A REVIEW OF LIVER PATIENT ANALYSIS

TEAM ID:NM2023TMID19450
TEAM SIZE :4
TEAM LEADER:J.NARASIMMAN(20UCS2537)
TEAM MEMBERS:M.MASOODH(20UCS2535)
A.MATHAN(20UCS2536)
D.POOVARASAN(20UCS2539)

PROJECT REPORT

1. INTRODUCTION

Overview

A Review Of Liver Patient Analysis Methods Using Machine Learning

Project Description:

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

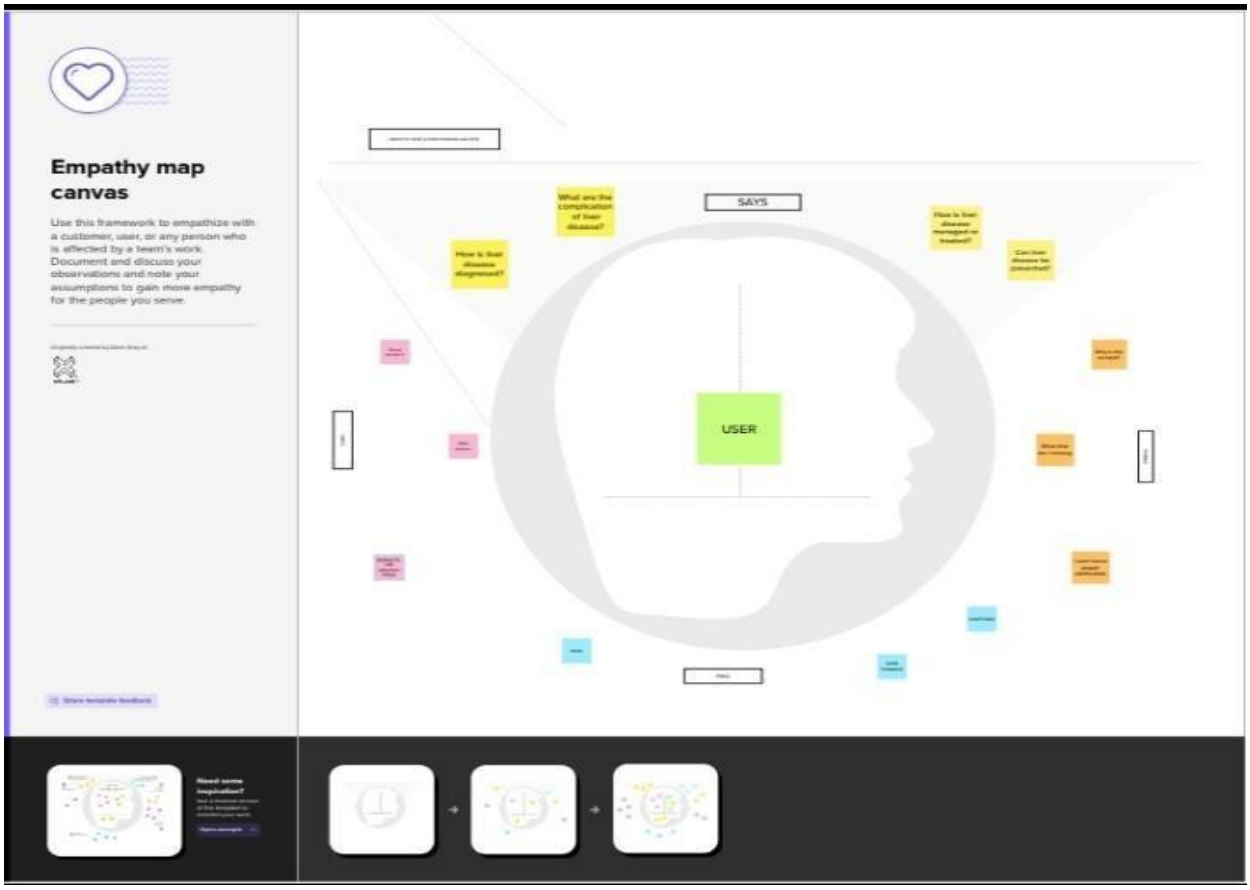
cost of liver disease diagnosis. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time. In this project we will analyse the parameters of various classification algorithms and compare 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.

PURPOSE

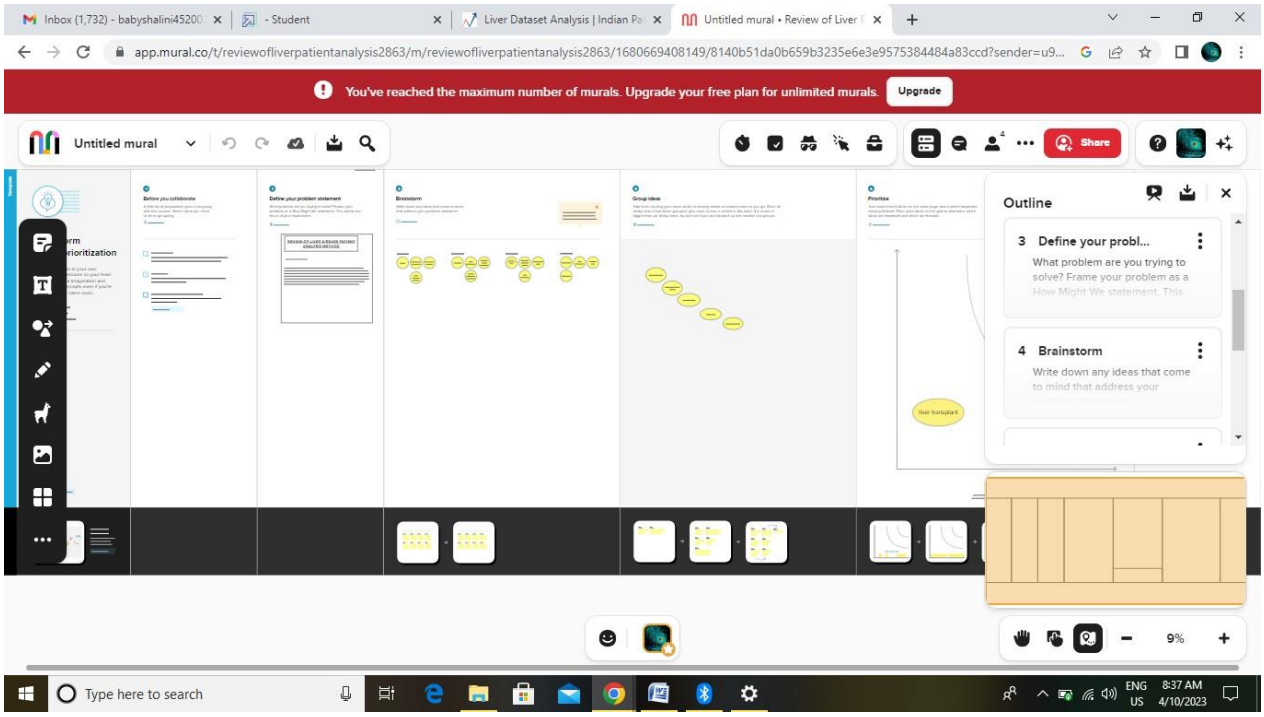
- Liver cirrhosis is the biggest health problem posed by alcohol use, with 1.4 lakh deaths every year.
- Sadly, no. In fact, it is getting more common in younger people than ever before. Dr. Amrish said that liver disease can set in childhood too as it can pass through genes.
- Cirrhosis isn't curable, but it's treatable. Alcohol abuse, hepatitis, and fatty liver disease are some of the main causes.

- Then you people will get answers like these as I mentioned above, So the purpose and inspiration of this project clearly simplifies the devastating answers from the data available with Google. We do need a system that in some stage reduces the burden on doctors, and today in this article I'll try to frame a practical logic that will help our healthcare system in a long run.

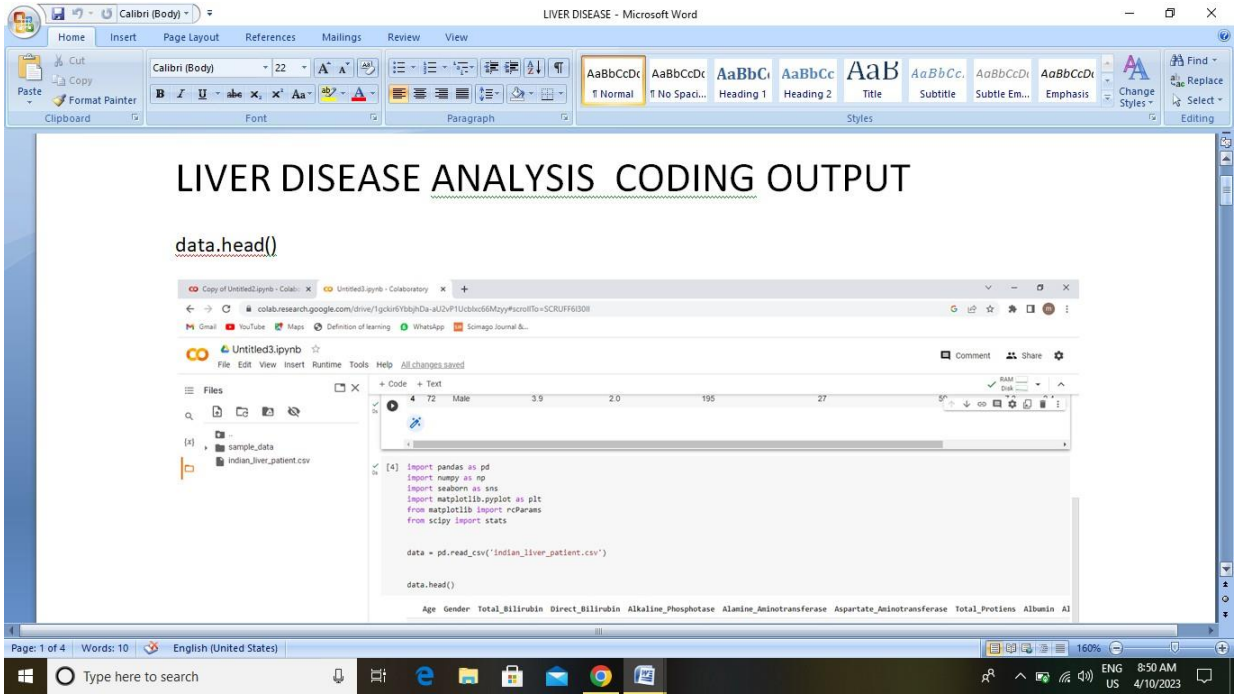
2. PROBLEM DEFINITION & DESIGN THINIKING



IDEATION & BRAINSTORMING MAP

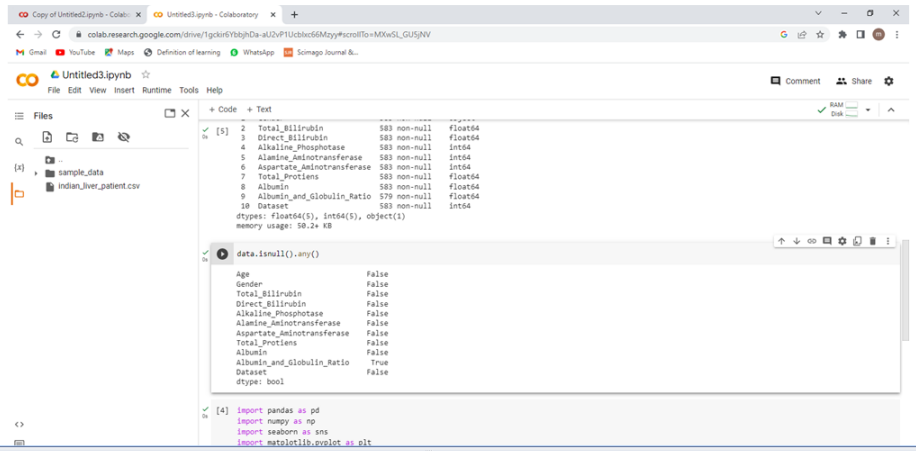

Untitled mural
Share
Outline
3 Define your probl...
What problem are you trying to solve? Frame your problem as a
4 Brainstorm
Write down any ideas that come to mind that address your
Type here to search
ENG US 8:37 AM 4/10/2023

3. RESULT


LIVER DISEASE - Microsoft Word
LIVER DISEASE ANALYSIS CODING OUTPUT
data.head()
Untitled3.ipynb
sample_data
indian_liver_patient.csv
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
data = pd.read_csv('indian_liver_patient.csv')
data.head()
Age Gender Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens Albumin
Page: 1 of 4 Words: 10 English (United States)
Type here to search
ENG US 8:50 AM 4/10/2023

LIVER DISEASE - Microsoft Word

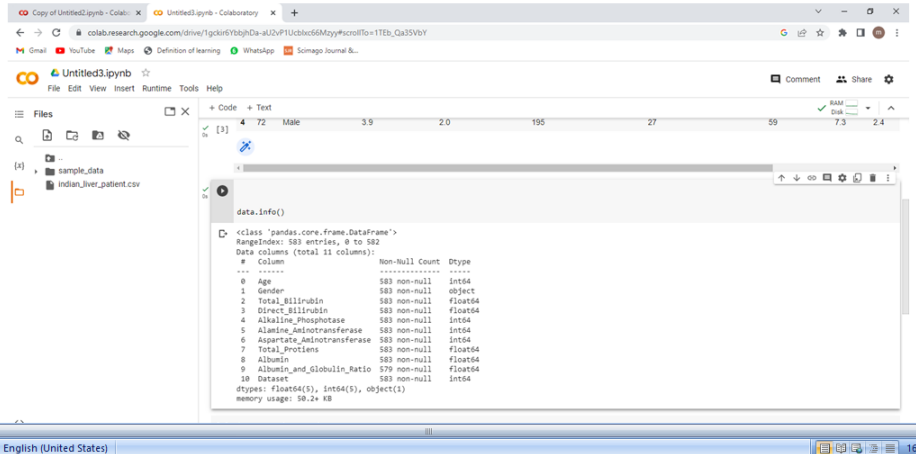
data.isnull().any()



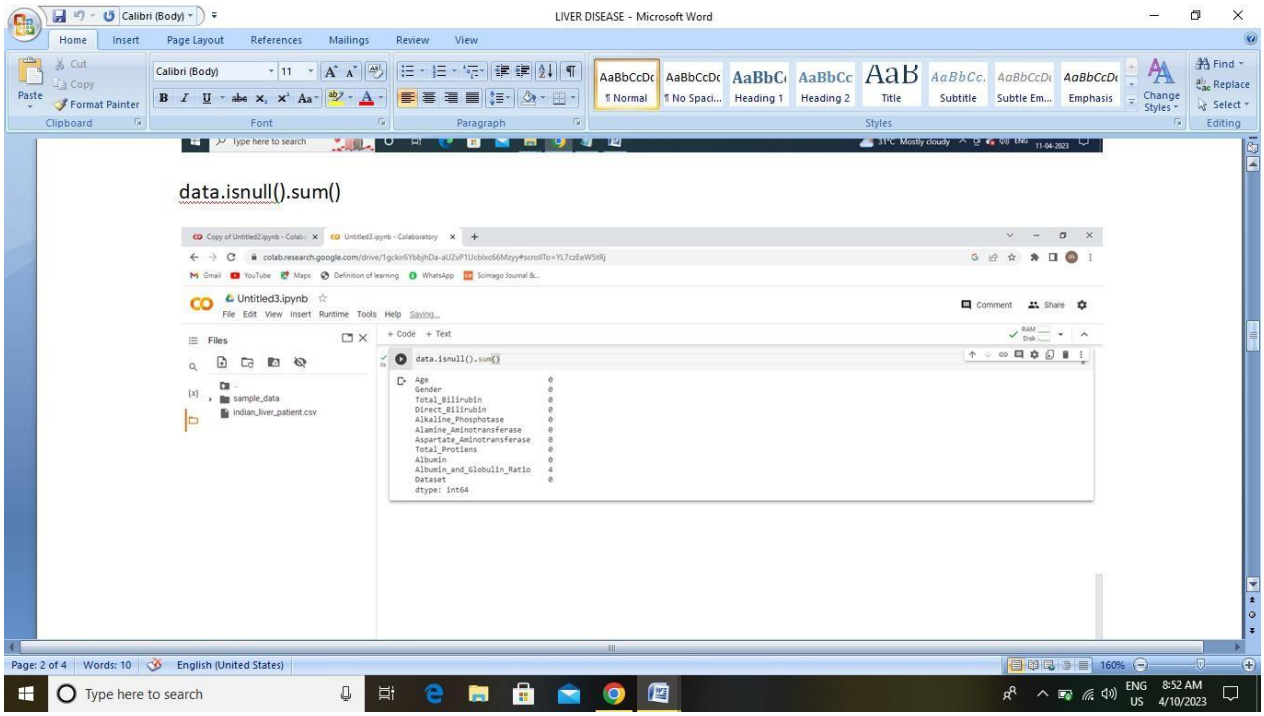
Page: 2 of 4 Words: 10 English (United States)

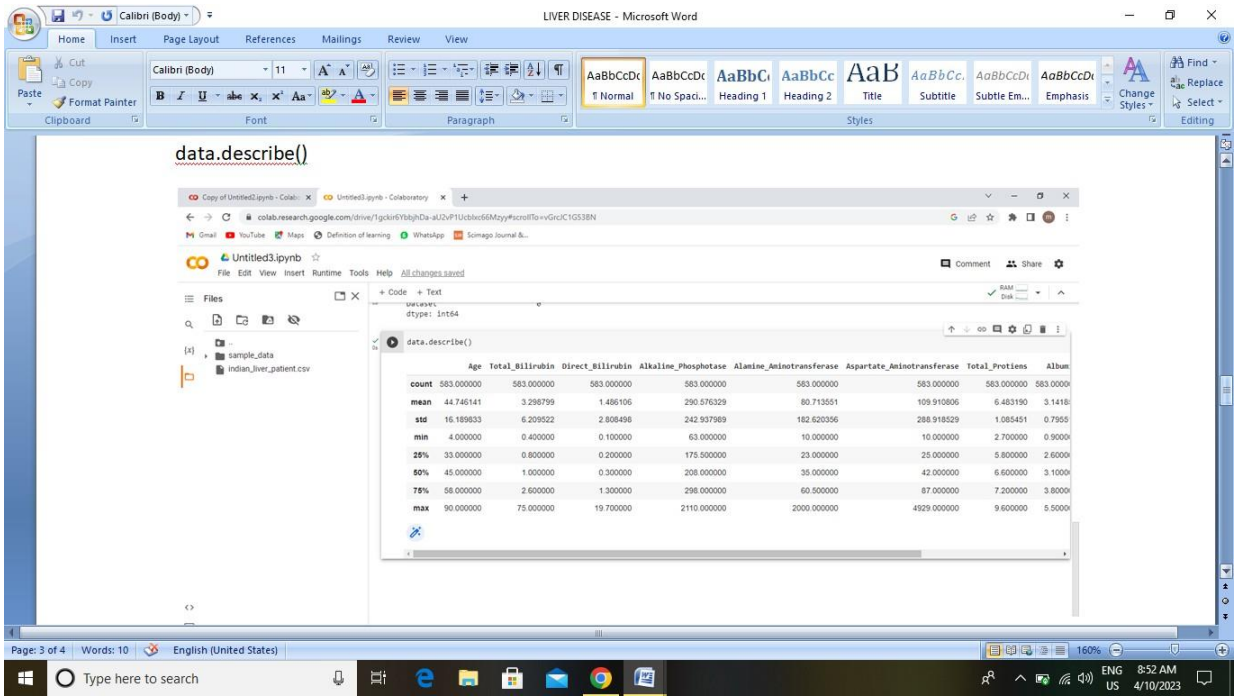
LIVER DISEASE - Microsoft Word

data.info()



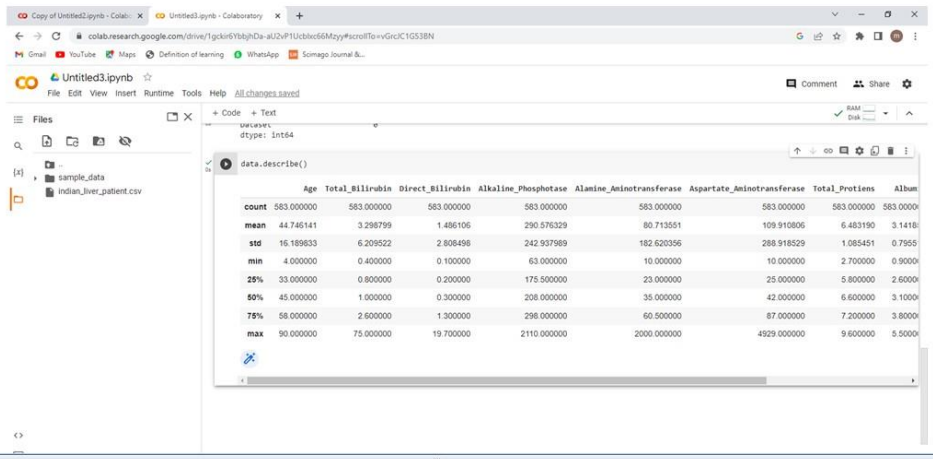
Page: 1 of 4 Words: 10 English (United States)


LIVER DISEASE - Microsoft Word
data.isnull().sum()
Untitled3.ipynb
Files
sample_data
indian_liver_patient.csv
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
Page: 2 of 4
Words: 10
English (United States)
Type here to search


LIVER DISEASE - Microsoft Word
data.describe()
Untitled3.ipynb
Files
sample_data
indian_liver_patient.csv
data.describe()
Age Total_Bilirubin Direct_Bilirubin Alkaline_Phosphotase Alamine_Aminotransferase Aspartate_Aminotransferase Total_Protiens
count 583.000000 583.000000 583.000000 583.000000 583.000000 583.000000 583.000000
mean 44.746141 3.298799 1.486106 290.576329 80.713551 109.910806 6.483190
std 16.189833 6.209522 2.808498 242.937989 182.620356 288.918529 1.085451
min 4.000000 0.400000 0.100000 63.000000 10.000000 10.000000 2.700000
25% 33.000000 0.800000 0.200000 175.500000 23.000000 25.000000 5.800000
50% 45.000000 1.000000 0.300000 208.000000 35.000000 42.000000 6.600000
75% 58.000000 2.600000 1.300000 298.000000 60.500000 87.000000 7.200000
max 90.000000 75.000000 19.700000 2110.000000 2000.000000 4929.000000 9.600000
Page: 3 of 4
Words: 10
English (United States)
Type here to search

LIVER DISEASE - Microsoft Word

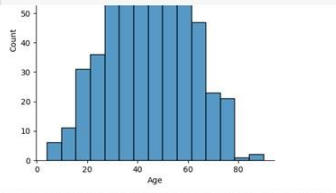
data.describe()



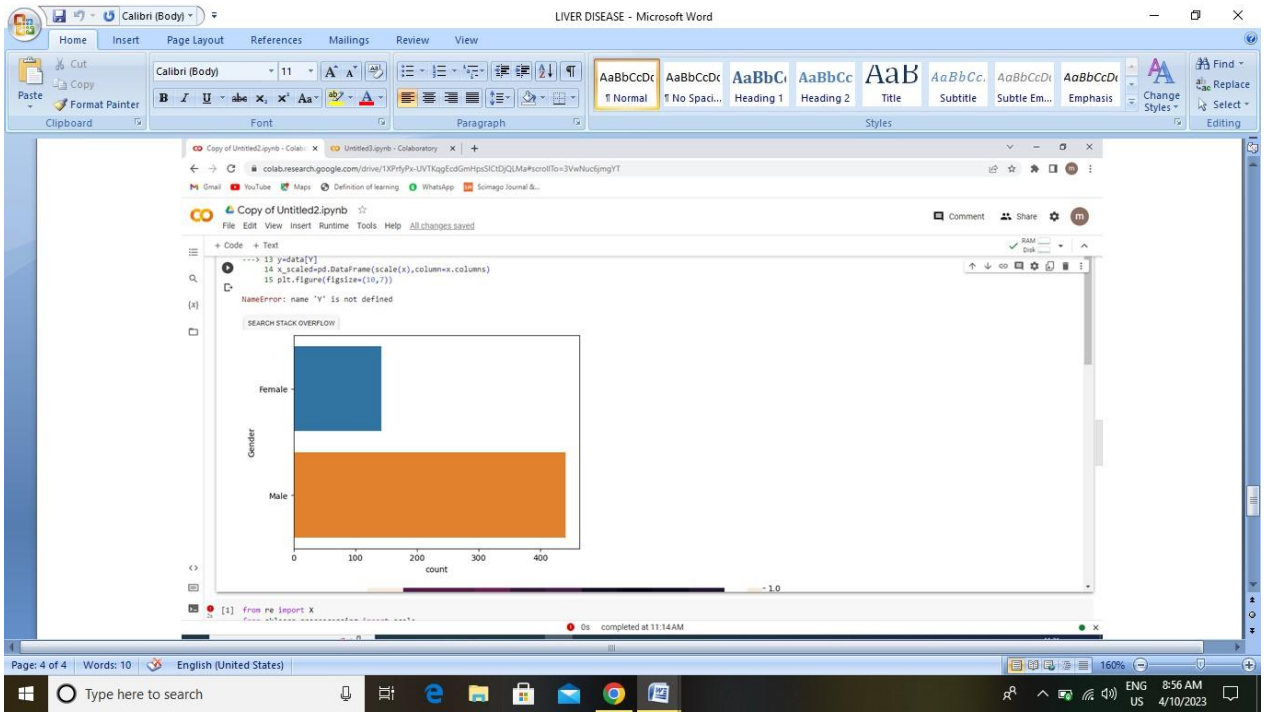
	Age	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albu
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000
mean	44.746141	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.1418
std	16.189833	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.7955
min	4.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.9000
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.6000
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.1000
75%	58.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.8000
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.5000

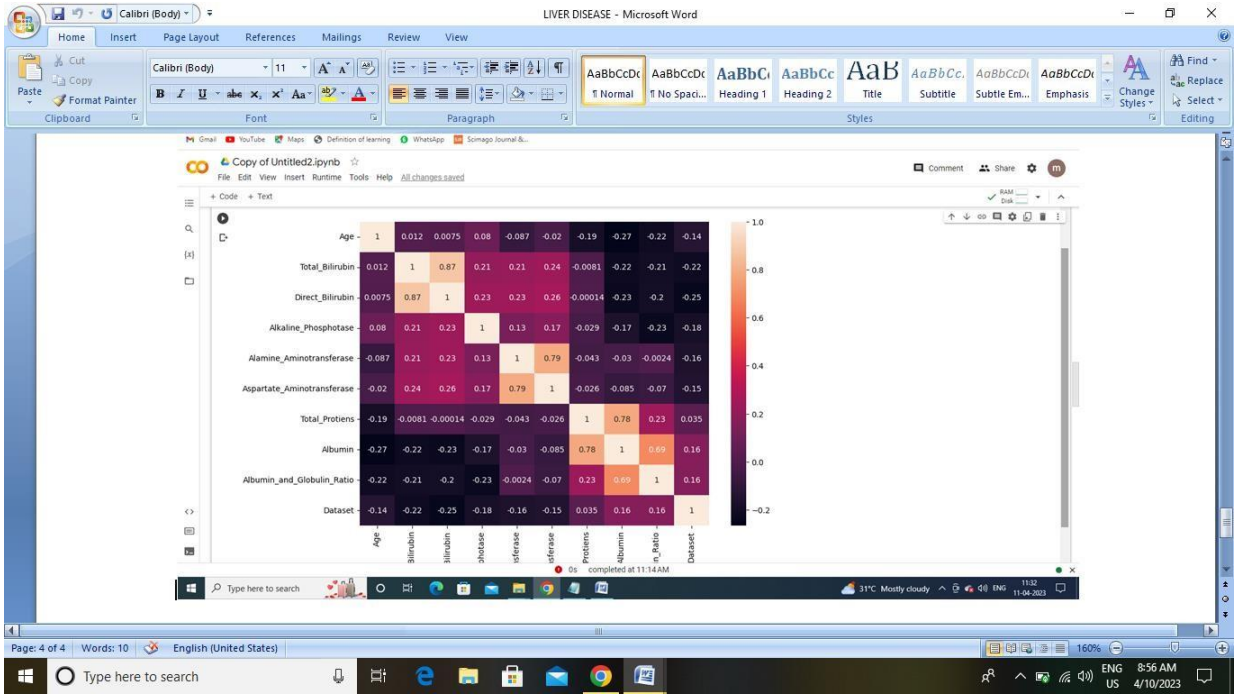
LIVER DISEASE - Microsoft Word

```
sns.heatmap(df.corr(),annot=True)
from sklearn.preprocessing import scale
x=data.iloc[:,:-1]
y=data[y]
x_scaled=pd.DataFrame(scale(x),column=x.columns)
plt.figure(figsize=(10,7))
sns.heatmap(df.corr(),annot=True)
```




```
NameError                                 Traceback (most recent call last)
<ipython-input-47-c2a82aa14360> in <cell line: 13>()
     11 from sklearn.preprocessing import scale
     12 x=data.iloc[:,:-1]
---> 13 y=data[Y]
```


LIVER DISEASE - Microsoft Word
Copy of Untitled2.ipynb
NameError: name 'Y' is not defined
Female
Male
Gender
count
0
100
200
300
400
completed at 11:14AM
Page: 4 of 4
Words: 10
English (United States)
Type here to search
8:56 AM
4/10/2023


LIVER DISEASE - Microsoft Word
Copy of Untitled2.ipynb
Age
Total_Bilirubin
Direct_Bilirubin
Alkaline_Phosphotase
Alamine_Aminotransferase
Aspartate_Aminotransferase
Total_Protiens
Albumin
Albumin_and_Globulin_Ratio
Dataset
completed at 11:14AM
Page: 4 of 4
Words: 10
English (United States)
Type here to search
8:56 AM
4/10/2023

4. ADVANTAGES & DISADVANTAGES

Liver biopsy	
Benefits	***Disadvantages***
Clear diagnostic criteria	Major invasive test
Diagnostic value confirmed	Complications include death
May suggest the etiology	Significant sampling errors
Can perform differential diagnosis	High cost
Assess the degree and stage of liver damage	Inter-observer variability
It can decide the therapy	

5. APPLICATIONS

The liver filters all of the blood in the body and breaks down poisonous substances, such as alcohol and drugs. The liver also produces bile, a fluid that helps digest fats and carry away waste.

- Hospitals.
- Specialty Clinics.
- Medical Research Department.

6. CONCLUSION

The main roles of the liver include removing toxins, processing food nutrients and regulating body metabolism. Important causes of liver disorders are fatty liver, hepatitis virus infections and alcohol. Cirrhosis (liver scarring), the end-result of many liver disorders, can lead to liver failure.

7.FUTURE SCOPE

- Hospitals.
- Specialty Clinics.
- Medical Research Department.
- Patient(Body)

8. APPENDIX

Source Code

```
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


data = pd.read_csv('indian_liver_patient.csv')


data.head()


data.info()


data.isnull().any()


data.isnull().sum()


data['Albumin_and_Globulin_Ratio'] =
data.fillna(data['Albumin_and_Globulin_Radio'].mode()[0])
data.isnull().sum()
```

```
from sklearn.preprocessing import LabelEncoder
lc = LabelEncoder()
data['gender']= lc.fit_transform(data['gender'])


data.describe()


sns.displot(data['age'])
plt.title('Age Distribution Graph')
plt.show()


sns.countplot(data['outcome'], hue=data['gender'])



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


from sklearn.proprocessing import scale
X_scaled=pd.DataFrame (scale(X), column=X.columns)



X_scaled.head()



x=data.iloc[:,:-1]
y=data.outcome



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_samplig 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.ensemble import RandomForestClassifier
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))
```

```
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))
```

```
import tensorflow.keras
from tensorflow.keras.models import sequential
from tensorflow.keras.layers import Dense



classifier = Sequential()


classifier.add(Dense(units=100, activation='relu', input_dim=10))


classifier.add(Dense(units=50, activation='relu'))


classifier.add(Dense(units=1, activation='sigmoid'))


classifier.compile(optimizer='adam', loss='binary_crossentopy', metrics=['accuracy'])


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.hs")

y_pred = classifier.predict(x_test)
y_pred
```

```
y_pred = (y_pred > 0.5)
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
```

```
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,palette = "Set2")
```

```
from sklearn.ensemble import ExtraTreesClassifier
model=ExtraTreesClassifier()
model.fit(x,y)
```

```
ExtraTreesClassifier()

model.feature_importances_
dd=pd.DataFrame(model.feature_importances_,index=X.columns).sort_values(0,ascendi
ng=False)
dd



dd.plot(kind='barch', figsize=(7,6))
plt.title("FEATURE IMPORTANCE",fontsize=14)

import joblib
joblib.dump(model1,  'ETC.pkl')



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():



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

```
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()
```