LIVER PATIENT ANALYSIS USING MACHINE LEARNING

A UG PROJECT PHASE-1 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

BAJJURI ANUHYA	**18UK1A0505**
SINDHU MECHINENI	**18UK1A0552**
DENGU AJAY	**18UK1A0570**
JULURI VAMSHI	**18UK1A0577**

Under the esteemed guidance of

Mr. P.NIRANJAN REDDY

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
(Affiliated to JNTUH, Hyderabad)
Bollikunta, Warangal – 506005

2018– 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005
2018 – 2022

CERTIFICATE OF COMPLETION

UG PROJECT PHASE-1

This is to certify that the UG Project Phase-1 entitled **"LIVER PATIENT ANALYSIS USING MACHINE LEARNING"** is being submitted by ***B.ANUHYA(*H.NO:18UK1A0505) , *M.SINDHU* (H.NO:18UK1A0552),*D.AJAY*(H.NO:18UK1A0570),*J.VAMSHI*(H.NO:18UK1A0577)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2021-22**, is a record of work carried out by them under the guidance and supervision.

Project Guide
Mr. P. Niranjan Reddy
(Assistant Professor)

Head of the Department
Dr. R. Naveen Kumar
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **Mr. P. Niranjan Reddy** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

B.ANUHYA (18UK1A0505)
M.SINDHU (18UK1A0552)
D.AJAY (18UK1A0570)
J.VAMSHI (18UK1A0577)

ABSTRACT

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 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.

Keywords – Liver diseases , enzymes , machine learning.

TABLE OF CONTENTS:-

LIST OF FIGURES PAGE NO

1. INTRODUCTION

1.1. MOTIVATION:

Liver disease is any disturbance of liver function that causes illness. The liver is responsible for many critical functions within the body and should it become diseased or injured, the loss of those functions can cause significant damage to the body. Liver disease is also referred to as hepatic disease. Liver disease is a broad term that covers all the potential problems that cause the liver to fail to perform its designated functions. Usually, more than 75% or three quarters of liver tissue needs to be affected before a decrease in function occurs.

1.2. PROBLEM DEFINITION:

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.

1.3. PROJECT OBJECTIVE:

The main objective of this project is to 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 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. User can predict the disease by entering parameters in the web application.

1.4. LIMITATIONS OF PROJECT:

Many liver diseases are manageable if you catch them early. Without treatment, however, they can cause permanent damage. The complications of untreated or unmanaged liver disease can lead to cirrhosis, severe scarring that cannot be reversed. If cirrhosis has gone too far, a liver transplant may be your only option. Because some liver diseases can

develop without symptoms, making it a point to schedule annual physicals, along with the typical physical blood work, can help you and your doctor stay one step ahead.Focusing on a nutritious diet, physical exercise, and other healthy lifestyle choices such as limiting alcohol can also help with prevention or management.

1.5. ORGANIZATION OF DOCUMENTATION:

Actually there has been many theoretical projects and several experimental projects individually done based on liver patient analysis and many different algorithms have been developed for forecasting the liver diseases. But Machine Learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Our aim from the project is to make use of NumPy or pandas libraries in Machine Learning and predict whether a person has Liver disease or not. Secondly, to learn how to visualize the data by using graphs. In the end, we are predicting the liver disease based on the key list of liver enzymes, proteins, age and gender. The prediction is to be done using Machine Learning algorithms and withdrawing the conclusions.

2. PROBLEM STATEMENT

The liver is an organ about the size of a football. It sits just under your rib cage on the right side of your abdomen. The liver is essential for digesting food and ridding your body of toxic substances. Liver disease can be inherited (genetic). Liver problems can also be caused by a variety of factors that damage the liver, such as viruses, alcohol use and obesity. Over time, conditions that damage the liver can lead to scarring (cirrhosis), which can lead to liver failure, a life-threatening condition. But early treatment may give the liver time to heal.Liver failure occurs when your liver isn't working well enough to perform its functions (for example, manufacturing bile and ridding the body of harmful substances). Symptoms include **nausea, loss of appetite, and blood in the stool**. Treatments include avoiding alcohol and avoiding certain foods.

Other conditions that can lead to liver failure include:

- **Hepatitis A:** Contact with food or water contaminated with the hepatitis A virus, or with a person who's infected with virus, can cause liver inflammation. This type usually goes away on its own.
- **Autoimmune hepatitis:** In this type, your body's immune system, not a virus, attacks your liver and causes inflammation
- **Cirrhosis**: Things like drinking alcohol for many years or having hepatitis scar your liver can make it hard or impossible for your liver to work.
- **Primary sclerosing cholangitis**: This disease slowly damages your bile ducts. It mostly affects young men.
- **Oxalosis:** This is when your kidneys can't get rid of calcium oxalate crystals through your urine.
- **Wilson's disease:** People with this rare inherited disease store too much copper in their brain and liver.
- **Alpha-1 antitrypsin deficiency**: This genetic condition can lead to lung or liver disease.
- **Liver cancer:** People with long-term hepatitis B or hepatitis C often get this.
- **Liver adenoma:** This is when benign liver tumors are on an otherwise healthy liver. This often affects women between ages 20 and 44.

- **Fatty liver disease:** Extra fat cells can build up on your liver. Nonalcoholic fatty liver disease often affects people who are overweight, obese, or have high cholesterol. Alcohol-related fatty liver disease affects heavy drinkers.
- **Alcoholic hepatitis:** Liver inflammation that results from heavy or long-term drinking.
- **Alagille syndrome:** A genetic disorder that results in fewer bile ducts than normal in the liver.
- **Primary biliary cholangitis (PBC):** Over time, this disease destroys your small bile ducts. You might still hear it called by its former name, primary biliary cirrhosis.
- **Galactosemia:** People with this condition can't process galactose, a sugar found in many foods. It can cause liver damage.
- **Lysosomal acid lipase deficiency (LAL-D):** With this genetic condition, you can't produce an enzyme called lysosomal acid lipase (LAL), which helps your body break down fats and cholesterol in your cells. As a result, fats stay in your liver and cause damage.

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 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.

3. LITERATURE SURVEY

3.1. INTRODUCTION:

The liver is an organ about the size of a football. It sits just under your rib cage on the right side of your abdomen. The liver is essential for digesting food and ridding your body of toxic substances.Liver disease can be inherited (genetic). Liver problems can also be caused by a variety of factors that damage the liver, such as viruses, alcohol use and obesity.

Over time, conditions that damage the liver can lead to scarring (cirrhosis), which can lead to liver failure, a life-threatening condition. But early treatment may give the liver time to heal. Liver disease doesn't always cause noticeable signs and symptoms. If signs and symptoms of liver disease do occur, the may include:

- Skin and eyes that appear yellowish (jaundice)
- Abdominal pain and swelling
- Swelling in the legs and ankles
- Itchy skin
- Dark urine colour
- Pale stool colour
- Chronic fatigue
- Nausea or vomiting
- Loss of appetite
- Tendency to bruise easily

3.2. EXISTING SYSTEM:

The purpose of the present model is to predict the whether a person has a liver disease or not. The model 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.

3.3. DISADVANTAGES OF EXISTING SYSTEM:

- Collecting large amount of data set.
- Large number of training data and annotations are needed which may not be practical in some problems.

3.4. PROPOSED SYSTEM:

Here we are building a model by applying various machine learning algorithms find the best accurate model. Thus a person will get to know whether he/she is having a liver disease or not. Some of the machines learning algorithms are:

1. Linear Regression:

Linear Regression is **a supervised machine learning algorithm where the predicted output is continuous and has a constant slope**. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog).

Steps to implement Linear regression model:

1. Initialize the parameters.
2. Predict the value of a dependent variable by given an independent variable.
3. Calculate the error in prediction for all data points.
4. Calculate partial derivative w.r.t a0 and a1.
5. Calculate the cost for each number and add them.

2. Multiple Linear Regression:

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable. Multiple regression is a broader class of regressions that encompasses linear and nonlinear regressions with multiple explanatory variables.

3.Random Forest:

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

Working of Random Forest Algorithm

Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result...

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

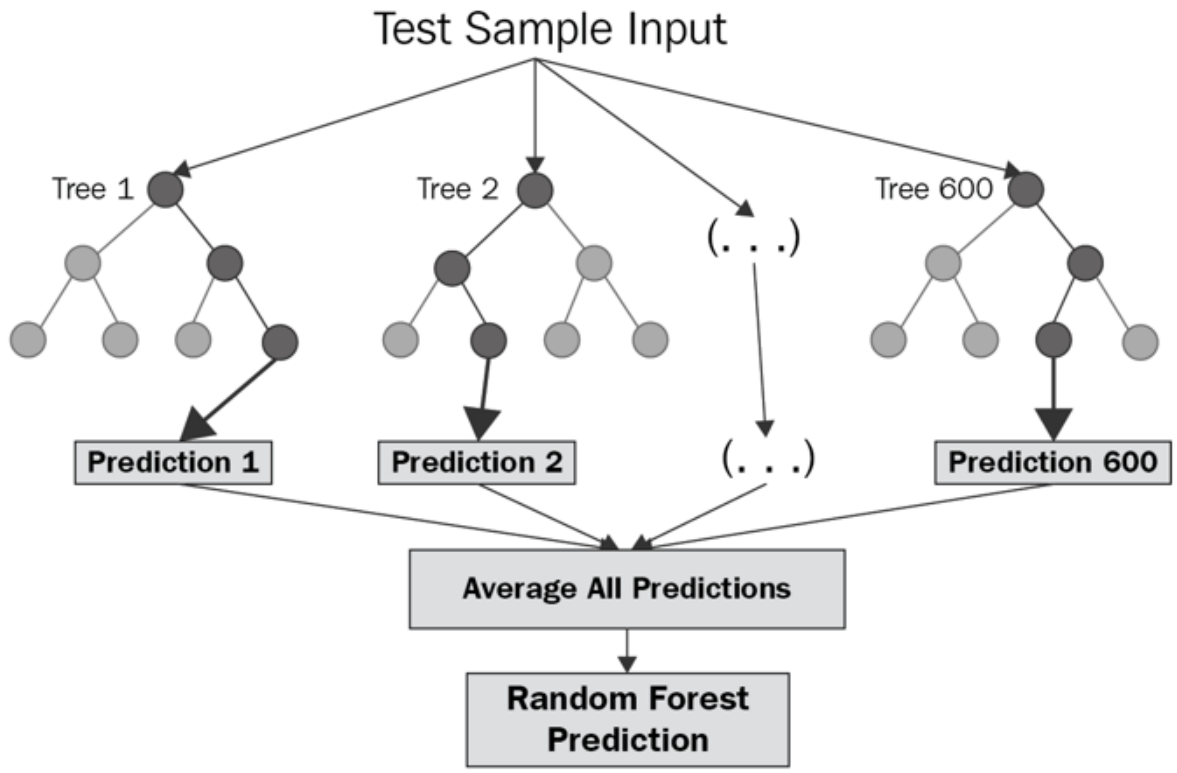


Figure 1 : Random Forest

4.Logistic regression:

- Logistic regression is **a supervised learning classification algorithm used to predict the probability of a target variable**. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.... Mathematically, a logistic regression model predicts P(Y=1) as a function of X.
- Logistic Regression is used when the dependent variable (target) is categorical. For example,
- To predict whether an email is a spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)

You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate.

5. k-nearest neighbor algorithm:

- It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbor's to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- K-Nearest Neighbors (KNN) is one of the simplest algorithms used in **Machine Learning for regression and classification problem**. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). The data is assigned to the class which has the nearest neighbors.
- It's also worth noting that the KNN algorithm is also part of a family of “lazy learning” models, meaning that it only stores a training dataset versus undergoing a training stage. This also means that all the computation occurs when a classification or prediction is being made. Since it heavily relies on memory to store all its training data, it is also referred to as an instance-based or memory-based learning method.

- The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of **K number of neighbors**

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

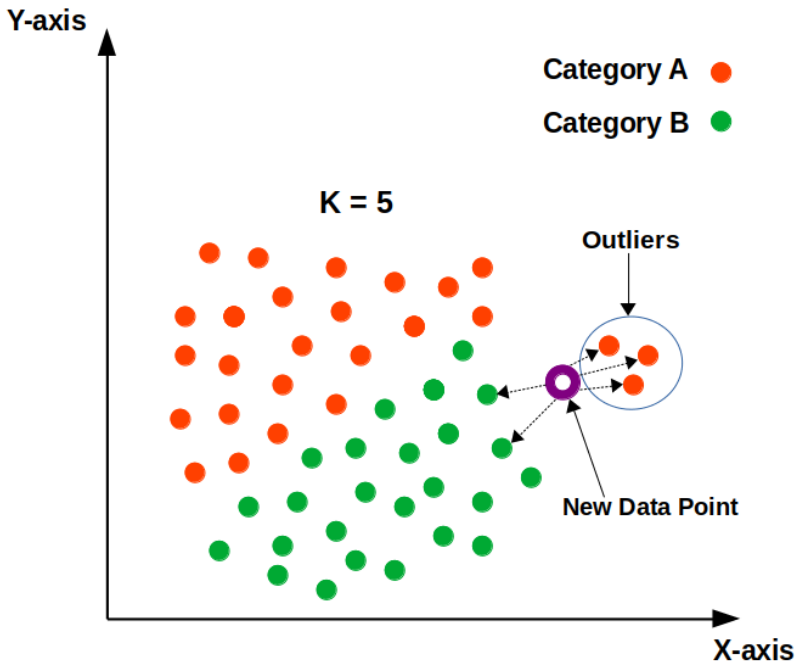


Figure 2 : k-nearest neighbor

4. EXPERIMENTAL ANALYSIS

Milestone 1: Data Collection

ML depends heavily on data, without data, a machine can't learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions.

You can collect datasets from different open sources like kaggle.com, data.gov; UCI machine learning repository etc. The dataset used for this project was obtained from Kaggle.

Milestone 2: Data Pre-processing

Data Pre-processing includes the following main tasks

- Importing the libraries.
- Importing the dataset.
- Analyse the data.
- Taking care of Missing Data.
- Data Visualisation.
- Splitting Data into Train and Test.

Milestone 3: Model Building

The model building process involves setting up ways of collecting data, understanding and paying attention to what is important in the data to answer the questions you are asking, finding a statistical, mathematical or a simulation model to gain understanding and make predictions. Model Building Includes:

- Import the model building libraries.
- Initialising the model.
- Training the model.
- Model Evaluation.
- Save the Model.

Milestone 4: Application Building

- Create an HTML File.
- Build python code.
- Run the app in local browser.
- Show casting the prediction on UI.

4.1. PROJECT ARCHITECTURE:

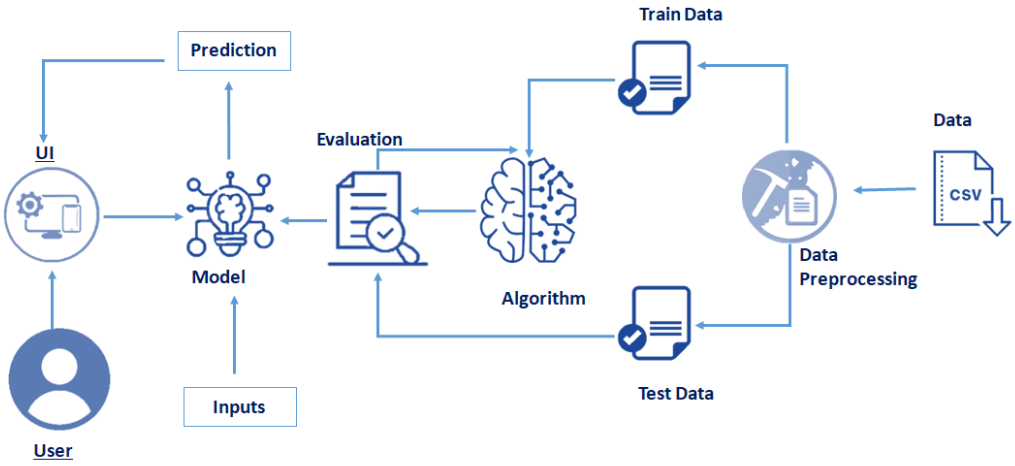


Figure 3 : Project Architecture

4.2. SOFTWARE AND HARDWARE REQUIREMENTS:

Software Requirements:

- Anaconda Environment
- Flask
- Python 3.9
- And other python libraries like NumPy, pandas.

Hardware Requirements:

- Operating system
- Processing
- RAM
- Operating system specifications
- Disk space

4.3. BLOCK DIAGRAM:

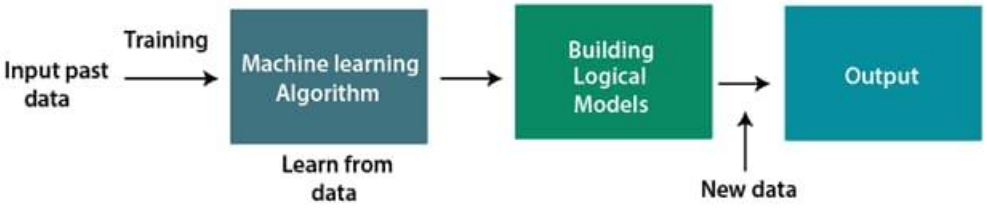


Figure 4 : Block Diagram

4.4. PROJECTFLOW:

- User interacts with the UI (User Interface) to upload the input features.
- Uploaded features/input is analysed by the model which is integrated.
- Once a model analyses the uploaded inputs, the prediction is showcased on the UI.

5. DESIGN

5.1. USE CASEDIAGRAM:

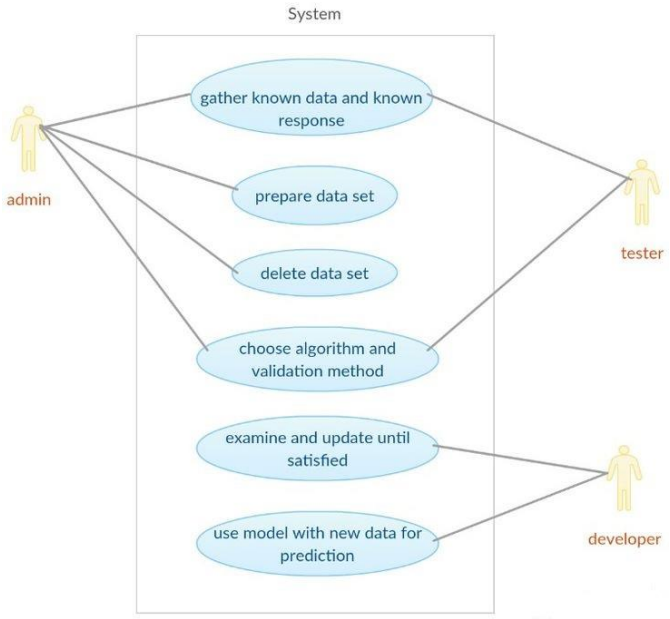


Figure 5: Use case Diagram

5.2. FLOWCHART:

Figure 6: Flowchart

5.3. DECISION TREE:

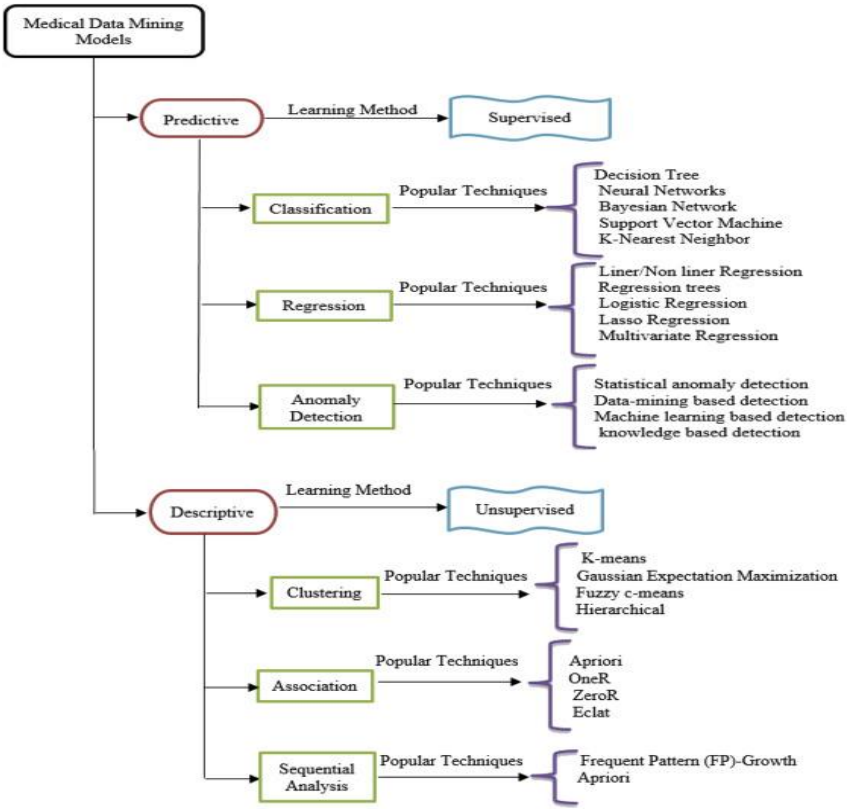


Figure 7: Decision Tree

5.3. SEQUENCE DIAGRAM:

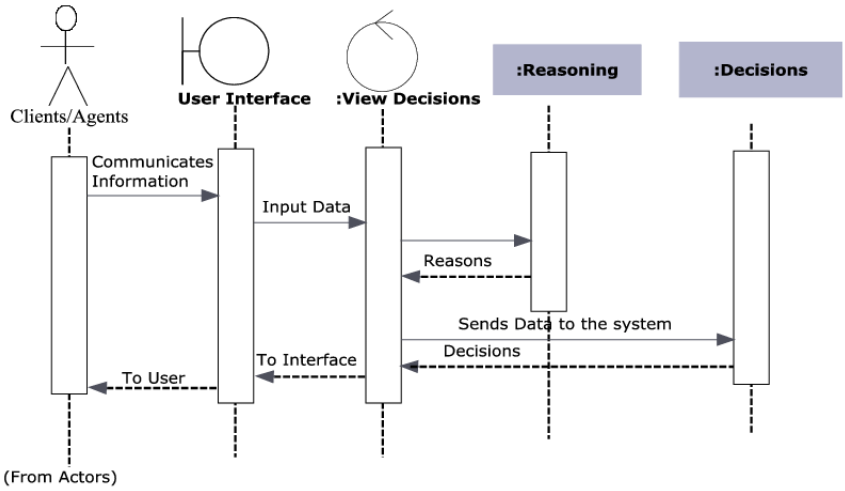


Figure 8: Sequence Diagram

5.4. COMMUNICATIO DIAGRAM:

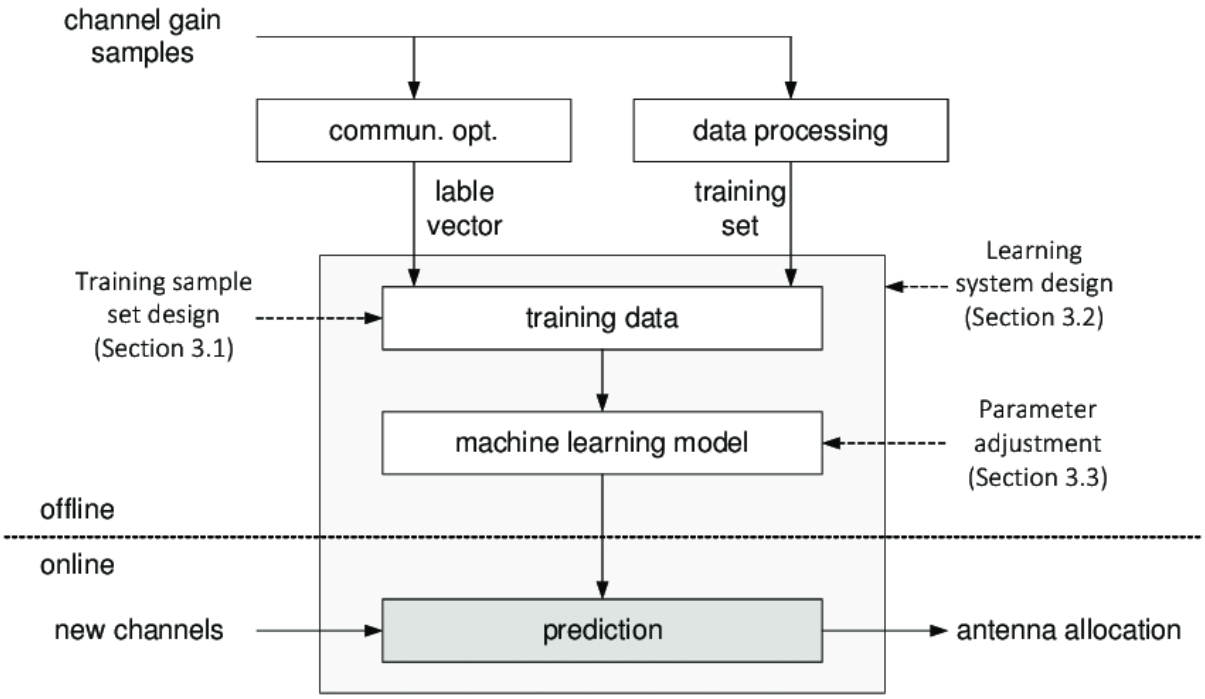


Figure 9: Communication Diagram

6. CONCLUSION

In UG Project Phase-1, we have worked on problem statement, literature survey and also done the experimental analyses which are required for the project to move forward. In experimental analysis we have discussed about the machine learning concepts and models and explained the algorithms to be used in the project. We also discussed about the flowcharts, use case diagrams, decision tree and sequence diagrams which are used in the project. Based on the experimental analysis we have designed the model for the project. Entire designing part is involved in UG Project Phase-1.

7. FUTURE SCOPE

UG Project Phase-2 is the extension of UG Project Phase-1. UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusions will be retrieved in the phase. We will also work on the applications, advantages, and disadvantages of the project in this phase. Future scope of the project will be also discussed in the UG Project Phase-2.

LIVER PATIENT ANALYSIS USING MACHINE LEARNING

A UG PROJECT PHASE-2 REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

BAJJURI ANUHYA	**18UK1A0505**
SINDHU MECHINENI	**18UK1A0552**
DENGU AJAY	**18UK1A0570**
JULURI VAMSHI	**18UK1A0577**

Under the esteemed guidance of

Mr. P.NIRANJAN REDDY

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
(Affiliated to JNTUH, Hyderabad)
Bollikunta, Warangal – 506005

2018– 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE
BOLLIKUNTA, WARANGAL – 506005
2018 – 2022

CERTIFICATE OF COMPLETION

UG PROJECT PHASE-2

This is to certify that the UG Project Phase-2 entitled **"LIVER PATIENT ANALYSIS USING MACHINE LEARNING"** is being submitted by ***B.ANUHYA(*H.NO:18UK1A0505) , *M.SINDHU*(H.NO:18UK1A0552),*D.AJAY*(H.NO:18UK1A0570),*J.VAMSHI*(H.NO:18UK1A0577)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** during the academic year **2021-22**, is a record of work carried out by them under the guidance and supervision.

Project Guide
Mr. P. Niranjan Reddy
(Assistant Professor)

Head of the Department
Dr. R. Naveen Kumar
(Professor)

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-2 in the institute.

We extend our heartfelt thanks to **Dr.R.NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-2.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-2 and for their support in completing the UG Project Phase-2.

We express heartfelt thanks to the guide, **Mr. P. Niranjan Reddy** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-2.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

B.ANUHYA (18UK1A0505)
M.SINDHU (18UK1A0552)
D.AJAY (18UK1A0570)
J.VAMSHI (18UK1A0577)

TABLE OF CONTENTS:

LIST OF FIGURES PAGE NO

1. INTRODUCTION

The liver is responsible for many critical functions within the body and should it become diseased or injured, the loss of those functions can cause significant damage to the body. 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.

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Here we are building a model by applying various machine learning algorithms find the best accurate model. Thus a person will get to know whether he/she is having a liver disease or not. On our Dataset, we have applied Random Forest Regression and KNN algorithm. Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. The abbreviation KNN stands for “K-Nearest Neighbour”. It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbours to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'.

UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusions are retrieved in this phase. We will also work on the applications, advantages, and disadvantages of the project in this phase. Future scope of the project will be also discussed in the UG Project Phase-2.

2. CODE SNIPPETS

2.1.MODELCODE:

Data Preprocessing

Data Pre-processing includes the following main tasks

1.Import the Libraries.

2.Importing the dataset.

3.Checking for Null Values.

4.Data Visualization.

5.Label Encoding.

6.OneHot Encoding.

7.Splitting Data into Train and Test.

In [1]:
```
import pandas as pd #Helps to read the csv files
import numpy as np #responsible for mathematical functions
import matplotlib.pyplot as plt #Data Visualisation
import seaborn as sns
import pickle #Function used for saving the model.
```

In [2]:
```
#import the dataset from specified location
data = pd.read_csv('D:\\indian_liver_patient.csv')
```

In [3]:
```
# showing the data from top 5
data.head()
```

Out[3]:

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

In [4]:
```
data.tail() #Bottom 5 rows
```

Out[4]:

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_
578	60	Male	0.5	0.1	500	20	34	5.9	1.6	
579	40	Male	0.6	0.1	98	35	31	6.0	3.2	
580	52	Male	0.8	0.2	245	48	49	6.4	3.2	
581	31	Male	1.3	0.5	184	29	32	6.8	3.4	
582	38	Male	1.0	0.3	216	21	24	7.3	4.4	

Figure 1: .ipynb code describing importing libraries and displaying the few rows from the Dataset.

```
In [5]: data.describe() #shows the descriptive statistics
```

Out[5]:

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

```
In [6]: data.info() #shows small information about data
```

```
<class 'pandas.core.frame.DataFrame'>
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
```

Figure 2: .ipynb code describing describe( ) and info( ) methods .

```
In [7]: data.isnull().any()
```

```
Age                         False
Gender                      False
Total_Bilirubin             False
Direct_Bilirubin            False
Alkaline_Phosphotase        False
Alamine_Aminotransferase    False
Aspartate_Aminotransferase  False
Total_Protiens              False
Albumin                     False
Albumin_and_Globulin_Ratio   True
Dataset                     False
dtype: bool
```

We can see that there are null values in the Albumin_and_Globulin_Ration Column.

```
In [8]: data.isnull().sum()
```

```
Age                         0
Gender                      0
Total_Bilirubin             0
Direct_Bilirubin            0
Alkaline_Phosphotase        0
Alamine_Aminotransferase    0
Aspartate_Aminotransferase  0
Total_Protiens              0
Albumin                     0
Albumin_and_Globulin_Ratio  4
Dataset                     0
dtype: int64
```

Figure 3: .ipynb code describing whether they are any NULL values in the Dataset.

```
In [10]: data['Dataset'].unique()
Out[10]: array([1, 2], dtype=int64)

In [11]: data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].mode()[0], inplace=True)

In [12]: #checking for the missing data after cleaning data
         data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].mode()[0], inplace=True)
         data.isnull().sum()
Out[12]: Age                         0
         Gender                      0
         Total_Bilirubin             0
         Direct_Bilirubin            0
         Alkaline_Phosphotase        0
         Alamine_Aminotransferase    0
         Aspartate_Aminotransferase  0
         Total_Protiens              0
         Albumin                     0
         Albumin_and_Globulin_Ratio  0
         Dataset                     0
         dtype: int64
```

We can see that all the null values are filled.

Figure 4 : .ipynb code describing filling of Null Values.

```
In [13]: 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')


         plt.subplot(3,3,7)
         plt.scatter(data['Aspartate_Aminotransferase'], data['Dataset'],)
         plt.ylabel('Dataset')
         plt.xlabel('Aspartate_Aminotransferase')

         plt.subplot(3,3,8)
         plt.scatter(data['Total_Protiens'], data['Dataset'],)
         plt.ylabel('Dataset')
         plt.xlabel('Total_Protiens')
```

```
plt.subplot(3,3,9)
plt.scatter(data['Albumin_and_Globulin_Ratio'], data['Dataset'])
plt.ylabel('Dataset')
plt.xlabel('Albumin_and_Globulin_Ratio')
```

```
Out[13]: Text(0.5, 0, 'Albumin_and_Globulin_Ratio')
```

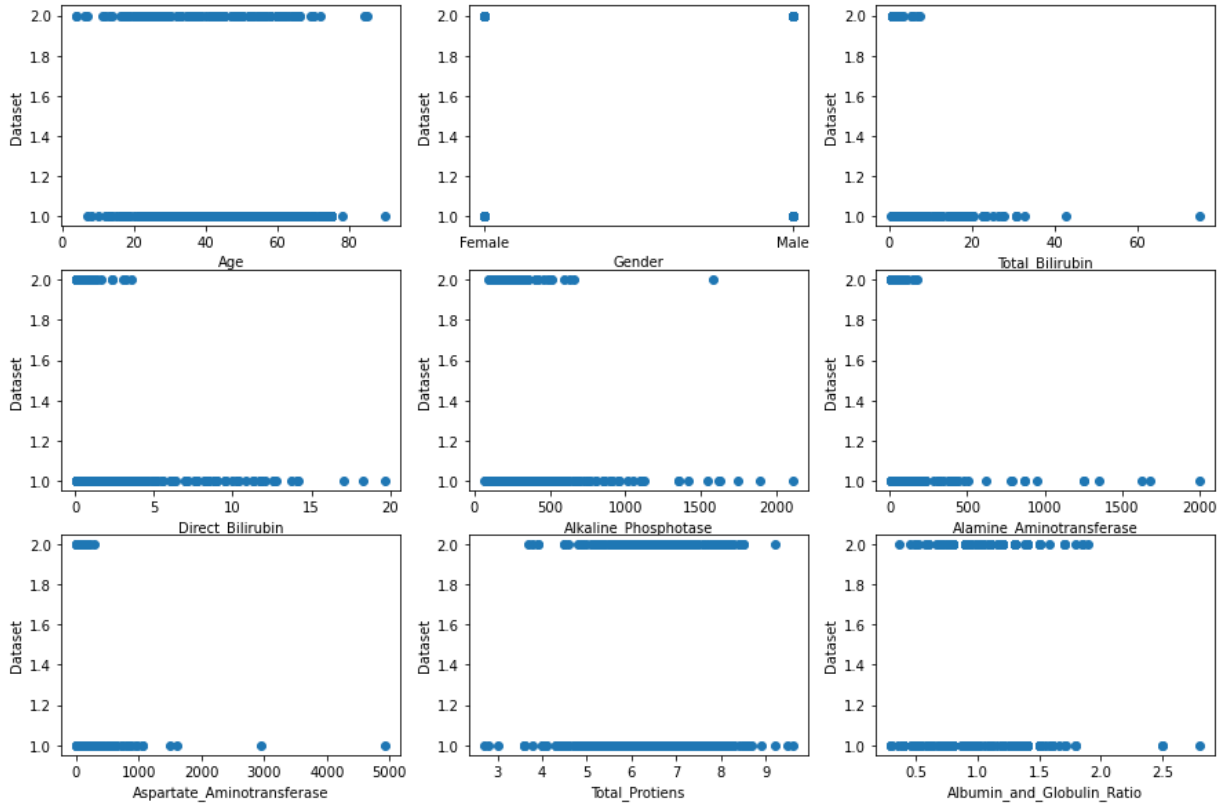
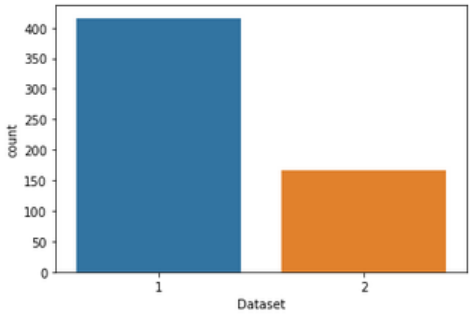


Figure 5 : .ipynb code describing Scatter Plot.

In [14]:
```
# Counting patients who are diagnosed and not diagnoised with liver disease
sns.countplot(data=data, x = 'Dataset')
LD,NLD=data['Dataset'].value_counts()
print("liver disease patinets:",LD)
print("Non-liver disease patinets:",NLD)
```

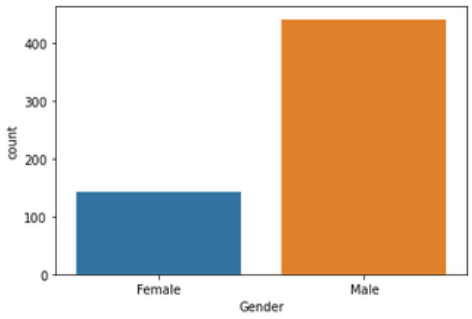
```
liver disease patinets: 416
Non-liver disease patinets: 167
```



Bar Plot which describe about the total number of diseases

In [15]:
```
# Counting patients who are Male and who are Female
sns.countplot(data=data, x = 'Gender', label='Count')
m,f=data['Gender'].value_counts()
print("No of Males:",m)
print("No of Females:",f)
```

```
No of Males: 441
No of Females: 142
```



Bar plot between Gender and Count

Figure 6 : .ipynb code describing Bar Plot.

In [16]:
```
# Importing the LabelEncoder libraray from scikit-learn
#encoding technique to convert categorical data to numeric data
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
# Converting Textual data into numeric data
data['Gender'] = le.fit_transform(data['Gender'])
data.head()
```

Out[16]:

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

In [17]:
```
# dividing the data into input and output
x=data.iloc[:,0:-1]
y=data.iloc[:,-1]
```

In the above code we are creating array or list of the independent variable x with our selected columns and for dependent variable y we are only taking the dependent or output or target column.

In [18]:
```
# importing the train_test_split from scikit-learn
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=10)
```

In [19]:
```
# Returns size of xtrain
xtrain.shape
```

Out[19]: (466, 10)

In [20]:
```
# Returns size of xtest
xtest.shape
```

Out[20]: (117, 10)

Figure 7: .ipynb code describing label encoding and splitting dataset into independent and dependent variables

Model Building

-Predictive modeling is a mathematical approach to create a statistical model to forecast future behavior based on input test data.

Model building includes the following main tasks

Training and testing the model, Evaluation of Model, Save the model, Predicting the output using the model.

In [21]:
```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

In [22]:
```
# Importing the machine learning model
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
```

In [23]:
```
# Initializing the machine learning models
RFmodel=RandomForestClassifier()
KNNmodel=KNeighborsClassifier()
```

In [24]:
```
#Random Forest Classifier Model
from sklearn.ensemble import RandomForestClassifier
RFmodel=RandomForestClassifier()
```

In [25]:
```
# train the data with Random Forest model
RFmodel.fit(xtrain, ytrain)
```

Out[25]: RandomForestClassifier()

Figure 8 : .ipynb code describing model building.

```
In [27]: # Checking for accuracy score from actual data and predicted data
         RFaccuracy=accuracy_score(RFpred, ytest)
         RFaccuracy

Out[27]: 0.7008547008547008

In [28]: # showing the confusion matrix
         RFcm=confusion_matrix(RFpred, ytest)
         RFcm

Out[28]: array([[72, 24],
                [11, 10]], dtype=int64)

In [29]: # K-Nearest Neighbors Model
         from sklearn.neighbors import KNeighborsClassifier
         KNN = KNeighborsClassifier()

In [30]: # train the data with K-Nearest Neighbors Model
         KNN.fit(xtrain, ytrain)

Out[30]: KNeighborsClassifier()

In [31]: KNNpred=KNN.predict(xtest)

In [32]: # Checking for accuracy score from actual data and predicted data
         KNNaccuracy=accuracy_score(KNNpred, ytest)
         KNNaccuracy

Out[32]: 0.6581196581196581

In [33]: # showing the confusion matrix
         KNNcm=confusion_matrix(KNNpred, ytest)
         KNNcm

Out[33]: array([[67, 24],
                [16, 10]], dtype=int64)

In [34]: 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))

         Random Forest Algorithm accuracy score : 70.09 %
         K-Nearest Neighbors Algorithm accuracy score : 65.81 %

In [35]: # saving the model
         import pickle
         pickle.dump(RFmodel, open('liver_analysis_RF.pkl','wb'))
```

Figure 9 : .ipynb code describing finding Accuracy .

2.2.HTML CODE AND PYTHON CODE

1. app.py code:

Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\hp\Desktop\Flask

C:\Users\hp\Desktop\Flask\app.py

app.py | chance.html | home.html | index.html | nochance.html

```
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 out application
# render_template is used for rendering the html pages
import pickle # pickle is used for serializing and de-serializing Python object structures


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_analysis_RF.pkl', 'rb'))

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

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

Figure 10 : .python code used for rendering all the HTML pages.

2. home.html:

Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\hp\Desktop\Flask

C:\Users\hp\Desktop\Flask\templates\home.html

app.py | home.html | index.html | chance.html | nochance.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

  <style type="text/css">
    body{
            background-color: #FFFEF9;
            background-image:url('../static/img/home.jpg');
            background-position: center;
            background-attachment: fixed;
            background-size:Fill;
            background-size: 60% 60%;
            background-repeat: no-repeat;
            background-position: center bottom;
        }
    nav{
        background-color: #ad38c2;
        height: 60px;
    }
    .navbar-brand{
        color: white;
        font-size: 30px
    }

    nav ul li a{
        color: white;
        font-size: 20px
    }
  </style>
</head>
<body>

<nav class="navbar">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand">Liver Patient Analysis</a>
    </div>
    <ul class="nav navbar-nav navbar-right">
        <li><a href="#">Home</a></li>
      <li><a href="/predict">Goto Predict</a></li>
    </ul>
  </div>
</nav>

<div class="container">
  <h3>Introduction</h3>
  <p><b>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 al
</div>
```

Figure 11 : home.html page is the code for home page of our Web Application.

3.index.html:

Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\hp\Desktop\Flask

C:\Users\hp\Desktop\Flask\templates\index.html

app.py | home.html | index.html | chance.html | nochance.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Liver Patient Analysis</title>
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

    <style type="text/css">
        body{
            background-color:#FFFEF9;
            background-image:url('../static/img/index.jpg');
            background-position: center;
            background-attachment: fixed;
            background-size:Fill;
            background-size: 110% 100%;
            background-repeat: no-repeat;
            background-position: center bottom;
        }
        .page-header{
            background-color: coral;
            width: 100%;
            height: auto;
            text-align: center;
            padding-top: 5px;
            color: #fff;
        }
        h1{
            font-size: 40px;
            font-weight: bold;
        }
    </style>
</head>
<body>

    <div class="container">
        <div class="row">
            <div class="col-md-3"></div>
            <div class="col-md-6">
                 <div class="page-header">
                    <h1>Liver Patient Prediction</h1>
                </div>
            </div>
        </div>
    </div>

    <div class="container">
        <div class="row">
            <div class="col-md-3"></div>
            <div class="col-md-6">
                <form action="/data_predict" method="POST">
                    <div class="row">
                        <div class="col-md-6">
                            <div class="form-group">
                                <label for="age">Age:</label>
                                <input type="text" class="form-control" id="age" name="age">
                            </div>
                        </div>
                        <div class="col-md-6">
                            <div class="form-group">
                                <label for="gender">Gender:</label>
                                <input type="text" class="form-control" id="gender" name="gender" placeholder="Enter 0 as male, 1 as female">
                            </div>
                        </div>
                    </div>

                    <div class="row">
                        <div class="col-md-6">
                            <div class="form-group">
                                <label for="tb">Total_Bilirubin:</label>
                                <input type="text" class="form-control" id="tb" name="tb">
                            </div>
                        </div>
                        <div class="col-md-6">
                            <div class="form-group">
                                <label for="db">Direct_Bilirubin:</label>
                                <input type="text" class="form-control" id="db" name="db">
                            </div>
                        </div>
                    </div>


                    <div class="row">
                        <div class="col-md-6">
                            <div class="form-group">
                                <label for="ap">Alkaline_Phosphotase:</label>
                                <input type="text" class="form-control" id="ap" name="ap">
                            </div>
                        </div>

                        <div class="col-md-6">
                            <div class="form-group">
                                <label for="aa1">Alamine_Aminotransferase:</label>
                                <input type="text" class="form-control" id="aa1" name="aa1">
                            </div>
                        </div>
                    </div>

                    <div class="row">
                        <div class="col-md-6">
                            <div class="form-group">
                                <label for="aa2">Aspartate_Aminotransferase:</label>
                                <input type="text" class="form-control" id="aa2" name="aa2">
                            </div>
                        </div>
                        <div class="col-md-6">
                            <div class="form-group">
                                <label for="tp">Total_Protiens:</label>
                                <input type="text" class="form-control" id="tp" name="tp">
                            </div>
                        </div>
                    </div>
```

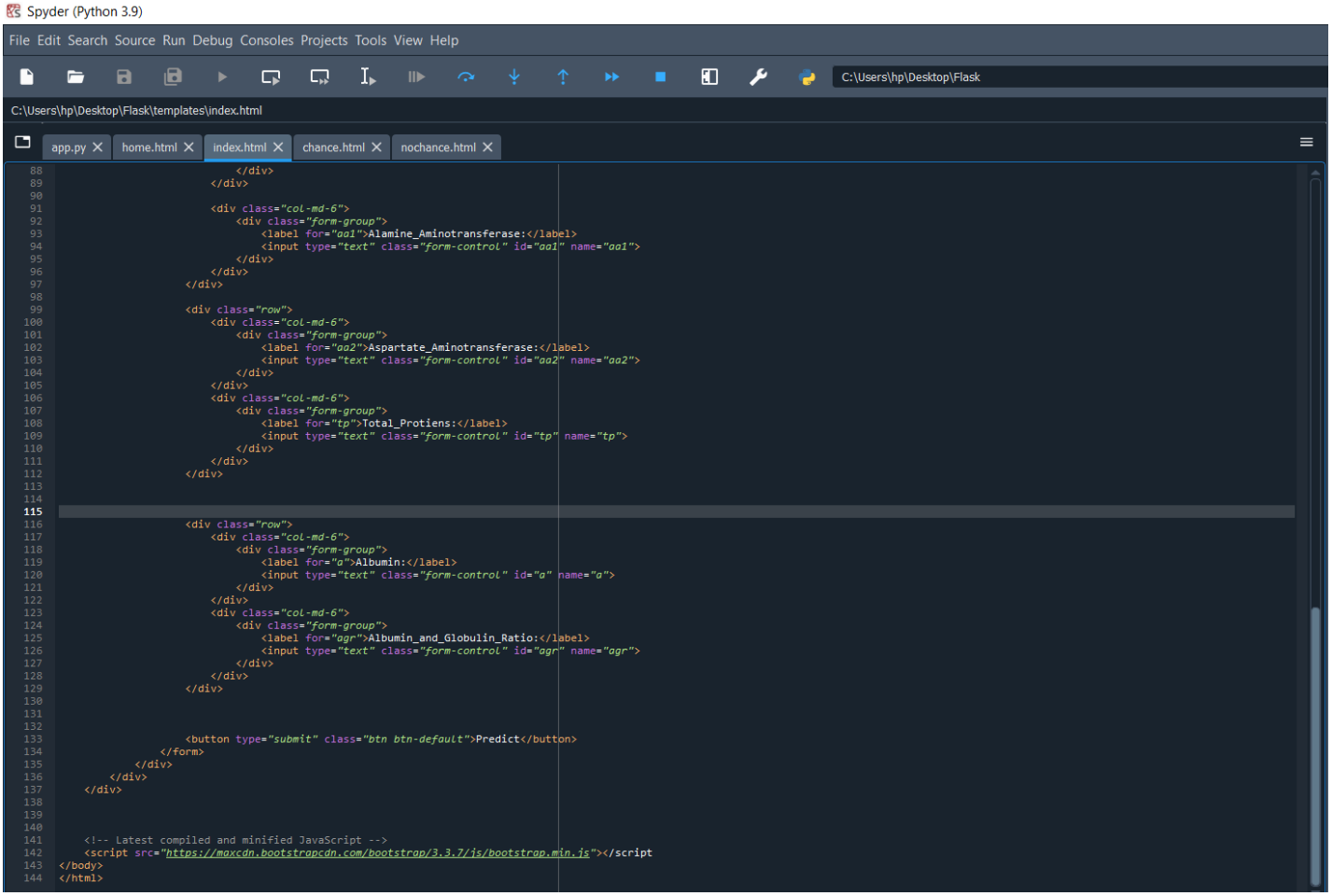


Figure 12: index.html is the page which displays all the inputs that are needed to be given by the user.

4.chance.html:

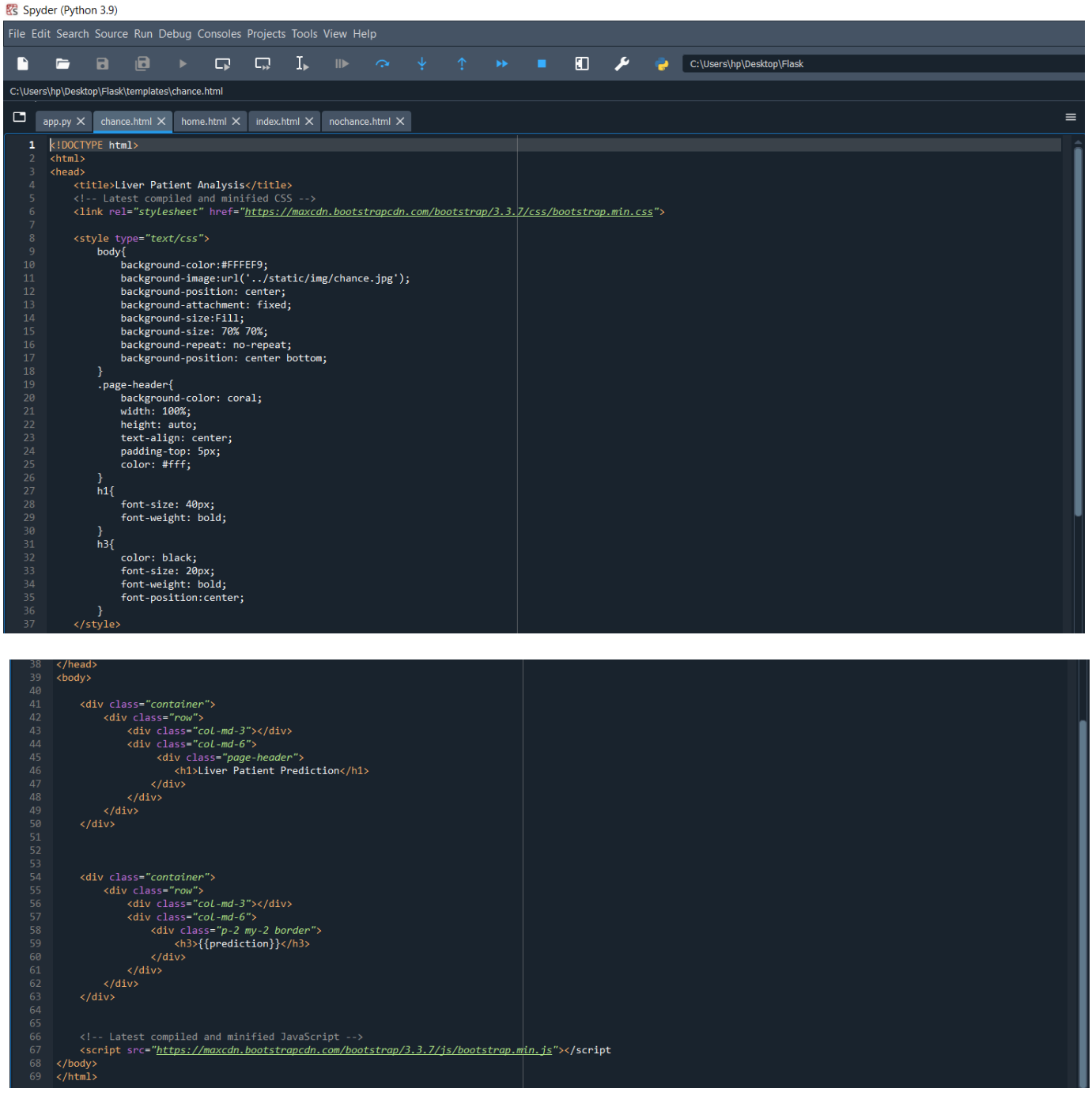


Figure 13: chance.html is the page which displays that the user is having liver disease.

5.nochance.html:

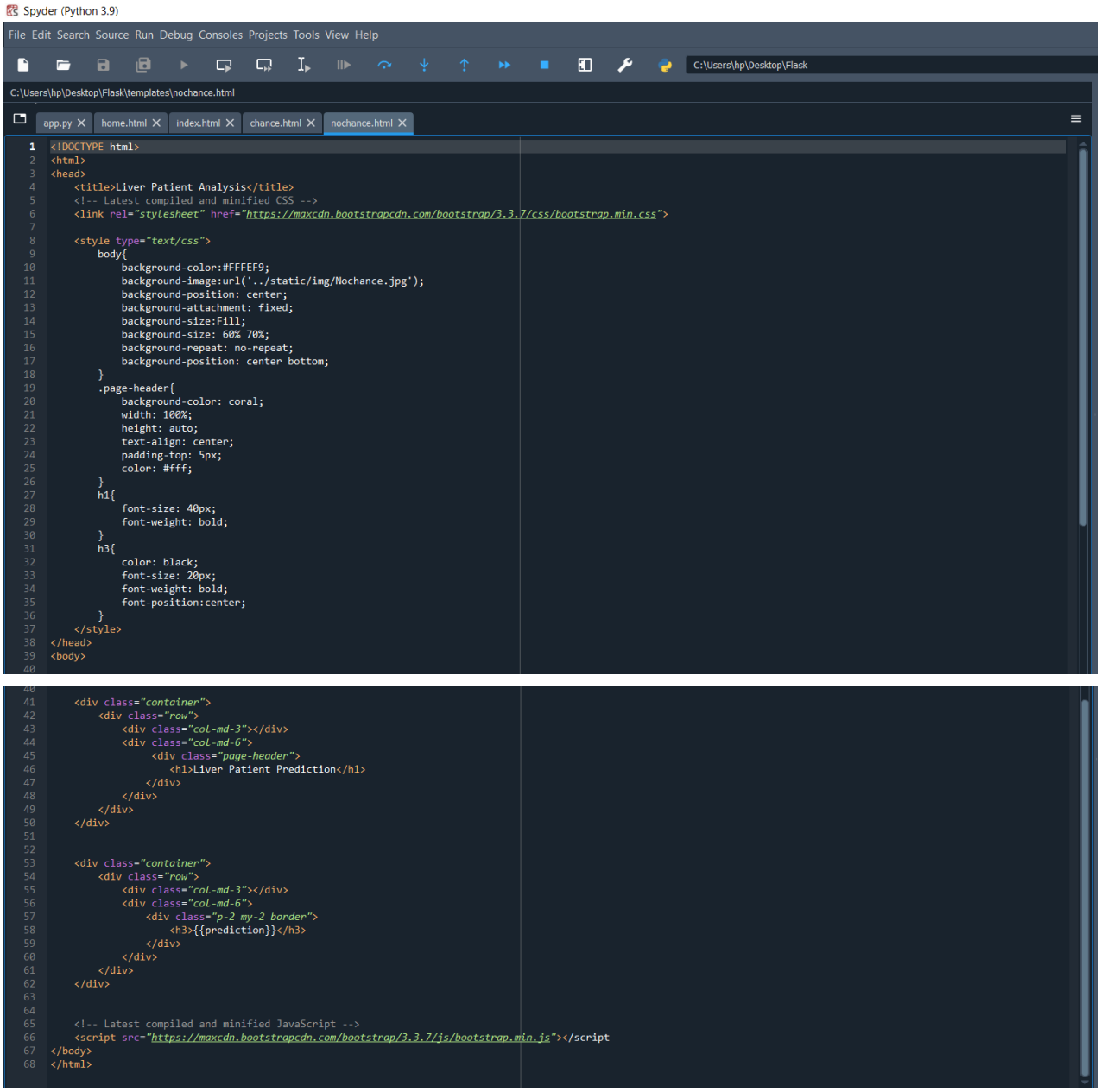


Figure 14: nochance.html is the page which displays that the user is not having liver disease.

3.CONCLUSION

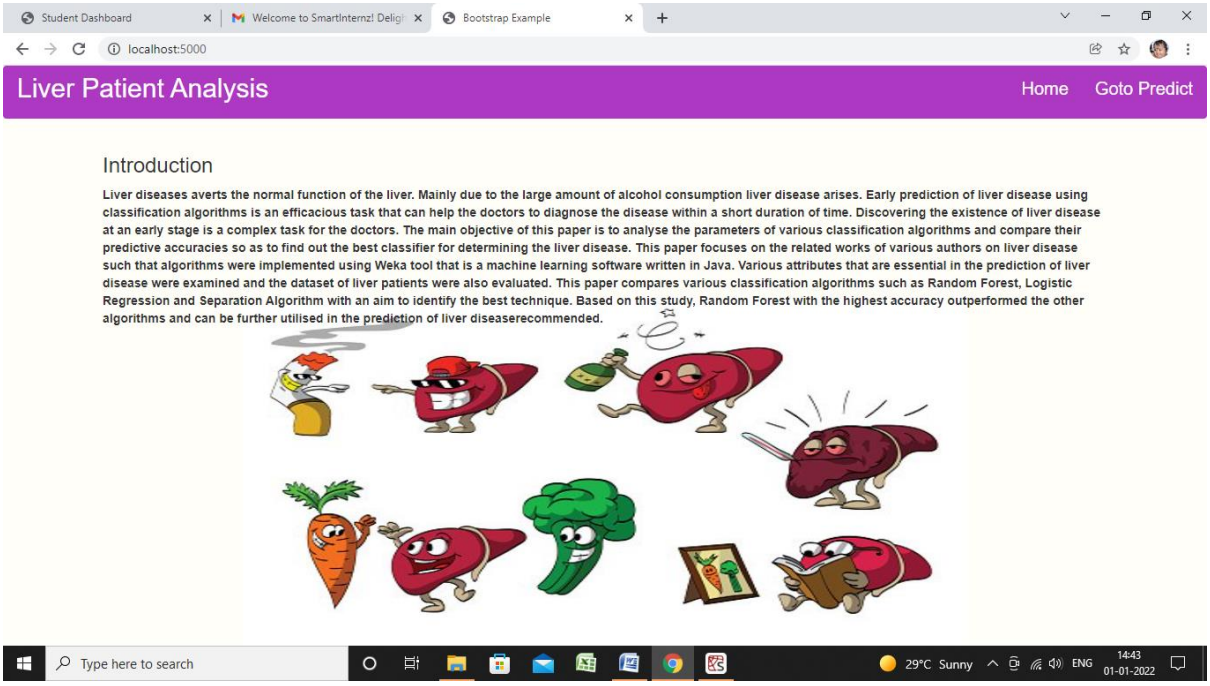


Figure:15: Home page (Which gives introduction to Liver Patient Analysis)

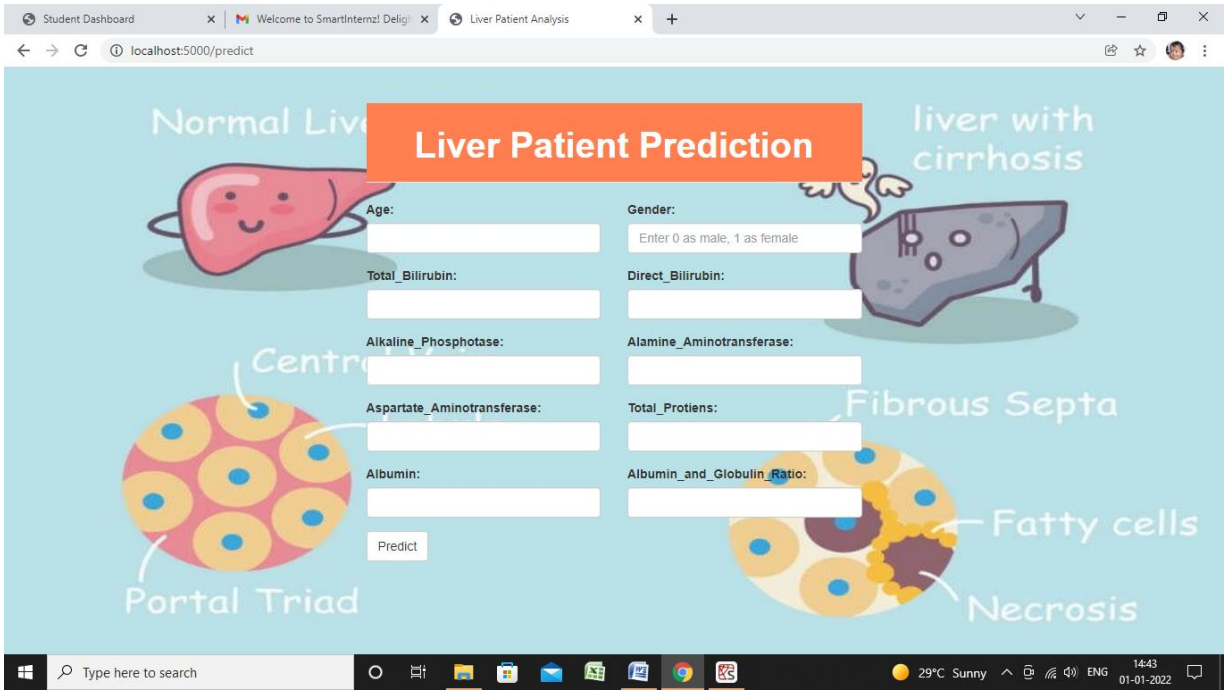


Figure: 16: Input pages (Which takes inputs from User)

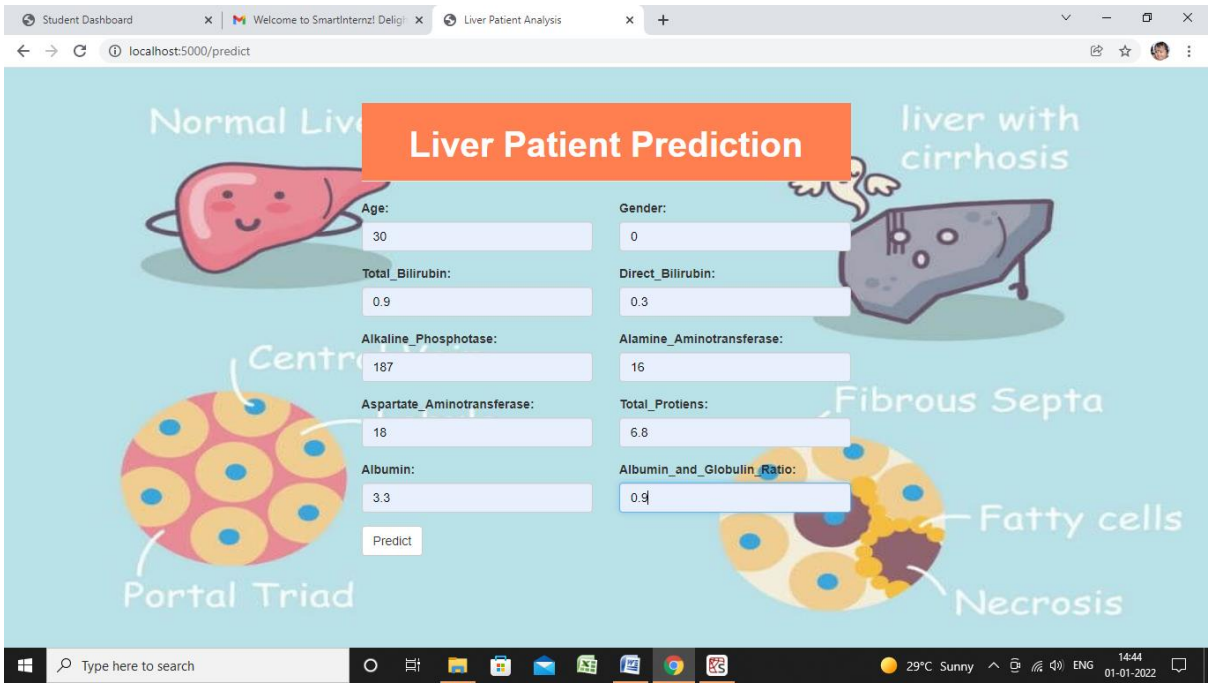


Figure: 17: Input pages (Which takes inputs from User)

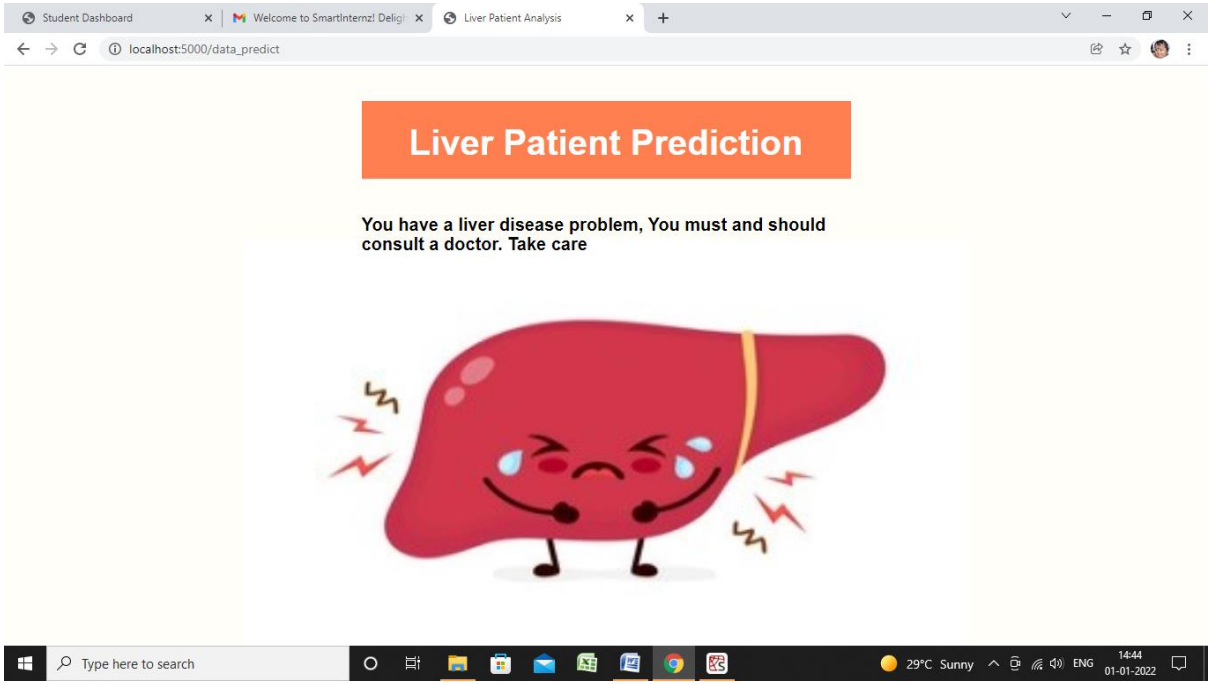


Figure: 18: Output page (Displays that the person is having liver disease)

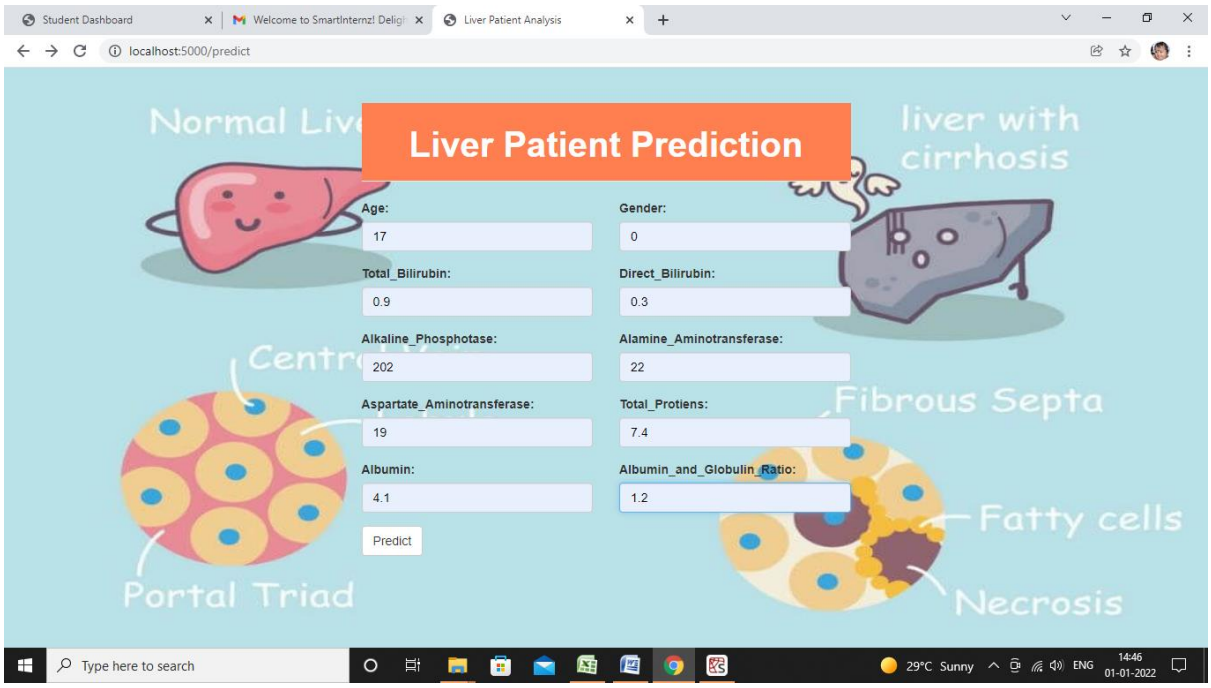


Figure: 19: Input pages (Which takes inputs from User)

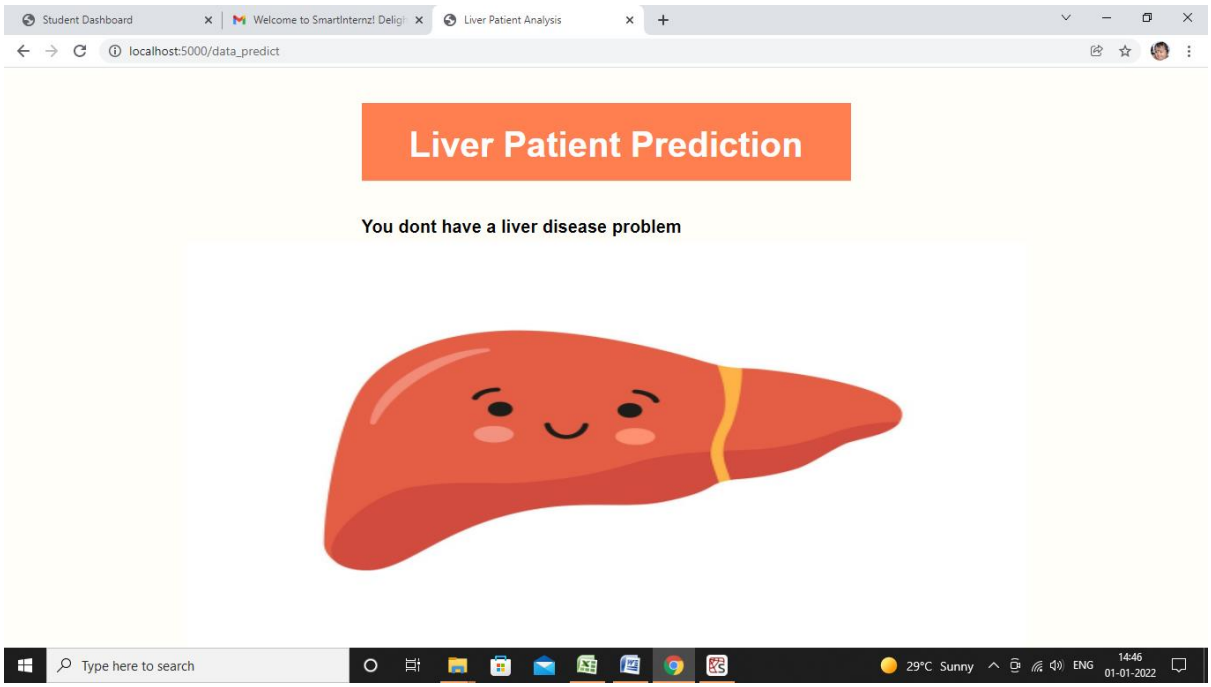


Figure: 20: Output page (Displays that the person is not having liver disease)

4. APPLICATION

The areas where this solution can be applied:

- Can be applied in each and every individual’s Daily Life.
- Can be used in medical fields for faster prediction in determining the disease.

5. ADVANTAGES

Those with liver disease may qualify for **Social Security disability**. If you suffer from autoimmune hepatitis, cirrhosis, and other chronic liver conditions, you may qualify for Social Security disability benefits if the condition meets the Social Security blue book's listing.

6. DISADVANTAGES

Liver disease can progress to cirrhosis and liver failure. Associated complications may include **increased risk of bleeding and infection, malnutrition and weight loss**, and decreased cognitive function. Some liver diseases are associated with an increased risk of developing liver cancer.

Liver failure, or liver failure that occurs over many years, may cause:

- Fatigue
- Nausea
- Loss of appetite
- Diarrhoea
- Vomiting blood
- Blood in the stool

As liver failure advances, symptoms become more severe. In later stages, symptoms of liver failure may include:

- Jaundice(yellowing of the skin and eyes)
- Extreme tiredness
- Disorientation (confusion and uncertainty)
- Fluid build-up in the abdomen and extremities (arms and legs)

Sometimes, the liver fails suddenly, which is known as acute liver failure. People with acute liver failure may have the following symptoms:

- Bleeding
- Changes in mental status
- Musty or sweet breath odour
- Movement problems
- Loss of appetite
- General feeling of being unwell

7. FUTURE SCOPE

On our Dataset, we have applied Random Forest Regression and KNN algorithm. Random forest has got the highest accuracy of 70%.

Enhancements that can be made in the future:

This model can be further developed to suggest what are the preventions that a person diagnosed with liver disease should be follow and Can also enhance by adding the strict diet schedules which should be followed to be strong and fit.And we can further classify type of disease also based on the inputs given by the user.

8. BIBILOGRAPHY

[1] Parminder Kaur1and Aditya Khamparia, ―Classification Of Liver Based Diseases Using Random Tree", International Journal of Advances in Engineering & Technology, June, 2015, ISSN: 22311963, Vol. 8, Issue 3, pp. 306-313

[2] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[3] Bendi Venkata Ramana, Prof. M.Surendra Prasad Babu,, Prof. N. B. Venkateswarlu,‖A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis‖, International Journal of Database Management Systems (IJDMS), Vol.3, No.2, May 2011.

[4] Chuan Choong Yang, Chit Siang Soh and Vooi Voon Yap, ―A nonintrusive appliance load monitoring for efficient energyconsumption based on Naive Bayes classifier‖, Sustainable Computing: Informatics and Systems 14 (2017) 34–42.

[5] Cleary, J. and L. Trigg, ―K*: An Instance-based Learner Using an Entropic Distance Measure‖, in 12th International Conference on Machine Learning. 1995. p. 108-114.

[6] Ross J. Quinlan: ―Learning with Continuous Classes‖ In Proceedings AI'92 (Adams & Sterling, Eds), 343-348, Singapore: World Scientic, 1992.

[7] Youvrajsinh Chauhan and Jignesh Vania, ―J48 Classifier Approach to Detect Characteristic of Bt Cotton base on Soil Micro Nutrient‖, International Journal of Computer Trends and Technology (IJCTT) – volume 5 number 6 –Nov 2013.

[8] S. Muthuselvan and Dr. K. Soma Sundaram, ―An Analysis of Knowledge Discovery Process Over a Cloud Environment — A Survey‖, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015).

[9] Inderjit Kaur, Deep Mann,‖Data Mining in Cloud Computing‖, International Journal of Advanced Research in Computer Science and Software Engineering.

[10] Emmanuel Ahishakiye, Elisha Opiyo Omulo, Danison Taremwa and Ivan Niyonzima, ―Crime Prediction Using Decision Tree (J48) Classification Algorithm‖ International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 06 – Issue 03, May 2017.

9.HELP FILE

PROJECT EXECUTION:

STEP-1: Go to **Start,** search and launch **ANACONDA NAVIGATOR.**

STEP-2: After launching of **ANACONDA NAVIGATOR,** launch **JUPYTER NOTEBOOK.**

STEP-3: Open **"Major project code" IPYNB file.**

STEP-4: Then run all the cells.

STEP-5: All the **data preprocessing**, **training and testing**, **model building**, **accuracy** of the model can be showcased.

STEP-6: And a pickle file will be generated.

STEP-7: Create a Folder named **FLASK** on the **DESKTOP.** Extract the pickle file into this Flask Folder.

STEP-8: Extract all the html files (home.html, index.html, chance.html, nochance.html) and python file(app.py) into the **FLASK Folder.**

STEP-9: Then go back to **ANACONDA NAVIGATOR** and the launch the **SPYDER.**

STEP-10: After launching Spyder, give the path of **FLASK FOLDER** which you have created on the DESKTOP.

STEP-11: Open all the app.py and html files present in the Flask Folder.

STEP-12: After running of the app.py, open **ANACONDA PROMPT** and follow the below steps:

cd File Path→click enter

python app.py→click enter (We could see running of files).

STEP-13: Then open **BROWSER,** at the URL area type "**localhost:5000".**

STEP-14: Home page of the project will be displayed.

STEP-15: Click on "**Go to Predict".** Directly it will be navigated to index page.

STEP-16:A index page will be displayed where the user needs to give the inputs and then click on "**Predict"**. Output will be generated whether a person is having liver disease or not.