TEAM ID : **PNT2022TMID15752**

# CHAPTER 1

# INTRODUCTION

## 1.1 PROJECT OVERVIEW

Liver diseases avert the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of liver disease at an early stage is a complex task for the doctors. The main objective of this project is to analyze the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This Project examines data from liver patients concentrating on relationships between a key list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease. Here we are building a model by applying various machine learning algorithms find the best accurate model. And integrate to flask based web application. User can predict the disease by entering parameters in the webapplication.

## 1.2 PURPOSE

Current screening strategies for liver disease focus on detection of subclinical advanced liver fibrosis but cannot identify those at high future risk of severe liver disease. Our aim was to develop and validate a risk prediction model for incident liver disease in the general population based on widely available factors. The diagnosis is often delayed until severe complications occur and prognosis becomes poor. In order to identify individuals in the general population who have a high risk of developing severe liver disease in the future, we developed and validated a Liver Disease risk prediction with or without measurement of the liver enzyme gamma-glutamyl transferase. The Liver Disease score can be used as part of health counseling, and for planning further liver investigations and follow-up.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 EXISTING PROBLEMS

| S.No. | YEAR | AUTHOR NAME | TITLE | ALGORITHM | DRAWBACKS |
|---|---|---|---|---|---|
| 1 | 2009 | P.Karule, Dr. Sanjay Vasant Dudul | Intelligent Diagnosis of Liver Diseases from Ultrasonic Liver Images: Neural Network Approach | MLPNN classifier | In the early stages, HV tends to be asymptomatic and can be detected only through screening. |
| 2 | 2010 | G.Sakr, I.Elhajj, H. Huijer | Support Vector Machines to Define and Detect Agitation Translation | Support Vector Machine Classifier | Unlike generative machine learning approaches, which require computations of conditional probability distributions |
| 3 | 2012 | H. Al-Angari, A. Sahakian | Automated Recognition of Obstructive Sleep Apnea Syndrome Using Support Vector Machine Classifier. | Support Vector Machine (SVM) Classifier | The features can be incorporated into automatic algorithms for portable OSA monitoring using the available respiratory and oxygen saturation devices. |
| 4 | 2021 | Adekola Olubukola Daniel, Ekanem Edikan Uwem | Prediction Diagnosis of Liver Disease in Human of using Machine Learning | Neural network | The need to develop a tool that could aid doctors and prevent from unwarranted errors and unwanted biases in diagnosis is established in this research. |
| 5 | 2021 | Sana Ansari, I. Shafi, J.Ahmad, Syed Ismail Shah | Neural network-based approach for the non-invasive diagnosis and classification of hepatotropic viral disease, December2021 | Neural network | The respiratory and magnitude features showed sensitivity in the apnea minute classification compared to the other features. |

**Table 2.1 - Existing Problems**

## 2.2 REFERENCES


1. P.Karule, Dr. Sanjay Vasant Dudul, "Intelligent Diagnosis of Liver Diseases from Ultrasonic Liver Images: Neural Network Approach", July 2009.
2. G.Sakr, I.Elhajj, H. Huijer, "Support vector machine to define and detect Agitation translation", February 2010.
3. H. Al-Angari, A. Sahakian, "Automated Recognition of Obstructive Sleep Apnea Syndrome Using Support Vector Machine Classifier", May 2012.
4. Adekola Olubukola Daniel, Ekanem Edikan Uwem, "Prediction and Diagnosis of Liver Disease in Human Using Machine Learning", August 2021.
5. Sana Ansari, I. Shafi, J.Ahmad, Syed Ismail Shah, "Neural network-based approach for the non-invasive diagnosis and classification of hepatotropic viral disease", December2021.


## 2.3 PROBLEM STATEMENT DEFINITION

Liver diseases avert the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of liverdisease at an early stage is a complex task for the doctors. The main objective of this project is to analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This Project examines data from liver patients concentrating on relationships between a key list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease. Here we are building a model by applying various machine learning algorithms find the best accurate model and integrate to flask based web application.

| | |
|---|---|
| I am | The person to predict the liver disease using Machine Learning techniques |
| I'm trying to | Use the recent technologies to predict the human liver disease. |
| But | I am unaware of the existing technology that can help me a lot to predict the disease and I don’t know to use the correct technology. |
| Because | I don’t want to waste the cost and time. |
| Which makes me feel | I want a best accuracy which can predict the disease so that the people can move with their necessary treatments. |

**Table 2.2 - Problem Statement Definition**

# CHAPTER 3

# IDEATION AND PROPOSED SOLUTION

## 3.1 EMPATHY MAP CANVAS

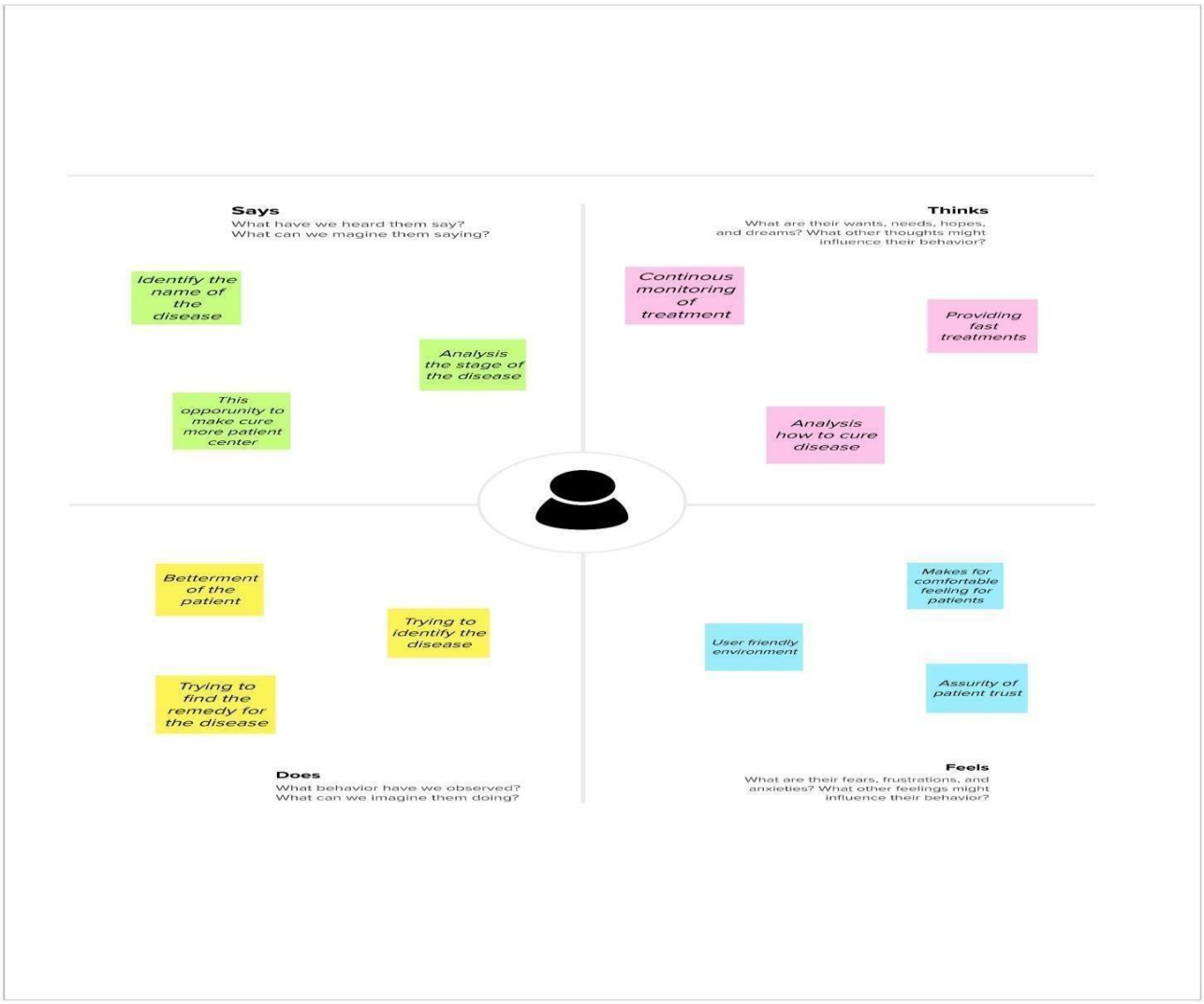


**Figure 3.1 - Empathy Map**

An empathy map is a widely-used visualization tool within the field. In relation to **empathetic** design, the primary purpose of an empathy map is to bridge the understanding of the end user.

## 3.2 IDEA ION

## AND

## BRAINS ORMIN

**GSte p- 1:**

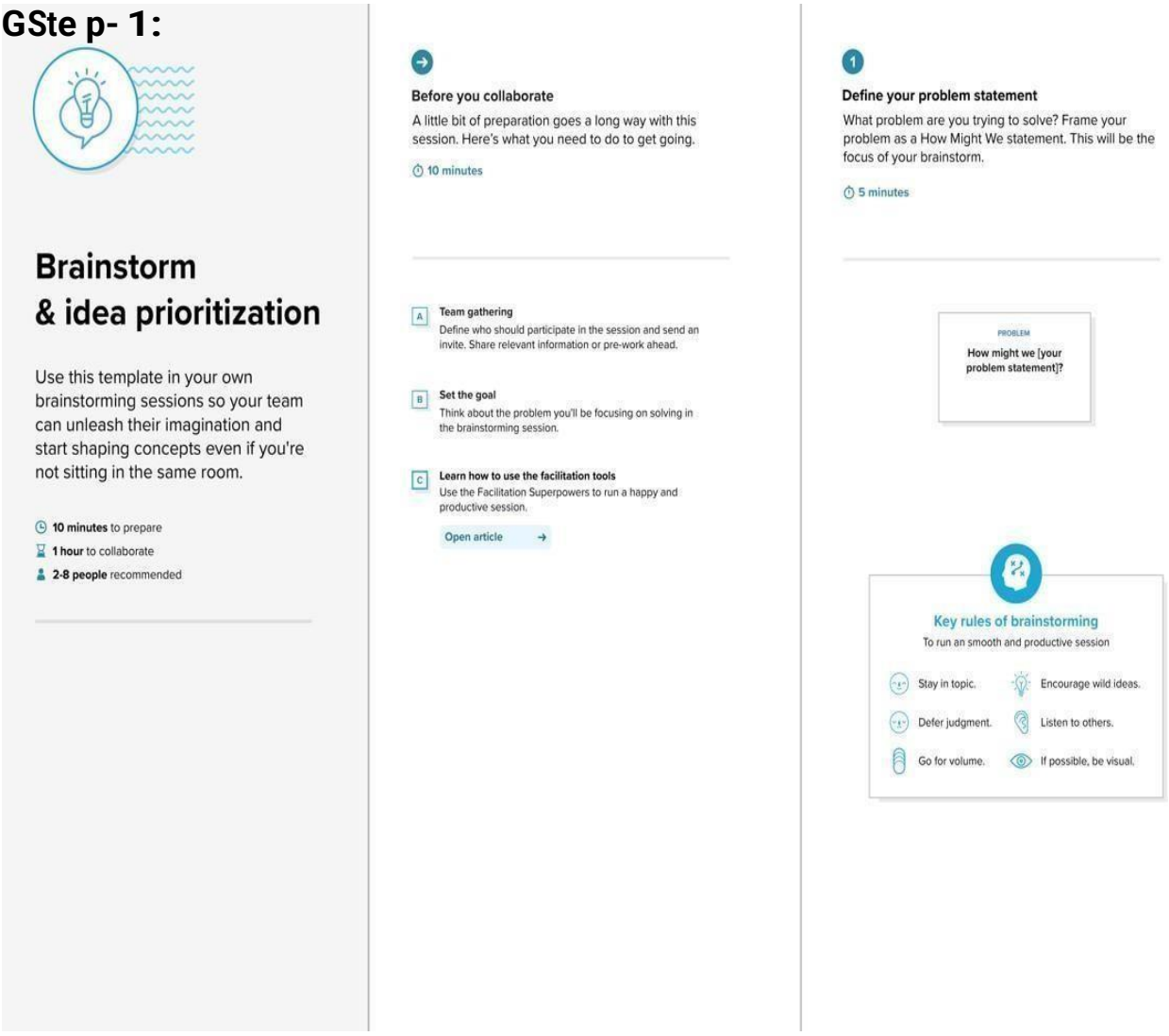


**Figure 3.2 - Ideation and Brainstorming**

A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

## Step 2:

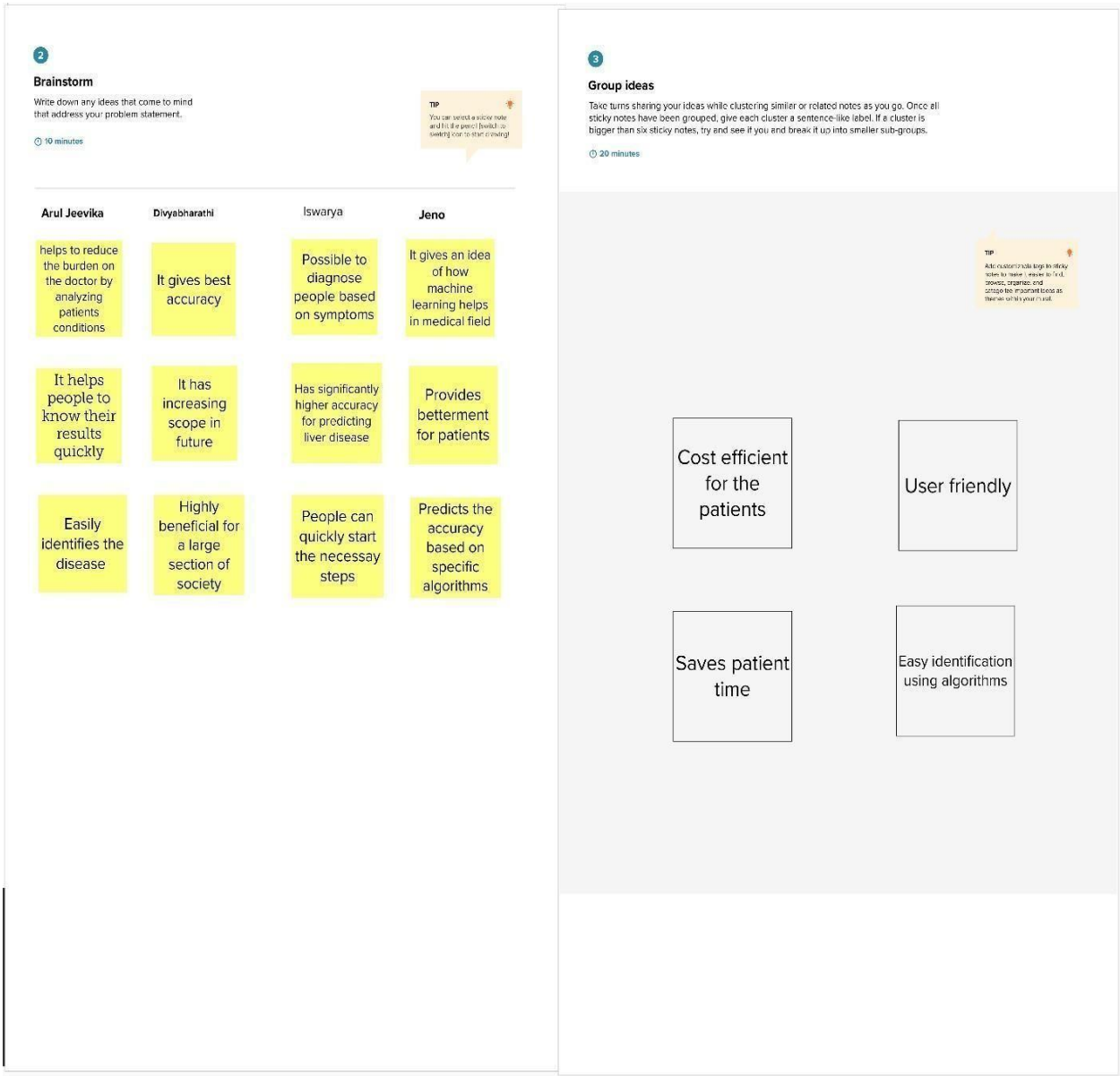


**Figure 3.3 - Brainstorm, Idea Listing and Grouping**

The idea listing and grouping is used to organize and analyse large numbers of ideas by categorising them. By organising and reorganising ideas, students gain a better appreciation of, and dialogue about, their ideas. As students create idea clusters, new contexts and connections among themes emerge.

## Step 3:

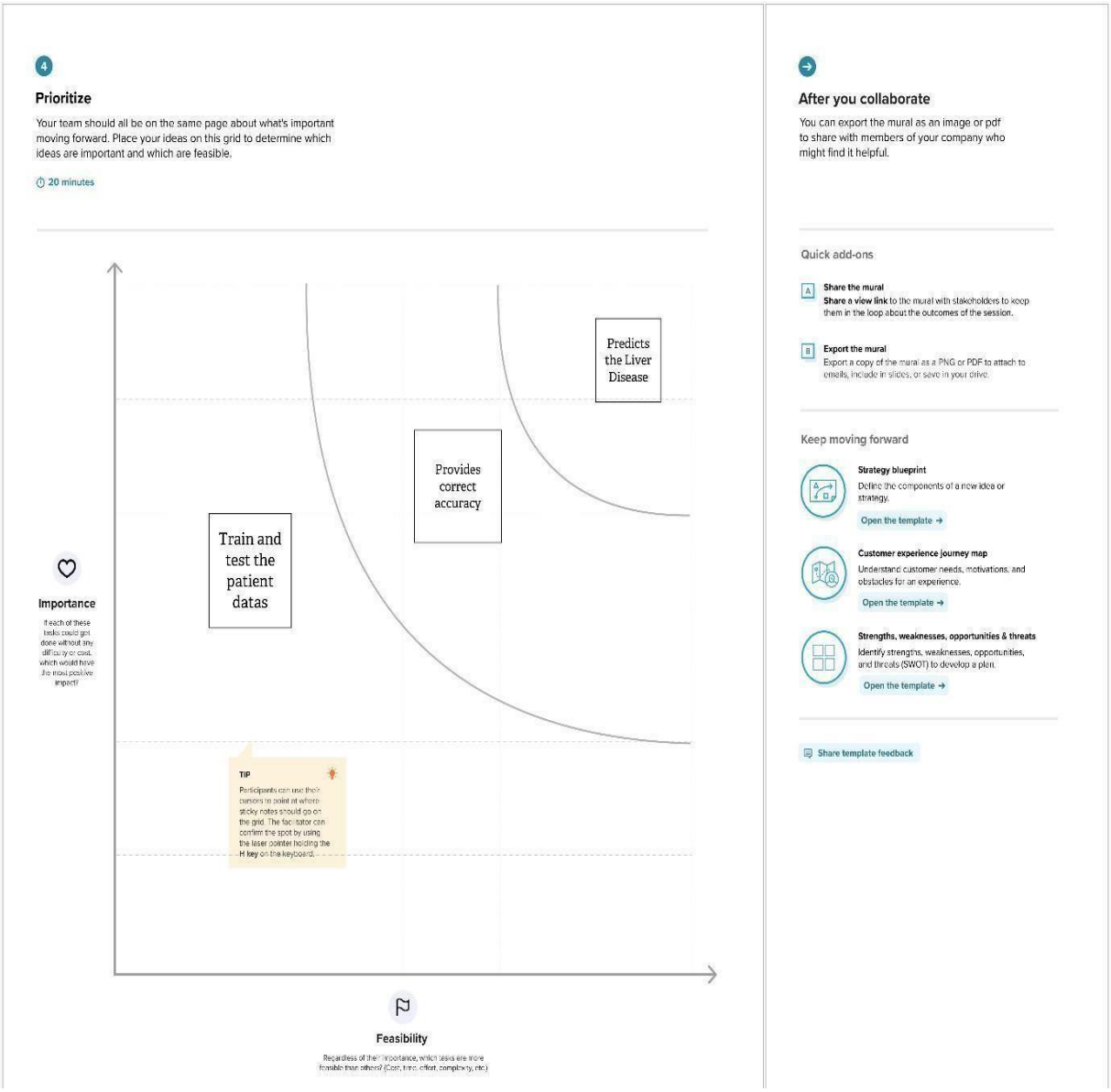


**Figure 3.4 - Idea Prioritization**

Idea prioritization is just a part of the idea management process. Having a structured idea management process and a systematic way of gathering, evaluating and prioritizing new ideas takes time. To make it work, the entire idea management process should be integrated to the everyday ways of working.

## 3.3 PROPOSED SOLUTION

| S.No. | PARAMETERS | DESCRIPTION |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | Liver diseases avert the normal function of the liver. Early prediction of liver disease using classification algorithms are an effective task that can help the doctors to diagnose the disease within a short duration |
| 2. | Idea/Solution description | One of the easiest solutions to predict the liver disease using Machine Learning techniques |
| 3. | Novelty / Uniqueness | This project provides the best accuracy for predicting the liver disease |
| 4. | Social Impact / Customer Satisfaction | It helps to identify the liver disease in effective way, reduce the cost and user friendly |
| 5. | Scalability of the Solution | This project can be improved by giving medical suggestion for patients |

**Table 3.1 - Proposed Solution**

## 3.4 PROBLEM SOLUTION FIT

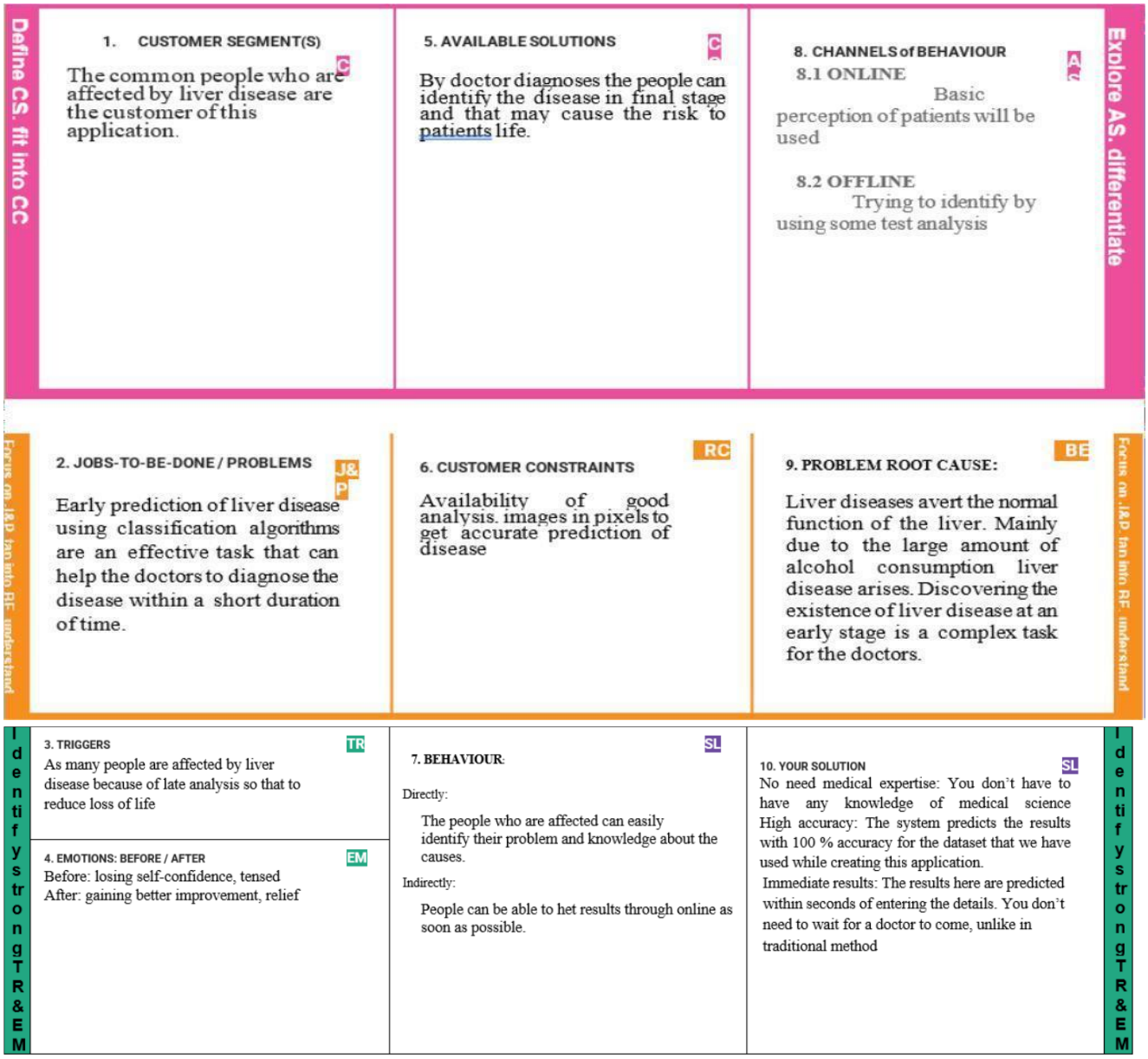


**Figure 3.5 - Solution fit of design with user requirements**

This occurs when the user have evidence that customers care about certain jobs, pains, and gains. At this stage the user proved the existence of a problem and have designed a value proposition that addresses customers' jobs, pains and gains

# CHAPTER 4

# REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENTS

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | Upload image | The user can upload the medical test image. The image should be in jpeg, png or jpg format |
| FR-2 | Read image | The image will be scanned before augmentation takes place. |
| FR-3 | Transform image | The scanned image is then transformed into a format that is needed by the saved model. |
| FR-4 | Evaluate image | The saved model creates a feature map of the uploaded medical test image and predicts the image. |
| FR-5 | Determine and predict the output | The predicted output is then analyzed and converted to a user-friendly language. |
| FR-6 | Display the output | The analyzed result is then diaplayed to the user. |

**Table 4.1 - Functional Requirements**

## 4.2 NON-FUNCTIONAL REQUIREMENTS

| FR No. | Non-Functional Requirements | Description |
|---|---|---|
| NFR-1 | Usability | Datasets of all the liver is used to detecting the disease that present in the liver. |
| NFR-2 | Security | The information belongs to the user and liver are secured highly. |
| NFR-3 | Reliability | It is important for predicting the disease in liver. |
| NFR-4 | Performance | The performance is based on the technology used for disease prediction |
| NFR-5 | Availability | It is available for all user to predict the disease in the liver. |
| NFR-6 | Scalability | Increasing the prediction of the disease in the liver. |

**Table 4.2 - Non Functional Requirements**

# CHAPTER 5

# PROJECT DESIGN

## 5.1 DATAFLOW DIAGRAMS

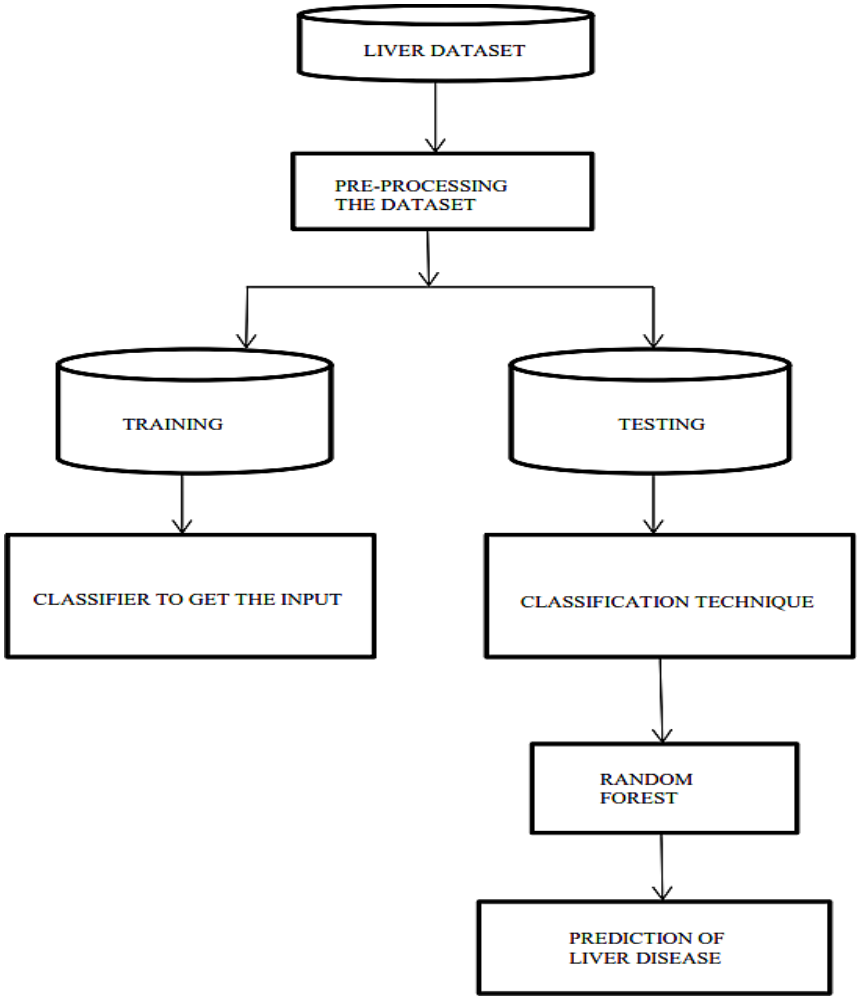


**Figure 5.1 - Data flow Diagram**

A data flow diagram is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method.

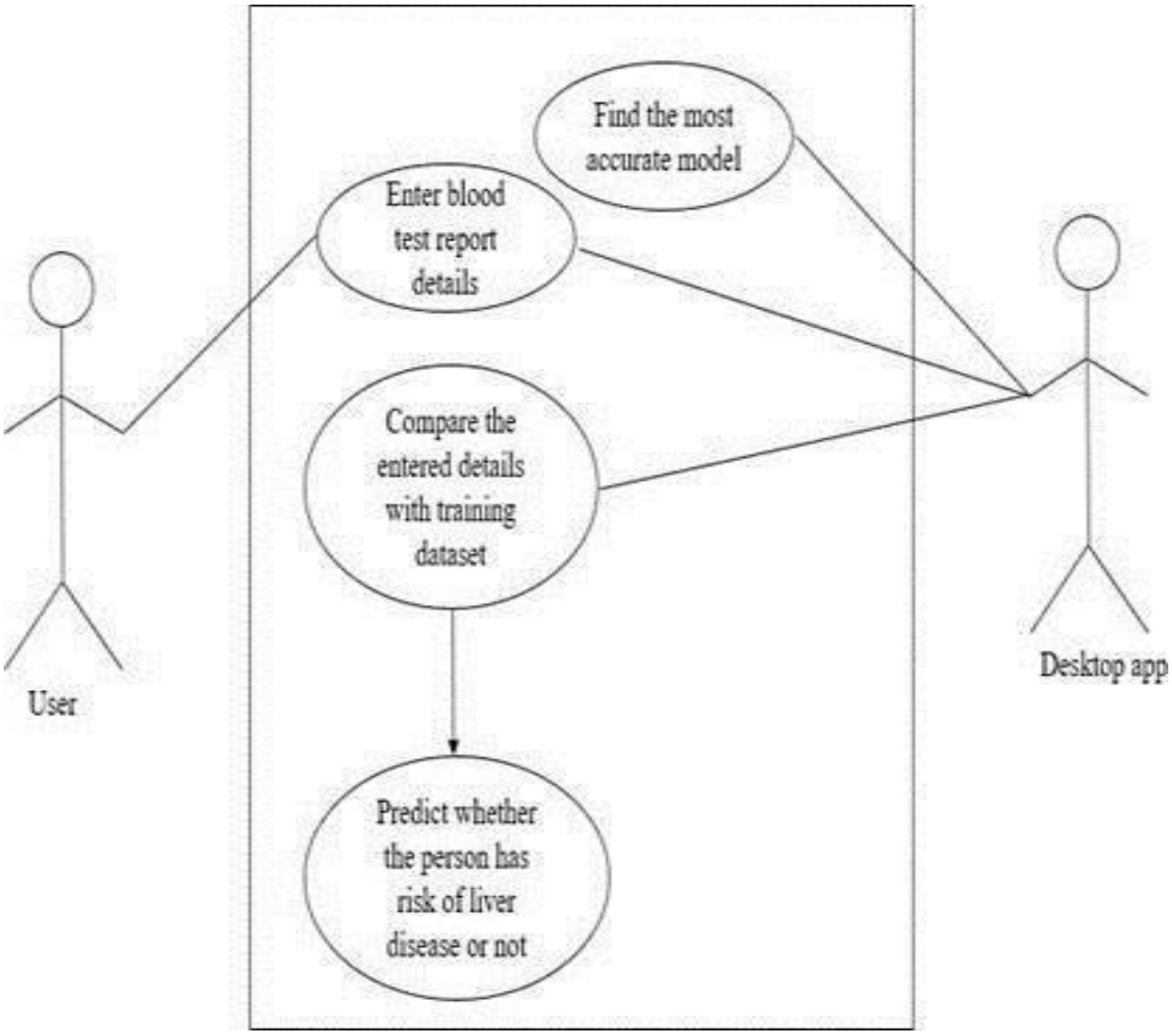


**Figure 5.2 - Describes the flow of the project deployment**

The flow through which applications, modules, updates, and patches are delivered from developers to users. The methods used by developers to build, test and deploy new code will impact how fast a product can respond to changesin customer preferences or requirements and the quality of each change.

## 5.2 SOLUTION AND TECHNICAL ARCHITECTURE

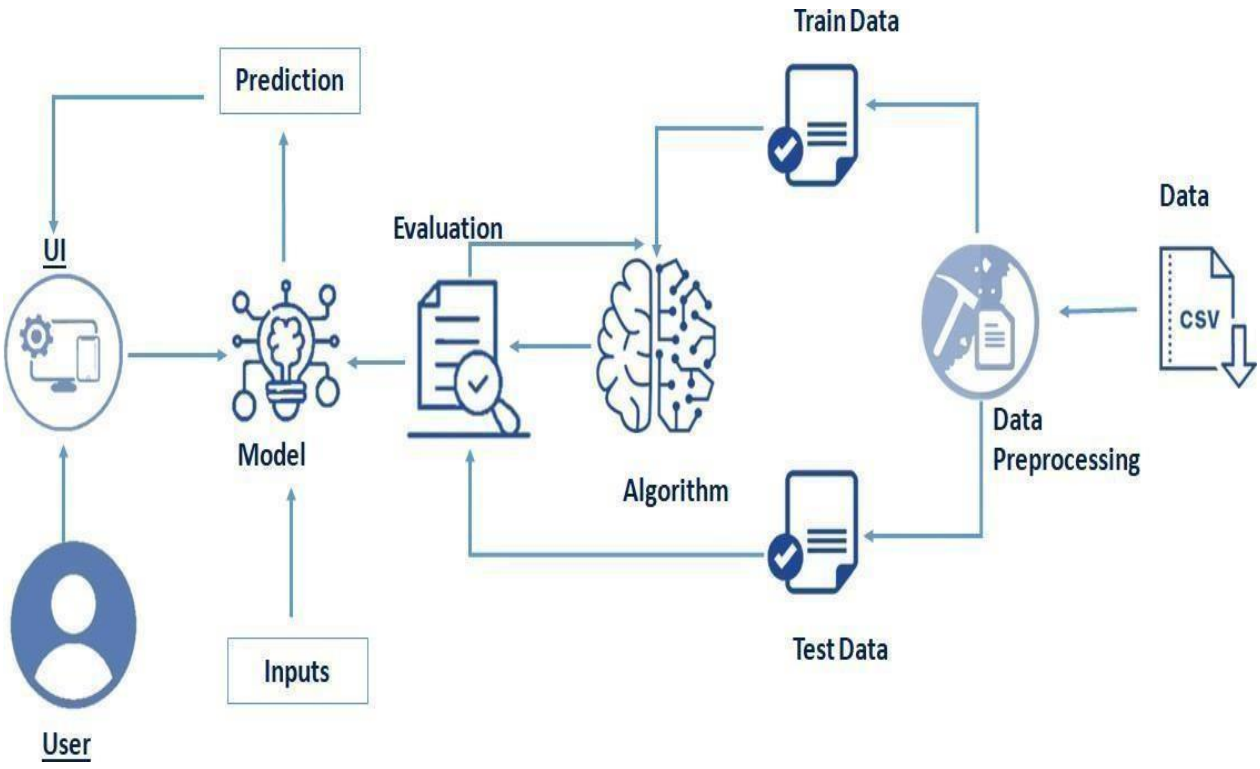


**Figure 5.3 - The process of architectural description**

Solution Architects are most similar to project managers, ensuring that all parties, including stakeholders, are on the same page and moving in the right direction at all stages. Technical architects manage all activities leading to the successful implementation of a new application. A solution architect must have a technical background with at least eight years of work experience in one or more IT areas including but not limited to: IT architecture, infrastructure, and cloud development.

## 5.3 USER STORY

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer (Web user) Step 1 | 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 |
| Customer Step 2 | Login | USN-2 | As a user, I can log into the application by entering email & password | I can access my account / dashboard | Medium | Sprint-1 |
| Customer Step 3 | Dashboard | USN-3 | In our dashboard it will display about the symptoms of the liver disease | User can know the details about the liver disease | Medium | Sprint-2 |
| Customer Step 4 | Liver predication page | USN-4 | The user should enter the details of the parameter | Details of the parameter | Medium | Sprint-2 |

**Table 5.1 – User Story**

# CHAPTER 6

# PROJECT PLANNING AND SCHEDULING

## 6.1 SPRINT PLANNING AND ESTIMATION

| Sprint | Functional Requirement (Epic) | User Story Number | User Story/Task | Story Points | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Data Collection | USN-1 | Collect the suitable dataset for predicting the chronic kidney disease. | 10 | High | Jeno J |
| Sprint-1 | Data Pre-Processing | USN-2 | Datasets are transformed into useful format. | 7 | Medium | Arul Jeevika M<br>Divyabharathi T<br>Iswarya M<br>Jeno J |
| Sprint-2 | Model Building | USN-3 | Calculate the Index values | 10 | High | Arul Jeevika M |
| Sprint-2 | | USN-4 | Splitting the Model into Training and Testing from the overall dataset. | 7 | Medium | Iswarya M<br>Jeno J |
| Sprint-3 | Training and Testing | USN-5 | Train the Model using Regression algorithm and testing the performance of the model. | 10 | High | Iswarya M<br>Jeno J |
| Sprint-3 | Application Building | USN-6 | Build the HTML and python code | 7 | Medium | Arul Jeevika M<br>Divyabharathi T |
| Sprint-4 | | USN-7 | Run Flask App | 10 | High | Arul Jeevika M |
| Sprint-4 | Implementation of the Application | USN-8 | Deploy the model on IBM cloud. | 7 | Medium | Iswarya M |

**Table 6.1 - Sprint planning and estimation**

## 6.2 SPRINT DELIVERY SCHEDULE

| 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 | 10 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 8 | 29 Oct 2022 |
| Sprint-2 | 10 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 7 | 05 Nov 2022 |
| Sprint-3 | 10 | 6 Days | 06 Oct 2022 | 12 Nov 2022 | 8 | 12 Nov 2022 |
| Sprint-4 | 10 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 7 | 19 Nov 2022 |

**Table 6.2 - Sprint Delivery Schedule**

## 6.3 REPORTS FROM JIRA

### Burndown Chart

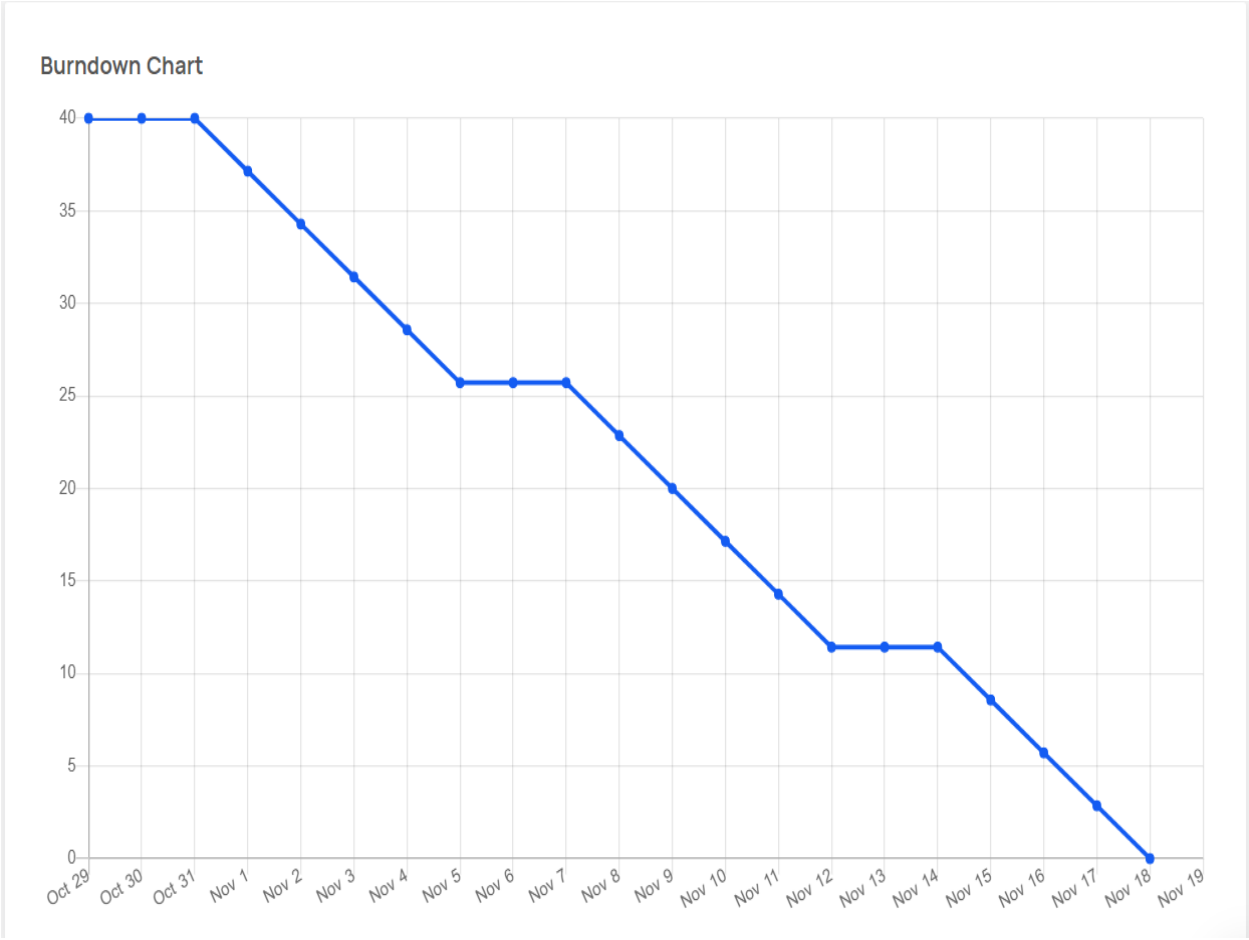


**Figure 6.1 - Burndown Chart**

A burndown chart shows the amount of work that has been completed in an epic or sprint, and the total work remaining. Burndown charts are used to predict your team's likelihood of completing their work in the time available.

## Burnup Chart

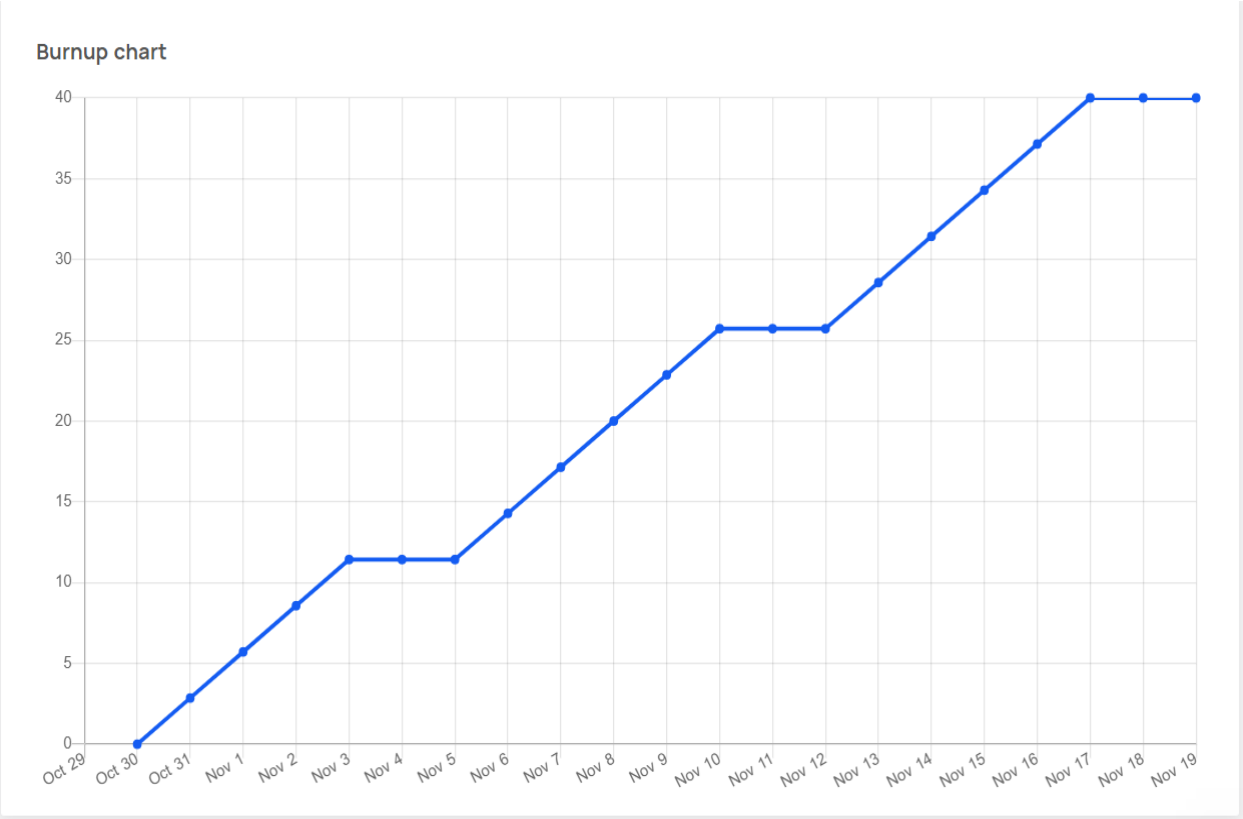


**Figure 6.2 - Burnup Chart**

A burnup chart highlights the work you've completed against your total project scope while a burn down chart highlights the amount of work remaining in a project. A burnup chart contains a work completed line and a project scope line. It displays the scope of a project and the work completed

# CHAPTER 7

# CODING AND SOLUTION

## 7.1 FEATURE 1

The first feature of the deployment is the process of Random Forest Classifier is used to train and test the model for detecting the Liver Disease withthe help of collected and pre-processed dataset collections.

```
# Train Test Split:
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X_smote,y_smote,
test_size=0.3, random_state=33)
```

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

```
# RandomForestClassifier:
from sklearn.ensemble import
RandomForestClassifierRandomForest =
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))
```

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.

```
# GradientBoostingClassifier:
from sklearn.ensemble import
GradientBoostingClassifierGradientBoost =
GradientBoostingClassifier() GradientBoost =
GradientBoost.fit(X_train,y_train)

# Predictions:
y_pred = GradientBoost.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))
```

AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classification problem. The most suited and therefore most common algorithm used with AdaBoost are decision trees with one level.

```
# AdaBoostClassifier:
from sklearn.ensemble import
AdaBoostClassifierAdaBoost =
AdaBoostClassifier()
AdaBoost = AdaBoost.fit(X_train,y_train)
```

```
# Predictions:
y_pred = AdaBoost.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))
```

### 7.2 FEATURE 2

Python flask is the first feature that helps to complete this project. It allows the user to create local server and host the website in a local machine.

```
from flask import Flask, render_template,
requestimport numpy as np
import pickle
import
requestsimport
json
```

Here we import all the necessary features of this project involving in Python flask.

```
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}
app = Flask(_name_)
model = pickle.load(open('liver2.pkl', 'rb'))

@app.route('/',methods=['GE
T']) def Home():
  return  render_template('index.html')
```

Here we created a local client's own server which serves the .html pages to the users.

```
@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_Aminotran
sferase,Aspartate_Aminotransferase,Total_Protiens,Albumin,Albumin_and_Glob
ulin_Ratio]])
        prediction = model.predict(values)
        return render_template('result.html', prediction=predictionif
_name_ == "_main_":
    app.run(debug=True
    e)
```

Here we use the inputs from the html pages which has to be get by using request method in Python Flask. By validating the values from the database, we allow the user to access the home page. render_template: Used for rendering html pages on browser. url_for: Passing the control of the program to anotherfunction. session: Creates a separate session for the individual use.

# CHAPTER 8

# TESTING

## 8.1 TEST CASES

| | | | |
|---|---|---|---|
| **TEST CASE ID** | 15358 | **TEST CASE DESCRIPTION** | STATISTICAL MACHINE LEARNING APPROACHES TO LIVER DISEASE PREDICTION |

| S.No | PREREQUISITES | TEST DATA |
|---|---|---|
| 1 | Access to Chrome Browser | By clicking the website link |
| 2 | Entering the details required | Details should be in a integer format |
| 3 | Check for correct values | Data should be filled |
| 4 | Application to train the model | Provide the datasets for model training |

**Table 8.1 - Test Details**

**Test Scenario :**

| Step | Step Details | Expected Results | Actual Results | Pass/Fail/Not/ Executed/ Suspended |
|---|---|---|---|---|
| 1 | Navigate to website link | Site should open | As Expected | pass |
| 2 | Enter the details | Details should be entered | As Expected | pass |
| 3 | Click Submit | Check the result | As Expected | Pass |
| 4 | Output results | Result are generated | As Expected | Pass |

**Table 8.2 - Test Cases**

## 8.2 USER ACCEPTANCE TESTING

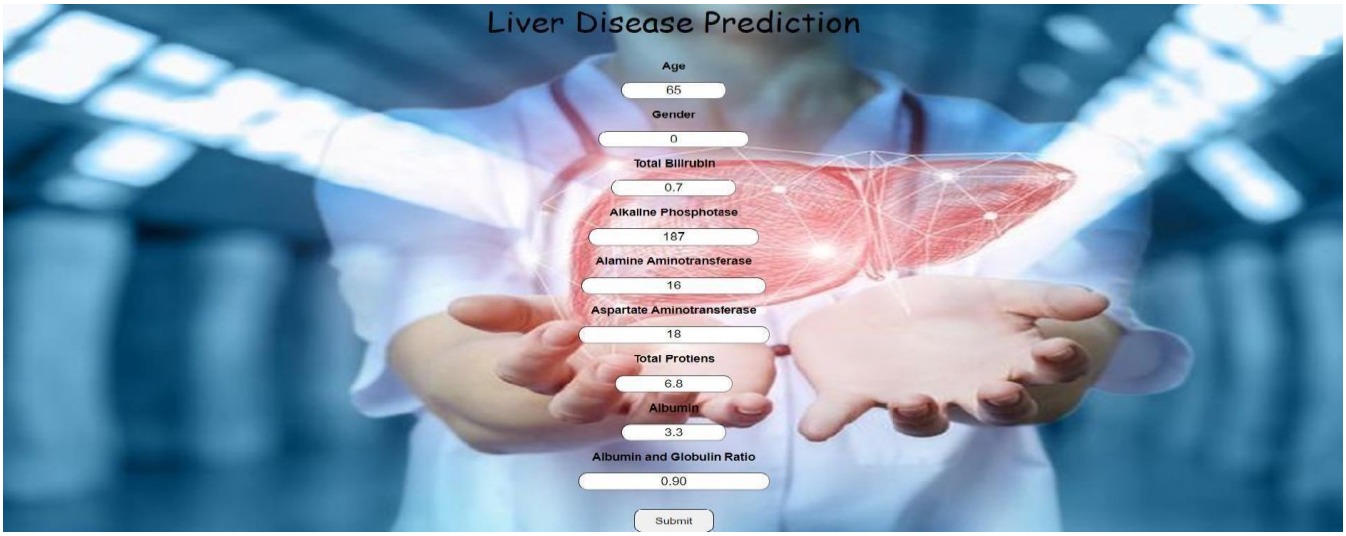


**Figure 8.1 – User Acceptance Test 1**

In this the user will be entering tested values which is moved for prediction.

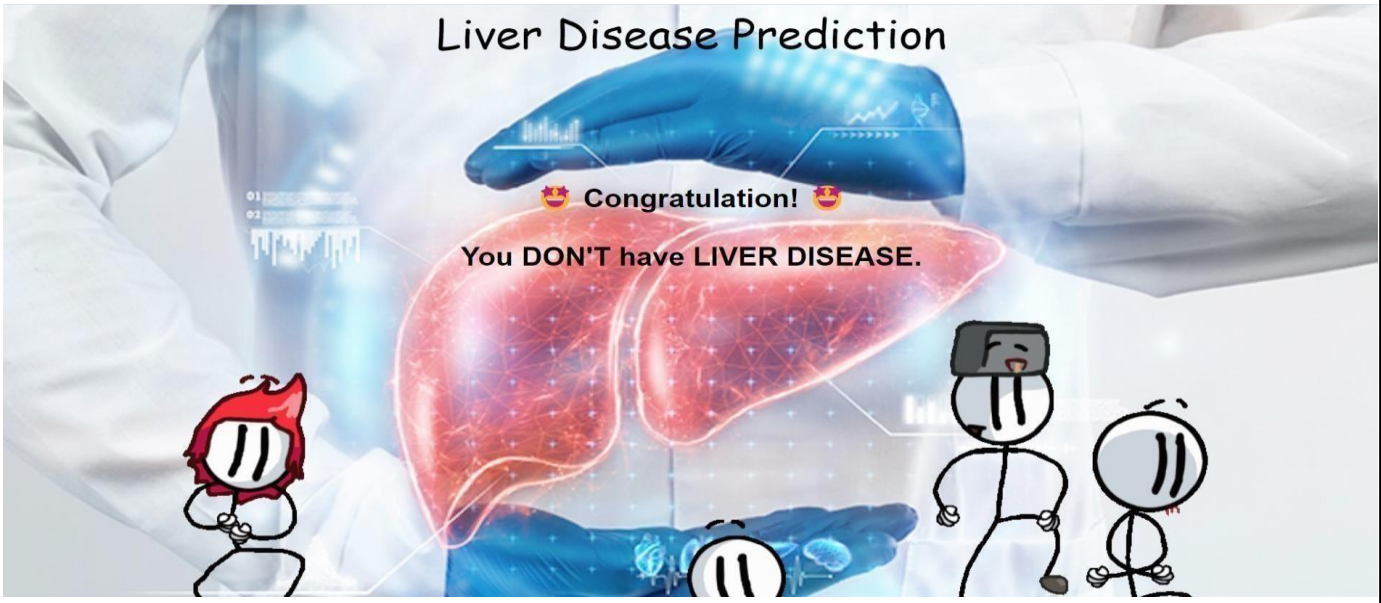


**Figure 8.2 – User Acceptance Test 2**

In this the data which was entered by the user will be analyzed. This figure show the person does not have liver disease.

.

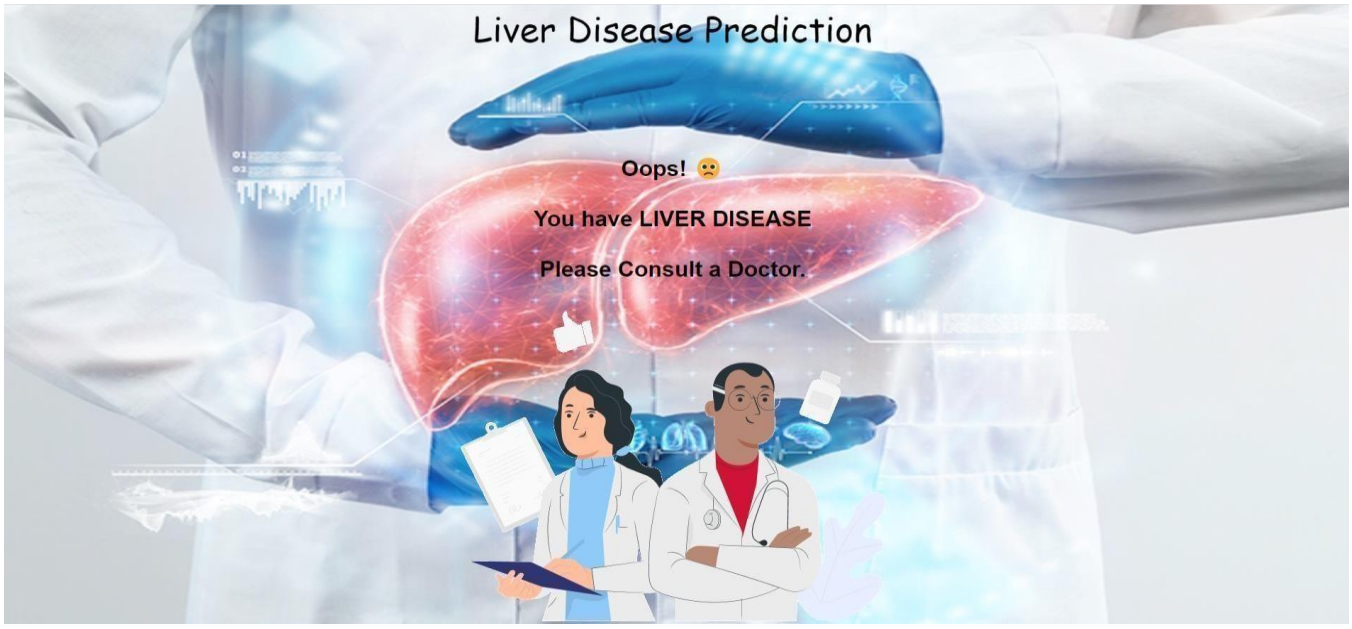


**Figure 8.3 – User Acceptance Test 3**

In this the data which was entered by the user will be analyzed. This figure shows the person have a liver disease.

# CHAPTER 9

# RESULTS

## 9.1 PERFORMANCE METRICS

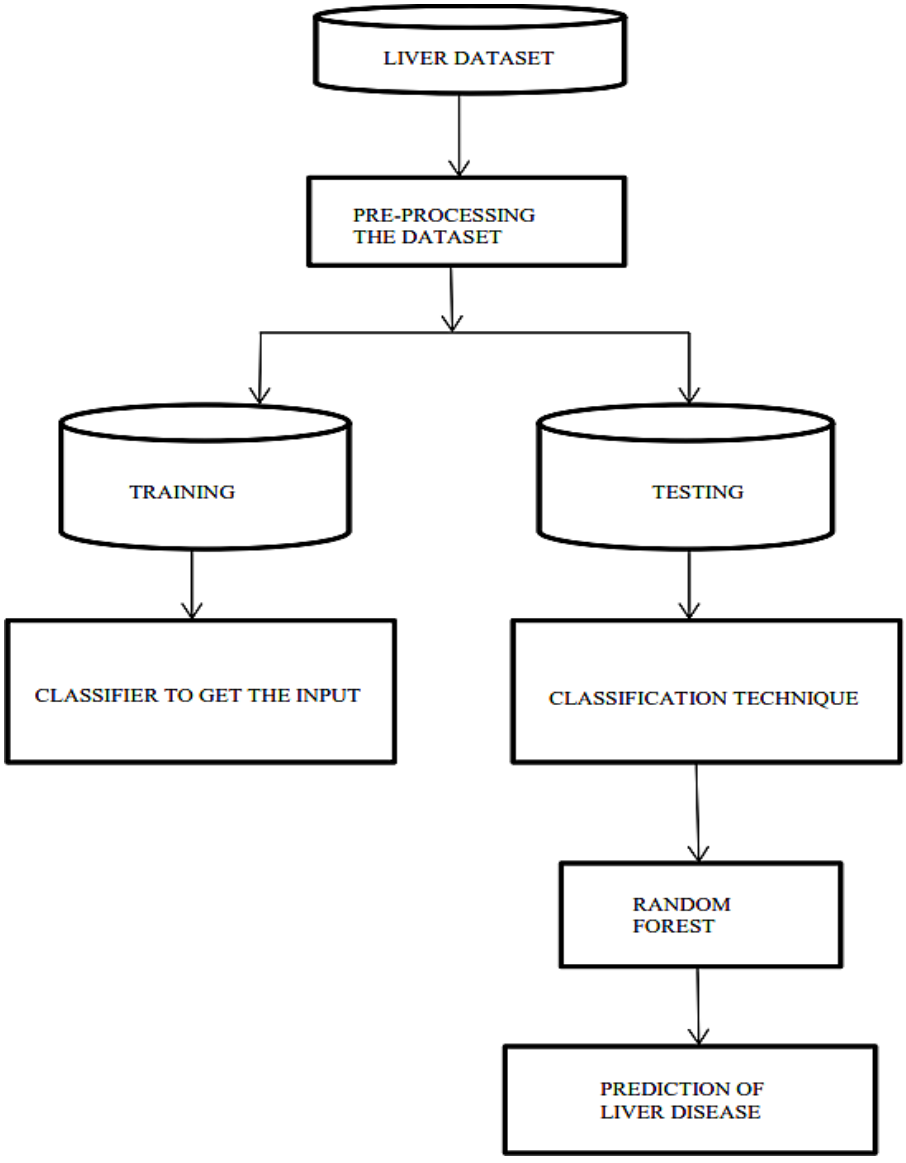


**Figure 9.1 - Flow of the Performance**

Performance Metrics is performed along for all the above assigned tasks. In each and every tasks various metrics is performed in order to provide the optimum outcome. Results gives the best outcome as expected in project.

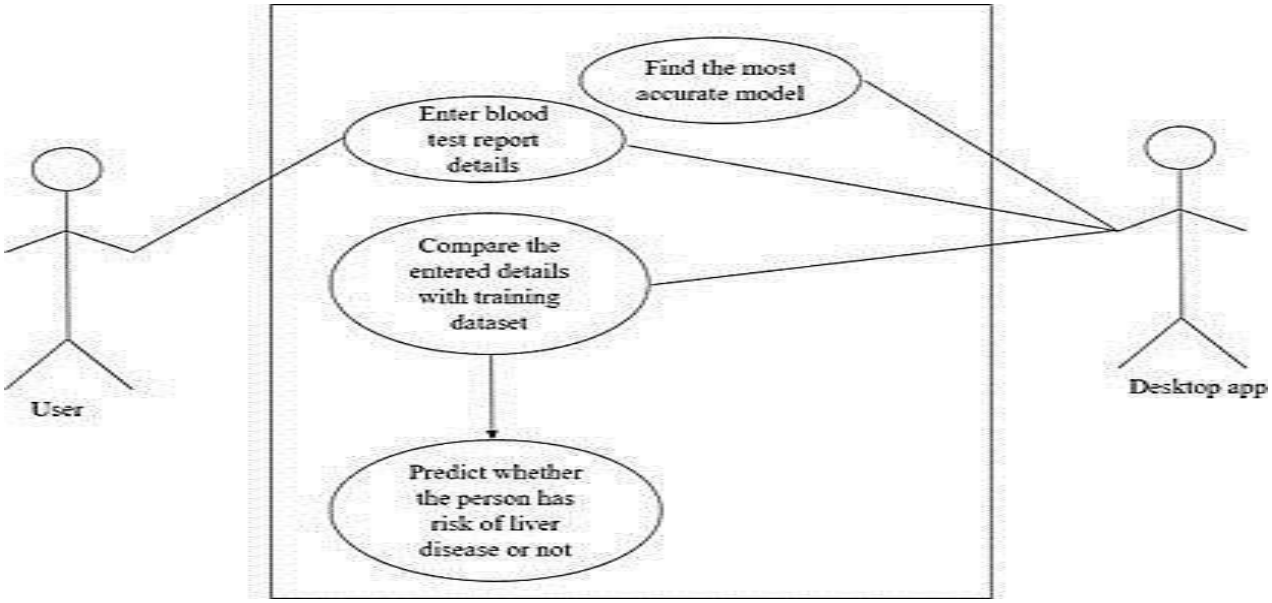


**Figure 9.2 - Performance Matrix**

One can use following execution measures for the request and figure of imperfection slanted module as shown by his/her own need.

**Confusion Matrix:** The confusion matrix is used to measure the introduction of two class issue for the given instructive record. The right corner to corner parts TP (True positive) and TN (True Negative) adequately describe instances similarly as FP (false positive) and FN (false negative) wrongly request instances. Confusion Matrix correctly classify instance TP+TN incorrectly classify instances.

- True positives imply the positive liver tuples that were precisely named bythe classifier,
- True negatives are the negative liver tuples that were precisely set apart bythe classifier.
- False positives are the negative liver tuples that were erroneously set apartas positive tuples.
- False negatives are the positive liver tuples that were incorrectly stamped negative tuples.

# CHAPTER 10

# ADVANTAGES AND DISADVANTAGES

## ADVANTAGES

In pharmaceutical industries, random forest can be used to identify the potential of a certain medicine or the composition of chemicals required for medicines. Random forest can also be used in hospitals to identify the diseases suffered by a patient ,risk of liver disease and many other diseases where early analysis and research play a crucial role. Our study has successfully developed and validated the first risk prediction model and subsequent user-friendly scoring tool, the algorithm for Liver Function Investigations, for liver condition diagnosis in patients with no obvious liver condition at the time of incident liver function testing in primary care.

## DISADVANTAGES

The idea for the approach of this project arises from the current situation regarding the increase in the confirmatory diagnosis of liver, and lack of treatment or the user's ignorance of its pathologies leads to irreversible kidney failure in the final stages of disease, such as dialysis for life, financially affecting the health system, as it is a costly treatment that generates the most significant amount of absorption of the resources available for health. This could be reduced by using tools such as machine learning to classify from the initial stages. Although the application of machine learning in healthcare and other areas is favorable, the field of liver disease has not yet exploited its full potential.

# CHAPTER 11

# CONCLUSION

The principal part of this work is to make an effective diagnosis system for liver infection patients. The application will have the option to predict liver infection prior and advise the wellbeing condition. This application can be surprisingly gainful in low-salary nations where our absence of medicinal foundations and just as particular specialists. In our study, there are a few bearings for future work in this field. 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.

# CHAPTER 12

## FUTURE SCOPE

Diseases related to liver is 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 the sedentarylifestyle and luxuries that are continuously being introduced and enhanced; theproblem is going to last long. So, in such a scenario, our project will beextremely helpful to the society. With the dataset that we used for this project,we got 81% accuracy for Random forest model, and though it might be difficultto get such accuracies with very large datasets, from this projects results, onecan clearly conclude that we can predict the risk of liver diseases with accuracyof 90 % or more. Also it can be incorporated into a website and these app andwebsite will be highly beneficial fora large section of society.

## CHAPTER 13

## APPENDIX

### SOURCE CODE:

**Algorithm :**

```
#Importing The Libraries import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns import pickle
#
Reading the Dataset
data=pd.read_csv(path)
#
Exploratory Data Analysis
data.head()
data.tail()
data.info()
data.describe()
#
Checking ForNull Values And Handling Null Values
data.isnull().any()
data.isnull().sum()
data['Albumin_and_Globulin_Ratio']=data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
data.isnull().sum()
#
Data Visualization
sns.countplot(data=data,x='Gender',lable='Count')
m,f=data['Gender'].value_counts()
print("No of Males:",m)
print("No of Females:",f)
```

```
sns.countplot(data=data, x='Dataset')
LD,NLD=data['Dataset'].value_counts()
print("liverdisease patients:",LD)
print("Non-liverdisease patients:",NLD)

#Splitting The Dataset Into Dependent And Independent
Variable x=data.iloc[:,0:-1]
y=data.iloc[:,-1]

#Split The Dependent And Independent Features Into Train Set And
TestSet from sklearn.model_selction import train_test_split
xtrain.xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)

#Check the shape of both xtrain and xtest.
xtrain.shape
xtest.shape
#importing the machine learning model
from sklearn.svm import svc
from sklearn.ensemble import RandomForestClassifierfrom
sklearn.neighbors import KNeighborsClassifier
#initiailizing the machine learning models
svm=SVC()
RFmodel=RandomForestClassifier()
KNNmodel=KNeighborsClassifiers()
svm=SVC()
#Train the data with SVM model
svm.fit(xtrain,ytrain)
#Random forest classifiers model
from sklearn.ensemble import
RandomForestClassifierRFmodel=RandomForestClassifier()
#Train the data with random forest model
RFmodel.fit(xtrain,ytrain)
SVMaccuracy=accuracy_score(SVMpred,ytest)
SVMaccuracy
```

```
from sklearn.ensemble import
RandomForestClassifierRFmodel=RandomForestClass
ifier() RFmodel.fit(xtrain,ytrain)
RFpred=RFmodel.predict(xtest)
RFaccuracy=accuracy_score(RFpred,ytest)
RFaccuracy
RFcm=confusion_matrix(RFpred,ytest)
RFcm
from sklearn.neighbors import KNeighborsClassifierKNN=KNeighborsClassifier()
KNN.fit(xtrain,ytrain)
KNNpred=KNN.preduct(xtest)
KNNaccuracy=accuracy_score(KNNpred,ytest)
KNNaccuracy
KNNcm=confusion_matrix(KNNpred,ytest)
KNNcm
# saving the model
import pickle
pickle.dump(svm,open('liver_analysis.pkl','wb'))
```

**Flask Connection**

```
from flask import Flask, render_template,
requestimport 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_Amin
o
transferase,Aspartate_Aminotransferase,Total_Protiens,Albumin,Albumin
_a nd_Globulin_Ratio]])
        prediction = model.predict(values)
        return render_template('result.html', prediction=prediction)
ifname == " main ":
   app.run(debug=True)
```

**HTML**
**Templates**
**index.htm**

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>LiverPrediction Model</title>


</head>
<body>
   <div class="container">
```

<h2 class='container-heading'>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>
      <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>
      </form>
   </div>
<style>
/* Background Image */
body
{
background-
image:url("https:/    akcdn.detik.net.id/community/media/visual/2020/12/17
/6-makanan-untuk-mencegah-penyakit-liver-salah-satunya-kopi-5_43.jpeg?w=250&q=");
height: 100%;
```

```
/* Centerand 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;
}
/* Box */
        #first {
            border-radius:
            14px;height: 25px;
```

```
    width:    150px;
    font-size:    20px;
    text-align: center;
}
#second {
    border-radius:
    14px;height: 25px;
    width:    220px;
    font-size:    20px;
    text-align: center;
}
#third {
    border-radius:
    14px;height: 25px;
    width:    180px;
    font-size:    20px;
    text-align: center;
}
#fourth {
    border-radius:
    14px;height: 25px;
    width:    250px;
    font-size:    20px;
    text-align: center;
}
#fifth {
    border-radius:
    14px;height: 25px;
    width: 270px;
    font-size: 20px;
```

```
    text-align: center;
}
#sixth {
    border-radius:
    14px;height: 25px;
    width:    280px;
    font-size:    20px;
    text-align: center;
}
#seventh {
    border-radius:
    14px;height: 25px;
    width:    170px;
    font-size:    20px;
    text-align: center;
}
#eight {
    border-radius:
    14px;height: 25px;
    width:    150px;
    font-size:    20px;
    text-align: center;
}
#ninth {
    border-radius:
    14px;height: 25px;
    width:    280px;
    font-size:    20px;
    text-align: center;
}
```

```
/* Submit Button */
#sub {
        width: 120px;
        height:  43px;
        text-align: center;
        border-radius:
        14px; font-size:
        18px;
    }
</style>
</body>
</html>
```

**result.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"  content="width=device-width,  initial-scale=1.0">
    <title>LiverDisease  Result</title>
</head>
<body>
    <div class="container">
        <form action="{{ url_for('predict')}}" method="post">
            <h2 class='container-heading'><span class="heading_font">LiverDisease
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
LIVERDISEASE <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>
<style>
/* Background Image */
body
{
background-image:url("https:/ d1vbn70lmn1nqe.cloudfront.net/prod/wp-
content/uploads/2022/05/11050117/hepatitis-kronis-halodoc.jpg");
height: 100%;
/* Centerand 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>
```

**Github and Project Video Demo**

**LinkGithub Link:**

https://github.com/IBM-EPBL/IBM-Project-17559-1659673380

**Project Video Demo Link:**

https://drive.google.com/file/d/1NGI351dv4rdYhfoXA9GQ3VSJl2z9PMk9/view?usp=sharing