Project Report

Statistical Machine Learning Approaches to Liver Disease Prediction

Team ID : PNT2022TMID17773

Team members: B.Swetha , Balamurugan Ramya , M.Pushpa Lathaa, J.Jayasri.

1.INTRODUCTION

1.1 Project Overview

Many people are affected by liver disease due to large amount of alcohol consumption and also because of some other reasons. So this system is used to predict the liver disease using machine learning techniques which helps the doctors for earlier diagnosis

1.2 Purpose

- To predict liver disease at an earlier stage itself
- To build a web application to gather information from user for prediction of liver disease

2. LITERATURE SURVEY

2.1 Existing problem

Detection of liver disease by doctors is a time consuming and tedious process. It is also expensive. So detection using prediction system is essential for use for patients as well as doctors .

2.2 References

- Somaya Hashem a,e, Mahmoud ElHefnawi a,e, Shahira Habashy b, Mohamed El-Adawy b,GamalEsmat c "**Machine Learning Prediction Models for Diagnosing Hepatocellular Carcinoma with HCV-related Chronic Liver Disease**" Computer Methods and Programs in Biomedicine ,Volume 196 ,2020 worked on a dataset consists of 4423 patients details for prediction of HCC presence using classification and regression tree, alternating decision tree, reduce pruning error tree and linear regression algorithm .
- Rayyan AzamKhan ,YigangLuo ,Fang-XiangWu "**Machine learning based liver disease diagnosis: A systematic review" Neurocomputing, Volume468, 2022** investigated the potential of CAD system for detection of liver disease using the image acquisition modalities and machine learning algorithms like SVM, KNN , Neural Network.
- Varun Vats, Lining Zhang, Sreejit Chatterjee, Sabbir Ahmed, Elvin Enziama and Kemal Tepe **"A Comparative Analysis of Unsupervised Machine Techniques for Liver Disease Prediction**" **2018** proposed a paper by comparing the unsupervised machine learning techniques such as K-Means, Affinity propogation, and DBSCAN to predict liver disease. Among the three algorithms K-Means is proved as an optimal method for liver disease prediction by using Silhouette Coefficient. The algorithms are applied on a medical dataset containing liver disease related data.
- Golmei Shaheamlung, Harshpreet Kaur, Mandeep Kaur "**A Survey on machine learning techniques for the diagnosis of liver disease" 2020** has discussed about the machine learning techniques used for liver disease prediction by various authors previously. The common machine learning techniques used for liver disease prediction are SVM, KNN, K-Means, Decision tree, and Neural network. All these are measured by analysis methods like accuracy, sensitivity, specificity, and precision. Different algorithm has different performance based on different scenarios. So a hybrid machine learning model can improve the performance and accuracy.
- Satessh Ambesange, Vijayalaxmi A, Rashmi Uppin, Shruthi Patil, Vilaskumar Patil "**Optimizing Liver disease prediction with Random Forest by various Data balancing Techniques**" worked on Random Forest (RF) algorithm to predict the disease with different preprocessing techniques. Data set is checked for skewness, outliers and imbalance using univariate and bivariate analysis and then suitable algorithms used to remove outliers and various oversampling and under sampling techniques are used to balance the data.

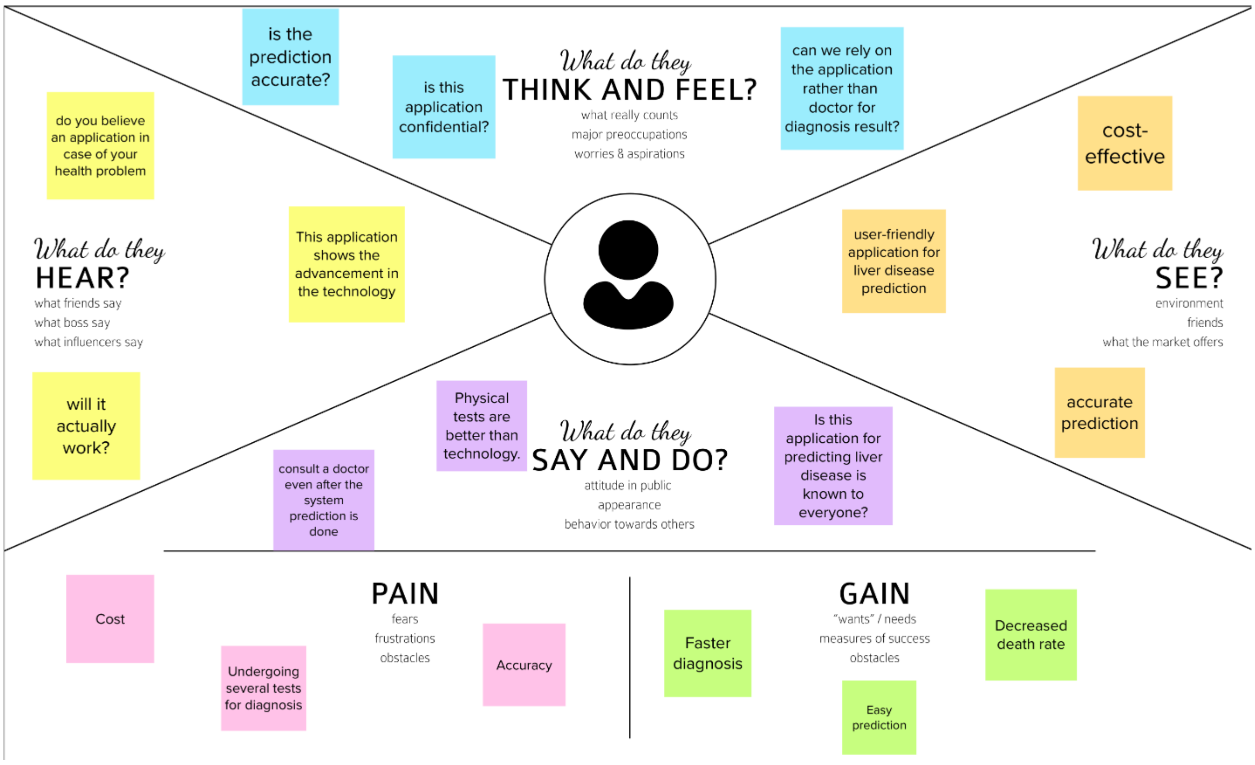
- Sateesh Ambesange, Ranjana Nadagoudar, Rashmi Uppin, Vilaskumar Patil, Shruti " **Liver Diseases Prediction using KNN with Hyper Parameter Tuning Techniques**" worked on prediction of liver disease by Machine learning based model trained with the dataset. Feature analysis ,transformation techniques have been used to transform the data .Here K-Nearest Neighbor model is used to diagnose and predict liver disease. Grid Search is used for tuning the model's hyper parameters.
- Rakshith D B Mrigank Srivastava Ashwani Kumar Gururaj S P" **Liver Disease Prediction System using Machine Learning Techniques"(2021)** worked with a diagnosis of liver disease using various machine learning models. Machine Learning models such as Naïve Bayes, Artificial Neural Network , KNN are used to predict the liver disease by using different attributes .
- Sivasangari "**Diagnosis of Liver Disease using Machine Learning Models "(2020)** worked with a diagnosis of liver disease using various machine learning models. Three machine learning models namely Support Vector Machine, Decision Tree and Random Forest is used to predict the liver disease by using different attributes .

2.3 Problem Statement Definition

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	a Patient	know whether I am affected by liver disease.	it is a hard, costly, time consuming process	the process undergoes many tests	tensed and bad.
PS-2	a Doctor	know whether the patient has the liver disease at earlier stage	it takes long time for me to diagnose	the process undergoes many tests	difficult to cure my patient at an earlier stage.

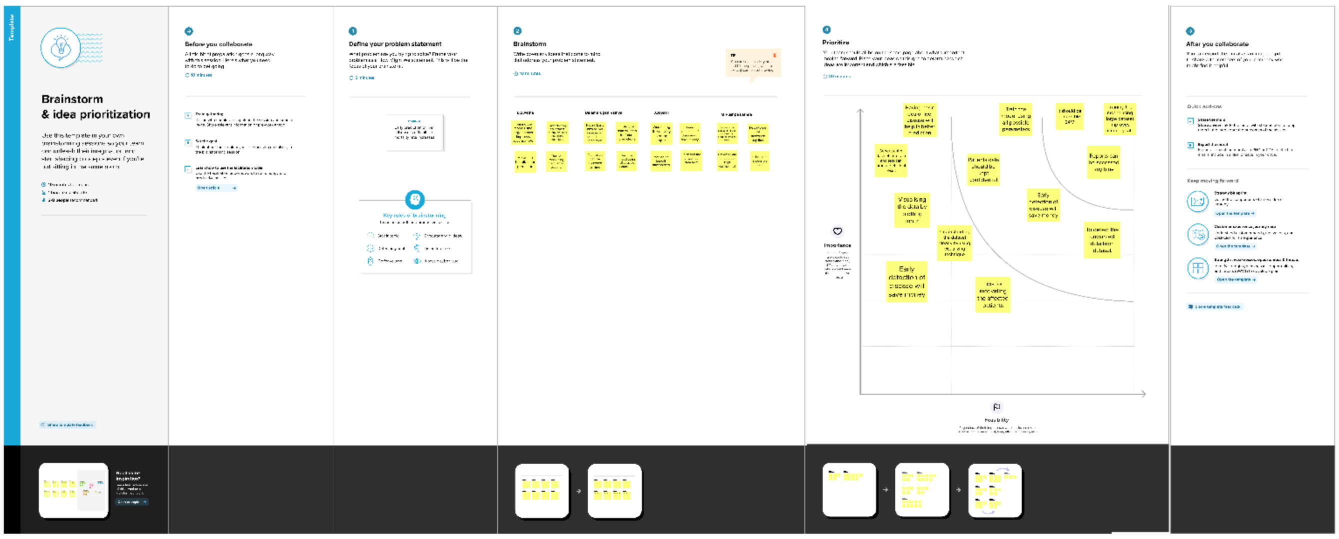
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



is the prediction accurate?
is this application confidential?
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
can we rely on the application rather than doctor for diagnosis result?
do you believe an application in case of your health problem
cost-effective
This application shows the advancement in the technology
user-friendly application for liver disease prediction
What do they
HEAR?
what friends say
what boss say
what influencers say
What do they
SEE?
environment
friends
what the market offers
will it actually work?
Physical tests are better than technology.
accurate prediction
consult a doctor even after the system prediction is done
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Is this application for predicting liver disease is known to everyone?
Cost
PAIN
fears
frustrations
obstacles
Undergoing several tests for diagnosis
Accuracy
Faster diagnosis
GAIN
"wants" / needs
measures of success
obstacles
Decreased death rate
Easy prediction

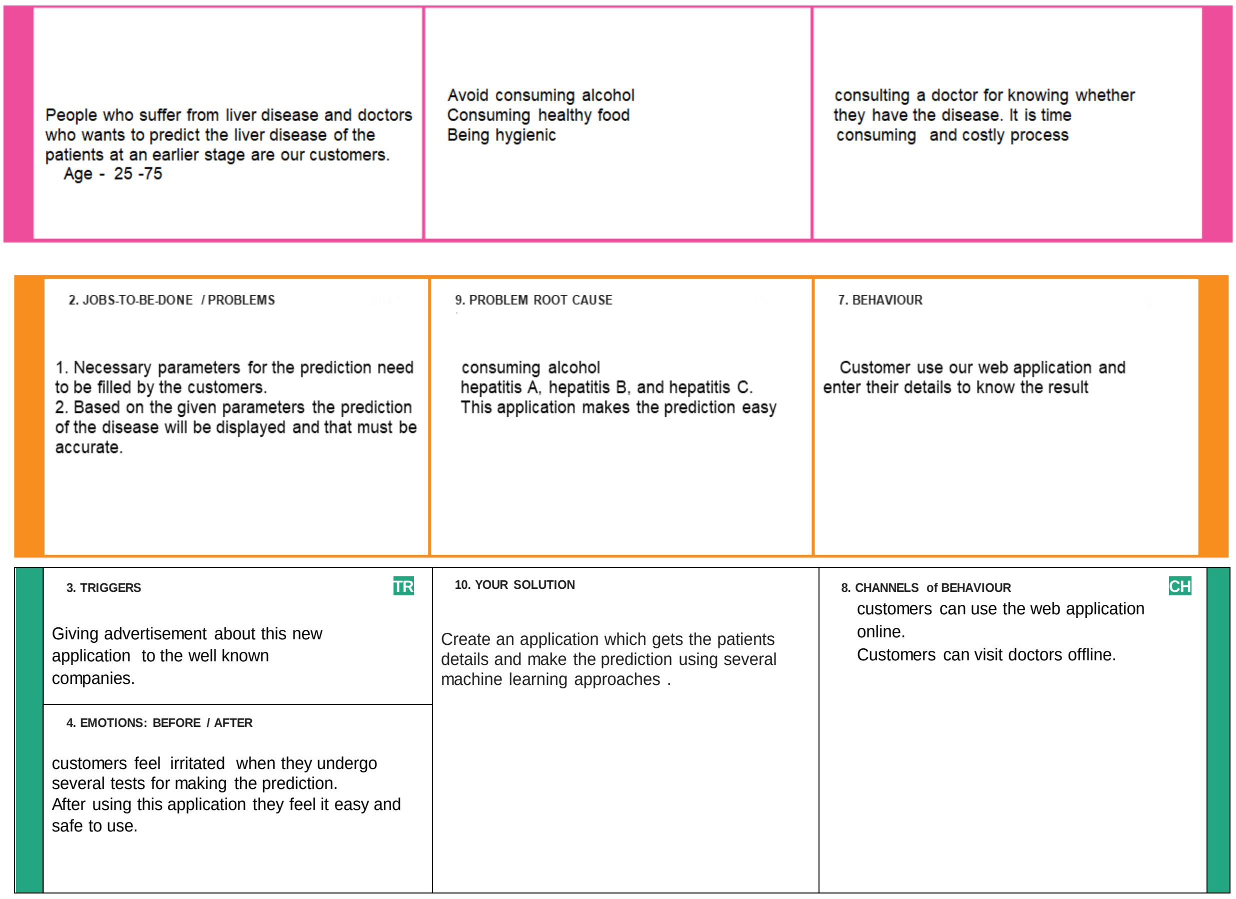
3.2 Ideation & Brainstorming



3.3 Proposed Solution

S .No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Many people are affected by liver diseases due to large amount of alcohol consumption. Doctors also find it difficult to predict the disease at an early stage which leads to increased mortality rate. So earlier prediction of liver disease using machine learning techniques helps the doctors for diagnosis.
2.	Idea / Solution description	The idea is to use machine learning algorithm that analyses the parameters of the patient and predict the liver disease.
3.	Novelty / Uniqueness	This system predicts the accurate result and also gives mental tips for motivating the affected patients.
4.	Social Impact / Customer Satisfaction	The customer is satisfied in using the user friendly application at low cost.
5.	Business Model (Revenue Model)	This model can be used at the hospitals for the earlier prediction of liver disease.
6.	Scalability of the Solution	The system fits into any health sector application and can withstand increased work loads .

3.4 Problem Solution fit



People who suffer from liver disease and doctors who wants to predict the liver disease of the patients at an earlier stage are our customers.
Age - 25 -75

Avoid consuming alcohol
Consuming healthy food
Being hygienic

consulting a doctor for knowing whether they have the disease. It is time consuming and costly process

2. JOBS-TO-BE-DONE / PROBLEMS

1. Necessary parameters for the prediction need to be filled by the customers.
2. Based on the given parameters the prediction of the disease will be displayed and that must be accurate.

9. PROBLEM ROOT CAUSE

consuming alcohol
hepatitis A, hepatitis B, and hepatitis C.
This application makes the prediction easy

7. BEHAVIOUR

Customer use our web application and enter their details to know the result

3. TRIGGERS

TR

Giving advertisement about this new application to the well known companies.

4. EMOTIONS: BEFORE / AFTER

customers feel irritated when they undergo several tests for making the prediction.
After using this application they feel it easy and safe to use.

10. YOUR SOLUTION

Create an application which gets the patients details and make the prediction using several machine learning approaches .

8. CHANNELS of BEHAVIOUR

CH

customers can use the web application online.
Customers can visit doctors offline.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

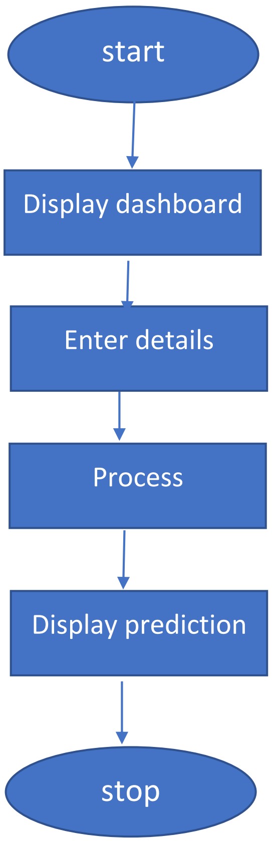
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User details	This system gets the information about the users
FR-2	Blood test details	This system gets the blood test report details for making the prediction
FR-3	Make prediction	This system makes the prediction using various machir learning techniques
FR-4	View prediction	This system will display the predicted output to the us

4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	UI is provided in this application that is used to fill details about the blood test of the patients
NFR-2	**Security**	The details of the patients are maintained confidential
NFR-3	**Reliability**	The software does not fail during usage and the prediction made is accurate
NFR-4	**Performance**	The web page is loaded without any delay irrespective of various user request
NFR-5	**Availability**	This application is available 24/7
NFR-6	**Scalability**	This application can handle any number of user request simultaneously.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

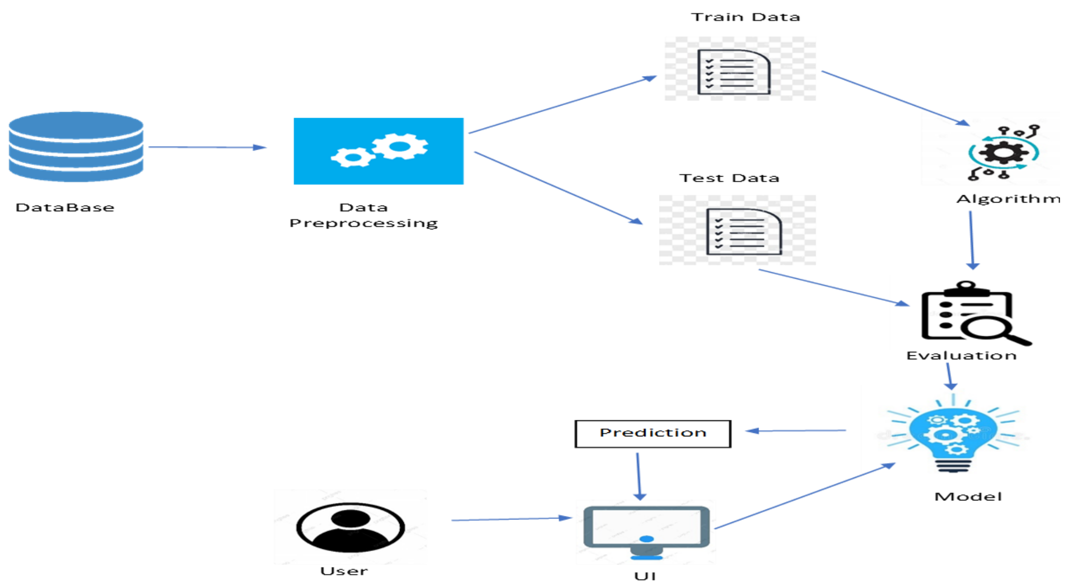


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interface receives input from user and predict the result.	HTML, CSS, JavaScript / Angular Js / React Js etc.
	Application Logic-1	User will upload the input feature by using UI	HTML,Python-Flask
	Application Logic-2	UI input the data to trained model.	Python
2.	Application Logic-3	Model will predict whether the person have liver disease or not and display using UI.	Python
3.	Machine Learning Model	ML Model are implemented and used for classification.	SVM,KNN,Decision Tree model
4.	Cloud Database	Deploying the model in cloud	IBM cloud
5.	File Storage	To store data in hierarchical structure .	Local Filesystem

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Python
2.	Security Implementations	User data are not stored in server so there is no security issuse.	-
3.	Availability	Application will be available for 24/7	Load Balancer
4.	Performance	Application can handle any number of users	-

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	User details	USN-1	As a user , I will give my details to the system		Low	Sprint-1
	Blood test details	USN-2	As a user , I will give my blood test details to the system		High	Sprint-1
	Make prediction	USN-3	As a user , I can request the system to make the prediction	The prediction will be in progress	High	Sprint-2
	View prediction	USN-4	As a user , I can view the prediction made by the system		High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

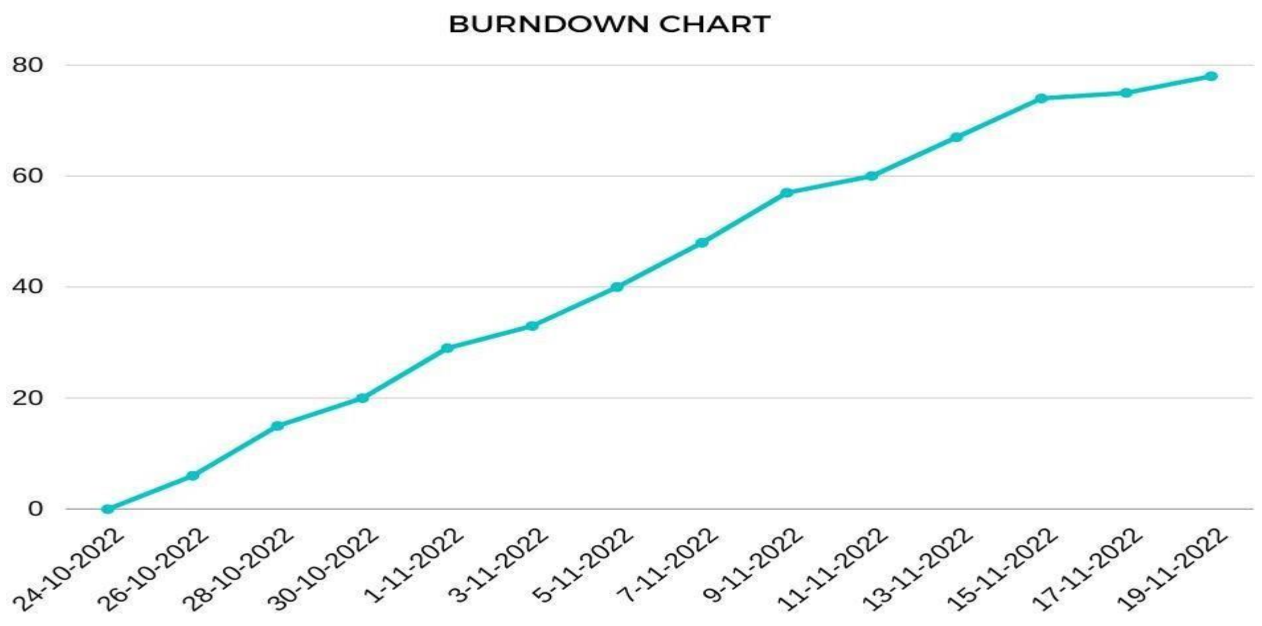
6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Pre-Processing	USN-1	Removing rows having null values, converting string values to numeric value.	10	Easy	B.Swetha M.Pushpa Lathaa J.JayaSri Balamurugan Ramya.
Sprint-1		USN-2	Data visualization	10	Easy	B.Swetha M.Pushpa Lathaa J.JayaSri Balamurugan Ramya.
Sprint-2	Model building	USN-3	Developing machine learning models for liver disease prediction.	10	High	B.Swetha M.Pushpa Lathaa J.JayaSri Balamurugan Ramya.
Sprint-2		USN-4	Improving accuracy of the built model	10	Medium	B.Swetha M.Pushpa Lathaa J.JayaSri Balamurugan Ramya.
Sprint-3	Integrating model with html page	USN-5	Designing html page for getting the user inputs for making the prediction	5	Easy	B.Swetha M.Pushpa Lathaa J.JayaSri Balamurugan Ramya.

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	18	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	19	20 Nov 2022

6.3 Reports from JIRA



7.CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

Login page – The user can login to the site.

7.2 Feature 2

User can enter details and predict liver disease.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Patient Liver Prediction</title>
                        <style>
        body{
                                        background-repeat: no-repeat;
                                        background-attachment: fixed;
                                        background-size: 100% 100%;
                                        background-image: url({{ url_for('static', filename='fatty-
liver.jpg') }});
                                }
                                #t1{
                                        color: red;
                                        background-color: white;
                                }

                        </style>
</head>
<body>
                        <table style="margin-left:auto;margin-right:auto;margin-top: auto;margin-
bottom: auto;border: 5px solid black;" id="t1">
                                <tr>
                        <td>
                                <h2><p>Your test result is</p></h2>
                        </td>
                </tr>
                <tr>
                        <td>
                                <h1>{{res}}</h1>
                        </td>
                </tr>

                <tr>
                        <td>
                        {% if res==1 %}
                                <div id="d1"><h4><!-- 🙁<br>--><br>You have LIVER
DISEASE <br><br>Please Consult a Doctor.</h4></div>
                                <!--<img class="gif" src="{{ url_for('static',
filename='dr.gif')}}" alt="LIVER Image">-->
                        {% elif res==0 %}
                                <div id="d1"><h4><!--🤩 Congratulation! 🤩<br>--
><br>You DON'T have LIVER DISEASE.</h4></div>
                                <!--<img class="gif1" src="{{ url_for('static',
filename='yes.webp')}}" alt="Not LIVER Image">-->
                        {% endif %}
                        </td>

</tr>
                <tr>
                        <td><a href="/"> Go back </a></td>
                </tr>
            </table>
</body>
</html>
```

8.TESTING

8.1 Test Cases

Date	3-Nov-22
Team ID	PNT2022TMID17773
Project Name	Project - Statistical Machine Learning Approaches to Liver Disease Prediction
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Verify login/Singup popup displayed or not		Login/Signup popup should display	Working as expected	Pass				
HomePage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 3.Verify login/Singup popup with below UI elements:		Application should show the homepage	Working as expected	pass			BUG-1234	
PredictionPageTC_OO3	Functional	Main page	User will enter the [arameteres for prediction		Enter the parameter such as age gender,blood test details .	65 Female 0.7 0.1 187 16 18 6.8 3.3 0.9 1	application should navigate to result page and user should get the result after entering the predict button	works	pass				
PositiveResultPage_TC_OO4	Functional	Result page	displays the result as positive as predicted		click the predict button in the previous page to get the result in this page		page should display the positive result for the patients	works	pass				
NegativeResultPage	Functional	Result page	displays the result as positive as predicted		click the predict button in the previous page to get the result in this page		page should display the negative result for the patients	works	pass				

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20

Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3. Test Case Analysis

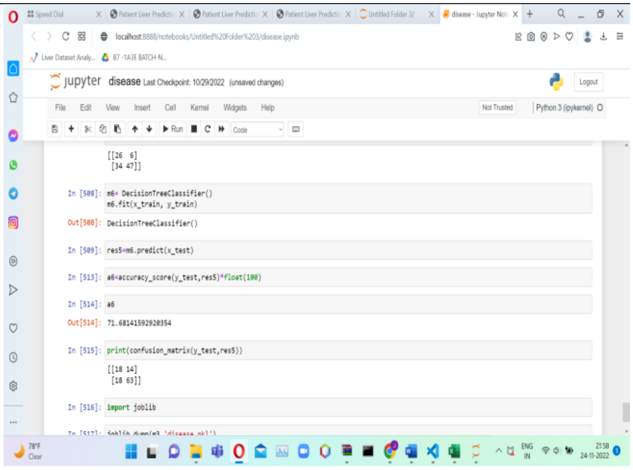
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

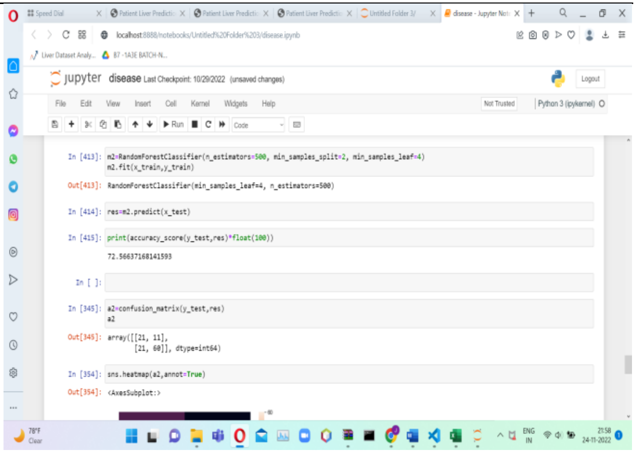
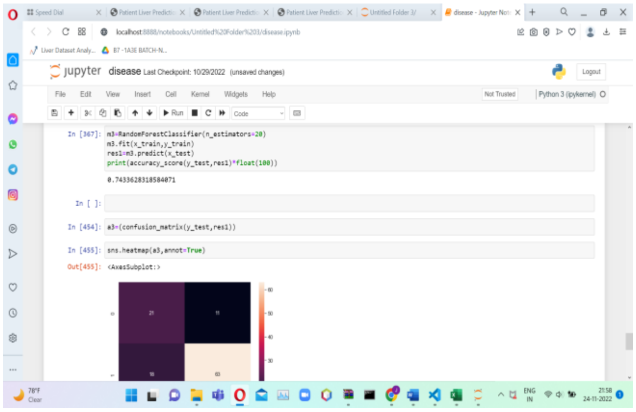
9.RESULTS

9.1 Performancemetrics

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	

```
[[26  6]
 [34 47]]

In [508]: m6= DecisionTreeClassifier()
          m6.fit(x_train, y_train)

Out[508]: DecisionTreeClassifier()

In [509]: res5=m6.predict(x_test)

In [513]: a6=accuracy_score(y_test,res5)*float(100)

In [514]: a6

Out[514]: 71.68141592920354

In [515]: print(confusion_matrix(y_test,res5))

          [[18 14]
           [18 63]]

In [516]: import joblib
```

S.No.	Parameter	Values	Screenshot
1.	Tune the Model	Hyperparameter Tuning - Validation Method -	 

```
In [413]: m2=RandomForestClassifier(n_estimators=500, min_samples_split=2, min_samples_leaf=4)
          m2.fit(x_train,y_train)

Out[413]: RandomForestClassifier(min_samples_leaf=4, n_estimators=500)

In [414]: res=m2.predict(x_test)

In [415]: print(accuracy_score(y_test,res)*float(100))

          72.56637168141593

In [ ]:

In [345]: a2=confusion_matrix(y_test,res)
          a2

Out[345]: array([[21, 11],
                 [21, 60]], dtype=int64)

In [354]: sns.heatmap(a2,annot=True)

Out[354]: <AxesSubplot:>
```

```
In [367]: m3=RandomForestClassifier(n_estimators=20)
          m3.fit(x_train,y_train)
          res1=m3.predict(x_test)
          print(accuracy_score(y_test,res1)*float(100))

          0.7433628318584071

In [ ]:

In [454]: a3=(confusion_matrix(y_test,res1))

In [455]: sns.heatmap(a3,annot=True)

Out[455]: <AxesSubplot:>
```

10 ADVANTAGES

- Less time consumption for making the liver disease prediction
- Easy process that any user can follow
- Less Expensive when compared to lab tests
- Early prediction helps in reducing the death rate

11 CONCLUSION

This system helps the users (patients and doctors) for prediction of liver disease at an earlier stage .The user can enter the required parameters in the website for prediction. The result will be displayed on the screen within few seconds

12 FUTURE SCOPE

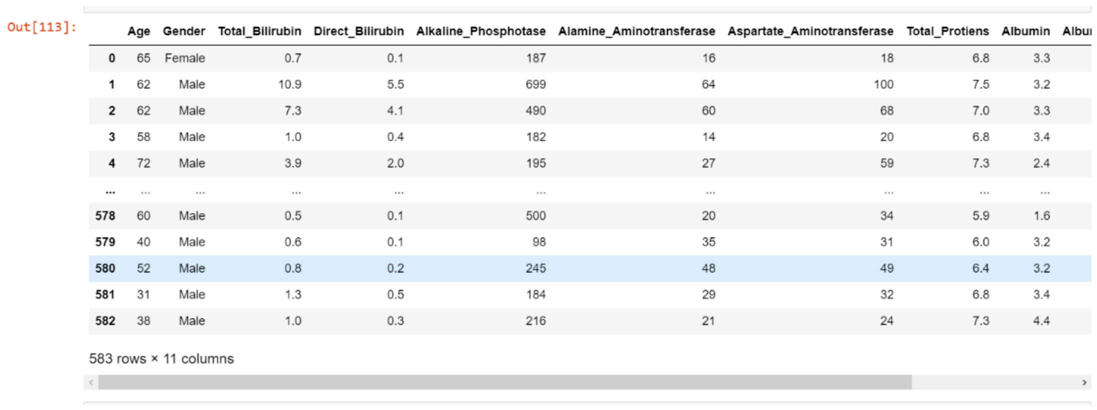
Further the system can enhanced with more feature and made available to user through online ,that can be used to predict the liver disease .

13 APPENDIX

Source code

```
import numpy as np
# for dataframes
import pandas as pd
# for easier visualization
import seaborn as sns
# for visualization and to display plots
from matplotlib import pyplot as plt
# import color maps
from matplotlib.colors import ListedColormap
```

```
df=pd.read_csv('indian_liver_patient.csv')
df
```



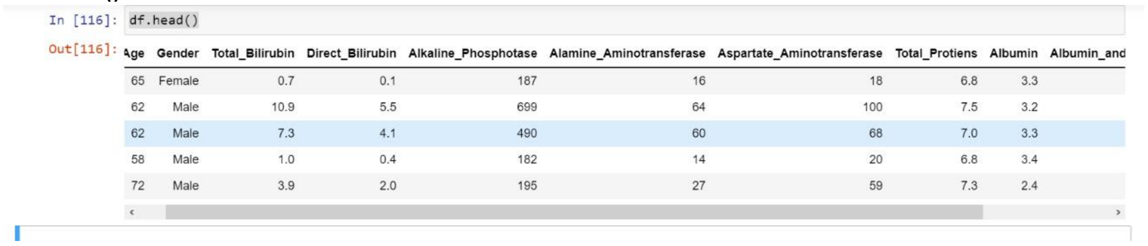
Out[113]:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albu
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	
...	...	...	...	...	...	...	...	...	...	
578	60	Male	0.5	0.1	500	20	34	5.9	1.6	
579	40	Male	0.6	0.1	98	35	31	6.0	3.2	
580	52	Male	0.8	0.2	245	48	49	6.4	3.2	
581	31	Male	1.3	0.5	184	29	32	6.8	3.4	
582	38	Male	1.0	0.3	216	21	24	7.3	4.4	

583 rows × 11 columns

```
df.shape
(583, 11)
df.columns
Index(['Age', 'Gender', 'Total_Bilirubin', 'Direct_Bilirubin',
       'Alkaline_Phosphotase', 'Alamine_Aminotransferase',
       'Aspartate_Aminotransferase', 'Total_Protiens', 'Albumin',
        'Albumin_and_Globulin_Ratio', 'Dataset'],
      dtype='object')
```

df.head()



```
In [116]: df.head()
```

Out[116]:

Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and
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	7.0	3.3	
58	Male	1.0	0.4	182	14	20	6.8	3.4	
72	Male	3.9	2.0	195	27	59	7.3	2.4	

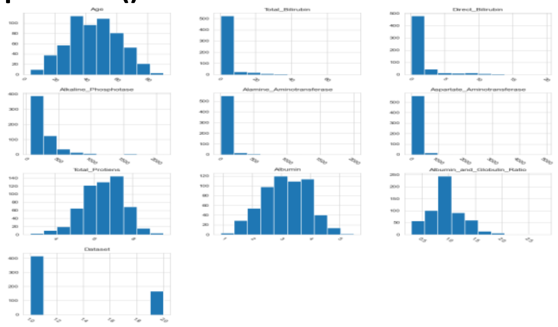
Exploratory analysis

filtering categorical data

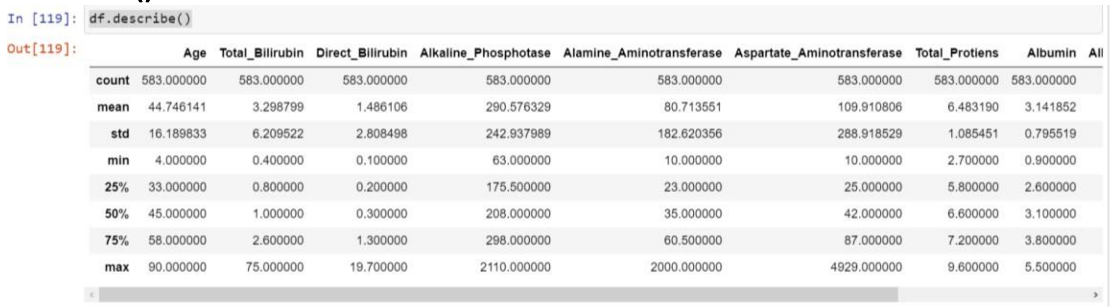
df.dtypes[df.dtypes=='object']

Distribution of Numerical Features

df.hist(figsize=(15,15), xrot=-45, bins=10)
plt.show()



df.describe()

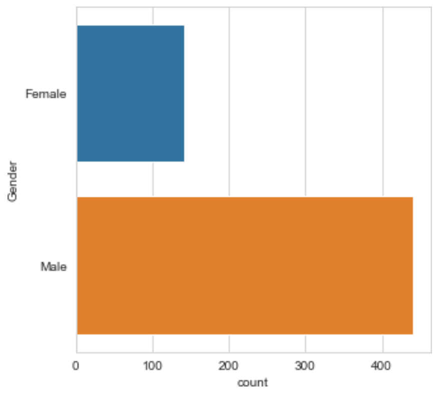


```
In [119]: df.describe()
Out[119]:
```

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Al
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	
mean	44.746141	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.141852	
std	16.189833	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.795519	
min	4.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.900000	
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.600000	
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.100000	
75%	58.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.800000	
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.500000	

Dataset i.e output value has '1' for liver disease and '2' for no liver disease so let's make it 0 for no disease to make it convineint

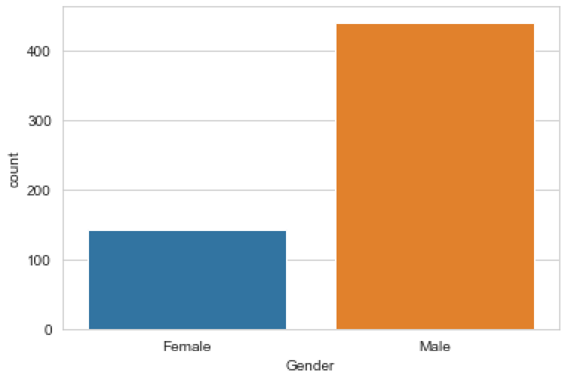
```
def partition(x):
    if x == 2:
        return 0
    return 1

df['Dataset'] = df['Dataset'].map(partition)
plt.figure(figsize=(5,5))
sns.countplot(y='Gender', data=df)
```



sns.countplot(data=df, x = 'Gender', label='Count')

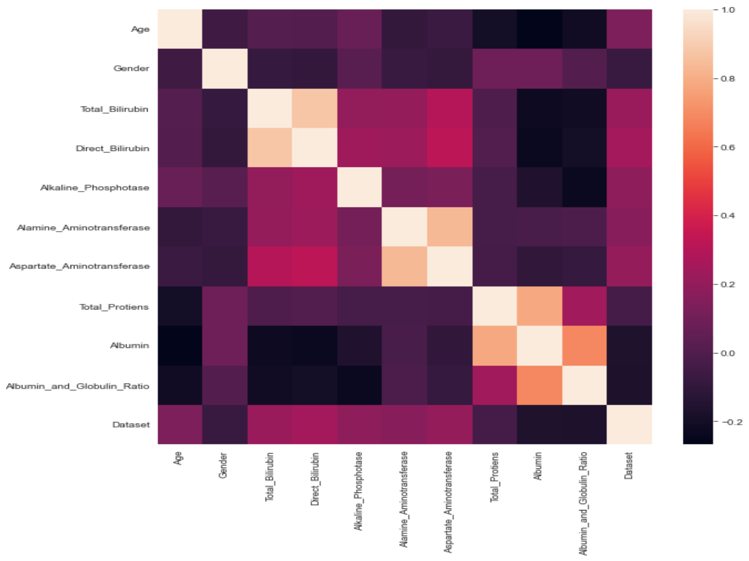
M, F = df['Gender'].value_counts()
print('Number of patients that are male: ',M)
print('Number of patients that are female: ',F)

```
Number of patients that are male:  441
Number of patients that are female:  142
```



Label Male as 0 and Female as 1

```
def partition(x):
   if x =='Male':
       return 0
    return 1
df['Gender'] = df['Gender'].map(partition)
df.corr()
plt.figure(figsize=(10,10))
sns.heatmap(df.corr())
```



Data cleaning

```
df = df.drop_duplicates()
print( df.shape )
```

```
(564, 11)
```

Removing outliers

```
df = df[df.Aspartate_Aminotransferase <=3000 ]
df.shape
```

```
(564, 11)
```

```
df = df[df.Aspartate_Aminotransferase <=2500 ]
df.shape
```

```
(564, 11)
```

Dropping null values

```
df=df.dropna(how='any')
df.shape
```

```
(564, 11)
```

Model Building:

```
y = df.Dataset
x = df.drop('Dataset', axis=1)
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.2,
                                                    random_state=42,
                                                   )
!pip install imblearn
from imblearn.over_sampling import SMOTE
smote=SMOTE()
x_train,y_train=smote.fit_resample(x_train,y_train)
```

K-NEIGHBORS:

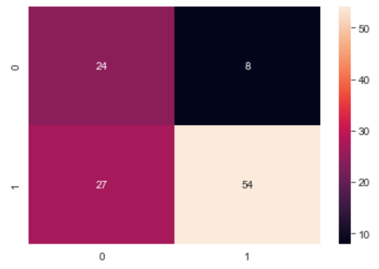
```
m1=KNeighborsClassifier(n_neighbors=21)
m1.fit(x_train,y_train)
ans=m1.predict(x_test)
print(accuracy_score(y_test,ans)*float(100))
```

```
69.02654867256636
```

```
a1=confusion_matrix(y_test,ans)
a1
```

```
array([[24,  8],
       [27, 54]], dtype=int64)
```

```
sns.heatmap(a1,annot=True)
```

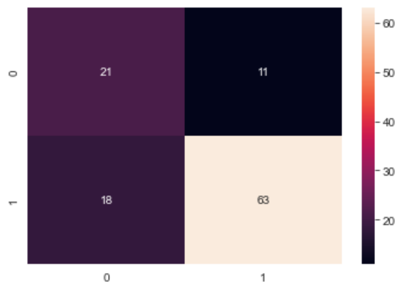


RANDOM FOREST:

```
m3=RandomForestClassifier(n_estimators=20)
m3.fit(x_train,y_train)
res1=m3.predict(x_test)
print(accuracy_score(y_test,res1)*float(100))
```

```
0.7433628318584071
```

```
a3=(confusion_matrix(y_test,res1))
sns.heatmap(a3,annot=True)
```



SVM:

```
m4=SVC(C=1, gamma=1)
m4.fit(x_train, y_train)
res2=m4.predict(x_test)
print(accuracy_score(y_test,res2)*float(100))
71.68141592920354
print(confusion_matrix(y_test,res2))
[[ 0 32]
 [ 0 81]]
```

LOGISTIC REGRESSION:

```
m5=LogisticRegression()
m5.fit(x_train,y_train)
res4=m5.predict(x_test)
a5=accuracy_score(y_test,res4)
a5
0.6460176991150443
print(confusion_matrix(y_test,res4))
[[26  6]
 [34 47]]
```

DECISION TREE:

```
m6= DecisionTreeClassifier()
m6.fit(x_train, y_train)
res5=m6.predict(x_test)
a6=accuracy_score(y_test,res5)*float(100)
```

```
a6
71.68141592920354
```

```
print(confusion_matrix(y_test,res5))
[[18 14]
 [18 63]]
```

SAVING MODEL

```
import joblib
joblib.dump(m3,'disease.pkl')
['disease.pkl']
```

HTML FILES:

INDEX PAGE :

```
<!DOCTYPE html>
<html>
                <head>
                    <title>Patient Liver Prediction</title>
                    <style>
      body{
                                background-repeat: no-repeat;
                                background-attachment: fixed;
                                background-size: 100% 100%;
                                background-image: url({{ url_for('static', filename='fatty-
liver.jpg') }});
                        }
                        #t1{
                                background-color: aliceblue;
                                color: black;
                        }
                        #h{
                                background-color: slateblue;
                                text-align: center;
                        }
                        #gen{
                                opacity: .5;
                        }
                        #b1 {
                                color: black;
                                text-align: center;
                                display: inline-block;
```

```
                                        font-size: 16px;
                                        margin: 4px 2px;
                                }
                        </style>
                </head>
                <body>
                        <h1 id="h">Patient Liver Prediction</h1>
                        <form method="POST" action="/predict">
                                <table style="margin-left:auto;margin-right:auto;" id="t1

                                         <tr>
                                                <td><label>Age : </label></td>
                                                <td><input type="text" name="age"><br></td>
                                        </tr>
                                        <tr>
                                                <td><label>Gender : </label></td>
                                <td><input type="text" name="gen" placeholder="Enter 0 as
Male, 1 as Female"><br></td>
                                        </tr>
                                        <tr>
                                                <td><label>Total Bilirubin : </label></td>
                                                <td><input type="text" name="tbil"
required><br></td>
                                        </tr>
                                        <tr>
                                                <td><label>Direct Bilirubin : </label></td>
                                                <td><input type="text" name="dbil"
required><br></td>
                                        </tr>
                                        <tr>
                                                <td><label>Alkaline Phosphate : </label></td>
                                                <td><input type="text" name="alk"
required><br></td>
                                        </tr>
                                        <tr>
                                                <td><label>Alanine aminotransferase :
</label></td>
                                                <td><input type="text" name="ala"
required><br></td>
                                        </tr>
                                        <tr>
                                                <td><label>Aspartate aminotransferase :
</label></td>
                                                <td><input type="text" name="asp"
```

```
required><br></td>
                                                </tr>
                                                <tr>
                                                        <td><label>Total Protiens : </label></td>
                                                        <td><input type="text" name="tpro"
required><br></td>
                                                </tr>
                                                <tr>
                                                        <td><label>Albumin : </label></td>
                                                        <td><input type="text" name="alb"
required><br></td>
                                                </tr>
                                                <tr>
                                                        <td><label>Albumin and Globulin ratio :
</label></td>
                                                        <td><input type="text" name="albglo" required><
</tr>
                                                <tr>
                                                        <td><button class="button"
id="b1">Predict</button></td>
                                                </tr>
                                        </table>
                                </form>
                        </body>
</html>
```

PREDICTION PAGE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Patient Liver Prediction</title>
                        <style>
    body{
                                        background-repeat: no-repeat;
                                        background-attachment: fixed;
                                        background-size: 100% 100%;
                                        background-image: url({{ url_for('static', filename='fatty-liver.jpg')
}});
                                }
                                #t1{
                                        color: red;
                                        background-color: white;
                                }

                        </style>
</head>
<body>
                        <table style="margin-left:auto;margin-right:auto;margin-top: auto;margin-
bottom: auto;border: 5px solid black;" id="t1">
                                <tr>
                                        <td>
                                                <h2><p>Your test result is</p></h2>
                                        </td>
                                </tr>
                                <tr>
                                        <td>
                                                <h1>{{res}}</h1>
                                        </td>
                                </tr>

                                <tr>
                                        <td>
                                        {% if res==1 %}
                                                <div id="d1"><h4><!-- 🙁<br>--><br>You have LIVER
DISEASE <br><br>Please Consult a Doctor.</h4></div>
                                                <!--<img class="gif" src="{{ url_for('static',
filename='dr.gif')}}" alt="LIVER Image">-->
                                        {% elif res==0 %}
                                                <div id="d1"><h4><!--🤩 Congratulation! 🤩<br>--
><br>You DON'T have LIVER DISEASE.</h4></div>
                                                <!--<img class="gif1" src="{{ url_for('static',
filename='yes.webp')}}" alt="Not LIVER Image">-->
                                        {% endif %}
                                        </td>

</tr>
                                <tr>
                                        <td><a href="/"> Go back </a></td>
                                </tr>
                        </table>
</body>
</html>
```

FLASK FILE:

```
import flask
```

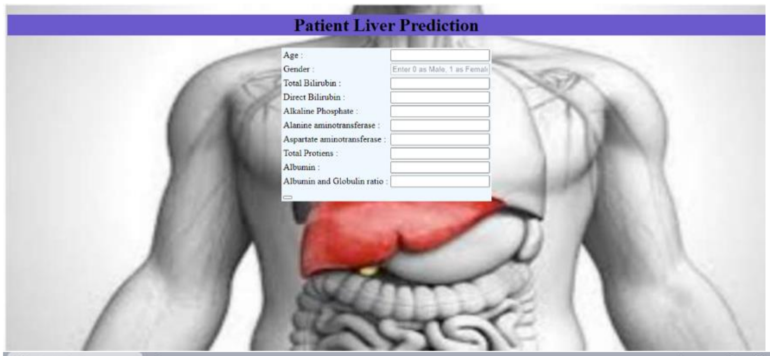
```
from flask import request, render_template, url_for
from flask_cors import CORS
import joblib

app = flask.Flask(__name__)
CORS(app)
@app.route('/', methods=['GET'])
def homePage():
   return render_template('index.html')


#@app.route('/favicon.ico')
#def favicon():
#    return url_for('static', filename='image/favicon.ico')


@app.route('/predict', methods=['POST'])
def predict():
   a = int(request.form['age'])
   g = int(request.form['gen'])
   tb = float(request.form['tbil'])
   db = float(request.form['dbil'])
   ap = int(request.form['alk'])
   ala = int(request.form['ala'])
   asa = int(request.form['asp'])
   tp = float(request.form['tpro'])
   alb = float(request.form['alb'])
   ag = float(request.form['albglo'])
   X = [[a, g, tb, db, ap, ala, asa, tp, alb, ag]]
   model = joblib.load('disease.pkl')
   disease = model.predict(X)[0]
   return render_template('predict.html',res=disease)

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

DEPLOYING IN CLOUD

```
import requests

import json
```

```
    # NOTE: you must manually set API_KEY below using information retrieved from your IBM
Cloud account.

    API_KEY = "kLomfc2FUoTTlDe3yRiXQ5pSsrEyYyDtdl8bkXLmoasX"

    token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":

     API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})

    mltoken = token_response.json()["access_token"]

    print("ml token",mltoken)



    header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}



    # NOTE: manually define and pass the array(s) of values to be scored in the next line

    payload_scoring = {"input_data": [{"field":
[["Age","Gender","Total_Bilirubin","Direct_Bilirubin","Alkaline_Phosphotase","Alamine_Ami
notransferase","Aspartate_Aminotransferase","Total_Protiens","Albumin","Albumin_and_Globu
lin_Ratio"]], "values": [[63,0,0.9,0.2,194,52,45,6,3.9,1.85]]}]}



    response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/8867b336-70c2-4ac8-baf3-
39b2a1ca913b/predictions?version=2022-11-21', json=payload_scoring,

     headers={'Authorization': 'Bearer ' + mltoken})

    print("Scoring response")

    predictions=response_scoring.json()

    pred=predictions['predictions'][0]['values'][0][0]

    if(pred==1):

        print("Liver disease")

    else:

        print("No liver disease")
```

OUTPUT:



Patient Liver Prediction

Age :

Gender : Enter 0 as Male, 1 as Femal

Total Bilirubin :

Direct Bilirubin :

Alkaline Phosphate :

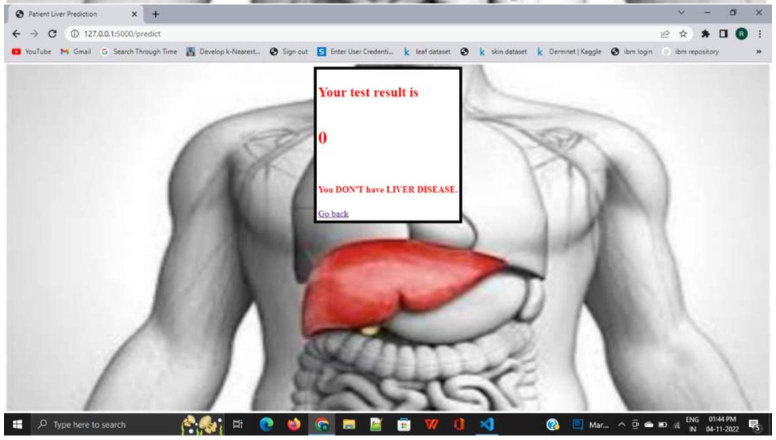
Alanine aminotransferase :

Aspartate aminotransferase :

Total Protiens :

Albumin :

Albumin and Globulin ratio :



127.0.0.1:5000/predict

Your test result is

0

You DON'T have LIVER DISEASE.

Go back

Git hub link: https://github.com/IBM-EPBL/IBM-Project-2338-1658469925

Demo link:

https://drive.google.com/file/d/1kDOQnaNnPkk45UwN9WRoe_wM8uYpIx-h/view?usp=share_link