Statistical Machine Learning Approaches to Liver Disease Prediction

TEAM ID: PNT2022TMID24640
Team Members:
- Kenny Mathew
- P A Mohammed Arshad
- Aarthi A
- Sankeerth S Narayan

1. INTRODUCTION
1.1 Project Overview

Statistical Machine Learning Approaches to liver disease prediction is used to find the liver disease in a human being. Cirrhosis is a leading cause of mortality and morbidity across the world. To detect disease, healthcare professionals need to collect samples from patients which can cost both time and money. Often, more than one kind of test or many samples are needed from the patient to accumulate all the necessary information for a better diagnosis. The most routine tests are urinalysis, complete blood count (CBC), and comprehensive metabolic panel (CMP). These tests are generally less expensive and can still be very informative. The liver has many functions such as glucose synthesis and storage, detoxification, production of digestive enzymes, erythrocyte regulation, protein synthesis, and various other features of metabolism. Chronic liver diseases include chronic hepatitis, fibrosis, and cirrhosis. Hepatitis can occur from viral infection (e.g., hepatitis c virus) or auto-immune origin. Inflammation from hepatitis infection can cause tissue damage and scarring to occur in the liver. Moderate scarring is classified as fibrosis, while severe liver damage/scarring is classified as cirrhosis. Fibrosis and cirrhosis can also occur from alcoholism and non-alcoholic fatty liver disease. It is the 11th leading cause of death and 15th leading cause of morbidity, accounting for 2.2% of deaths and 1.5% of disability. Among the various liver disease types, Chronic Liver Disease (CLD) caused 1.32 million deaths in 2017, approximately two-thirds among men and one-third among women.

1.2 Purpose

The purpose of this project is to identify whether a user has liver disease or not based upon the data entered as input. Based on the inputs entered by the user, a result is displayed. If the user has been diagnosed with any form of liver disease, then it is highly recommended that they seek medical attention immediately. Our project's objective is to

analyse data from liver patients with a focus on the correlations between a vital list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease. We are creating a model in this instance by utilizing different machine learning algorithms to identify the most accurate model. and incorporate into web applications built with flask. By entering certain parameters into the web application, users can predict the disease.

2. LITERATURE SURVEY

2.1 Existing Problem

[1] In this system they described how to predict risk of liver disease for a person, based on the blood test report results of the user using various machine learning algorithms. The final output was predicted based on the most accurate machine learning algorithm. Based on the accurate model they designed a system which asks a person to enter the details of his/her blood test report. Then the system uses the most accurate model which is trained to predict, whether a person has risk of liver disease or not.

[2] In the 21st-century, the issue of liver disease has been increasing all over the world. The overall percentage of death by liver disease is 3.5% worldwide. Chronic Liver disease is also considered to be one of the deadly diseases, so early detection and treatment can recover the disease easily. This research work is based on liver disease prediction using machine learning algorithms. Liver disease prediction has various levels of steps involved, pre-processing, feature extraction, and classification. In this s research work, a hybrid classification method is proposed for liver disease prediction. And Datasets are collected from the Kaggle database of Indian liver patient records. The proposed model achieved an accuracy of 77.58%.

[3] Machine learning (ML) utilizes artificial intelligence to generate predictive models efficiently and more effectively than conventional methods through detection of hidden patterns within large data sets. In this review, we examine the literature pertaining to machine learning in hepatology and liver transplant medicine. We provide an overview of the strengths and limitations of ML tools and their potential applications to both clinical and molecular data in hepatology. ML has been applied to various types of data in liver disease research, including clinical, demographic, molecular, radiological, and pathological data. We anticipate that use of ML tools to generate predictive algorithms will change the face of clinical practice in hepatology and transplantation.

[4] Data Mining technologies have been widely used in the process of medical diagnosis and prognosis, extensively. In this project, the patient data sets are analysed for the predictability of the subject to have a liver disease based purely on a widely analysed classification model. This System predict the same conclusive result with a higher rate

of accuracy. J48 algorithm is considered to be a better performing algorithm when it comes to feature selection with an accuracy rate of 95.04%.

[5] Various kinds of pressure and unbalanced eating behaviours, along with alcohol inhalation and on-going toxic gases, etc, cause liver disease in patients. For this purpose, the type of data mining algorithms can help medical doctors to diagnose patients in hospital. This paper analyses meta learning algorithms to classify the Indian liver patient dataset. Ad boost, logit boost, Bagging and Grading meta learning algorithms are applied to this data set. Key role is played by Grading algorithm in shaping enhanced classification accuracy (Correct Classification Rate) of a data set.

2.2 Reference

[1] Rakshith D B, Mrigank Srivastava, Ashwani Kumar, Gururaj S P, “Liver Disease Prediction System Using Machine Learning Techniques”, IJERT, Vol-10, Issue 6, June 2021.

[2] Shaheamlung, Golmei & Kaur, Harshpreet. (2021). The Diagnosis of Chronic Liver Disease using Machine Learning Techniques. INFORMATION TECHNOLOGY IN INDUSTRY. 9. 10.17762/itii.v9i2.382.

[3] Spann A, Yasodhara A, Kang J, Watt K, Wang B, Goldenberg A, Bhat M. Applying Machine Learning in Liver Disease and Transplantation: A Comprehensive Review. Hepatology (Baltimore, Md.). 2020 Mar;71(3):1093-105.

[4] Durai V, Ramesh S, Kalthireddy D. Liver disease prediction using machine learning. Int. J. Adv. Res. Ideas Innov. Technol. 2019;5(2):1584-8.

[5] Pasha M, Fatima M. Comparative Analysis of Meta Learning Algorithms for Liver Disease Detection. J. Softw.. 2017 Dec 1;12(12):923-33

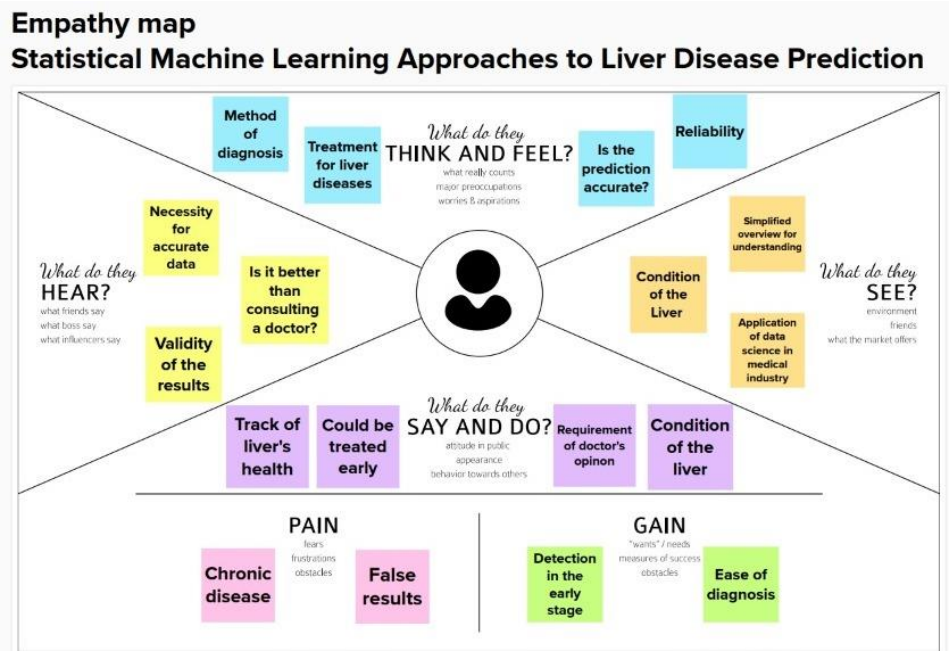
2.3 Problem Statement Definition

Statistical Machine Learning Approaches to Liver Disease Prediction
Liver Disease prevent the liver's normal function. An effective task that can assist clinicians in quickly diagnosing the disease is the early prediction of liver disease using classification algorithms. This project's main goal is to compare the predicted accuracy of different categorization algorithms by analysing their parameters. A core list of liver enzymes, proteins, age, and gender are the focus of this project's analysis of patient data from the liver, which aims to forecast the likelihood of liver illness. ML algorithms are new techniques to handle many hidden problems in medical data sets. This approach can help

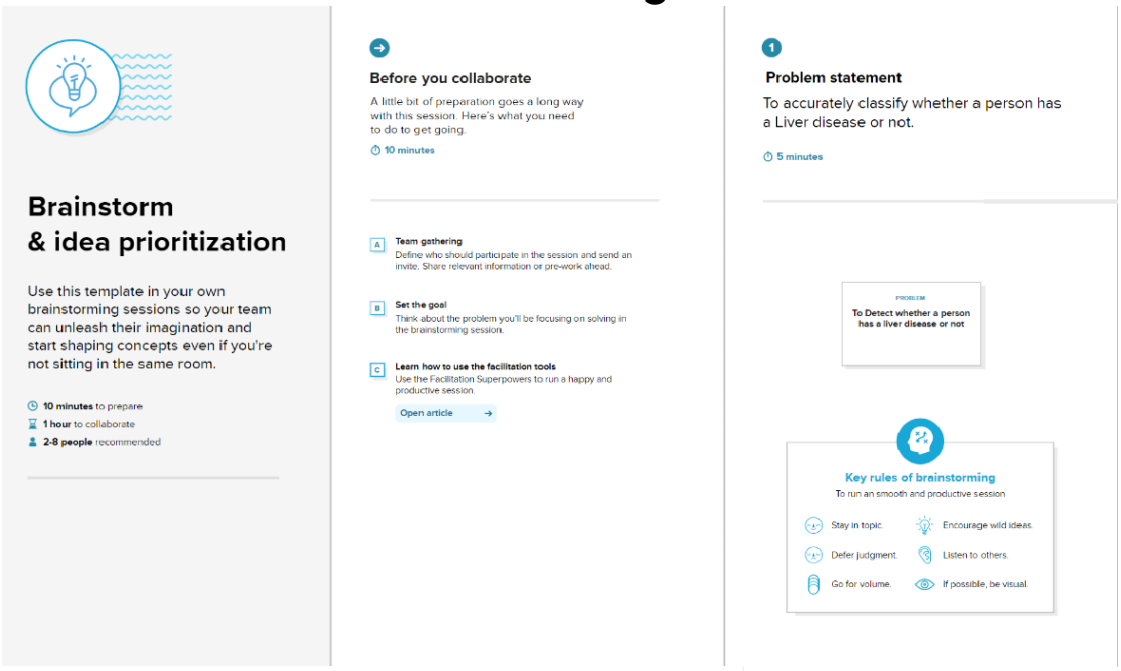
healthcare management and professionals to explore better results in numerous clinical applications, such as medical image processing, language processing, and tumour or cancer cell detection, by finding appropriate features.

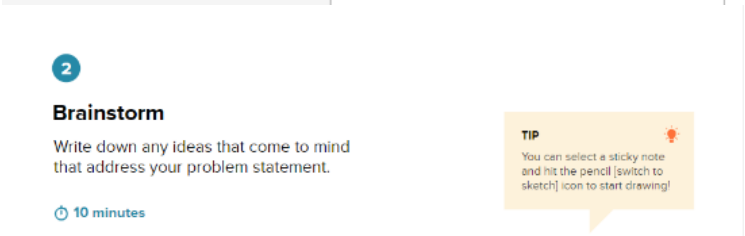
3. IDEATION AND PROPOSED SOLUTION

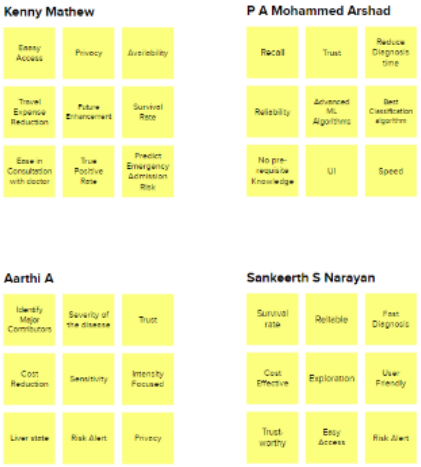
3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming





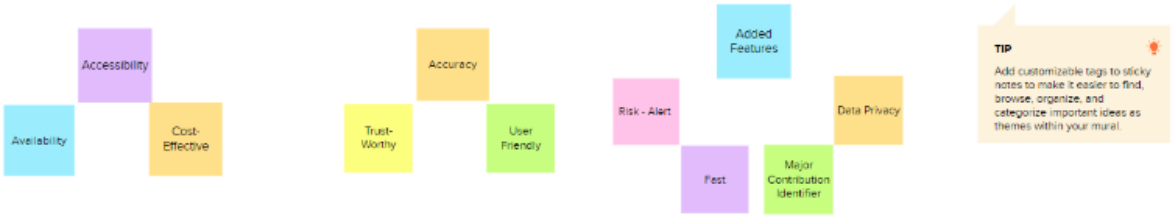


3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

3.3 Proposed Solution

Feasibility and Social Impact

The system being proposed here uses concept of machine learning and deep learning, and the models are first trained, then tested. Finally, the most accurate model will predict the final result. At first, the system asks you to enter your details including age, gender, total Bilirubin, direct Bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT and Alkphos. Values of last eight parameters mentioned here, can be known by blood test report of the user. After taking these inputs from the user, the system compares the data input with the training dataset of most accurate model and then predicts the result accordingly as risk or no risk.

Novelty

The proposed solution uses advanced machine learning models like Random Forest, neural networks etc which provide better accuracy than traditional machine learning models and the is suitable for prediction with a greater number of features.

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

High accuracy: The system predicts the results with 100 % accuracy for the dataset that we have used while creating this application. While the accuracy might be different in some cases, it will still be high enough to be trustworthy at a large scale.

Immediate results: The results here are predicted within seconds of entering the details. You don't need to wait for a doctor to come, unlike in traditional method.

Building and training the system: The phase is totally worked upon by developer of the system, and end user has nothing to do with it. In this phase, we split the dataset into training dataset and test dataset, and then trained the models using training dataset.

Testing the models: In this phase we tested the accuracy of the models with the test dataset that was formed in previous phase and the most accurate model is figured out.

Entering details and prediction: In this phase, the end user comes into picture. He/she enters the details of blood test report using GUI of the application. The application then

matches the details with the training dataset of the most accurate model, and then predicts final result displaying, Risk or No Risk on the screen.

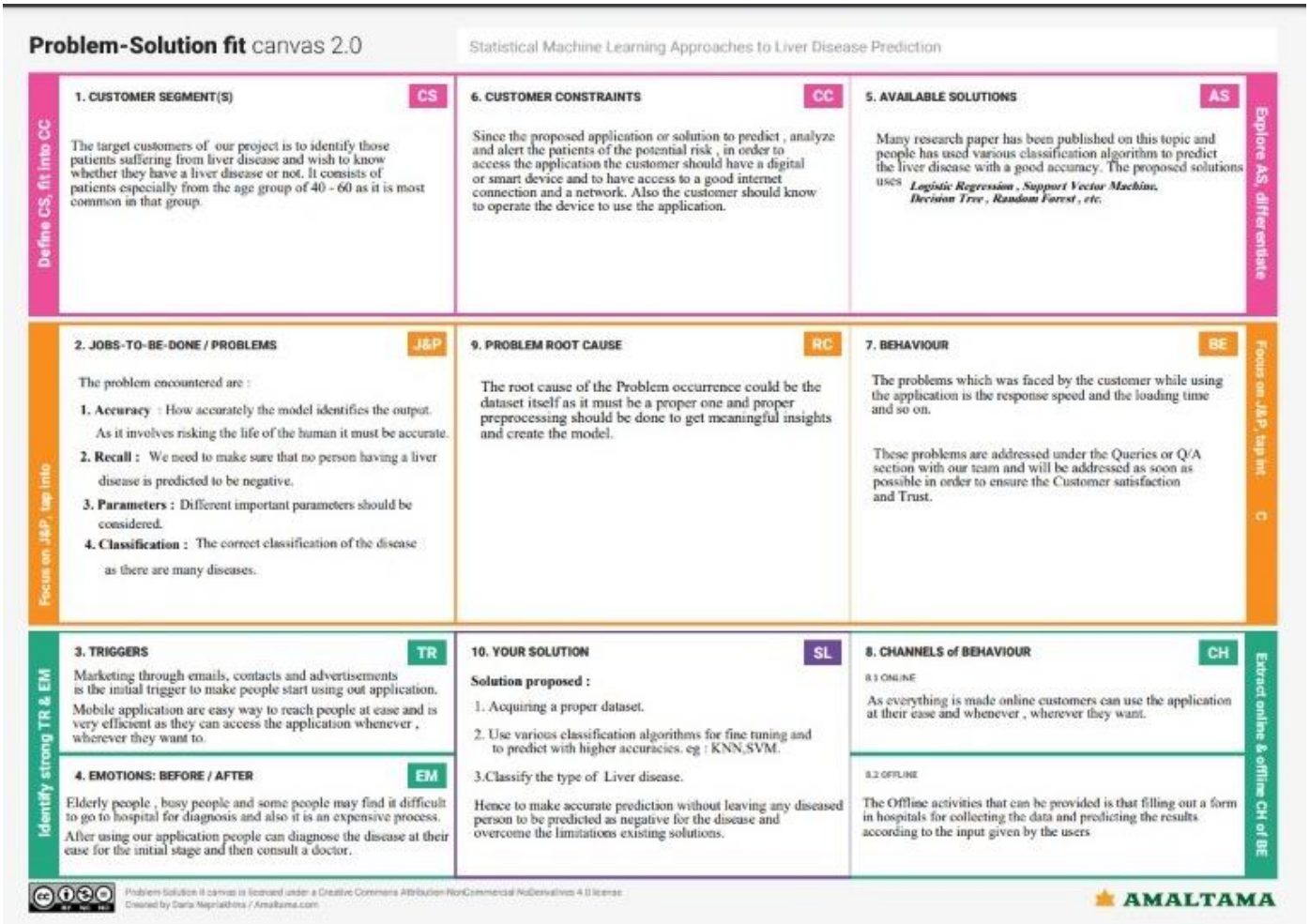
Diagrammatic representations:
Use case diagram: As we can see from the use case diagram first the user enters the blood test details and desktop app takes it as an input and predicts the output based on trained accuracy model and displays the result to the user whether the person is at the risk of liver disease or not.

Scope: The scope is to create a model that predicts whether a person has a liver disease or not using Neural networks and Machine Learning.

Business Model: The application created can be used by the medical sector for faster prediction of liver disease rather than using traditional methods that take a long time to provide results.

Scalability of the System: The model can be integrated with equipment's that measure different levels in the patient's blood and can be used to immediately predict the results.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

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
FR-2	User Confirmation	Confirmation via Email Confirmation via Password

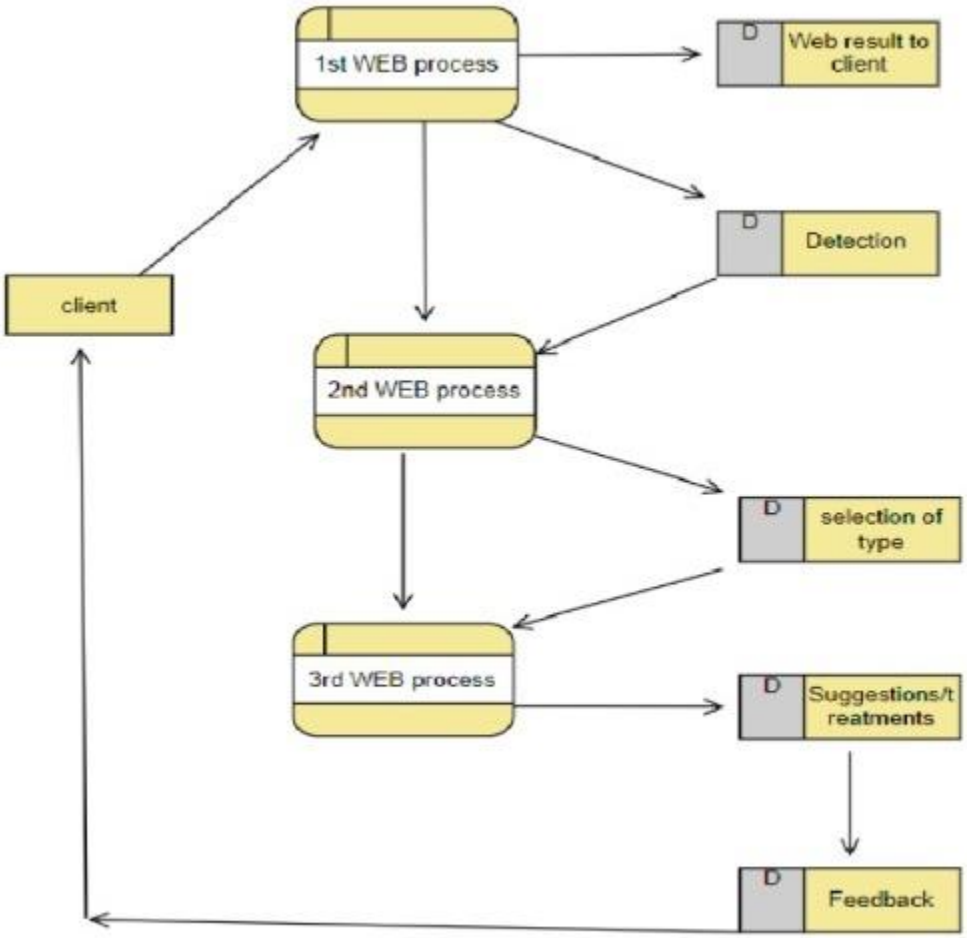
FR-3	User Input	Get necessary details for prediction
FR-4	Data Processing	Data Cleaning Data Scaling Augmentation Feature selection
FR-5	Prediction	Predicting whether the user has Liver disease or not and its type

4.2 Non-functional Requirements:

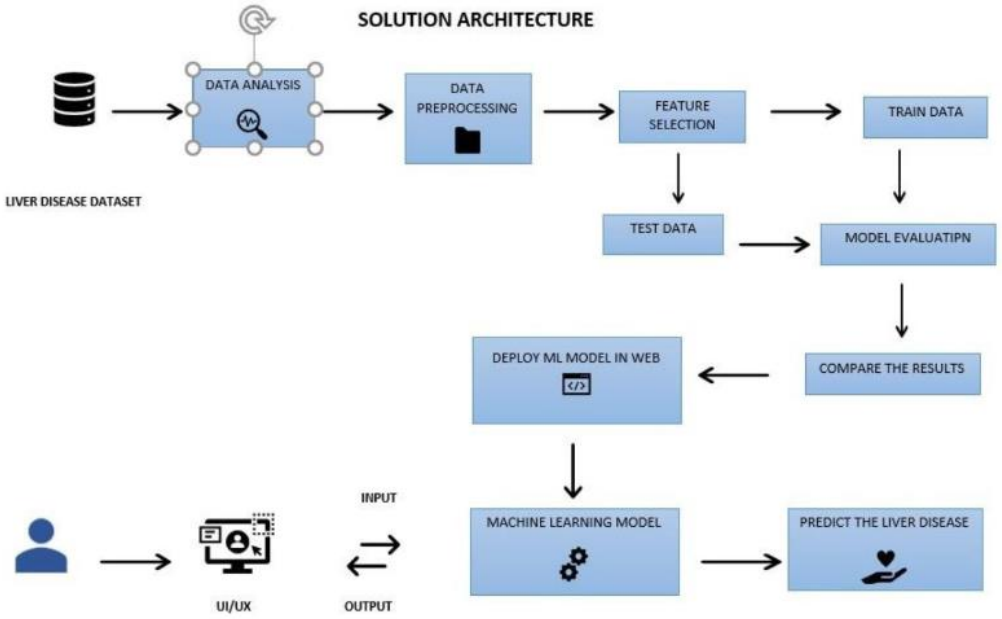
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Defines the difficulty faced by the user to learn and operate the application.
NFR-2	**Security**	Ensures that the software is protected from unauthorized access and stored data.
NFR-3	**Reliability**	Reliability defines how likely it is for the software to work without failure for a given period. It decreases with because of the bugs, the software failures, system failures and the system requirements.
NFR-4	**Performance**	It is a quality attribute that indicates how the app is functioning and how responsive the app is to the end-user.
NFR-5	**Scalability**	It the ability of an application to handle a growing number of users and load, without compromising on performance and causing disruptions to user experience.
NFR-6	**Availability**	It is the extent to which an application is operational, functional period of functionality and usable for completing or fulfilling a user’s or business's requirements.

5.PROJECT DESIGN
5.1 Data Flow Diagrams



5.2 Solution and Technical Architecture



Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	A Web page which gets user input and send it tothe backend for predicting the given input data	HTML, CSS, JavaScript
2.	Predicting Model	Model which takes user input and predict whetherthe person have liver disease or not	Python, NumPy, Pandas, Scikit-learn
3.	Web Server	A web server which serves static HTML user interface files and uses Predicting ML model to process output and send back to the client	Python, Flask
4.	Machine Learning Model	The model used for classify whether the personhave liver disease or not	Support Vector Machine Model
5.	Cloud Deployment	The ML model is bind with web server and deployed in to the IBM cloud	IBM cloud, IBM Watson Studio

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	There are several opensource frameworks used for data pre-processing, data analysis, Model building, pickling and web servers	NumPy, Pandas, Seaborn, Scikit-learn , Pickle, Flask
2.	Security Implementations	Since no user data is stored in the server, there is no security issues in the application side	-
3.	Scalable Architecture	It is a monolithic architecture and, if needed the model which is used to predict can be developed separately as a microservice	Microservices using Docker and Kubernetes
4.	Availability	If the load increases a load balancer can be used to handle the huge request	Nginx Server, Load Balancer
5.	Performance	The performance is still good and has no need the interference of external CDNs, It can able to handle adequate amount of network requests	-

5.3 User Stories:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
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
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	10	High
Sprint-2	Input Necessary Details	USN-4	As a user, I can give Input Details to Predict Likeliness of Liver Disease.	15	High
Sprint-2	Data pre-processing	USN-5	Transform raw data into suitable format for prediction.	5	High
Sprint-3	Prediction of Liver Disease	USN-6	As a user, I can predict Liver Disease using machine learning model.	15	High
Sprint-3		USN-7	As a user, I can get accurate prediction of liver disease.	5	Medium
Sprint-4	Review	USN-8	As a user, I can give feedback of the application.	20	High

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	Kenny Mathew
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Kenny Mathew
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	10	High	Sankeerth S
Sprint-2	Input Necessary Details	USN-4	As a user, I can give Input Details to Predict Likeliness of Liver Disease.	15	High	Sankeerth S
Sprint-2	Data pre-processing	USN-5	Transform raw data into suitable format for prediction.	5	High	Aarthi A
Sprint-3	Prediction of Liver Disease	USN-6	As a user, I can predict Liver Disease using machine learning model.	15	High	Aarthi A
Sprint-3		USN-7	As a user, I can get accurate prediction of liver disease.	5	Medium	P A Mohammed Arshad
Sprint-4	Review	USN-8	As a user, I can give feedback of the application.	20	High	P A Mohammed Arshad

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	18	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	17	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	18	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	17	19 Nov 2022

7. CODING & SOLUTIONING

The application being proposed here uses concept of machine learning, and the models are first trained, then tested. The user can interact with the system through the developed web application. At first, the system asks you to register for the application by entering email id, password and confirming my password. One we have register; we will get a confirmation email. Then log in to the application by entering the email and password. A Dashboard will be opened, the registered user can access the dashboard and can also retrieve information anywhere. The user can navigate through different pages using the dashboard. For the prediction process, the user should upload the image that is required for finding whether liver disease is there are not. The main feature is the result of the tested data will be generated as report in the form of PDF and stored in the user login and also it will automatically download to the user system.

PREDICTION

app.py

```python
from flask import Flask, render_template, request, redirect, url_for, flash
import pickle

app = Flask(__name__)
model = pickle.load(open('liver.pkl', 'rb'))

@app.route("/")
def login():
    return render_template("login.html")
@app.route('/home')
def my_form():
    return render_template("index.html")

@app.route("/predict",methods=["POST"])
def predict():
    Age=int(request.form['Age'])
    Gender=int(request.form['Gender'])
    Total_Bilirubin=float(request.form['Total_Bilirubin'])
    Direct_Bilirubin=float(request.form['Direct_Bilirubin'])
    Alkaline_Phosphotase=int(request.form['Alkaline_Phosphotase'])
    Alamine_Aminotransferase=int(request.form['Alamine_Aminotransferase'])
    Aspartate_Aminotransferase=int(request.form['Aspartate_Aminotransferase'])
    Total_Protiens=float(request.form['Total_Protiens'])
    Albumin=float(request.form['Albumin'])
    Albumin_and_Globulin_Ratio=float(request.form['Albumin_and_Globulin_Ratio'])
    pre=model.predict([[Age,Gender,Total_Bilirubin,Direct_Bilirubin,Alkaline_Phosphotase,Alamine_Aminotransferase,Aspartate_Aminotransferase,Total_Protiens,Albumin
    pre=str(pre[0])
    if pre=="0":
        return render_template("Positive.html")
    else:
        return render_template("Negative.html")



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

index.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">
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="stylesheet" href=".\style.css" />
    <title>Liver Disease Predictor</title>
</head>
<body class="text-center h-full w-full">
    <div class="w-full h-100 bg-gradient-to-r from-orange-500 via-red-500 to-yellow-500 background-animate">

        <nav class="flex items-center justify-center flex content-center">


            <h1 class="text-center text-7xl mt-8 font-semibold text-teal-100 ">Liver Disease Predictor</h1>
        </nav>
    </div>

    <div class="w-full h-100 bg-gradient-to-r from-orange-500 via-red-500 to-yellow-500 background-animate">
        <section class="px-8 py-12 text-slate-800  text-2xl font-sans ">
            Liver diseases avert the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease
            using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of
            liver disease at an early stage is a complex task for the doctors. The main objective of this project is to analyze the various parameters pertinent to a liver disease
            and predict using classification algorithms whether a disease is present or not.
        </section>


        <!-- <button id="#predict" class="absolute bottom-4  left-1/2 bg-red-300 p-3 rounded-lg text-2xl ">Check Your Liver Health</button> -->
    </div>

    <div class="w-full h-100 bg-gradient-to-r from-orange-500 via-red-500 to-yellow-500 background-animate" id="predict">
        <h3 class="text-center px-12 mb-12  font-semibold text-3xl">Check your Liver Health</h3>

        <form class="grid grid-cols-2 gap-8 px-10" action="{{ url_for('predict') }}" method="POST">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Age" placeholder="Age " name="Age" required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Gender" placeholder="Gender (Male:1 , female:0)" name="Gender" required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Total_Bilirubin" placeholder="Total Bilirubin" name="Total_Bilirubin" required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Direct_Bilirubin" placeholder="Direct Bilirubin" name="Direct_Bilirubin" required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Alkaline_Phosphotase" placeholder="Alkaline Phosphotase" name="Alkaline_Phosphotase"required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Alamine_Aminotransferase" placeholder="Alamine Aminotransferase" name="Alamine_Aminotransferase"
            required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Aspartate_Aminotransferase" placeholder="Aspartate Aminotransferase" name="Aspartate_Aminotransferase"
            required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Total_Protiens" placeholder="Total Protiens" name="Total_Protiens" required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Albumin" placeholder="Albumin " name="Albumin" required="required">
            <input class="text-2xl px-4 py-2 rounded-lg" type="text" id="Albumin_and_Globulin_Ratio" placeholder="Albumin and Globulin Ratio" name="Albumin_and_Globulin_Ratio"
            required="required">

            <input id="button" class=" bg-yellow-500 w-40 border-solid px-6 py-3 col-span-2 rounded-2xl text-2xl hover:cursor-pointer " type="submit">
        </form>
```

login.html

```
<!DOCTYPE html>
<html>
<head>
<title>Login Form</title>
<style>
 body
 {
   margin:0px; background: linear-gradient(90deg, #cc2b5e, #c779d0, #753a88); color:#f7f7f7; font-family:Arial, Helvetica, sans-serif;
 }
 #main
 {
   width:600px; height:260px; margin-left:auto; margin-right:auto; border-radius:5px; padding-left:10px; margin-top:100px;
   border-top:3px double #f1f1f1; border-bottom:3px double #f1f1f1; padding-top:20px;
 }
 #main table
 {
   font-family:"Comic Sans MS", cursive;
 }
 /* css code for textbox */
 #main .tb
 {
   height:28px; width:230px; border:1px solid black; color:black; font-weight:bold; border-left:5px solid #f7f7f7; opacity:0.9;
 }

  /* css code for button*/
  #main .btn
  {
   width:80px; height:32px; outline:none;  font-weight:bold; border:0px solid black; text-shadow: 0px 0.5px 0.5px #fff;
   border-radius: 2px; font-weight: 600; color: black; letter-spacing: 1px; font-size:14px; -webkit-transition: 1s; -moz-transition: 1s; transition: 1s;
  }

  #main .btn:hover
  {
   background: linear-gradient(90deg, #cc2b5e, #c779d0, #753a88); outline:none;  border-radius: 2px; color:#f1f1f1; border:1px solid #f1f1f1;
  }
     #title{
         text-align: center;
         font-size: 35px;
     }
</style>
<script>

    function login()
  {
      var uname = document.getElementById("email").value;
      var pwd = document.getElementById("pwd1").value;
      var filter = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;
      if(uname =='')
      {
          alert("please enter user name.");
      }
      else if(pwd=='')
      {
          alert("enter the password");
      }
      else if(!filter.test(uname))
      {
```

```
      else if(!filter.test(uname))
      {
          alert("Enter valid email id.");
      }
      else if(pwd.length < 6 || pwd.length > 6)
      {
          alert("Password min and max length is 6.");
      }
      else
      {
  alert('Thank You for Login & You will be redirected to the application');
//Redirecting to other page or webste code or you can set your own html page.
    window.location.href = "{{ url_for('my_form') }}";
          }
  }
  //Reset Inputfield code.
  function clearFunc()
  {
      document.getElementById("email").value="";
      document.getElementById("pwd1").value="";
  }
</script>
</head>

  <body>
  <!-- Main div code -->
  <div id="main">
  <div class="h-tag">
  <h2 id="title" >Liver Disease Prediction</h2>
  </div>
  <!-- Login box -->
  <div class="login">
  <table cellspacing="2" align="center" cellpadding="8" border="0">
  <tr>
  <td>Enter User Name :</td>
  <td><input type="text" placeholder="Enter user name here" id="email" class="tb" /></td>
  </tr>
  <tr>
  <td>Enter Password :</td>
  <td><input type="password" placeholder="Enter Password here" id="pwd1" class="tb" /></td>
  </tr>
  <tr>
  <td></td>
  <td>
  <input type="submit" value="Reset" onclick="clearFunc()" class="btn" />
  <input type="submit" value="Login" class="btn" onClick="login()" /></td>
  </tr>
  </table>
  </div>
   <!-- login box div ending here.. -->
  </div>
  <!-- Main div ending here... -->

</body>
</html>
```

negative.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">
    <script src="https://cdn.tailwindcss.com"></script>
    <title>Result</title>
</head>
<body class="text-center w-full h-100 bg-gradient-to-r from-orange-500 via-red-500 to-yellow-500 background-animate flex justify-center items-center">
    <div class="bg-white border-3 border-black rounded-3xl flex-row p-8 mt-10 w-3/4 justify-center items-center">
        <h1 class="text-4xl text-center font-semibold">You are not healthy!!</h1>
        <br><br>
        <!-- <img class="w-3/4 ml-6" src="./images/healthy.jpg" alt="healthy"> -->

        <br><br>

        <button class="bg-yellow-400 px-8 py-3 rounded-3xl "><a href="{{ url_for('my_form') }}">Back to Home</a></button>
    </div>


</body>
</html>
```

positive.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">
    <script src="https://cdn.tailwindcss.com"></script>
    <title>Result</title>
</head>
<body class="text-center w-full h-100 bg-gradient-to-r from-orange-500 via-red-500 to-yellow-500 background-animate flex justify-center items-center">
    <div class="bg-white border-3 border-black rounded-3xl flex-row p-8 mt-10 w-3/4 justify-center items-center">
        <h1 class="text-4xl text-center font-semibold">You are healthy!!</h1>
        <br><br>
        <!-- <img class="w-3/4 ml-6" src="./images/healthy.jpg" alt="healthy"> -->

        <br><br>
        <button class="bg-yellow-400 px-8 py-3 rounded-3xl "><a href="{{ url_for('my_form') }}">Back to Home</a></button>
    </div>


</body>
</html>
```

SPRINT-1: PROJECT DELIVERABLES
Data Preparation and Pre-process Reading the dataset:

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
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='v5beUinwA8mQv6o5ac52k0LKMOUAt43AEPh2bqE8qkqS',
    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 = 'statisticalmachinelearningapproac-donotdelete-pr-txlpxwdlezxkwa'
object_key = 'indian_liver_patient.xlsx'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']

data = pd.read_excel(body.read())
data.head()
```

Out[2]:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40

Dataset Description:

In [4]:
```
data.info()
```

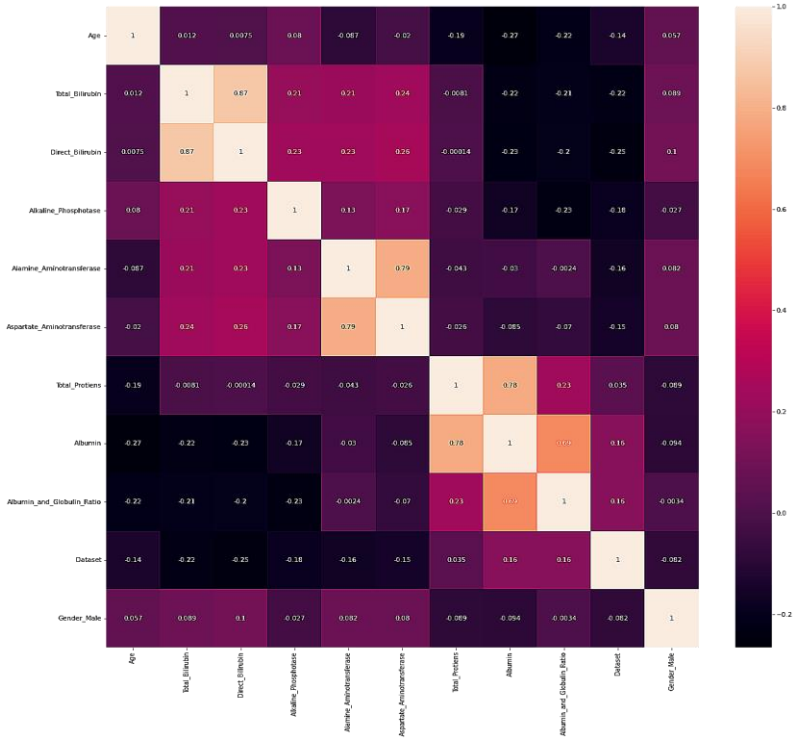
```
RangeIndex: 583 entries, 0 to 582
Data columns (total 11 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Age                         583 non-null    int64
 1   Gender                      583 non-null    object
 2   Total_Bilirubin             583 non-null    float64
 3   Direct_Bilirubin            583 non-null    float64
 4   Alkaline_Phosphotase        583 non-null    int64
 5   Alamine_Aminotransferase    583 non-null    int64
 6   Aspartate_Aminotransferase  583 non-null    int64
 7   Total_Protiens              583 non-null    float64
 8   Albumin                     583 non-null    float64
 9   Albumin_and_Globulin_Ratio  579 non-null    float64
 10  Dataset                     583 non-null    int64
dtypes: float64(5), int64(5), object(1)
memory usage: 50.2+ KB
```

In [5]:
```
data.describe()
```

Out[5]:

	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_R
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	579.00
mean	44.746141	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.141852	0.94
std	16.189833	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.795519	0.31
min	4.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.900000	0.30
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.600000	0.70
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.100000	0.93
75%	58.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.800000	1.10
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.500000	2.80

Correlation map:

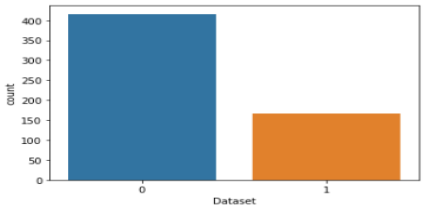


EDA: Exploratory Data Analysis:

In [11]:
```
import seaborn as sns
sns.countplot('Dataset', data = data)
```

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or mis
interpretation.
  warnings.warn(
```

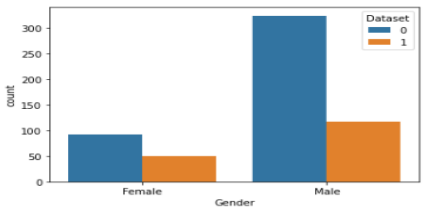
Out[11]:



In [12]:
```
sns.countplot('Gender', data = data, hue = 'Dataset')
```

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or mis
interpretation.
  warnings.warn(
```

Out[12]:



Multi – variate Analysis:

8. TESTING

Test Scenarios

S. No	Testing Scenarios
1	Verify user is able to see home page
2	Verify user is able to view the about section of the home page
3	Verify user is able to navigate to predict the liver disease page
4	Verify user is able to come back to home page from result and prediction page
5	Verify predict page elements
	Input Fields
1	Verify user is able to enter float values in input field of the prediction
2	Verify user is able to enter number in some of the input field in prediction page
3	Verify user is able to predict only after entering all the fields

8.1 Test Cases

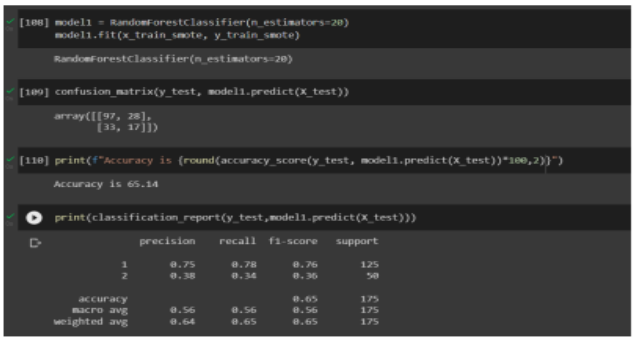
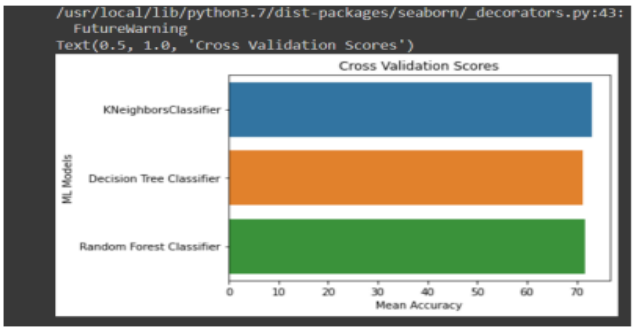
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_001	UI	Home Page	Verify user is able to see all the buttons and all the contents of the page		1.Enter URL and click go 2.Explore the whole page to verify that all the fonts are in correct size and readable	http://127.0.0.1:5000/	All the elements and contents should be readable font color and size and with uniformity	Working as expected	Pass				Lavanya M
HomePage_TC_002	Functional	Home Page	Verify user is able to see the about section of the page when clicked about on the home page		1.Enter URL and click go 2.Click on about hyperlink on the page 3.Verify it moves down to about section of the page	http://127.0.0.1:5000/	About section of the home page should display	Working as expected	Pass				Lavanya M
HomePage_TC_003	Functional	Home Page	Verify the prediction form page is displayed when the predict button is clicked		1.Enter URL and click go 2.Click on predict button on the home page 3.Verify the prediction page is displayed.	http://127.0.0.1:5000/	Application should show prediction page to enter details	Working as expected	Pass				Lavanya M
PredictionPage_TC_001	Functional	Prediction page	Verify user is able to enter values in the input fields in both float and integer values.		1.Enter URL(http://127.0.0.1:5000) and click go 2.Click on Predict button 3.Enter the details in the input field box in float and integer wherever necessary 4.click predict button.	Age: 17 Gender: 0 Total Bilirubin: 0.9 Direct Bilirubin: 0.3 Alkaline Phosphatase: 202 Alamine Aminotransferase: 22 Asparate Aminotransferase: 19 Albumin: 7.4 Total Proteins: 4.1 Albumin and Globulin Ratio: 1.2	User should navigate to result page	Working as expected	Pass				Lavanya M
PredictionPage_TC_002	Functional	Prediction page	Verify user can view the predicted result only after entering all the required field data		1.Enter URL(http://127.0.0.1:5000) and click go 2.Click on Predict button 3.Enter the details in the input field box in float and integer wherever necessary 4.click predict button.	Age: 17 Gender: 0 Total Bilirubin: 0.9 Direct Bilirubin: 0.3 Alkaline Phosphatase: 202 Alamine Aminotransferase: 22 Asparate Aminotransferase: 19 Albumin: - Total Proteins: 4.1 Albumin and Globulin Ratio: 1.2	Application should show "Enter all the required fields"(Albumin)	Working as expected	Pass				Lavanya M
PredictionPage_TC_003		Login page	Verify user is able to click the Home and predict again button		1.Enter URL(http://127.0.0.1:5000) and click go 2.Click on Predict button 3.Enter the details in the input field box in float and integer wherever necessary 4.click predict button. 5. View the result 6. Click the predict button/home button		Application should move to Prediction page or home page respectively.	Working as expected	Pass				Lavanya M

8.2 User Acceptance Testing

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	2	0	0	2
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	1	0	0	1
Exception Reporting	1	0	0	1
Final Report Output	3	0	0	3
Version Control	1	0	0	1

9. RESULTS

9.1 Performance Metrics

S. No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix -, Accuracy Score- & Classification Report -	 **FIG 9.1.1 Metrics of Classification Model**
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	 **FIG 9.1.2 Hyperparameter tuning**

10. ADVANTAGES AND DISADVANTAGES

Advantages

- This helps in early diagnosis of liver disease.
- It makes the process simple and easier.
- We are able to monitor the patient in an effective manner.
- We can also help patients who are not in a condition to directly consult a doctor.

Disadvantages

- There are inaccuracies which cannot be avoided.
- For people who have difficulty in accessing internet or those who do not have electronic gadgets, this is not a feasible solution.

11.CONCLUSION

In this paper, we proposed and built a machine learning based on a hybrid classifier to be used as a classification model for liver diseases diagnosis to improve performance and experts to identify the chances of disease and conscious prescription of further treatment healthcare and examinations. In future work. 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.

12. FUTURE SCOPE

In future, along with this analysis of the input data, using Machine Learning algorithms for predictive analysis may be more effective. The predictions and classification of user data can be done by using the ML algorithms easily and a full-fledged application can be created for analysis and predictions of the Patient data can be done near future. And the results can be directly sent to the patient email address without delay.

13. APPENDIX

Source code

https://github.com/IBM-EPBL/IBM-Project-42415-1660661955/tree/main/Final%20Deliverables/Source%20code

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-42415-1660661955

Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-42415-1660661955/tree/main/Final%20Deliverables/Demo%20video