A Review of Liver Patient Analysis Methods Using Machine Learning

Done By

1.Alagananthi .E

2.Ayisha Shirin .S

3.Joselin Sabitha .S

4.Muthu Bhuvaneswari .C

5.Muthu Selvi Karthika .D

1.INTRODUCTION:

1.1 Overview

The liver is one of the most critical organs of the human body. It plays an essential role in the body's function. Primary purposes include removing toxins from the body, fighting against infections, and balancing the hormones and secretion of bile juice (Devikanniga et al., 2020). If these functions are not performed by the liver correctly, it will result in several complications and liver diseases. Therefore ifa virus infects the liver or chemicals that injure the liver are consumed, or the immune system's dysfunction occurs, severe damage to the liver or malfunctioning may happen, which ultimately might cause death (Nahar & Ara, 2018). Liver disease is one of the most chronic and threatening diseases globally that can cause various side effects if not treated early (Dutta et al., 2022). According to World Health Organization (WHO) report in 2018, the number of deaths due to liver diseases is around one million and ranked 11th in the world with a critical number of fatalities (World Total Deaths, n.d.). As the symptoms of liver diseases cannot be visible until the condition becomes chronic, it is challenging and daunting for medical health professionals to identify liver disease at its early stages (Devikannigaet al., 2020).

In addition, the traditional testing methods like sonography, MRI scans and CT scans that are available for detecting liver diseases are expensive and harmful with numerous side effects (Joloudari et al., 2019). Thus, a significant constraint found by health care workers is to predict liver diseases at an early stage, at minimal cost and at the same time provide a better health care system to treat liver diseases. Severe liver diseases include problems with indigestion, dry mouth, pain in the abdomen, skin colour turning yellow, numbness, memory loss and fainting problems (Shaheamlunget al., 2020). Unnoticed at the initial stages, these symptoms are only visible when the disease turns chronic. However, even though the liver is partially infected, it can still function (Devikanniga et al., 2020). Diagnosis of liver diseases can be divided into three stages i.e., the first stage is liver inflammation, the second is liver scarring (cirrhosis), and the final stage is liver cancer or failure. Since these scenarios are present in liver disease, early prediction is significant to provide better health for New Zealanders. If liver disease is diagnosed early, there will be a chance of early treatment and control of deaths due to liver diseases (Arbain& Balakrishnan, 2019). But when the liver fails to function, few treatments are available except liver transplantation (Shaheamlung et al., 2020), which is very expensive, particularly in New Zealand (Hepatitis C, 2021). Apparently, in New Zealand, 35 - 40% of the population are not diagnosed with Hepatitis C at the early stages because of the asymptomatic behaviour of liver disease. Unfortunately, most of these individuals do not know the risks linked to liver disease. Due to the asymptomatic behaviour and higher costs of liver disease treatment, it is essential to prevent or diagnose early for better treatment. With advancements in biomedical sciences, the health care system has significantly

improved by predicting disease using machine learning techniques (El-Shafeiyet al., 2018). Machine Learning algorithms are one of the potential solutions to this problem due to their handling large amounts of data and employing different approaches like classification, association and clustering, which benefits in realistic arbitration of disease prediction (Naseem et al., 2020). There are different learning techniques in ML methods, one of which is supervised learning. Supervised learning techniques use labelled data and map the input and output data. These supervised learning methods are widely used for prediction and classification (Osisanwo et al., 2017). Supervised learning techniques would be appropriate as this research predicts whether the patient has liver diseaseor has no liver disease.

The supervised learning methods used in this study are Support Vector Machine(SVM) (Boser et al., 1992), Naïve Bayes (McCallum & Nigam,1998),K-Nearest Neighbors (K-NN)(Fix & Hodges, 1951), Classification and Regression Trees (CART) (Breiman et al., 1984), and LinearDiscriminant Analysis (LDA) (Kemp, 2003). The main objective of this research is to compare the accuracies using five supervised learning algorithms, i.e., SVM, Naïve Bayes, K-NN, CART, LDA and autoencoders, for predicting whether the patient has liver disease or not. This study also proposes the liver disease prediction (LDP) method to help relevant stakeholders pursue an effective healthcare strategy.

Moreover, this paper examines the techniques that indicate liver diseases at an acceptable level ofaccuracy and determines the methods that produce the best accuracy. This study selects a single data set of liver patients with five supervised learning techniques that are applied to that data set in R. The accuracy results from other learning techniques are also used to compare the best algorithm for predicting liver diseases. The stakeholders, including doctors, researchers, lab technicians, or companies dealing with healthcare improvements, can use these results to predict liver diseases at a lower cost and provide better health care in liver treatment.

A Brief Description about Project:

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 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 compares various classification algorithms such as Random Forest, Logistic Regression, KNN and ANN Algorithm with an aim to identify the best technique. Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver disease and can be recommended to the user.

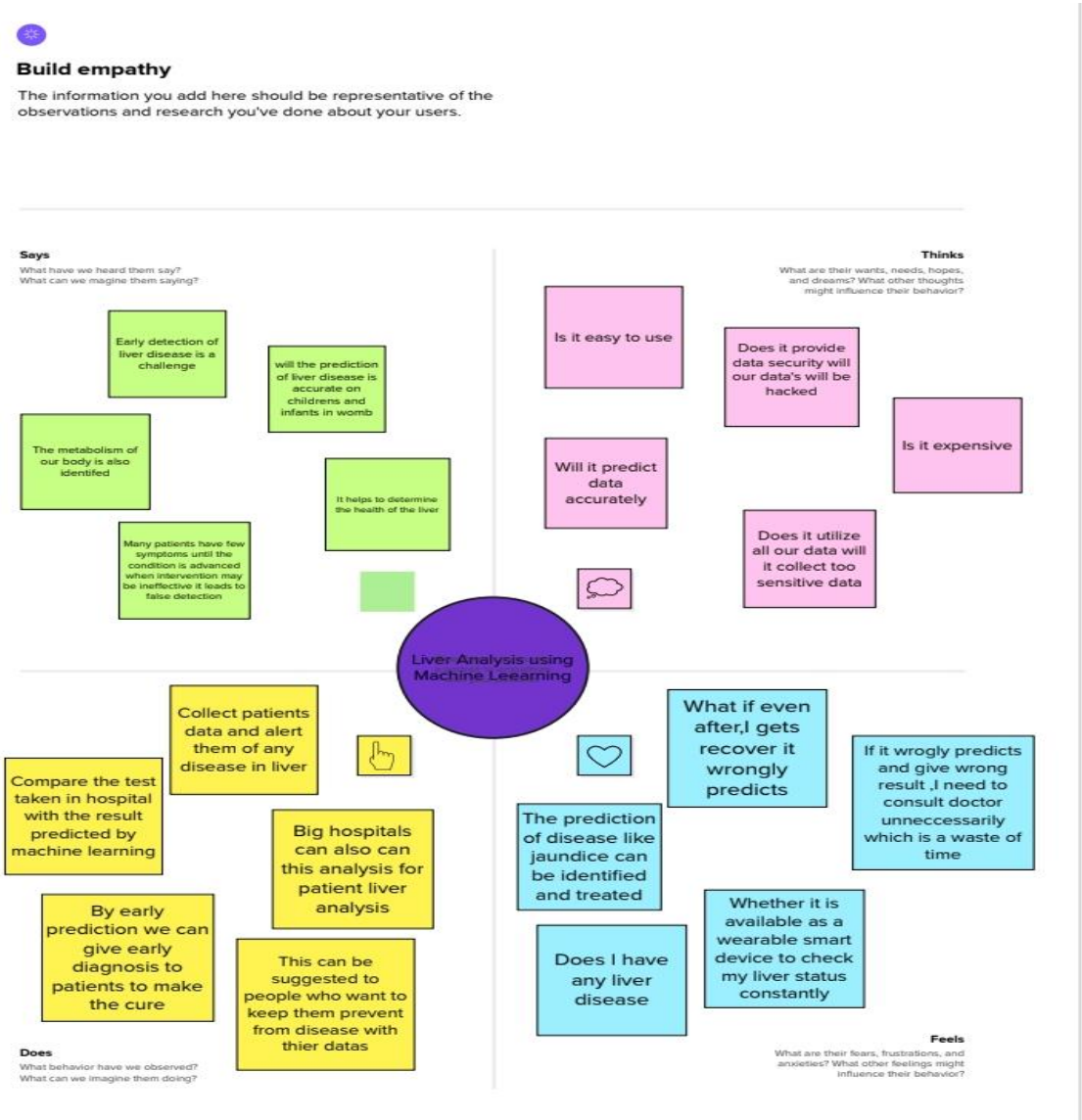
1.2Purpose:

The use of this project what can be achieved using this

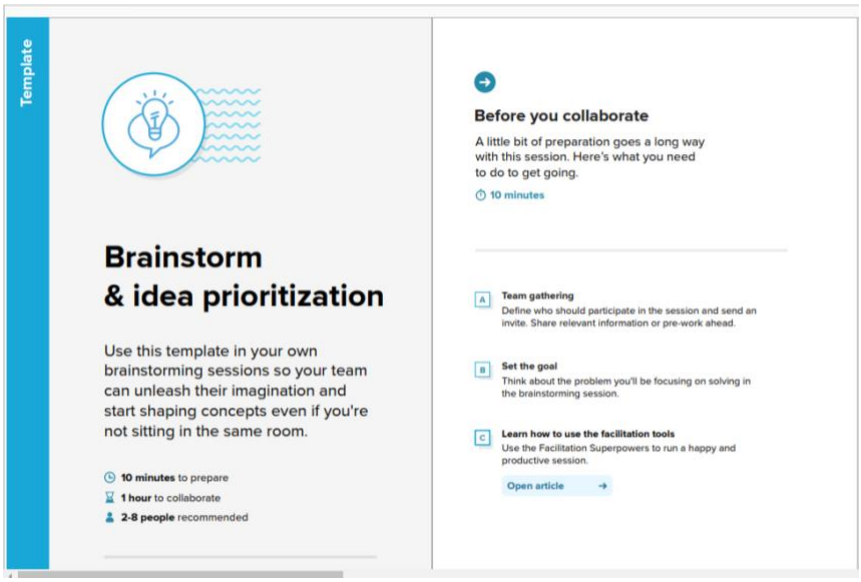
Early prediction of liver disease is very important to save human life and take proper steps to control the disease. Decision Tree algorithms have been successfully applied in various fields especially in medical science.This research work explores the early prediction of liver disease using various decision tree techniques. The liver disease dataset which is select for this study is consisting of attributes like total bilirubin, direct bilirubin, age, gender, total proteins, albumin and globulin ratio. The main purpose of this work is to calculate the performance of various decision tree techniques and compare their performance. The decision tree techniques used in this study are J48, LMT, Random Forest, Random tree, REPTree, Decision Stump, and Hoeffding Tree. The analysis proves that Decision Stump provides the highest accuracy than other techniques

2.Problem Definition and Design Thinking

2.1 EMPATHY MAP



2.2 Ideation & Brainstorming Map



Person 1

The right data's are to be collected

The right machine learning algorithm should be applied for predicting liver disease

Interactive and real time data should be collected for analysis

Person 2

The prdictions should be accurate

We can use K nearest neighbour to solve classification and regression problems

The data collected should be true

Person 3

Should avoid false prediction

For Collecting data we should also collect patient who have liver diesase from hospitals

Support vector machine can be used to build regression and classification models

Person 4

The entry of any virus can also causes liver problem

The training and testing of data should be divided in a 80:20 ratio

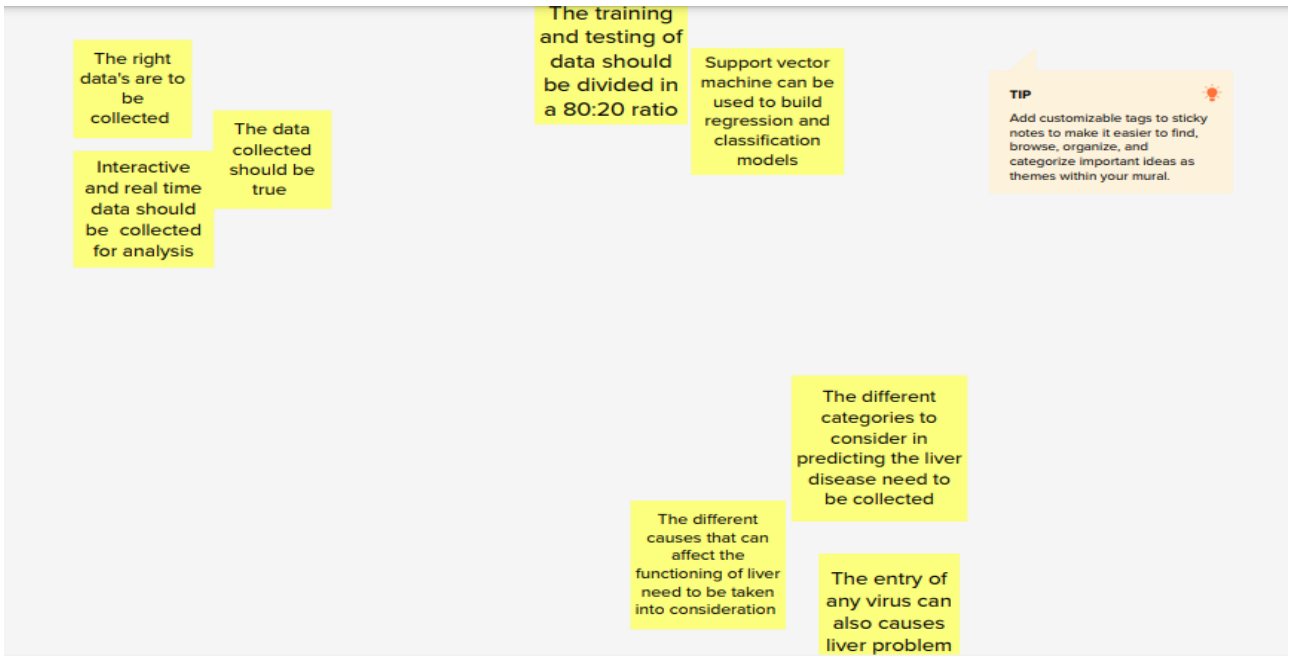
Data preprocessing should be done for this we can use jupiter notebook

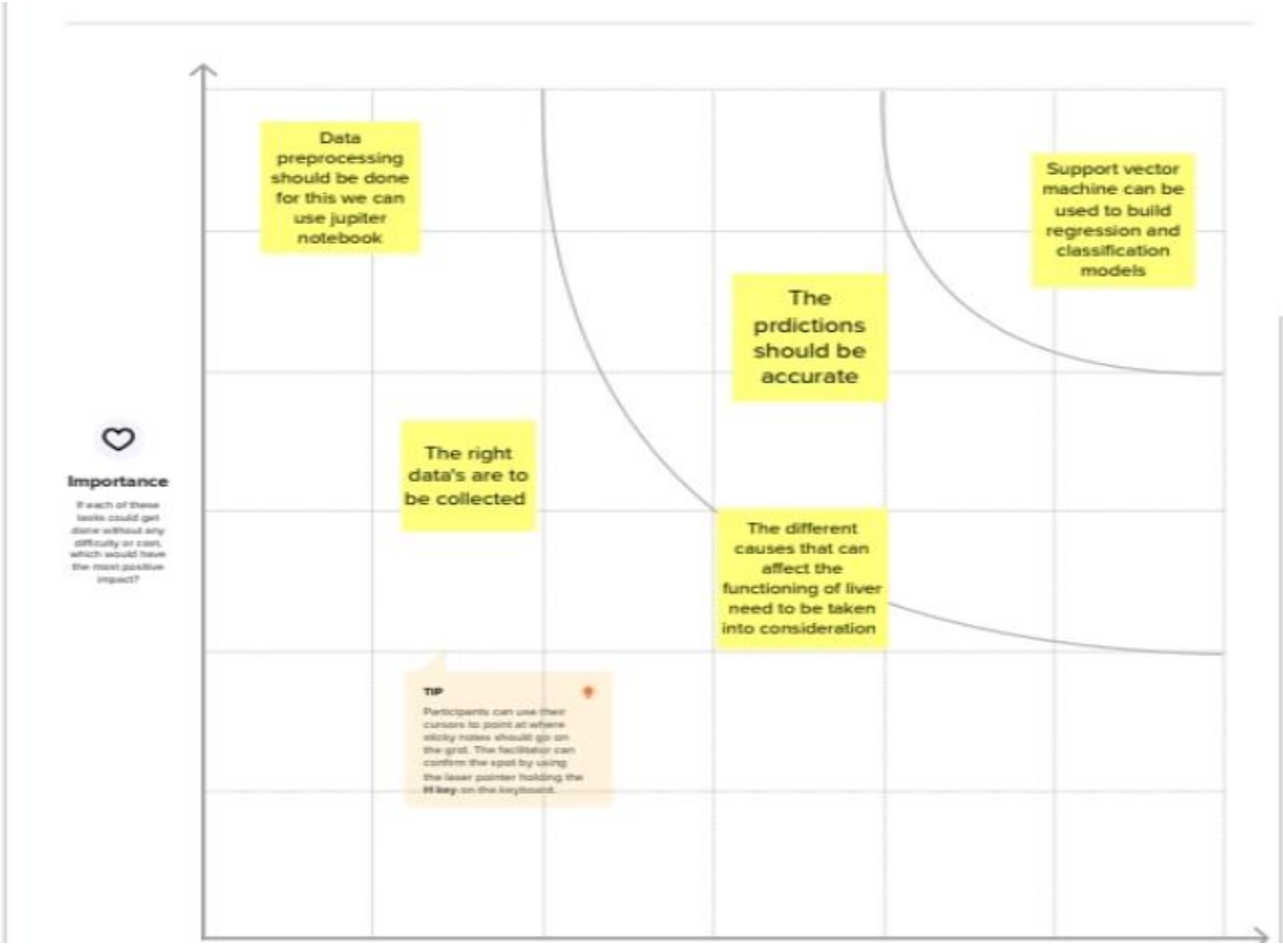
Person 5

The false data and incomplete data should be filtered

The different categories to consider in predicting the liver disease need to be collected

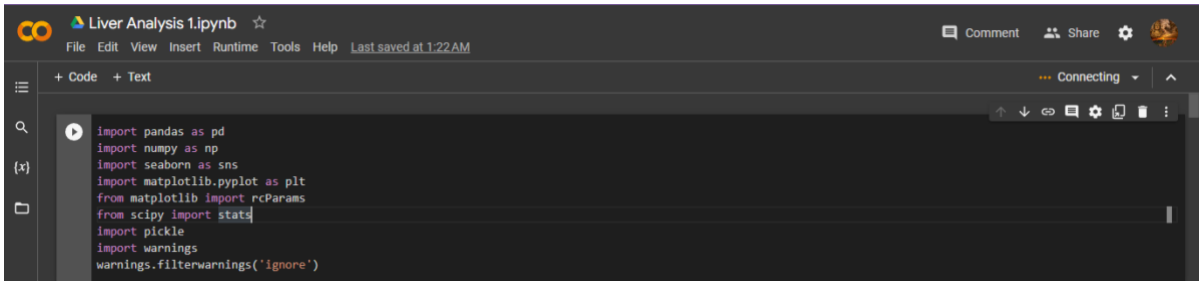
The different causes that can affect the functioning of liver need to be taken


The right data's are to be collected
The data collected should be true
Interactive and real time data should be collected for analysis
The training and testing of data should be divided in a 80:20 ratio
Support vector machine can be used to build regression and classification models
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
The different categories to consider in predicting the liver disease need to be collected
The different causes that can affect the functioning of liver need to be taken into consideration
The entry of any virus can also causes liver problem

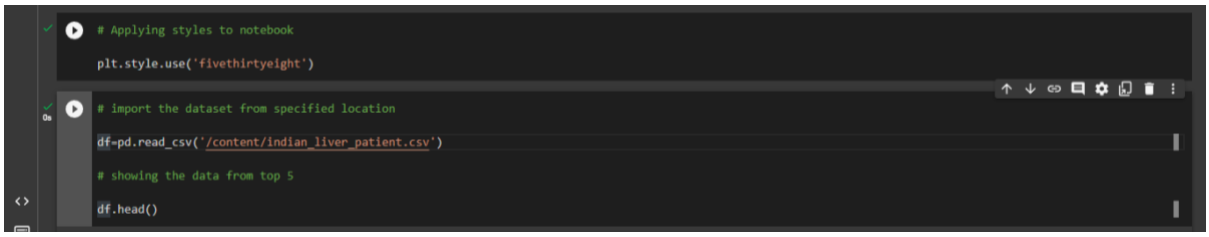

Data preprocessing should be done for this we can use jupiter notebook
Support vector machine can be used to build regression and classification models
The prdictions should be accurate
The right data's are to be collected
Importance
The different causes that can affect the functioning of liver need to be taken into consideration
TIP

Results:

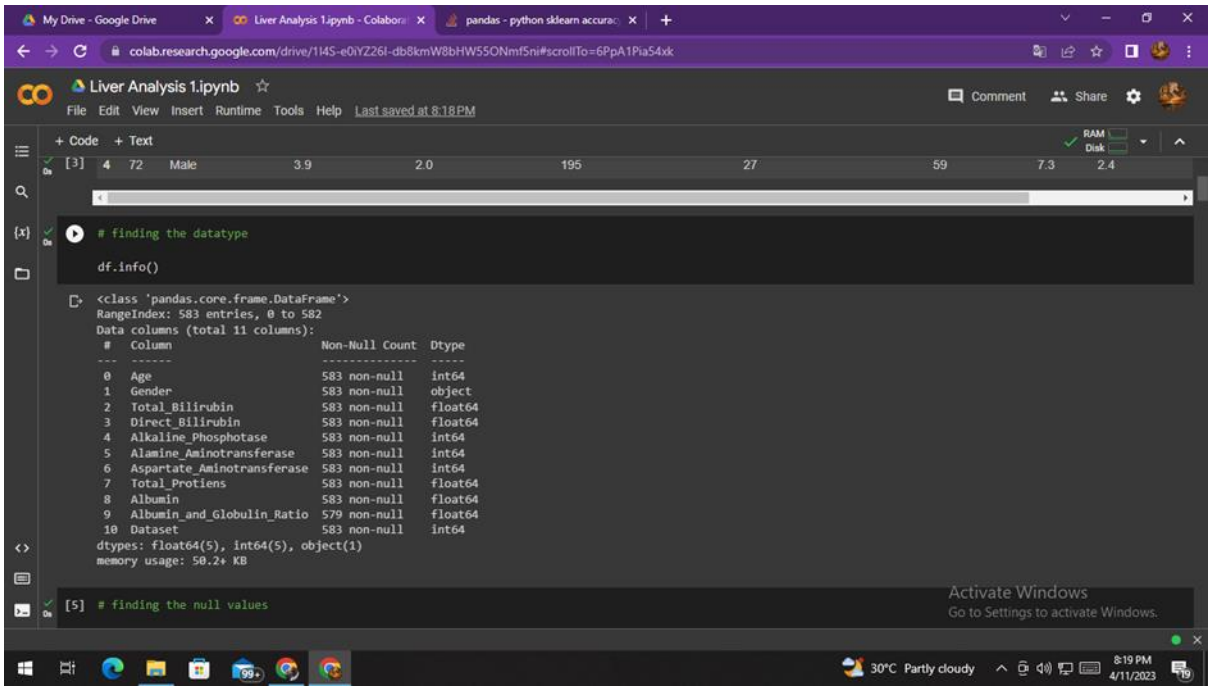
Importing required libraries:



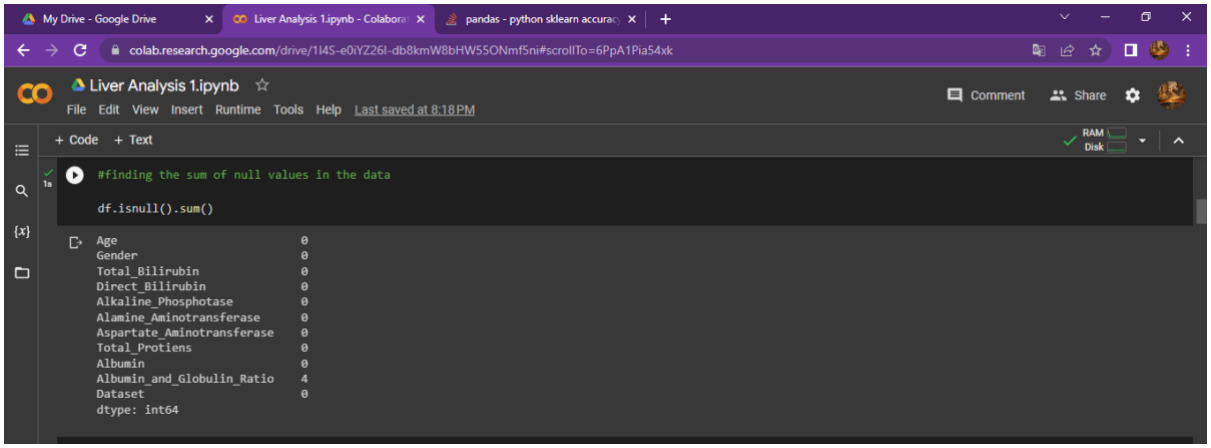
Read the csv file:



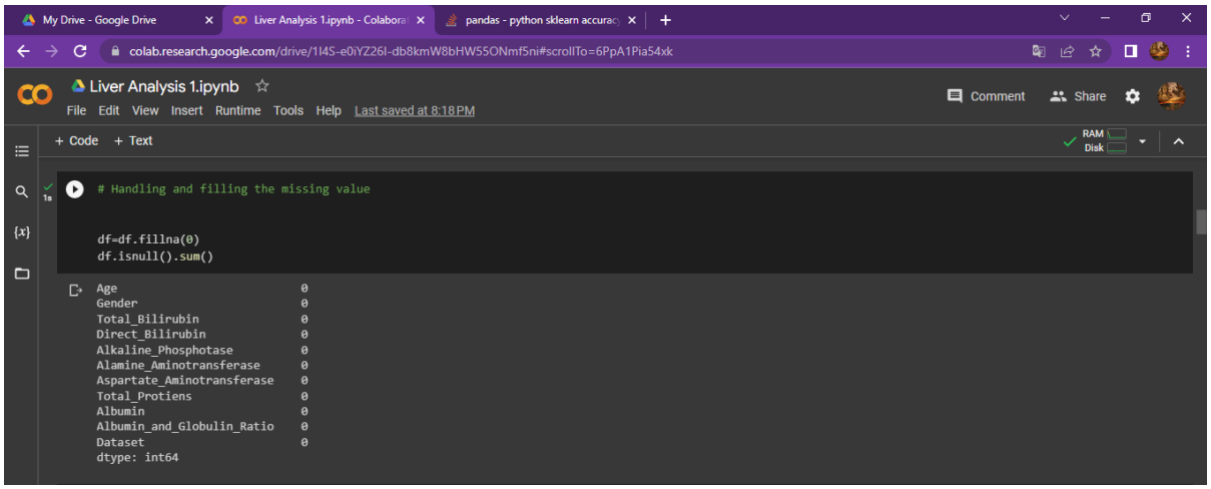
Getting the information using info() :



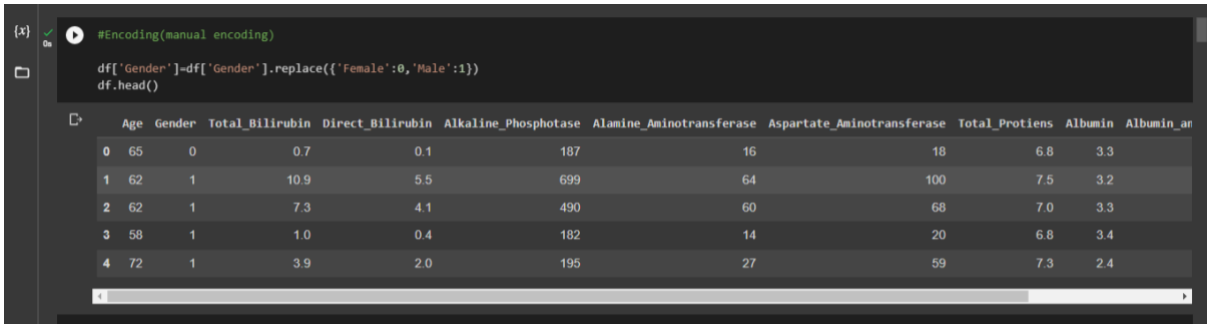
Finding sum of Null Values isnull().sum() function:



Handling and Filling the missing values:

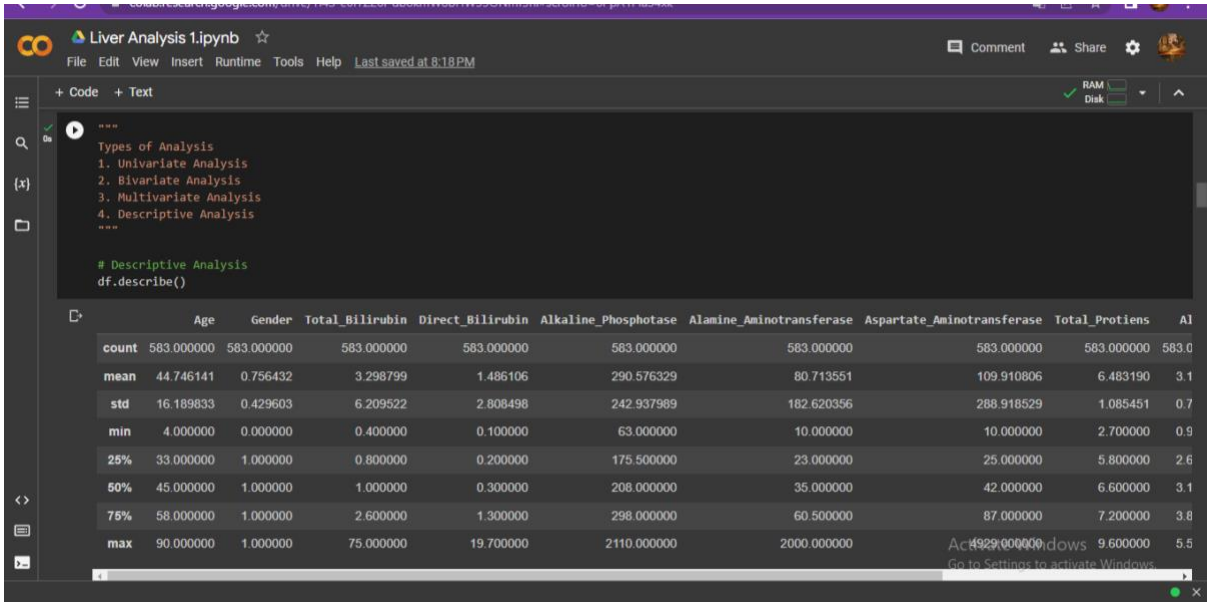


Manual Encoding:



	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_an
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	

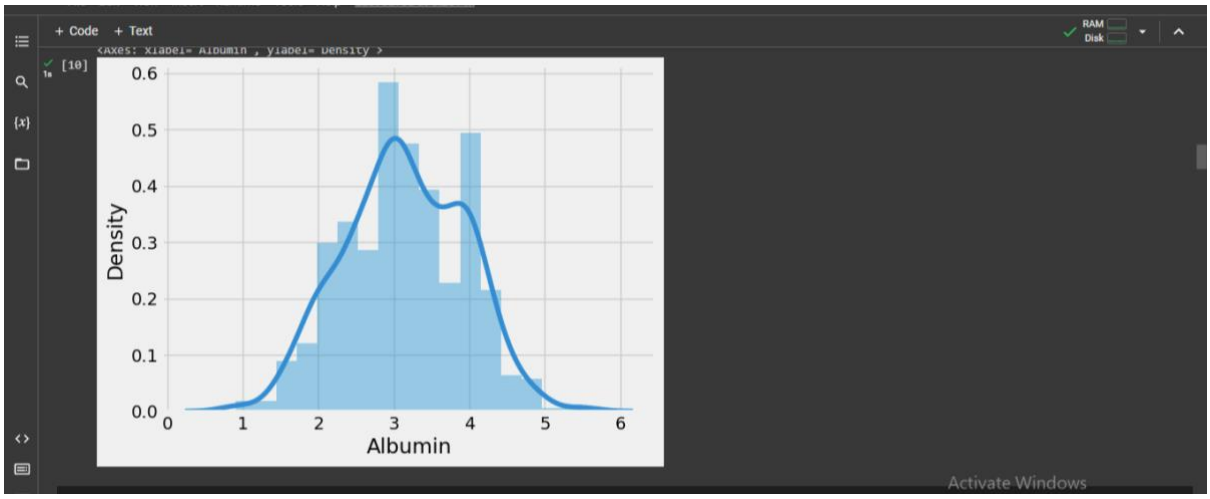
TYPES OF ANALYSIS (Descriptive Analysis):



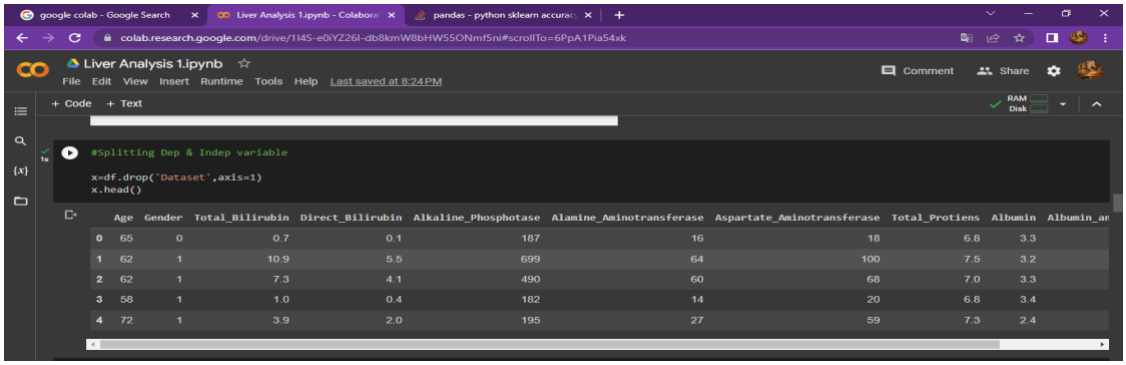
	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Al
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.0
mean	44.746141	0.756432	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.1
std	16.189833	0.429603	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.7
min	4.000000	0.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.9
25%	33.000000	1.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.6
50%	45.000000	1.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.1
75%	58.000000	1.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.8
max	90.000000	1.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.5

i)Univariate Analysis:

- Univariate Analysis is defined as carried out on only one variable to summarize or describe the variable.



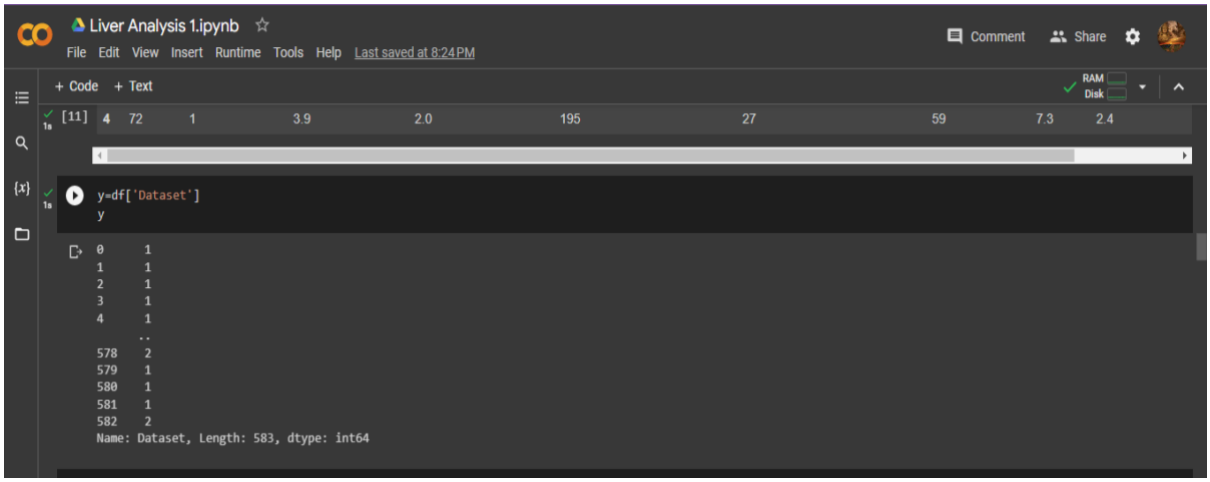
Splitting the data into dependent and independent variable:



```
#Splitting Dep & Indep variable

x=df.drop('Dataset',axis=1)
x.head()
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_ar
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	

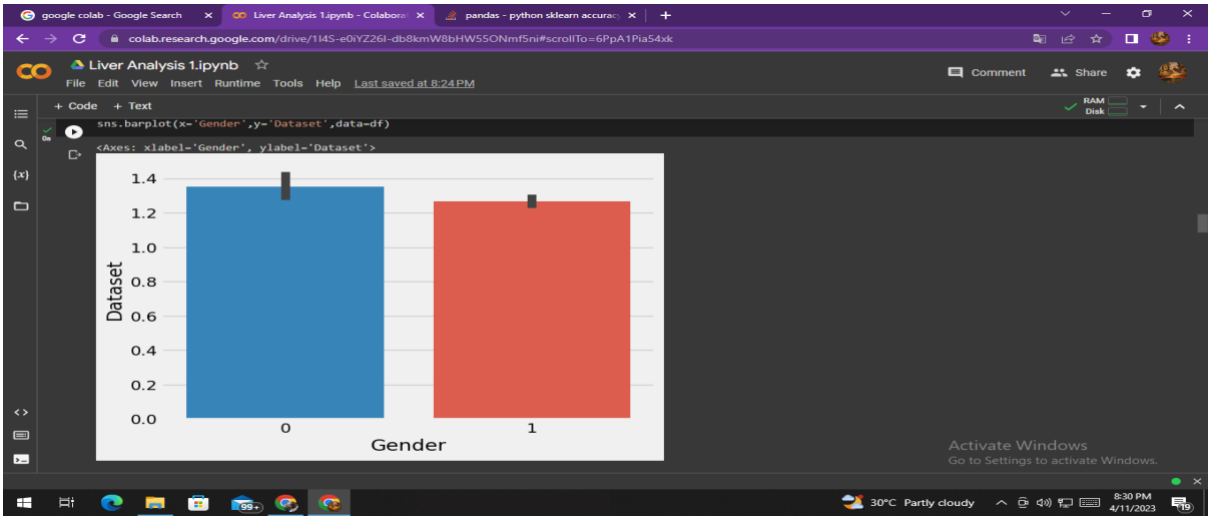


```
y=df['Dataset']
y
```

```
0      1
1      1
2      1
3      1
4      1
      ..
578    2
579    1
580    1
581    1
582    2
Name: Dataset, Length: 583, dtype: int64
```

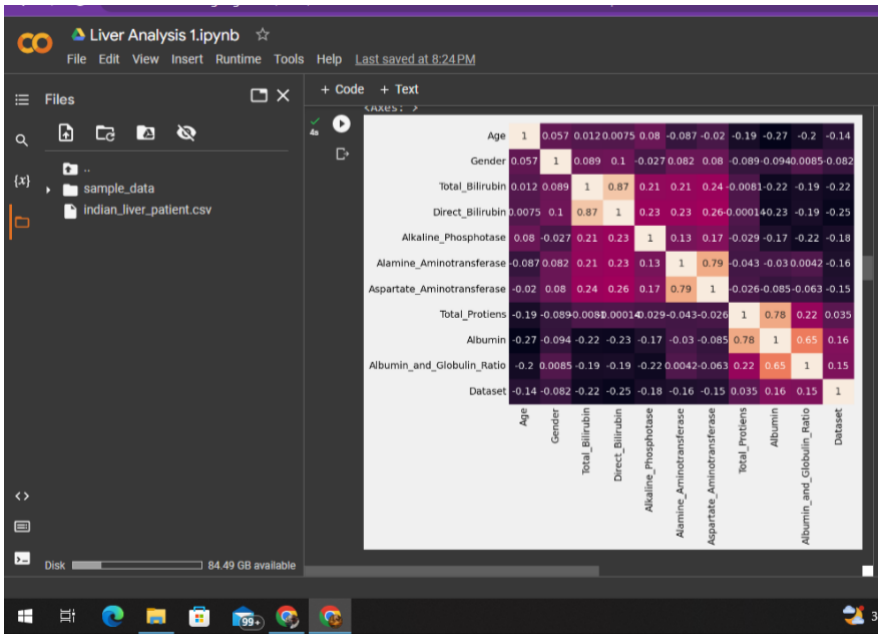
ii)Bivariate Analysis:

- ❖ Bivariate Analysis refers to the analysis of two variable to determine relationship between them.



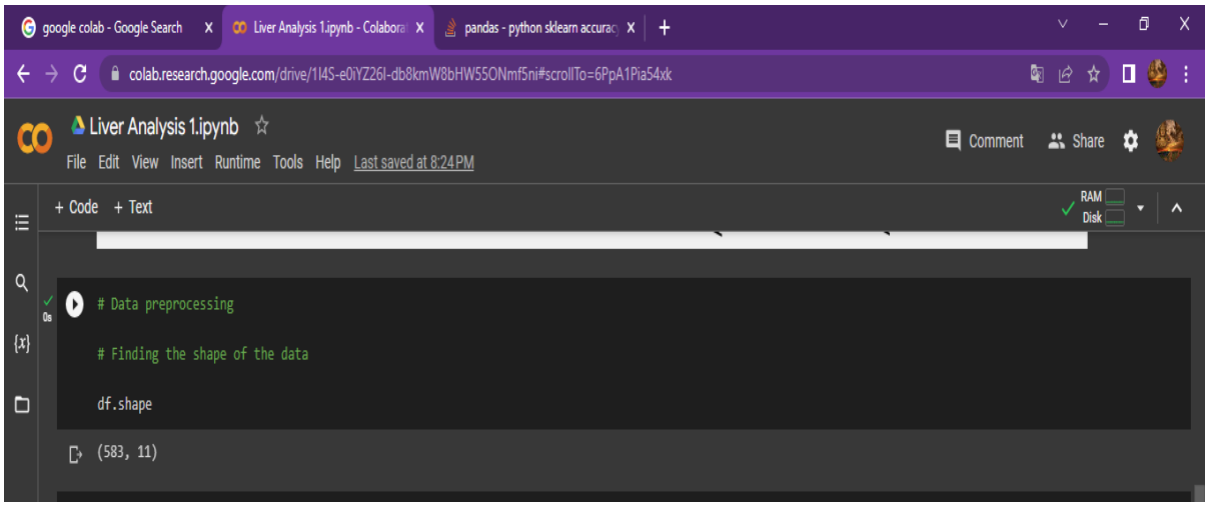
iii)Multivariate Analysis:

- ❖ Multivariate Analysis is based in observation and analysis of more than one statistical outcome variable at a time.

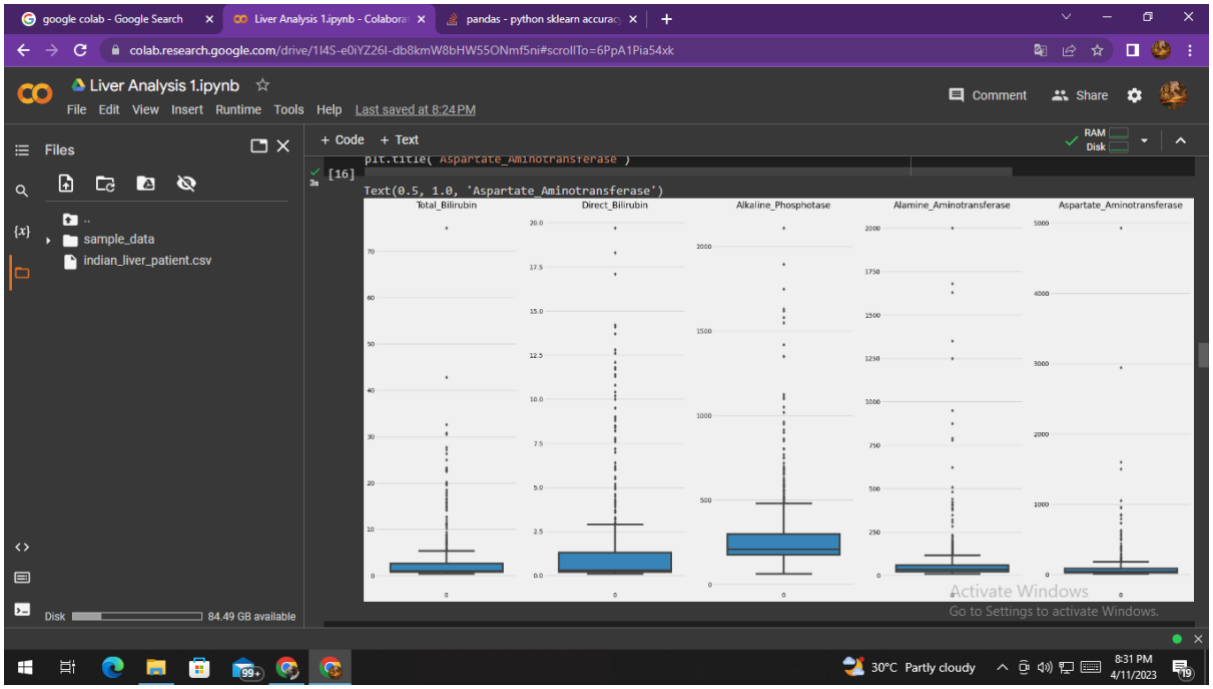


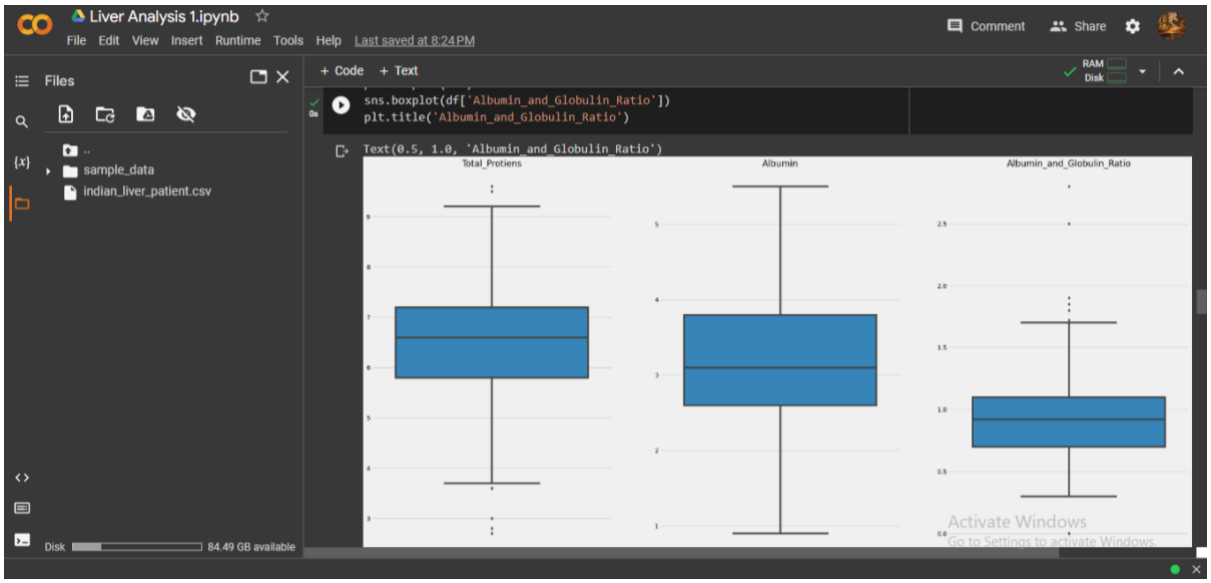
DATA PREPROCESSING:

Finding the shape of the data

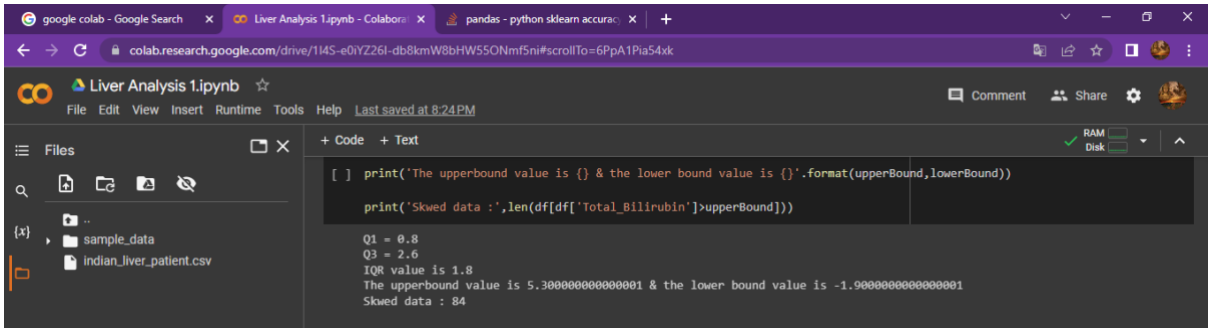


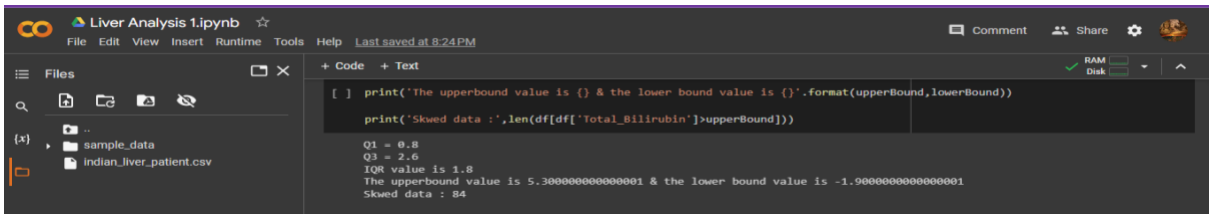
Finding outliers:

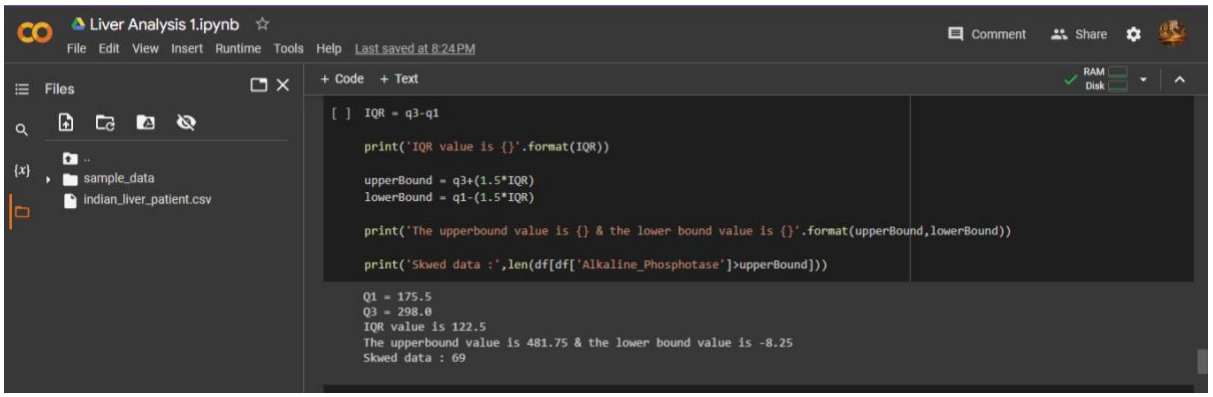


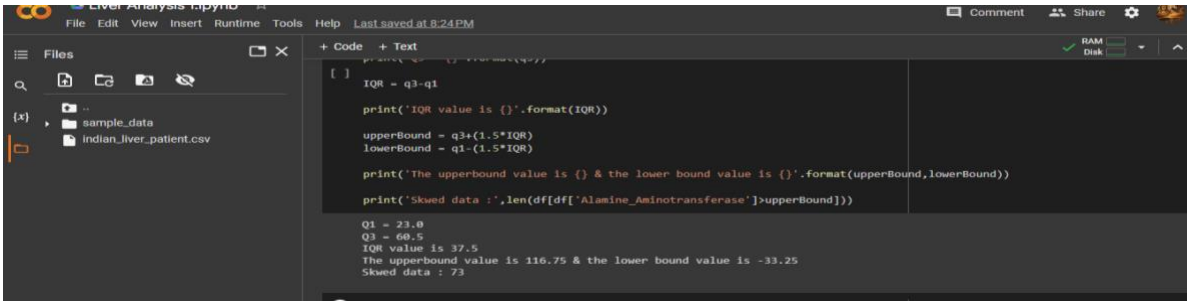


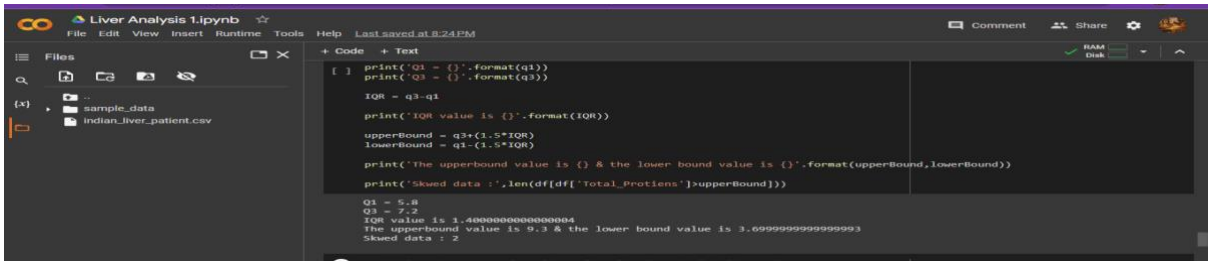
Count of Outliers:

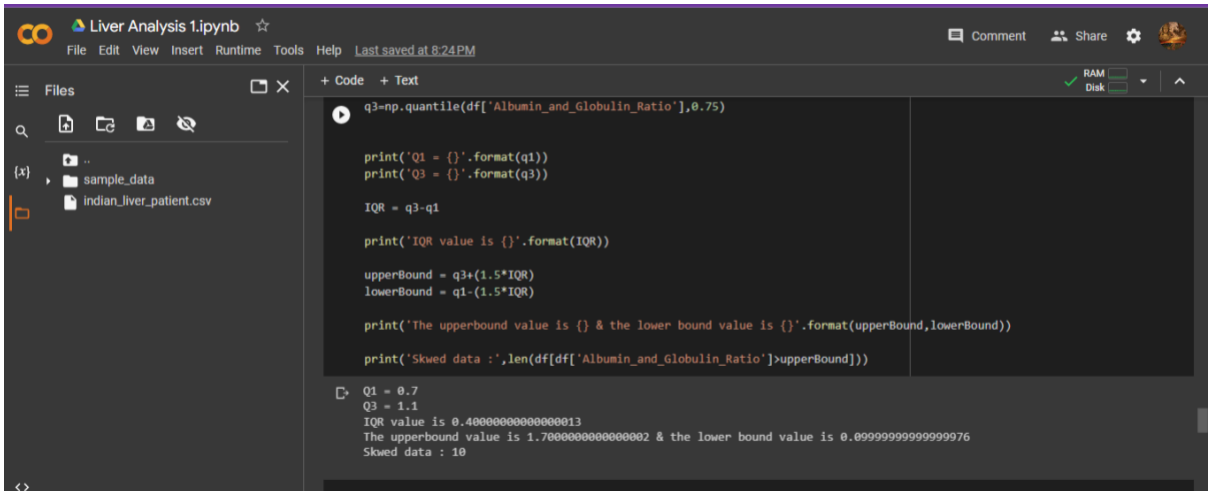


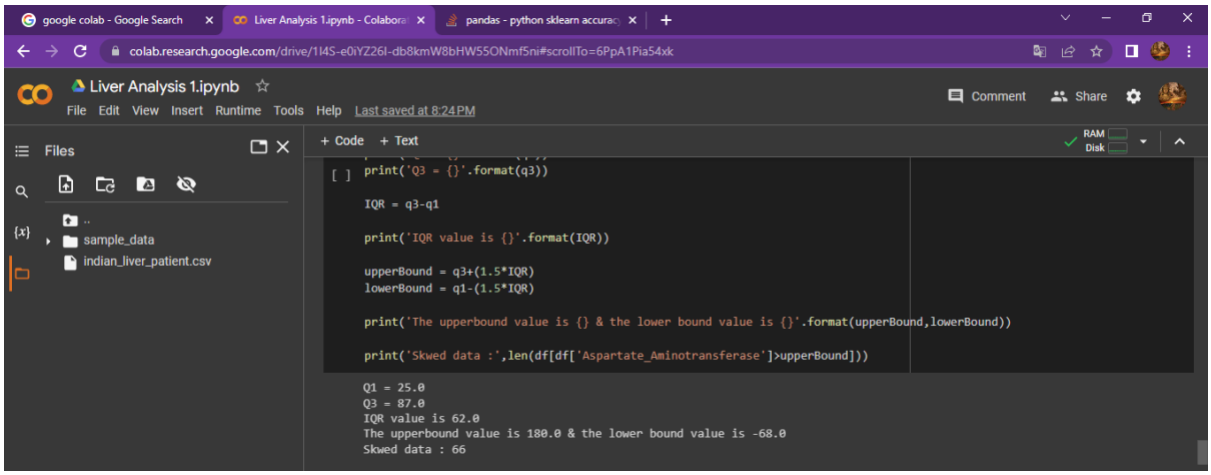




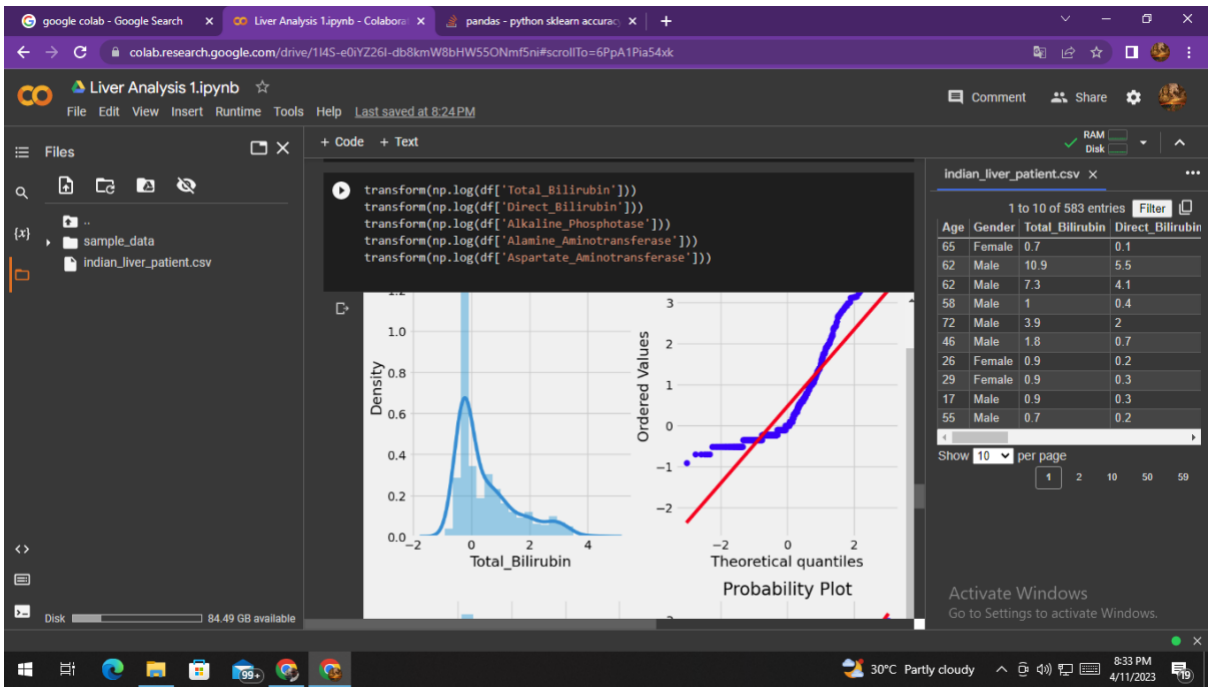


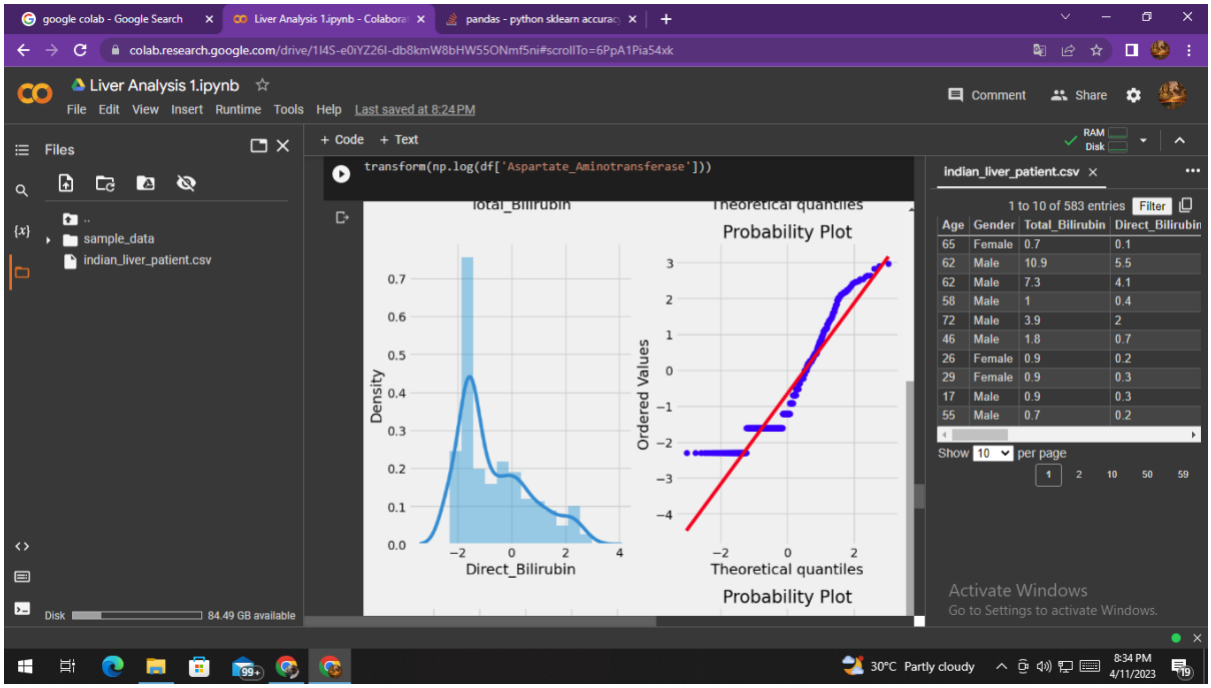


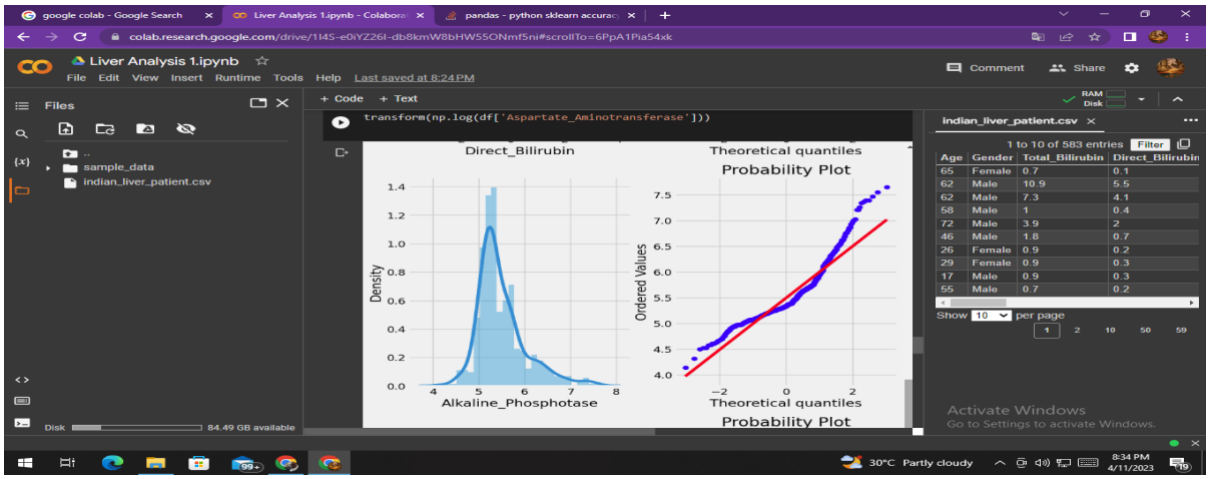


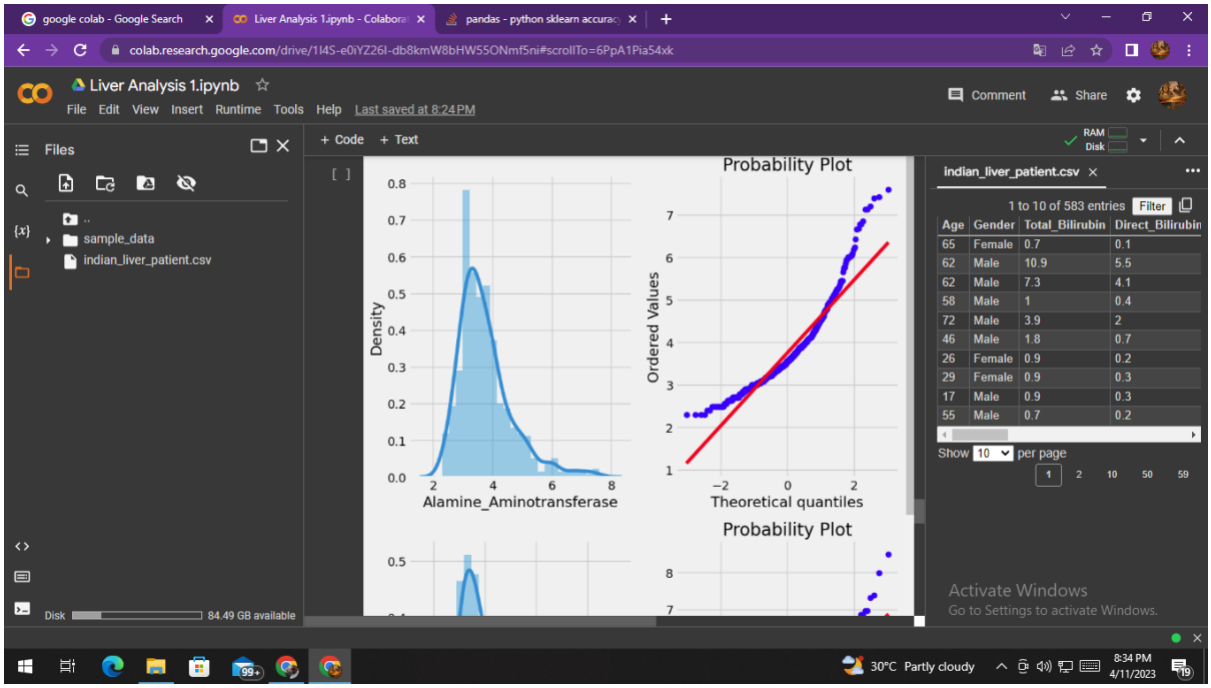


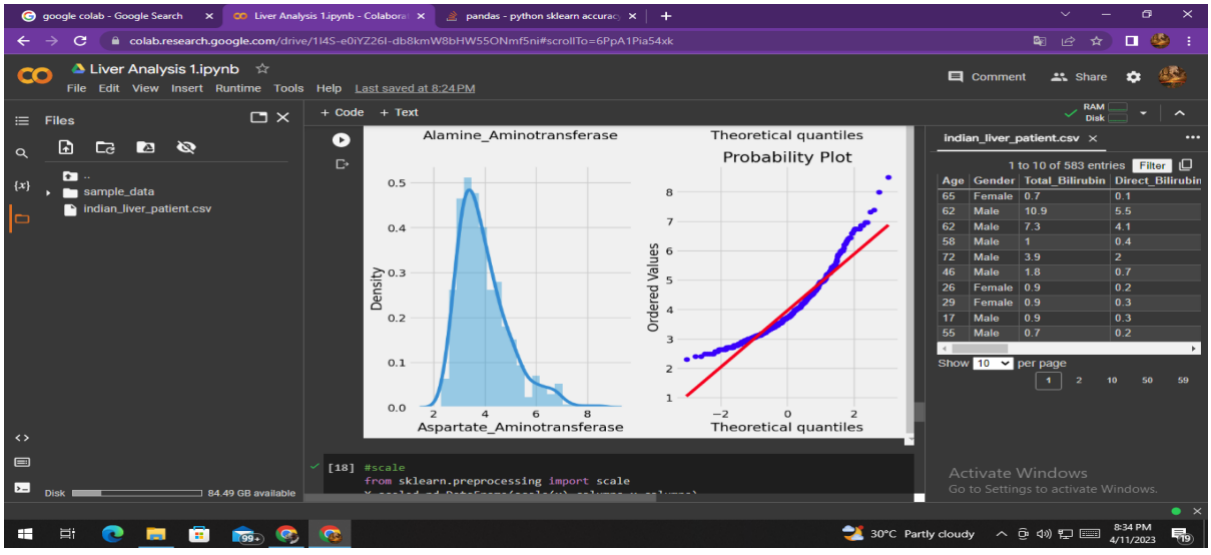
Transformation log:



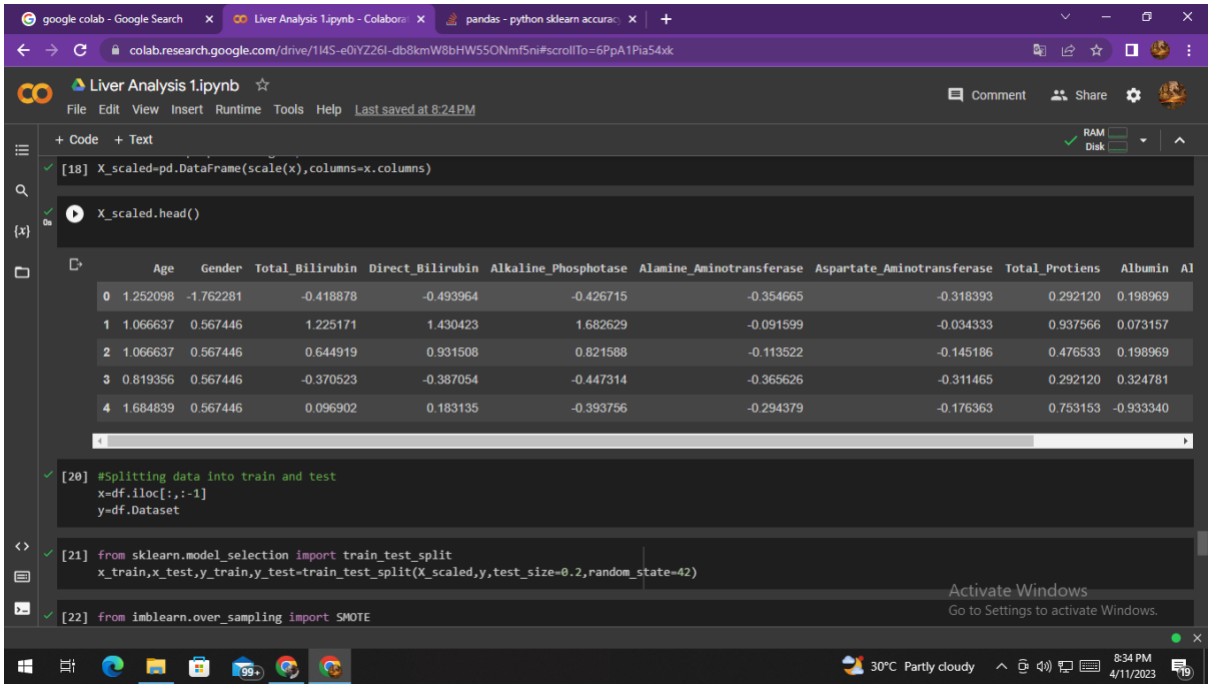

google colab - Google Search
Liver Analysis 1.ipynb - Colabora
pandas - python sklearn accurac
colab.research.google.com/drive/1I4S-e0iYZ26I-db8kmW8bHW55ONmf5ni#scrollTo=6PpA1Pia54xk
Liver Analysis 1.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:24 PM
Comment
Share
Files
sample_data
indian_liver_patient.csv
+ Code + Text
transform(np.log(df['Aspartate_Aminotransferase']))
Total_Bilirubin
Density
Direct_Bilirubin
Theoretical quantiles
Probability Plot
Ordered Values
Theoretical quantiles
Probability Plot
indian_liver_patient.csv
1 to 10 of 583 entries
Filter
Age Gender Total_Bilirubin Direct_Bilirubin
65 Female 0.7 0.1
62 Male 10.9 5.5
62 Male 7.3 4.1
58 Male 1 0.4
72 Male 3.9 2
46 Male 1.8 0.7
26 Female 0.9 0.2
29 Female 0.9 0.3
17 Male 0.9 0.3
55 Male 0.7 0.2
Show 10 per page
1 2 10 50 59
Disk 84.49 GB available
Activate Windows
Go to Settings to activate Windows.
30°C Partly cloudy
8:34 PM
4/11/2023


google colab - Google Search
Liver Analysis 1.ipynb - Colabora
pandas - python sklearn accurac
colab.research.google.com/drive/1I4S-e0iYZ26I-db8kmW8bHW55ONmf5ni#scrollTo=6PpA1Pia54xk
Liver Analysis 1.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:24 PM
Comment
Share
Files
sample_data
indian_liver_patient.csv
+ Code + Text
transform(np.log(df['Aspartate_Aminotransferase']))
Direct_Bilirubin
Density
Alkaline_Phosphotase
Theoretical quantiles
Probability Plot
Ordered Values
Theoretical quantiles
Probability Plot
indian_liver_patient.csv
1 to 10 of 583 entries
Filter
Age Gender Total_Bilirubin Direct_Bilirubin
65 Female 0.7 0.1
62 Male 10.9 5.5
62 Male 7.3 4.1
58 Male 1 0.4
72 Male 3.9 2
46 Male 1.8 0.7
26 Female 0.9 0.2
29 Female 0.9 0.3
17 Male 0.9 0.3
55 Male 0.7 0.2
Show 10 per page
1 2 10 50 59
Disk 84.49 GB available
Activate Windows
Go to Settings to activate Windows.
30°C Partly cloudy
8:34 PM
4/11/2023

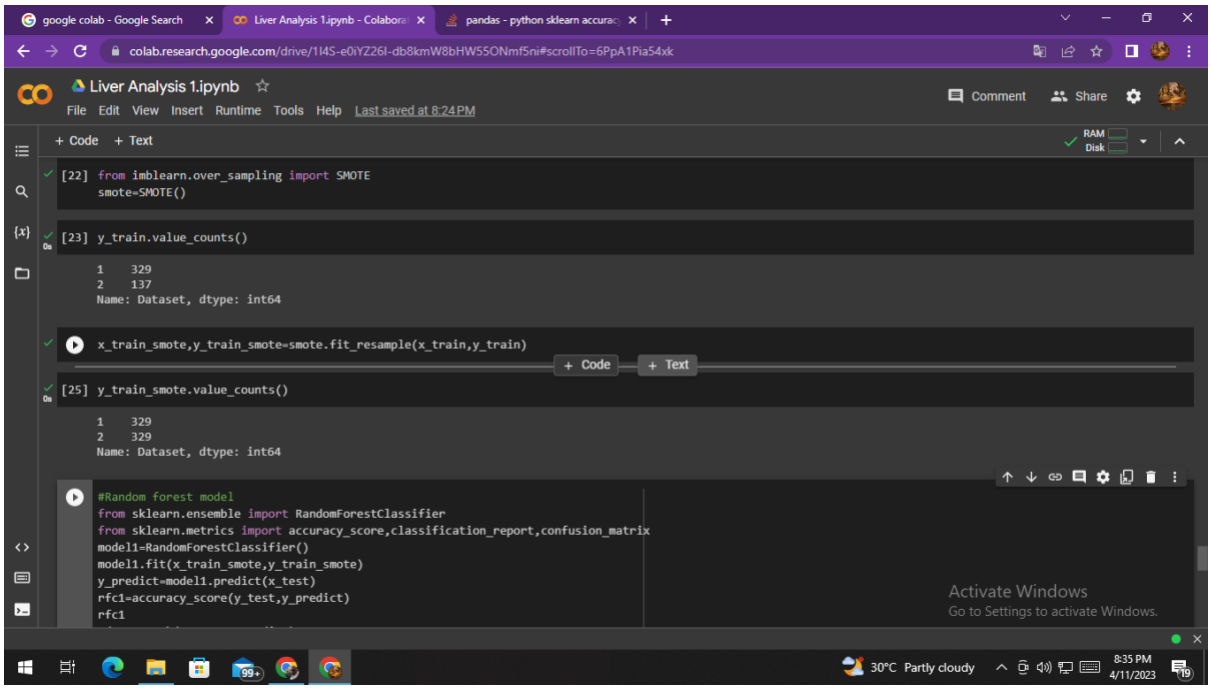

Liver Analysis 1.ipynb
Probability Plot
Density
Ordered Values
Alamine_Aminotransferase
Theoretical quantiles
indian_liver_patient.csv
1 to 10 of 583 entries
Age Gender Total_Bilirubin Direct_Bilirubin
65 Female 0.7 0.1
62 Male 10.9 5.5
62 Male 7.3 4.1
58 Male 1 0.4
72 Male 3.9 2
46 Male 1.8 0.7
26 Female 0.9 0.2
29 Female 0.9 0.3
17 Male 0.9 0.3
55 Male 0.7 0.2
Activate Windows


Liver Analysis 1.ipynb
Alamine_Aminotransferase
Theoretical quantiles
Probability Plot
Density
Ordered Values
Aspartate_Aminotransferase
Theoretical quantiles
indian_liver_patient.csv
1 to 10 of 583 entries
Age Gender Total_Bilirubin Direct_Bilirubin
65 Female 0.7 0.1
62 Male 10.9 5.5
62 Male 7.3 4.1
58 Male 1 0.4
72 Male 3.9 2
46 Male 1.8 0.7
26 Female 0.9 0.2
29 Female 0.9 0.3
17 Male 0.9 0.3
55 Male 0.7 0.2
[18] #scale
from sklearn.preprocessing import scale
Activate Windows

Scaling the Data:



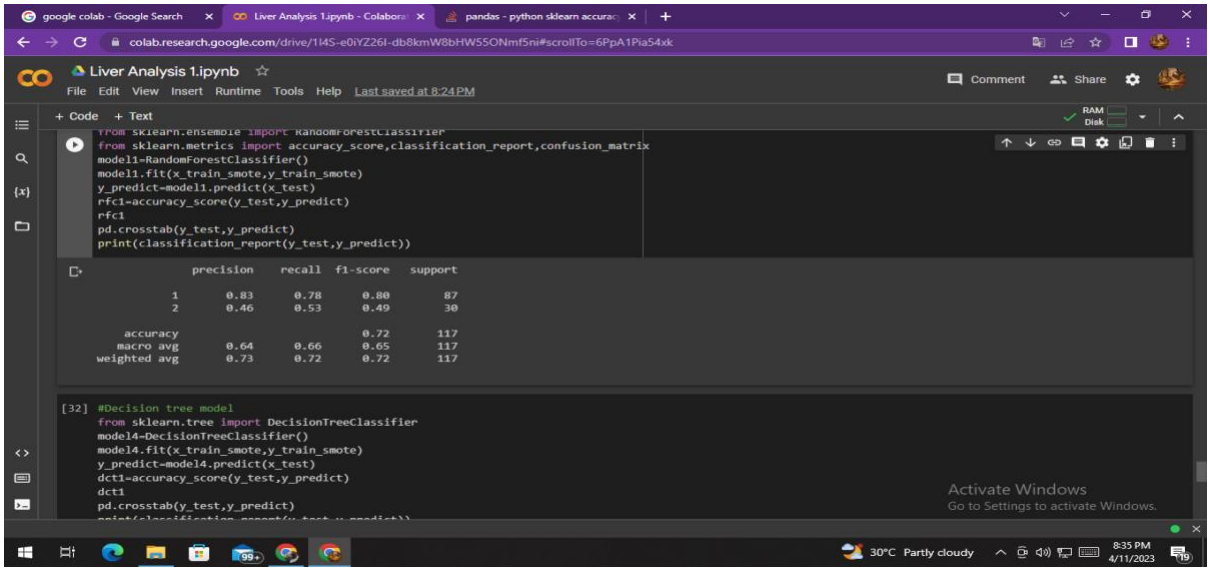
Splitting the data into training and testing data:



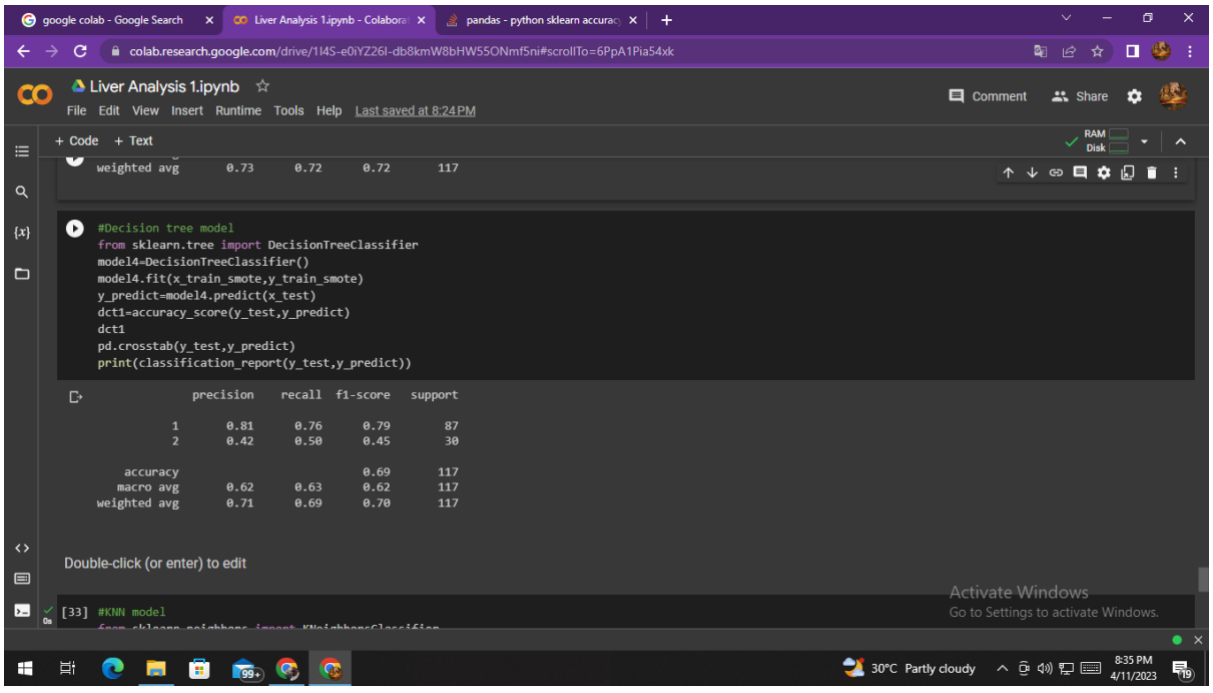
MODEL BUILDING:

Applying Algorithms

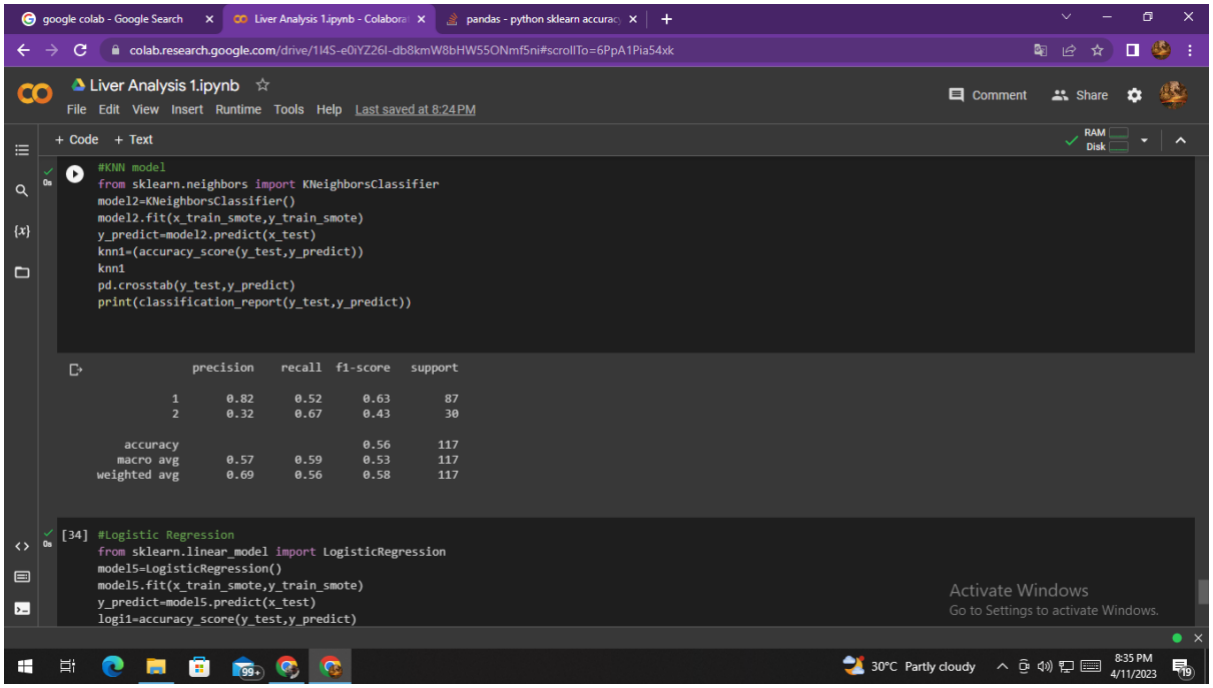
i) Random Forest Algorithm



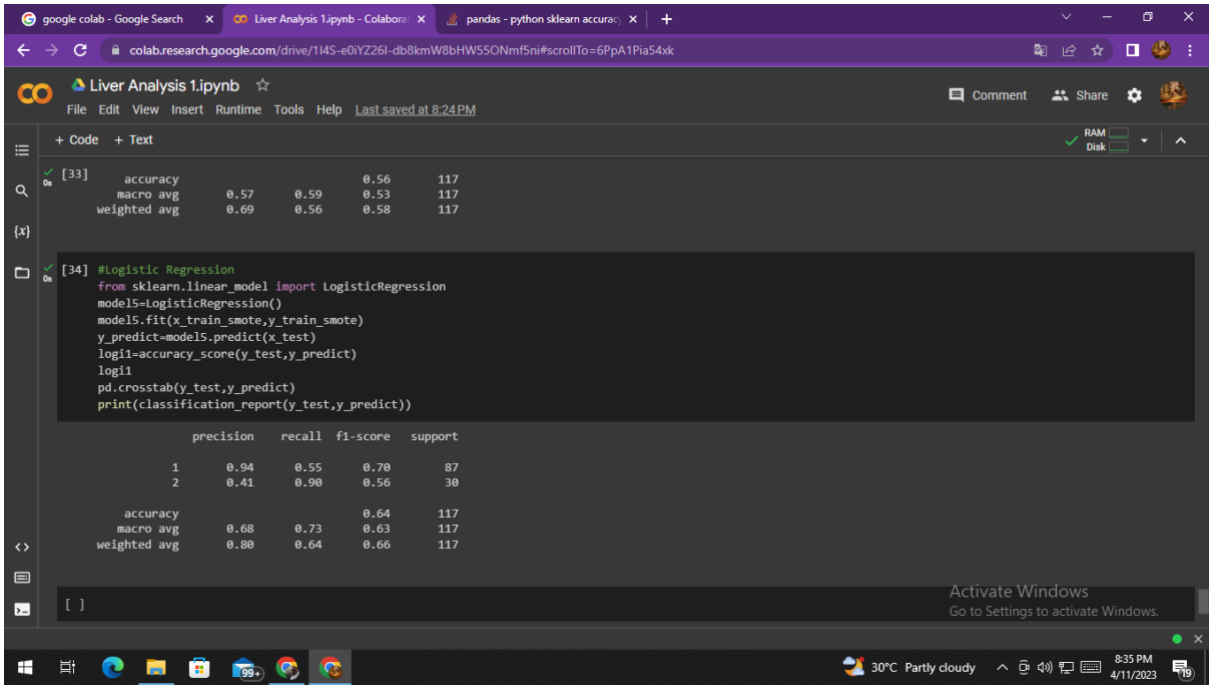
ii)Decision Tree



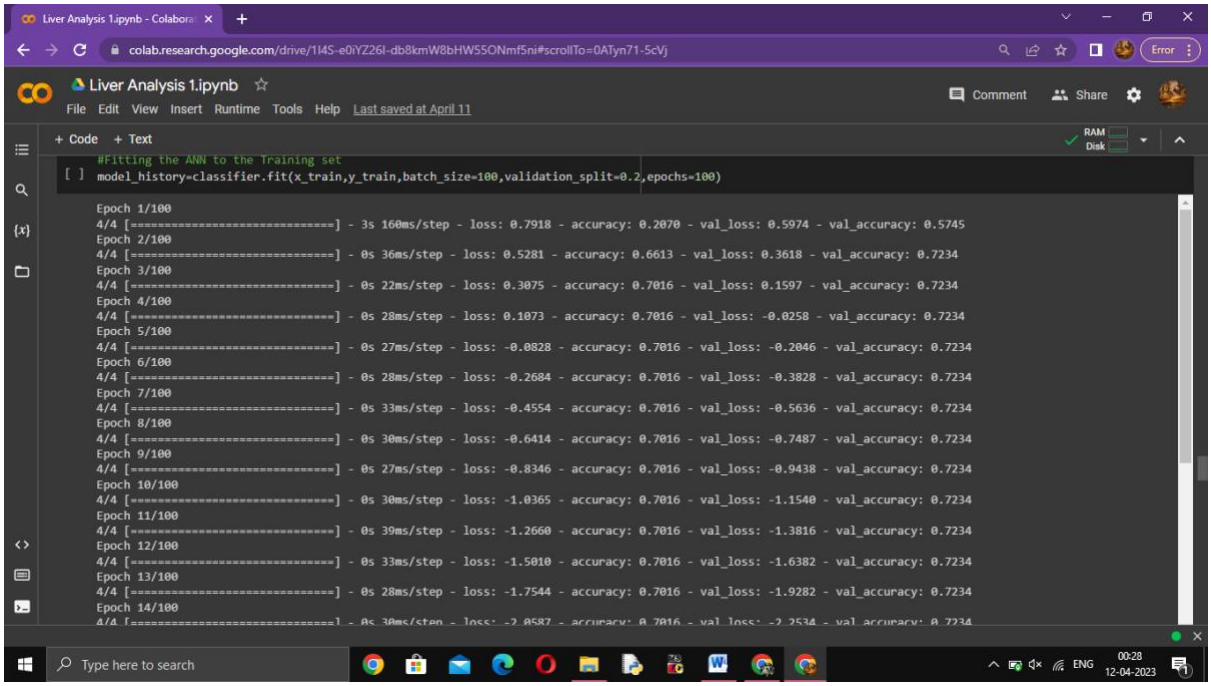
iii)K- Nearest Neighbour Algorithm



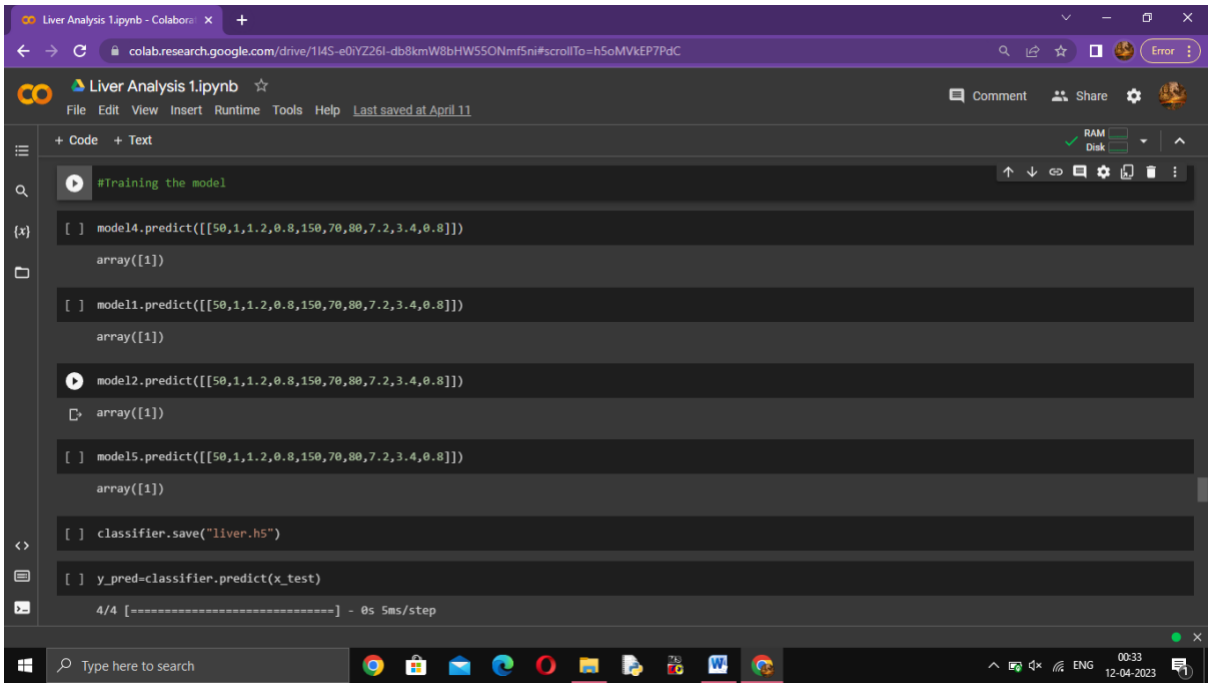
iv)Logistic Regression

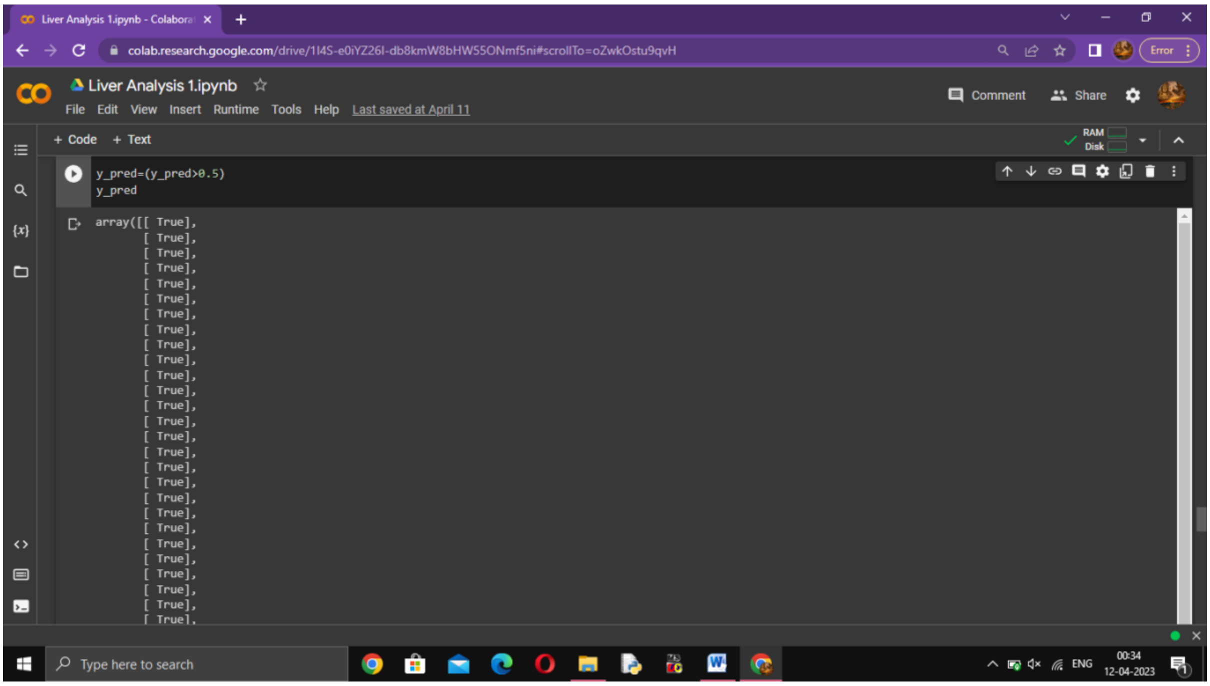


Fitting the ANN to the training set:

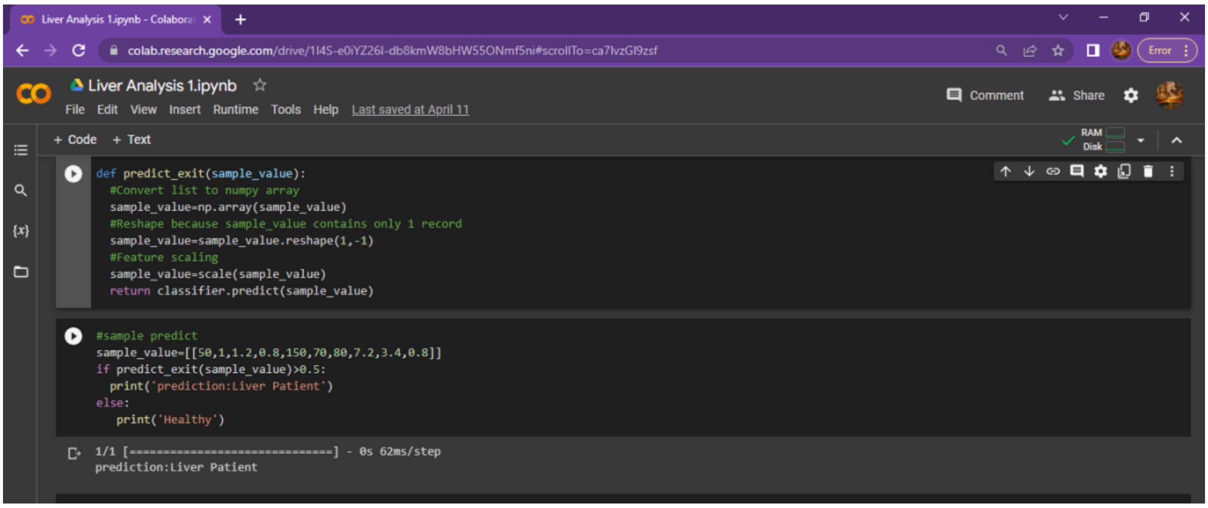


Testing the model:



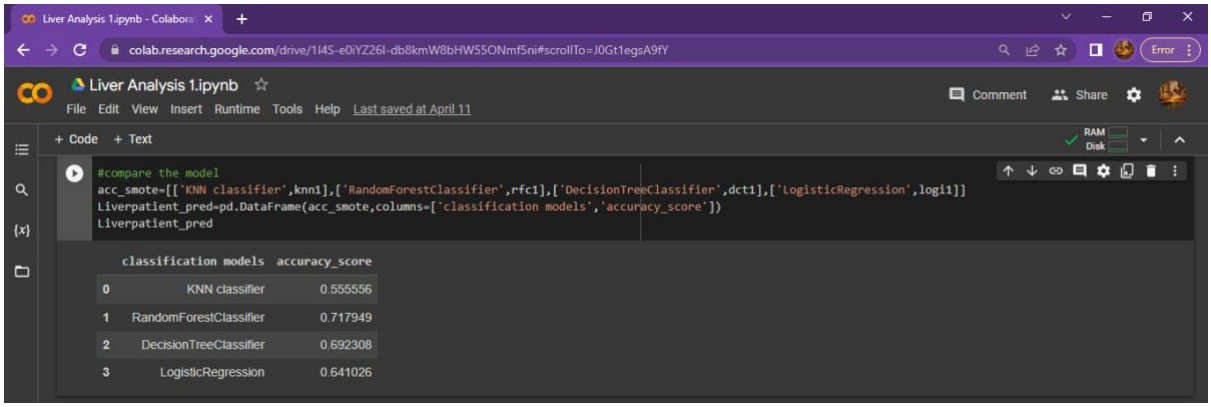


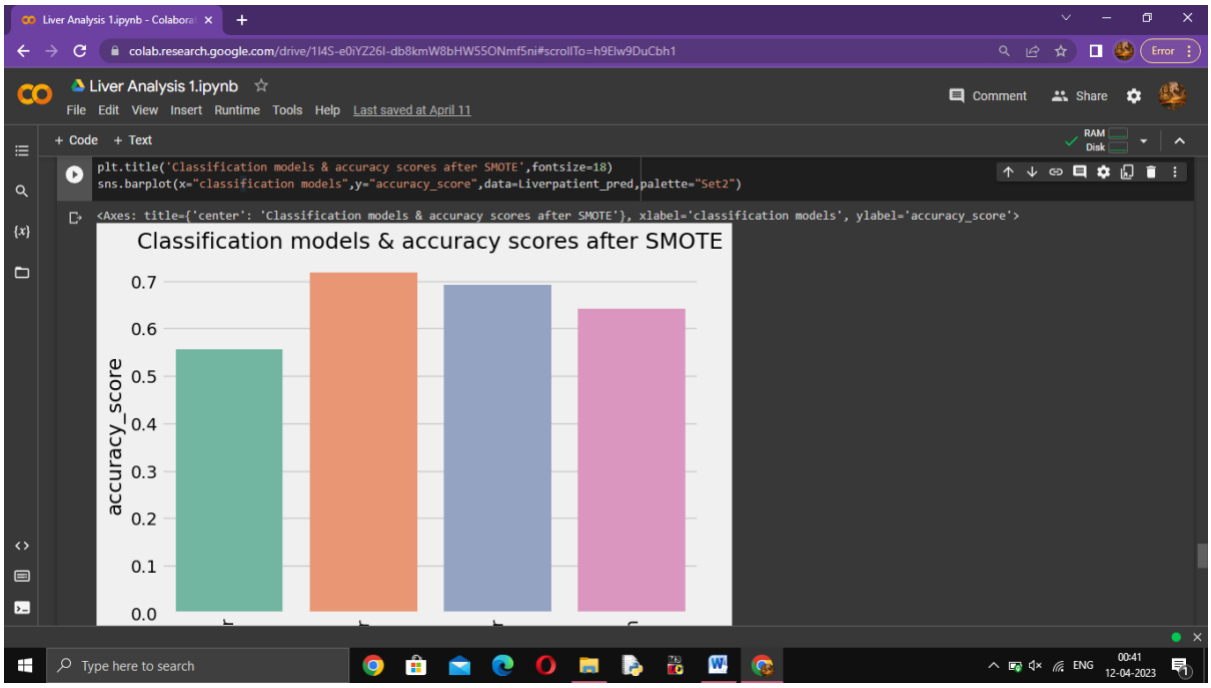
Predicting the Test data:

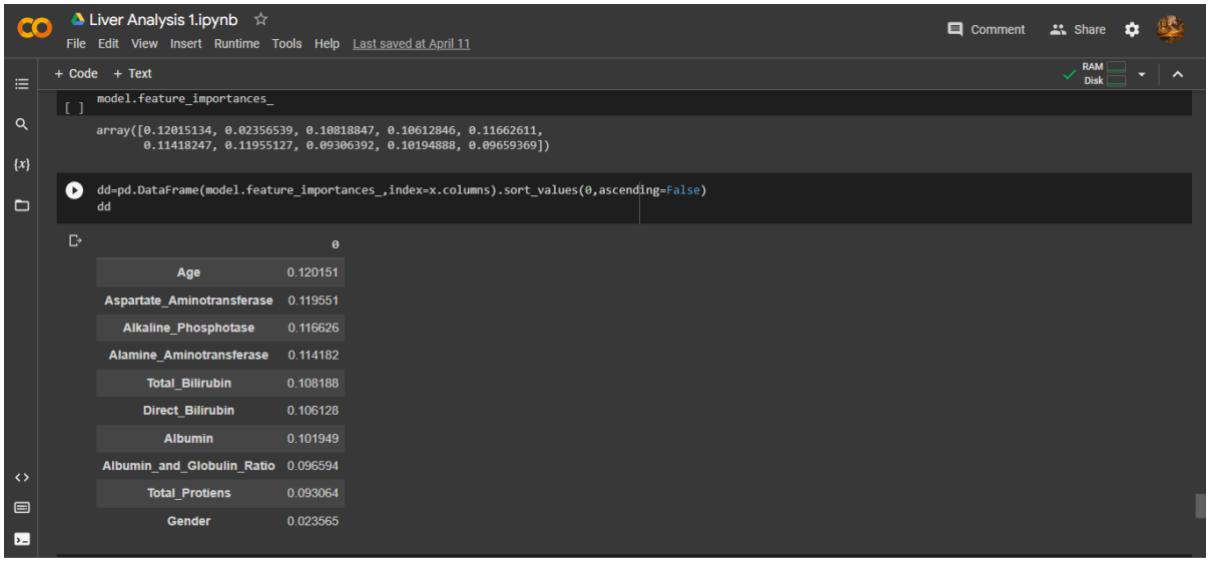


PERFORMANCE TESTING AND HYPER PARAMETER TUNING

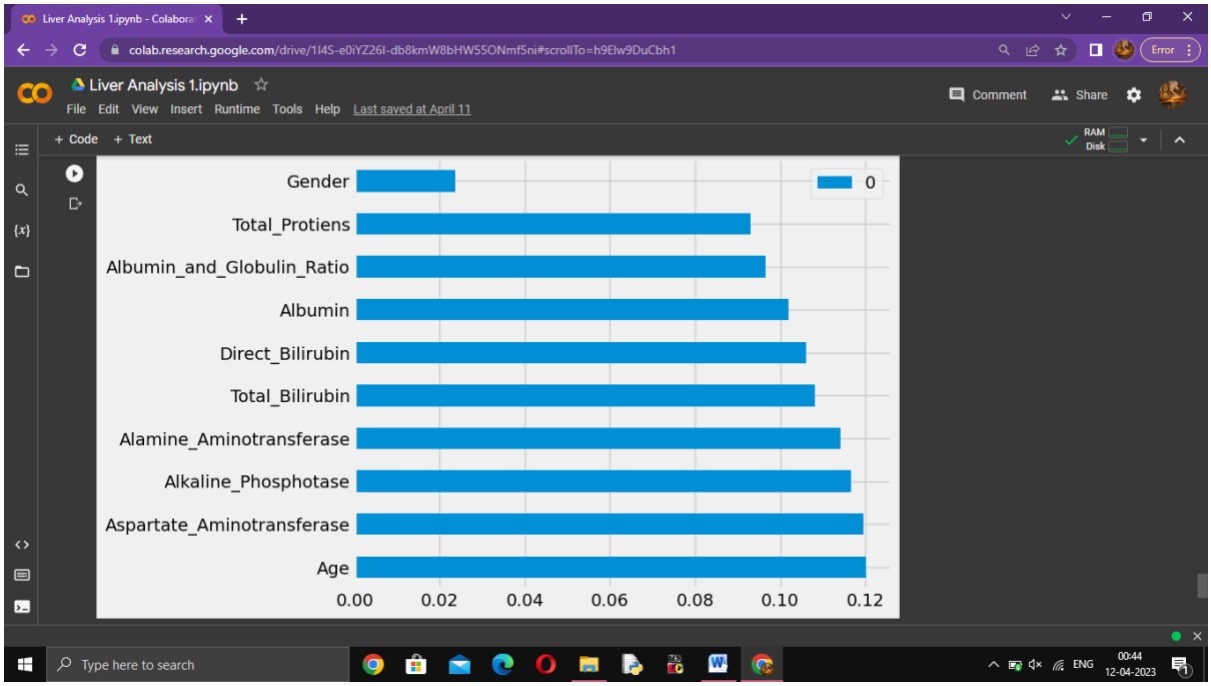
i)comparing the model







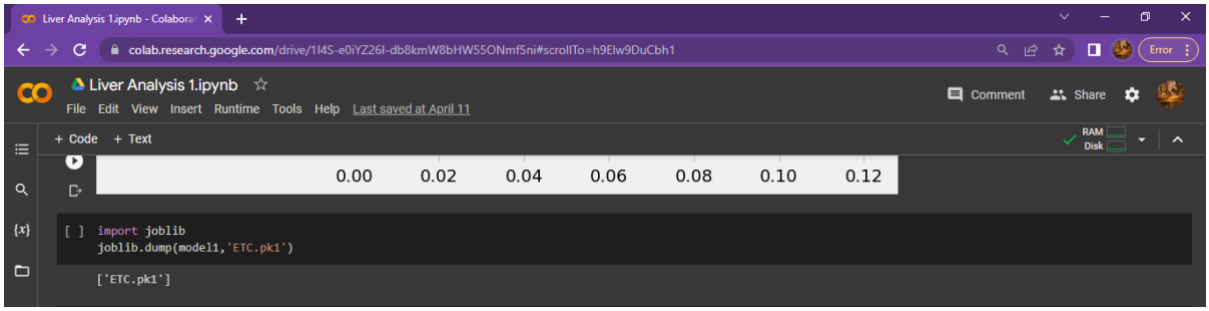
ii)Identifying important features



Model Deployment

i)Save the best model :

❖



Advantages:

1.No medical expertise required: You don’t need to have any knowledge of medical science and liver diseases to predict the liver disease using this application. All you need to do is enter the details being asked, which are already present in the blood test report(some like age, gender are already known) and then you will get the results of prediction.

2. High accuracy: The system predicts the results with 100 % accuracy for the dataset that we have used while creating this application. While the accuracy might be different in some cases, it will still be high enough to be trustworthy at a large scale.

3. Immediate results: The results here are predicted within seconds of entering the details. You don’t need to wait for a doctor to come, unlike in traditional method.

4.An early diagnosis of liver problems will increases patients survival rate liver failures are at risk among Indian

5. To performance classification of liver based diseases is further improved

6.Improving accurancy in diseae progession tracking:Machine learning algorthms can track disease progression and predict outcomes with greater accurancy and than additional methods allowing for more precise and personalized treatment plans.

7.Reduced health cost:By improving diagnosis accurancy and reducing the need for unnecessary tests and procedures,machine learning can help reduce healthcare costs and for patientsand providers

8.Improving effienciency in medical research:Machine learning can analyze vast amount of medical data quickly and accurately ,allowing researchers to identify patterns and insights that might have been missing using traditional research methods

9.Enhanced patient safety:Machine learning can help identify potential adverse events or complications associate with liver disease and treatment,and improving patient safety

Disadvantages:

1. Limited interpretability:Machine learning can be difficult to interpret,make it challenging for Medical professionals to understand how the algorithm arrived at its decision
2. Data quality:Machine learning algorithms rely on high quality data to make accurate predictions.If the data is incomplete or inaccurate,the algorithm predictions may also be unreliable
3. Privacy concerns:patient data used in machine learning algorithms must be protected to prevent unauthorized access or misuse
4. Bias:Machine learning can be biased if they are training on data that is not representative of the general population or if the data contains inherent biases
5. Need for large amounts of data:Machine learning algorithms required large amounts of data to train and produce accurate pretictions.obtaining this data can be consuming and costly.
6. Complex implementation process: Implementing machine learing in clinical settings can be challenging due to the need for integration with existing healthcare system and the need for specialized technical expertise.
7. Lack of standardization:there are currently no standardization and the development and implementation of machine learning algorithms in healthcare,which can lead to inconsistencies in result and patient care.
8. Ethical concerns:The use of machine learning algorithms in healthcare raises ethical concerns around data privacy,bias,and the potential for harm to patients

Applications:

1.Diagnosis and treatment of liver disease:Machine learning algorithm can analyze patient data to identify signs of liver disease and recommend personalized treatment plans based on the patient's medical history and risk factors

2.Early detections of liver cancer:Machine learning can identify early signs of liver cancer,allowing to earlier intervention and better patient outcomes

3.Monitoring disease progression:Machine learning algorithms can track disease progression over time,allowing doctors to monitor the effectiveness of treatments and adjust treatment plans as needed

4.predictive analytics:Machine learning can be used to predict furture outcomes and complications of liver disease,alllowing doctors to proactively address potential issues and improve patient outcomes

5.Drug development:Machine learnig algorithms can analyze large amounts of data from clinical trials and others sources to identify potential drug targets for liver disease

6.Medical imaging analysis:Machine learning can analysis medical imaging data to identify early of liver disease or liver cancer,allowing for earlier intervention and better

Conclusion

ML approach is widely used in analyzing clinical information, genetic records, and medical images of patients. Various studies have proved the unmatchable potential of data mining and ML tools in the medical domain. These tools can discover hidden significant predictive parameters from medical datasets that provide early prediction and diagnosis of diseases. In this study, we tried to find the research gap of the various researches that use ML concepts in liver disease diagnosis. Moreover, the future scope has also been mentioned regarding the same. From the above review, it is evident that ML techniques are highly promising in diagnosing liver diseases. But further data proving its validity and efficiency is required for its constant use by physicians.

Liver future scope

In this thesis the proposed system concludes that PSO feature selection methods for Indian Liver Patient Dataset. This thesis analyzed the liver disease using algorithms such as J48, MLP, SVM, Random Forest, and Bayesnet Classification. These algorithm gives various result based on PSO feature selection model .It has been seen that bayes net and J48 Classification gives better results compare to other classification algorithms. There are many criterions for evaluating the selected feature subset, here this thesis used features such as Total bilirubin, Direct_ bilirubin, Total_protiens, Albumin, A/G ratio, SGPT, SGOT, Alkphos to evaluate the performance of different classification algorithm. In future, we have attempted toclassify different feature selection algorithms into four groups: complete search, heuristic search, meta-heuristic methods and methods that use artificial neural network. PSO has been widely used for feature selection to improve liver classification performance. Further, a lot of work is being done using multi-objective PSO for feature selection to improve liver classification performance and to reduce number of features selected as well. Most of the existing multi-objective feature selection based on PSO algorithms use binary tournament selection to select gbest and uniform and non-uniform mutation.

There is a scope to further reduce search space for better liver classification accuracy if enhanced selection and mutation procedures are being used. The future methodology is used to analyze the liver region into separable compartments i.e. liver etc. However, the method requires further improvement mostly regarding feature selection of the liver into multiple components: renal cortex, renal column, renal medulla and renal pelvis. Apart from that, it is planned to expand the database on which the system will be tested. And also the proposed method in this thesis can be employed for detecting the heart diseases in future with the heart dataset and classification of the diseases.

It is observed that some approaches do not show efficient outcomes when used on a large dataset and classifies a sample of datasets with greater accuracy than the remaining dataset. Different methods have a varied range of efficiency and sensitivity. It is also seen that specific approaches do not show the same level of accuracy when applied to real-time data. Collecting large-scale medical data for research and selecting the most significant features are the challenging aspects of using ML concepts for liver disease detection. These problems can be solved by carrying out large scale studies on multiple centres with emphasizing the collection, segmentation, and pre-processing of data.

In the future it would help the doctors to easily detect the disease in a patient bodyandit could be developed further by using different datasets and also by using different algorithms. Through this project we have increased the efficiency of the prediction. We have increasedthe accuracy of the prediction algorithms where we have used different algorithms topredict the accuracy of the disease at different accuracy levels. This project can also be developedtodetect which kind of a liver disease a patient has and if it is conformed it can also be toldat which percentage it is in the patient's body. This can be taken further by usingvariousalgorithms and also lot of datasets. This can also be done using artificial intelligencetechniques using different tools.

APPENDIX

CODING

Source Code in Colab(.ipynb):

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import rcParams
from scipy import stats
import warnings
warnings.filterwarnings('ignore')
# Applying styles to notebook

plt.style.use('fivethirtyeight')
# import the dataset from specified location

df=pd.read_csv('/content/indian_liver_patient.csv')

# showing the data from top 5

df.head()
# finding the datatype

df.info()
# finding the null values

df.isnull().any()
#finding the sum of null values in the data

df.isnull().sum()
# Handling and filling the missing value


df=df.fillna(0)
df.isnull().sum()
#Encoding(manual encoding)

df['Gender']=df['Gender'].replace({'Female':0,'Male':1})
df.head()
"""
Types of Analysis
1. Univariate Analysis
2. Bivariate Analysis
```

3. Multivariate Analysis
4. Descriptive Analysis

```
# Descriptive Analysis
df.describe()

# Univariate analysis -Extracting info from a single column

#checking data distribution

sns.distplot(df['Albumin'])
#Splitting Dep & Indep variable

x=df.drop('Dataset',axis=1)
x.head()
y=df['Dataset']
y
# Bivariate
sns.barplot(x='Gender',y='Dataset',data=df)
#Multivariate

plt.figure(figsize=(10,7))
sns.heatmap(df.corr(),annot=True)
# Data preprocessing

# Finding the shape of the data

df.shape
# Finding outliers using boxplot for Total_Bilirubin,Direct_Bilirubin,Alkaline_Phosphot
ase,Alamine_Aminotransferase and Aspartate_Aminotransferase
plt.figure(figsize=(30,15))
plt.subplot(151)
sns.boxplot(df['Total_Bilirubin'])
plt.title('Total_Bilirubin')
plt.subplot(152)
sns.boxplot(df['Direct_Bilirubin'])
plt.title('Direct_Bilirubin')
plt.subplot(153)
sns.boxplot(df['Alkaline_Phosphotase'])
plt.title('Alkaline_Phosphotase')
plt.subplot(154)
sns.boxplot(df['Alamine_Aminotransferase'])
plt.title('Alamine_Aminotransferase')
plt.subplot(155)
sns.boxplot(df['Aspartate_Aminotransferase'])
plt.title('Aspartate_Aminotransferase')
# Finding the outliers for Total_Protiens,Albumin and Albumin_and_Globulin_Ratio
plt.figure(figsize=(30,15))
plt.subplot(131)
```

```
sns.boxplot(df['Total_Protiens'])
plt.title('Total_Protiens')
plt.subplot(132)
sns.boxplot(df['Albumin'])
plt.title('Albumin')
plt.subplot(133)
sns.boxplot(df['Albumin_and_Globulin_Ratio'])
plt.title('Albumin_and_Globulin_Ratio')
# Finding the count of outliers for Total_Bilirubin

# IQR =q3-q1.....,ub = q3+(1.5*IQR) , lb = q1-(1.5*IQR)

q1=np.quantile(df['Total_Bilirubin'],0.25)
q3=np.quantile(df['Total_Bilirubin'],0.75)

print('Q1 = {}'.format(q1))
print('Q3 = {}'.format(q3))

IQR = q3-q1

print('IQR value is {}'.format(IQR))

upperBound = q3+(1.5*IQR)
lowerBound = q1-(1.5*IQR)


print('The upperbound value is {} & the lower bound value is {}'.format(upperBound,lo
werBound))

print('Skwed data :',len(df[df['Total_Bilirubin']>upperBound]))
# Finding the count of outliers for Direct_Bilirubin

# IQR =q3-q1.....,ub = q3+(1.5*IQR) , lb = q1-(1.5*IQR)

q1=np.quantile(df['Direct_Bilirubin'],0.25)
q3=np.quantile(df['Direct_Bilirubin'],0.75)

print('Q1 = {}'.format(q1))
print('Q3 = {}'.format(q3))

IQR = q3-q1

print('IQR value is {}'.format(IQR))

upperBound = q3+(1.5*IQR)
lowerBound = q1-(1.5*IQR)

print('The upperbound value is {} & the lower bound value is {}'.format(upperBound,lo
werBound))
```

```
print('Skwed data :',len(df[df['Direct_Bilirubin']>upperBound]))
# Finding the count of outliers for Alkaline_Phosphotase

# IQR =q3-q1.....,ub = q3+(1.5*IQR) , lb = q1-(1.5*IQR)

q1=np.quantile(df['Alkaline_Phosphotase'],0.25)
q3=np.quantile(df['Alkaline_Phosphotase'],0.75)

print('Q1 = {}'.format(q1))
print('Q3 = {}'.format(q3))

IQR = q3-q1

print('IQR value is {}'.format(IQR))

upperBound = q3+(1.5*IQR)
lowerBound = q1-(1.5*IQR)

print('The upperbound value is {} & the lower bound value is {}'.format(upperBound,lo
werBound))

print('Skwed data :',len(df[df['Alkaline_Phosphotase']>upperBound]))
# Finding the count of outliers for Alamine_Aminotransferase

# IQR =q3-q1.....,ub = q3+(1.5*IQR) , lb = q1-(1.5*IQR)

q1=np.quantile(df['Alamine_Aminotransferase'],0.25)
q3=np.quantile(df['Alamine_Aminotransferase'],0.75)

print('Q1 = {}'.format(q1))
print('Q3 = {}'.format(q3))

IQR = q3-q1

print('IQR value is {}'.format(IQR))

upperBound = q3+(1.5*IQR)
lowerBound = q1-(1.5*IQR)

print('The upperbound value is {} & the lower bound value is {}'.format(upperBound,lo
werBound))

print('Skwed data :',len(df[df['Alamine_Aminotransferase']>upperBound]))
# Finding the count of outliers for Aspartate_Aminotransferase

# IQR =q3-q1.....,ub = q3+(1.5*IQR) , lb = q1-(1.5*IQR)

q1=np.quantile(df['Aspartate_Aminotransferase'],0.25)
q3=np.quantile(df['Aspartate_Aminotransferase'],0.75)
```

```
print('Q1 = {}'.format(q1))
print('Q3 = {}'.format(q3))

IQR = q3-q1

print('IQR value is {}'.format(IQR))

upperBound = q3+(1.5*IQR)
lowerBound = q1-(1.5*IQR)

print('The upperbound value is {} & the lower bound value is {}'.format(upperBound,lo
werBound))

print('Skwed data :',len(df[df['Aspartate_Aminotransferase']>upperBound]))
# Finding the count of outliers for Total_Protiens

# IQR =q3-q1.....,ub = q3+(1.5*IQR) , lb = q1-(1.5*IQR)

q1=np.quantile(df['Total_Protiens'],0.25)
q3=np.quantile(df['Total_Protiens'],0.75)

print('Q1 = {}'.format(q1))
print('Q3 = {}'.format(q3))

IQR = q3-q1

print('IQR value is {}'.format(IQR))

upperBound = q3+(1.5*IQR)
lowerBound = q1-(1.5*IQR)

print('The upperbound value is {} & the lower bound value is {}'.format(upperBound,lo
werBound))

print('Skwed data :',len(df[df['Total_Protiens']>upperBound]))
# Finding the count of outliers for Albumin_and_Globulin_Ratio

# IQR =q3-q1.....,ub = q3+(1.5*IQR) , lb = q1-(1.5*IQR)


q1=np.quantile(df['Albumin_and_Globulin_Ratio'],0.25)
q3=np.quantile(df['Albumin_and_Globulin_Ratio'],0.75)


print('Q1 = {}'.format(q1))
print('Q3 = {}'.format(q3))

IQR = q3-q1
```

```
print('IQR value is {}'.format(IQR))

upperBound = q3+(1.5*IQR)
lowerBound = q1-(1.5*IQR)

print('The upperbound value is {} & the lower bound value is {}'.format(upperBound,lo
werBound))

print('Skwed data :',len(df[df['Albumin_and_Globulin_Ratio']>upperBound]))
#function for handling outlier
def transform(variable):
  plt.figure(figsize=(14,6))
  plt.subplot(131)
  sns.distplot(variable)
  plt.subplot(132)
  stats.probplot(variable,plot=plt)
transform(np.log(df['Total_Bilirubin']))
transform(np.log(df['Direct_Bilirubin']))
transform(np.log(df['Alkaline_Phosphotase']))
transform(np.log(df['Alamine_Aminotransferase']))
transform(np.log(df['Aspartate_Aminotransferase']))

#scale
from sklearn.preprocessing import scale
X_scaled=pd.DataFrame(scale(x),columns=x.columns)
X_scaled.head()
#Splitting data into train and test
x=df.iloc[:,:-1]
y=df.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)
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()
#Random forest model
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
model1=RandomForestClassifier()
model1.fit(x_train_smote,y_train_smote)
y_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))
#Decision tree model
from sklearn.tree import DecisionTreeClassifier
model4=DecisionTreeClassifier()
```

```
model4.fit(x_train_smote,y_train_smote)
y_predict=model4.predict(x_test)
dct1=accuracy_score(y_test,y_predict)
dct1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test,y_predict))
#KNN model
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))
#Logistic Regression
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))
#Ann
import tensorflow.keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
#Initializing 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,epoch
s=100)
#Training the model
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]])
model2.predict([[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]])
model5.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
```

```
y_pred=(y_pred>0.5)
y_pred
def predict_exit(sample_value):
  #Convert list to numpy array
  sample_value=np.array(sample_value)
  #Reshape because sample_value contains only 1 record
  sample_value=sample_value.reshape(1,-1)
  #Feature scaling
  sample_value=scale(sample_value)
  return classifier.predict(sample_value)
#sample predict
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('Healthy')
#Testing model with multiple evaluation metrics
#compare the model
acc_smote=[['KNN classifier',knn1],['RandomForestClassifier',rfc1],['DecisionTreeCla
ssifier',dct1],['LogisticRegression',logi1]]
Liverpatient_pred=pd.DataFrame(acc_smote,columns=['classification models','accur
acy_score'])
Liverpatient_pred
plt.figure(figsize=(7,5))
plt.xticks(rotation=90)
plt.title('Classification models & accuracy scores after SMOTE',fontsize=18)
sns.barplot(x="classification models",y="accuracy_score",data=Liverpatient_pred,pal
ette="Set2")
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,asc
ending=False)
dd
dd.plot(kind='barh',figsize=(7,6))
plt.title("Feature Importance",fontsize=14)
import pickle
pickle.dump(model,open('Liver_analysis.pkl','wb'))
```

Creating Templates

1.Home.html

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<meta charset ="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Home</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<style>
 body
 {
 background-image:url('https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTwSxYjKzzE1OVK6CNCOOve4swwz1Fwd
P2IXg&usqp=CAU');
background-size:50%;
background-align:left;
background-size:cover;
 }
p.big
 {
 font-size:20px;
 line-height:1.8;

text-align:center;
```

```
 outline-style:;
color:black;}
 h1
{
color:black;
 }
</style>
</head>


<body>
<br>
<div class="container">
<div class="row">
<div class="col-md-12 bg-light text-right">
<a href="/home" class="btnbtn-info btn-lg">Home</a>
<a href="/predict" class="btnbtn-primary btn-lg">Predict</a>
</div>
</div>


<center>
<h1><strong>Liver Analysis</strong></h1><br>
<p class="big">The liver is a complex organ, but it is necessary to know its function within our body in order to take the required care and avoid suffering from liver diseases that affect our health and lifestyle. The liver is located under the diaphragm and crosses the abdominal cavity longitudinally. <br>
It weighs between 1.4 and 1.6 kilograms and measures about 10 centimeters.
This can vary according to the age and anatomy of the people. <br>
```

This organ is formed by a left lobe and a right lobe. The gallbladder is located at the level of the right lobe and acts as a reservoir for bile. In turn, it receives blood by two different routes: the hepatic artery, which supplies the blood coming from the heart, and the portal vein, which transports the blood sent from the intestine.

The hepatic veins are responsible for ensuring the evacuation of the blood. Liver function tests (also known as a liver panel) are blood tests that measure different enzymes, proteins, and other substances made by the liver.

These tests check the overall health of your liver.

</p>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</body>

</html>

2.Predict.html

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<meta charset ="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Home</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<style>
 body
 {
 background-
image:url("https://i.etsystatic.com/19980602/r/il/015454/3983272814/il_794xN.39832
72814_epj2.jpg");
 background-size:130%;
color:blue;
 }
 h1
{
color:blue;
  }
 h3.big
 {
 line-height:1.8;
 }
</style>
```

```
</head>
<body>
<br>
<div class="container">
<div class="row">
<div class="col-md-12 bg-light text-right">
<a href="/home" class="btnbtn-info btn-lg">Home</a>
<a href="/predict" class="btnbtn-primary btn-lg">Predict</a>
</div>
</div>
<br>

<h1><strong><center>Liver Patient analysis</center></strong></h1><br><br>
<h4>
<form action="/pred"  method="post">

<div class="form-group row">
<div class="col-md-3">
<label >Age</label>
<input type="text" class="form-control" name="Age" placeholder="Age">
</div>
</div><br>

<div class="form-group mb-3">
<div class="input-group-prepend">
<label  class="input-group-text" for="Gender">Gender</label>
</div>
```

```
<select class="custom-select" id="Gender" name="Gender">

<option value="1">Male</option>

<option value="2">Female</option>

</select>

</div><br>

<div class="form-group row">

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

<label >Total_bilirubin</label>

<input type="text" class="form-control" name="Total_bilirubin"
placeholder="Total_bilirubin">

</div>

</div><br>

<div class="form-group row">

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

<label >Direct_bilirubin</label>

<input type="text" class="form-control" name="Direct_bilirubin"
placeholder="Direct_bilirubin">

</div>

</div><br>

<div class="form-group row">

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

<label >Alkaline_Phosphotase</label>

<input type="text" class="form-control" name="Alkaline_Phosphotase"
placeholder="Alkaline_Phosphotase">

</div>

</div><br>

<div class="form-group row">

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

```
<label>Alamine_Aminotransferase</label>

<input type="text" class="form-control" name="Alamine_Aminotransferase"
placeholder="Alamine_Aminotransferase">

</div>

</div><br>

<div class="form-group row">

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

<label >Aspartate_Aminotransferase</label>

<input type="text" class="form-control" name="Aspartate_Aminotransferase"
placeholder="Aspartate_Aminotransferase">

</div>

</div><br>

<div class="form-group row">

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

<label >Total_protiens</label>

<input type="text" class="form-control" name="Total_protiens"
placeholder="Total_protiens">

</div>

</div><br>

<div class="form-group row">

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

<label >Albumin</label>

<input type="text" class="form-control" name="Albumin" placeholder="Albumin">

</div>

</div><br>

<div class="form-group row">

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

<label >Albumin_and_Globulin_Ratio</label>
```

```
<input type="text" class="form-control" name="Albumin_and_Globulin_Ratio"
placeholder="Albumin_and_Globulin_Ratio">

</div>

</div><br>


<div class="form-group row">

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

<button type="submit" class="btnbtn-success btn-lg">Submit</button>


</div></div>

</form>

<br>

</h4>

</div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</body>

</html>
```

3.Submit.html

```
<!DOCTYPE html>
<html lang=”en”>
<head>
<meta charset ="UTF-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Home</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<style>
 body
 {
 background-
image:url("https://i.etsystatic.com/19980602/r/il/015454/3983272814/il_794xN.39832
72814_epj2.jpg");
 background-size:130%;
color:blue;
 }
 h1
{
color:blue;
  }
 h3.big
 {
 line-height:1.8;
 }
</style>
```

```
</head>
<body>
<br>
<div class="container">
<div class="row">
<div class="col-md-12 bg-light text-right">
<a href="/home" class="btnbtn-info btn-lg">Home</a>
<a href="/predict" class="btnbtn-primary btn-lg">Predict</a>
</div>
</div>
<br>

<h1><strong><center>Liver Patient analysis</center></strong></h1><br><br>
<h4>
<form action="/pred"  method="post">

<div class="form-group row">
<div class="col-md-3">
<label >Age</label>
<input type="text" class="form-control" name="Age" placeholder="Age">
</div>
</div><br>

<div class="form-group mb-3">
<div class="input-group-prepend">
<label  class="input-group-text" for="Gender">Gender</label>
</div>
```

```
<select class="custom-select" id="Gender" name="Gender">

<option value="1">Male</option>

<option value="2">Female</option>

</select>

</div><br>

<div class="form-group row">

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

<label >Total_bilirubin</label>

<input type="text" class="form-control" name="Total_bilirubin"
placeholder="Total_bilirubin">

</div>

</div><br>


<div class="form-group row">

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

<label >Direct_bilirubin</label>

<input type="text" class="form-control" name="Direct_bilirubin"
placeholder="Direct_bilirubin">

</div>

</div><br>

<div class="form-group row">

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

<label >Alkaline_Phosphotase</label>

<input type="text" class="form-control" name="Alkaline_Phosphotase"
placeholder="Alkaline_Phosphotase">

</div>

</div><br>
```

```
<div class="form-group row">

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

<label>Alamine_Aminotransferase</label>

<input type="text" class="form-control" name="Alamine_Aminotransferase"
placeholder="Alamine_Aminotransferase">

</div>

</div><br>

<div class="form-group row">

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

<label >Aspartate_Aminotransferase</label>

<input type="text" class="form-control" name="Aspartate_Aminotransferase"
placeholder="Aspartate_Aminotransferase">

</div>

</div><br>

<div class="form-group row">

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

<label >Total_protiens</label>

<input type="text" class="form-control" name="Total_protiens"
placeholder="Total_protiens">

</div>

</div><br>

<div class="form-group row">

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

<label >Albumin</label>

<input type="text" class="form-control" name="Albumin" placeholder="Albumin">

</div>

</div><br>

<div class="form-group row">
```

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

<label >Albumin_and_Globulin_Ratio</label>

<input type="text" class="form-control" name="Albumin_and_Globulin_Ratio"
placeholder="Albumin_and_Globulin_Ratio">

</div>

</div><br>


<div class="form-group row">

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

<button type="submit" class="btnbtn-success btn-lg">Submit</button>


</div></div>

</form>

<br>

</h4>

</div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</body>

</html>
```

4.App.py

```
from flask import Flask, render_template, request
import pickle
import sklearn
import numpy as np
app = Flask(__name__)
model = pickle.load(open('Liver_analysis.pkl', 'rb'))
@app.route('/')
def about():
    return render_template('home.html')

@app.route('/home')
def home():
    return render_template('home.html')
@app.route('/predict')
def index():
    return render_template('predict.html')
@app.route('/pred', methods=['post'])
def pred():
    age = request.form['Age']
    gender = request.form['Gender']
    tb = request.form['Total_bilirubin']
db = request.form['Direct_bilirubin']
    ap = request.form['Alkaline_Phosphotase']
    aa1 = request.form['Alamine_Aminotransferase']
    aa2 = request.form['Aspartate_Aminotransferase']
tp = request.form['Total_protiens']
```

```
    a = request.form['Albumin']
agr = request.form['Albumin_and_Globulin_Ratio']
    data = [[float(age), float(gender), float(tb), float(db), float(ap), float(aa1), float(aa2), float(tp), float(a),float(agr)]]
    prediction = model.predict(data)


    if prediction == 1:


        return render_template('Submit.html', prediction_text='You have liver disease')
    else:


        return render_template('Submit.html', prediction_text='You dont have liver disease')


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

Video Demonstration Link: https://youtu.be/TBWwoqFCE6U