

# LIVER PATIENT ANALYSIS



# 1. INTRODUCTION

## 1.1 overview

In India, delayed diagnosis of diseases is a fundamental problem due to a shortage of medical professionals. A typical scenario, prevalent mostly in rural and somewhat in urban areas is:

1. A patient going to a doctor with certain symptoms.
2. The doctor recommending certain tests like blood test, urine test etc depending on the symptoms.
3. The patient taking the aforementioned tests in an analysis lab.
4. The patient taking the reports back to the reports back to the hospital, where they are examined and the disease is identified.

## 1.2 Purpose

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 purpose of this project is to analyse the parameters of various classification algorithms and compare their predictive accuracies to discover the best classifier for determining the liver disease.

## **2.LITERATURE SURVEY**

### **2.1 Existing Problem (OR) Problem Statement**

Given a dataset containing various attributes of 584 Indian patients, use the features available in the dataset and define a supervised classification algorithm which can identify whether a person is suffering from liver disease or not. This data set contains 416 liver patient records and 167 non- liver patient records. The data set was collected from north east of Andhra Pradesh, India. This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age "90".

### **2.2 Proposed Solution**

This is a classic example of supervised learning. We have been provided with a fixed number of features for each data point, and our aim will be to train a variety of Supervised Learning algorithms on this data, so that, when a new data point arises, our best performing classifier can be used to categorize the data point as a positive example or negative. Exact details of the number and types of algorithms used for training is included in the 'Algorithms and Techniques' sub-section of the 'Analysis' part.

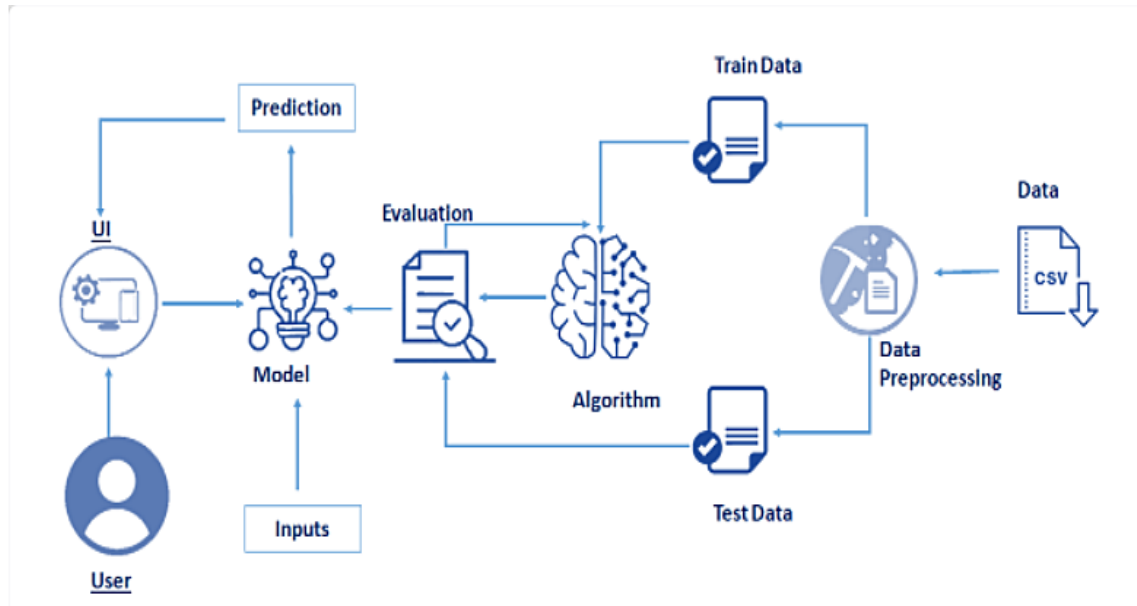
This project focuses on the related works of various authors on liver disease such that algorithms were implemented using Jupyter that is a machine learning software written in Python. Various attributes that are essential in the prediction of liver disease were examined and the dataset of liver patients were also evaluated. This project compares various classification algorithms such as Random Forest, Support Vector Machine and KNN classification 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 to the user.

Later by using Flask app create html files and create an user interface to display whether the patient has liver problem or not.

### 3.THEORITICAL ANALYSIS

#### 3.1 Block Diagram



#### 3.2 Hardware / Software designing

The following is the Hardware required to complete this project:

- Internet connection to download and activate
- Administration access to install and run Anaconda Navigator
- Minimum 10GB free disk space
- Windows 8.1 or 10 (64-bit or 32-bit version) OR Cloud: Get started free, \*Cloud account required.

Minimum System Requirements To run Office Excel 2013, your computer needs to meet the following minimum hardware requirements:

- 500 megahertz (MHz)
- 256 megabytes (MB) RAM
- 1.5 gigabytes (GB) available space
- 1024x768 or higher resolution monitor

The following are the software s required for the project:

- Jupyter Notebook
- Spyder
- Microsoft Excel 2013

## **4.EXPERIMENTAL INVESTIGATIONS**

Coming to analysis or investigations three supervised learning approaches are selected for this problem. Care is taken that all these approaches are fundamentally different from each other, so that we can cover as wide an umbrella as possible in term of possible approaches.

For each algorithm, we will try out different values of a few hyper parameters to arrive at the best possible classifier. This will be carried out with the help of grid search cross validation technique.

There are several Machine learning algorithms to be used depending on the data you are going to process such images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have may be classification algorithms and Regression algorithms.

### **(1) Support Vector Machine**

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. Support Vectors are simply the coordinates of individual observation. The goal of a support vector machine is not only to draw hyperplanes and divide data points, but to draw the hyperplane the separates data points with the largest margin, or with the most space between the dividing line and any given data point.

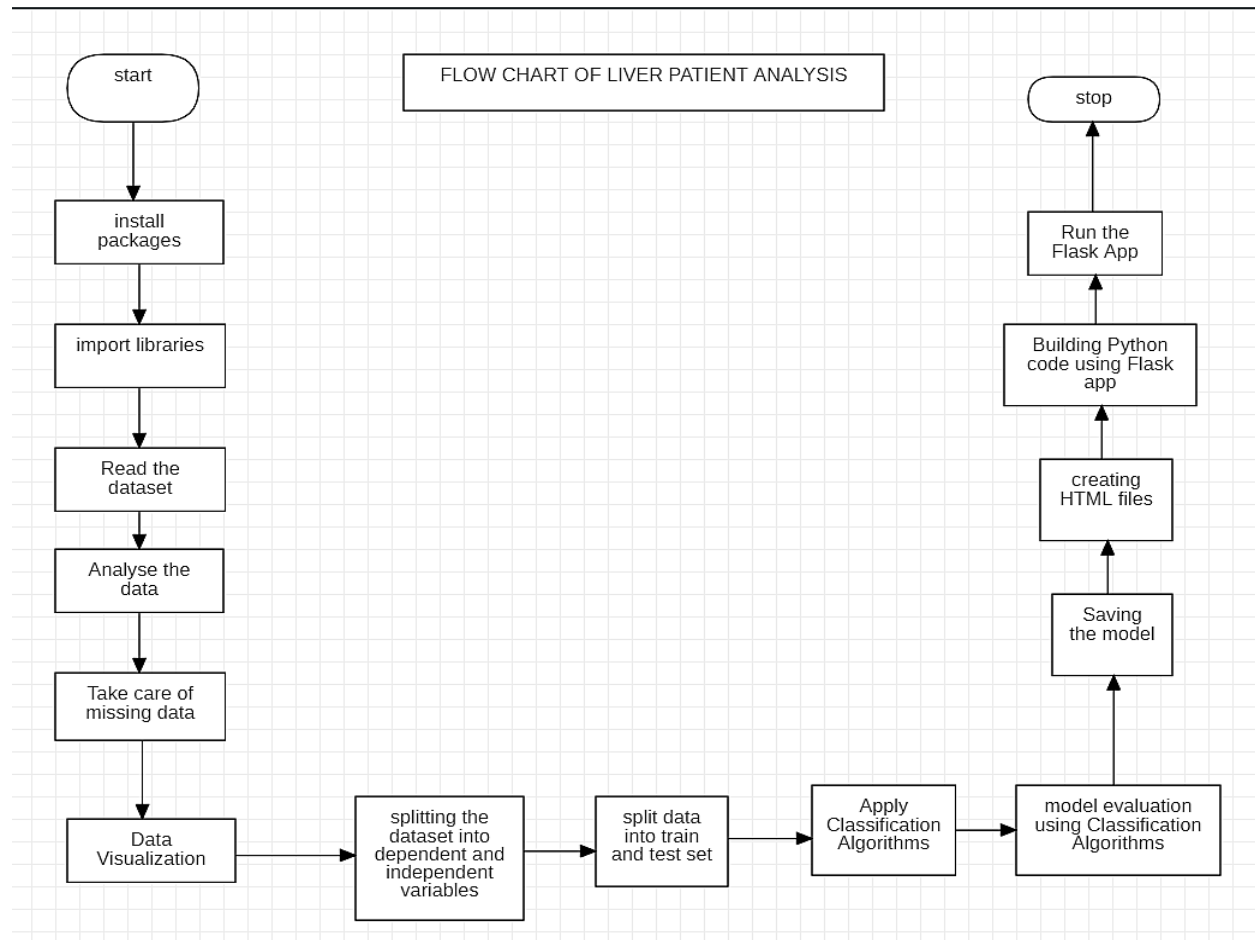
## (2) Random Forest Classification

Random Forest or Random decision forests are an ensemble method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees.

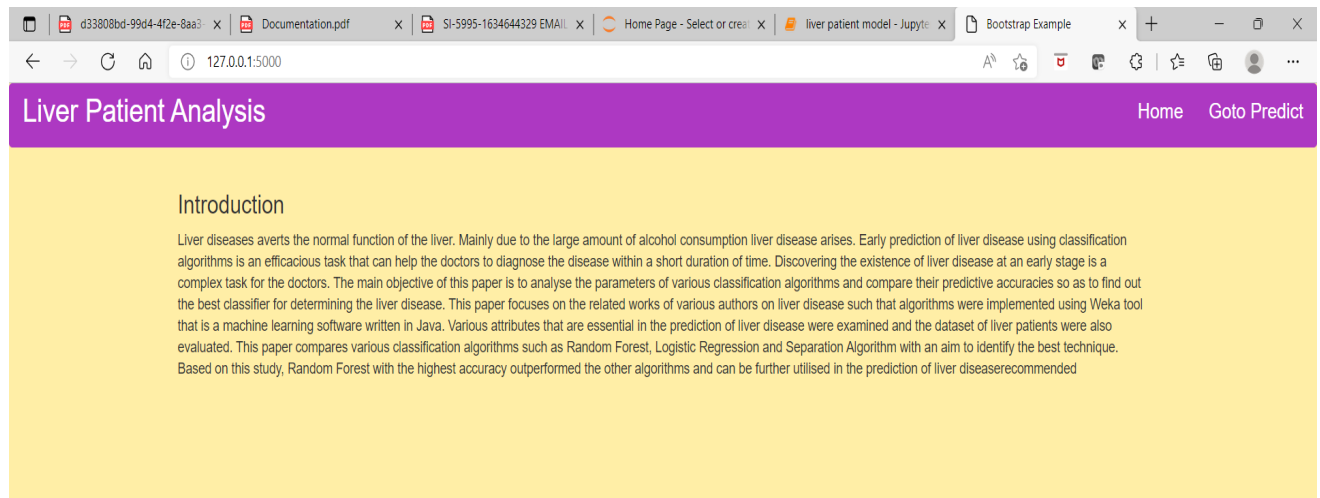
## (3) KNN Classification algorithm or K-Nearest Neighbour algorithm

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

## 5.FLOW CHART



## 6.RESULT



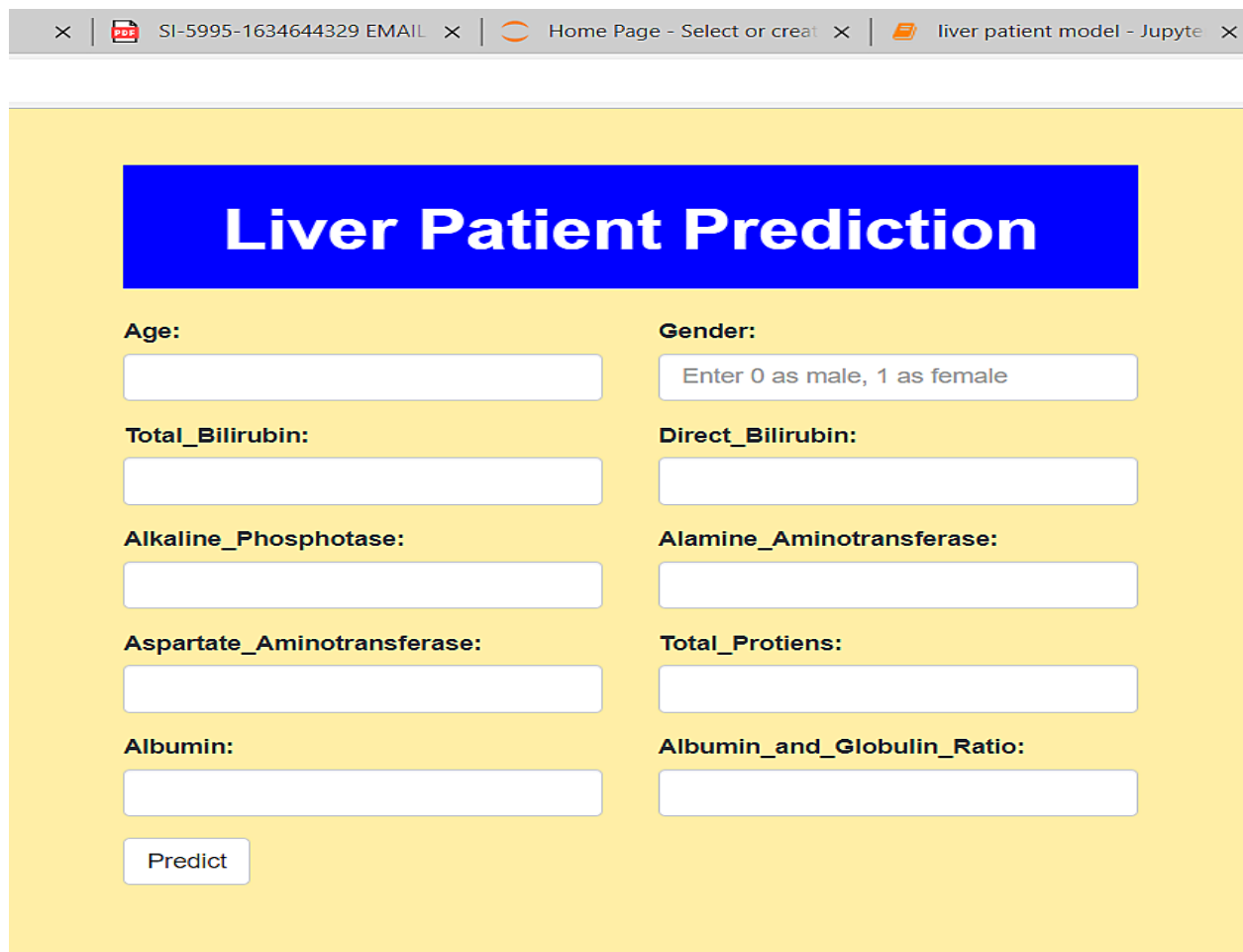
The screenshot shows a web browser with multiple tabs. The active tab is titled "Liver Patient Analysis". The page has a purple header bar with the title "Liver Patient Analysis" on the left and navigation links "Home" and "Goto Predict" on the right. The main content area has a yellow background and is titled "Introduction". The text describes the project's goal: to analyze various classification algorithms for liver disease prediction. It mentions that Random Forest performed best among the algorithms tested.

**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 diseases recommended.

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



The screenshot shows a web browser with multiple tabs. The active tab is titled "Liver Patient Prediction". The page has a yellow background and a blue header bar with the title "Liver Patient Prediction". Below the header, there are two columns of input fields for patient data. The left column contains fields for Age, Total\_Bilirubin, Alkaline\_Phosphotase, Aspartate\_Aminotransferase, and Albumin. The right column contains fields for Gender (with a hint "Enter 0 as male, 1 as female"), Direct\_Bilirubin, Alamine\_Aminotransferase, Total\_Protiens, and Albumin\_and\_Globulin\_Ratio. At the bottom left, there is a "Predict" button.

## Liver Patient Prediction

<b>Age:</b>	<b>Gender:</b>
<input type="text"/>	<input type="text" value="Enter 0 as male, 1 as female"/>
<b>Total_Bilirubin:</b>	<b>Direct_Bilirubin:</b>
<input type="text"/>	<input type="text"/>
<b>Alkaline_Phosphotase:</b>	<b>Alamine_Aminotransferase:</b>
<input type="text"/>	<input type="text"/>
<b>Aspartate_Aminotransferase:</b>	<b>Total_Protiens:</b>
<input type="text"/>	<input type="text"/>
<b>Albumin:</b>	<b>Albumin_and_Globulin_Ratio:</b>
<input type="text"/>	<input type="text"/>

# Liver Patient Prediction

**You dont have a liver desease problem**

# Liver Patient Prediction

**You have a liver desease 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

### ADVANTAGES:

- **Efficiency in workflow** : One of the first desires that probably comes to mind is efficiency. When building your website, you want to be able to reach as many people as you can. You also want to reach them on a consistent basis in a way that doesn't involve you spending all your time waiting at the hospital for the long time.
- **Reduce costs** : You don't need a large time to wait for the results for the entire day. They can approach hospital when they're ready to get about their condition and you don't need additional teams to get to know about results. Paying for a large team to constantly contact prospects isn't needed.
- Using machine learning algorithms to predict disease is made possible by increasing access to hidden attributes in medical data sets. Various kinds of data sets, such as blood panels with liver function tests, histologically stained slide images, and the presence of specific molecular markers in blood or tissue samples, have been used to train classifier algorithms to predict liver disease with good accuracy.
- 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.

### DISADVANTAGES:

- Any single error in data set can change the entire data.
- Correct accuracy must be needed while doing the project using supervised machine learning algorithms.

- Python code should be correct without any error.

## **8. APPLICATIONS:**

This application can further be developed with more idea and implementation and by using different algorithms. The accuracy score of the model can be further improved by using decision tree and also by increasing the data set, K-Nearest Neighbors algorithm is also one of the pertinent methods which can be used to predict the disease accurately. It proposes to improve the accuracy further.

## **9.CONCLUSIONS**

Prompt and timely detection of liver disease prediction plays a vital role in increasing life span of patient. In this paper, an attempt is made to detect the presence of liver disease using Support vector machine, Random Forest , K-NN classification methods of Machine Learning.

Liver disease is detected by clinicians who are well trained in identifying significant observations and classifying them as normal or abnormal using background information and other context clues. ML algorithms can be trained to detect the possibility of liver disease in a similar way to assist healthcare workers. Using the correlation of each variable with the risk of liver disease to train the model, ML methods were able to identify which blood donors were healthy and which had liver disease with high accuracy.

Among three ML classification methods, SVM and RF performed better than K-NN classification. Although, the accuracy levels for all three methods performed well based on the testing data set. SMOTE produced very effective results in classification performance by oversampling the minority group.

The machine learning algorithms presented in this study can support medical experts but are not the alternative when making decisions from ML classifiers for diagnostic pathways. These methods can reduce many of the limitations that occur in healthcare associated with inaccuracy in diagnoses, missing data, cost, and time. Application of the ML methods can help reduce the total burden of liver disease on public health worldwide by improving recognition of risk factors and diagnostic variables. More

importantly, for chronic liver disease, detecting liver disease at earlier stages or in hidden cases by ML could decrease liver-related mortality, transplants, and/or hospitalizations. Early detection improves prognosis, since treatment can be given before progression of the disease to later stages. Invasive tests, such as biopsy, would occur less in this case as well. Although this study focused on hepatitis and chronic liver disease variables for ML training, it can be hypothesized that the methods can be used to distinguish other types of liver disease from healthy individuals. Applying all of the mentioned methods to other areas of medicine could open the doors for AI/ML-facilitated diagnosis.

## **10.FUTURE SCOPE**

In the future, the local interpretable model-agnostic explanation (LIME) method will be used to understand the model's interpretability. Instead of binary classification, one may use multinomial classification by separating the types of liver disease. In this way, each model's performance can be compared. The described ML methods can assist health sectors to achieve a better diagnosis providing effective results in identifying groups or levels within medical data to facilitate healthcare workers. Moreover, ML methods are data driven, and they directly use diagnostic variables from patients' medical tests. Thus, it is a more reliable process. The applied ML methods in this project can save time, costs, and potentially lives for the betterment of disease diagnosis.

## **11.BIBLIOGRAPHY**

We referred some books and surfed the internet for the better outcome of the project

- W. Raghupathi, V. Raghupathi, Big data analytics in healthcare: promise and potential, Health information science and systems (2014), 2-3.
- A. Charleonnann, T. Fufaung, T. Niyomwong, W Chokchueypattanakit, S. Suwannawach, N. Ninchawee "Predictive Analytics for Chronic Kidney Disease Using Machine Learning Techniques" MITiCON2016.
- Youtube videos by simplilearn, Datacamp, Codebasics, eduraka!.
- Smartinternz tutorial classes help me for completion of project.

## APPENDIX

### A. SOURCE CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
data = pd.read_csv('C:/Users/ABHILASH REDDY/Desktop/smartbridge/indian_liver_patient.csv')
data.head()
data["Dataset"].value_counts()
data.tail()
data.info()
data.describe()
data.isnull().any()
data.isnull().sum()
data[data['Dataset']==1]
data['Dataset'].unique()
data.isnull().sum()
data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].mode()[0],
inplace=True)
data.isnull().sum()
plt.figure(figsize=(15,10))
plt.subplot(3,3,1)
plt.scatter(data['Age'], data['Dataset'])
plt.ylabel('Dataset')
plt.xlabel('Age')

plt.subplot(3,3,2)
plt.scatter(data['Gender'], data['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Gender')

plt.subplot(3,3,3)
plt.scatter(data['Total_Bilirubin'], data['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Total_Bilirubin')
```

```

plt.subplot(3,3,4)
plt.scatter(data['Direct_Bilirubin'], data['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Direct_Bilirubin')

plt.subplot(3,3,5)
plt.scatter(data['Alkaline_Phosphotase'], data['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Alkaline_Phosphotase')

plt.subplot(3,3,6)
plt.scatter(data['Alamine_Aminotransferase'], data['Dataset'],)
plt.ylabel('Dataset')
plt.xlabel('Alamine_Aminotransferase')
sns.countplot(data=data, x = 'Gender', label='Count')
m,f=data['Gender'].value_counts()
print("No of Males:",m)
print("No of Females:",f)
sns.countplot(data=data, x = 'Dataset')
LD,NLD=data['Dataset'].value_counts()
print("Liver Disease Patients:",LD)
print("Non-liver Desease Patients:",NLD)
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
# Converting Textual data into numeric data
data['Gender'] = le.fit_transform(data['Gender'])
data.head()
data['Gender'] = le.fit_transform(data['Gender'])
data.head()
x=data.iloc[:,0:-1]
y=data.iloc[:, -1]
from imblearn.over_sampling import SMOTE
smt = SMOTE()
x_resample, y_resample = smt.fit_resample(x,y)
x_resample
y_resample
y.shape, y_resample.shape
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x_resample,y_resample,test_size=0.2)

```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
svm=SVC()
RFmodel=RandomForestClassifier()
KNNmodel=KNeighborsClassifier()
from sklearn.svm import SVC
svm=SVC()
svm.fit(xtrain, ytrain)
SVMpred=svm.predict(xtest)
SVMaccuracy=accuracy_score(SVMpred, ytest)
SVMaccuracy
SVMcm=confusion_matrix(SVMpred, ytest)
SVMcm
from sklearn.ensemble import RandomForestClassifier
RFmodel=RandomForestClassifier()
RFmodel.fit(xtrain,ytrain)
RFpred=RFmodel.predict(xtest)
RFaccuracy=accuracy_score(RFpred, ytest)
RFaccuracy
RFcm=confusion_matrix(RFpred, ytest)
RFcm
from sklearn.neighbors import KNeighborsClassifier
KNN = KNeighborsClassifier()
KNN.fit(xtrain, ytrain)
KNNpred=KNN.predict(xtest)
KNNaccuracy=accuracy_score(KNNpred, ytest)
KNNaccuracy
KNNcm=confusion_matrix(KNNpred, ytest)
KNNcm
print("Support Vector Machine Algorithm accuracy score : {value:.2f}
%.format(value=SVMaccuracy*100))
print("Random Forest Algorithm accuracy score : {value:.2f} %.format(value=RFaccuracy*100))
print("K-Nearest Neighbors Algorithm accuracy score : {value:.2f}
%.format(value=KNNaccuracy*100))
import pickle
pickle.dump(RFmodel, open('liver_analysis1.pkl','wb'))
```

## FLASK APP CODE:

```
from flask import Flask, render_template, request # Flask is a application
# used to run/serve our application
# request is used to access the file which is uploaded by the user in our application
# render_template is used for rendering the html pages
import pickle # pickle is used for serializing and de-serializing Python object structures
model = pickle.load(open('liver_analysis1.pkl','rb'))

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=['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)]]

    # loading model which we saved
    model = pickle.load(open('liver_analysis1.pkl', 'rb'))
```

```
prediction= model.predict(data)[0]
if (prediction == 1):
    return render_template('chance.html', prediction='You have a liver disease problem, You
must and should consult a doctor. Take care')
else:
    return render_template('noChance.html', prediction='You dont have a liver disease
problem')

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

**SUBMITTED BY**  
YARRAM ABHILASH REDDY