**INTRODUCTION**

* 1. **OVERVIEW**

Liver diseases averts the normal function of the liver. This disease is caused by an assortment of elements that harm the liver. Diagnosis of liver infection at the preliminary stage is important for better treatment. In today’s scenario devices like sensors are used for detection of infections. Accurate classification techniques are required for automatic identification of disease samples. This disease diagnosis is very costly and complicated. Therefore, the goal of this work is to evaluate the performance of different Machine Learning algorithms in order to reduce the high cost of liver disease diagnosis. 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.

In this project we will analyze the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This project compares various classification algorithms such as Random Forest, Logistic Regression, KNN and ANN Algorithm with an aim to identify the best technique. Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver disease and can be recommended to the user.

* 1. **PURPOSE**

In India, delays in diagnosing diseases are a major problem due to a lack of medical professionals. The typical scenario, which is mainly in rural and slightly urban areas:

* A patient who sees a doctor with certain symptoms.  
  The doctor will perform some tests, such as blood and urine tests, depending on the symptoms.
* The patient undergoes the above tests in the analytical laboratory.
* The patient takes the reports back to the hospital, where they are examined and diagnosed.

The goal of this project is to reduce some of the delays caused by unnecessary detours between the hospital and the pathology laboratory.

**PROBLEM DEFINITION AND DESIGN THINKING**

**2.1 EMPATHY MAP**

An empathy map is a square divided into four quadrants with the user or client in the middle. Each of the four quadrants comprises a category that helps us delve into the mind of the user. The four empathy map quadrants look at what the user says, thinks, feels, and does.

WHY IS THIS SO HARD?

I DON’T KNOW WHAT TO DO TO IMPROVE MY LIVER HEALTH

WHAT ELSE AM I MISSING?

I ‘M WORRIED ABOUT MY LIVER HEALTH

**THINKS**

**SAYS**

**FEELS**

**DOES**

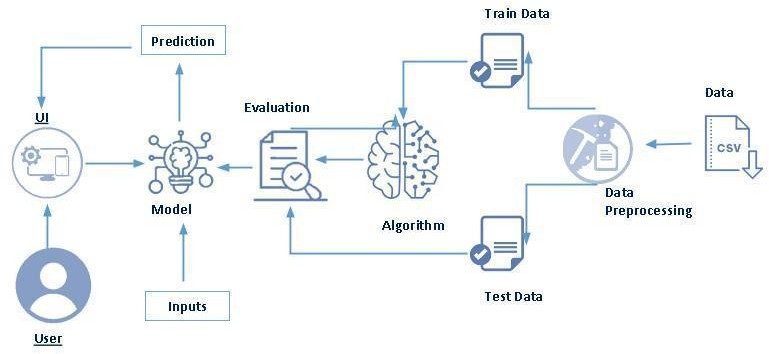
ANXIOUS

MORE RESEARCH

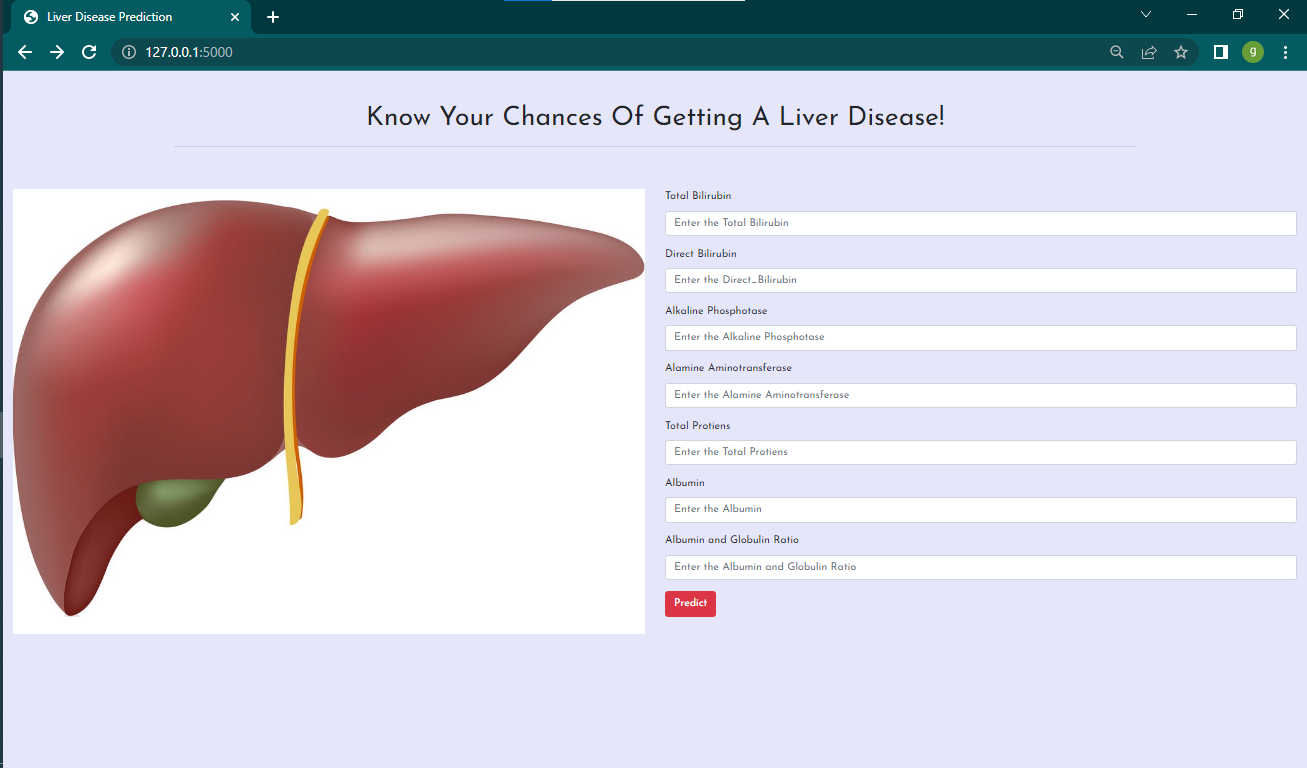
FEAR

COMPARES PRODUCTS

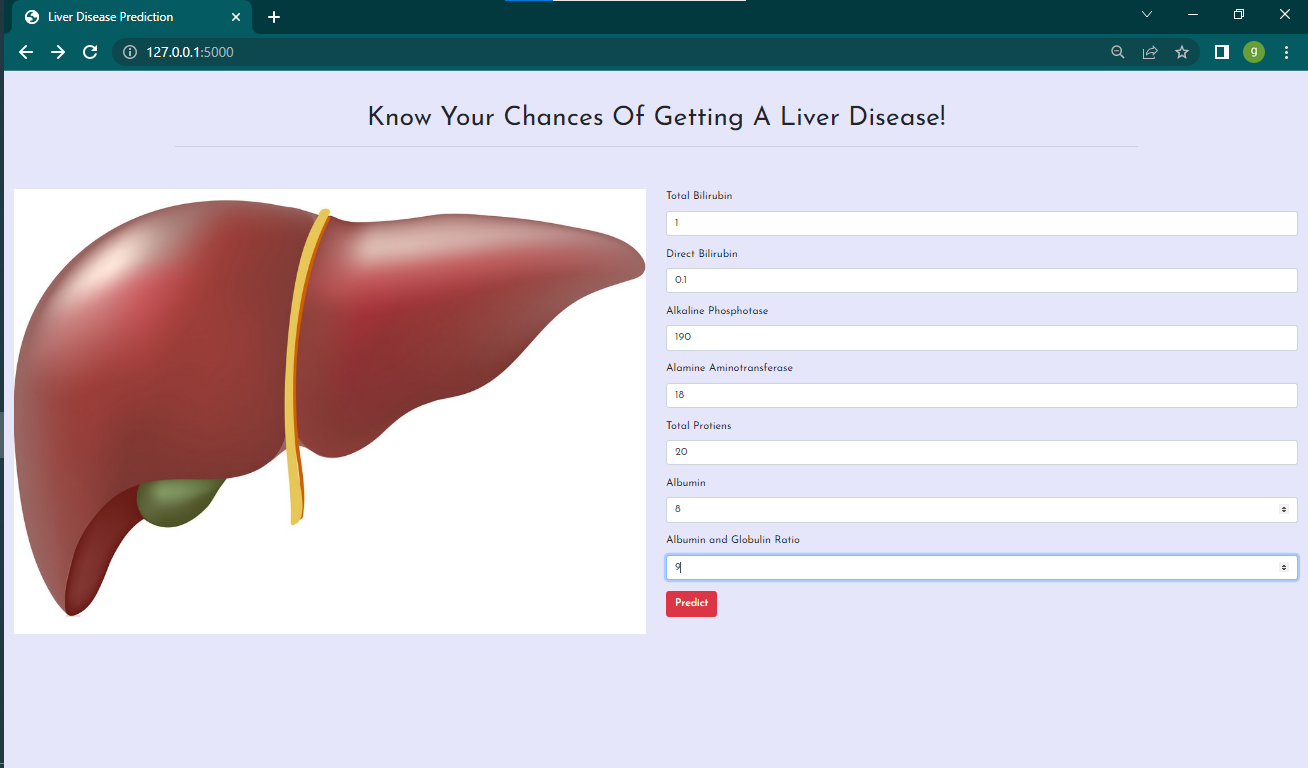
**2.2 IDEATION MAP**



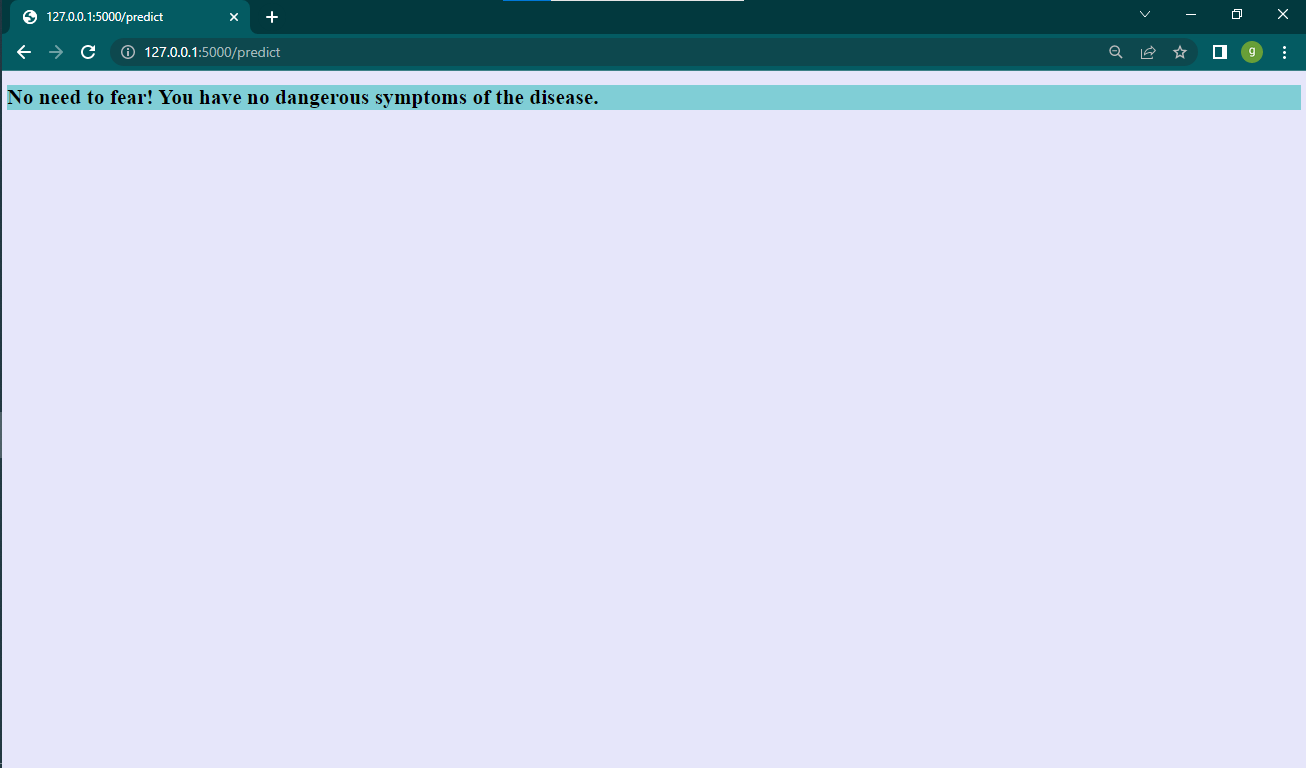
**RESULT**

****

**HOME PAGE**

****

**INPUT PAGE**

****

**OUTPUT (PREDICTION)**

**ADVANTAGES AND DISADVANTAGES**

**4.1 ADVANTAGES**

* This is used to predict the liver disease of a patient on the basis of dataset available.
* The system processes the symptoms provided by the user as input and gives the output as the probability of the disease.
* It will also recommend necessary precautionary measures required to treat the predicted disease.
* It saves time and money.
* Using predictive analytics in healthcare can improve the quality of healthcare, collect more clinical data for personalized treatment, and successfully diagnose the medical condition of individual patient.
* This could help to improve patient care and the safety and effectiveness of medical procedures.

**4.2 DISADVANTAGES**

* It will cause delays in providing the output. So, machine learning significantly depends on the data and its quality.
* The data that machines process remains huge in quantity and differs greatly.
* It requires massive and expensive resources and high-quality expertise to set up that quality of infrastructure. Trials runs are costly as they would cost in terms of time and expenses.
* It will have some degree of inaccuracy. For a high degree of accuracy, algorithms should be developed so that they give reliable results.

**APPLICATION**

**5.1 APPLICATION**

* Machine learning algorithm can be used in medical imaging (such as X-rays or MRI scans) using pattern recognition to look for patterns that indicate a particular disease. This could potentially help doctors make quicker, more accurate diagnoses.
* Can be used in health monitoring system.
* No need of expensive devices.
* Patients who are not willing to take test, can use this system.
* Can be used in daily checkup.

**CONCLUSION**

**6.1 CONCLUSION**

With the passage of time, diseases of the liver are becoming increasingly widespread. These are solely going to get worse in the future, thanks to ongoing technological improvements. Despite the truth that people are becoming more health-conscious and enrolling in yoga and dancing classes, the sedentary lifestyle and facilities that are continuously being delivered and improved will proceed to be an issue. As a result, in this situation, our project will be rather recommended to society.

**FUTURE SCOPE**

**7.1 FUTURE SCOPE**

A future selection-based machine learning algorithm is proposed to predict three chronic diseases, namely, diabetes, heart attack, and cancer. Machine learning techniques, such as natural language processing (NLP) and image processing, help streamline data collection and convert data into a standard format. This can lead to enhancement in identifying clinical patterns and assist with better predictions. Machine learning with Quantum can improve the analysis of data and get more profound insights.

**APPENDIX**

**8.1 SOURCE CODE**

from flask import Flask, render\_template, url\_for, flash, redirect

import joblib

import numpy as np

from flask import request

app = Flask(\_\_name\_\_, template\_folder = "templates")

@app.route("/")

@app.route("/liver")

def cancer():

return render\_template("liver.html")

def ValuePredictor(to\_predict\_list, size):

to\_predict = np.array(to\_predict\_list).reshape(1, size)

if (size == 7):

loaded\_model = joblib.load(r"liver\_model.pkl")

result = loaded\_model.predict(to\_predict)

return result[0]

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

def predict():

if request.method == "POST":

to\_predict\_list = request.form.to\_dict()

to\_predict\_list = list(to\_predict\_list.values())

to\_predict\_list = list(map(float, to\_predict\_list))

if (len(to\_predict\_list) == 7):

result = ValuePredictor(to\_predict\_list, 7)

if(int(result)==1):

prediction = "Sorry, you have chances of getting the disease. Please consult the doctor immediately"

else:

prediction = "No need to fear! You have no dangerous symptoms of the disease."

return(render\_template("result.html", prediction\_text=prediction))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)