Government of Tamil Nadu

Naan Muthalvan - Project-Based Experiential Learning

A Review of Liver Patient Analysis Methods Using Machine Learning

Submitted by

Team ID: NM2023TMID22698

R.BACKIA LAKSHMI -(20326ER043)

S.DEVIPRIYA-(20326ER045)

S.DIVYA(20326ER046)

M.DURGASRI(20326ER047)

Under the guidance of

Mrs. P.SANGEETHA M.SC., M.Phil.,

Guest Lecturer

PG and Research Department of Computer Science

M.V.MUTHIAH GOVERNMENT ART SCOLLEGE FOR WOMEN

(Affiliated To Mother Teresa Women's University, Kodaikanal)

Reaccredited with "A" Grade by NAAC

DINDIGUL-624001.

APRIL-2023

M.V.MUTHIAH GOVERNMENT ARTS COLLEG FORWOMEN
(Affiliated to Mother Teresa Women's University, Kodaikanal)
Reaccredited with "A" Grade by NAAC
Dindigul-624001

PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of the project entitled**, "A REVIEW OF LIVER PATIENT ANALYSIS METHODS USING MACHINE LEARNING"** .done by **Ms.R. BACKIA LAKSHMI (20326ER043) ,Ms.S.DEVIPRIYA (20326ER045)**, **Ms.S.DIVYA (20326ER046)and Ms.M.DURGASRI(20326ER047).**This is submitted in partial fulfillment for the award of the degree of **Bachelor of Science in Computer Science in M.V.MUTHIAHGOVERNMENT ARTS COLLEGE FOR WOMEN,DINDIGUL** during the period of December 2022 to April 2023.

Project Mentor(s) **Head of the Department**

Submitted for viva-voce Examination held on _____

CONTENT

ABSTRACT

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 utilized in the prediction of liver disease and can be recommended to the user.

INTRODUCTION

1.INTRODUCTION

1.1 OVER VIEW:

Liver disease is a significant health concern worldwide, and early prediction of liver disease can greatly aid in effective management and treatment. Machine learning algorithms, which are a type of artificial intelligence, have shown great potential in predicting liver disease by analyzing large amounts of data. Liver disease prediction using machine learning algorithms typically involves several steps. First, a dataset containing relevant features or variables, such as patient age, gender, liver function tests, imaging findings, and medical history, is collected. This dataset is then used to train and evaluate different machine learning algorithms.There are several commonly used machine learning algorithms for liver disease prediction, including logistic regression, decision trees, support vector machines, random forests, and neural networks. These algorithms use different approaches to analyze the dataset and identify patterns or relationships between the features and the presence of liver disease.Once the machine learning algorithms are trained on the dataset, they can be tested using a separate dataset to evaluate their performance. Common performance metrics used in liver disease prediction include accuracy, sensitivity, specificity, and area under the receiver operating characteristic (ROC) curve.

1.2 PURPOSE

Liver disease prediction using machine learning algorithms has several potential benefits. It can help clinicians identify individuals at high risk for liver disease, allowing for early intervention and preventive measures. It can also assist in determining the optimal treatment plan for patients with liver disease and monitor disease progression over time. However, there are also limitations to using machine learning algorithms for liver disease prediction. The quality and representativeness of the dataset used for training and testing the algorithms can greatly impact their performance. Additionally, machine learning algorithms are not a substitute for clinical judgment and should be used in conjunction with other diagnostic tools and medical expertise.

SYSTEM SPECIFICATION

2.1 HARDWARE SPECIFICATION

Processor	:	Dual Core
Hard Disk	:	500 GB
RAM	:	4 GB
Board	:	Mercury
Key Board	:	106 keys (TVS Key board)
Monitor	:	15” Digital Color

2.2 SOFTWARE SPECIFICATION

Operating System	:	Windows 10
Language	:	Python and Machine Learning

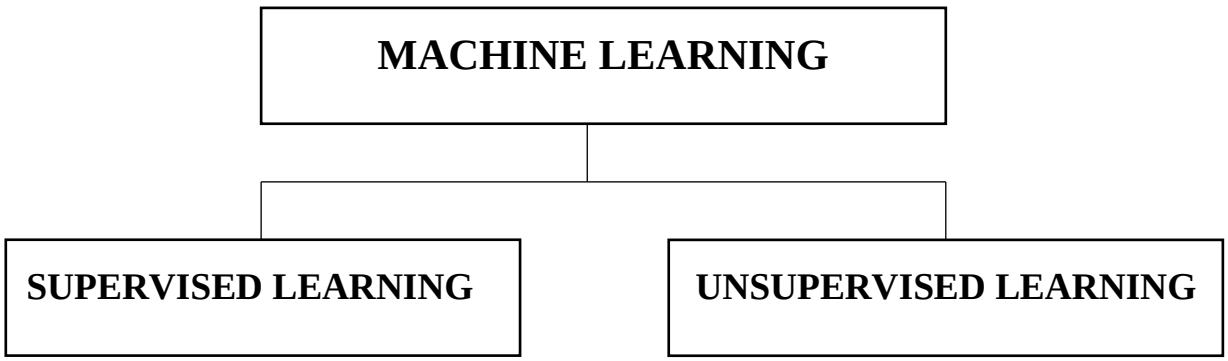
2.3 SOFTWARE DESCRIPTION

MACHINE LEARNING:-

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them.

Machine learning algorithms are typically created using frameworks that accelerate solution development, such as TensorFlow and PyTorch.



SUPERVISED MACHINE LEARNING:-

In supervised learning, a model is trained on labeled data, where the input data is paired with corresponding output labels. The goal is to learn a mapping between inputs and outputs, so that the model can make predictions on unseen data. Examples of supervised learning algorithms include linear regression, logistic regression, decision trees, support vector machines (SVM), and neural networks.

UNSUPERVISED MACHINE LEARNING:-

In unsupervised learning, a model is trained on unlabeled data, where the input data does not have any associated output labels. The goal is to find patterns, relationships, or structures within the data. Examples of unsupervised learning algorithms include clustering algorithms such as k-means clustering, hierarchical clustering, and DBSCAN, as well as dimensionality reduction techniques like principal component analysis (PCA) and t-SNE.

LOGISTIC REGRESSION:

Logistic Regression is a simple and interpretable algorithm that works well for binary classification tasks. It is computationally efficient and can provide insights into the importance of different features in predicting liver diseases.

RANDOM FOREST:

Random Forest is an ensemble learning method that combines multiple decision trees to create a robust and accurate model. It is known for handling noisy data and avoiding overfitting, making it a good choice for liver patient analysis.

K -NEAREST NEIGHBORS (KNN):

KNN is a non-parametric algorithm that can be used for both classification and regression tasks. It is simple to implement and can capture non-linear relationships in the data. However, it may suffer from the curse of dimensionality and be sensitive to the choice of K value.

ARTIFICIAL NEURAL NETWORKS (ANN):

ANN, particularly deep learning models, such as multilayer perceptron (MLP) and convolutional neural networks (CNN), have shown great success in various domains, including medical diagnosis. They can learn complex patterns from large datasets and may outperform other methods when the dataset is large and complex.

DECISION TREE MODEL

A function named Decision Tree Classifier is imported and train and test data are passed as the parameters. Inside the function, Decision Tree Classifier algorithm is initialized and training data is passed to the model with .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

PROBLEM DEFINITION & DESIGN THINKING

3.1 EMPATHY MAP

An empathy map is a visual tool used in design thinking and user-centered design processes to help understand and empathize with a particular user or customer. It is typically a simple diagram or template that is used to capture and organize information about a user's experiences, behaviors, needs, and emotions.

An empathy map is divided into four quadrants, each representing a different aspect of the user's experience:

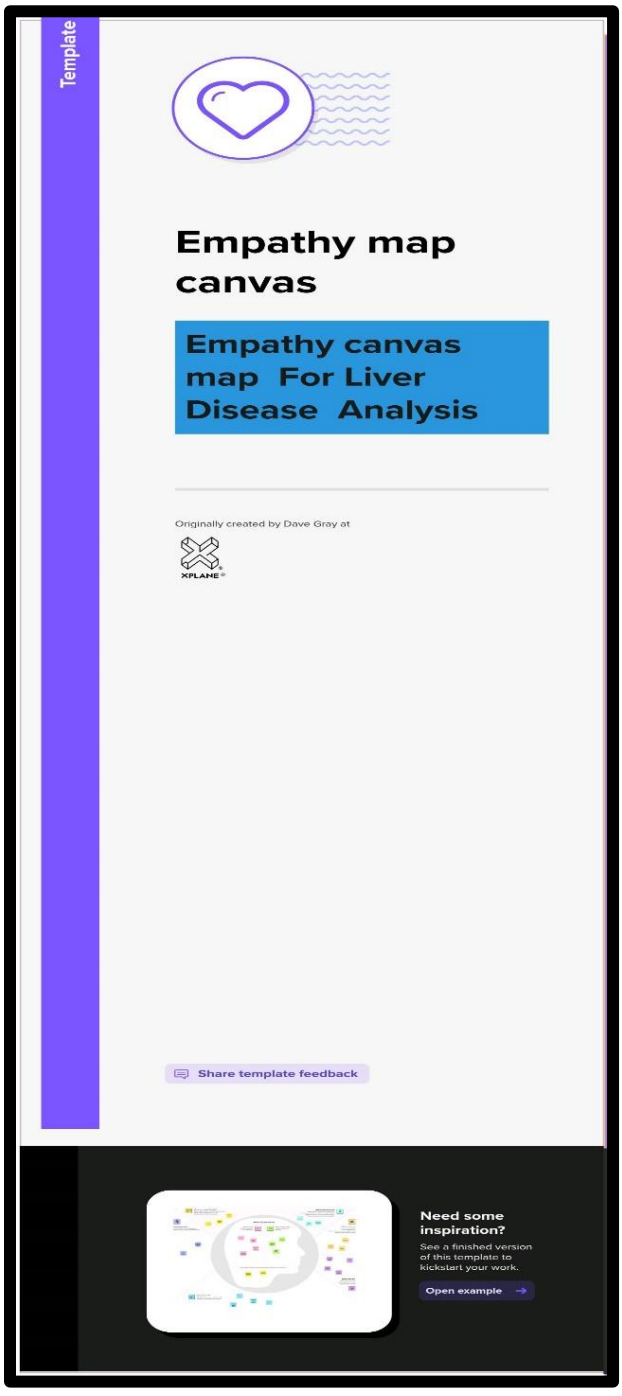
"Say" – This quadrant captures what the user says or verbalizes, such as their spoken words, complaints, or feedback.

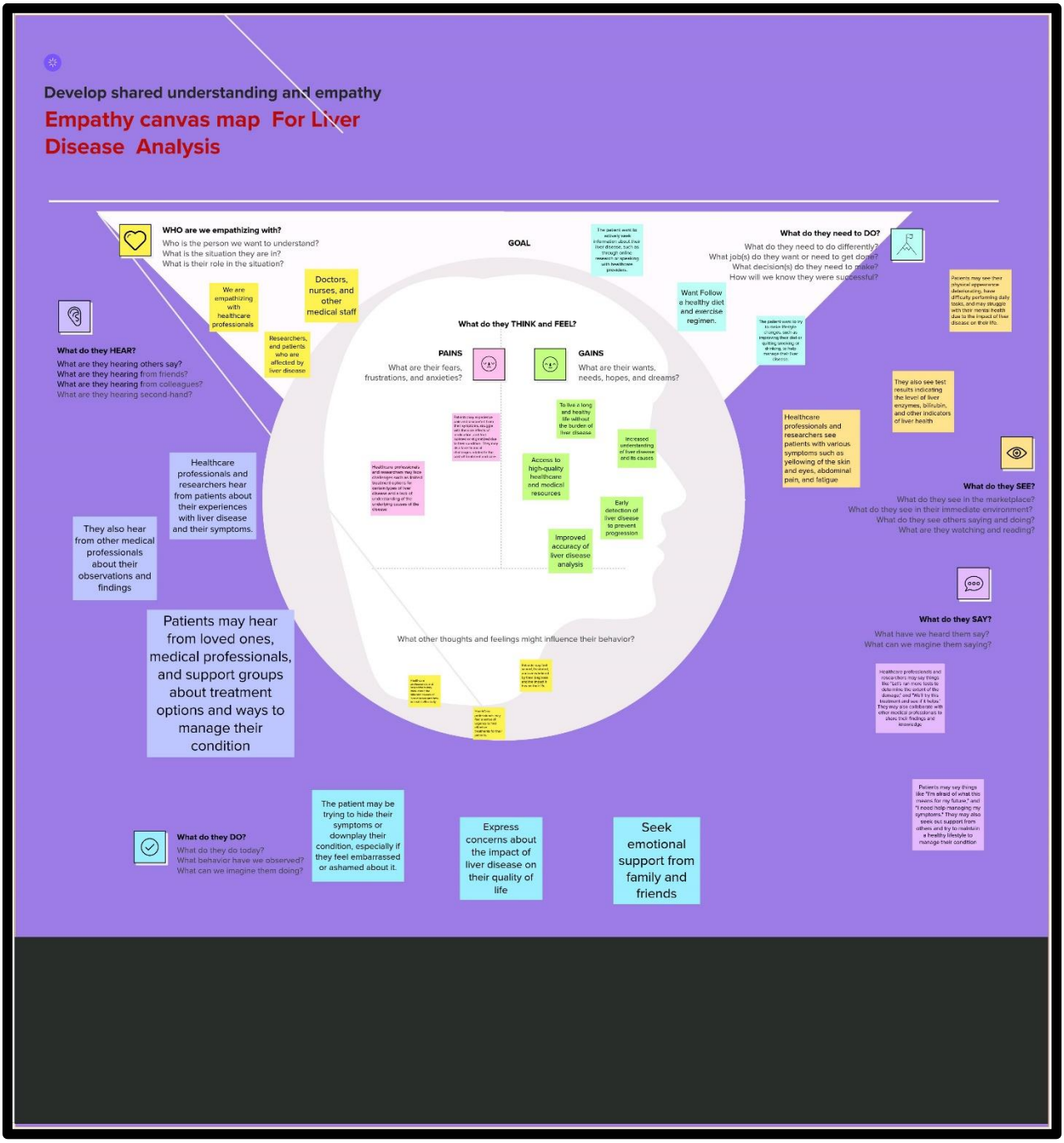
"Do" – This quadrant focuses on what the user does or their observable behaviors, such as their actions, gestures, and interactions.

"Think" – This quadrant delves into what the user thinks or their internal thoughts, beliefs, and assumptions.

"Feel" – This quadrant explores what the user feels or their emotions, motivations, and desires.

Empathy maps are typically filled out by a team through research and observation of users, and they help the team gain a deeper understanding of the user's perspective and needs. By visualizing and consolidating information in an empathy map, design teams can develop insights and uncover opportunities to create more user-centric solutions. Empathy maps can be used in various stages of the design process, from problem definition and ideation to prototyping and testing, to ensure that the user's perspective is considered throughout the design process.


Template
Empathy map canvas
Empathy canvas map For Liver Disease Analysis
Originally created by Dave Gray at
XPLANE
Share template feedback
Need some inspiration?
See a finished version of this template to kickstart your work.
Open example


Develop shared understanding and empathy
Empathy canvas map For Liver Disease Analysis
WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
We are empathizing with healthcare professionals
Doctors, nurses, and other medical staff
Researchers, and patients who are affected by liver disease
GOAL
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
Want Follow a healthy diet and exercise regimen.
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
GAINS
What are their wants, needs, hopes, and dreams?
To live a long and healthy life without the burden of liver disease
Increased understanding of liver disease and its causes
Access to high-quality healthcare and medical resources
Early detection of liver disease to prevent progression
Improved accuracy of liver disease analysis
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
Healthcare professionals and researchers hear from patients about their experiences with liver disease and their symptoms.
They also hear from other medical professionals about their observations and findings
Patients may hear from loved ones, medical professionals, and support groups about treatment options and ways to manage their condition
Healthcare professionals and researchers see patients with various symptoms such as yellowing of the skin and eyes, abdominal pain, and fatigue
They also see test results indicating the level of liver enzymes, bilirubin, and other indicators of liver health
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What can we imagine them saying?
What other thoughts and feelings might influence their behavior?
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
The patient may be trying to hide their symptoms or downplay their condition, especially if they feel embarrassed or ashamed about it.
Express concerns about the impact of liver disease on their quality of life
Seek emotional support from family and friends

3.2 IDEATION & BRAINSTORMING MAP

Ideation and brainstorming maps are visual tools used to generate and organize ideas during the creative process. They are often used in design thinking, innovation, and problem-solving workshops to encourage free-flowing and collaborative idea generation.

An ideation or brainstorming map typically consists of a central concept, problem, or theme in the middle of the map, with branches or nodes radiating outwards that represent different ideas, concepts, or solutions. These branches can further have sub-branches or nodes that capture related ideas or details. The map can be created on a whiteboard, a flip chart, a digital tool, or even on a piece of paper

Ideation and brainstorming maps are flexible and dynamic tools that encourage open thinking, collaboration, and exploration of different ideas. They can be used to capture a large number of ideas, stimulate further ideation, and help in identifying patterns, connections, and opportunities among the ideas generated

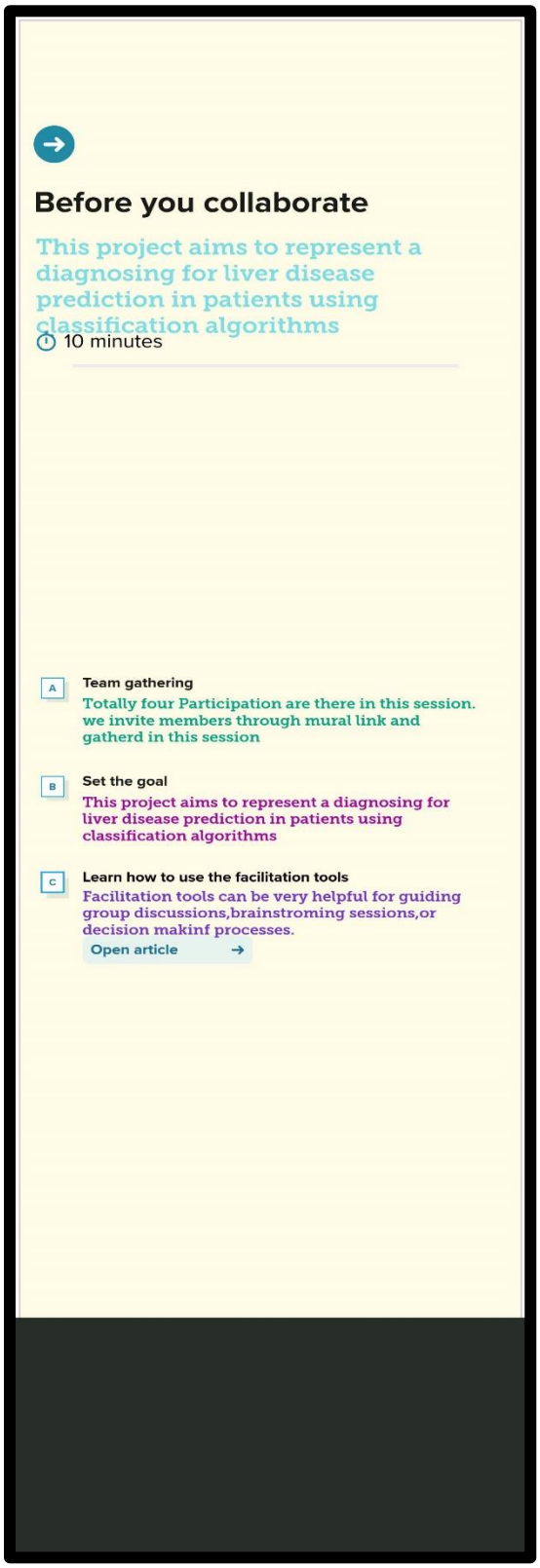
Template

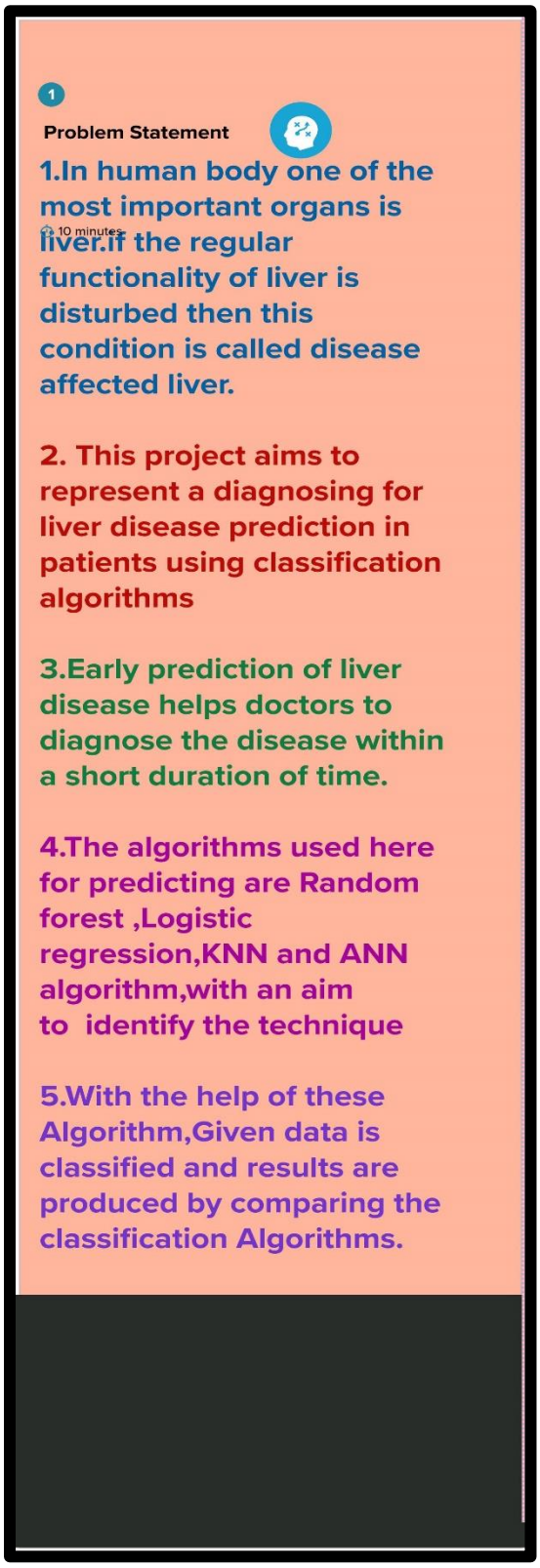
Brainstorm & idea prioritization

A Review of Liver Patient Analysis Methods Using Machine Learning

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback


Before you collaborate
This project aims to represent a diagnosing for liver disease prediction in patients using classification algorithms
10 minutes
A
Team gathering
Totally four Participation are there in this session. we invite members through mural link and gatherd in this session
B
Set the goal
This project aims to represent a diagnosing for liver disease prediction in patients using classification algorithms
C
Learn how to use the facilitation tools
Facilitation tools can be very helpful for guiding group discussions,brainstroming sessions,or decision makinf processes.
Open article


1
Problem Statement
1.In human body one of the most important organs is liver.If the regular functionality of liver is disturbed then this condition is called disease affected liver.
10 minutes
2. This project aims to represent a diagnosing for liver disease prediction in patients using classification algorithms
3.Early prediction of liver disease helps doctors to diagnose the disease within a short duration of time.
4.The algorithms used here for predicting are Random forest ,Logistic regression,KNN and ANN algorithm,with an aim to identify the technique
5.With the help of these Algorithm,Given data is classified and results are produced by comparing the classification Algorithms.

2

Brainstorm

Person Idea

10 minutes

Hardware eqipments are needed

Operating System is used

Person 1

GoogleCollab Notebook is used

Storage and internet is used

Person 4

Classification algorithm is used

Web FrameWork is build

Liver Patient Data set is used

Libraries are imported

Python language is used

Machine learning is used

Person 2

Tensorflow model is used

HTML pages are Build

Person 3

Logistic Regression is used

Randomforest algorithm

ANN Algorihm is used

KNN algorithm is used

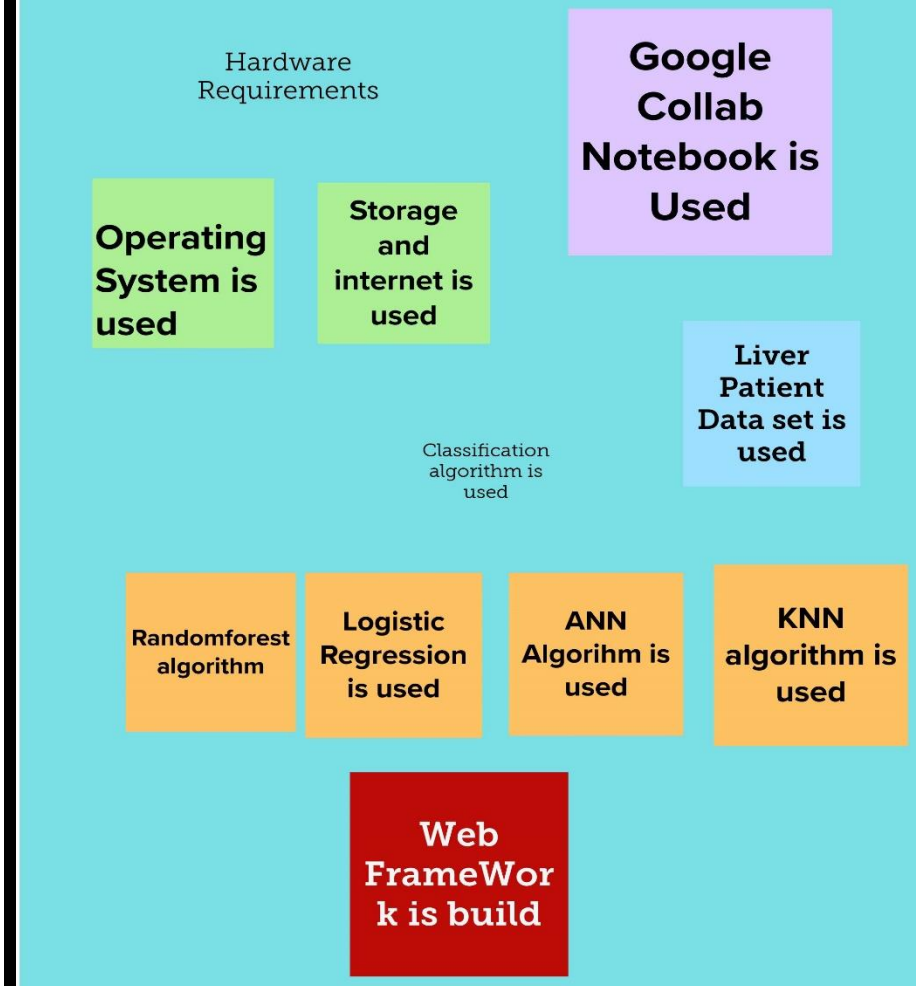
3

Group ideas

1.Hardware requirements is required
2.Google Collab Notebook is Used To Run the commands
3.Liver Patient Dataset is Used

20 minutes

4.Classification Algorithm Like Random forest, Logistic Regression, KNN and ANN Algorithm
5.HTML ,Web frame work is build and Python Language is used

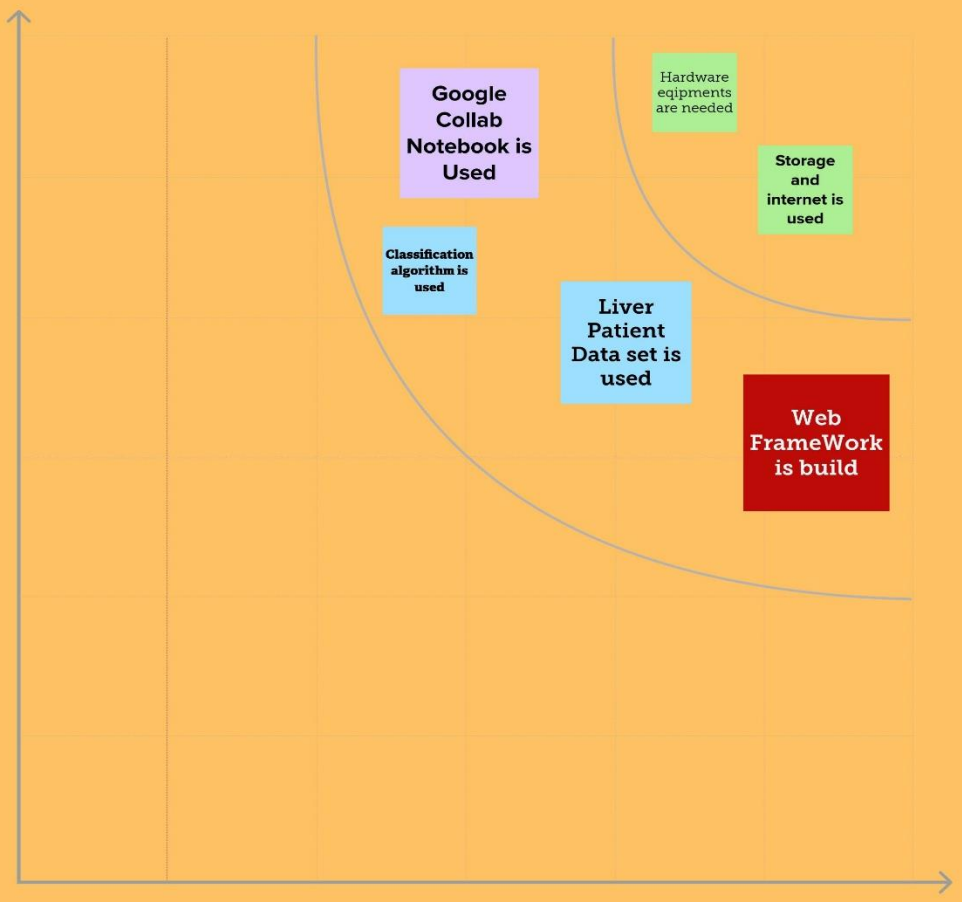


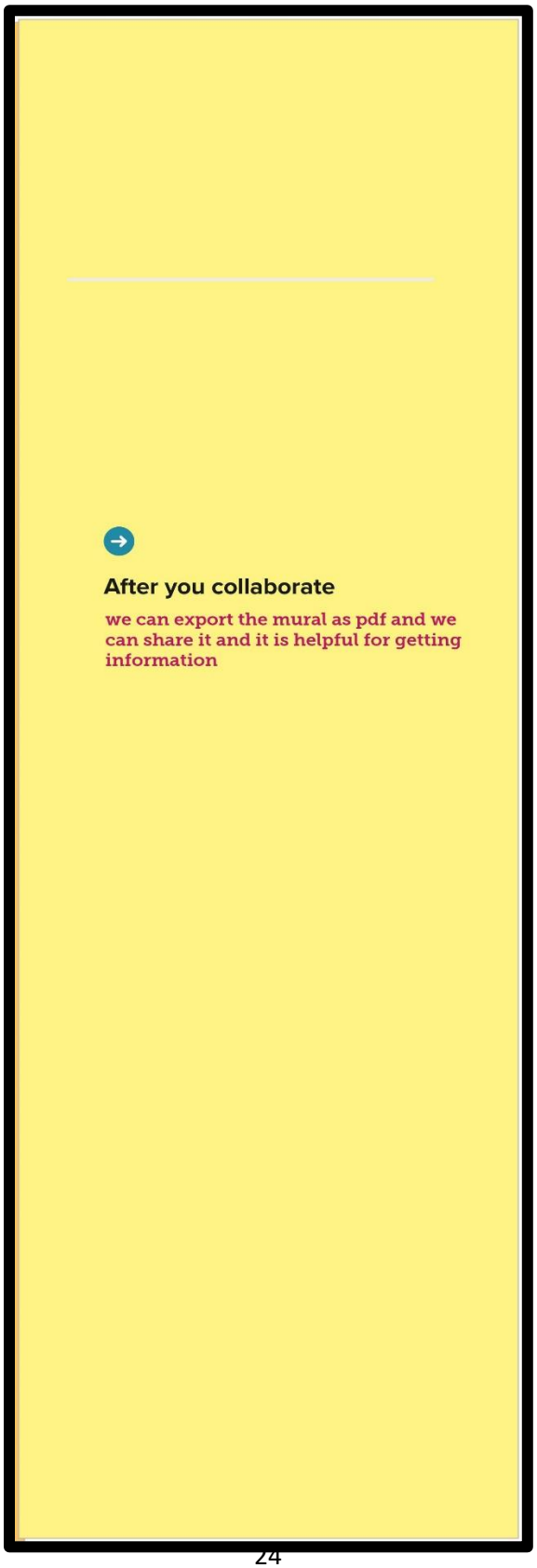
4

Prioritize

Group ideas are Prioritized

20 minutes




After you collaborate
we can export the mural as pdf and we can share it and it is helpful for getting information

PROJECT DESCRIPTION

4. PROJECT DESCRIPTION

4.1 MODULES:

DATA COLLECTION & PREPARATION

COLLECT THE DATASET

IMPORTING THE LIBRARIES

READ THE DATASET

DATA PREPARATION

HANDLING MISSING VALUES

HANDLING CATEGORICL VALUES

EXPLORATORY DATA ANALYSIS

DESCRIPTION STATISTICAL

VISUAL ANALYSIS

UNIVARIATE ANALYSIS

BIVARIATE ANALYSIS

MULTIVARIATE ANALYSIS

HANDLING IMBLANCED DATA

MODEL BUILDING

TESTING THE MODEL

PERFORMANCE TESTING & HYPERPARAMETER TUNING

TESTING MODEL WITH MULTIPLE EVALUATION METRICS

COMPARE THE MODEL

IDENTIFYING IMPORTANT FEATURES

MODEL DEPLOYMENT

SAVE THE BEST MODEL

INTEGRATE WITH WEB FRAMEWORK

BUILDING HTML PAGES

BUILD PYTHON CODE

RUN THE WEB APPLICATION

4.2 MODULE DESCRIPTION

DATA COLLECTION & PREPARATION

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So the section allows you to download the required dataset.

COLLECT THE DATASET

There are many popular open sources for collecting the data. E.g: Kaggle.com, UCI repository, etc. In this project we have used .csv data. This data is downloaded from Kaggle.com Please refer to the link given below to download the dataset.

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analysing techniques.

IMPORTING THE LIBRARIES

Import the necessary libraries as shown in the image. (Optional) Here we have used visualization style as fivethirtyeight.

READ THE DATASET

Our dataset format might be in.csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas. In pandas we have a function called read_csv() to read the dataset. As a parameter we have to give the directory of the csv file.

DATA PREPARATION

As we have understood how the data is, let's pre-process the collected data. The download data set is not suitable for training the machine learning model as it might have so much, randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

- HANDLING MISSING VALUES
- HANDLING CATEGORICAL DATA

HANDLING MISSING VALUES

Let's find the shape of our dataset first. To find the shape of our data, the df.shape method is used. To find the data type, df.info() function is used.

For checking the null values, df.isnull() function is used. To sum those null values we use.sum() function.

We can see that there are null values in the Albumin_and_Globulin_Ration Column. Let us check how many numbers of null records present in the Closing value column using sum() function.

From the above code of analysis, we can infer that columns such as Albumin and Globulin Ratio is having the missing values, we need to treat them in a required way.

We will fill in the missing values in the numeric data type using the mean value of that particular column and categorical data type using the most repeated values.

HANDLING CATEGORICAL VALUES

As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding. To convert the categorical features into numerical features we use encoding techniques. There are several techniques but in our project. We are using manual encoding with the help of list comprehension. In our project, for Gender, encoding is done.

EXPLORATORY DATA ANALYSIS

In this milestone, we will see the exploratory data analysis.

DESCRIPTIVE STTISTICAL

Descriptive analysis is to study the basic features of data with the statistical process. Here pandas has a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And e can find mean, std, min, max and percentile values of continuous feature.

VISUAL ANALYSIS

Visual analysis is the process of using visual representations, such s charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

UNIVARIATE ANALYSIS

In simple words, univariate analysis is understanding the data with a single feature. Here we have displayed two different graphs such as distplot and countplot. The Seaborn package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot.

In our dataset we have some categorical features. With the count plot function, we are going to count the unique category in those features.

COUNTPLOT:-

A count plot can be thought of as a histogram across a categorical, instead of quantitative, variable. The basic API and options are identical to those for barplot(), so you can compare counts across nested variables.

BIVARIATE ANALYSIS

From the graph we can infer that, gender and outcome is a categorical variables with 2 categories, from gender column we can infer that 1- category is having more weightage than category-0, and outcome with 0, it means healthy is a underclass when compared with category-1, which means liver patient.

MULTIVARIATE ANALYSIS

In simple words, multivariate anlysis is to find the relation between multiple features. Here we have used a het plot from the seaborn package.

Now, the code would be normalizing the data by scaling it to have a similar range of values, and then splitting that data into a training set and a test set for training the model and testing its performance, respectively.

SCALING THE DATA

Scaling is one the important process, we have to perform on the dataset, because of data measures in different ranges can leads to mislead in prediction. Models such as KNN, Logistic regression need scaled data, as they follow distance based method and Gradient Descent concept

We will perform scaling only on the input values. Once the dataset is scaled, it will be converted into an array and we need to convert it back to a dataframe

SPLITTING DATA INTO TRAIN AND TEST

Now let's split the Dataset into train and test sets

CHANGES: first the dataset into x and y then split the data set

Here x and y variables are created. On x variables, df is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using the train_test_split() function from Sklearn. As parameters, we are passing x,y test_size, random state.

HANDLING IMBALANCE DATA

Data Balancing is one of the most important step, which need to be performed for classification models, because when we train our model on imbalanced dataset, we will get biased results, which means our model is able to predict only one class element

For balancing the data we are using the SMOTE Method.

SMOTE: Synthetic minority over sampling techniques, which will create new synthetic data points for under class as per the requirements given by us using KNN method.

From the above picture, we can infer that, previously our dataset had 329 class 1, and 132 class items, after applying smote technique on the dataset the size has become equal.

MODEL BUILDING

TESTING THE MODEL

This code defines a function named “predict_exit” which takes in a sample_values as an input. The function then converts the input sample_value from a list t a numpy array. It reshapes the sample_value array as it contains only one record. Then, it applies feature scaling to the reshaped sample_value array using a scaler object ‘scale’ that should have been previously defined and fitted. Finally, the function returns the prediction of the classifier on the scaled sample_value

PERFORMENCE TESTING & HYPERPARAMETER TUNING

TESTING MODEL MULTIPLE EVALUATION METRICS

Multiple evalution metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using accuracy, score to compare between models.

COMPARE THE MODEL

For comparing the above four models, the Accuracy function is defined.

After calling the function, the results of models are displayed as output. From the five models Random Forest Classifier is performing well. From the above image, We can see the

IDENTIFYING IMPORTANT FEATURES

10 attributes are passed to predict the actual outcome, Its necessary to identify the l important feature to determine the output. Here we are using function called feature_importance to identify the important features among the available attributes and understand with a visualization.

MODEL DEPLOYMENT

SAVE THE BEST MODEL

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.

INTEGRATE WITH WEB FRAMEWORK

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- Building HTML Pages
- Building server side script
- Run the web application

BUILDING HTML PAGES

For this project create three HTML files namely

- home.html
- predict.html
- index.html

and save them in the templates folder.

BUILD PYTHON CODE

Import the libraries. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument. And render HTML page:

Here we will be using a declared constructor to route to the HTML page which we have created earlier.

In the above example, '/' URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

RETRIEVES THE VALUE FROM UI:

Here we are routing our app to predict() function. This function retrieves all the values from the HTML page using Post request. That is stored in an array. This array is passed to the model.predict() function. This function returns the prediction. And this prediction value will be rendered to the text that we have mentioned in the submit.html page earlier.

RUN THE WEB APPLICATION

- Open anaconda prompt from the start menu
- Navigate to the folder where your python script is.
- Now type "python app.py" command
- Navigate to the localhost where you can view your web page.
- Click on the predict button from the top left corner, enter the inputs, click on the submit button, and see the result/prediction on the web.

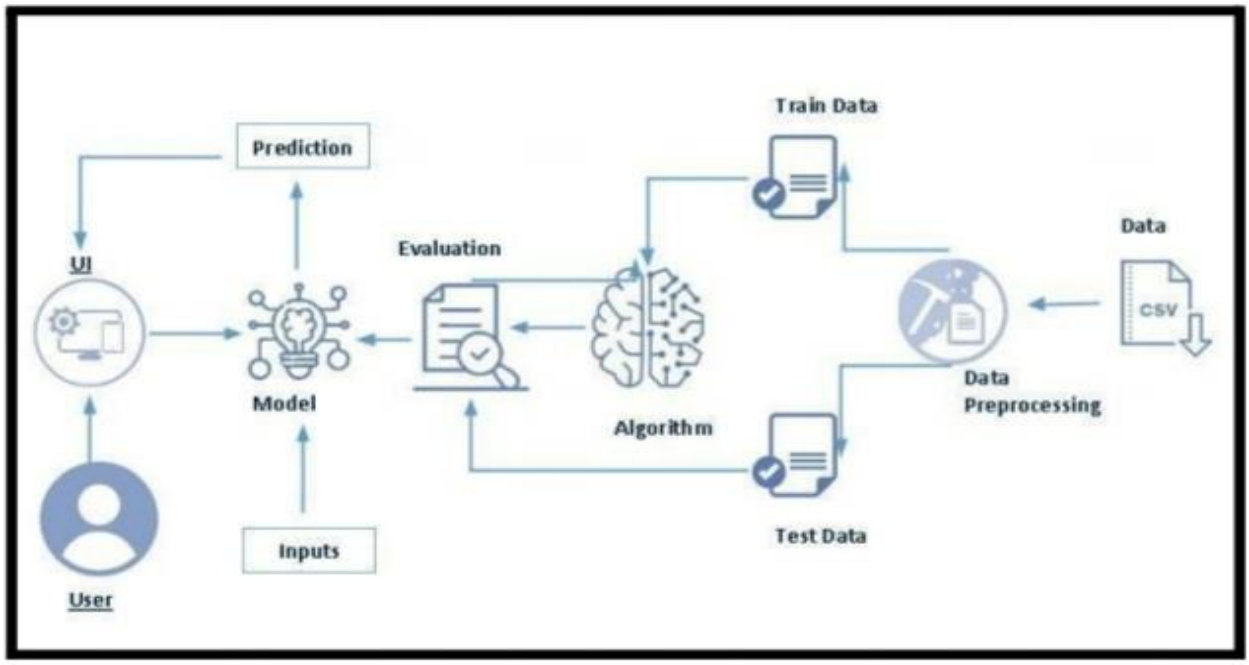
Now, Go the web browser and write the localhost url (http://127.0.0.1:5000) to get the below result.

Now, when you click Go to predict the button from the banner you will get redirected to the prediction page.

Inputs- Now, the user will give inputs to get the predicted page after giving details user has to click on Predict Button to get the result.

4.3 DATA FLOW DIAGRAM

TECHNICAL ARCHITECTURE:



ADVANTAGES

5.1 ADVANTAGES

Liver patient analysis using machine learning methods has become an important research area in recent years. Machine learning algorithms have several advantages over traditional statistical methods in analyzing liver patient data. Some of the key advantages of using machine learning methods for liver patient analysis are:

Ability to handle large datasets: Machine learning algorithms can handle large amounts of data, which is crucial in analyzing complex medical data such as liver patient data. This allows for more accurate and reliable analysis.

Improved accuracy: Machine learning algorithms are designed to learn from data and improve their accuracy over time. This allows for more accurate predictions and diagnoses of liver diseases.

Ability to identify hidden patterns: Machine learning algorithms can identify patterns and relationships in data that are not immediately apparent. This can lead to new insights into liver diseases and their underlying causes.

Personalized medicine: Machine learning algorithms can be used to develop personalized treatment plans for liver patients based on their individual characteristics and medical history.

Automation: Machine learning algorithms can automate many of the tasks involved in liver patient analysis, such as data preprocessing and feature selection. This can save time and improve efficiency.

APPLICATIONS

6.1 APPLICATIONS

Liver patient analysis using machine learning methods has become an important area of research, with several machine learning applications being developed to aid in liver disease diagnosis, prediction, and treatment. Some of the most commonly used machine learning applications in liver patient analysis include:

Support Vector Machines (SVMs): SVMs are commonly used in liver disease diagnosis and prediction. SVMs can classify patients as having liver disease or being healthy based on their clinical and demographic data. SVMs are also used to predict the severity of liver disease and the likelihood of liver disease progression.

Artificial Neural Networks (ANNs): ANNs are used in liver disease diagnosis and prediction. ANNs can be trained to recognize patterns in liver patient data and make predictions about disease outcomes. ANNs have been used to predict liver fibrosis and cirrhosis progression.

Decision Trees (DTs): DTs are used in liver disease diagnosis and prediction. DTs can identify important features in liver patient data and classify patients based on those features. DTs have been used to predict the risk of hepatocellular carcinoma in patients with liver cirrhosis.

Random Forests (RFs): RFs are used in liver disease diagnosis and prediction. RFs can combine the predictions of multiple decision trees to improve the accuracy of liver disease prediction. RFs have been used to predict the risk of liver fibrosis in patients with chronic liver disease.

Deep Learning (DL): DL is a type of machine learning that is used in liver disease diagnosis and prediction. DL algorithms can automatically learn features from liver patient data, which can improve the accuracy of liver disease prediction. DL has been used to predict liver fibrosis and cirrhosis progression.

CONCLUSION AND FUTURE ENHANCEMENT

7.CONCLUSION

In conclusion, the analysis of liver patient data using machine learning techniques, specifically classification algorithms like Random Forest, Logistic Regression, KNN, and ANN, has shown promising results for identifying potential liver diseases. These algorithms have been utilized to build predictive models that can assist in early diagnosis and treatment planning. These algorithms are used to predict the liver disease at an early stage and the comparative analysis will help to predict the liver disease and will benefit in managing the health of the individuals.

8. FUTURE ENHANCEMENTS

Here are some potential future enhancements for a liver patient analysis and prediction project that uses classification algorithms such as Random Forest, Logistic Regression, KNN, and ANN:

Feature Engineering: Feature engineering is the process of selecting, transforming, and creating new features from the raw data to improve the performance of machine learning models. In the future, you could explore different feature engineering techniques to extract more relevant features from the liver patient data. This could include domain-specific feature engineering, such as incorporating liver-specific biomarkers or clinical indicators, or exploring advanced feature extraction techniques such as deep feature learning or feature selection algorithms.

Model Ensemble: Ensemble methods, such as stacking, bagging, and boosting, can often improve the prediction accuracy of machine learning models. In the future, you could explore ensemble techniques to combine the predictions of multiple classification algorithms, such as Random Forest, Logistic Regression, KNN, and ANN, to create a more robust and accurate prediction model for liver patient analysis. This could involve experimenting with different ensemble strategies and evaluating their performance on the dataset.

Hyperparameter Tuning: Hyperparameters are parameters of machine learning algorithms that are not learned during training but need to be tuned to optimize model performance. In the future, you could explore different hyperparameter tuning techniques, such as grid search, random search, or Bayesian optimization, to find the optimal hyperparameter settings for the classification algorithms used in your liver patient analysis project. This could help improve the performance and generalization of the models

Deployment and Integration: In the future, you could consider deploying the developed classification models into a real-world clinical setting, integrating them into a healthcare system or electronic health record (EHR) system for practical use. This could involve addressing challenges related to data privacy, security, regulatory compliance, and model deployment in a clinical environment. Deployment and integration of machine learning models in healthcare settings require careful consideration of ethical, legal, and practical aspects, and could be a potential future enhancement for your liver patient analysis project.

Continual Model Updating: Healthcare data is dynamic and constantly evolving, and machine learning models may need to be updated periodically to maintain their accuracy and relevance. In the future, you could consider implementing a continual model updating strategy, where the classification models are updated with new data over time to ensure their performance remains optimal. This could involve techniques such as online learning, transfer learning, or domain adaptation to adapt the models to changing data distributions or clinical practices.

APPENDIX

9.APPENDIX

9.1 SAMPLE CODING

```
import pandas as pd
import numpy as np
import seaborn as sns
import sklearn
import matplotlib.pyplot as plt
from matplotlib import rcParams
from scipy import stats
from pandas import DataFrame
import pip
data=pd.read_csv('/indian_liver_patient.csv')
data.head()
data.info()
data.isnull().any()
data.isnull().sum()
#checking for the missing data after cleansing data(
data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].median(),inplace=True)
data.isnull().sum()
from sklearn.preprocessing import LabelEncoder
lc = LabelEncoder()
```

```
data['Gender']=lc.fit_transform(data['Gender'])

data.describe()

sns.distplot(data['Age'])

plt.title('Age distribution Graph')

plt.show()

sns.countplot(data ['Dataset'],

hue=data['Gender'])

plt.figure(figsize=(10,7))

sns.heatmap(data.corr(),annot=True)

from sklearn.preprocessing import scale

X_scaled=pd.DataFrame(scale(X),columns=X.columns)

X_scaled.head()

X=data.iloc[:,:-1]

y=data.Dataset

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test= train_test_split(X_scaled,y,test_size=0.2, random_state=42)

pip install imblearn

from imblearn.over_sampling import SMOTE

smote = SMOTE()

y_train.value_counts()

X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

```
y_train_smote.value_counts()
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fit(X_train_smote, y_train_smote)
rf_predict=model1.predict(X_test)
rfc1=accuracy_score(y_test,y_predict)
rfc1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test,y_predict))
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
model4=DecisionTreeClassifier()
model4.fit(X_train_smote, y_train_smote)
y_predict=model4.predict(X_test)
dtc1=accuracy_score(y_test,y_predict)
dtc1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test, y_predict))
```

```
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
model2=KNeighborsClassifier()
model2.fit(X_train_smote, y_train_smote)
y_predict = model2.predict(X_test)
knn1=accuracy_score(y_test, y_predict)
knn1
pd.crosstab(y_test, y_predict)
print(classification_report(y_test, y_predict))
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
model5=LogisticRegression()
model5.fit(X_train_smote, y_train_smote)
y_predict=model5.predict(X_test)
logi1=accuracy_score(y_test, y_predict)
logi1
pd.crosstab(y_test,y_predict)
print (classification_report (y_test, y_predict))
import tensorflow.keras
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
#intialising the ANN
classifier = Sequential()
#adding the input layer and the first hidden layer
classifier.add(Dense(units=100, activation='relu', input_dim=10))
#adding the second hidden layer
classifier.add(Dense(units=50, activation='relu'))
#adding the output layer
classifier.add(Dense(units=1, activation='sigmoid'))
#compiling the ANN
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
#fitting the ANN to the training set
model_history = classifier.fit(X_train,y_train, batch_size=100,validation_split=0.2, epochs=100)
model4.predict([[50,1,1,2,0.8,150,70,80,7.2,3.4,0.8]])
model1.predict([50,1,1.2,0.8,150,70,80,7.2,3.4,0.8])
classifier.save("liver.h5")
y_pred = classifier.predict(X_test)
y_pred
def predict_exit(sample_value):
 sample_value = np.array(sample_value)
```

```
 sample_value = sample_value.reshape(1,-1)
 sample_value = scale(sample_value)
 return classifier.predict(sample_value)
sample_value =[[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]
if predict_exit(sample_value )>0.5:
print('prediction:liver patient')
else:
print('prediction:healthy')
acc_smote= [['KNN classifier',knn1],['RandomForestClassifier',rfc1],
['DecisionTreeClassifier',dtc1],['LogisticRegression',logi1]]
Liverpatient_pred= pd.DataFrame(acc_smote, columns=['classification models','accuracy_score']
Liverpatient_pred
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
plt.figure(figsize=(7,5))
plt.xticks(rotation=90)
plt.title('classificaton models & accuracy scores after SMOTE',fontsize=18)
sns.barplot(x="classification models", y="accuracy_score", data=Liverpatient_pred)
from sklearn.ensemble import ExtraTreesClassifier
model=ExtraTreesClassifier()
model.fit(X,y)
```

```
model.feature_importances_

dd=pd.DataFrame(model.feature_importances_,index=X.columns).sort_values(0,ascending=Fals
e)

dd

dd.plot(kind='barh',figsize=(7,6))

plt.title("FEATURE IMPORTANCE",fontsize=14)

import joblib

joblib.dump(model1, 'ETC.pk1')
```

9.2 SAMPLE CODING

1.INDEX.HTML

```
<!DOCTYPE html>

<html>

<head>

<title>LIVER PREDICTION</title>

<link rel="stylesheet" type="text/css" href="css/style.css">

<link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>

<header>

<div>

<ul>

<li class="active"><a href="index.html"><i class="fa fa-home"></i>HOME</a></li>

<li><a href="home.html">GO TO PREDICT</a></li>

</ul>

</div>

<div class="title">

<br><br><br><br><br><br><br><br><br><h1>INTRODUCTION<br><br>Liver is the largest internal organ in the human body,it is essential for digesting food and releasing the
```

toxic element of the body and plays a major role in metabolism and serving several vital functions. The

liver is the largest glandular organ of the body. It weighs about 3 lb (1.36 kg) .The liver's main job is to

strain the blood coming from the digestive tract, before passing it to the rest of the body. The liver also

detoxifies chemicals and metabolizes drugs.
As it does so, the liver hides bile that ends up back in the

intestines. The liver also makes proteins important for blood clotting and other functions.The liver supports

almost every organ in the body and is vital for our survival. Liver diseases are diagnosed based on the liver functional test.

Some of the diseases are Wilson's disease,liver cancer, and cirrhosis.

</header>

</body>

</html>

2. HOME.HTML

```
<!DOCTYPE html>
<html>
<head>
<title>LIVER PATIENT ANALYSIS</title>
<link rel="stylesheet" href="style1.css" type="text/css">
</head>
<body>
<div class="main">
<div class="liver patient analysis">
<h1>LIVER PATIENT ANALYSIS</h1>
<form id="liver" method="post">
<label>AGE:</label>
<br>
<input type="text" name="age" id="age" placeholder="enter your age">
<br><br>
<label>GENDER:</label>
<br>
   
<input type="radio" name="gender" id="male" >
<span id="male">male</span>
 
<input type="radio" name="gender" id="female" >
<span id="Female">female</span>
```

```
<br><br>
<label>TOTAL _BILIRUBIN:</label>
<br>
<input type="text" name="total_bilirubin" id="total_bilirubin" >
<br><br>
<label>DIRECT_BILIRUBIN:</label>
<br>
<input type="text" name="direct_bilirubin" id="direct_bilirubin" >
<br><br>
<label>ALKALINE_PHOSPOTASE:</label>
<br>
<input type="text" name="alkaline_phospotase" id="alkaline_phospotase" >
<br><br>
<label>ALAMINE_AMINOTRANSFERASE:</label>
<br>
<input type="text" name="alamine_aminotransferase" id="alamine_aminotransferase" >
<br><br>
<label>ASPARTATE_AMINOTRANSFERASE:</label>
<br>
<input type="text" name="aspartate_aminotransferase" id="aspartate_aminotransferase" >
<br><br>
<label>TOTAL _PROTEINS:</label>
<br>
<input type="text" name="total_proteins" id="total_proteins" >
<br><br>
<label>ALBUMIN:</label>
<br>
```

```
<input type="text" name="albumin" id="albumin" >
<br><br>
<label>ALBUMIN_AND_GLOBULIN_RATIO:</label>
<br>
<input type="text" name="albumin_and_globulin_ratio" id="albumin_and_globulin_ratio" >
<br><br>
</form>
<header>
<div class="button">
<a href="Predict.html"class="btn">PREDICT</a><center></center></a><br><br>
</div>
</header>
</body>
</html>
```

3. PREDICT.HTML

```
<!DOCTYPE html>
<html>
<head>
<title>LIVER PATIENT ANALYSIS</title>
<link rel="stylesheet" href="style2.css" type="text/css">
</head
<div class="col-md-3"></div>
<div class="col-md-6">
```

```
<body>

<div class="row"  style="margin-bottom: 477px;">

<div class="card card-body alert alert-danger"><br><br><br><br><br><center>You have a liver desease problem,You must and should consult a docter.Take care.</center></div><br>

<div class="card card-body alert alert-success"><br><br><br><br><center>You dont have a liver desease problem.</center></div><br><br>

<div class="row">

<div class="col-md-4"></div>

<a href="index.html"><center>Back to Index</center></a>

<div class="col-md-4"></div>

</div>

</div>

<div class="col-md-3"></div

</div>

</body>

</html>
```

STYLE 1

```
{
margin:0;
padding:0;
}
body{
background:url("liver.jpg");
background-size:cover;
background-position:center;
background-repeat:no-repeat;
}
div.main{
width:400px;
margin:100px auto 0px auto;
}
h1{
text align: center;
color:yellow;
padding: 20px;
font-family: sans-serif;
}
div.liver{
background-color:rgba(0,0,0,0.5);
width:100%;
font-size:18px;
border-radius:35px;
```

```
border:1px soid rgba(255,255,255,0.3);
box-shadow:2px,2px,15px
rgba(0,0,0,0.3);
color:#fff;
}
form#liver{
margin:40px;
}
label{
font-family:sans-serif;
font-size:18px;
font-style:italic;
}
input#name{
width:300px;
border:1px solid #ddd;
border-radius:3px;
outline:0;
padding:7px;
background-color:#fff;
box-shadow:inset 1px 1px 5px;
}
.btn{
border: 5px solid #fff;
padding:5px 20px;
color:yellow;
transition: 0.6s ease;
```

```
font-size: 18px;
text-decoration:center;
text align: center;
}
label,span,h1{
text-shadow:1px 1px 5px
rgba(0,0,0,0.3);
}
```

STYLE 2

```
{
margin:0;
padding:0;
}
body{
background:url("liver2.jpg");
background-size:cover;
background-position:center;
background-repeat:no-repeat;
}
div.main{
width:400px;
margin:100px auto 0px auto;
}
h1{
text align: center;
color:yellow;
padding: 20px;
```

```
font-family: sans-serif;
}
div.row{
background-color:rgba(0,0,0,0.5);
width:100%;
font-size:30px;
font_family:Times New Roman;
border-radius:35px;
border:1px soid rgba(255,255,255,0.3);
box-shadow:2px,2px,15px
rgba(0,0,0,0.3);
color:#fff;
}
form#liver{
margin:40px;
}
label{
font-family:sans-serif;
font-size:18px;
font-style:italic;
}
input#name{
width:300px;
border:1px solid #ddd;
border-radius:3px;
outline:0;
padding:7px;
```

```
background-color:#fff;
box-shadow:inset 1px 1px 5px;
}
.btn{
border: 5px solid #fff;
padding:5px 20px;
color:yellow;
transition: 0.6s ease;
font-size: 18px;
text-decoration:center;
text align: center;
}
label,span,h1{
text-shadow:1px 1px 5px
rgba(0,0,0,0.3);
}
```

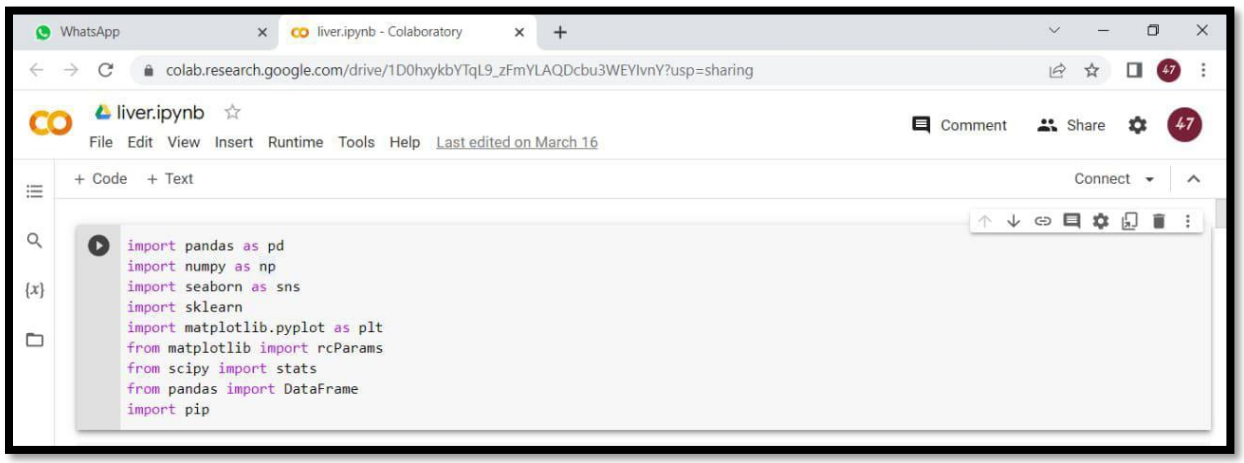
9.3 SAMPLE CODING

```
from flask import Flask,render_template,request
import numpy as np
import pickle
app=Flask(__name__)
@app.route('/')
def home():
return render_template('home.html')
@app.route('/predict')
def index():
return render_template("index.html")
@app.route('/data_predict', methods=['POST'])
def predict():
age= request.form['age']
Gender = request.form['Gender']
tb= request.form['tb']
db = request.form['db']
ap = request.form['ap']
aa1 = request.form['aa1']
aa2 = request.form['aa2']
tp = request.form['tp']
a = request.form['a']
agr = request.form['agr']
```

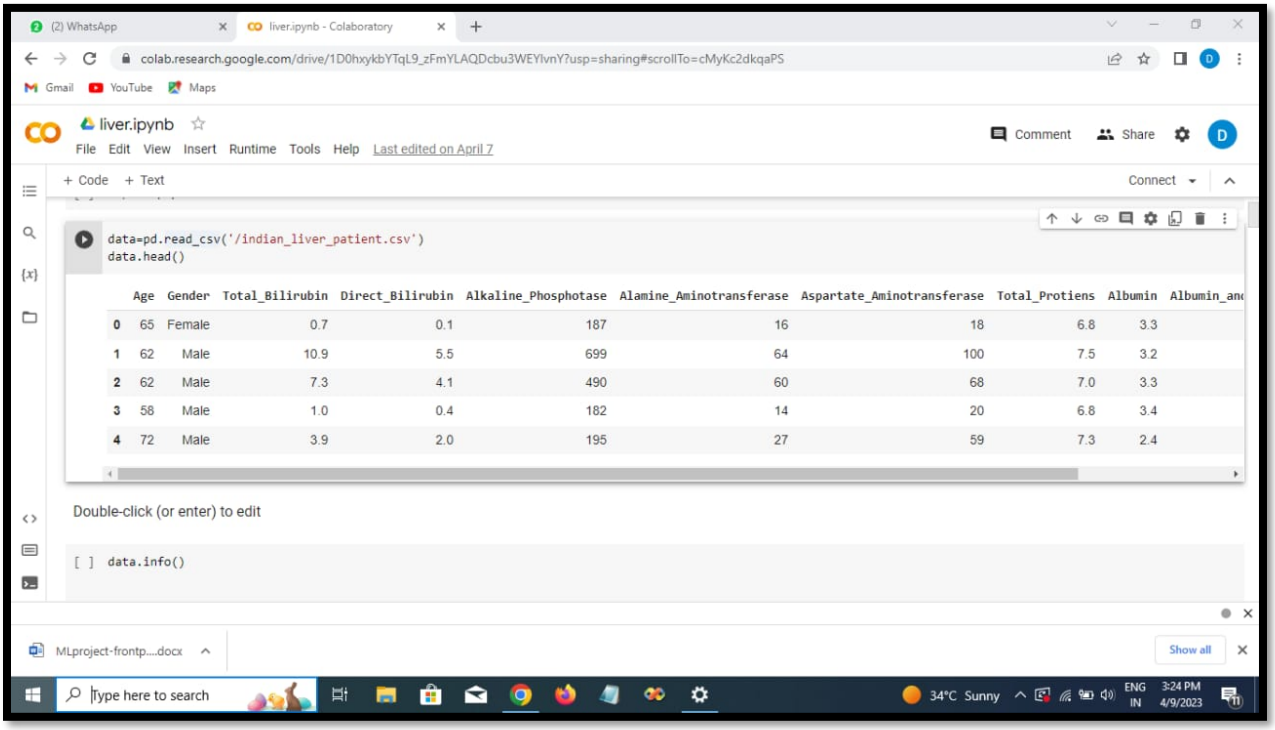
```
#coverting data into float format

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

# Loading model which we saved

model = pickle.load(open('liver_analysis.pkl', 'rb'))

prediction= model.predict(data)[0]

if(prediction==1):

return render_template('noChance.html', prediction='You have a liver desease
problem, You must and')

else:

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

if__name__=='__main__'

app.run()
```

10.RESULT SCREEN LAYOUT

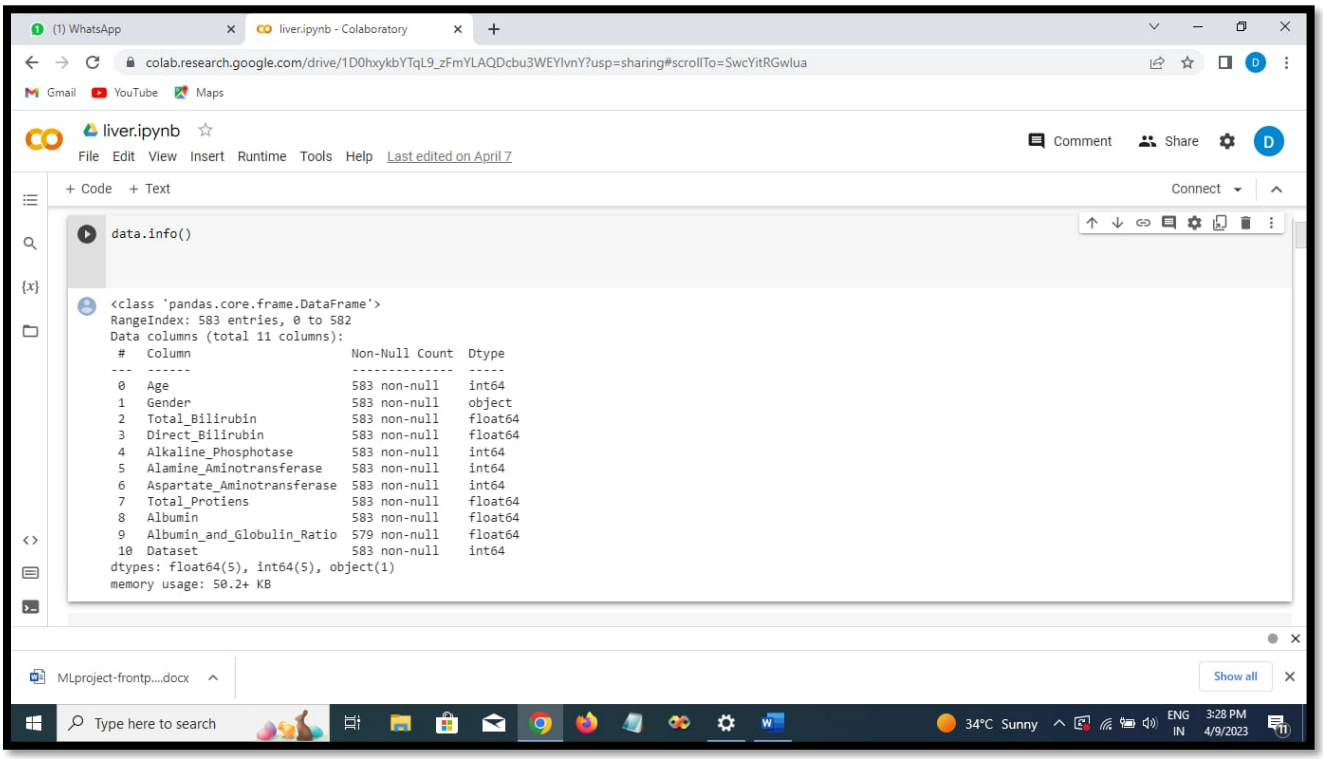
IMPORTING THE LIBRARIES

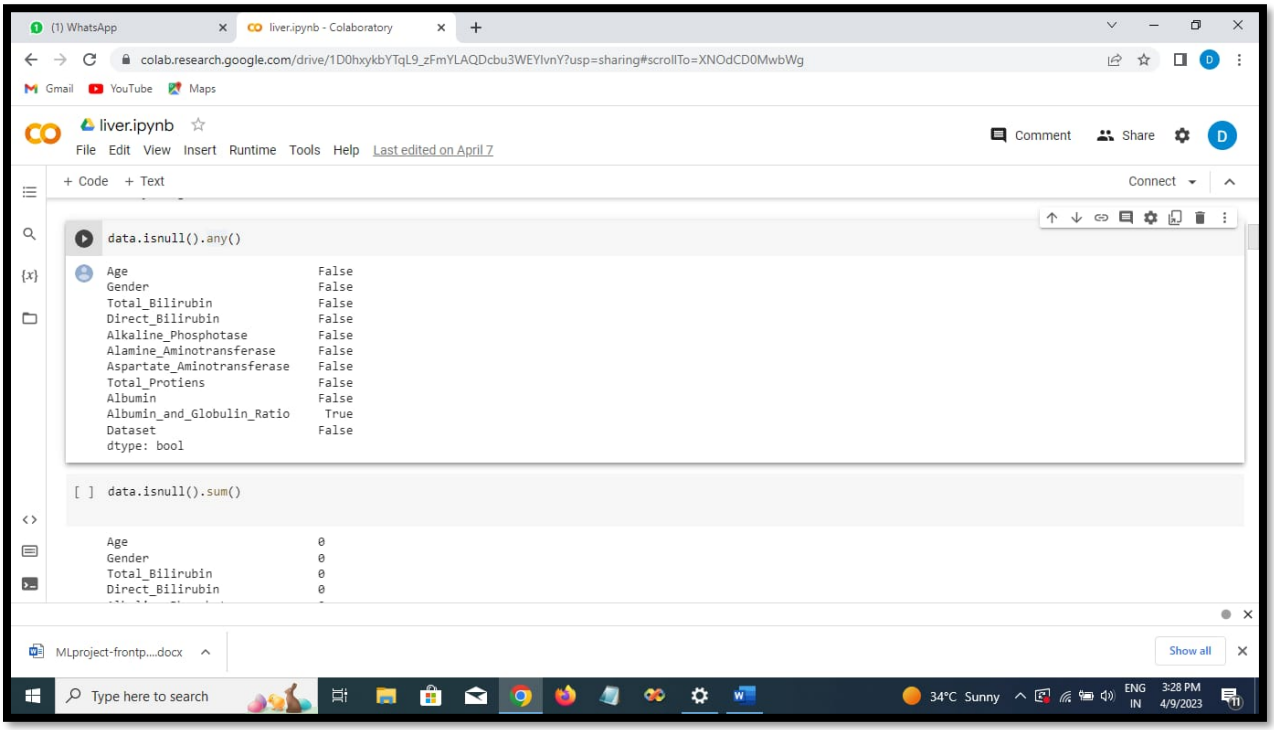


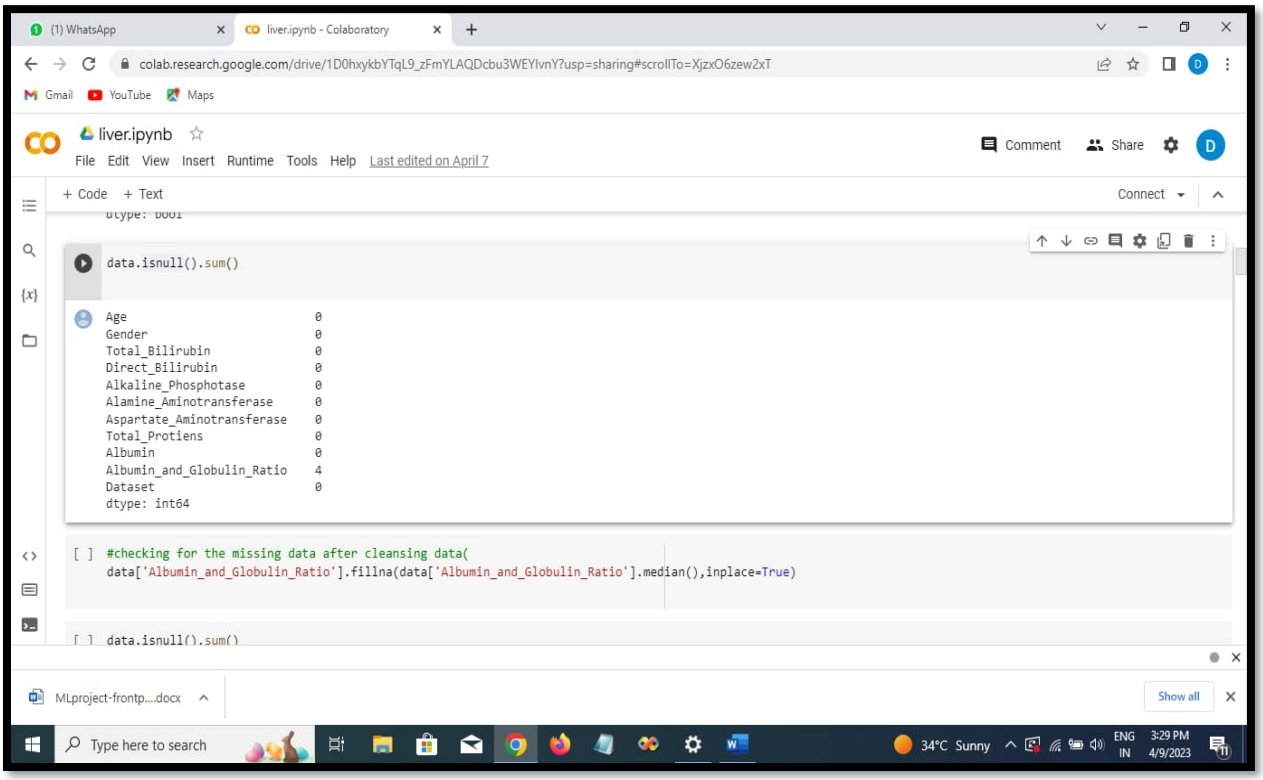
READ THE DATASET

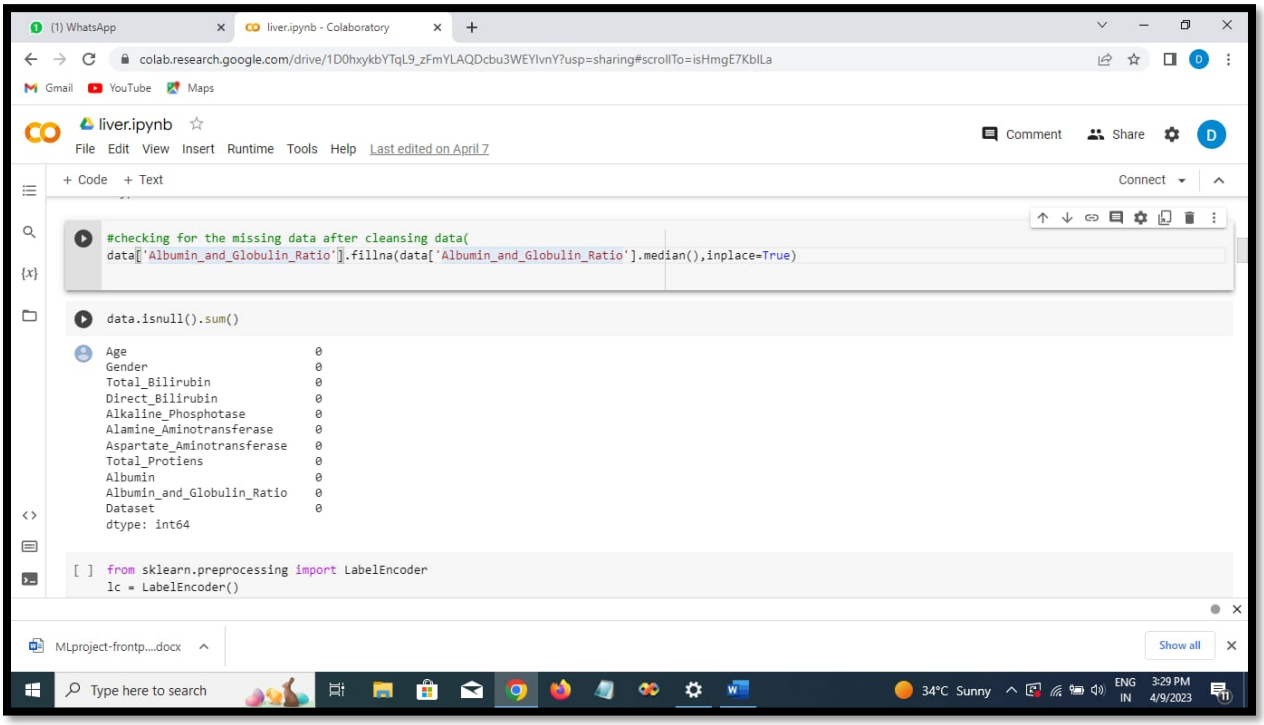


HANDLING MISSING VALUES

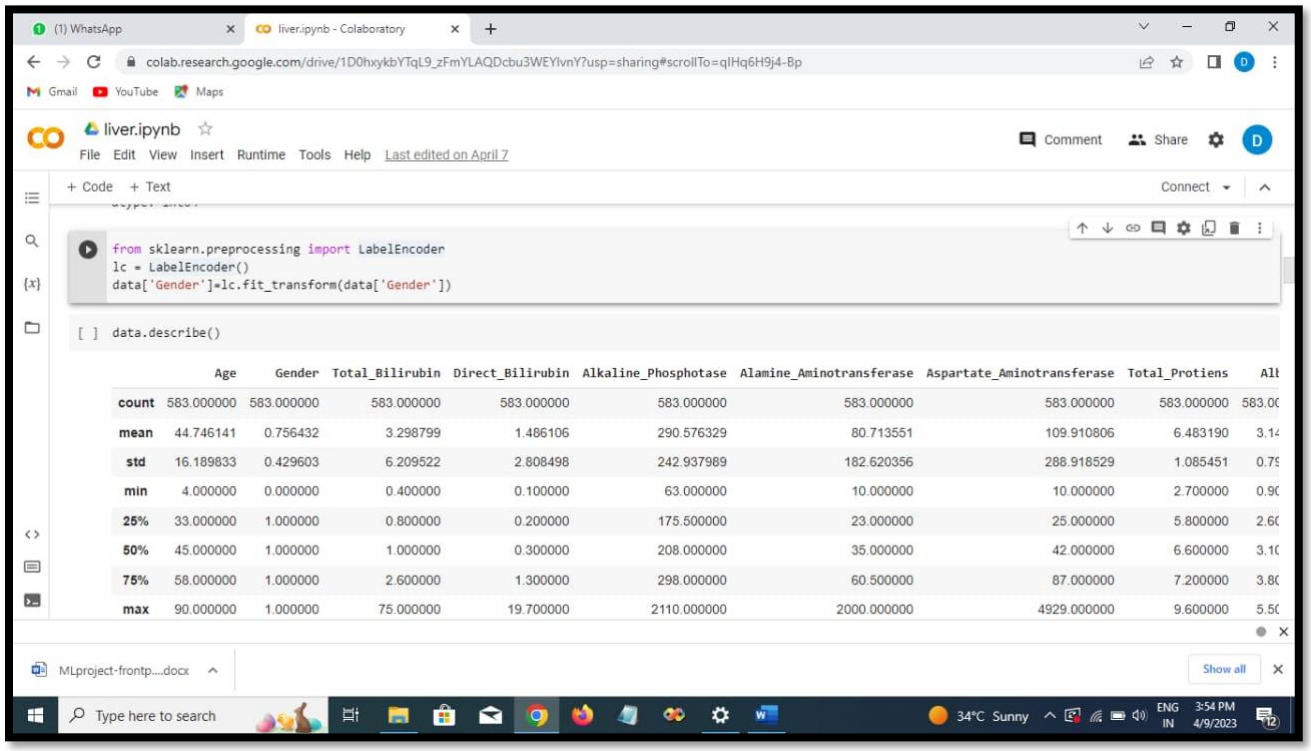



liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=XNOdCD0MwbWg
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
data.isnull().any()
Age False
Gender False
Total_Bilirubin False
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
[ ] data.isnull().sum()
Age 0
Gender 0
Total_Bilirubin 0
Direct_Bilirubin 0
MLproject-frontp....docx
Show all
Type here to search
34°C Sunny
ENG IN
3:28 PM 4/9/2023

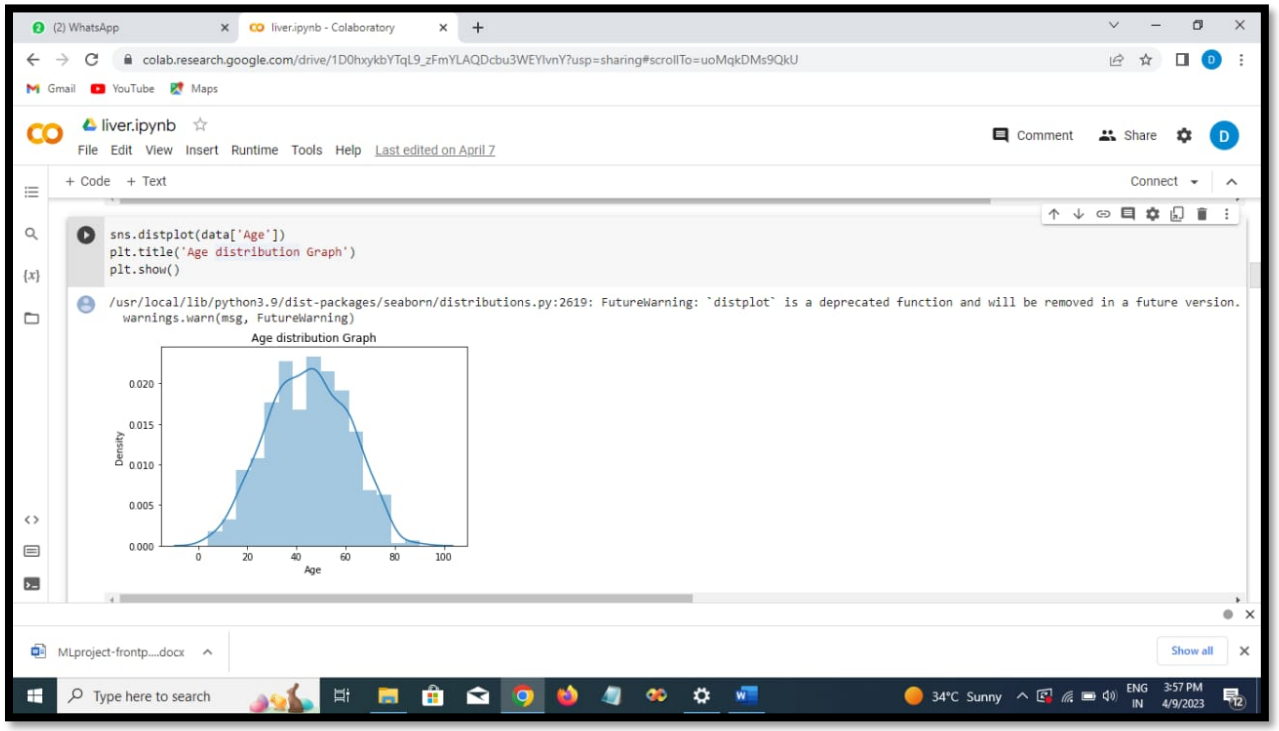

(1) WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=XjzxO6zew2xT
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
data.isnull().sum()
Age 0
Gender 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
[ ] #checking for the missing data after cleansing data(
data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].median(),inplace=True)
[ ] data.isnull().sum()
MLproject-frontp....docx
Show all
Type here to search
34°C Sunny
ENG IN
3:29 PM 4/9/2023


(1) WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=isHmgE7KblLa
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
#checking for the missing data after cleansing data(
data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].median(),inplace=True)
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 0
Dataset 0
dtype: int64
[ ] from sklearn.preprocessing import LabelEncoder
lc = LabelEncoder()
MLproject-frontp....docx
Show all
Type here to search
34°C Sunny
ENG
IN
3:29 PM
4/9/2023

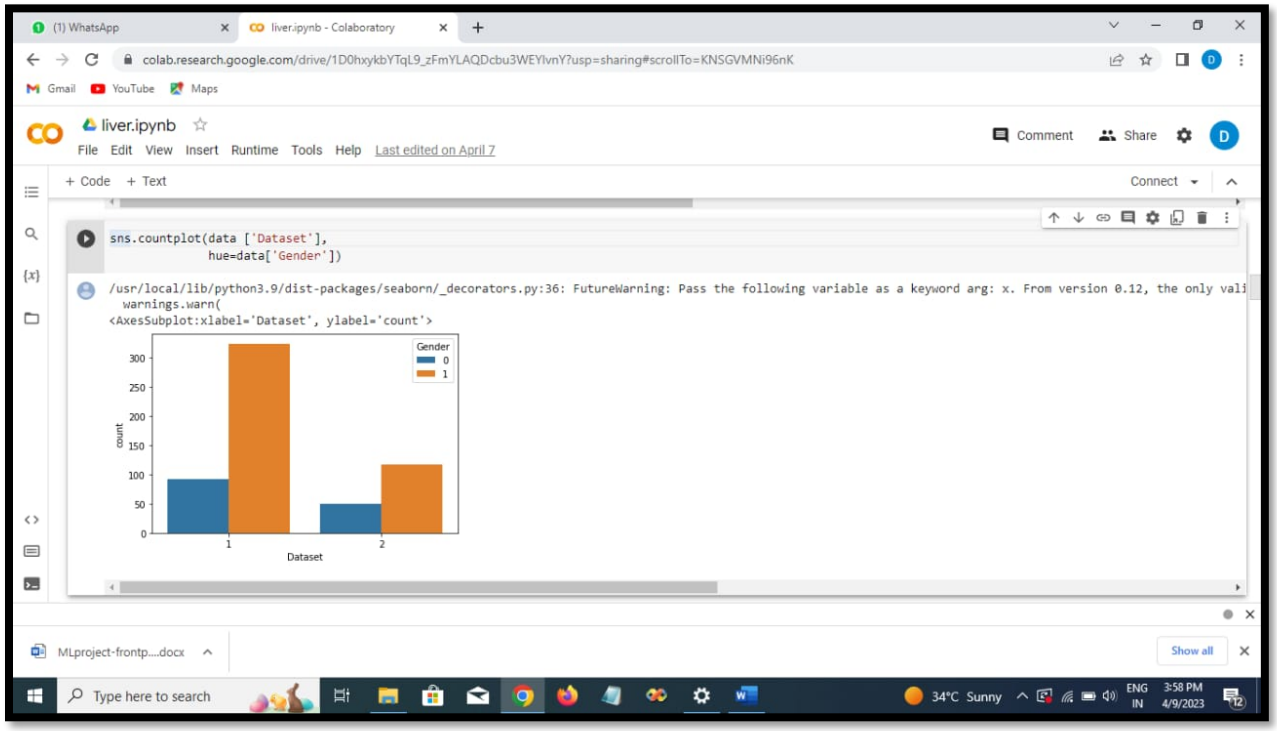
HANDLING CATEGORICAL VALUE



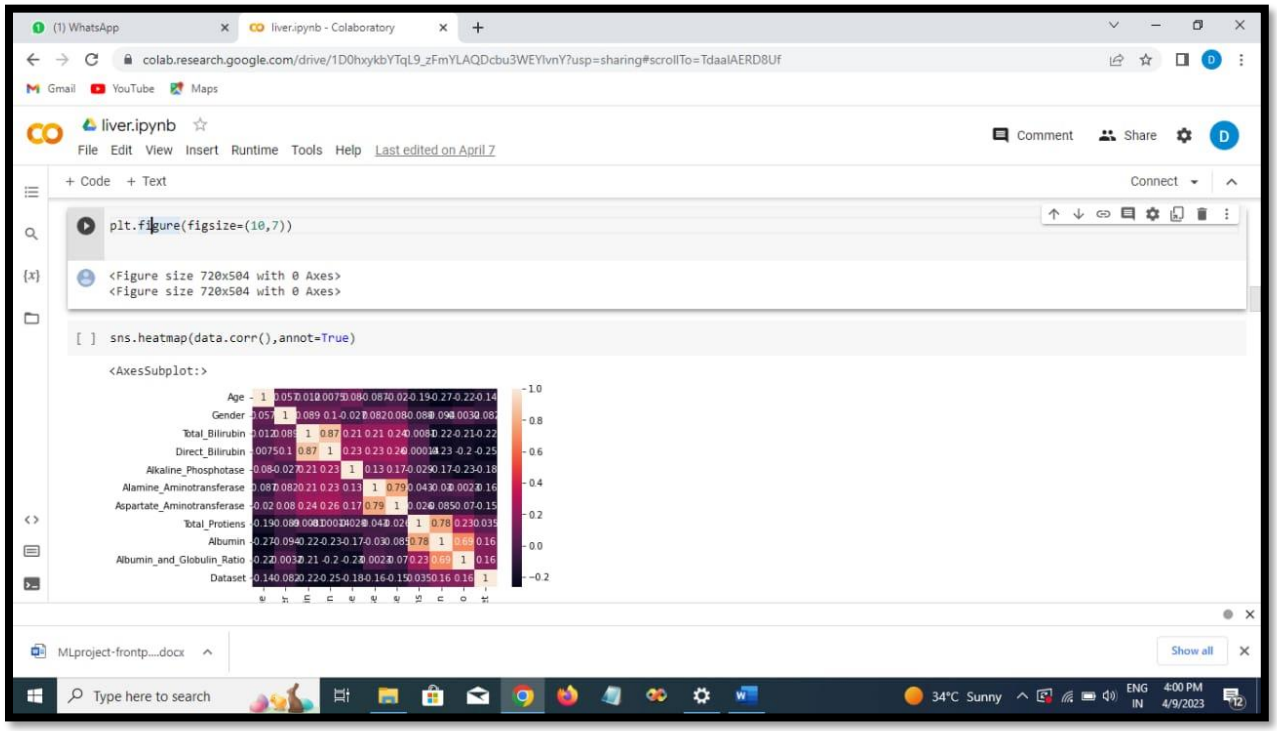
UNIVARIATE ANALYSIS



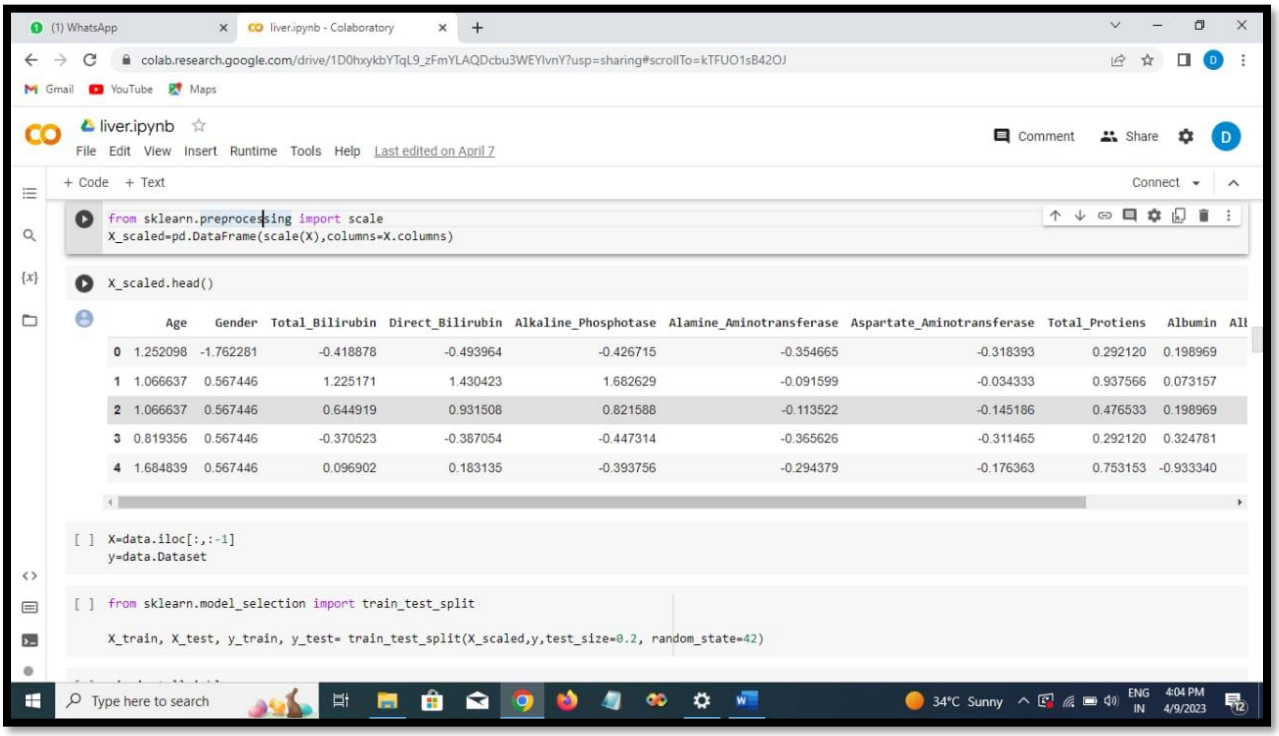
BIVARIATE ANAYSIS



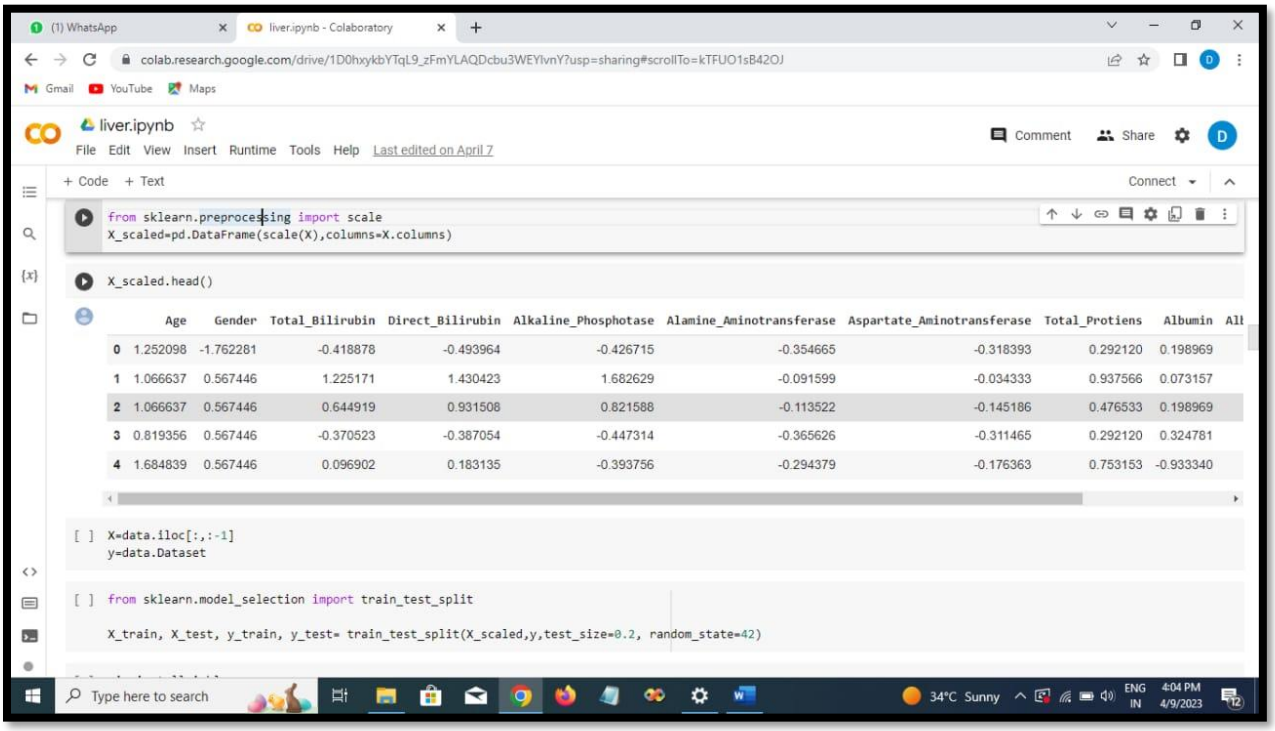
MULTIVARIATE ANALYSIS



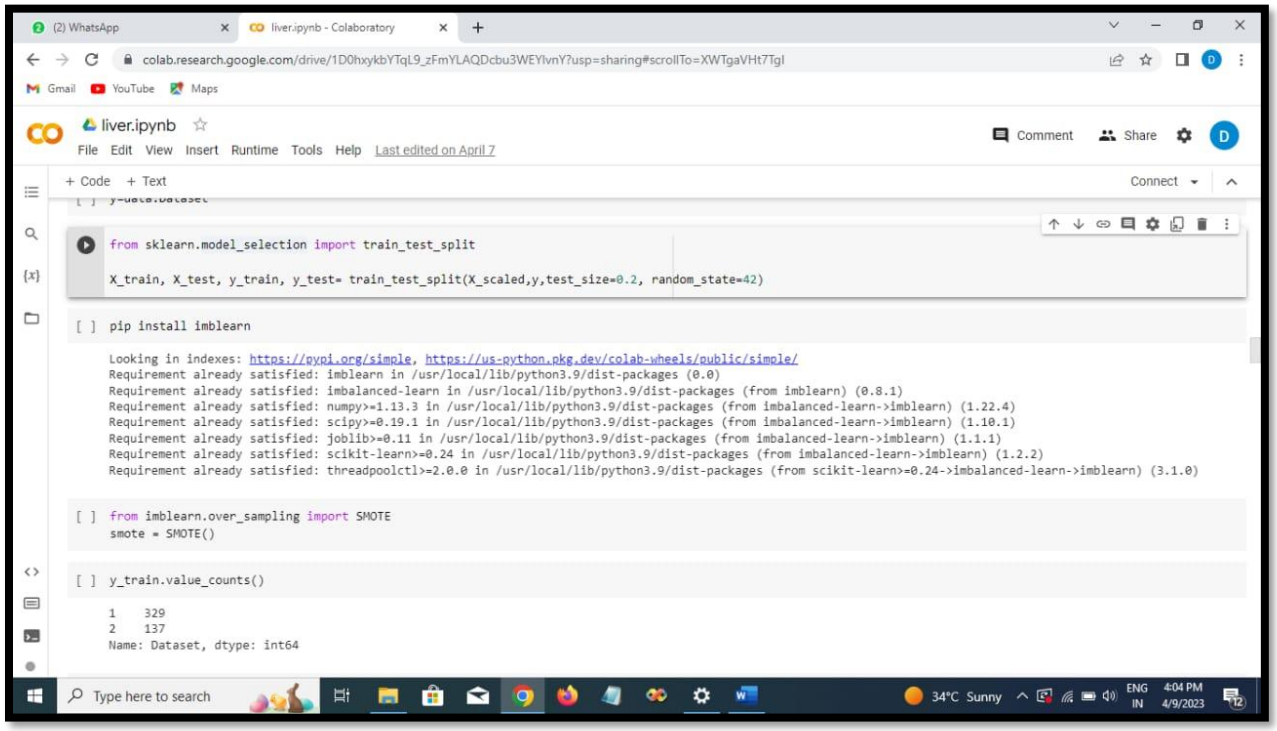
SCALING THE DATA

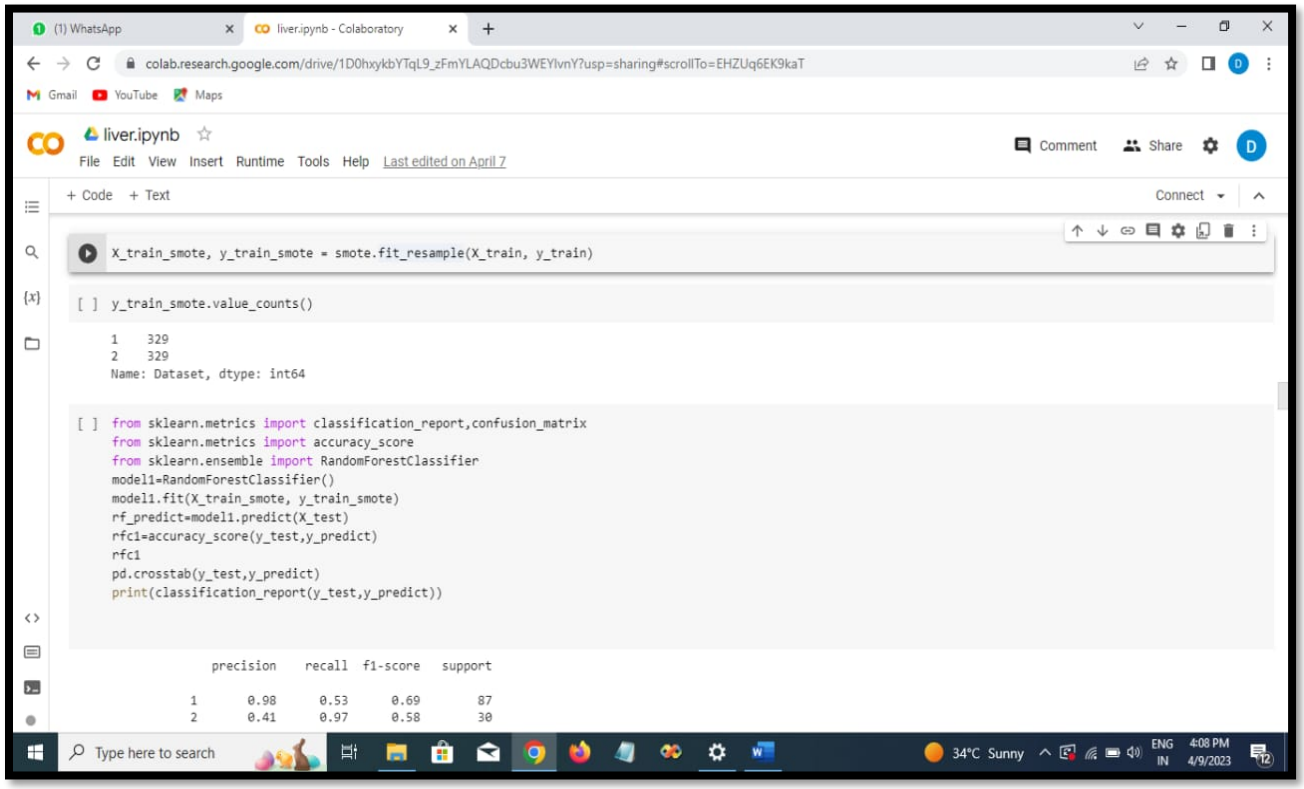


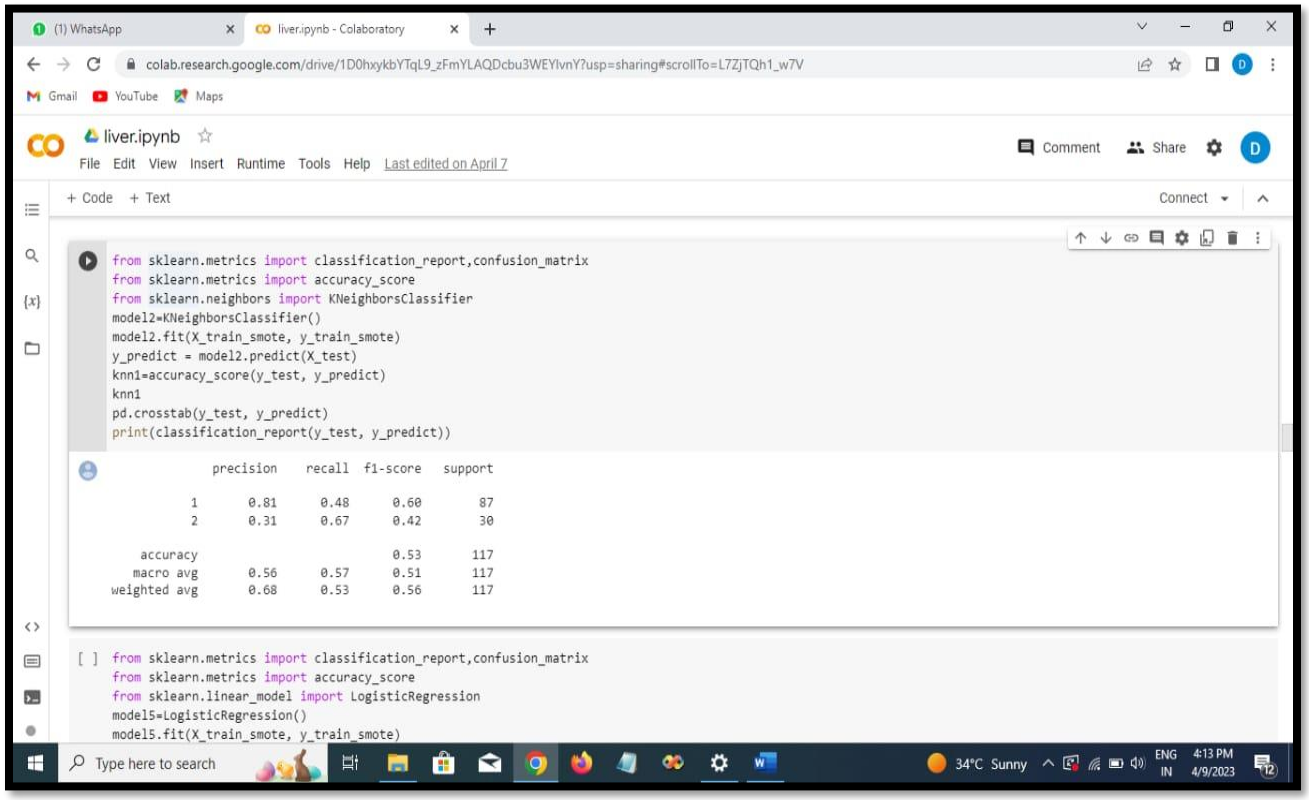
SPLITING A DATA INTO TRAIN AND TEST

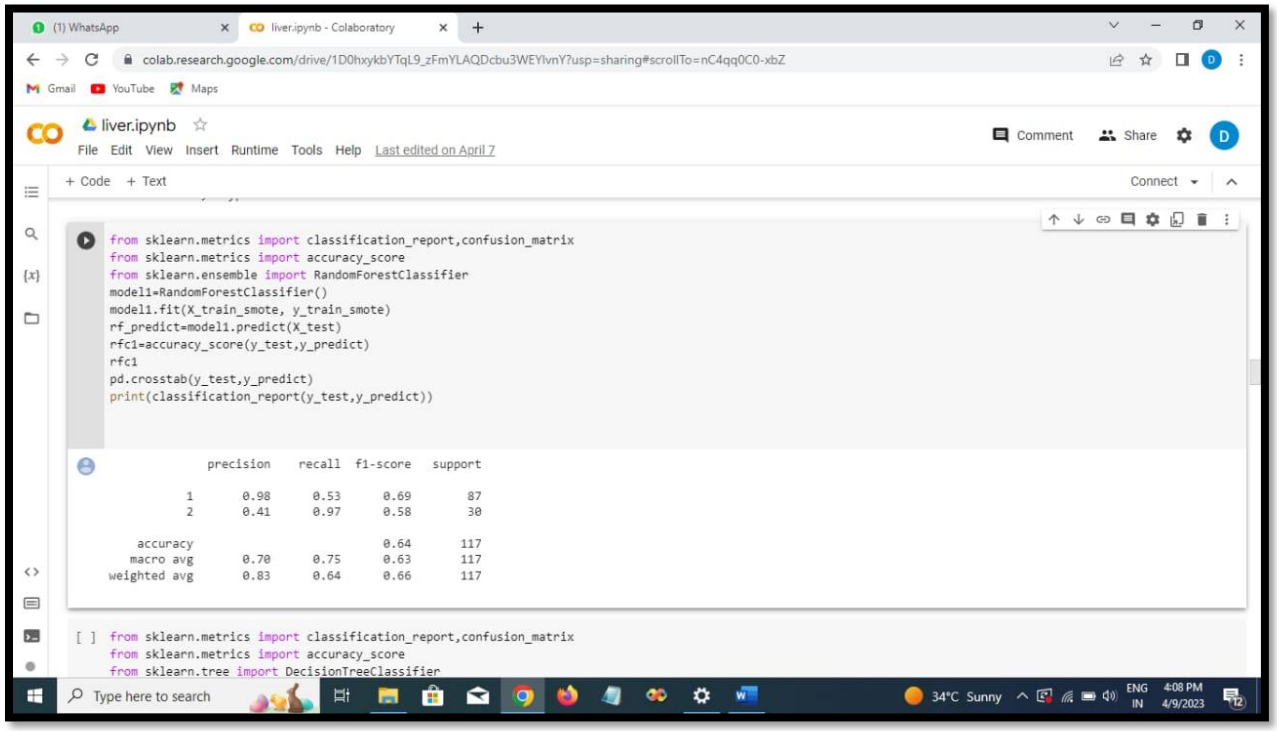


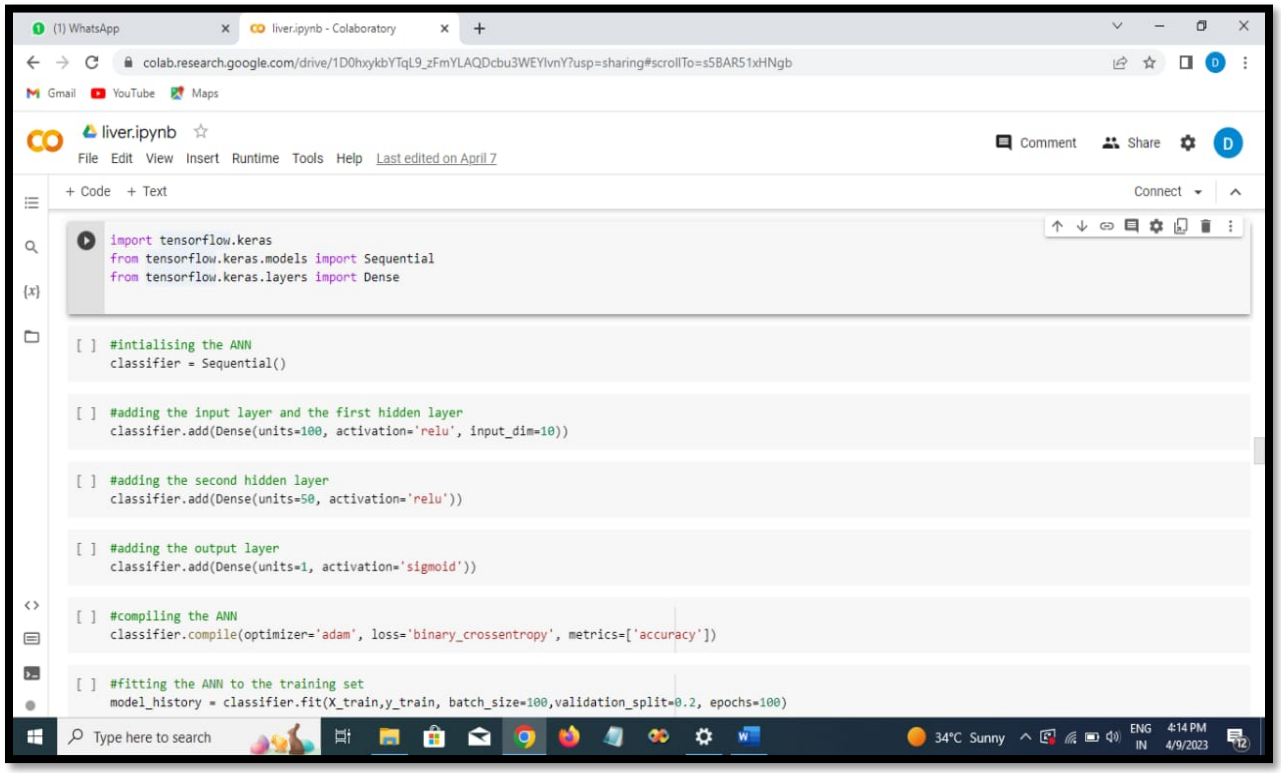
HANDLING IMBALANCE DATA

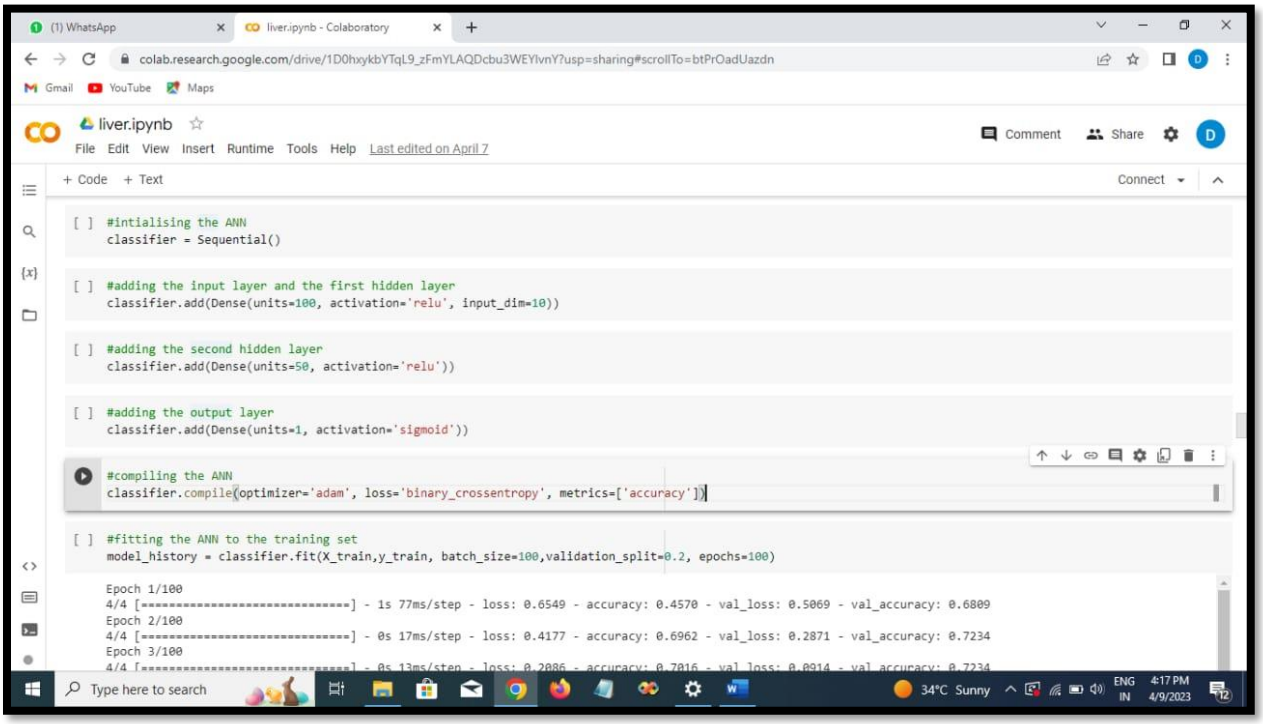


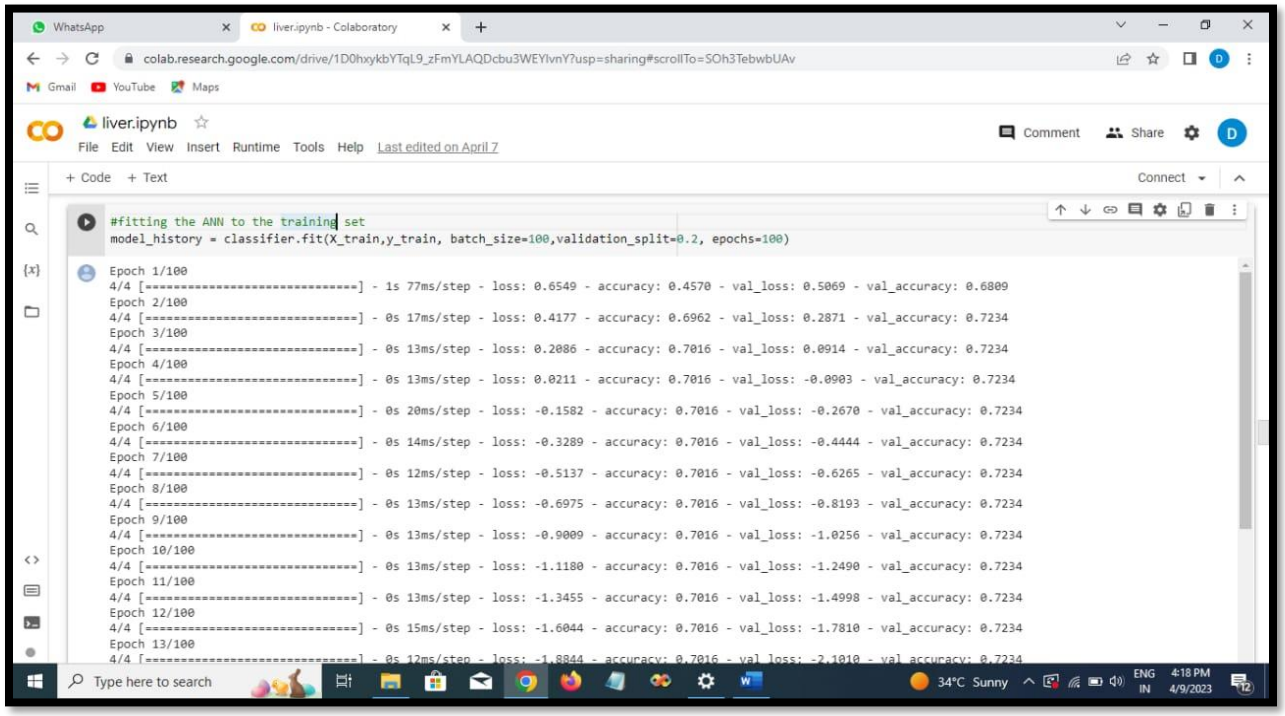

(1) WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYlvnY?usp=sharing#scrollTo=EHZUq6EK9kaT
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
[ ] y_train_smote.value_counts()
1 329
2 329
Name: Dataset, dtype: int64
[ ] from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fit(X_train_smote, y_train_smote)
rf_predict=model1.predict(X_test)
rfc1=accuracy_score(y_test,y_predict)
rfc1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test,y_predict))
precision recall f1-score support
1 0.98 0.53 0.69 87
2 0.41 0.97 0.58 30
Type here to search
34°C Sunny
ENG
IN
4:08 PM
4/9/2023

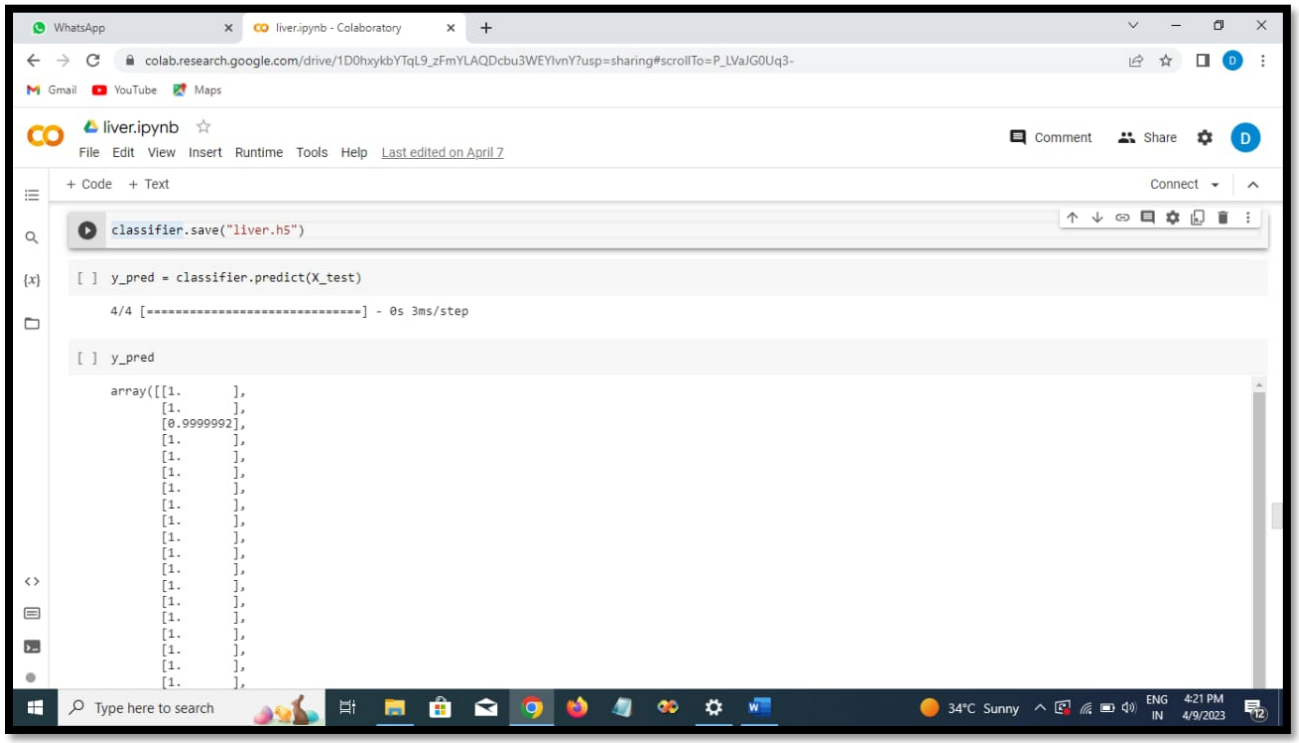

(1) WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=L7ZjTQh1_w7V
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
model2=KNeighborsClassifier()
model2.fit(X_train_smote, y_train_smote)
y_predict = model2.predict(X_test)
knn1=accuracy_score(y_test, y_predict)
knn1
pd.crosstab(y_test, y_predict)
print(classification_report(y_test, y_predict))
precision recall f1-score support
1 0.81 0.48 0.60 87
2 0.31 0.67 0.42 30
accuracy 0.53 117
macro avg 0.56 0.57 0.51 117
weighted avg 0.68 0.53 0.56 117
[ ] from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
model5=LogisticRegression()
model5.fit(X_train_smote, y_train_smote)
Type here to search
34°C Sunny
ENG IN
4:13 PM 4/9/2023

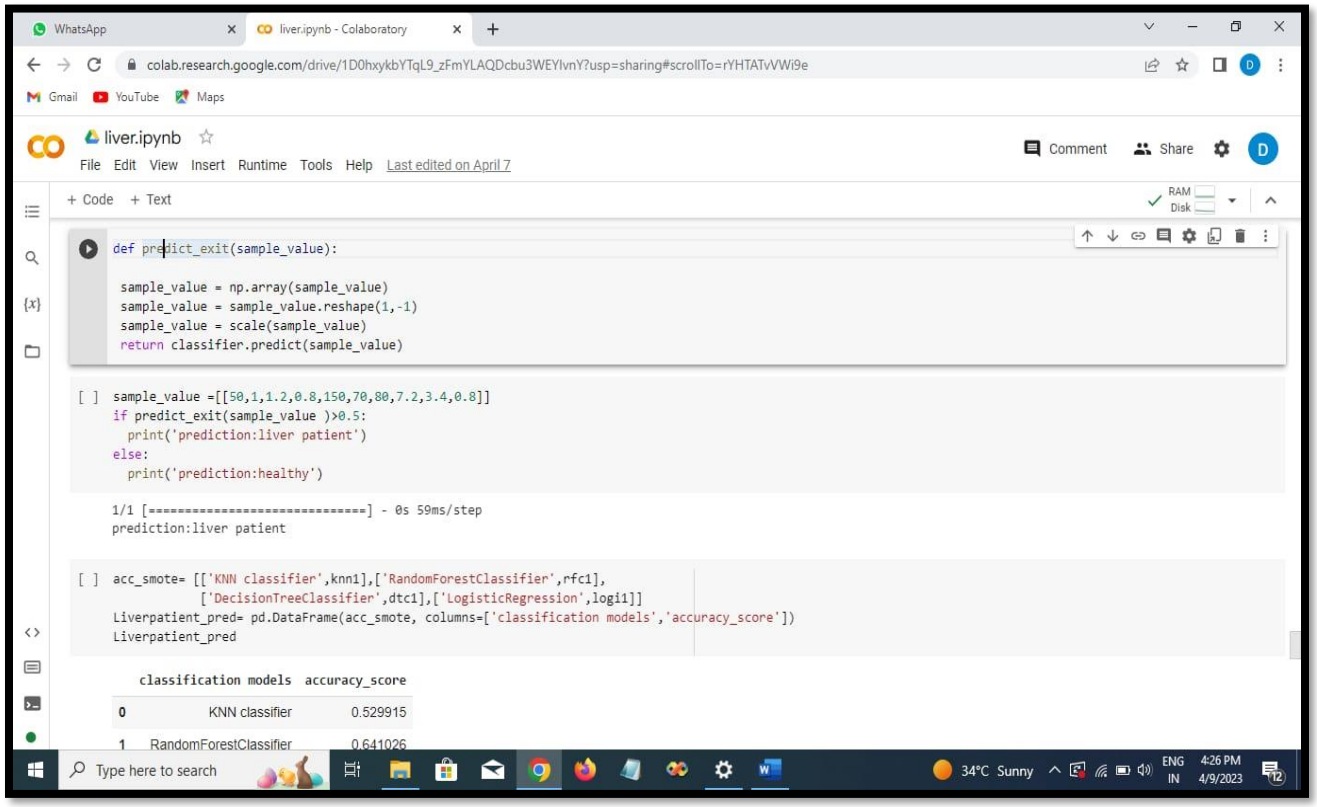

(1) WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYlvnY?usp=sharing#scrollTo=nC4qq0C0-xbZ
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fit(X_train_smote, y_train_smote)
rf_predict=model1.predict(X_test)
rfc1=accuracy_score(y_test,y_predict)
rfc1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test,y_predict))
precision recall f1-score support
1 0.98 0.53 0.69 87
2 0.41 0.97 0.58 30
accuracy 0.64 117
macro avg 0.70 0.75 0.63 117
weighted avg 0.83 0.64 0.66 117
[ ] from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
Type here to search
34°C Sunny
ENG IN
4:08 PM 4/9/2023

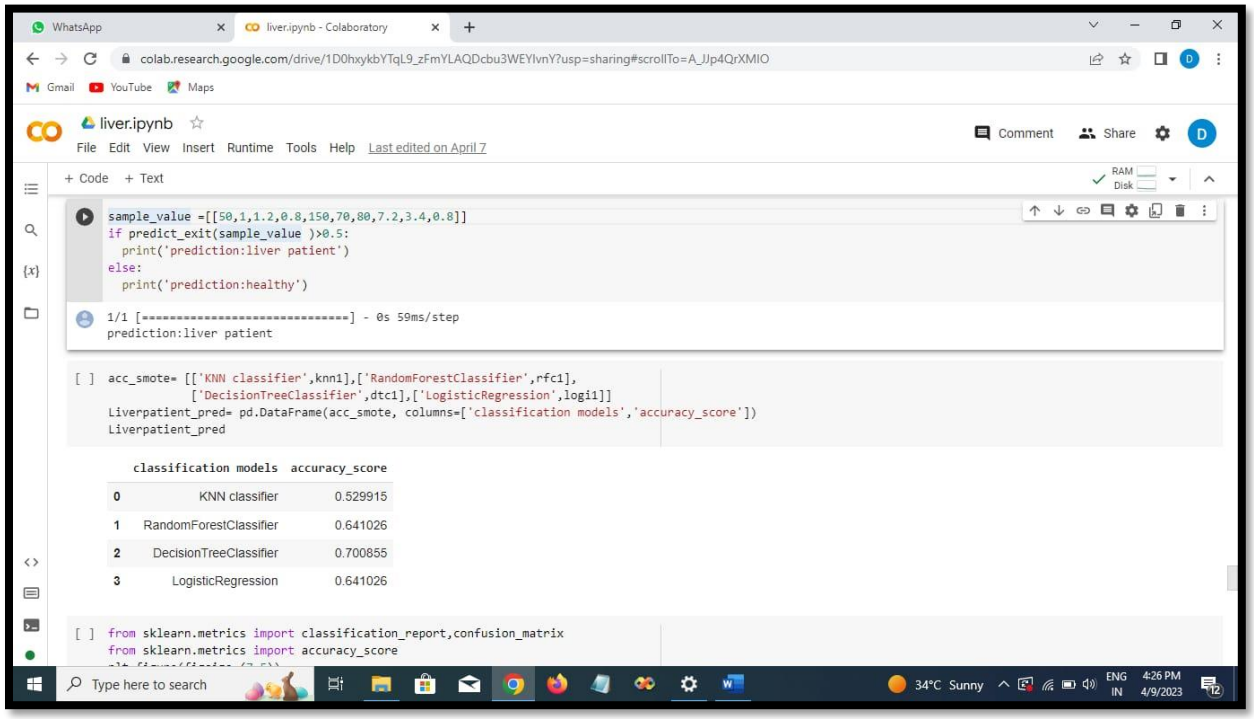

(1) WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=s5BAR51xHNgb
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
import tensorflow.keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
#intialising the ANN
classifier = Sequential()
#adding the input layer and the first hidden layer
classifier.add(Dense(units=100, activation='relu', input_dim=10))
#adding the second hidden layer
classifier.add(Dense(units=50, activation='relu'))
#adding the output layer
classifier.add(Dense(units=1, activation='sigmoid'))
#compiling the ANN
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
#fitting the ANN to the training set
model_history = classifier.fit(X_train,y_train, batch_size=100,validation_split=0.2, epochs=100)
Type here to search
34°C Sunny
ENG
IN
4:14 PM
4/9/2023

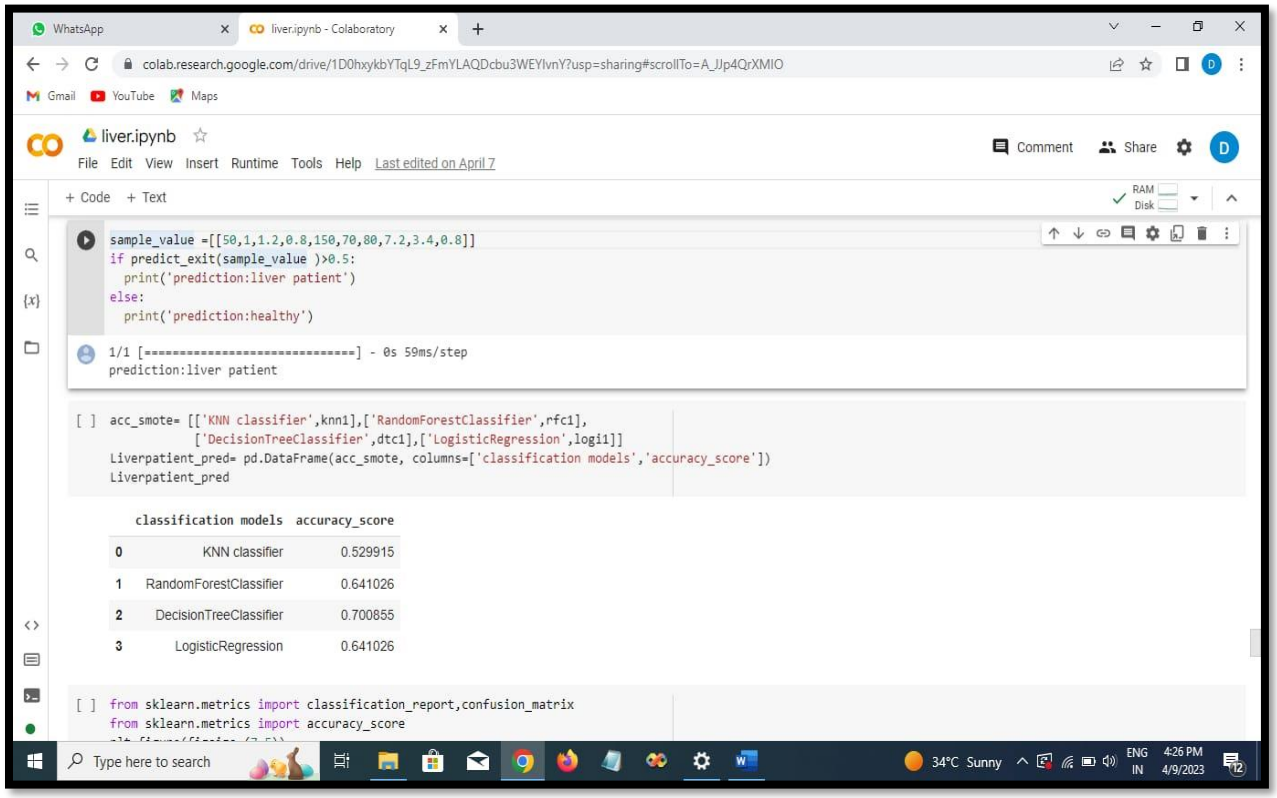

(1) WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYlvnY?usp=sharing#scrollTo=btPrOadUazdn
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
[ ] #intialising the ANN
classifier = Sequential()
[ ] #adding the input layer and the first hidden layer
classifier.add(Dense(units=100, activation='relu', input_dim=10))
[ ] #adding the second hidden layer
classifier.add(Dense(units=50, activation='relu'))
[ ] #adding the output layer
classifier.add(Dense(units=1, activation='sigmoid'))
#compiling the ANN
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
[ ] #fitting the ANN to the training set
model_history = classifier.fit(X_train,y_train, batch_size=100,validation_split=0.2, epochs=100)
Epoch 1/100
4/4 [==============================] - 1s 77ms/step - loss: 0.6549 - accuracy: 0.4570 - val_loss: 0.5069 - val_accuracy: 0.6809
Epoch 2/100
4/4 [==============================] - 0s 17ms/step - loss: 0.4177 - accuracy: 0.6962 - val_loss: 0.2871 - val_accuracy: 0.7234
Epoch 3/100
Type here to search
34°C Sunny
ENG
IN
4:17 PM
4/9/2023

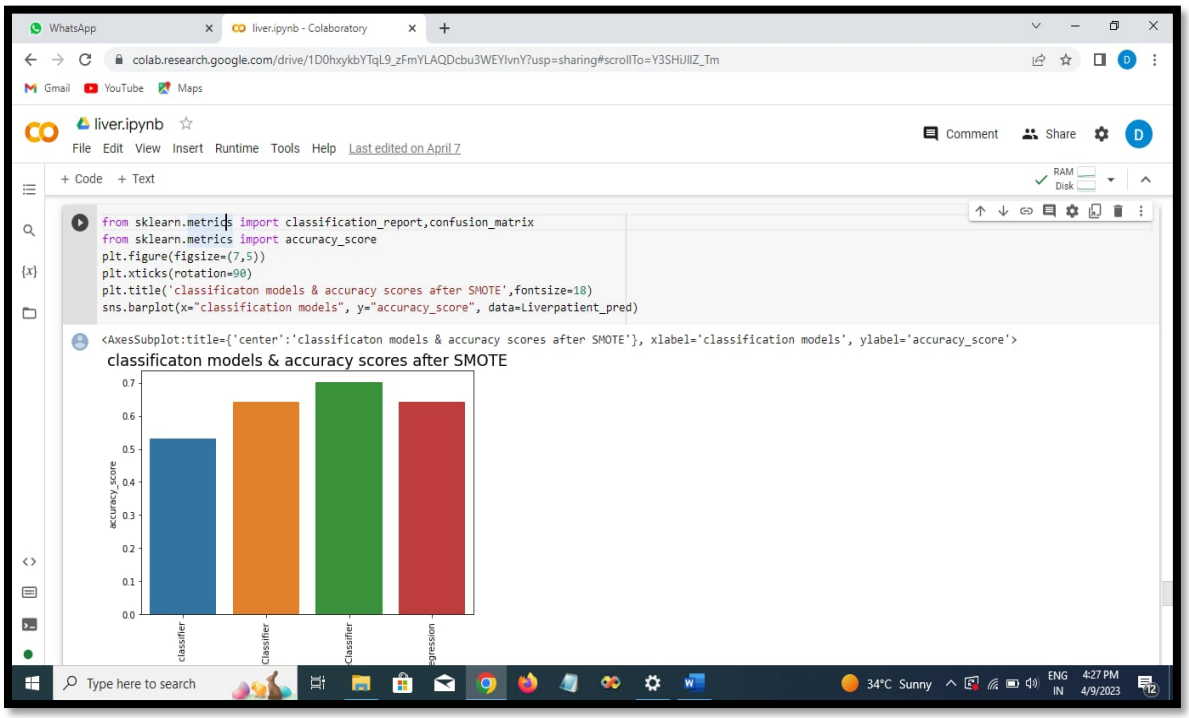

WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=SOh3TebwbUAv
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
#fitting the ANN to the training set
model_history = classifier.fit(X_train,y_train, batch_size=100,validation_split=0.2, epochs=100)
Epoch 1/100
4/4 [==============================] - 1s 77ms/step - loss: 0.6549 - accuracy: 0.4570 - val_loss: 0.5069 - val_accuracy: 0.6809
Epoch 2/100
4/4 [==============================] - 0s 17ms/step - loss: 0.4177 - accuracy: 0.6962 - val_loss: 0.2871 - val_accuracy: 0.7234
Epoch 3/100
4/4 [==============================] - 0s 13ms/step - loss: 0.2086 - accuracy: 0.7016 - val_loss: 0.0914 - val_accuracy: 0.7234
Epoch 4/100
4/4 [==============================] - 0s 13ms/step - loss: 0.0211 - accuracy: 0.7016 - val_loss: -0.0903 - val_accuracy: 0.7234
Epoch 5/100
4/4 [==============================] - 0s 20ms/step - loss: -0.1582 - accuracy: 0.7016 - val_loss: -0.2670 - val_accuracy: 0.7234
Epoch 6/100
4/4 [==============================] - 0s 14ms/step - loss: -0.3289 - accuracy: 0.7016 - val_loss: -0.4444 - val_accuracy: 0.7234
Epoch 7/100
4/4 [==============================] - 0s 12ms/step - loss: -0.5137 - accuracy: 0.7016 - val_loss: -0.6265 - val_accuracy: 0.7234
Epoch 8/100
4/4 [==============================] - 0s 13ms/step - loss: -0.6975 - accuracy: 0.7016 - val_loss: -0.8193 - val_accuracy: 0.7234
Epoch 9/100
4/4 [==============================] - 0s 13ms/step - loss: -0.9009 - accuracy: 0.7016 - val_loss: -1.0256 - val_accuracy: 0.7234
Epoch 10/100
4/4 [==============================] - 0s 13ms/step - loss: -1.1180 - accuracy: 0.7016 - val_loss: -1.2490 - val_accuracy: 0.7234
Epoch 11/100
4/4 [==============================] - 0s 13ms/step - loss: -1.3455 - accuracy: 0.7016 - val_loss: -1.4998 - val_accuracy: 0.7234
Epoch 12/100
4/4 [==============================] - 0s 15ms/step - loss: -1.6044 - accuracy: 0.7016 - val_loss: -1.7810 - val_accuracy: 0.7234
Epoch 13/100
Type here to search
34°C Sunny
ENG IN
4:18 PM 4/9/2023

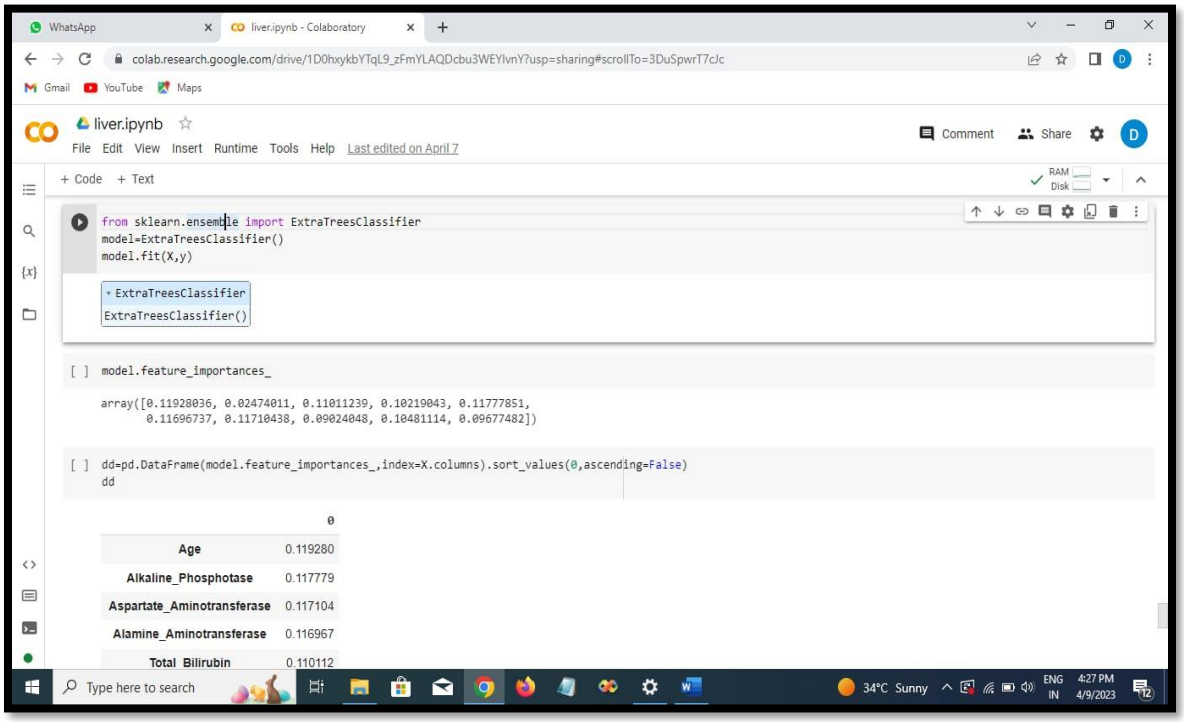

WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYlvnY?usp=sharing#scrollTo=P_LVaJG0Uq3-
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
Connect
classifier.save("liver.h5")
[ ] y_pred = classifier.predict(X_test)
4/4 [==============================] - 0s 3ms/step
[ ] y_pred
array([[1.
[1.
[0.9999992],
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
[1.
Type here to search
34°C Sunny
ENG
IN
4:21 PM
4/9/2023

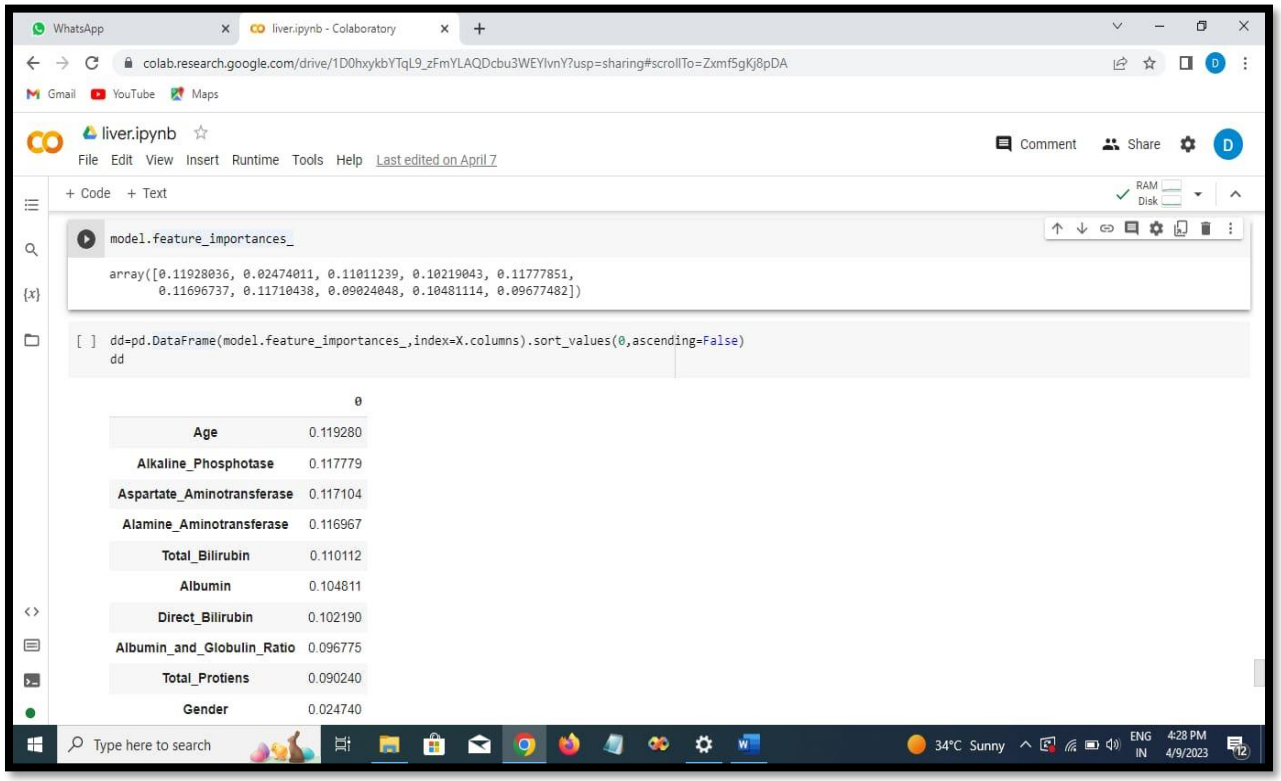

liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=rYHTATvVWi9e
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
RAM
Disk
def predict_exit(sample_value):
sample_value = np.array(sample_value)
sample_value = sample_value.reshape(1,-1)
sample_value = scale(sample_value)
return classifier.predict(sample_value)
sample_value =[[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]
if predict_exit(sample_value)>0.5:
print('prediction:liver patient')
else:
print('prediction:healthy')
1/1 [==============================] - 0s 59ms/step
prediction:liver patient
acc_smote= [['KNN classifier',knn1],['RandomForestClassifier',rfc1],
['DecisionTreeClassifier',dtc1],['LogisticRegression',log1]]
Liverpatient_pred= pd.DataFrame(acc_smote, columns=['classification models','accuracy_score'])
Liverpatient_pred
classification models accuracy_score
0 KNN classifier 0.529915
1 RandomForestClassifier 0.641026
Type here to search
34°C Sunny
ENG
IN
4:26 PM
4/9/2023

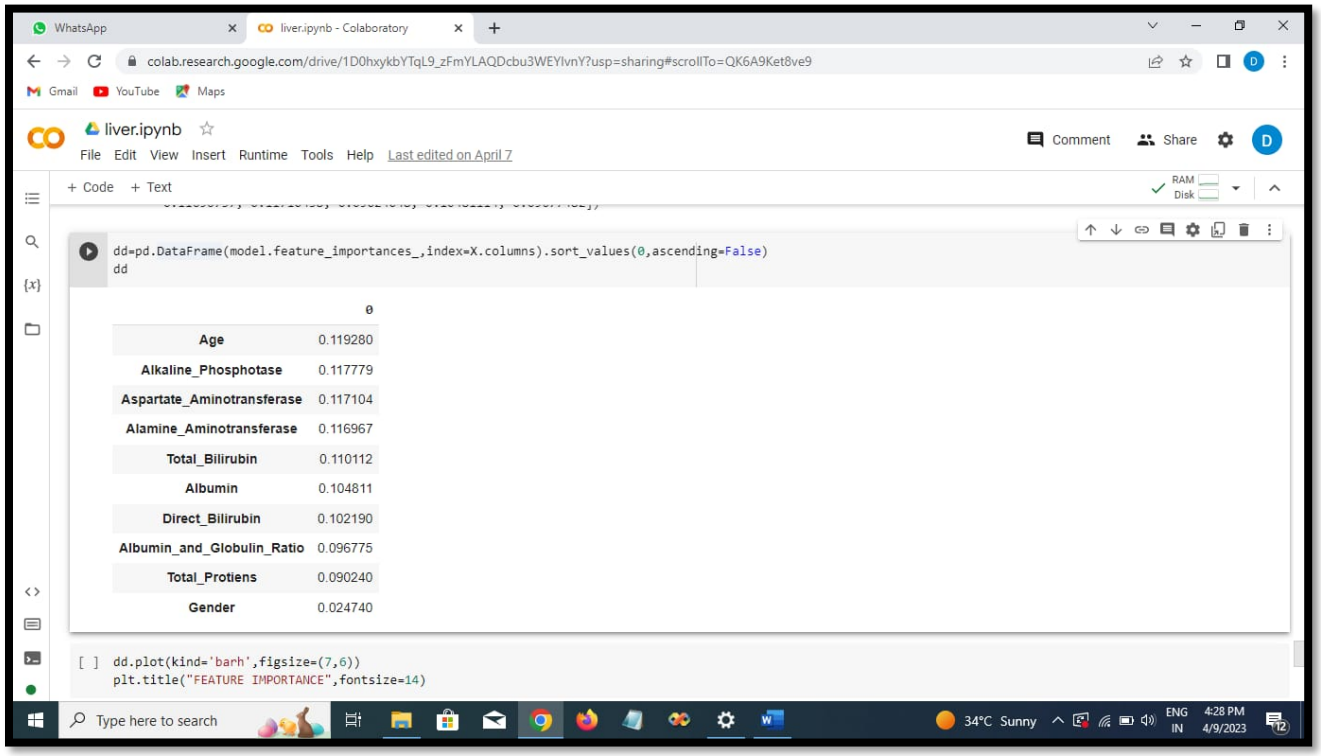

WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=A_JJp4QrXMIO
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
RAM
Disk
sample_value =[[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]
if predict_exit(sample_value)>0.5:
print('prediction:liver patient')
else:
print('prediction:healthy')
1/1 [==============================] - 0s 59ms/step
prediction:liver patient
[ ] acc_smote= [['KNN classifier',knn1],['RandomForestClassifier',rfc1],
['DecisionTreeClassifier',dtc1],['LogisticRegression',logi1]]
Liverpatient_pred= pd.DataFrame(acc_smote, columns=['classification models','accuracy_score'])
Liverpatient_pred
classification models accuracy_score
0 KNN classifier 0.529915
1 RandomForestClassifier 0.641026
2 DecisionTreeClassifier 0.700855
3 LogisticRegression 0.641026
[ ] from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
Type here to search
34°C Sunny
ENG
IN
4:26 PM
4/9/2023

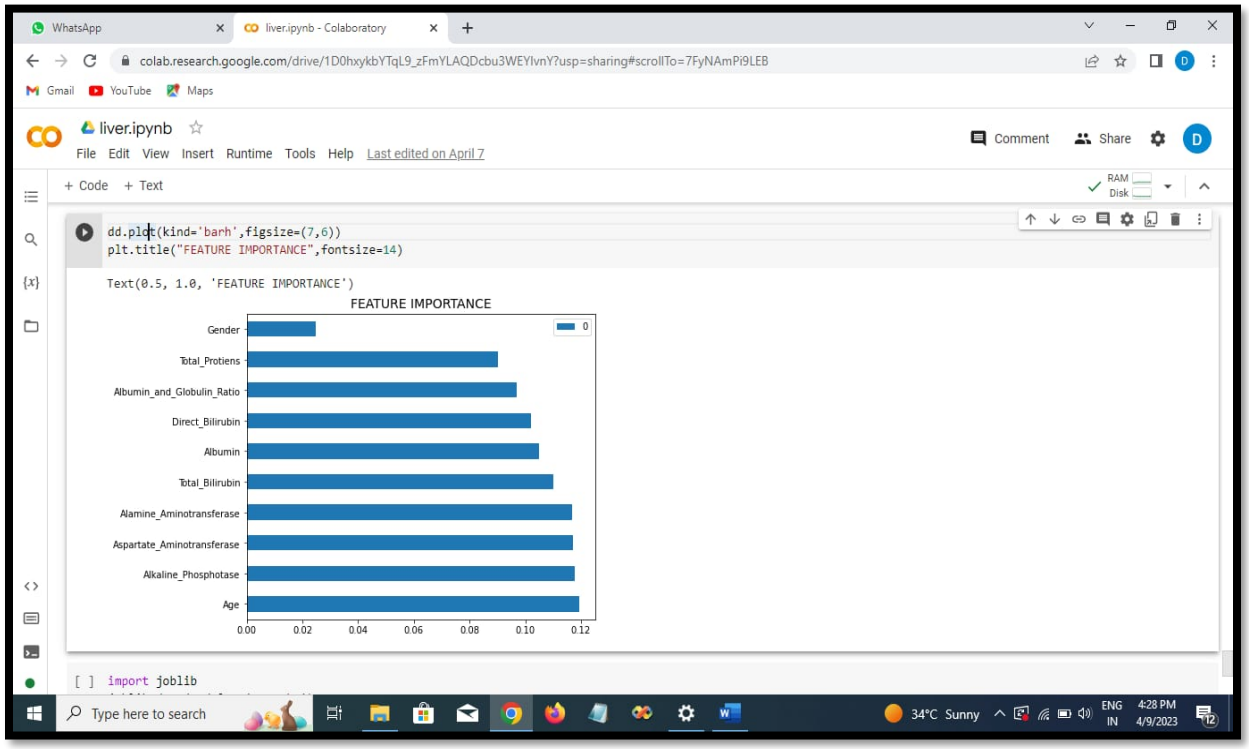

WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=A_JJp4QrXMIO
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
RAM
Disk
sample_value =[[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]
if predict_exit(sample_value)>0.5:
print('prediction:liver patient')
else:
print('prediction:healthy')
1/1 [==============================] - 0s 59ms/step
prediction:liver patient
[ ] acc_smote= [['KNN classifier',knn1],['RandomForestClassifier',rfc1],
['DecisionTreeClassifier',dtc1],['LogisticRegression',logi1]]
Liverpatient_pred= pd.DataFrame(acc_smote, columns=['classification models','accuracy_score'])
Liverpatient_pred
classification models accuracy_score
0 KNN classifier 0.529915
1 RandomForestClassifier 0.641026
2 DecisionTreeClassifier 0.700855
3 LogisticRegression 0.641026
[ ] from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
Type here to search
34°C Sunny
ENG
IN
4:26 PM
4/9/2023

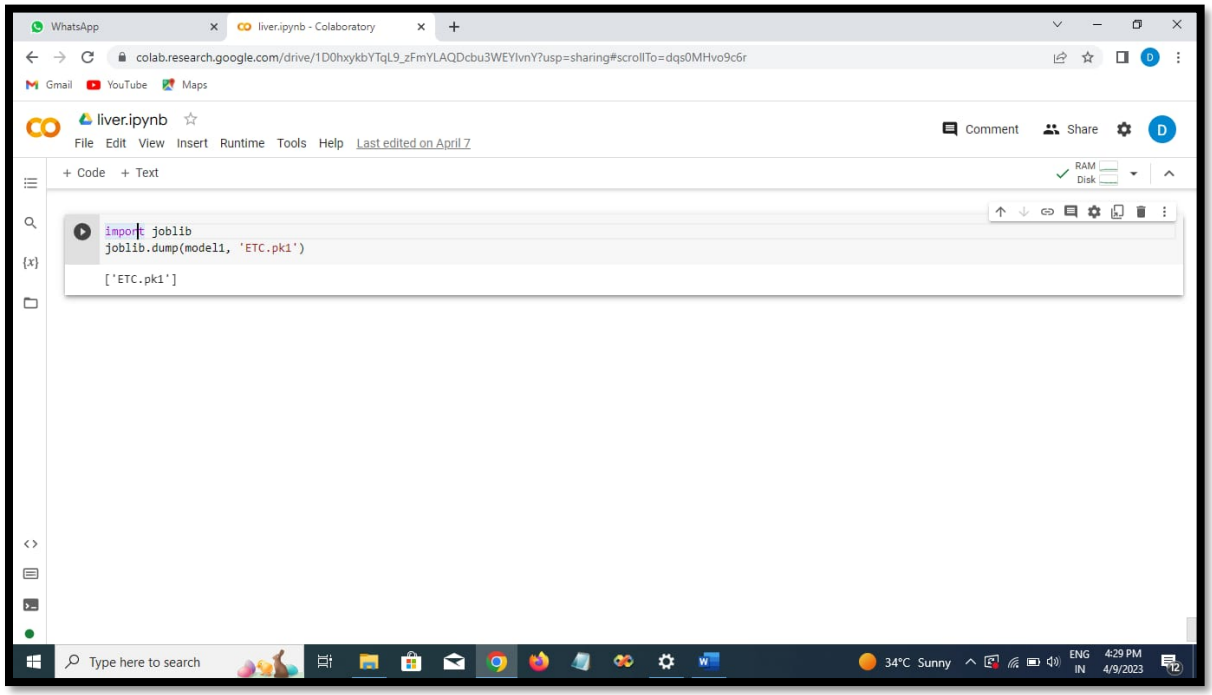

liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=Y3SHiJIIZ_Tm
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
plt.figure(figsize=(7,5))
plt.xticks(rotation=90)
plt.title('classificaton models & accuracy scores after SMOTE',fontsize=18)
sns.barplot(x="classification models", y="accuracy_score", data=Liverpatient_pred)
<AxesSubplot:title={'center':'classificaton models & accuracy scores after SMOTE'}, xlabel='classification models', ylabel='accuracy_score'>
classificaton models & accuracy scores after SMOTE
accuracy_score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Classifier
Classifier
Classifier
gression
Type here to search
34°C Sunny
ENG
IN
4:27 PM
4/9/2023


WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=3DuSpwrT7cJc
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
RAM
Disk
from sklearn.ensemble import ExtraTreesClassifier
model=ExtraTreesClassifier()
model.fit(X,y)
ExtraTreesClassifier
ExtraTreesClassifier()
[ ] model.feature_importances_
array([0.11928036, 0.02474011, 0.11011239, 0.10219043, 0.11777851,
0.11696737, 0.11710438, 0.09024048, 0.10481114, 0.09677482])
[ ] dd=pd.DataFrame(model.feature_importances_,index=X.columns).sort_values(0,ascending=False)
dd
0
Age 0.119280
Alkaline_Phosphotase 0.117779
Aspartate_Aminotransferase 0.117104
Alamine_Aminotransferase 0.116967
Total_Bilirubin 0.110112
Type here to search
34°C Sunny
ENG
IN
4:27 PM
4/9/2023

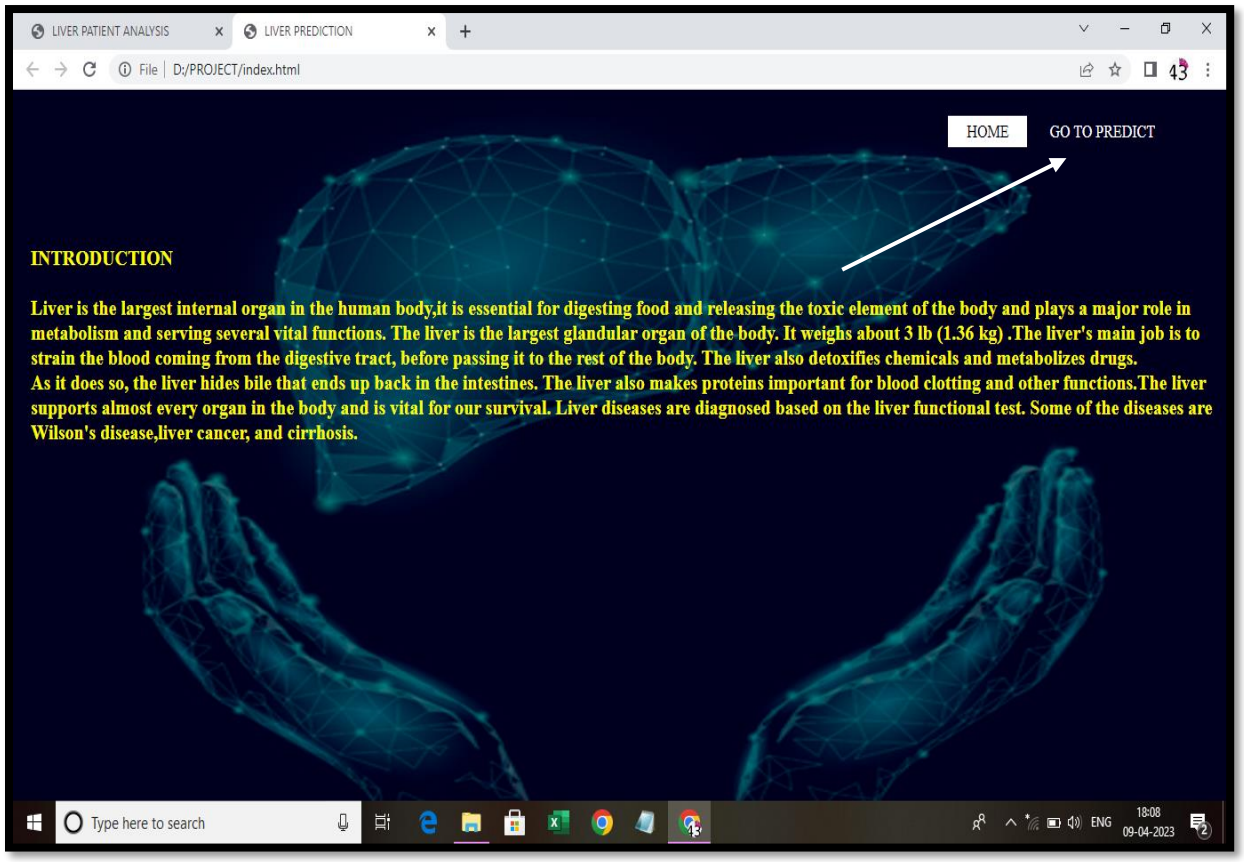

WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=Zxmf5gKj8pDA
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
RAM
Disk
model.feature_importances_
array([0.11928036, 0.02474011, 0.11011239, 0.10219043, 0.11777851,
0.11696737, 0.11710438, 0.09024048, 0.10481114, 0.09677482])
[ ] dd=pd.DataFrame(model.feature_importances_,index=X.columns).sort_values(0,ascending=False)
dd
0
Age 0.119280
Alkaline_Phosphotase 0.117779
Aspartate_Aminotransferase 0.117104
Alamine_Aminotransferase 0.116967
Total_Bilirubin 0.110112
Albumin 0.104811
Direct_Bilirubin 0.102190
Albumin_and_Globulin_Ratio 0.096775
Total_Protiens 0.090240
Gender 0.024740
Type here to search
34°C Sunny
ENG
IN
4:28 PM
4/9/2023

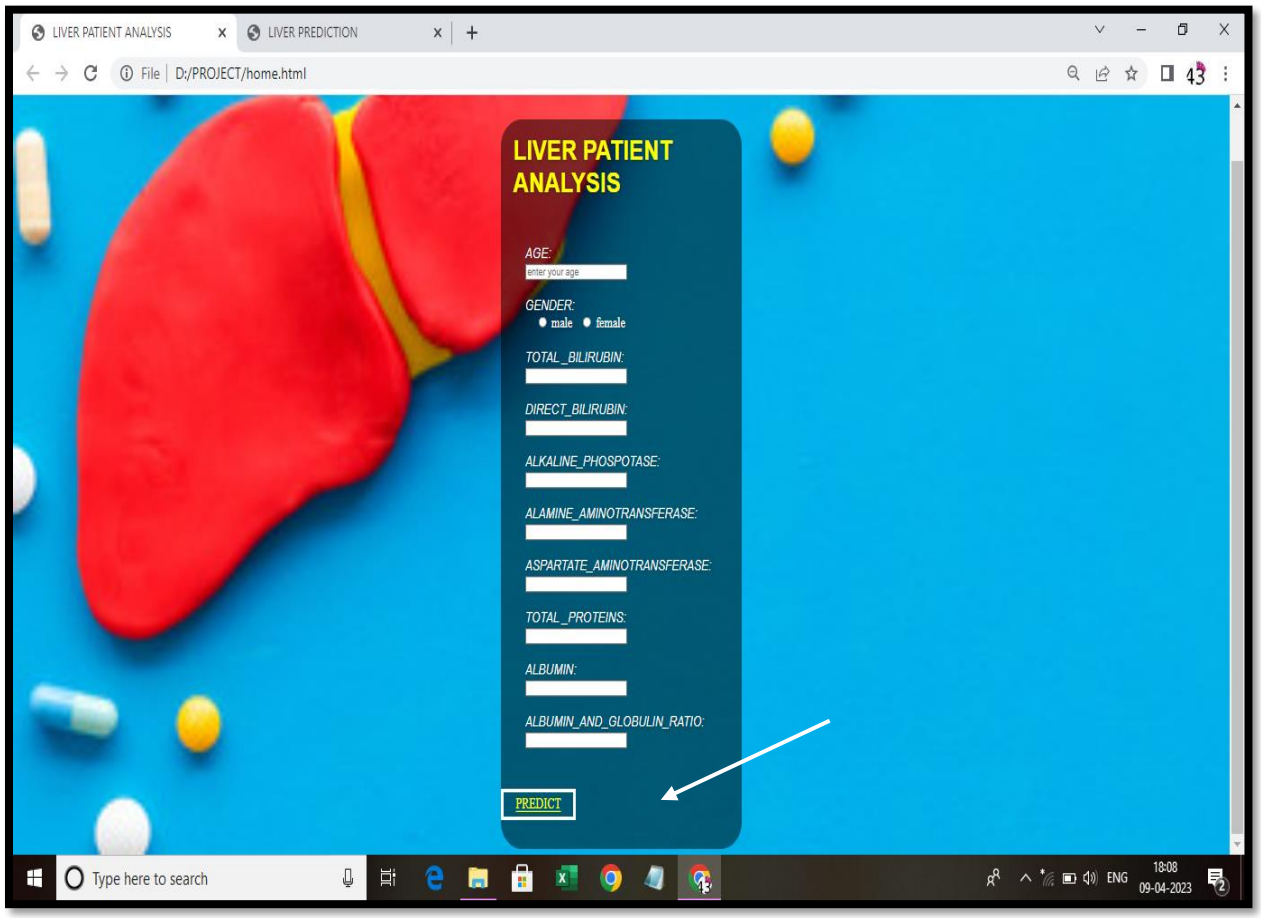

WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYlvnY?usp=sharing#scrollTo=QK6A9Ket8ve9
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
RAM
Disk
dd=pd.DataFrame(model.feature_importances_,index=X.columns).sort_values(0,ascending=False)
dd
0
Age 0.119280
Alkaline_Phosphotase 0.117779
Aspartate_Aminotransferase 0.117104
Alamine_Aminotransferase 0.116967
Total_Bilirubin 0.110112
Albumin 0.104811
Direct_Bilirubin 0.102190
Albumin_and_Globulin_Ratio 0.096775
Total_Protiens 0.090240
Gender 0.024740
[ ] dd.plot(kind='barh',figsize=(7,6))
plt.title("FEATURE IMPORTANCE",fontsize=14)
Type here to search
34°C Sunny
ENG
IN
4:28 PM
4/9/2023

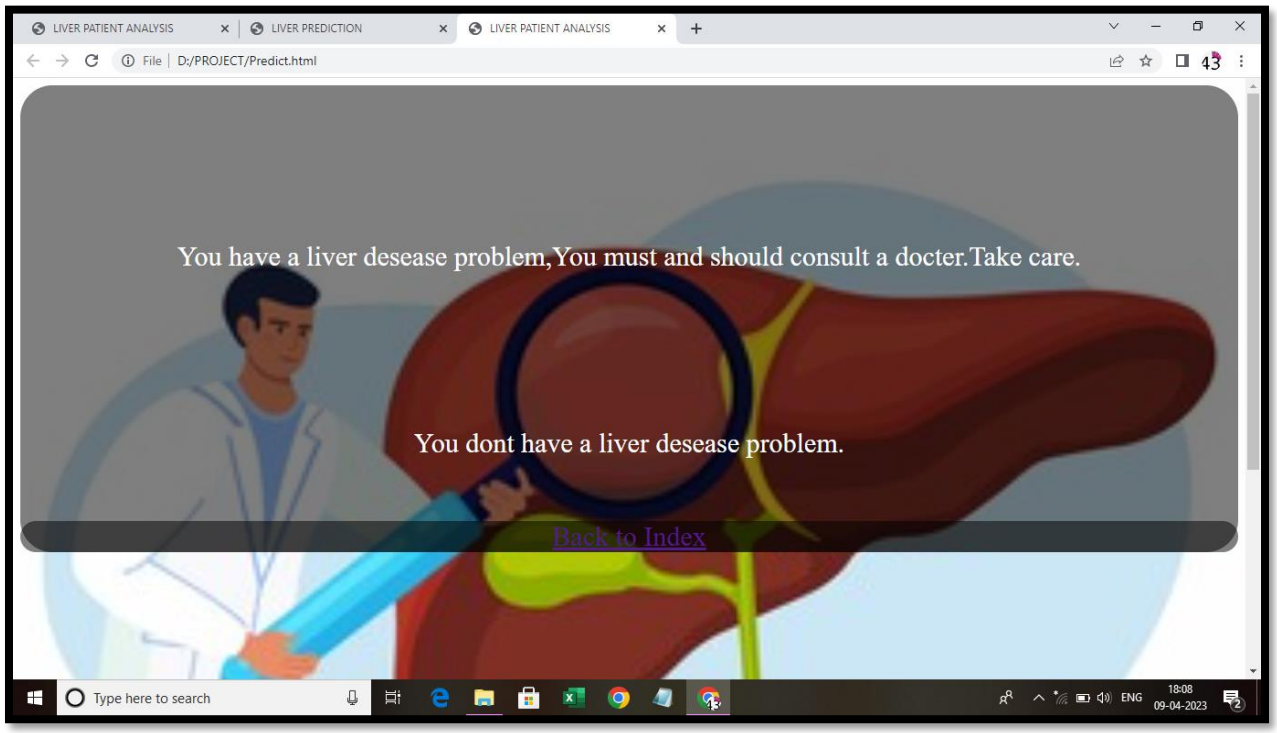

WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=7FyNAmPi9LEB
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
RAM
Disk
dd.plot(kind='barh',figsize=(7,6))
plt.title("FEATURE IMPORTANCE",fontsize=14)
Text(0.5, 1.0, 'FEATURE IMPORTANCE')
FEATURE IMPORTANCE
0
Gender
Total_Protiens
Albumin_and_Globulin_Ratio
Direct_Bilirubin
Albumin
Total_Bilirubin
Alamine_Aminotransferase
Aspartate_Aminotransferase
Alkaline_Phosphotase
Age
0.00
0.02
0.04
0.06
0.08
0.10
0.12
[ ] import joblib
Type here to search
34°C Sunny
ENG
IN
4:28 PM
4/9/2023


WhatsApp
liver.ipynb - Colaboratory
colab.research.google.com/drive/1D0hxykbYTqL9_zFmYLAQDcbu3WEYIvnY?usp=sharing#scrollTo=dqs0MHvo9c6r
Gmail
YouTube
Maps
liver.ipynb
File Edit View Insert Runtime Tools Help Last edited on April 7
Comment
Share
+ Code + Text
RAM
Disk
import joblib
joblib.dump(model1, 'ETC.pk1')
['ETC.pk1']
Type here to search
34°C Sunny
ENG
IN
4:29 PM
4/9/2023

LIVER DISEASE PREDICTION




LIVER PATIENT ANALYSIS
LIVER PREDICTION
File | D:/PROJECT/home.html
LIVER PATIENT ANALYSIS
AGE:
enter your age
GENDER:
male
female
TOTAL_BILIRUBIN:
DIRECT_BILIRUBIN:
ALKALINE_PHOSPOTASE:
ALAMINE_AMINOTRANSFERASE:
ASPARTATE_AMINOTRANSFERASE:
TOTAL_PROTEINS:
ALBUMIN:
ALBUMIN_AND_GLOBULIN_RATIO:
PREDICT
Type here to search
ENG
18:08
09-04-2023


LIVER PATIENT ANALYSIS
LIVER PREDICTION
LIVER PATIENT ANALYSIS
File | D:/PROJECT/Predict.html
You have a liver desease problem,You must and should consult a docter.Take care.
You dont have a liver desease problem.
Back to Index
Type here to search
ENG
18:08
09-04-2023