PROJECT REPORT

Statistical Machine Learning Approaches To Liver Disease Prediction

submitted by

PNT2022TMID52707

Najila M	-	CITC1907026
Priyanka S	-	CITC1907032
Selvi K	-	CITC1907043
Swethaa Shri J	-	CITC1907053

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Diseases related to the liver and heart are becoming more and more familiar with time. Future developments in technology will only lead to an increase in these. Even if more individuals are becoming health-conscious nowadays and are enrolling in dancing and yoga courses, the issue will persist for a long time due to sedentary lifestyles and extravagances that are always being introduced and improved. The ageing of the population and the rise in the incidence and prevalence of chronic diseases result in an increased risk of liver disease-related hospitalisation or death.

Early prediction of liver disease is very important to save human life and take proper steps to control the disease. The correct prediction of liver disease can prevent life threats, and incorrect prediction can prove to be fatal at the same time. Machine learning algorithms have been playing a vital role in solving complex, highly nonlinear classification and prediction problems. Further, different machine learning algorithms are ensembled in order to increase the classification and prediction accuracy. Here different machine learning techniques like k-Nearest Neighbours (KNN), Decision Tree (DT) and Random Forest (RF) algorithms have been ensembled using the majority voting technique to predict Liver diseases.

1.2 PURPOSE

The purpose of this project is to predict the presence of liver disease with high efficiency.Instead of using individual classifier algorithms, an ensemble model that combines KNN, DT, RF is used to increase accuracy. The main objective of this project is to make socially healthy living. 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. It is also helpful for the doctors to get patients treated at the earliest. It is cost effective and user-friendly.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

Today, everyone's health is a very essential concern, so it is necessary to offer medical services that are freely accessible to everyone. Discovering the existence of liver disease at an early stage is a complex task for the doctors. But early treatment may give the liver time to heal. Since there are pre-existing processes to analyze the patient data and the classifier data, the more important fact here is to predict the same with conclusive result with a higher rate of accuracy. Existing Models based on feature selection and classification raised some issues regarding with training dataset and Test dataset.

2.2 REFERENCES

Title: Automated Prediction of Liver Disease using Machine Learning (ML) Algorithms
Author: A. Srivastava, V. V. Kumar, M. T. R and V. Vivek

In this paper, Information systems and strategic tools have lately been offered as new strategies in medical research to improve the disease detection process. A variety of machine learning (ML) algorithms are being used to predict liver diseases. They recommend employing Logistics Regression (LR), Naive Bayes Model (NB), K-Nearest Neighbor (Knn) in the project. The data was separated into two categories: patients with liver disease and sicknesses the most accurate machine learning method was used to predict the final result. The different algorithms are compared against various performance metrics and the best one is identified.

Title: A Machine Learning Based Framework to Identify and Classify Non- alcoholic Fatty Liver Disease in a Large-Scale Population
Author: Weidong Ji, Mingyue Xue, Yushan Zhang, Hua Yao, Yushan Wang

Non-alcoholic fatty liver disease (NAFLD) is a common serious health problem worldwide, which lacks efficient medical treatment. They aimed to develop and validate the machine learning (ML) models which could be used to the accurate screening of large number of people. This paper included 304,145 adults who have joined in the national physical examination and used their questionnaire and physical measurement parameters as model's candidate covariates. Absolute shrinkage and selection operator (LASSO) was used to feature selection from candidate covariates, then four ML algorithms were used to build the screening model for NAFLD, used a classifier with the best performance to output the importance score of the covariate in NAFLD.

Title: Liver Disease Diagnosis Using Machine Learning
Author: Manas Minnoor, Veeky Baths

This paper evaluates the performance of various supervised machine learning algorithms such as Logistic Regression, K-Nearest Neighbors (KNN), Extra Trees, LightGBM as well as a Multilayer Perceptron (MLP) neural network in the detection and diagnosis of liver disease. A total of 11 attributes are used to train the models. The usage of machine learning algorithms alongside human medical expertise may help drastically reduce errors in clinical diagnosis. This paper establishes the feasibility of applying machine learning in various medical fields including the diagnosis of other diseases.

Title: Statistical Machine Learning Approaches to Liver Disease Prediction
Author: Fahad Mostafa, Easin Hasan,Morgan Williamson,Hafiz khan

The study compared binary classifier machine learning algorithms (i.e., artificial neural network, random forest (RF), and support vector machine), which were utilized on a published liver disease data set to classify individuals with liver diseases, which will allow health professionals to make a better diagnosis. The synthetic minority oversampling technique was applied to oversample the minority class to regulate overfitting problems. The purpose of this study was to extract significant predictors for liver disease from the medical analysis of 615 humans using ML algorithms. Thus, this suggests that ML methods predict liver disease by

incorporating the risk factors, which may improve the inference-based diagnosis of patients.

Title: Diagnosis of Liver Disease using Machine Learning Models

Author: A. Sivasangari; Baddigam Jaya Krishna Reddy; Annamareddy Kiran; P. Ajitha

Liver-related disease poses more problems for people living and is more important nowadays to recognize the causes, and identification phase. So, for early detection of liver disease, an automated program is needed to build with more accuracy and reliability. Specific machine learning models are developed for this purpose to predict the disease. In this paper, the methods of Support Vector Machines (SVM), Decision Tree (DT) and Random Forest (RF) is proposed to predict liver disease with better precision, accuracy and reliability.

2.3 PROBLEM STATEMENT DEFINITION

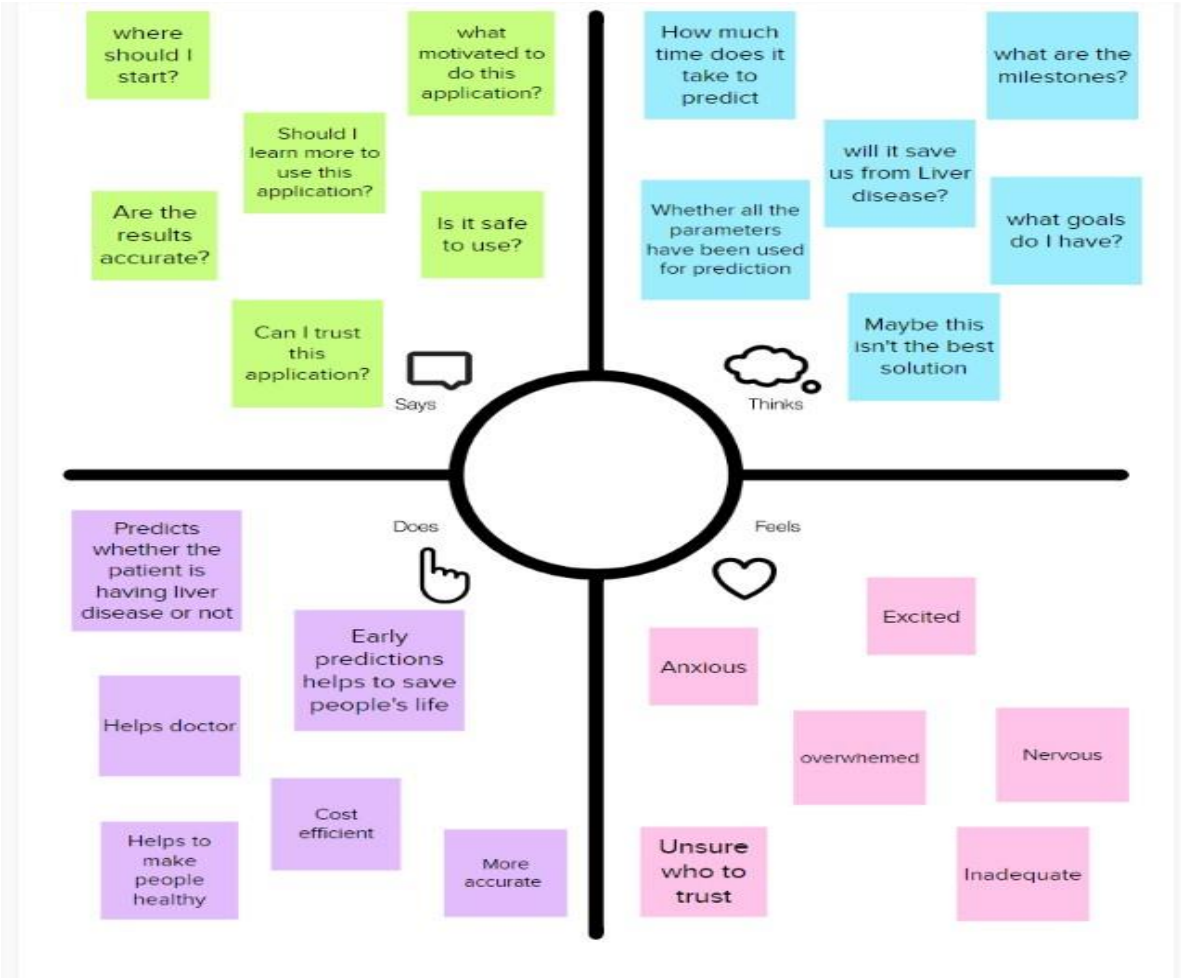
The liver plays an important role in many bodily functions from protein production and blood clotting to cholesterol, glucose (sugar), and iron metabolism. It has a range of functions, including removing toxins from the body, and is crucial to survival. The loss of those functions can cause significant damage to the body. Liver illnesses have become one of the leading causes of death worldwide in recent decades, and they have also become a life-threatening disease. The prediction of liver disease is a challenging task, which can offer automated information about the liver condition of patient so that further treatment can be made effective. The most crucial challenge is to identify possible high-risk patients to improve health care service provision and also to reduce costs.

I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

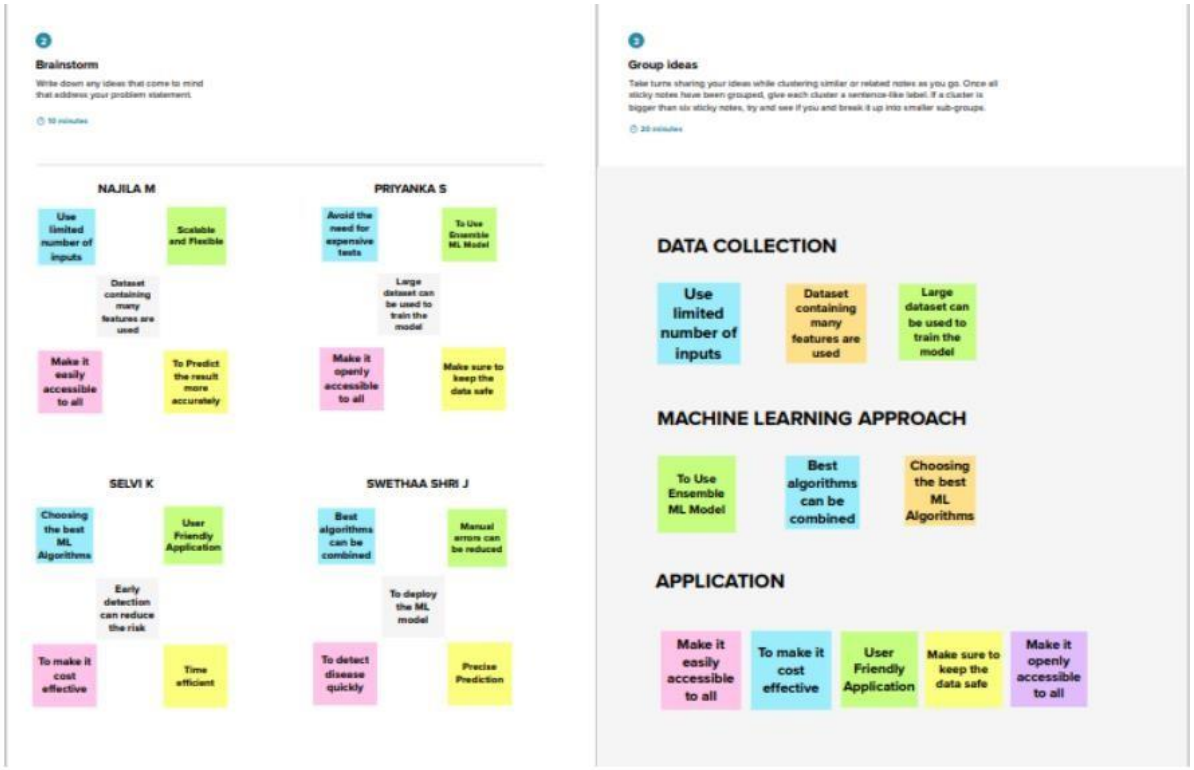
CHAPTER 3

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING



3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The number of patients with liver disease has been steadily rising as a result of heavy alcohol usage, exposure to dangerous gases, and use of contaminated food. Health Care Professionals need to obtain patient samples to identify the liver disease, which could be expensive both money and time. The key problem is doctor cannot provide a diagnosis based on test variation results.
2.	Idea / Solution description	The application will accurately and quickly identify which individuals have liver disease and which ones do not by using patient records that include blood test report results.
3.	Novelty / Uniqueness	To predict the presence of Liver disease with high efficiency.Insteadof using individual classifier algorithms, an ensemble model that combines KNN, DT, RF is used to increase accuracy. Model is deployed using Heroku cloud platform.
4.	Social Impact / Customer Satisfaction	The proposed system will make socially healthy living by decreasing mortality rate. It is also helpful for the doctors to get patients treated at the earliest.
5.	Business Model (Revenue Model)	• Health Care Sector (Hospitals). • Can generate revenue through direct customers. • Can collaborate with health care sector and generate revenue from their customers.
6.	Scalability of the Solution	It is cost effective and user friendly.

3.4 PROBLEM SOLUTION FIT

Project Title: Statistical Machine Learning Approaches To Liver Disease Prediction **Project Design Phase-I - Solution Fit Template** **Team ID**: PNT2022TMID52707

Problem-Solution fit canvas 2.0

To understand the solution proposed in coherence to the problem statement

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

People (patients/doctors) who wants to know whether he/she has liver disease or not

6. CUSTOMER CONSTAINTS CC

- should not consume alcohol, drugs, tobacco etc.
- Avoid smoking
- Maintain a balanced diet and do exercise.
- Keep track of blood sugar level

5. AVAILABLE SOLUTIONS AS

- Liver Biopsy
- Liver transplant
- Blood testing/ Imaging Testing
- Biomarkers.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- Loss of appetite
- Skin and eyes that appear yellowish (Jaundice).
- Abdominal pain and swelling.
- Swelling in the legs and ankles.
- Itchy skin.

9. PROBLEM ROOT CAUSE RC

- Family history of Liver disease
- Heavy consumption of alcohols/drugs.
- Fat accumulation in the liver
- Due to obesity.
- Increase in blood sugar level (Type 2 diabetes)

7. BEHAVIOUR BE

- Make an appointment with your doctor if he/she has any persistent signs or symptoms
- Consult local medical authority for advice.
- Follow the proper diet

Focus on J&P, tap into BE, understand

Identify strong TR & EM

3. TRIGGERS TR

Pain in the joints and upper right part of the belly triggers patients to consult a doctor.

4. EMOTIONS: BEFORE / AFTER EM

- Before: Doubt, ambiguous, stressed, disoriented.
- After accurate prediction: Happiness, determined, explicit calmness.

10. YOUR SOLUTION

An application which uses ensemble machine learning model by combining K-Nearest Neighbours, Decision Tree, Random Forest to quickly identify whether the patient is having liver disease or not more accurately. SL

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

Patients can get their results as per their input data in online.

8.2 OFFLINE

Patients can consult doctor based on the results.

Extract online & offline CH of BE

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Patients with symptoms of Liver disease	Patient having liver disease dataset contains age of the patient, gender, Alkaline Phosphotase,Total Bilirubin etc.
FR-2	Predicting the liver disease using Ensemble model	Machine Learning
FR-3	Pre-processing of liver disease dataset	Principal Component Analysis (PCA)
FR-4	Ensemble Model Training	K-Nearest Neighbors , Decision Tree, Random Forest
FR-5	Model Evaluation	Predicting the accuracy of our ensemble model and comparing it with other algorithms such as Support Vector Machine (SVM) etc.
FR-6	Model Deployment	Deploying the Machine learning model in cloud platform.

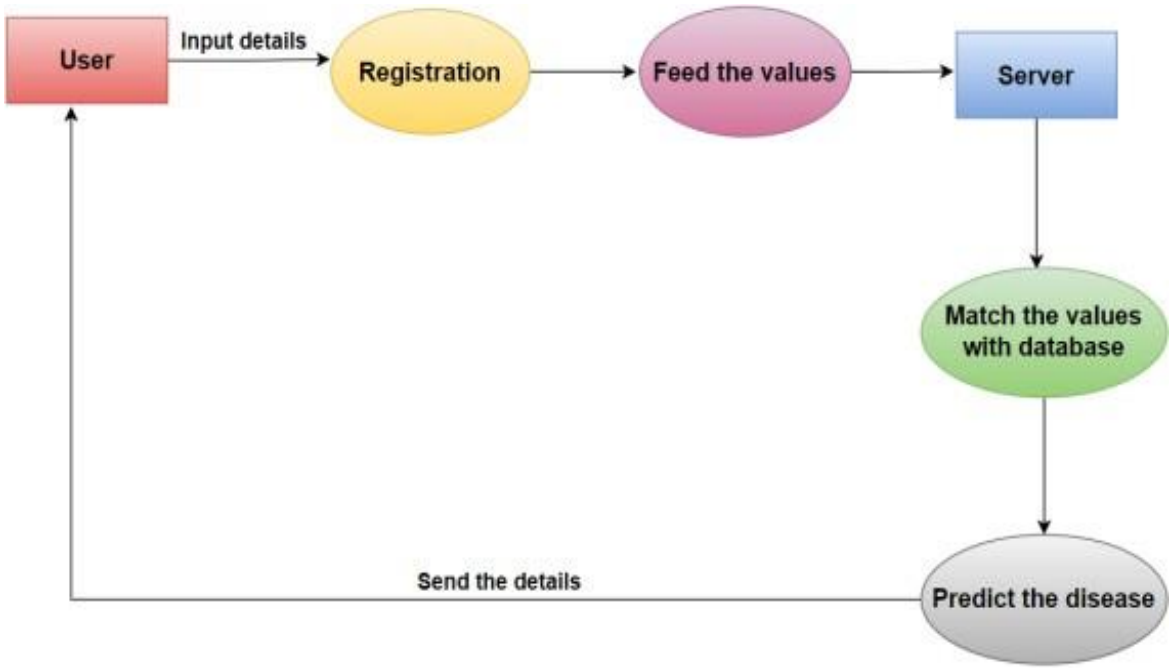
4.2 NON FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system provides a natural interaction with the users. It is user-friendly.
NFR-2	**Security**	The model enables with the high security system, as the user's data won't be shared to the other sources. Only the authorised person can access the system.
NFR-3	**Reliability**	As the system is build using a rich Ensemble model, mostly all the user input can be processed without failure in 95 per cent of use cases and since all the processing are done on cloud, the system is consider to be highly reliable.
NFR-4	**Performance**	Our system should run on 32 bit (x86) or 64 bit (x64) Dual-core 2.66-GHZ or faster processor.
NFR-5	**Availability**	The system should be available for the duration of the user access, until the user terminate the access. The system response to request of the user in less time and the recovery is done is less time.
NFR-6	**Scalability**	It provides an efficient outcome and has the ability to increase or decrease the performance of the system based on the datasets. It is cost effective and user friendly.

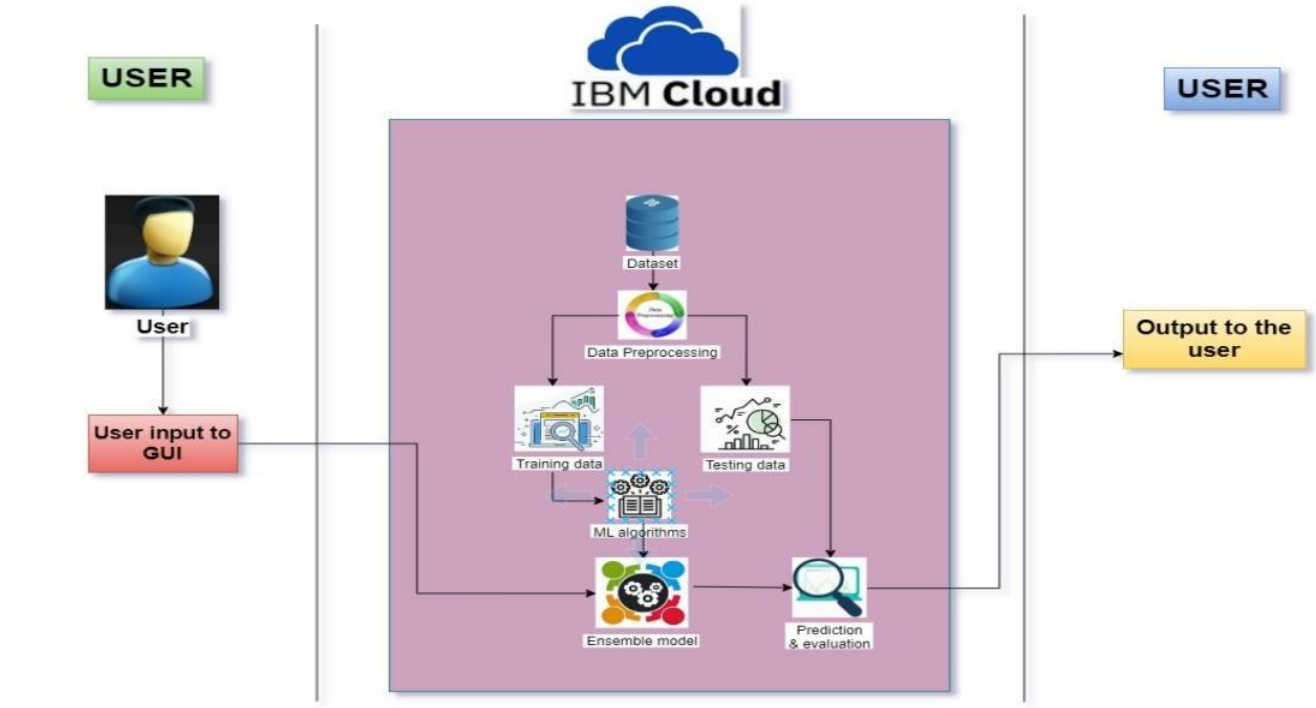
CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Patient/ Doctor (Web user)	Access Web page	USN-1	As a user, anyone can access the web page to check whether they have liver disease or not.	I can access my webpage through online at any time.	High	Sprint-1
Patient/Doctor	Usage of patient input data	USN-2	As per the symptoms of user, the disease should be predicted in easy way.	Predictio n can be done in easy way	High	Sprint-2
Patient/Doctor	Accuracy of Liver disease	USN-3	By using the prediction model the user can check whether they have liver disease or not frequently.	Prediction of liver disease will be accurate	High	Sprint-3
Administrator	Manage the web page	USN-4	As an admin, he/she can manage user details and update parameters essential for prediction	Make changes on User Interface (UI)	High	Sprint-3

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	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Najila M Selvi K
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Priyanka S Swethaa Shri J
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	10	High	Najila M Priyanka S
Sprint-2	Input Necessary Details	USN-4	As a user, I can give Input Details to Predict Likeliness of Liver Disease.	15	High	Swethaa Shri J Selvi K
Sprint-2	Data Pre-Processing	USN-5	Transform raw data into suitable format for prediction.	5	High	Selvi K
Sprint-3	Prediction of Liver Disease	USN-6	As a user, I can predict Liver Disease using machine learning model.	15	High	Najila M Swethaa Shri J
Sprint-3		USN-7	As a user, I can get accurate prediction of liver disease.	5	Medium	Priyanka S
Sprint-4	Review	USN-8	As a user, I can give feedback of the application.	20	High	Najila M Selvi K

6.2 SPRINT DELIVERY SCHEDULE

Sprint	**Total Story Point**	**Duration**	**SprintStart Date**	**SprintEnd Date(Planned)**	**Story PointsCompleted (as onPlanned EndDate)**	**Sprint ReleaseDate(Actual)**
Sprint-1	20	6Days	24Oct2022	29Oct2022	10	29 Oct 2022
Sprint-2	20	6Days	31Oct2022	05Nov2022	10	03 Nov 2022
Sprint-3	20	6Days	07Nov2022	12Nov2022	10	11 Nov 2022
Sprint-4	20	6Days	14Nov2022	19Nov2022	10	19 Nov 2022

CHAPTER 7

CODING & SOLUTIONING

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
from sklearn.model_selection import train_test_split, StratifiedKFold, GridSearchCV
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report
```

```
data=pd.read_csv('/content/indian_liver_patient.csv')
```

```
data.info()
```

```
def partition(x):
    if x=='Male':
        return 1
    return 0
data['Gender']=data['Gender'].map(partition)
```

```
def partition(x):
    if x==2:
        return 0
    return 1
data['Dataset']=data['Dataset'].map(partition)
```

```
data['Dataset']
```

```
x=data.drop(columns='Dataset',axis=1)
y=data['Dataset']
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,stratify=y,random_state=42)
```

```
print(x.shape,x_train.shape,x_test.shape)
```

```
(1636, 10) (1145, 10) (491, 10)
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
xtrain=sc.fit_transform(x_train)
xtest=sc.transform(x_test)
```

```
def my_confusion_matrix(y_test, y_pred, plt_title, accuracy_title):
    cm=confusion_matrix(y_test, y_pred)
    print(f'{accuracy_title} accuracy score:', '{:.2%}'.format(accuracy_score(y_test, y_pred)))
    print(classification_report(y_test, y_pred))
    sns.heatmap(cm, annot=True, fmt='g', cbar=False, cmap='BuPu')
    plt.xlabel('Predicted Values')
    plt.ylabel('Actual Values')
    plt.title(plt_title)
    plt.show()
    return cm
```

```
random_state = 42
classifier = [KNeighborsClassifier(),
              DecisionTreeClassifier(random_state = random_state),
             RandomForestClassifier(random_state = random_state),
             ]

knn_param_grid = {"n_neighbors": np.linspace(1,19,10, dtype = int).tolist(),
                  "weights": ["uniform","distance"],
                  "metric":["euclidean","manhattan"]}

dt_param_grid = {"min_samples_split" : range(10,500,20),
                 "max_depth": range(1,20,2)}


rf_param_grid = {"max_features": [1,3,10],
                 "min_samples_split":[2,3,10],
                 "min_samples_leaf":[1,3,10],
                "bootstrap":[False],
                 "n_estimators":[100,300],
                 "criterion":["gini"]}
classifier_param = [knn_param_grid,
                    dt_param_grid,
                   rf_param_grid,
                   ]
```

```
cv_result = []
best_estimators = []
for i in range(len(classifier)):
    clf = GridSearchCV(classifier[i], param_grid=classifier_param[i], cv = StratifiedKFold(n_splits = 10), scoring = "accuracy", n_jobs = -1,verbose
    clf.fit(x_train,y_train)
    cv_result.append(clf.best_score_ * 100)
    best_estimators.append(clf.best_estimator_)
    print(cv_result[i])
```

```
cv_results = pd.DataFrame({"Cross Validation Means":cv_result, "ML Models":[ "KNeighborsClassifier", "Decision Tree Classifier",
             "Random Forest Classifier",
            ]})

g = sns.barplot("Cross Validation Means", "ML Models", data = cv_results)
g.set_xlabel("Mean Accuracy")
g.set_title("Cross Validation Scores")
```

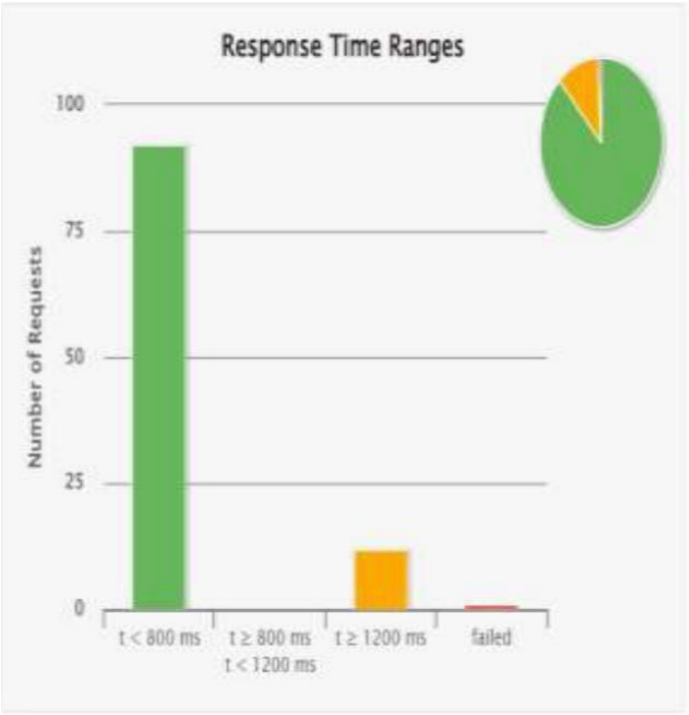
```
knn = KNeighborsClassifier(n_neighbors = 9)
knn.fit(x_train, y_train)
y_head_knn = knn.predict(x_test)

dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
y_head_dt = dt.predict(x_test)

rf = RandomForestClassifier(n_estimators = 250, random_state = 1)
rf.fit(x_train,y_train)
y_head_rf = rf.predict(x_test)
```

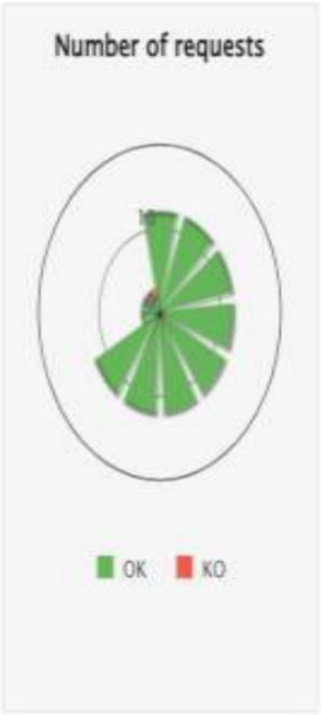
```
votingC = VotingClassifier(estimators = [("knn",best_estimators[0]),
                                         ("dt",best_estimators[1]),
                                         ("rf",best_estimators[2])],
                                         voting = "hard", n_jobs = -1)
votingC = votingC.fit(x_train, y_train)
y_pred=votingC.predict(x_test)
my_confusion_matrix(y_test, y_pred, 'Ensemble Model CM', 'Ensemble Model')
```

CHAPTER 8

TESTING

8.1 TEST CASES





8.2 User Acceptance Testing

Test Case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status
HomePage_ TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on User accountbutton	1. Click on User Account icon 2. Verify login/Signuppopup displayed or not	-	Login/Signup popupshould display	Working as expected	Pass
RegisterPage _TC_OO1	UI	Registration Page	Verify the UI elements in Register/Signup page	1. Enter URL and clickgo 2. Click on User Account dropdown button 3. Verify Register/Signup popup with below UI elements: a.name text box b. email text box c. phone number text box d. password text box e.occupation text box f. Already a member? login	-	Application should show below UI elements: a.name text box b.email text box c.phone number textbox d.password text box e.occupation text box f.Already have an account? Click login	Working as expected	Pass

RegisterPage _TC_OO2	Functional	Registration Page	Verify the users entering the unique email	1. Click on User Account dropdown button 2. Verify Register/Signup page accepts only unique email	-	Application should allow only unique email address	Working as expected	Pass
RegisterPage _TC_OO2	Functional	Registration Page	Verify that the user can able to register with valid credentials	1. Click on User Account dropdown button 2. Click Register/Signuppopup a.Enter name b.Enter email c.Enter phone number d.Enter password e.Enter occupation f.Click Register button	-	User should navigate to sign in page	Working as expected	Pass
LoginPage_ TC_OO1	UI	Login page	Verify the UI elements in Login/Sign in page	1. Click on User Accountdropdown button 2. Verify login/Signup popup with below UI elements: a.email text box b.password text box c.Login button d.Not a member? Create account	-	Application should show below UI elements: a.email text box b.password text boxc.Login button with orange color d. Not a member? Create account	Working as expected	Pass
LoginPage_ TC_OO2	Functional	Login page	Verify user is able to loginto application with Valid credentials	1. Click on User Account dropdown button andclick on sign in/login pop up 2. Enter Valid email in Email text box 3. Enter valid password in password text box 4.Click on login button	Email: 123@gmail.com password: 123456	User should navigate to Prediction page	Working as expected	Pass
LoginPage_ TC_OO3	Functional	Login page	Verify user is not able to log into application with Invalid credentials	1. Click on User Accountdropdown button andclick on sign in/login pop up 2. Enter Invalid email in Email text box 3.Enter valid password	Email: 12@gmail.com password: 123456	User will be at the same page without navigating	Working as expected	Pass

				in password text box 4.Click on login button				
LoginPage_TC_OO4	Functional	Login page	Verify user is not able to log into application with Invalid credentials	1.Click on User Account dropdown button and click on sign in/login pop up 2.Enter Valid username/email in Email text box 3.EnterInvalid password in password text box 4.Click on login button	Email:123@gmail.com password: 12345	User will be at the same page without navigating	Working as expected	Pass
Prediction_Page_TC_OO1	UI	Prediction page	Verify user is able to see the prediction form, prediction and go back button	1.Click on User Account dropdown button 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button		Application should navigate to Prediction page and user can able to view the prediction form, predict and go back button	Working as expected	Pass
Prediction_Page_TC_OO2	UI	Prediction page	Verify the UI elements prediction form	1.Click on User Account dropdown button 2.Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button 5. Verify Prediction form popup with below UI elements: a.Age		User can able to enter the details in prediction form	Working as expected	Pass

				b.Gender c.Total Bilirubin d.Direct Bilirubin e.Alkalin Phosphate f.Alamine Amino Transferase g.Aspartate Amino Trandferase h.Total proteins i.Albumin j.Albumin Globulin Ratio				
Prediction_Page_TC_OO3	Functional	Prediction Page	Verify user is able to click the predict button	1. Click on User Accountdropdown button 2. Enter Valid email in Email text box 3.Enter valid password in password text box 4.Click on login button 5.Enter details in Prediction form 6.Click on predict button		User Should navigate to Predicted result page	Working as expected	Pass
Prediction_Page2_TC_OO1	Functional	Prediction page	Verify user is able to click the Home and Predict Again button			User should navigate to login page after clicking on Home button and prediction page after clicking on Predict Again button	Working as expected	Pass

8.2.1 DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Total
By Design	1	0	1	0	2
Duplicate	0	0	0	0	0
External	0	0	2	0	2
Fixed	4	1	0	1	6
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	1	1
Won’t Fix	1	0	1	0	2
Total	6	1	4	3	14

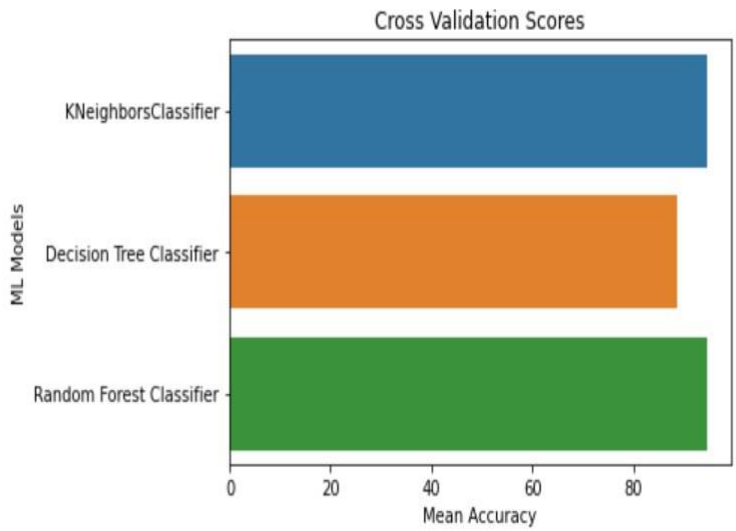
8.2.2 TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Client Application	10	0	3	7
Security	2	0	1	1
Performance	3	0	1	2
Exception Reporting	2	0	0	2

CHAPTER 9

RESULTS

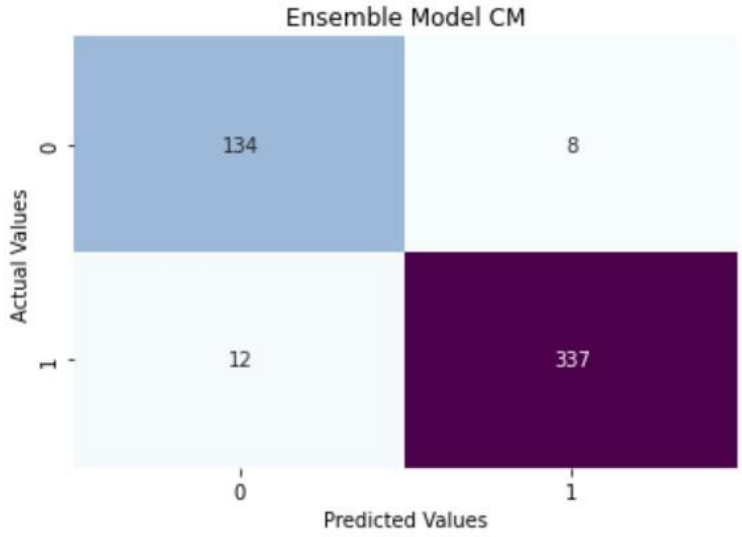
9.1 PERFORMANCE METRICS



83]:
```
best_estimators
```

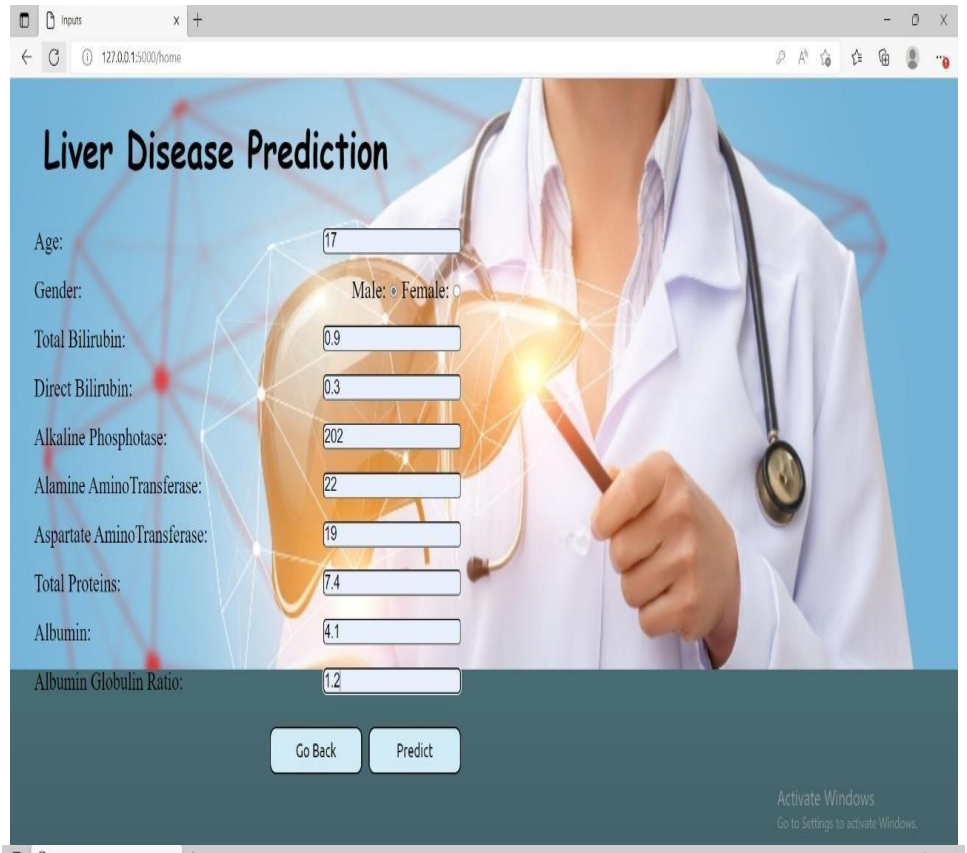
83]:
```
[KNeighborsClassifier(metric='euclidean', n_neighbors=3, weights='distance'),
 DecisionTreeClassifier(max_depth=17, min_samples_split=10, random_state=42),
 RandomForestClassifier(bootstrap=False, max_features=1, min_samples_split=10,
                        random_state=42)]
```

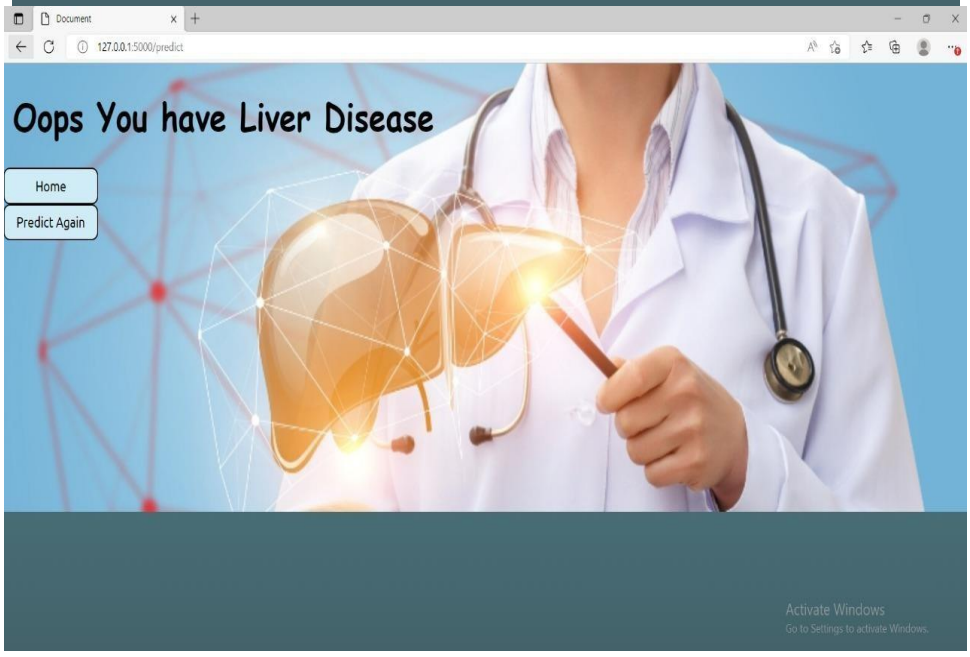
```
Ensemble Model accuracy score: 95.93%
              precision    recall  f1-score   support

           0       0.92      0.94      0.93       142
           1       0.98      0.97      0.97       349

    accuracy                           0.96       491
   macro avg       0.95      0.95      0.95       491
weighted avg       0.96      0.96      0.96       491
```



t[85]:
```
array([[134,   8],
       [ 12, 337]])
```

OUTPUT SCREENSHOT FOR LIVER DISEASE





OUTPUT SCREENSHOT FOR NORMAL PERSON

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Results are projected in a matter of seconds after entering the information. In contrast to the conventional procedure, you don't need to wait for a doctor to arrive.
- The application will accurately and quickly identify which individuals have liver disease.
- The performance classification of liver based diseases is further improved in our proposed ensemble model.
- Risky factors can be predicted early by machine learning models.

DISADVANTAGES

- Certain approaches being applicable only for small data.
- Certain combination of classifier over fit with data set while others are under fit.
- Some approaches are not adoptable for real time collection of database implementation.

CHAPTER 11
CONCLUSION

The ageing of the population and the rise in the incidence and prevalence of chronic diseases result in an increased risk of liver disease-related hospitalisation or death.This is notably high for people who have several diseases, which results in large resource consumption. Finding potential high-risk patients is the biggest difficulty in order to increase the quality of medical care and cut expenditures. The main goal of the research is to put ensemble algorithms, such as K-Nearest Neighbors, Random Forest, and Decision Tree, into practise in order to forecast the likelihood of hospitalisation or mortality starting from administrative and socioeconomic information. Our goal is to get prediction on the basis of given datasets of people whether the person is having the chronic disease or liver disease symptoms or not. This system will be very useful for many hospitals and even professional doctors to easily detect the disease. Also, general user can use this system for their finding out the disease. This system will change the way and can be early as possible as it will lead to save the person's life. This whole work is focused on how we can predict the disease by given datasets so that will help in preventing and curing the disease of the patients.

CHAPTER 12
FUTURE SCOPE

This project can be further developed by establishing an alarm system for the patient's relatives and doctor according to the risk level. Deep learning algorithms can be used to enhance the performance. However, in future we are planning to collect the very recent data from various regions across the world for liver disease prediction.

APPENDIX

SOURCE CODE

MODEL CREATION

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
from sklearn.model_selection import train_test_split, StratifiedKFold, GridSearchCV
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,classification_report
```

```
data=pd.read_csv('/content/indian_liver_patient.csv')
```

```
data.info()
```

```
def partition(x):
    if x=='Male':
        return 1
    return 0
data['Gender']=data['Gender'].map(partition)
```

```
def partition(x):
    if x==2:
        return 0
    return 1
data['Dataset']=data['Dataset'].map(partition)
```

```
data['Dataset']
```

```
x=data.drop(columns='Dataset',axis=1)
y=data['Dataset']
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,stratify=y,random_state=42)
```

```
print(x.shape,x_train.shape,x_test.shape)
```

```
(1636, 10) (1145, 10) (491, 10)
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
xtrain=sc.fit_transform(x_train)
xtest=sc.transform(x_test)
```

```
def my_confusion_matrix(y_test, y_pred, plt_title, accuracy_title):
    cm=confusion_matrix(y_test, y_pred)
    print(f'{accuracy_title} accuracy score:', '{:.2%}'.format(accuracy_score(y_test, y_pred)))
    print(classification_report(y_test, y_pred))
    sns.heatmap(cm, annot=True, fmt='g', cbar=False, cmap='BuPu')
    plt.xlabel('Predicted Values')
    plt.ylabel('Actual Values')
    plt.title(plt_title)
    plt.show()
    return cm
```

```
random_state = 42
classifier = [KNeighborsClassifier(),
              DecisionTreeClassifier(random_state = random_state),
             RandomForestClassifier(random_state = random_state),
             ]

knn_param_grid = {"n_neighbors": np.linspace(1,19,10, dtype = int).tolist(),
                  "weights": ["uniform","distance"],
                  "metric":["euclidean","manhattan"]}

dt_param_grid = {"min_samples_split" : range(10,500,20),
                 "max_depth": range(1,20,2)}


rf_param_grid = {"max_features": [1,3,10],
                 "min_samples_split":[2,3,10],
                 "min_samples_leaf":[1,3,10],
                "bootstrap":[False],
                 "n_estimators":[100,300],
                 "criterion":["gini"]}
classifier_param = [knn_param_grid,
                    dt_param_grid,
                   rf_param_grid,
                   ]
```

```
cv_result = []
best_estimators = []
for i in range(len(classifier)):
    clf = GridSearchCV(classifier[i], param_grid=classifier_param[i], cv = StratifiedKFold(n_splits = 10), scoring = "accuracy", n_jobs = -1,verbose
    clf.fit(x_train,y_train)
    cv_result.append(clf.best_score_ * 100)
    best_estimators.append(clf.best_estimator_)
    print(cv_result[i])
```

```
cv_results = pd.DataFrame({"Cross Validation Means":cv_result, "ML Models":[ "KNeighborsClassifier", "Decision Tree Classifier",
            "Random Forest Classifier",
           ]})

g = sns.barplot("Cross Validation Means", "ML Models", data = cv_results)
g.set_xlabel("Mean Accuracy")
g.set_title("Cross Validation Scores")
```

```
knn = KNeighborsClassifier(n_neighbors = 9)
knn.fit(x_train, y_train)
y_head_knn = knn.predict(x_test)

dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
y_head_dt = dt.predict(x_test)

rf = RandomForestClassifier(n_estimators = 250, random_state = 1)
rf.fit(x_train,y_train)
y_head_rf = rf.predict(x_test)
```

```
votingC = VotingClassifier(estimators = [("knn",best_estimators[0]),
                                         ("dt",best_estimators[1]),
                                         ("rf",best_estimators[2])],
                                         voting = "hard", n_jobs = -1)
votingC = votingC.fit(x_train, y_train)
y_pred=votingC.predict(x_test)
my_confusion_matrix(y_test, y_pred, 'Ensemble Model CM', 'Ensemble Model')
```

FLASK APP

main.py

```python
from flask import Flask, render_template, request,redirect,session, url_for
from flask_mail import Mail, Message
from itsdangerous import URLSafeTimedSerializer, SignatureExpired
import mysql.connector
import os
import pickle
from flask_login import UserMixin, login_user, LoginManager, login_required, logout_user, current_user

from flask_mysqldb import MySQL

app = Flask(__name__)
app.secret_key=os.urandom(24)
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'liver'

mysql = MySQL(app)

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

@app.route('/register/')
def about():
    return render_template('register.html')

@app.route('/home')
def home():
    if 'email' in session:
        return render_template('form.html')
    else:
        return redirect('/')

@app.route('/login_validation',methods=['POST'])
def login_validation():
    email=request.form.get('email')
```

main.py

```
    email = request.form.get('email')
    password = request.form.get('password')
    occupation = request.form.get('occupation')
    phone = request.form.get('phone')
    if mysql:
        print("Connection Successful!")
        cursor = mysql.connection.cursor()
        cursor.execute(
            """INSERT INTO `user_details` (`username`,`email`,`phone`,`occupation`,`password`) VALUES ('{}','{}','{}','{}','{}')""".form
        mysql.connection.commit()
        cursor.close()
    else:
        print("Connection Failed!")
    return "User Registered Successfully."


@app.route('/logout')
def logout():
    session.pop('email')
    return redirect('/')
@app.route('/form',methods=['POST'])
def form():
    print("HOME")
    return redirect('form.html')
@app.route('/predict', methods=['POST'])
def predict():

    age = request.form['age']
    gender = request.form['gender']
    tb = request.form['tb']
    dbi = request.form['dbi']
    ap = request.form['ap']
    aa1 = request.form['aa1']
    aa2 = request.form['aa2']
    tp = request.form['tp']
    a = request.form['a']
    agr = request.form['agr']
    if gender == "Male":
```

main.py

```
        aa1 = request.form['aa1']
        aa2 = request.form['aa2']
        tp = request.form['tp']
        a = request.form['a']
        agr = request.form['agr']
        if gender == "Male":
            gender = 1
        else:
            gender = 0
        data = [[float(age),
                 float(gender),
                 float(tb),
                 float(dbi),
                 float(ap),
                 float(aa1),
                 float(aa2),
                 float(tp),
                 float(a),
                 float(agr)]]

        model = pickle.load(open('liver1.pkl', 'rb'))

        prediction = model.predict(data)
        if (prediction == 1):
            return render_template('noChance.html',
                                   prediction='You don\'t have disease.')
        else:
            return render_template('chance.html',
                                   prediction='You dead boy.')



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


```

HOME PAGE (HTML)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Home</title>
<style>
    @import url('https://fonts.googleapis.com/css?family=Roboto:700&display=swap');
*{
    padding: 0;
    margin: 0;
}
.wrapper{
    background: url(pic1.jpg) ;
    background-size: cover;
    height: 100vh;
}
.wrapper .center{
    position: absolute;
    left: 50%;
    top: 55%;
    transform: translate(-50%, -50%);
    font-family: sans-serif;
    user-select: none;
}
.center h1{
    color: black;
    font-size: 50px;
    width: 885px;
    font-weight: bold;
    text-align: center;
}

.center .buttons{
    margin: 40px 390px;
}
.buttons button{
    height: 50px;
```

```
.center .buttons{
    margin: 40px 390px;
}
.buttons button{
    height: 50px;
    width: 150px;
    font-size: 18px;
    font-weight: 600;
    color: #ffb3b3;
    background: red;
    outline: none;
    cursor: pointer;
    border: 1px solid #cc0000;
    border-radius: 25px;
    transition: .4s;
}
.buttons button:hover{
    background: #cc0000;
}
</style>
</head>
<body>
    <div class="wrapper">
                <div class="center">
            <h1>Liver Disease Prediction System</h1>
            <div class="buttons">
                <form action="/form" method="POST">
                <input type="submit" value="predict">
                </form>

            </div>
                </a>
            </div>
    </div>
</body>
</html>
```

LOGIN PAGE

templates > login.html > html > head > style

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <style>
      .bg-nav{
background: #003973;  /* fallback for old browsers */
background-image: url("{{ url_for('static',  filename='Images/3.jpg') }}");;
background-size: cover;
}
.row{
margin-top:20px;
}
</style>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv

    <title>Liver Disease predictor</title>

  </head>
  <body class="bg-nav">
    <nav class="navbar">
      <a href="" class="navbar-brand text-light">Liver Disease Predictor</a>
    </nav>
<div id="content" style="margin-top: 100px">
    <div class="container">
      <div class="row">
        <div class="col-md-8">
            <h1 class="text-light display-15 mt" style="font-size:80px"> Liver Disease Predictor.</h1>
        </div>
        <div class="col-md-4">
            <div class="card">
              <div class="card-body">
                <form class="form" method="post" action="/login_validation">
```



```
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv

    <title>Liver Disease predictor</title>

  </head>
  <body class="bg-nav">
    <nav class="navbar">
      <a href="" class="navbar-brand text-light">Liver Disease Predictor</a>
    </nav>
<div id="content" style="margin-top: 100px">
    <div class="container">
      <div class="row">
        <div class="col-md-8">
            <h1 class="text-light display-15 mt" style="font-size:80px"> Liver Disease Predictor.</h1>
        </div>
        <div class="col-md-4">
            <div class="card">
              <div class="card-body">
                <form class="form" method="post" action="/login_validation">
                  <label>Email</label><br>
                  <input type = "email" class="form-control" name="email"><br>
                  <label>Password</label><br>
                  <input type="password" class="form-control" name="password"><br><br>
                  <input type="submit" class="btn btn-primary btn-block btn-lg" value="Login">
                </form>
                  <p style="margin-top:2px">Not a member? <a href="/register/">Create Account</a></p>
              </div>
            </div>
        </div>
      </div>
    </div>
</div>


  </body>
</html>
```

PREDICTION PAGE(HTML)

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Inputs</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Ubuntu&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="/static/home.css">
    <link rel="stylesheet" href="/static/predict.css">
</head>

<body>
    <div class="content">
        <h1> Liver Disease Prediction</h1>

        <form class="container" action="/predict" method="post">
            <ul class="formfields">
                <li class="fields">
                    <label for="age">Age:</label>
                    <input type="text" id="age" name="age" placeholder=" Enter your age : 3 - 90" required>
                </li>
                <li class="fields">
                    <label for="gender">Gender:</label>
                    <div class="gender_selection">
                        <label for="Male">Male:</label>
                        <input type="radio" id="Male" name="gender" required>
                        <label for="Female">Female:</label>
                        <input type="radio" id="Female" name="gender">
                    </div>
                </li>
                <li class="fields">
                    <label for="tb">Total Bilirubin:</label>
                    <input type="text" id="tb" name="tb" placeholder=" 0 - 80 " required>
```


templates > form.html > html > body > div.content > form.container > div.buttons

```html
                </li>
                <li class="fields">
                    <label for="db">Direct Bilirubin:</label>
                    <input type="text" id="dbi" name="dbi" placeholder=" 0 - 20 " required>
                </li>
                <li class="fields">
                    <label for="ap">Alkaline Phosphotase:</label>
                    <input type="text" id="ap" name="ap" placeholder=" 50 - 2200 " required>
                </li>
                <li class="fields">
                    <label for="aa1">Alamine AminoTransferase:</label>
                    <input type="text" id="aa1" name="aa1" placeholder=" 10 - 2000 " required>
                </li>
                <li class="fields">
                    <label for="aa2">Aspartate AminoTransferase:</label>
                    <input type="text" id="aa2" name="aa2" placeholder=" 10 - 5000 " required>
                </li>
                <li class="fields">
                    <label for="tp">Total Proteins:</label>
                    <input type="text" id="tp" name="tp" placeholder=" 0 - 10 " required>
                </li>
                <li class="fields">
                    <label for="a">Albumin:</label>
                    <input type="text" id="a" name="a" placeholder=" 0 - 10 " required>
                </li>
                <li class="fields">
                    <label for="agr">Albumin Globulin Ratio:</label>
                    <input type="text" id="agr" name="agr" placeholder=" 0 - 10" required>
                </li>
            </ul>
            <div class="buttons">
                <form action="/">
                    <button class="button"> Go Back </button>
                </form>
                <input type="submit" class="button" value="Predict">
            </div>
        </form>
```

GITHUB

https://github.com/IBM-EPBL/IBM-Project-13374-1659517516

PROJECT DEMO

https://drive.google.com/drive/folders/1OLpFaqpdzRiuNqjj3zQ7X65bJNHMyh-p