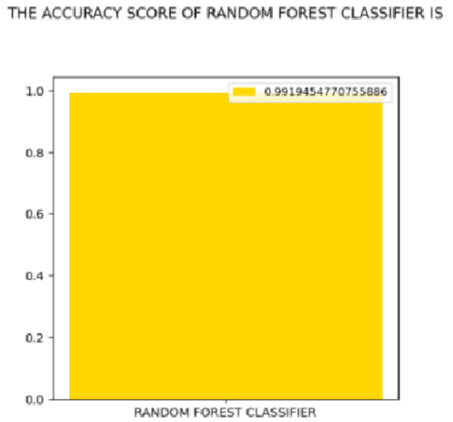


Figure 4 Accuracy score of Random forest classsifier

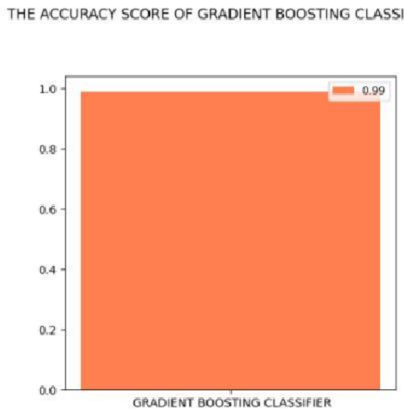


Figure 5 Accuracy score of Gradient Boosting Classifier

Precision:The proportion of positive predictions that are actually correct Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = TP / (TP + FP)

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = TP / (TP + FN)

The recall score of random forest Classifier is : 100.0

The recall score of gradient boosting Classifier is : 98.0

General Formula

F- Measure = 2TP / (2TP + FP + FN)

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

B.Webpage Outlook

"CIRRHOSIS AND KIDNEY DISEASE PREDICTION"

"MACHINE LEARNING"

Figure 6 Webpage

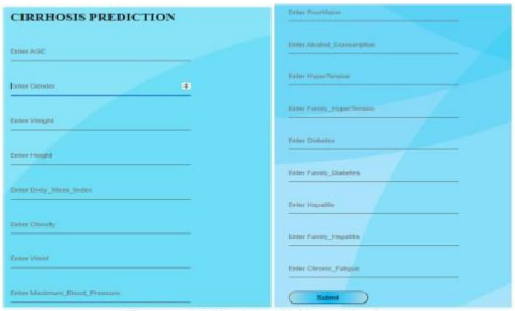


Figure 7 Cirrhosis Prediction

Figure 8 Kidney Disease Prediction

V.CONCLUSION

In conclusion, the application of advanced machine learning techniques, specifically Random Forest and Gradient Boosting, has proven to be instrumental in predicting the onset of liver cirrhosis and kidney disease. Through the analysis of relevant medical data, these models demonstrated a high level of accuracy and reliability in identifying key indicators and patterns associated with the progression of these conditions.The Random Forest algorithm, with its ensemble approach and ability to handle complex datasets, showcased robust predictive performance. Similarly, Gradient Boosting, through its iterative optimization process, exhibited superior predictive power by capturing subtle relationships within the data and improving overall model precision.The successful implementation of these machine learning techniques not only enhances our understanding of the factors contributing to liver cirrhosis and kidney disease but also holds significant promise for early detection and intervention. This, in turn, could lead to improved patient outcomes and more efficient healthcare resource allocation.

REFERENCES

[1] Zhangdaihong Liu,Xuan Wu, Yang Yang,David A.Clifton (2022)DUKA for Multi-modality EHR Data Fusion and Organ Failure Prediction

[2] Golmei Shaheamlung, Harshpreet Kaur(March 2021) The Diagnosis of Chronic Liver Disease using Machine Learning Techniques

[3] Shambel Kefelegn, Pooja Kamat(2018) Prediction and Analysis of Liver Disorder Diseases by using Data Mining Technique: Survey

[4] Nazmun Nahar and Ferdous Ara((2018) Liver disease prediction by using different decision tree techniques

[5] A.K.M Sazzadur Rahman, F. M. Javed Mehedi Shamrat, Zarrin Tasnim, Joy Roy, Syed Akhter Hossain(Nov 2019) A Comparative Study On Liver Disease Prediction Using Supervised Machine Learning Algorithms

[6] Sumedh Sontakke, Jay Lohokare, Reshul Dani(Feb 2017) Diagnosis of Liver Diseases using Machine Learning

[7] Adekola Olubukola Daniel ,Ekanem Edikan Uwem , Omidiran Daniel Tolulope(2020) Prediction and Diagnosis of Liver Disease in Human Using Machine Learning

[8] Hira Khalid,Ajab Khan,Muhammad Zahid Khan,Gulzar Mehmood(March 2023) Machine Learning Hybrid Model for the Prediction of Chronic Kidney Disease

[9] Mai, S., Tim, T., & Rob, S., Using Data Mining Techniques in Heart Disease Diagnosis and Treatment. Northcott Drive: University of New South Wales at the Australian Defense Force Academy, (2012).

[10] Kathleen, M. H., Julia, M. H., & George, M. J., Diagnosing Coronary Heart Disease Using Essemble Machine Learning. International Journal of Adanced Computer Science and Application, (2016).

[11] Soomrani, R., Adul, M., & Jamil, A., Thyroid Disease Type Diagnostics. Sukkur: Sukker Institute of Bussiness Administeration, (2016).

[12] Chieh-ChenWu et.al —Prediction of fatty liver using machine learning algorithms‖, Computer methods and medicines in Bio Medicines .

[13] Joel Jacob, Joseph Chakkalakal Mathew, Johns Mathew, Elizabeth Issac, —Diagnosis of Liver Disease Using Machine Learning Techniques‖ International Research Journal of Engineering and Technology (IRJET) ,Volume: 05 Issue: 04 , Apr-2018

[14] V.V. Ramalingam , A.Pandian, R. Ragavendran, —Machine Learning Techniques on Liver Disease - A Survey‖, International Journal of Engineering & Technology, 7 (4.19) (2018) 485-495

[15] Nazmun Nahar, Ferdous Ara , —Liver Disease Prediction using by using different decision tree techniques‖ International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.8, No.2, March 2018.

[16] Dr. S. Vijayarani, S.Dhayanand , —Liver Disease Prediction using SVM and Naïve Bayes Algorithms‖, International Journal of Science, Engineering and Technology Research (IJSETR) Volume 4,Issue 4 ,April 2015,816-820

[17] Shambel Kefelegn, Pooja Kamat, —Prediction and Analysis of Liver Disorder Diseases by using Data Mining Technique: survey ‖,International Journal of pure and applied mathematics ,volume 118,No 9,765-770 ,2018

[18] k. Thirunavukkarasu ; Ajay S. Singh ; Md Irfan ; Abhishek Chowdhury, —Prediction of Liver Disease using Classification Algorithms‖, International Conference ,2018.

finalpaper report

ORIGINALITY REPORT

14%	15%	19%	23%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Manipal University Student Paper	2%
2	it-in-industry.org Internet Source	2%
3	www.ijartet.com Internet Source	1%
4	Submitted to University of Greenwich Student Paper	1%
5	ijarcce.com Internet Source	1%
6	technodocbox.com Internet Source	1%
7	Submitted to Technological University Dublin Student Paper	1%
8	www.researchgate.net Internet Source	1%
9	Submitted to Canterbury Christ Church University Student Paper	1%

10	Submitted to Institute of International Studies Student Paper	1%
11	Submitted to Kakatiya Institute of Technology and Science Student Paper	1%
12	Submitted to University of Nigeria Student Paper	1%
13	S Vidhya, V. Siva Vadivu Ragavi, J. K. Monica, B . Kanisha. "Chapter 24 Milk Quality Prediction Using Supervised Machine Learning Technique", Springer Science and Business Media LLC, 2023 Publication	1%
14	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
15	Submitted to Ibri College of Technology Student Paper	1%
16	Submitted to Taylor’s Education Group Student Paper	1%
17	www.ijitee.org Internet Source	1%
18	ijcttjournal.org Internet Source	1%
19	sourceforge.net Internet Source	1%

20	www.ijert.org Internet Source	1%
21	www.iosrjournals.org Internet Source	1%
22	"Artificial Intelligence: Theory and Applications", Springer Science and Business Media LLC, 2024 Publication	1%
23	Submitted to Coventry University Student Paper	1%
24	www.atlantis-press.com Internet Source	1%
25	ijcsmc.com Internet Source	1%
26	Submitted to Vel Tech University Student Paper	1%
27	www.ijstr.org Internet Source	1%
28	www.mdpi.com Internet Source	1%
29	journals.resaim.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

REFERENCES

[1] Zhangdaihong Liu,Xuan Wu, Yang Yang,David A.Clifton (2022)DUKA for Multi-modality EHR Data Fusion and Organ Failure Prediction

[2] Golmei Shaheamlung, Harshpreet Kaur(March 2021) The Diagnosis of Chronic Liver Disease using Machine Learning Techniques

[3] Shambel Kefelegn, Pooja Kamat(2018) Prediction and Analysis of Liver Disorder Diseases by using Data Mining Technique: Survey

[4] Nazmun Nahar and Ferdous Ara((2018) Liver disease prediction by using different decision tree techniques

[5] A.K.M Sazzadur Rahman, F. M. Javed Mehedi Shamrat, Zarrin Tasnim, Joy Roy, Syed Akhter Hossain(Nov 2019) A Comparative Study On Liver Disease Prediction Using Supervised Machine Learning Algorithms

[6] Sumedh Sontakke, Jay Lohokare, Reshul Dani(Feb 2017) Diagnosis of Liver Diseases using Machine Learning

[7] Adekola Olubukola Daniel ,Ekanem Edikan Uwem , Omidiran Daniel Tolulope(2020) Prediction and Diagnosis of Liver Disease in Human Using Machine Learning

[8] Hira Khalid,Ajab Khan,Muhammad Zahid Khan,Gulzar Mehmood(March 2023) Machine Learning Hybrid Model for the Prediction of Chronic Kidney Disease

[9] Mai, S., Tim, T., & Rob, S., Using Data Mining Techniques in Heart Disease Diagnosis and Treatment. Northcott Drive: University of New South Wales at the Australian Defense Force Academy, (2012).

[10] Kathleen, M. H., Julia, M. H., & George, M. J., Diagnosing Coronary Heart Disease Using Essemble Machine Learning. International Journal of Adanced Computer Science and Application, (2016).

[11] Soomrani, R., Adul, M., & Jamil, A., Thyroid Disease Type Diagnostics. Sukkur: Sukker Institute of Bussiness Administeration, (2016).

[12] Chieh-ChenWu et.al ―Prediction of fatty liver using machine learning algorithms‖, Computer methods and medicines in Bio Medicines .

[13] Joel Jacob, Joseph Chakkalakal Mathew, Johns Mathew, Elizabeth Issac, ―Diagnosis of Liver Disease Using Machine Learning Techniques‖ International Research Journal of Engineering and Technology (IRJET) ,Volume: 05 Issue: 04 , Apr-2018

[14] V.V. Ramalingam , A.Pandian, R. Ragavendran, ―Machine Learning Techniques on Liver Disease - A Survey‖, International Journal of Engineering & Technology, 7 (4.19) (2018) 485-495

[15] Nazmun Nahar, Ferdous Ara , ―Liver Disease Prediction using by using different decision tree techniques‖ International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.8, No.2, March 2018.

[16] Dr. S. Vijayarani, S.Dhayanand , ―Liver Disease Prediction using SVM and Naïve Bayes Algorithms‖, International Journal of Science, Engineering and Technology Research (IJSETR) Volume 4,Issue 4 ,April 2015,816-820

[17] Shambel Kefelegn, Pooja Kamat, ―Prediction and Analysis of Liver Disorder Diseases by using Data Mining Technique: survey ‖,International Journal of pure and applied mathematics ,volume 118,No 9,765-770,2018

[18] k. Thirunavukkarasu ; Ajay S. Singh ; Md Irfan ; Abhishek Chowdhury, ―Prediction of Liver Disease using Classification Algorithms‖, International Conference ,2018.