

1. INTRODUCTION

1.1 OVERVIEW

Liver diseases averts 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 analyse 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 examines data from liver patients concentrating on relationships between a key list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease. Here we are building a model by applying various machine learning algorithms find the best accurate model. And integrate to flask based web application. User can predict the disease by entering parameters in the web application.

1.2 PURPOSE

Using IBM AutoAI, we automate all of the tasks involved in building predictive models for different requirements. You see how AutoAI generates great models quickly, which saves time and effort, and aids in a faster decision-making processes.

Abstraction of this project we are going discuss how to predict risk of liver disease for a person, based on the blood test report results of the user. In this paper, the risk of liver disease was predicted using various machine learning algorithms. The final output was predicted based on the most accurate machine learning algorithm. Based on the accurate model we 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. LITERATURE SURVEY

2.1 EXISTING PROBLEM

There are limitations when dealing with the unpredictable and the unknown. yearly largely in the context of hospital records data where where researchers have applied various machine learning systems to try and tackle the problem of chance of diseases as well as the related more well known ,task of sentiment analysis. Liver patient analysis isn't infallible and isn't appropriate or useful for all situation. Data teams should use liver patient analysis when they understand the business question and have the appropriate data and capabilities to answer that question.

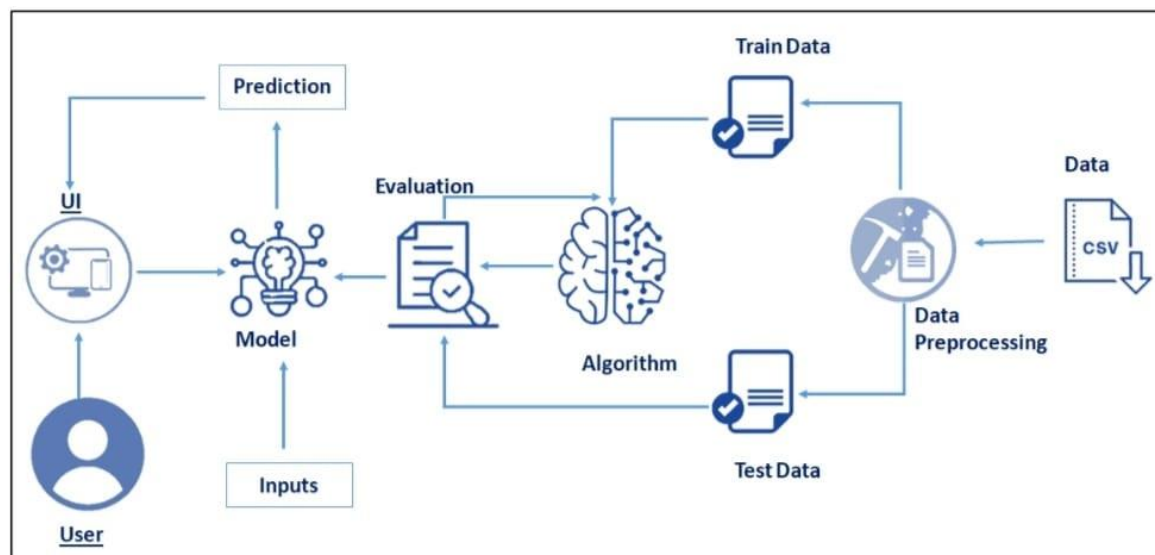
2. 2 PROPOSED SOLUTION

The system being proposed here uses concept of machine 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.

3.THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE/SOFTWARE DESIGNING

IBM Watson Studio - IBM Watson Studio helps data scientists and analysts prepare data and build models at scale across any cloud.

IBM Watson Machine Learning - IBM Watson Machine Learning helps data scientists and developers accelerate AI and machine-learning deployment.

IBM Cloud Object Storage - IBM Cloud Object Storage makes it possible to store practically limitless amounts of data, simply and cost effectively.

Hardware:

Processor : intel core i3

Ram : 4GB

Hard disk : 30GB
Operating System : Windows
Programming : python 3.6+
Software:
1.Anaconda Navigator
2.Jupyter Notebook
3.Spyder Notebook

To build Machine learning models you must require the following packages
Sklearn: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

NumPy: NumPy is a Python package that stands for 'Numerical Python'. It is the core library for

scientific computing, which contains a powerful n-dimensional array object

Pandas: pandas is a fast, powerful, flexible, and easy to use open-source data analysis and

manipulation tool, built on top of the Python programming language.

Matplotlib: It provides an object-oriented API for embedding plots into applications using general-

purpose GUI toolkits

Flask: Web framework used for building Web applications.

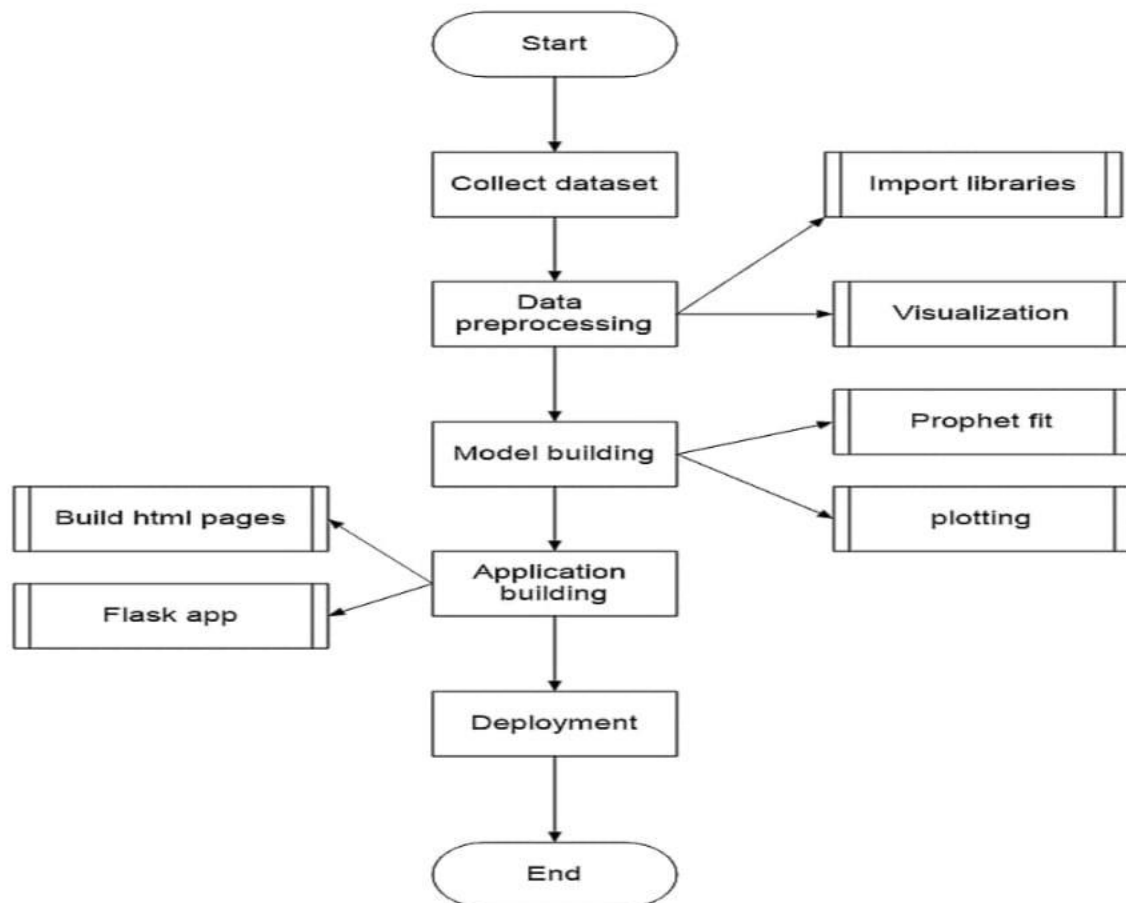
4.EXPERIMENTAL INVESTIGATIONS

Here we are going to build a machine learning model that predicts whether a person has a risk of liver disease or not.

For this purpose we searched for the details of his/her blood test report. Under the category of

Age, Gender, Total_Bilirubin, Direct_Bilirubin, Alkaline_Phosphotase, Alamine_Amino transferase, Aspartate_Aminotransferase, Total_Protiens, Albumin, Albumin_and_Globulin_Ratio.

5.FLOWCHART



6.RESULT

Liver Patient Analysis

Home Goto Predict

Introduction

Liver diseases averts 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 paper is to analyse 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 paper focuses on the related works of various authors on liver disease such that algorithms were implemented using Weka tool that is a machine learning software written in Java. Various attributes that are essential in the prediction of liver disease were examined and the dataset of liver patients were also evaluated. This paper compares various classification algorithms such as Random Forest, Logistic Regression and Separation 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 recommended.

This is our home page where we get to know the summary of the project

Liver Patient Prediction

Age:

Gender:

Enter 0 as male, 1 as female

Total_Bilirubin:

Direct_Bilirubin:

Alkaline_Phosphotase:

Alamine_Aminotransferase:

Aspartate_Aminotransferase:

Total_Protiens:

Albumin:

Albumin_and_Globulin_Ratio:

Predict

Liver Patient Prediction

You have a liver disease problem, You must and should consult a doctor. Take care

As we see the predicted output is displayed on the User Interface

7.ADVANTAGES AND DISADVANTAGES

7.1 ADVANTAGES

No medical expertise required: You dont 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 are 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 dont need to wait for a doctor to come, unlike in traditional method.

7.2 DISADVANTAGES

Prediction are never 100% accurate

It can be time consuming and resource intensive

8.APPLICATIONS

1. Disease identification
2. Personalized treatment
3. Drug discovery
4. Clinical research
5. Radiology

9.CONCLUSION AND FUTURE SCOPE

Diseases related to liver and heart are becoming more and more common with time. With continuous technological advancements, these are only going

to increase in the future. Although people are becoming more conscious of health nowadays and are joining yoga classes, dance classes; still the sedentary lifestyle and luxuries that are continuously being introduced and enhanced; the problem is going to last long.

So, in such a scenario, our project will be extremely helpful to the society. With the dataset that we used for this project, we got 100 % accuracy for SVM model, and though it might be difficult to get such accuracies with very large datasets, from this projects results, one can clearly conclude that we can predict the risk of liver diseases with accuracy of 90 % or more.

Today almost everybody above the age of 12 years has smartphones with them, and so we can incorporate these solutions into an android app or ios app. Also it can be incorporated into a website and these app and website will be highly beneficial for a large section of society.

10.BIBILOGRAPHY

<https://www.ijert.org/liver-disease-prediction-system-using-machine-learning-techniques>

APPENDIX

A.Source Code

```
from flask import Flask, render_template, request,jsonify,render_template #
Flask is a application

# used to run/serve our application

# request is used to access the file which is uploaded by the user in out
application

# render_template is used for rendering the html pages
```

```
import pickle # pickle is used for serializing and de-serializing Python object
structures

import os

import json

import requests

import numpy as np

import pickle

import joblib

import matplotlib

import matplotlib.pyplot as plt

import time

import pandas

API_KEY = "9417L-N8ej71KwqkN_GQdwUO5hFeBAOJO0FasObjrJ_t"

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

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

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

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

app=Flask(__name__) # our flask app

@app.route('/') # rendering the html template
def home():

    return render_template('home.html')

@app.route('/predict') # rendering the html template
def index() :
```



```
return render_template("index.html")
```

```
@app.route('/data_predict', methods=['GET','POST']) # route for our prediction
```

```
def predict():
```

```
    age = request.form['age'] # requesting for age data
```

```
    gender = request.form['gender'] # requesting for gender data
```

```
    tb = request.form['tb'] # requesting for Total_Bilirubin data
```

```
    db = request.form['db'] # requesting for Direct_Bilirubin data
```

```
    ap = request.form['ap'] # requesting for Alkaline_Phosphotase data
```

```
    aa1 = request.form['aa1'] # requesting for Alamine_Aminotransferase data
```

```
    aa2 = request.form['aa2'] # requesting for Aspartate_Aminotransferase data
```

```
    tp = request.form['tp'] # requesting for Total_Protiens data
```

```
    a = request.form['a'] # requesting for Albumin data
```

```
    agr = request.form['agr'] # requesting for Albumin_and_Globulin_Ratio data
```

```
    # coverting data into float format
```

```
    data = [[float(age), float(gender), float(tb), float(db), float(ap), float(aa1), float(aa2), float(tp), float(a), float(agr)]]
```

```
    payload_scoring = {"input_data": [{"field": ['age', 'gender', 'tb', 'db', 'ap', 'aa1', 'aa2', 'tp', 'a', 'agr']], "values": data}}
```

```
    response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/8d97274a-e988-4d2c-b707-7d1d092ed195/predictions?version=2022-03-31', json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
```

```
    print("Scoring response")
```

```
    print(response_scoring.json())
```

```
# loading model which we saved

model = pickle.load(open('C:/Users/id/Desktop/liverpatient/model1.pkl',
'rb'))

prediction= model.predict(data)

if (prediction == 1):

    return render_template('nochance.html', prediction='You have a liver
desease problem, You must and should consult a doctor. Take care')

else:

    return render_template('chance.html', prediction='You dont have a liver
desease problem')

if __name__ == '__main__':

    port=int(os.environ.get('PORT',8000))

    app.run(port=port,debug=True,use_reloader=False)
```