IBM PROJECT

Statistical Machine Learning Approaches to Liver Disease Prediction

Team ID	PNT2022TMID52974
Project Name	Statistical Machine Learning Approaches to Liver Disease Prediction

TEAM MEMBERS :

Shweatha J - Team Leader

S.Rakshana

Raveena R

Sakthi Devi S V

1.INTRODUCTION

1.1 Project Overview

Health is the state of physical, mental and social well-being, Leading a productive life is from having good health i.e., proper functioning of organs that are interconnected. The organ LIVER is crucial organ for one's existence as its one of main functions is detoxification, cleansing blood by eliminating toxins. Liver disease refers to several conditions that badly affects one's health. Liver disease can be predicted/detected by liver function test generally where it takes a day for result and it can go undetected if liver's function may decrease for a while. When undetected for a long time it leads to life threatening condition. So regular health examination is must. The project proposes a model /solution to above problem to predict the liver disease in faster manner using machine learning approaches or methods for early detection .

1.2 Project Purpose

Lets say leading a healthy and happy life is every single beings' motto. Leading a healthy life is by proper functioning of organs. It is said that 40% of Indians suffer from NAFLD(Non-alcoholic fatty liver disease) and 1 among 10 people in America suffer from some sort of liver disease. The cost of liver function tests is affordable by people but when undetected due to SLOW FUNCTIONING of liver leading to critical conditions the cost sums a lot that can't be afforded by all people. The proposed model/solution will be made as an app where a individual can take their own tests by answering some vital questions while not feeling well, and they can consult their doctors regarding their symptoms and taking medications as directed.

1.3 Existing Problems

PROBLEM STATEMENT:

To develop machine learning approaches to predict liver disease by training datasets in existing machine learning methods and the testing the dataset of patient to get the results. This model is made into an user interface app where it is reached to all corners of people effectively.

What does the problem affect? - Health of an individual

What are the boundaries of the problem? - Human life

Why is it important to fix this problem? – "No liver = No life"

2. LITERATURE SURVEY

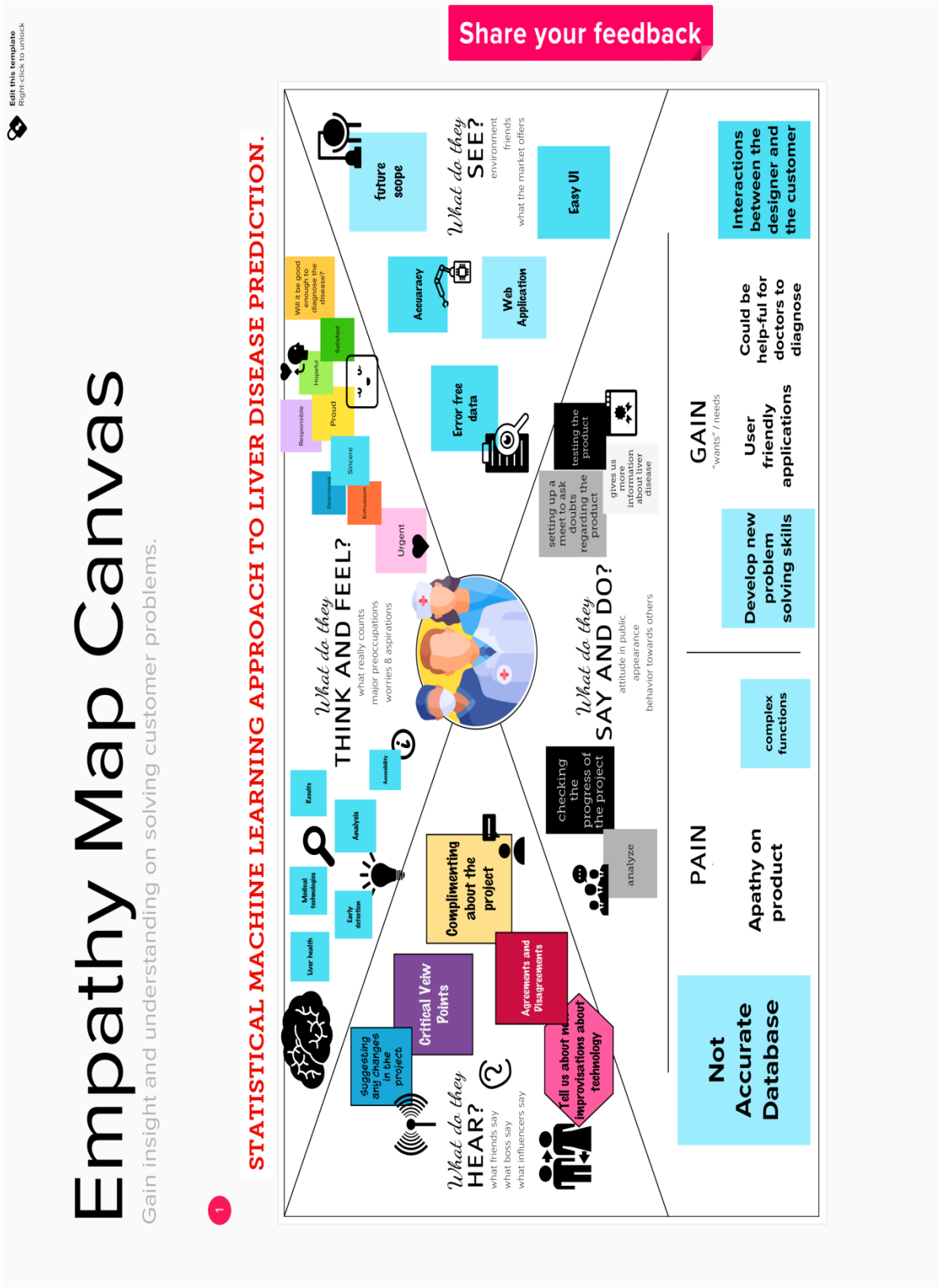
S.No	Title	Method used	Description	link
1	Supervised machine learning based liver disease prediction approach with LASSO feature selection	1.LASSO feature extraction method 2.10 fold cross validation approach 3.nearly 8 classification algorithms(LR, DT, RF, AdaBoost KNN, LDA,gradient boosting and SVM.)	Reduces overfitting using LASSO. Comparing classification algorithm based on accuracy, sensitivity, precision and f1-scores which help to identify the highest performing algorithm . 10 foldcross validation to split dataset for training and testing. LR has more accuracy than others followed by LDA&SVM.	https://beei.org/index.php/EEI/article/view/3242/2437
2	Intelligent Techniques and Comparative Performance Analysis of Liver Disease Prediction	Bioinfomatics &Random Forest (RF), Multilayer Perceptron (MLP) model, k Nearest Neighbour (kNN), and Support Vector Machine (SVM)	genetic data of patient is known by bioinformatics and drugs are invented into the area of defects in genetic structure. SVM has more accuracy than others	https://kalaharijournals.com/resources/IJME_Vol7.1_756.pdf

3	Accurate liver disease prediction system using convolutional neural network	Modified Principal component analysis - preprocessing Score based Artificial Fish Swarm Algorithm (SAFSA)- Optimal feature selection Modified CNN- classification	The attributes, ”Accuracy, Precision, Recall and F-measure” are compared between modified CNN (MCNN) and Multi layer perceptron neural network (MLPNN) MCNN has higher % than MLPNN	https ://doi.org/10.17485/IJST/v14i17.451
4	Liver Disease Prediction Using Machine Learning Algorithm	kNN Random Forest	Datasets are taken from open source platform and comparison is made between ML methods	https://doi.org/10.1007/978-981-16-0171-2_56
5	Evaluation based approaches for Liver disease prediction using Machine learning Algorithms	SVM LR	The main objective is to predict tumour or disease with respect to data mining techniques. Comparison of evaluation parameters for SVM and LR is made	https://doi.org/10.1109/ICCCI50826.2021.9402463

3. PROJECT DESIGN AND PLANNING

3.1 Ideation phase

3.1.1 Empathy Map :



3.1.2 Ideation and Brainstorming

Template

Idea prioritization

Use this framework to rank ideas based on their feasibility and impact to visually compare the merits of multiple ideas. Deliver a set of ideas that your team wants to try out, and identify which of them need to be prioritized.

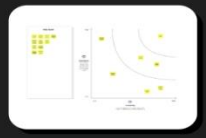
Share template feedback

Collect your ideas in one place

Jot down different ideas your team is interested in trying out. These could be different solutions, or different approaches to the same solution. As a team, go through the ideas in the Idea bank one by one and place them on the grid. Take the time to discuss each idea and come to a consensus on where it should go.

Idea bank

- Spread awareness to rural PHCs with the help of NGOs
- Updating with fast and highly performing and latest algorithms to already existing product if needed.
- Have a patient hotline
- Make liver test a part of annual health checkup packages
- Having doctors to spread awareness through consultation.

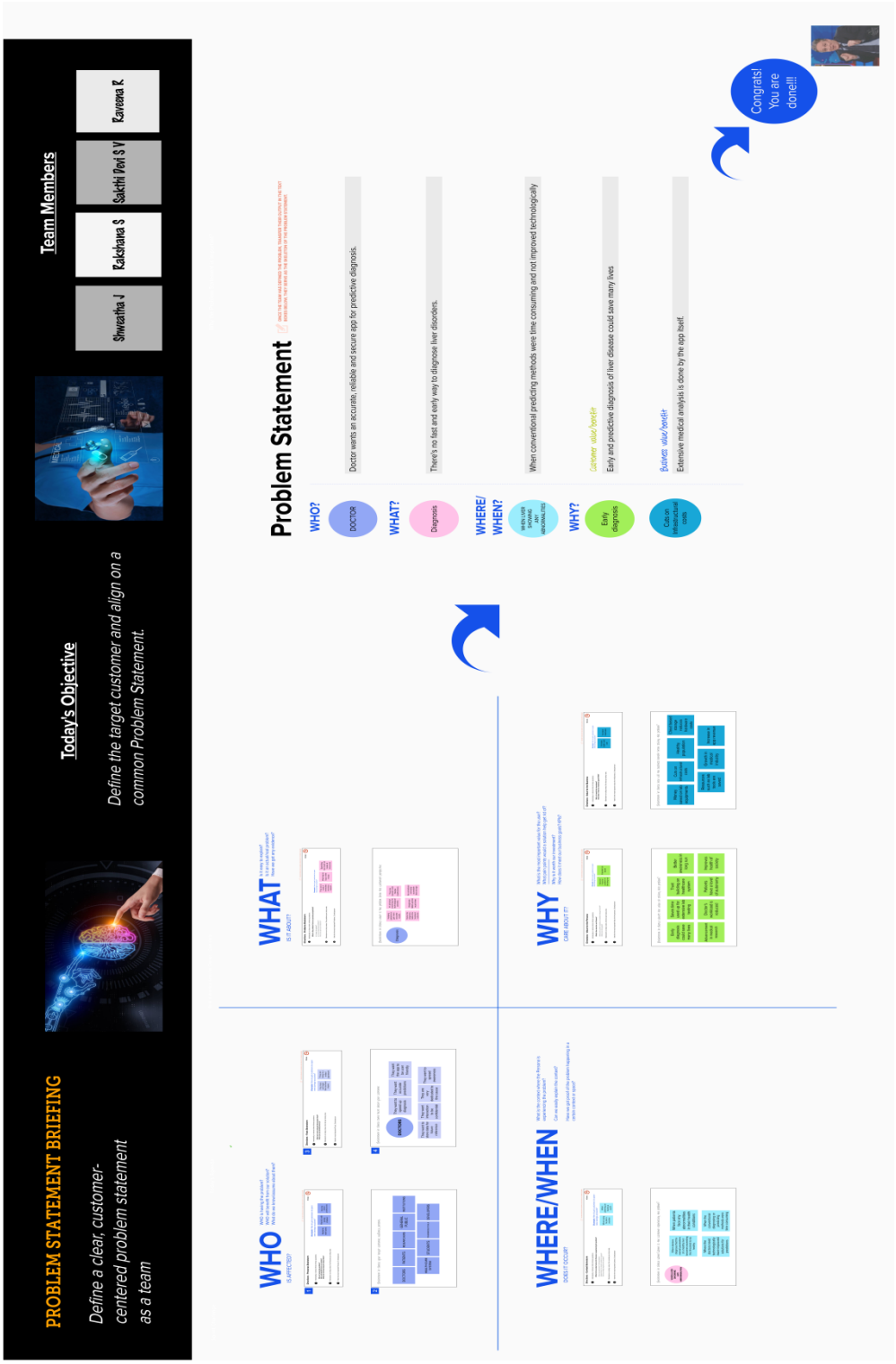


Need some inspiration?

See a finished version of this template to kickstart your work.

Open example

3.1.3 Problem statement



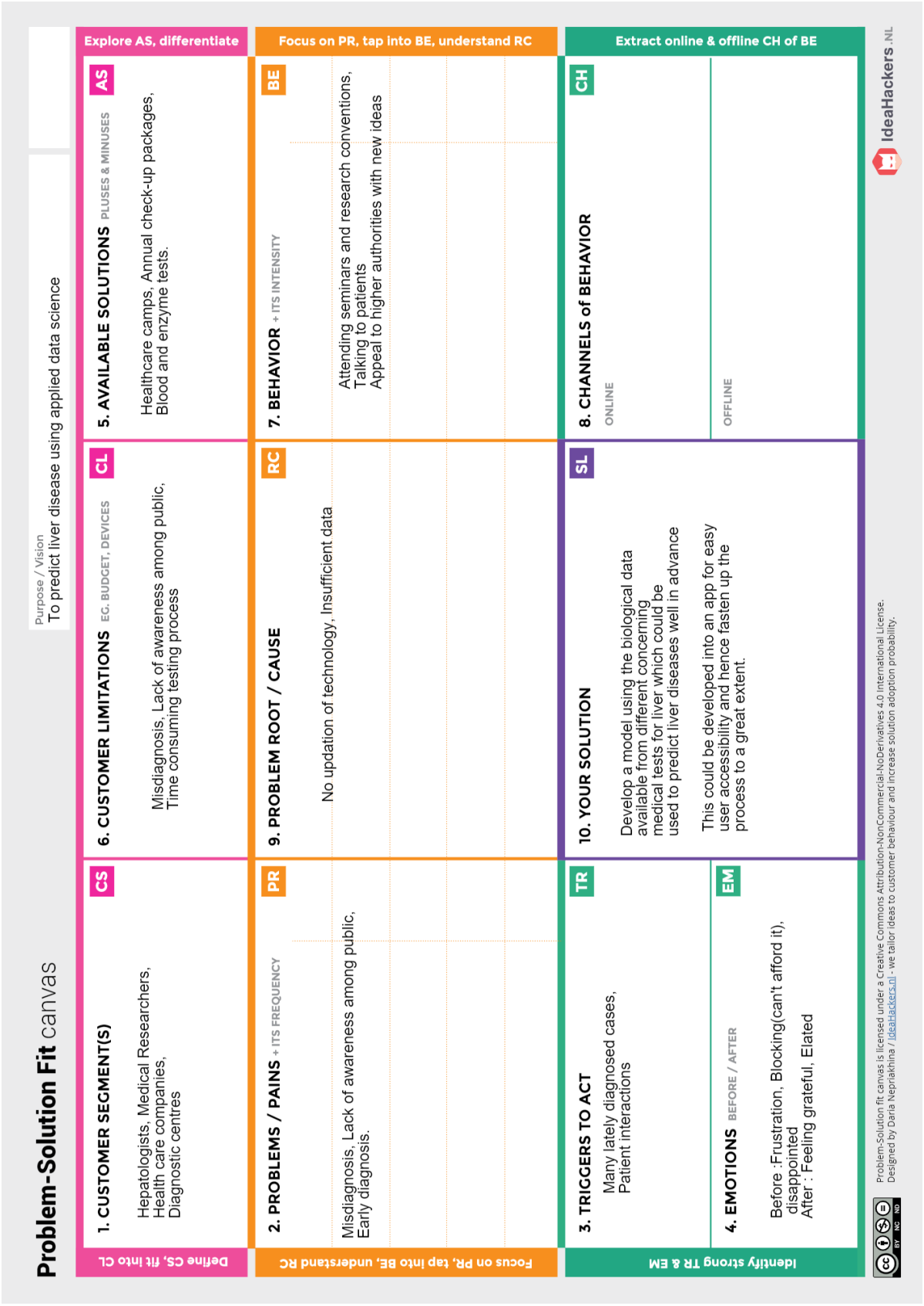
3.2 Project design Phase I

3.2.1. Proposed solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	To develop machine learning approaches topredict liver disease
2.	Idea / Solution description	Testing the data using existing machine learning methods and predicting the solution by giving input samples
3.	Novelty / Uniqueness	The proposed solution takes lesser time than recent testing time(i.e, few hours to day)
4.	Social Impact / Customer Satisfaction	People will get to know about their health condition thereby taking precautionary measures.
5.	Business Model (Revenue Model)	Subscription based health application
6.	Scalability of the Solution	Can be used as app in mobile, web function

3.2.2. Problem solution fit



3.2.3 Solution Architecture

Solution Architecture Diagram:

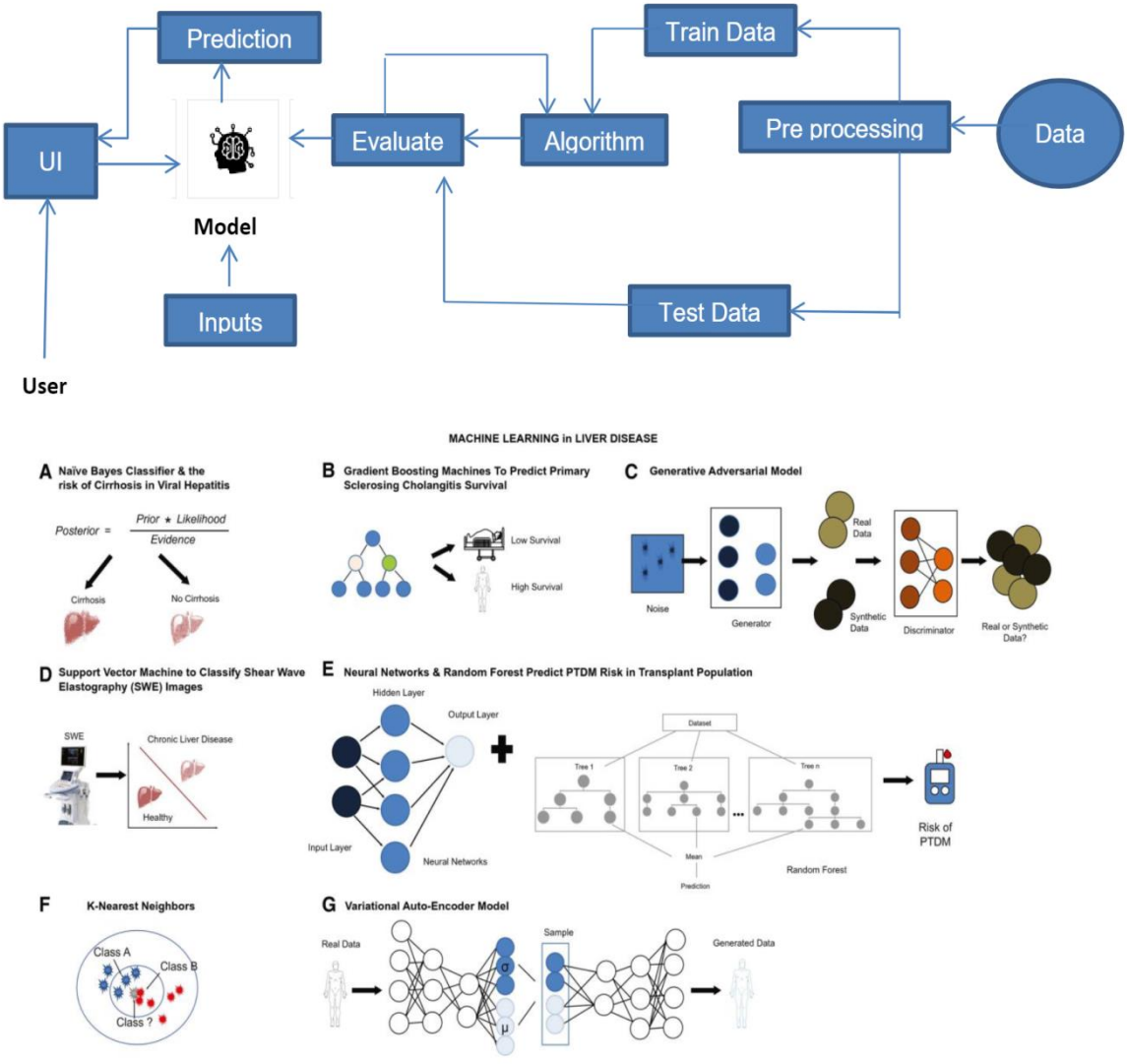
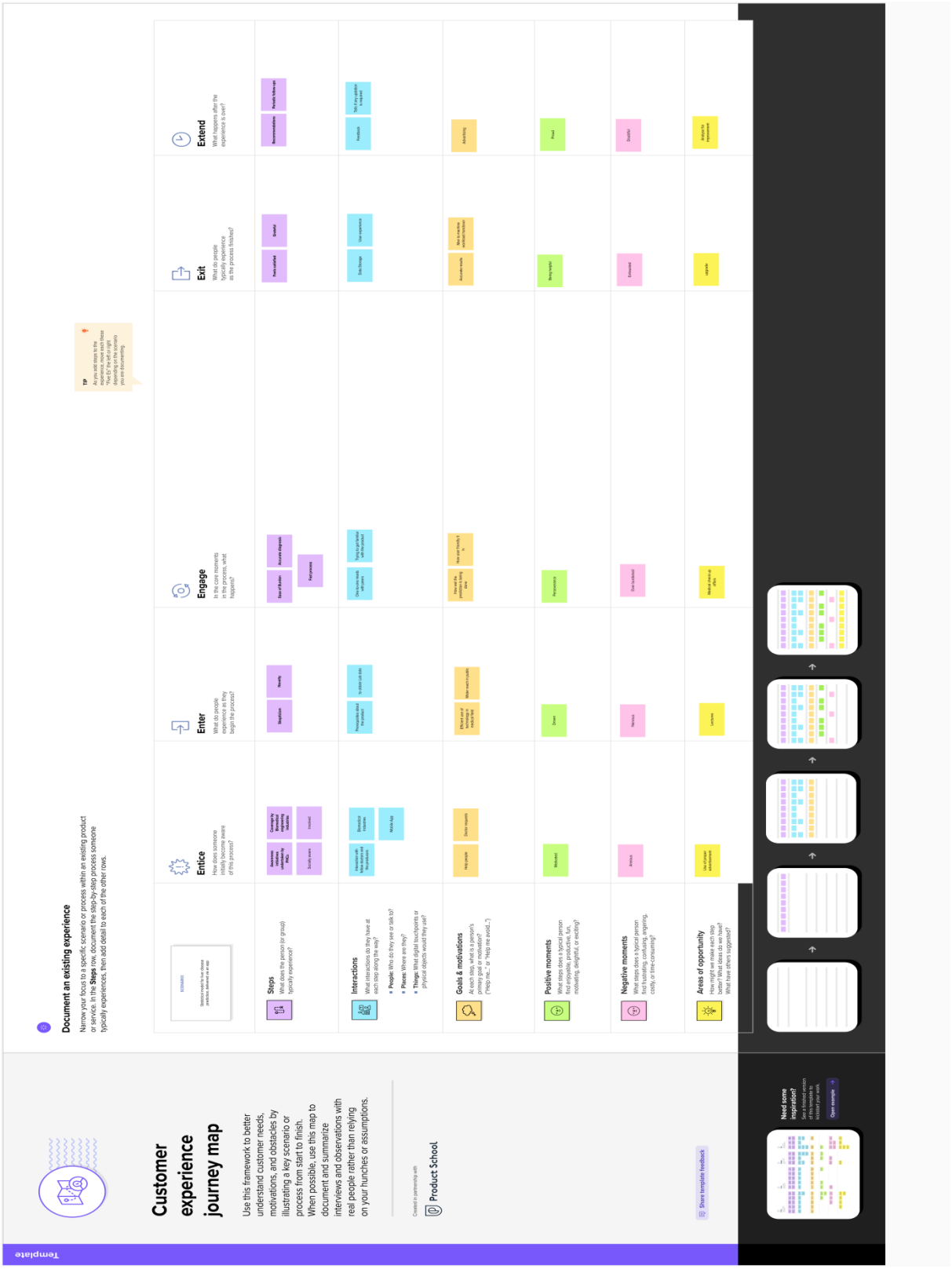


Figure 1: Architecture and data flow of the liver disease prediction system

3.3 Project design Phase II

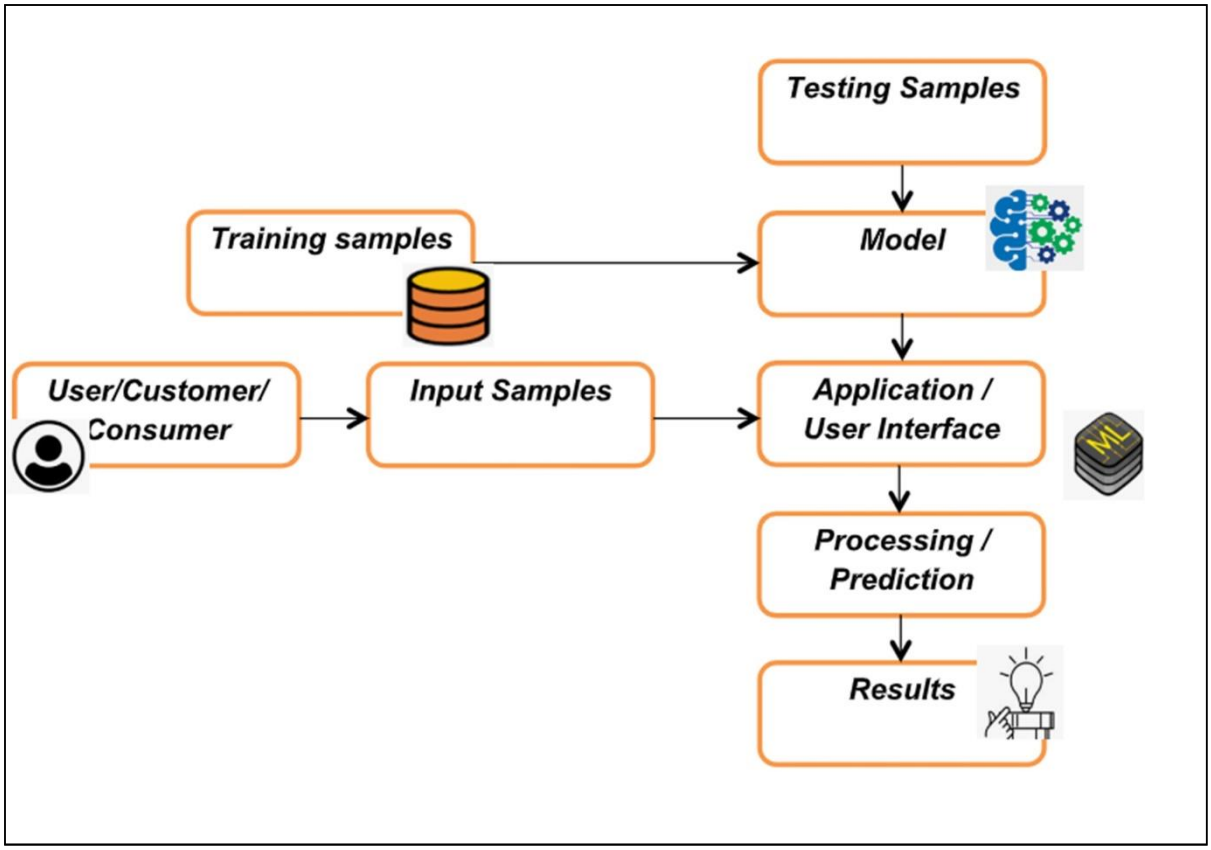
3.3.1 Customer Journey Map



3.3.2 Dataflow diagrams and User stories

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored

User Type	Functional Requirement(Epic)	User Story Number	User Story /Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Installation	USN-1	As a user, I want to easily access the app via appstore.	I could easily access the app via app store	High	Sprint 1
	Registration	USN-2	As a user, I should register to the app using my credentials.	I am able to register by providing my credentials.	High	Sprint 1
	Demo	USN-3	As a medical practitioner, I want to easily understand the app via a demo	I am able to learn about the functionalities via a demo.	Low	Sprint 1
	Dashboard	USN-4	As a user, I need a well- built user interface	I am able to efficiently use the dashboard	High	Sprint 1



3.3.3 Solution requirements

Functional Requirements:

Following are the functional requirements of the proposed solution.

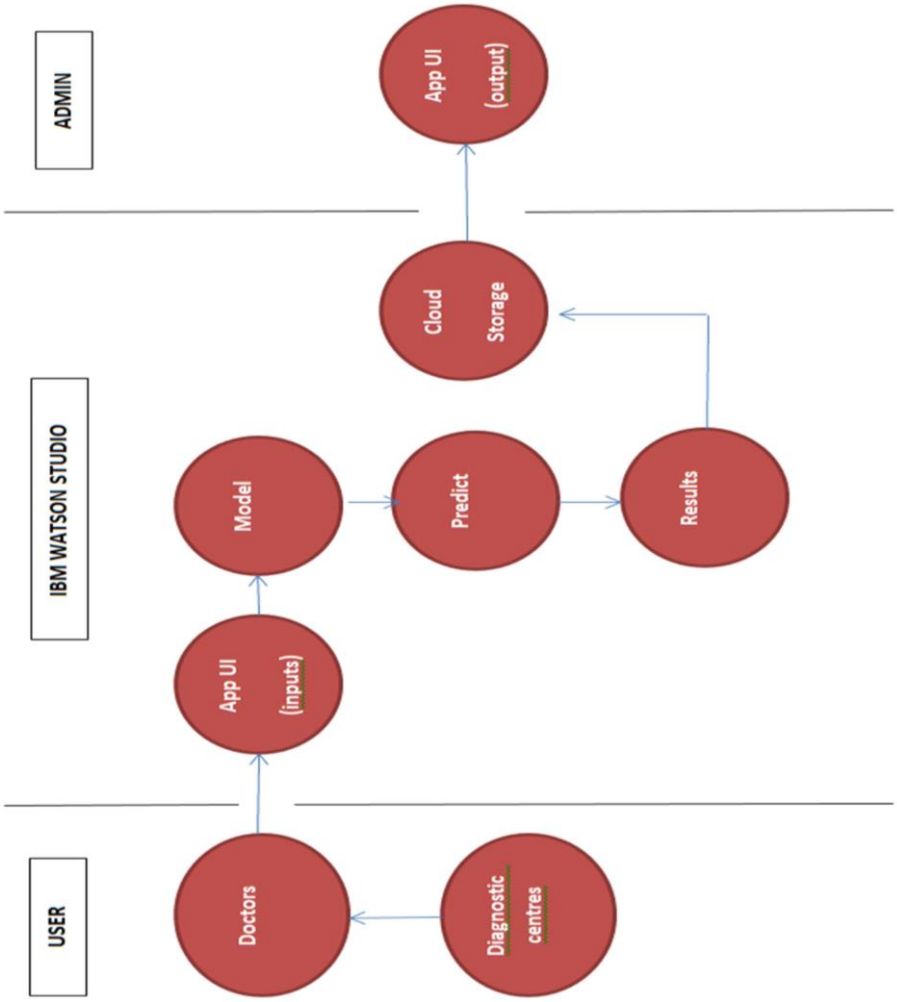
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Portal Registration	Registration to common Database

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Should User-friendly
NFR-2	**Security**	Authentication
NFR-3	**Reliability**	Results should be accurate
NFR-4	**Performance**	Fast
NFR-5	**Availability**	Widely available
NFR-6	**Scalability**	Mobile and Web app

3.3.4 Technology Stack (Architecture & Stack)



3.4 Project Planning

3.4.1 Milestone and Activity list

ASSIGNMENT AND MILESTONE TRACKER

Subject	Assigned to	Status
Pre requisites	Team	Done
Quiz 1	Shweatha	Done
Quiz 2	Sakthi	Done
Quiz 3	Rakshana	Done
Quiz 4	Raveena	Done
Assignment 1	Sakthi	Done
Assignment 2	Shweatha	Done
Assignment 3	Rakshana	Done
Assignment 4	Raveena	Done
Brain storming	Team	Done
Problem statement	Team	Done
Empathy map	Team	Done
Solution Architecture	Team	Done
Problem- solution fit	Team	Done
Solution Requirements	Team	Done
Customer/user journey map	Team	Done
Functional req	Team	Done
Data flow diagrams	Team	Done
Tech arch	Team	Done
Data pre processing	Raveena	Done
Data visualization	Sakthi	Done
Splitting the dataset	Rakshana	Done
Model building	Raveena	Done
Application building	Team	Done
Train on IBM Cloud	Team	Done
Run the application	Team	Done
Report	Team	Done

3.4.2 Sprint delivery plan

Sprint	**Functional Requirement**	**Task**	**Priority**	**Team Members**
Sprint-1	Registration	Register for IBM Cloud Services. Install Anaconda navigator and necessary packages. Complete design phase and assignments	High	Rakshana S Raveena R Sakthi Devi S V Shweatha J
Sprint-2	Data processing	Download dataset and process it using python tools. Visualize the dataset. Split the dataset into testing and training dataset.	High	Raveena R Sakthi Devi S V
Sprint-3	Model Building	Use various machine learning algorithms to build the model with best accuracy	Medium	Rakshana S Shweatha J
Sprint-4	Application & training	Build the web application and train on IBM cloud services.	Medium	Rakshana S Raveena R Sakthi Devi S V Shweatha J

4 PROJECT DEVELOPMENT PHASE

4.1 Dataset Collection

This data set contains 416 liver patient records and 167 non liver patient records collected from North East of Andhra Pradesh, India. The "Dataset" column is a class label used to divide groups into liver patient (liver disease) or not (no disease). This data set contains 441 male patient records and 142 female patient records.

Any patient whose age exceeded 89 is listed as being of age "90".

Columns:

- Age of the patient
- Gender of the patient
- Total Bilirubin
- Direct Bilirubin
- Alkaline Phosphotase
- Alamine Aminotransferase
- Aspartate Aminotransferase
- Total Protiens
- Albumin
- Albumin and Globulin Ratio
- Dataset: field used to split the data into two sets (patient with liver disease, or no disease)

indian_liver_patient.csv (23.93 kB)

Detail Compact Column

11 of 11 columns

About this file

Based on chemical compounds(bilrubin,albumin,protiens,alkaline phosphatase) present in human body and tests like SGOT , SGPT the outcome mentioned whether person is patient ie needs to be diagnosed or not.

# Age	A Gender	# Total_Bilirubin	# Direct_Bilirubin	# Alkaline_Phospho...	# Alamine_Aminotr...	# Aspartate_Amino...	# Total_Protiens	# Albumin
Age of the patients	Sex of the patients	Total Billirubin in mg/dL	Conjugated Billirubin in mg/dL	ALP in IU/L	ALT in IU/L	AST in IU/L	Total Proteins g/dL	Albumin in g/d
4 90	Male 76% Female 24%	0.4 75	0.1 19.7	63 2110	10 2000	10 4929	2.7 9.6	0.9
65	Female	0.7	0.1	187	16	18	6.8	3.3
62	Male	10.9	5.5	699	64	100	7.5	3.2
62	Male	7.3	4.1	490	60	68	7	3.3
58	Male	1	0.4	182	14	20	6.8	3.4
72	Male	3.9	2	195	27	59	7.3	2.4
46	Male	1.8	0.7	208	19	14	7.6	4.4
26	Female	0.9	0.2	154	16	12	7	3.5
29	Female	0.9	0.3	202	14	11	6.7	3.6
17	Male	0.9	0.3	202	22	19	7.4	4.1
55	Male	0.7	0.2	290	53	58	6.8	3.4

4.2 Dataset Pre-processing

Data Pre-processing includes the following main tasks

1. Import the Libraries.
2. Reading the dataset.
3. Analyse the data.
4. Taking Care of Missing data.
5. Data Visualization.
6. Splitting the Dataset into Dependent and Independent variables.
7. Splitting Data into Train and Test

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
```

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object
Storage. It includes your credentials.
# You might want to remove those credentials before you share
the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='81NwbGz-
nOfetpoQbtERs4vVLTRPbT5SX0ohN_GSTc4d',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-
storage.appdomain.cloud')

bucket = 'liverhealthmonitor-donotdelete-pr-z7bxt6q1mu2huc'
object_key = 'indian_liver_patient.csv'

body                                                    =
cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-
like object
if   not   hasattr(body,   "__iter__"):   body.__iter__   =
types.MethodType( __iter__, body )
```

```
data = pd.read_csv(body)
data.head()
data.tail()
data.info()
data.describe()
data.isnull().any()
data.isnull().sum()
data['Albumin_and_Globulin_Ratio']=data['Albumin_and_Globulin_
Ratio'].fillna(0)
data.isnull().sum()

sns.countplot(data=data,x='Gender',label='Count')
m,f=data['Gender'].value_counts()
print("Number of Males:",m)
print("Number of Females:",f)

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

x=data.iloc[:,0:-1]
y=data.iloc[:,-1]

from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)
xtrain.shape
xtest.shape
```

4.3 Model Building

Predictive modeling is a mathematical approach to create a statistical model to forecast future behavior based on input test data.

Model building includes the following main tasks

1. Training and testing the model using classification algorithms
    - Random Forest Classification.
    - Support Vector Machine
    - KNN Classification
2. Evaluation of Model
    - To check to see how well our model is performing on the test data.
    - Accuracy Score of 3 algorithms and choose the best out of three.
3. Save the model
4. Predicting the output using the model

```
#model building
from sklearn.svm import SVC
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(xtrain['Gender'].astype(str))
xtrain['Gender'] = le.transform(xtrain['Gender'].astype(str))
xtest['Gender'] = le.transform(xtest['Gender'].astype(str))

svm=SVC()
RFmodel=RandomForestClassifier()
KNNmodel=KNeighborsClassifier()

from sklearn.svm import SVC
svm=SVC(gamma='auto')
svm.fit(xtrain,ytrain)

from sklearn.ensemble import RandomForestClassifier
RFmodel=RandomForestClassifier()
RFmodel.fit(xtrain,ytrain)

from sklearn.neighbors import KNeighborsClassifier
KNN=KNeighborsClassifier()
KNN.fit(xtrain,ytrain)

import warnings
warnings.filterwarnings("ignore")
#accuracies
from sklearn.metrics import accuracy_score
SVMpred=svm.predict(xtest)
SVMaccuracy=accuracy_score(SVMpred,ytest)
print('SVM accuracy:',SVMaccuracy*100)

RFpred=RFmodel.predict(xtest)
RFaccuracy=accuracy_score(ytest,RFpred)
print('RFC accuracy:',RFaccuracy*100)

KNNpred=KNN.predict(xtest)
KNNaccuracy=accuracy_score(KNNpred,ytest)
print('KNN accuracy:',KNNaccuracy*100)
```

SVM accuracy: 73.50427350427351
RFC accuracy: 75.21367521367522
KNN accuracy: 75.21367521367522

```
#confusion matrix
from sklearn.metrics import confusion_matrix
cm1=confusion_matrix(ytest,SVMpred)
cm2=confusion_matrix(ytest,RFpred)
cm3=confusion_matrix(ytest,KNNpred)
print('SVM Confusion matrix',cm1)
print('RFC Confusion matrix',cm2)
print('KNN Confusion matrix',cm3)
```

```
SVM Confusion matrix [[86  0]
 [31  0]]
RFC Confusion matrix [[73 13]
 [16 15]]
KNN Confusion matrix [[69 17]
 [12 19]]
```

4.4 Application Building

Application Building involves following steps

1. Create an HTML file
2. Build a Python Code
3. Run the app

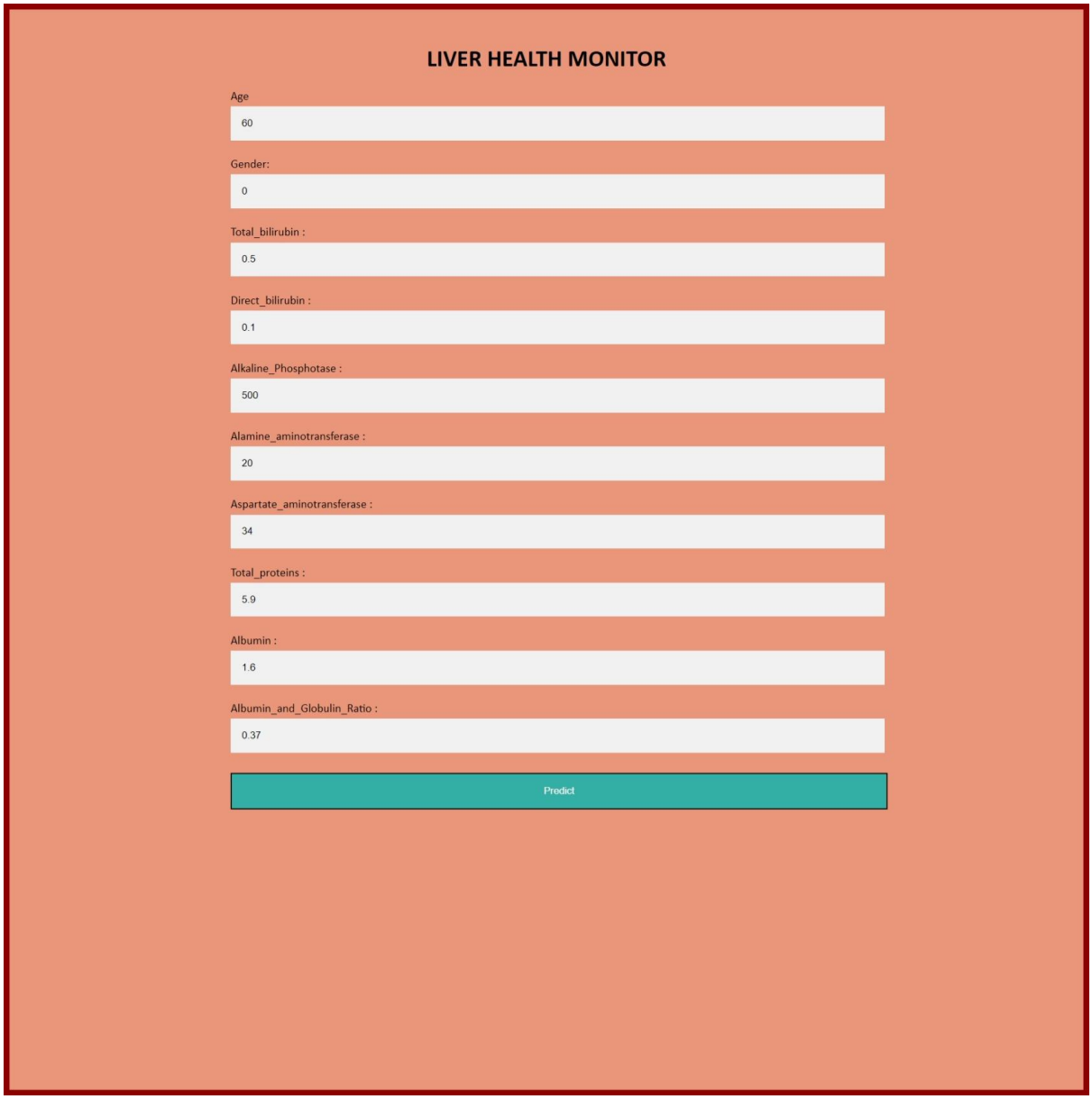
Create an HTML File:

```
<!DOCTYPE html>
<html>
<head>
<title>
Liver patient analysis
</title>
<meta  name="viewport"  content="width=device-width,  initial-
scale=1">
<style>
body{
font-family: Calibri, Helvetica, sans-serif;
background-color: DarkRed;
}
.container {
padding: 300px;
padding-top: 25px;
background-color:DarkSalmon ;
}

input[type=number] {
```

```
  width: 100%;
  padding: 15px;
margin: 5px 0 22px 0;
display: block;
 border: none;
 background: #f1f1f1;
}
input[type=number]:focus {
background-color: orange;
outline: none;
}
 div {
         padding: 10px 0;

}
.predictbtn {
  background-color: #20B2AA;
  color: white;
  padding: 15px;
  margin: 5px 0 22px 0;
  border: center;
  cursor: pointer;
  width: 104%;
  opacity: 0.9;
}
.predictbtn:hover {
opacity: 1;
}

</style>
</head>
<body>
<div class="container">
<center>  <h1> LIVER HEALTH MONITOR</h1> </center>
<form action="home.html" method="post">
<label> Age </label>
<input  type="number"  step=0.01  name="Age"  placeholder=  "Age"
size="15" required />

<label> Gender: </label>
<input       type="number"       step=0.01       name="Gender"
placeholder="Gender    (0    for    male,    1    for    female)"
size="15"required />

<label>
Total_bilirubin :
</label>
<input    type="number"    step=0.01    name="Total_bilirubin"
placeholder="Total_bilirubin" size="10" required>

<label>
```

```
Direct_bilirubin :
</label>
<input    type="number"    step=0.01    name="Direct_bilirubin"
placeholder="Direct_bilirubin" size="10" required>

<label>
Alkaline_Phosphotase :
</label>
<input   type="number"   step=0.01   name="Alkaline_Phosphotase"
placeholder="Alkaline_Phosphotase" size="10" required>

<label>
Alamine_aminotransferase :
</label>
<input type="number" step=0.01 name="Alamine_aminotransferase"
placeholder="Alamine_aminotransferase" size="10" required>


<label>
Aspartate_aminotransferase :
</label>
<input                  type="number"                  step=0.01
name="Aspartate_aminotransferase"
placeholder="Aspartate_aminotransferase" size="10" required>

<label>
Total_proteins :
</label>
<input    type="number"    step=0.01    name="Total_proteins"
placeholder="Total_proteins" size="10" required>

<label>
Albumin :
</label>
<input       type="number"       step=0.01       name="Albumin"
placeholder="Albumin" size="10" required>

<label>
Albumin_and_Globulin_Ratio :
</label>
<input                  type="number"                  step=0.01
name="Albumin_and_Globulin_Ratio"
placeholder="Albumin_and_Globulin_Ratio" size="10" required>
<button type="submit" class="predictbtn">Predict</button>

</form>

<br><h4 align="center"<b>{{pred}}</b></h4>
<br>
</div>
</body>
```

Web Design using HTML



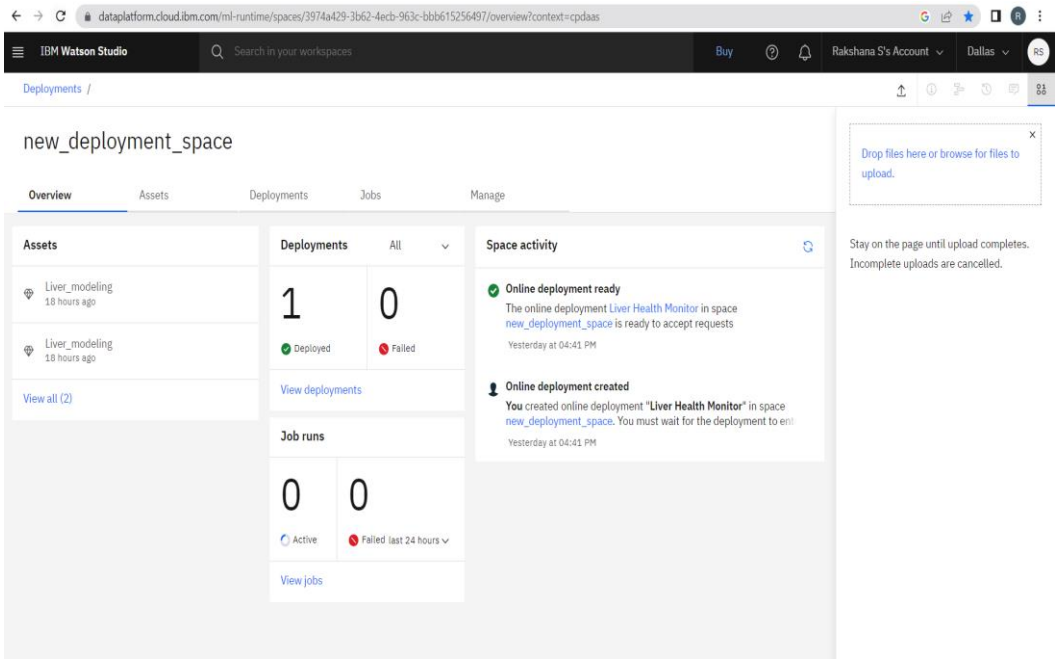
Build a Python code:

```
from flask import Flask, render_template,request
import pickle

app = Flask(__name__)

@app.route('/')
def bot():
    return render_template('home.html')

@app.route('/predict',methods=["POST"])

def predict():
    Age=request.form['Age']
    gender=request.form['Gender']
    tb=request.form['Total_bilirubin']
    db=request.form['Direct_bilirubin']
    ap=request.form['Alkaline_Phosphotase']
    aa1=request.form['Alamine_aminotransferase']
    aa2=request.form['Aspartate_aminotransferase']
    tp=request.form['Total_proteins']
    a=request.form['Albumin']
    agr=request.form['Albumin_and_Globulin_Ratio']


data=[[float(Age),float(gender),float(tb),float(db),float(ap),
float(aa1),float(aa2),float(tp),float(a),float(agr)]]
    model=pickle.load(open('liver_analysis.pkl','rb'))
    print(data)
    prediction=model.predict(data)

    if (prediction==1):
        output="You have liver disease."
    else:
        output="You do not have liver disease"
    return render_template('home.html',prediction_text=output)

if __name__=='__main__':
    app.run(debug=True)
```

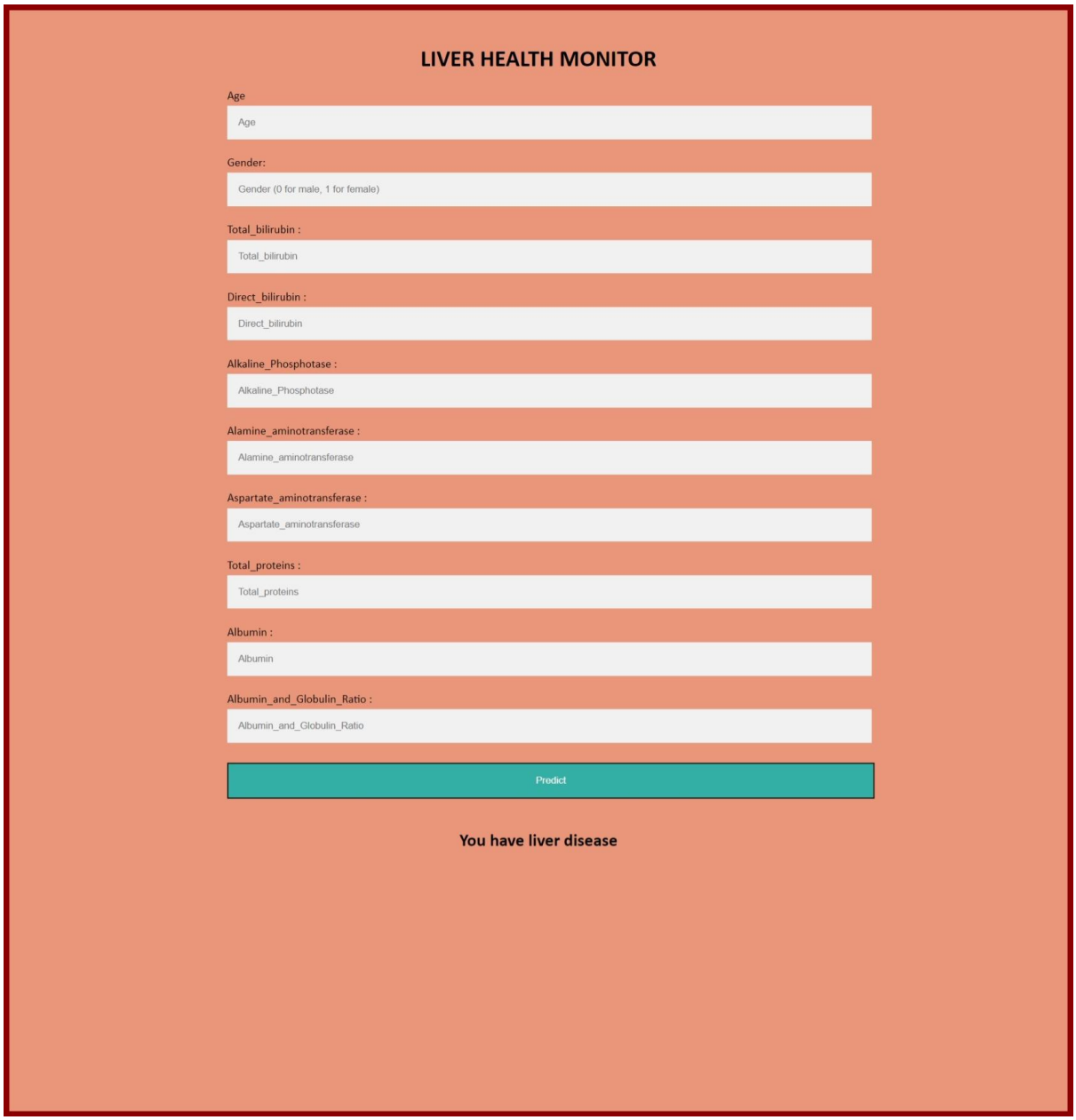
4.5 Train the model on IBM

To build a Machine Learning Model and deploy it on the IBM Cloud

```
!pip install ibm_watson_machine_learning
from ibm_watson_machine_learning import APIClient
wml_credentials={"url":"https://us-
south.ml.cloud.ibm.com","apikey":"eJqAgq9PC4DNCsDnTj0pfeO4-
Rk0jXwNWF61-LDwFYDL"}
client=APIClient(wml_credentials)
def guid_from_space_name(client,space_name):
    space=client.spaces.get_details()
    return(next(item for item in space['resources'] if
item['entity']["name"]==space_name)['metadata']['id'])
space_uid=guid_from_space_name(client,'new_deployment_space')
print("Space UID ="+space_uid)
client.set.default_space(space_uid)
client.software_specifications.list()
software_spec_uid=client.software_specifications.get_uid_by_na
me("runtime-22.1-py3.9")
software_spec_uid
model_details =
client.repository.store_model(model=svm,meta_props={
    client.repository.ModelMetaNames.NAME:"Liver_modeling",
    client.repository.ModelMetaNames.TYPE:"scikit-learn_1.0",

client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_sp
ec_uid})

model_id = client.repository.get_model_uid(model_details)
```



5. RESULTS



6. FUTURE SCOPE

Today almost everybody above the age of 12 years has smartphones 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.

The use of fast datasets technique like Apache Hadoop or Spark can be incorporated with this technique. In addition to this, we can use distributed refined algorithms like Forest Tree implemented in Apache Hadoop to increase scalability and efficiency.

7. CONCLUSION

Liver disease is one such critical disease that can be treated when detected early and it will become a life threating condition when undetected . There are many approaches and method for predicting liver disease like logistic regression Decision Tree, Random Forest, AdaBoost, k-Nearest Neighbour, convolution neural network, Multilayer perceptron and Support Vector Machine. Out of which we compared the accuracy of SVM ,KNN and random forest methods using various packages such as NumPy, Matplotlib, Pandas, Seaborn and so and resulted that Support Vector Machine has more accuracy compared to others. Also the proposed is made into an user interface app for well being of people.

APPENDIX

Demo link

https://vimeo.com/772712483

Github project page link

https://github.com/IBM-EPBL/IBM-Project-19384-1659697109