HX8001 - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

STATISTICAL MACHINE LEARNING APPROACHES TO LIVER DISEASE PREDICTION

A PROJECT REPORT

Submitted By

TEAM ID : PNT2022TMID14539
TEAM MEMBER : B.VENKATA SRAVAN KUMAR
TEAM MEMBER : B.SUDEEPTH
TEAM MEMBER : B.AAKASH
TEAM MEMBER : CH.V.S.YASWANTH KUMAR

In partial fulfillment for the award of the degree
Of

DEPARTMENI OF ELECTRONICS AND COMMUNICATION AND ENGINEERING

RMK COLLEGE OF ENGINEERING AND TECHNOLOGY

CHENNAI

ANNAUNIVERSITY : CHENNAI 600 02

NOV -DEC2022

ABSTRACT

Liver diseases has increased tremendously and it is considered to be very fatal disease in many countries. It becomes really difficult for the researchers to predict the disease from voluminous medical databases. To tackle this issue researchers use machine learning techniques such as classification, clustering and so on. The main aim of this project is to predict liver diseases using different classification algorithms. These classification algorithms are compared based on the performance factors i.e. classification accuracy and execution time. From the project we would observe which algorithm is better classifier to predict liver disease. A GUI, which can be used as a medical tool by hospitals and medical staff is implemented using the best classifier.

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. Diagnosis of liver infection at preliminary stage is important for better treatment. In today's scenario devices like sensors are used for detection of infections. Accurate classification techniques are required for automatic identification of disease samples.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 five algorithms Logistic Regression, Decision Tree, Support Vector Machine, Naïve Bayes, and Random Forest. The performance of different classification techniques was evaluated on different measurement techniques such as accuracy, precision, recall, and specificity.

TABLE OF CONTENT

CHAPTER	CONTENTS	PAGE NO
1.	**INTRODUCTION** 1.1 PROJECT OVERVIEW 1.2 PURPOSE	05
2.	**LITERATURE SURVEY** 2.1 EXISTING PROBLEM 2.2 REFERENCES 2.3 PROBLEM STATEMENT DEFINITION	06
3.	**IDEATION & PROPOSED SOLUTION** 3.1 EMPATHY MAP CANVAS 3.2 IDEATION & BRAINSTROMING 3.3 PROPOSED SOLUTION 3.4 PROBLEM SOLUTION FIT	09
4.	**REQUIREMENT ANALYSIS** 4.1 FUNCTIONAL REQUIREMENT 4.2 NON-FUNCTIONAL REQUIREMENTS	12
5.	**PROJECT DESIGN** 5.1 DATA FLOW DIAGRAMS 5.2 SOLUTION & TECHNICAL ARCHITECTURE 5.3 USER STORIES	14
6.	**PROJECT PLANNING & SCHEDULING** 6.1 SPRINT PLANNING & ESTIMATION	17

6.2 SPRINT DELIVERY SCHEDULE

7. CODING & SOLUTIONING **20**
7.1 FEATURE 1
7.2 DATABASE SCHEMA

8. TESTING **22**
8.1 TEST CASES
8.2 USER ACCEPTANCE TESTING

9. RESULTS **24**
9.1 PERFORMANCE METRICS

10. ADVANTAGES & DISADVANTAGES **27**

11. CONCLUSION **28**

12. FUTURE SCOPE **29**

13. APPENDIX **30**
SOURCE CODE
GITHUB & PROJECT DEMO LINK

CHAPTER- 1

INTRODUCTION

As per the World Health Organization's latest survey report, death due to liver disease is 2.95% of total death and India ranks 63rd position in the world. The main important role of the liver is to remove the toxic and harmful substances from the blood before distribution to different parts of our body. Liver disease is considered as one of the most dangerous and deadliest diseases faced in the globe. The reason behind the causes of liver disease are as follows, liver fibrosis, fatty liver, liver cirrhosis, hepatitis infection, excessive alcohol drink, drug and genetic abnormalities.

It is very difficult to identify liver disease at early stage when liver tissues has damaged moderately. This leads to failure in treatment and medication. If liver is 100% fail there is no option to recover but only one solution is available which is liver transplantation. Early detection of liver disease can help in treatment. There are different symptoms of chronic liver disease which include digestion problem including abdominal pain, dry mouth, constipation and internal bleeding, Dermatological issues like yellowish skin color, spider like veins, redness on feet and Brain and Nervous system abnormalities like memory problem, numbness and fainting.

Early prediction is crucial to give proper treatment and save life of patient. It can be easily done with the use of expert system. With the repeated improvements in Artificial intelligence different types of machine learning algorithms has been developed which helps in improving the quality and accuracy of the detection or prediction of the liver disease.

1.1. PROJECT OVERVIEW

Patients with liver disease have been continuously increasing because of excessive consumption of alcohol ,inhale of harmful gases, intake of contaminated food and drugs. To detect disease, Health Care Professionals need to collect samples from patients which can cost both time and money. The main problem is doctors cannot diagnose on the basis of variations in test results.

1.2 . PURPOSE

In this application, by using patient records that includes blood test report (some like age ,gender are already known)and then you will get the results of prediction.Thesystem predicts the results with 99%accuracy for the dataset that we have used while creating this application.

CHAPTER- 2

LITERATURE SURVEY

(i) Robin Bijou Department of Computer Application, Musial College of Engineering & Technology, Pathanamthitta, Kerala The APJ Abdul kalam Technological University, Statistical Machine Learning Approaches to Liver Disease Prediction, Volume 5 Issue 4, July-August 2022, Available at www.ijsred.com

Today, everyone's health is a very essential concern, so it is necessary to offer medical services that are freely accessible to everyone. The primary goal of this study is to forecast liver illness using a software engineering methodology that makes use of feature selection and classification techniques. The Indian Liver Patient Dataset (ILPD) from the University of California, Irvine database is used to carry out the proposed research. The many variables of the liver patient dataset, including age, direct bilirubin, gender, total bilirubin, Alkphos, sgpt, albumin, globulin ratio, and sgot, among others, are used to forecast the risk level of liver illnesses. On the Liver Patient dataset, several classification techniques are applied to determine accuracy, including Logistic Regression, Sequential Minimal Optimization, and K-Nearest Neighbor.

(ii) Fahad Mostafa , Easin Hasan , Morgan Williamson and Hafiz Khan, Statistical Machine Learning Approaches to Liver Disease Prediction, 1 December 2021 Published.

More widespread application of these methods to varying data sets can further improve accuracy in current deep learning methods. This study aimed to (i) impute missing data using the MICE algorithm; (ii) determine variable selection using eigen decomposition of a data matrix by PCA and to rank the important variables using the Gini index; (iii) compare among several statistical learning methods the ability to predict binary classifications of liver disease; (iv) use the synthetic minority oversampling technique (SMOTE) to oversample minority class to regulate over fitting; (v) obtain confusion matrices for comparing actual classes with predictive classes; (vi) compare several ML approaches to assess a better performance of liver disease diagnosis; (viii) evaluate receiver operating

characteristic (ROC) curves for determining the diagnostic ability of binary classification of liver disease.

2.1 EXISTING PROBLEM

There is no real time identity verification. Not using data mining technique to detect chances of getting liver disease. Time intense. Certain approaches being applicable only for the small data .Some approaches are not adoptable for real time collection of database implementation. Only two systems exist in the same domain, according to a thorough investigation into the subject. First, the system is entirely manual. It has the capacity to store patient information and medical records. The initial system's key characteristics are as follows. The second system is more effective than the first. It was discovered from a related research study that the system is constructed utilizing the KNN method.

2.2 REFERENCES

[1] Asrani, S.K.; Devarbhavi, H.; Eaton, J.;Kamath, P.S. “Burden of liver diseases in the world”. J. Hepatol. 2019.

[2] Chalasani, N.; Younossi, Z.; Lavine,J.E.; Charlton, M.; Cusi,K.; Rinella, M.;Harrison, S.A.; Brunt, E.M.; Sanyal, A.J.“The diagnosis and management of-nonalcoholic fatty liver disease: Practice guidance from the American Associationfor the Study of Liver Diseases”.Hepatology 2018.

[3] Wang, Y.; Li, Y.; Wang, X.; Gacesa,R.; Zhang, J.; Zhou, L.; Wang, B.“Predicting Liver Disease Risk Using aCombination of Common ClinicalMarkers: A Screening Model from RoutineHealth Check-Up”. Dis. Markers 2020.

[4] https://www.irjet.net/archives/V5/i1/IRJET-V5I142.pdf.

[5]Decision Trees, Retrieve from:https://dataaspirant.com/2017/01/30/how-decision-treealgorithmworks/,Last Accessed: 5 Octobor,2019.

2.3 PROBLEM STATEMENT DEFINITION

Liver damage caused by cirrhosis can't be undone, but further damage can be limited. Treatments focus on the underlying cause. In advanced cases, a liver transplant may be required. Predicting the stage of cirrhosis and beginning the treatment before it's too late can prevent the fatal consequences of the disease.

Problem Statement (PS)	**Iam (User)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	User	Trying to Predict My Liver Disease	it's difficult to calculate	it takes more time	Challenging.
PS-2	User	Finding the Accuracy	it's hard to gathering information	i don't have enough information	Disappointed
PS-3	User	Finding The Solution to the disease	it's hard	it takes long time	Disappointed

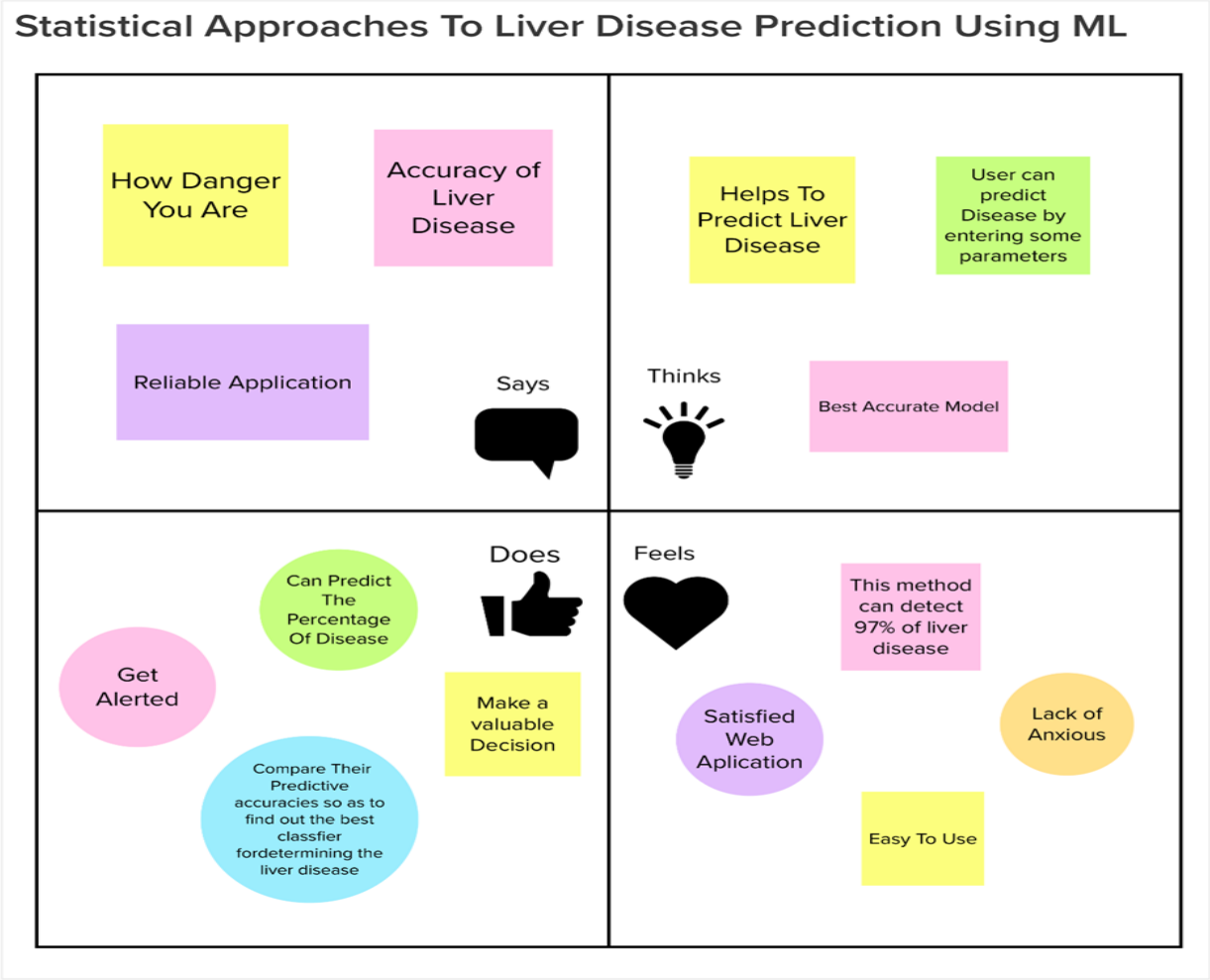
CHAPTER- 3

IDEATION & PROPOSED SOLUTION

We have analyzed different systems and proposed an ideation phase of our web application.

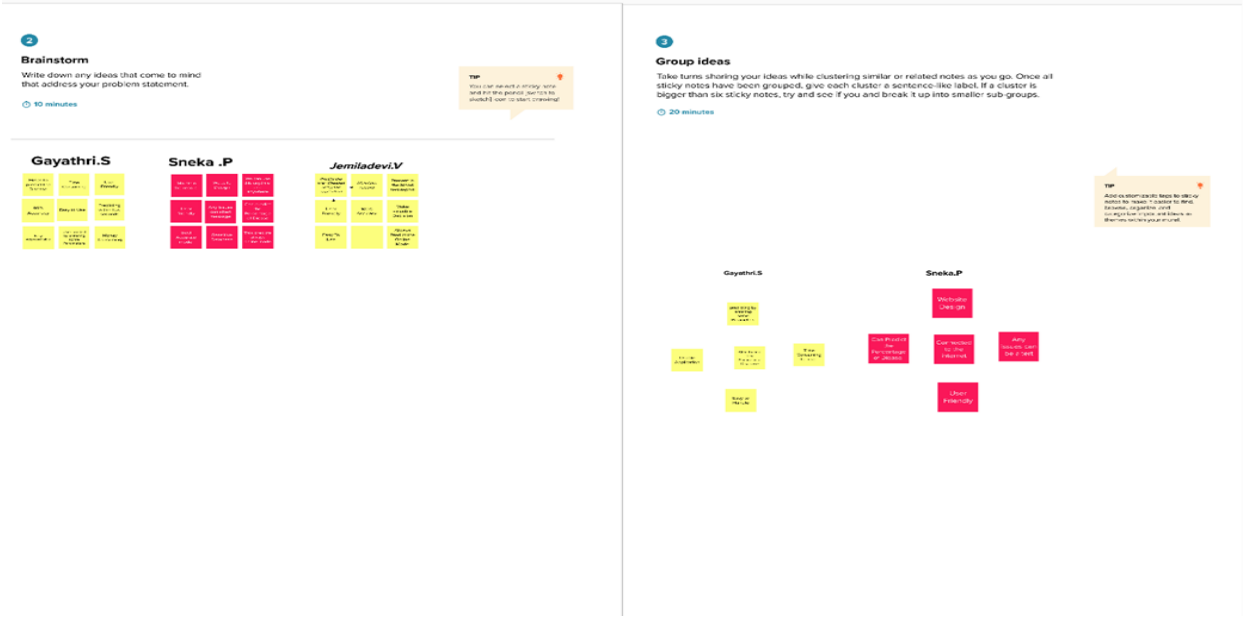
3.1. EMPATHY MAP CANVAS

An empathy map canvas helps brands provide a better experience for users by helping teams understand the perspectives and mindset of their users. Using a template to create an empathy map canvas reduces the preparation time and standardizes the process so you create empathy map canvases of similar quality.



3.2. IDEATION & BRAINSTORMING

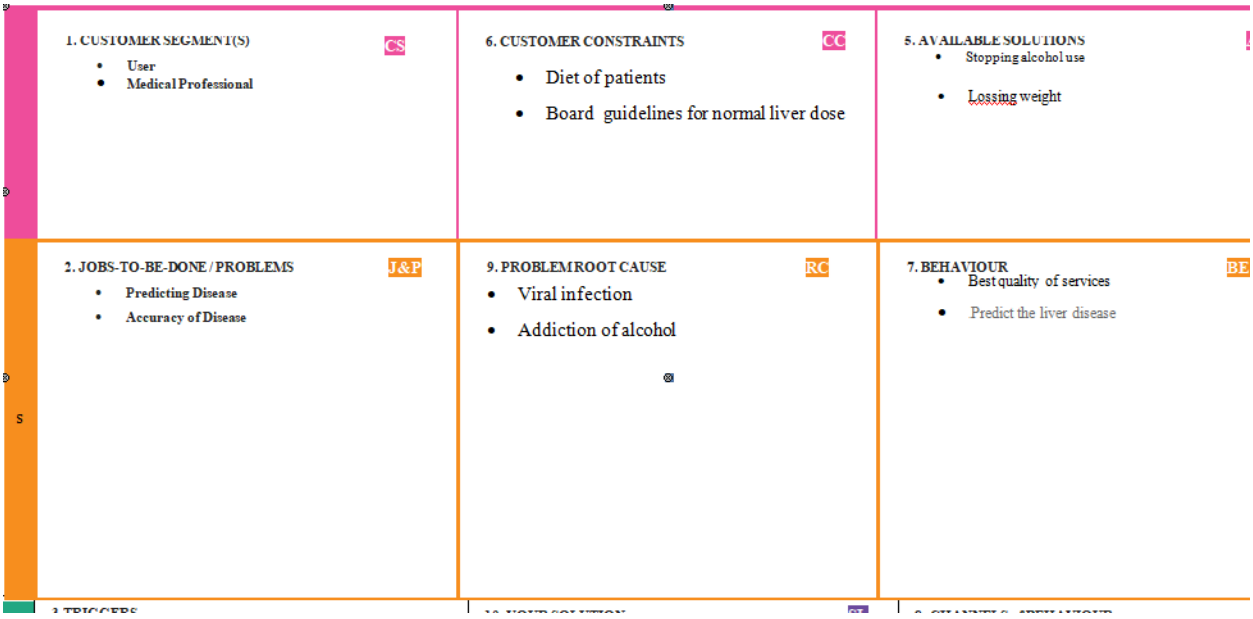
Noting down any ideas that come to mind that address your problem statement.



3.3. PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Predicting The Liver Disease.
2.	Idea / Solution description	By Entering some parameter User can predict the Accuracy of Disease.
3.	Novelty / Uniqueness	Determine patients having liver disease and which one do not in an accurate and faster way.
4.	Social Impact / Customer Satisfaction	Early Disease of Prediction will help in diagnosis and related treatment of the patients.
5.	Business Model (Revenue Model)	Can generate revenue through Direct customers.
6.	Scalability of the Solution	The Application used by any patients, users and medical professionals by ensuring their disease.

3.4. PROBLEM SOLUTION FIT



CHAPTER-4

REQUIREMENT ANALYSIS

4.1. FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Calender	In this application shall allow the user to add the data to their expenses
FR-4	Planner	This application should have represent in liver predict almost graphically in to the different level.
FR-5	Tracker	Liver diease preditct contains in graphically we have any of time form of report.
FR-6	Category	If any of the can be in this application for adding in to the category.

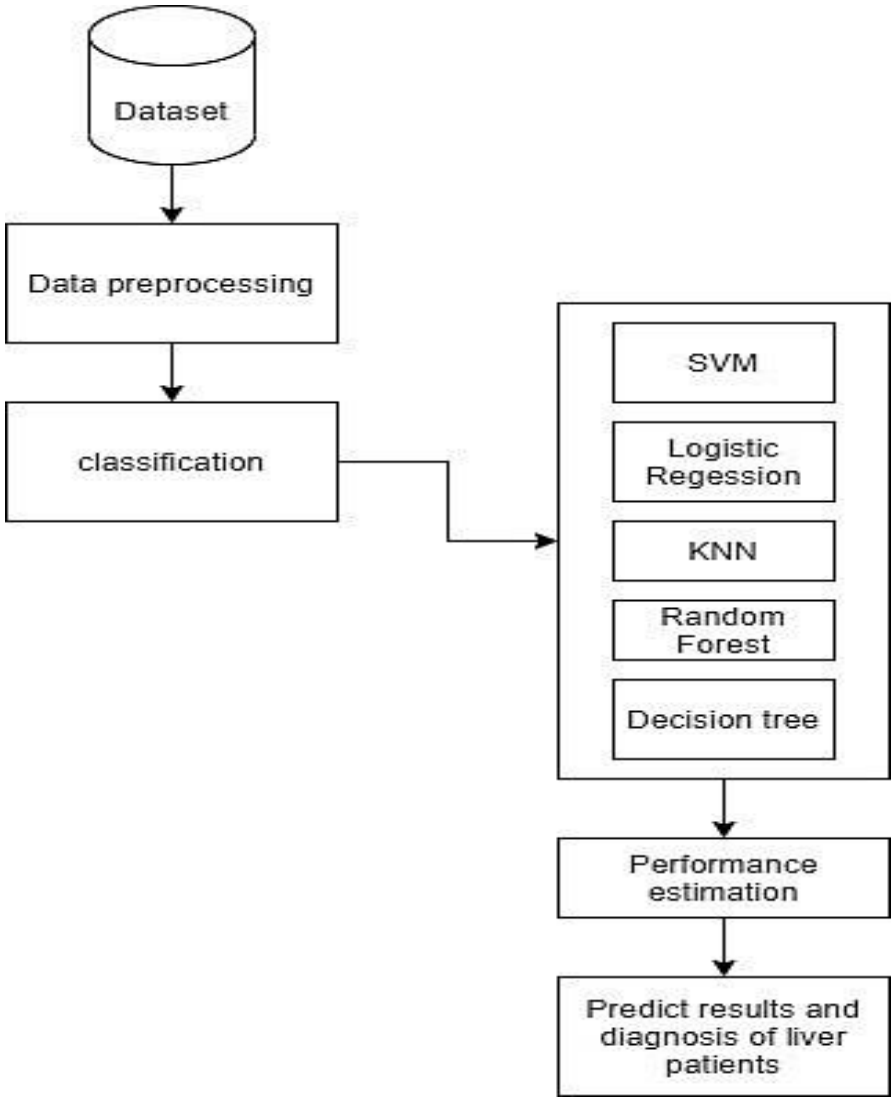
4.2.NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	It have any access we will go to previous stage in the liver diease prediction.
NFR-2	Security	More security of the patients data and user personal details.
NFR-3	Reliability	Each patients data records is stored on a well built efficient database schema and no risk in data loss.
NFR-4	Performance	It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.
NFR-5	Availability	It is always available of the time,no time constraint.
NFR-6	Scalability	The ability to appropriately handle increasing demands.

CHAPTER -5

PROJECT DESIGN

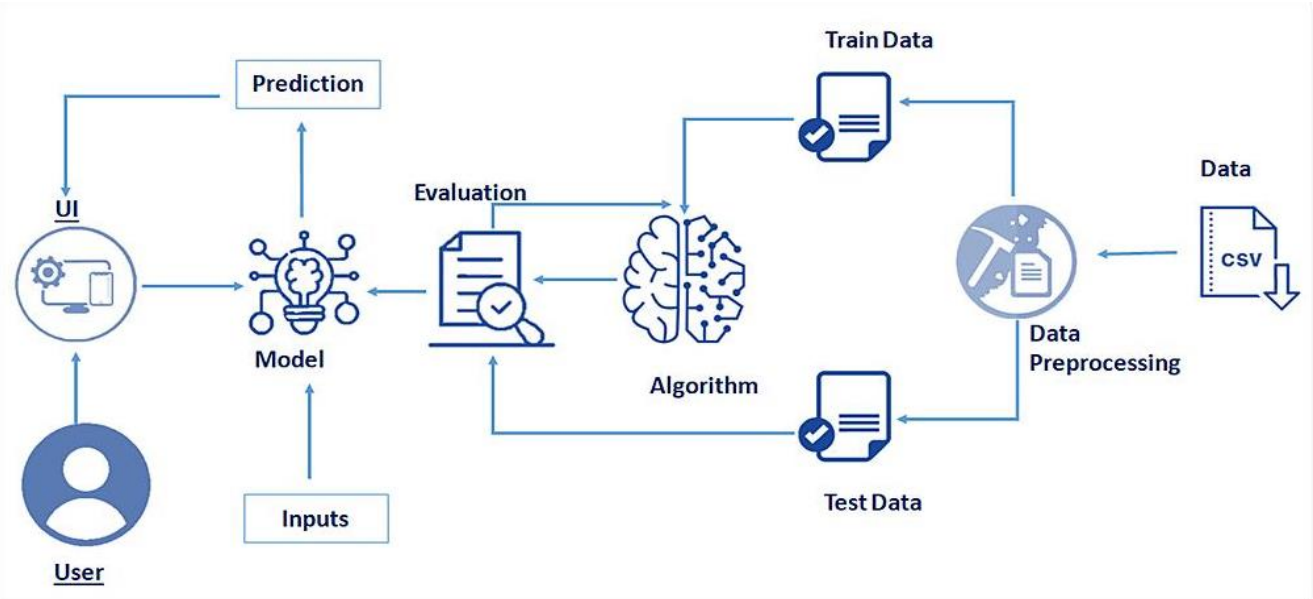
5.1. DATA FLOW DIAGRAMS



5.2. SOLUTION & TECHNICAL ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.



5.3. USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password	I can access my account	High	Sprint-2
	Dashboard	USN-3	As a user, I can log into my account for the mobile	I can access my account /Dashboard	High	Sprint-2
Customer (Web user)	Registration	USN-4	As a user, I can register for the application by entering my email, password, and confirming my password	I can access my account/Dashboard	High	Sprint-1
Customer Care Executive	Customer Support	USN-5	As a user, I can support for customers to handle queries and complaints from their customers	I can support for customers to clear complaints	High	Sprint-3
Administrator	Responsibility	USN-6	As a system administrator I want to be able to add new users when required so that	I Can add new users	High	Sprint-4

CHAPTER-6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint 1

1. We created a Flask Project.
2. Added all the routes needed for our project.
3. Created Tables in IBM Cloud.

Sprint 2

1. We added all the html templates needed for our project.
2. We styled those pages using CSS and Bootstrap.
3. We wrote Queries to connect IBM Cloud Database.
4. Finished all the Fetching and Posting Stuff of IBM Cloud Database Integration.

Sprint 3

1. Integration of Send grid into our application

Sprint 4

1. Finished in all release of the results.

6.2 SPRINT DELIVERY SCHEDULE

Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirment(Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering email, password and confirming my password	2	High	S. GAYATHRI V. JEMILADEVI N. KARTHIKA P. SNEKA
Sprint-2	Login	USN-2	As a user, I can log into the application by entering email and password	1	High	S. GAYATHRI V. JEMILADEVI N. KARTHIKA P. SNEKA
Sprint-2	Dashboard	USN-3	As a user, I can log into my account for the mobile	1	High	S. GAYATHRI V. JEMILADEVI N. KARTHIKA P. SNEKA
Sprint-3	Customer support	USN-4	As a user, I can support for customers to handle queries and complaints from their customers	1	High	S. GAYATHRI V. JEMILADEVI N. KARTHIKA P. SNEKA

Sprint-4	Responsibility	USN-5	As a system administrator I want to be able to add new users when required so that.	1	High	S. GAYATHRI V. JEMILADEVI N. KARTHIKA P. SNEKA

Project Tracker, Velocity & Burndown Chart

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed(as on planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07Nov 2022	12 NOV 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14Nov 2022	19 NOV 2022	20	19 Nov 2022

Velocity

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Burndown Chart

A burn down chart is a graphical representation of work left to do versus time.

It is often used in agile software development methodologies such as Scrum.

However, burn down charts can be applied to any project containing measurable

progress over time.

https://www.visual-paradigm.com/scrum/scrum-burndown-chart/

https://www.atlassian.com/agile/tutorials/burndown-charts

CHAPTER-7

CODING & SOLUTIONING

7.1 FEATURE 1

- ➤ One of the main segments in chronic liver disease prediction is the selection of important features of liver disorder.
- ➤ In this step, several features such as age, gender hat represent the personal information of each patient is v One of the main segments in chronic liver disease prediction is the selection of important features of liver disorder.
- ➤ The purpose of this research was to provide medical diagnosis information

7.2 DATABASE SCHEMA

Flask provides utilities for testing an application. This documentation goes over techniques for working with different parts of the application in tests.

We will use the pytest framework to set up and run our tests.

```
$ pip install pytest
```

Most web applications have a database. When running tests, you want to be certain that the tests don't hit the production database. At the same time, you want something like a database to be there.

I assume that you are using flask-sqlalchemy . It is part of the pallets project and thus an official part of the Flask ecosystem.

what about testing the SQL Queries?

You might wonder now how to test the SQL queries. Testing that they work at all should not be necessary if you use SQL Alchemy. And I really recommend to use SQL Alchemy when you use Flask with a relational database. If your queries are too complex for that, you can have a look at Query Builders. Avoid using raw SQL. In most cases it should not be necessary.

CHAPTER-8

TESTING

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Some prefer saying Software testing definition as a White Box and Black Box Testing. In simple terms, Software Testing means the Verification of Application Under Test (AUT). This Software Testing course introduces testing software to the audience and justifies the importance of software testing.

8.1 TEST CASES

We perform a testing analysis to validate its prediction capacity. we use subset of data that has not been used before, the testing instances.

The next table shows the confusion matrix for our problem. the confusion matrixs represents the real classes and the predicted classes 'columns for the testing data

	Predicted no_disease	**Predicted suspect_disease**	**Predicted hepatitis_c**	**Predicted fibrosis**	**Predicted cirrhosis**
Real no_disease	*107(87%)*	1(0.813%)	0	1(0.813%)	0
Real supset_diseases	0	0	0	0	0

Real hepatitis_c	5(4.07%)	0	1(0.813%)	1(0.813%)	1(0.813%)
Real fibrosis	0	0	1(0.813%)	2(1.63%)	0
Real cirrhosis	1(0.813%)	0	0	0	2(1.63%)

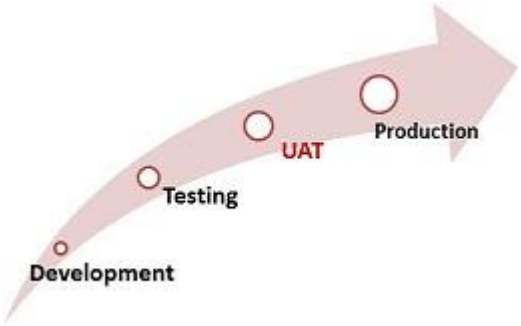
The number of instance the number of instances that the model can correctly predict is 123 (92%), while it misclassifies only 11 (8%) approximately. This shows that our predictive model has an excellent classification accuracy, and the biggest confusion is predicting no disease when the patient is suffering from hepatitis c.

8.2 USER ACCEPTANCE TESTING

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

Need of User Acceptance Testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

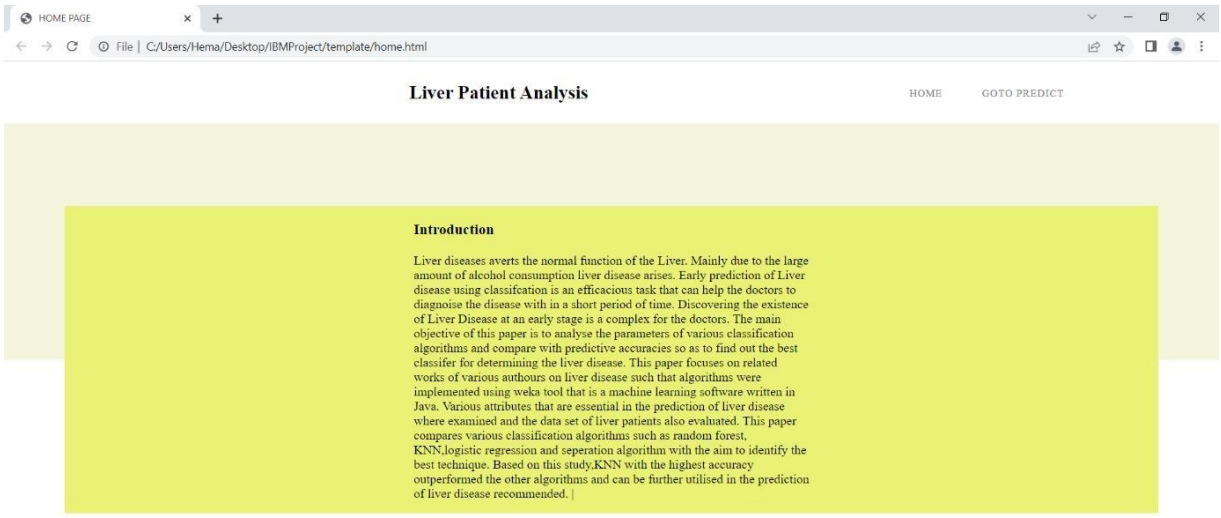
Purpose of UAT

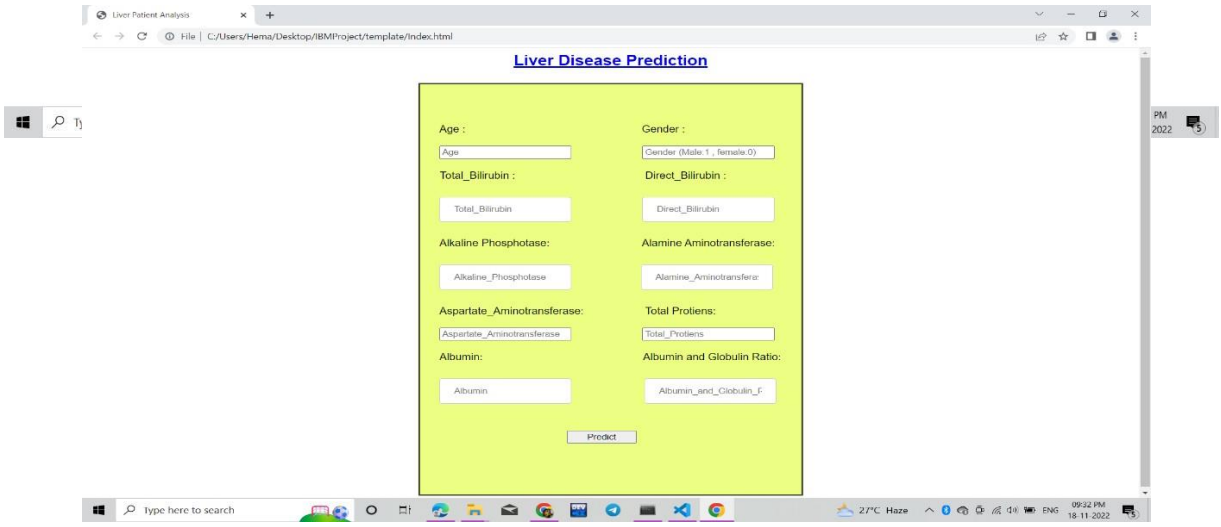


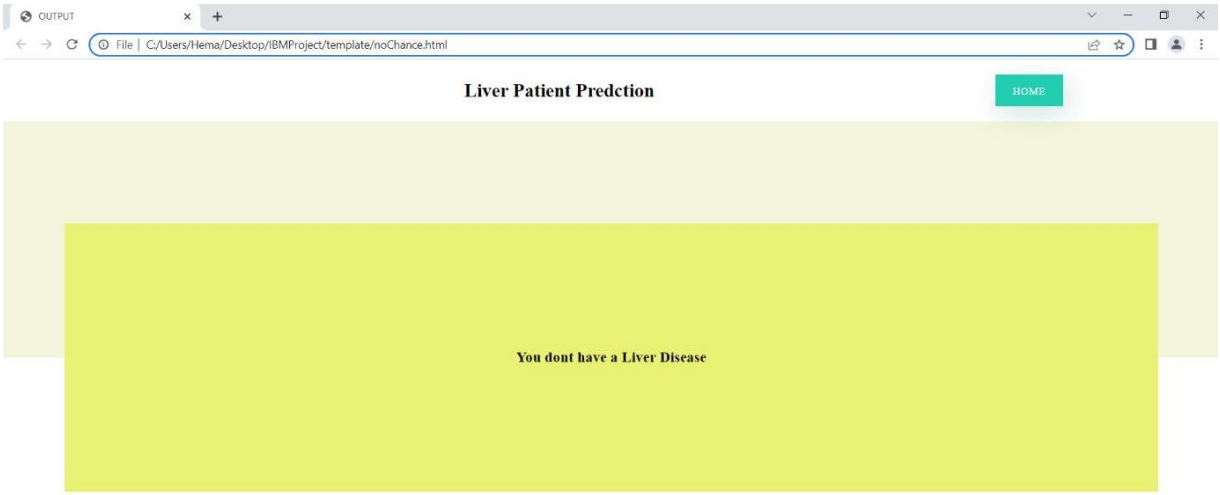
The main **Purpose of UAT** is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

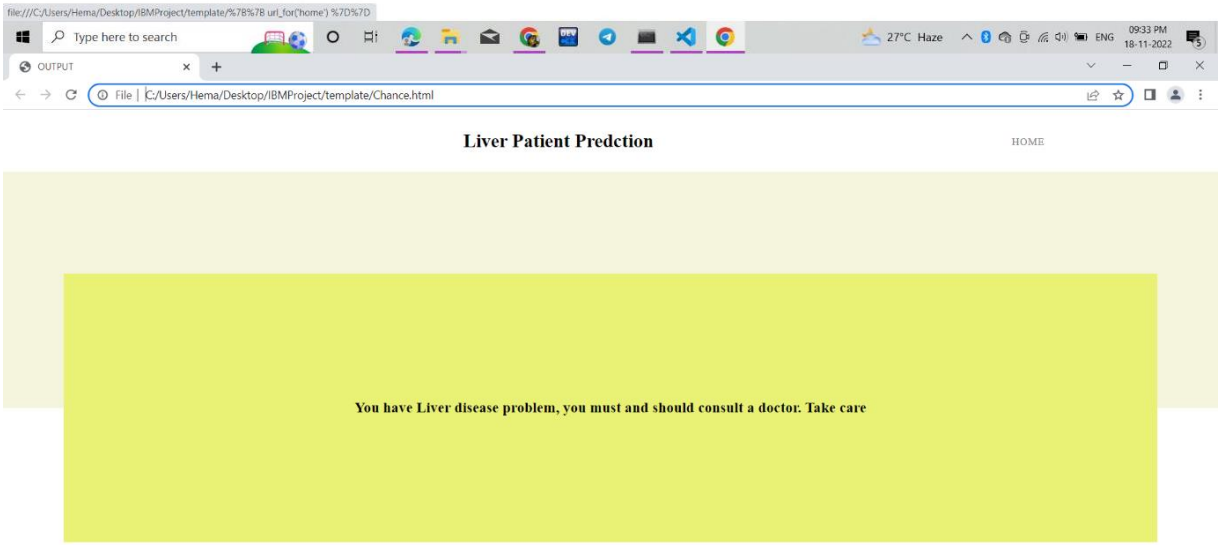
CHAPTER-9

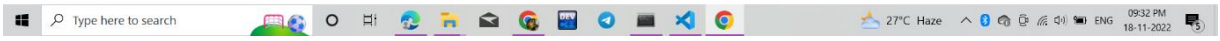
RESULTS






OUTPUT
File | C:/Users/Hema/Desktop/IBMProject/template/noChance.html
Liver Patient Predction
HOME
You dont have a Liver Disease


file:///C:/Users/Hema/Desktop/IBMProject/template/%7B%7B url_for('home') %7D%7D
Type here to search
27°C Haze
ENG
09:33 PM
18-11-2022
OUTPUT
File | C:/Users/Hema/Desktop/IBMProject/template/Chance.html
Liver Patient Predction
HOME
You have Liver disease problem, you must and should consult a doctor. Take care


Type here to search
27°C Haze
ENG
09:32 PM
18-11-2022

9.1. PERFORMANCE METRICS

True positives imply the positive liver tuples that were precisely named by the classifier,

True negatives are the negative liver tuples that were precisely set apart by the classifier.

False positives are the negative liver tuples that were erroneously set apart as positive tuples

False negatives are the positive liver tuples that were incorrectly stamped negative tuples

CHAPTER-10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

1.No medical expertise required: You don't need to have any knowledge of medical science and liver diseases to predict the liver disease using this application. All you need to do is enter the details being asked, which are already present in the blood test report(some like age, gender are already known) and then you will get the results of prediction.

2. High accuracy: The system predicts the results with100 % accuracy for the dataset that we have used while creating this application. While the accuracy might be different in some cases, it will still be high enough to be trustworthy at a large scale.

DISADVANTAGES

There is no real time identity verification.

Not using data mining technique to detect chances of getting liver disease.

Time instense.

Some approaches are not adoptable for real time collection of database implementation.

Certain approaches being applicable only for the small data.

CHAPTER-11

CONCLUSION

Diseases related to liver and heart are becoming more and more common with time. With continuous technological advancements, these are only going to increase in the future. Although people are becoming more conscious of health nowadays and are joining yoga classes, dance classes; still these dentary lifestyle and luxuries that are continuously being introduced and enhanced; the problem is going to last long. So, in such a scenario, our project will be extremely helpful to the society. With the dataset that we used for this project, we got 100 % accuracy for SVM model, and though it might be difficult to get such accuracies with very large datasets, from this project's results, one can clearly conclude that we can predict the risk of liver diseases with accuracy of 90 % or more.

Today almost everybody above the age of 12 years has smart phones with them, and so we can incorporate these solutions into an android app or ios app. Also it can be incorporated into a website and these app and website will be highly beneficial for a large section of society.

CHAPTER-12

FUTURE SCOPE

Database should be expanded on which the system will be tested much better.

Also, the model requires further improvement mostly regarding feature selection of the liver into multiple components.

The application must provide user interface for doctors input object of the prescription.

The application should have the capability for preprocessing of the given input.

The system should be capable to detect the chances of liver diease using past patient data.

Thus, combining the blood test features along with the features already used in the study can provide new directions.

Feature selection is the process of finding input features for a predictive model which removing irrelevant features that don’t contribute towards the model.

CHAPTER-13

APPENDIX

SOURCE CODE

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Liver Prediction Model</title>
</head>
<body>
   <div class="container">
      <h2 class='container-heading'><span class="heading_font">Liver Disease
Prediction</span></h2>
   </div>
 <div class="ml-container">
      <form action="{{ url_for('predict') }}" method="POST">
  <br>
<h3>Age</h3>
         <input id="first" name="Age" placeholder="in Year" required="required">
         <br>
 <h3>Gender</h3>
 <input id="second" name="Gender" placeholder="Male = 1, Female=0"
required="required">
         <br>
 <h3>Total Bilirubin</h3>
         <input id="third" name="Total_Bilirubin" placeholder="Total Bilirubin"
required="required">
        <br><br>
 <h3>Direct Bilirubin</h3>
         <input id="third" name="Direct_Bilirubin" placeholder="Direct Bilirubin"
required="required">
         <br>
         <h3>Alkaline Phosphotase</h3>
         <input id="fourth" name="Alkaline_Phosphotase" placeholder="Alkaline
Phosphotase" required="required">
         <br>
```

```
        <h3>Alamine Aminotransferase</h3>
        <input id="fifth" name="Alamine_Aminotransferase"
placeholder="Alamine Aminotransferase" required="required">
        <br>
        <h3>Aspartate Aminotransferase</h3>
        <input id="sixth" name="Aspartate_Aminotransferase"
placeholder="Aspartate Aminotransferase" required="required">
        <br>
        <h3>Total Protiens</h3>
        <input id="seventh" name="Total_Protiens" placeholder="Total Protiens"
required="required">
        <br>
        <h3>Albumin</h3>
        <input id="eight" name="Albumin" placeholder="Albumin"
required="required">
        <br>
        <h3>Albumin and Globulin Ratio</h3>
        <input id="ninth" name="Albumin_and_Globulin_Ratio"
placeholder="Albumin and Globulin Ratio" required="required">
        <br>
        <br>
        <br>
        <button id="sub" type="submit ">Submit</button>
        <br>
        <br>
        <br>
        <br>
        <p class='footer-description'>@JACSICE 2022</p>
 </form>
   </div>
```

Result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Liver Disease Result</title>
</head>
<body>
<div class="container">
```

```
      <form action="{{ url_for('predict')}}" method="post">
        <h2 class='container-heading'><span class="heading_font">Liver Disease
Prediction</span></h2>
  <br><br><br><br><br><br><br>
<!-- Result -->
        <div class="results">
              {% if prediction==2 %}
                    <h1><span class='danger'>Oops! ??<br><br>You have LIVER
DISEASE <br><br>Please Consult a Doctor.</span></h1>
                    <img class="gif" src="{{ url_for('static', filename='dr.gif')}}"
alt="LIVER Image">
              {% elif prediction==1 %}
                    <h1><span class='safe'>?? Congratulation! ??<br><br>You
DON'T have LIVER DISEASE.</span></h1>
                    <img class="gif1" src="{{ url_for('static',
filename='yes.webp')}}" alt="Not LIVER Image">
              {% endif %}
        </div>
      </form>
        </div>
   <div>
         <br><br> <br><br><br><br><br><br><br><br>
        <p class='footer-description'>@JACSICE 2022</p>
</div>
<style>
/* Background Image */
body
{
background-color:pink;
height: 100%;
/* Center and scale the image nicely */
background-position: center;
background-repeat: no-repeat;
background-size: 100% 100%;
}
/* Color */
body{
font-family: Arial, Helvetica,sans-serif;
   text-align: center;
   margin: 0;
```

```
    padding: 0;
    width: 100%;
height: 100%;
display: flex;
flex-direction: column;
}
/* Heading Font */
.container-heading{
    margin: 0;
}

.heading_font{
    color: #black;
font-family: 'Pacifico', cursive;
font-size: 50px;
font-weight: normal;
}</style>
</body>
</html>
```

Liver.py

```
# Importing Libraries:
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# for displaying all feature from dataset:
pd.pandas.set_option('display.max_columns', None)

# Reading Dataset:
dataset = pd.read_csv("Dataset/Liver_data.csv")
# Top 5 records:
dataset.head()

# RandomForestClassifier:
from sklearn.ensemble import RandomForestClassifier
RandomForest = RandomForestClassifier()
RandomForest = RandomForest.fit(X_train,y_train)

# Predictions:
```

```
y_pred = RandomForest.predict(X_test)

# Performance:
print('Accuracy:', accuracy_score(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

 # Creating a pickle file for the classifier
import pickle
filename =  'Liver.pkl'
pickle.dump(RandomForestClassifier, open(filename, 'wb'))
```

app.py

```
from flaskimport Flask, render_template, request
import numpy as np
import pickle
app = Flask(__name__)
model = pickle.load(open('Liver2.pkl', 'rb'))

@app.route('/',methods=['GET'])
def Home():
    return render_template('index.html')

@app.route("/predict", methods=['POST'])
def predict():
    if request.method == 'POST':
    Age = int(request.form['Age'])
    Gender = int(request.form['Gender'])
    Total_ Bilirubin = float(request.form['Total _Bilirubin']
Alkaline _Phosphotase = int(request.form['Alkaline_Phosphotase'])
Alamine_ Aminotransferase = int(request .form['Alamine _Aminotransferase'])
Aspartate_Aminotransferase=int(request.form['Aspartate_Aminotransferase'])
Total_ Protiens = float(request. form['Total_ Protiens'])
Albumin = float(request .form['Albumin'])
Albumin_and_Globulin_Ratio=float(request.form['Albumin_and_Globulin_Ratio']
)
values=np.array([[Age,Gender,Total_Bilirubin,Alkaline_Phosphotase,Alamine_A
minotransferase,Aspartate_Aminotransferase,Total_Protiens,Albumin,Albumin_an
d_Globulin_Ratio]])
 prediction = model. predict(values)
```

```
return render _template('result.html', prediction=prediction)
if__name__== "__main__":
   app .run(debug=True)
```

GitHub Link : https://github.com/IBM-EPBL/IBM-Project-37647-1660315483

Project Demo Link
https://drive.google.com/drive/folders/1IbkxlO3mHwh6sXk4z0ocUAWMaiBj8GQm