**A Review of Liver Patient Analysis Methods**

**Using**

**Machine Learning**

**CONCEPTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER** | **TITLE** | | **PAGE.NO.** |
| **1** | **INTRODUCTION**   * 1. **OVERVIEW**   2. **PURPOSE** | |  |
| **2** | **PROBLEM DEFINITION & DESIGN THINKING**  **2.1 EMPATHY MAP**  **2.2 IDEATION & BRAINSTOMING MAP** | |  |
| **3** | **RESULT** | |  |
| **4** | **ADVANTAGES & DISADVANTAGES** | |  |
| **5** | **APPLICATIONS** | |  |
| **6** | **CONCLUSION** | |  |
| **7** | **FUTURE SCOPE** |  | |
| **8** | **APPENDIX**  **8.1 SOURCE CODE** |  | |

**Abstract:**

In worldwide, Liver disease is a major health problem and diagnosing is crucial for

successful treatment. Liver disease is casued by Alcohocal, inhale polluted gas, drugs,contamination food and packing food pickle, so the medical expert system Hepatitis B,C virus(HBV,HCV) and mostly affected by Non-Alcoholic Steatohepatitis(NASH) otherwisecalled as Non-alcoholic fatty liver disease. This disease kills millions of people in the world. In this project we use the patient laboratory test results as the input which is retrieved from Git Hub. Git Hub is a codehosting platform for version control and collabration. The liver plays a very important role in life which supports the removal of toxins from the body.

Early prediction of liver disease is possible so that people can easily diagnosis the deadly disease in the early stage. This will give more useful in the Healthcare department and also a medical expert system can be used in a remote area. The liver plays a very important role in life which supports the removal of toxins from the body. So early prediction is very important to diagnosis the disease and recovers. Machine Learning is our present study mainly focused on the use of clinical data for liver disease prediction and explores different ways of representing such data through our analysis. The system can used the unsupervised algorithm to use for classification and analysis to updated with new data to improve the accuracy of the predictions over time. The Random Forests algorithm is used in this project which gives the High Accuracy, Sensitivity, Precision, Specificity and accurate result for analysis. This will give more useful in the Healthcare department and also a medical expert system can be used in a remote area. The proposed system provides an efficient and accurate

method for analyzing liver patient data. It has the potential to assist physicians in making timely and accurate diagnoses, leading to better patient outcomes.

**KEYWORDS:** Random Forests, Unsupervised algorithm, Machine Learning, Classification, Steatohepatitis (NASH), Fatty liver disease.

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW**

Liver disease is a significant health problem affecting millions of people worldwide. Early diagnosis and accurate prediction of disease progression are crucial for effective treatment and management of liver patients. Machine learning (ML) algorithms have shown great potential in assisting clinicians in the diagnosis and prognosis of liver diseases.

This review aims to provide an overview of the current state of research on the application of machine learning algorithms in the analysis of liver disease. The review will discuss the different ML techniques used in liver disease analysis, including their strengths and limitations. It will also highlight the challenges associated with the analysis of liver disease using ML, such as data availability and interpretability of the models.

The review will be of interest to clinicians, researchers, and practitioners working in the field of liver disease, as well as those interested in the application of machine learning algorithms in healthcare. By providing a comprehensive overview of the current state of research, the review will contribute to the development of more accurate and effective methods for the diagnosis and treatment of liver disease.

* 1. **PURPOSE**

Summarize the different ML techniques used in liver disease analysis, including their strengths and limitations.

Highlight the challenges associated with the analysis of liver disease using ML, such as data availability and interpretability of the models.

Discuss the potential applications of ML in the diagnosis, prognosis, and treatment of liver disease.

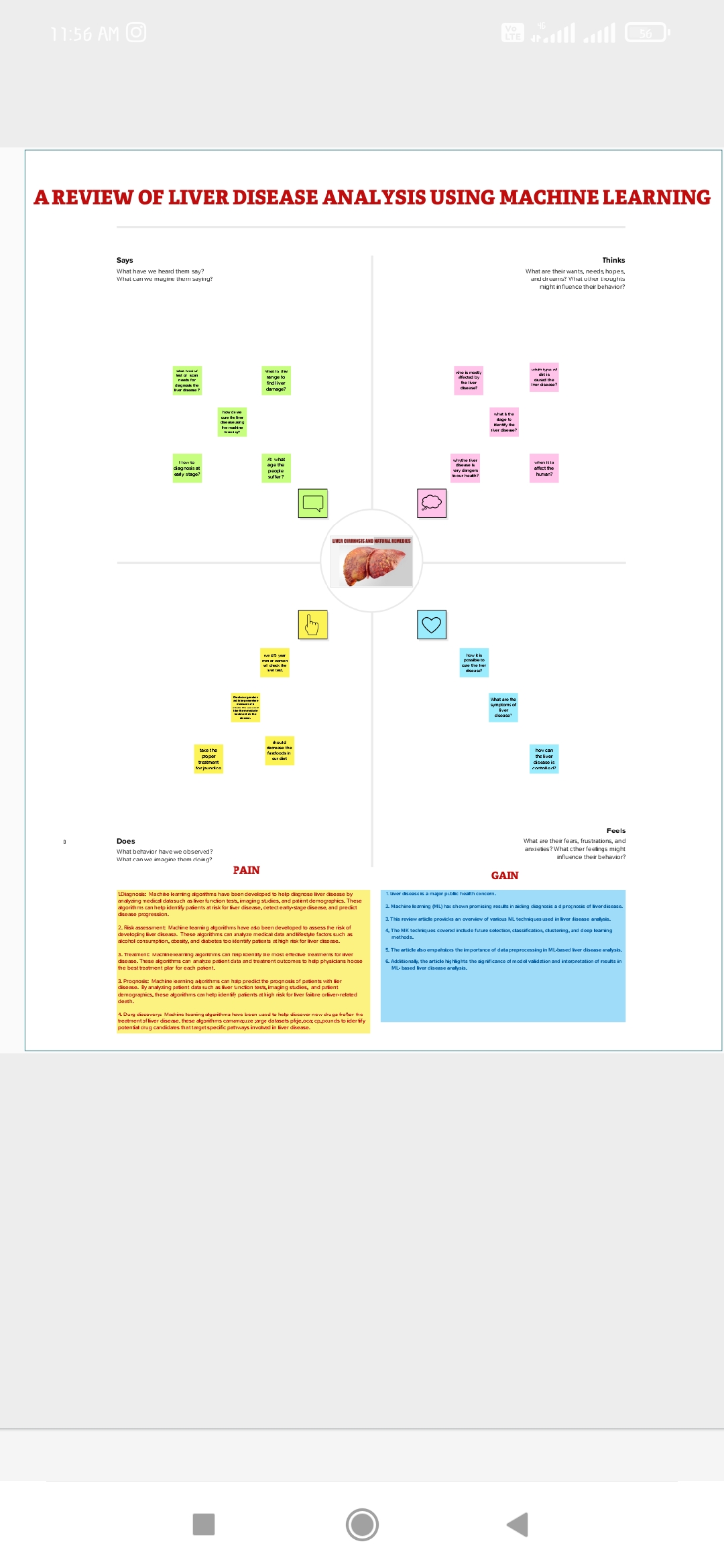
Provide insights into the future directions of research in this field.

The ultimate goal of this review is to contribute to the development of more accurate and effective methods for the diagnosis and treatment of liver disease using machine learning algorithms. By providing a comprehensive overview of the current state of research, the review aims to assist clinicians, researchers, and practitioners working in the field of liver disease in the development of new and innovative approaches to patient analysis and care.

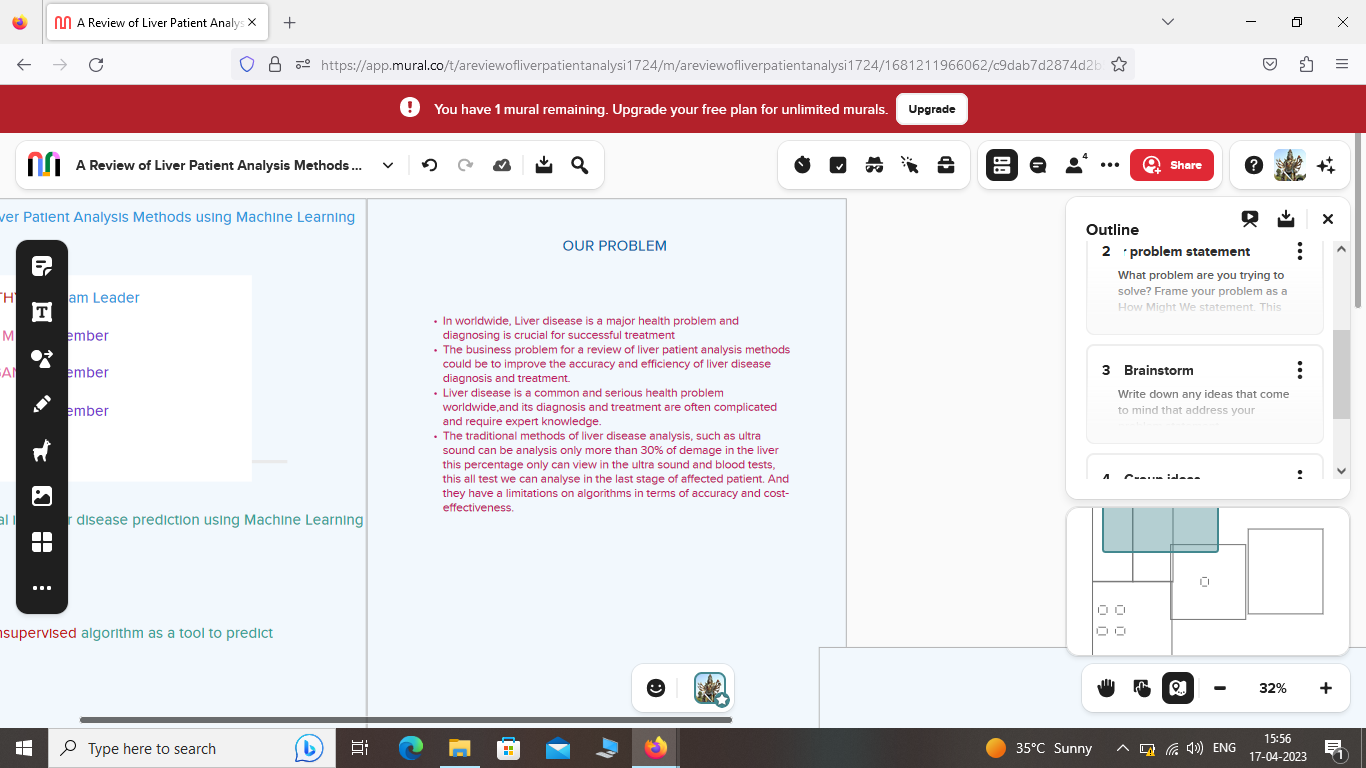
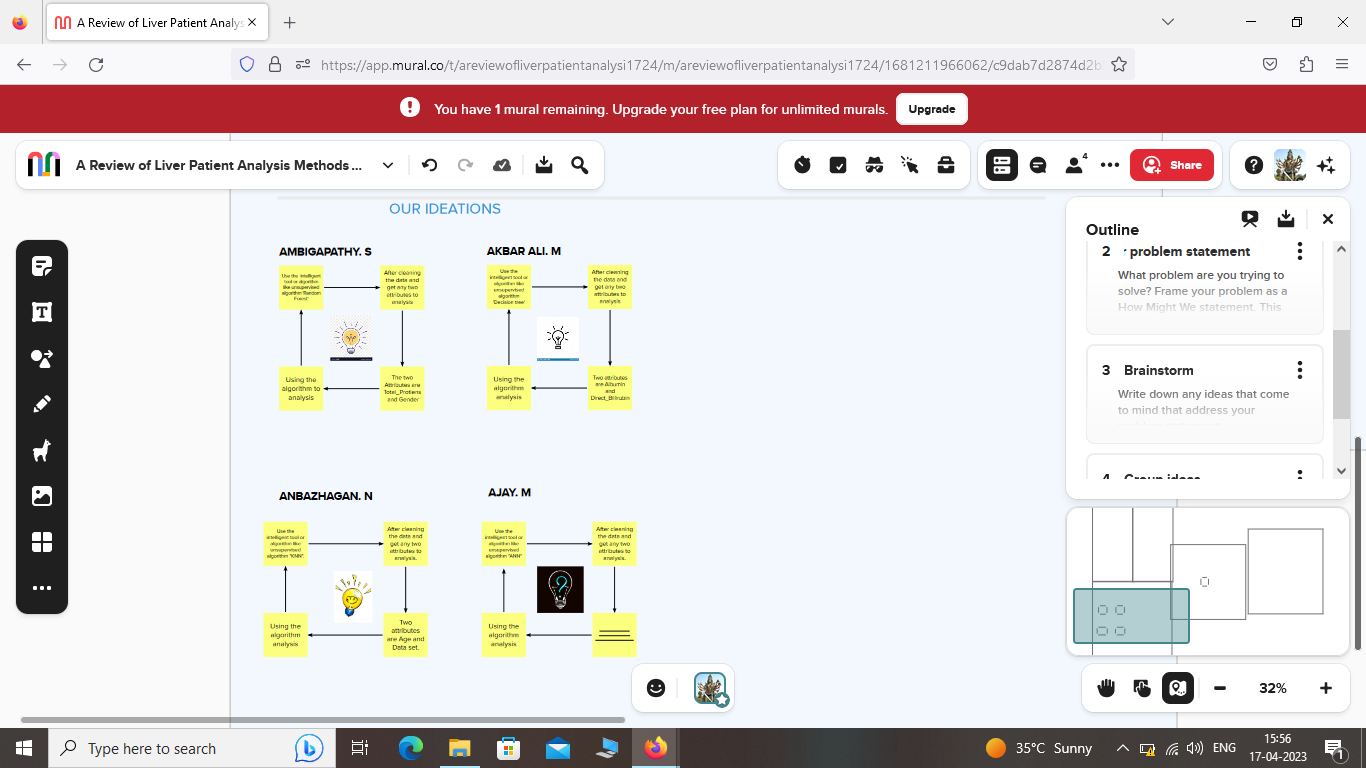
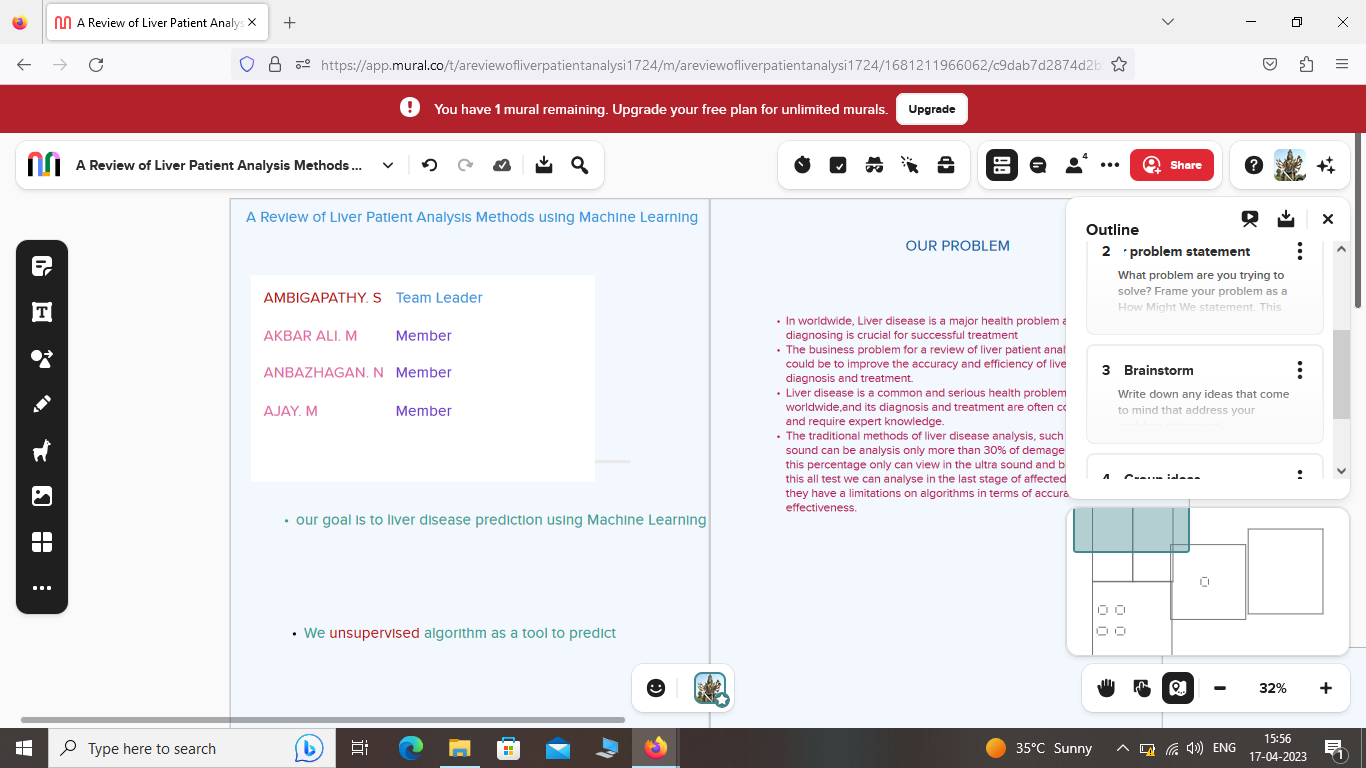
**CHAPTER 2**

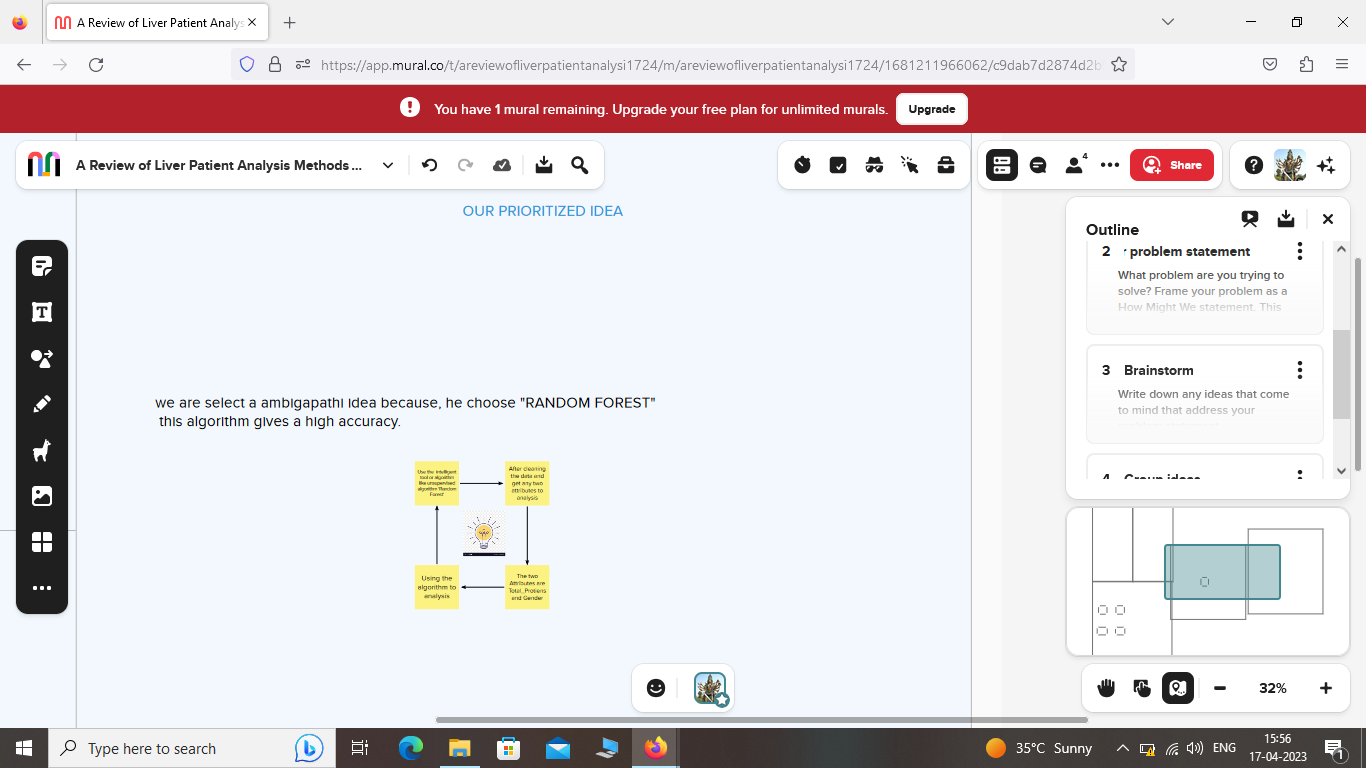
**PROBLEM DEFINITION & DESIGN THINKING**

**2.1 EMPATHY MAP**



**2.2 IDEATION & BRAINSTORMING MAP**

****

 **CHAPTER 3**

**RESULT :**

| **Age** | **Gender** | **Total\_Bilirubin** | **Direct\_Bilirubin** | **Alkaline\_Phosphotase** | **Alamine\_Aminotransferase** | **Aspartate\_Aminotransferase** | **Total\_Protiens** | **Albumin** | **Albumin\_and\_Globulin\_Ratio** | **Dataset** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 65 | Female | 0.7 | 0.1 | 187 | 16 | 18 | 6.8 | 3.3 | 0.90 | 1 |
| **1** | 62 | Male | 10.9 | 5.5 | 699 | 64 | 100 | 7.5 | 3.2 | 0.74 | 1 |
| **2** | 62 | Male | 7.3 | 4.1 | 490 | 60 | 68 | 7.0 | 3.3 | 0.89 | 1 |
| **3** | 58 | Male | 1.0 | 0.4 | 182 | 14 | 20 | 6.8 | 3.4 | 1.00 | 1 |
| **4** | 72 | Male | 3.9 | 2.0 | 195 | 27 | 59 | 7.3 | 2.4 | 0.40 | 1 |

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

RangeIndex: 583 entries, 0 to 582

Data columns (total 11 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Age 583 non-null int64

1 Gender 583 non-null object

2 Total\_Bilirubin 583 non-null float64

3 Direct\_Bilirubin 583 non-null float64

4 Alkaline\_Phosphotase 583 non-null int64

5 Alamine\_Aminotransferase 583 non-null int64

6 Aspartate\_Aminotransferase 583 non-null int64

7 Total\_Protiens 583 non-null float64

8 Albumin 583 non-null float64

9 Albumin\_and\_Globulin\_Ratio 579 non-null float64

10 Dataset 583 non-null int64

dtypes: float64(5), int64(5), object(1)

memory usage: 50.2+ KB

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

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

0 0.90

1 0.74

2 0.89

3 1.00

4 0.40

...

578 0.37

579 1.10

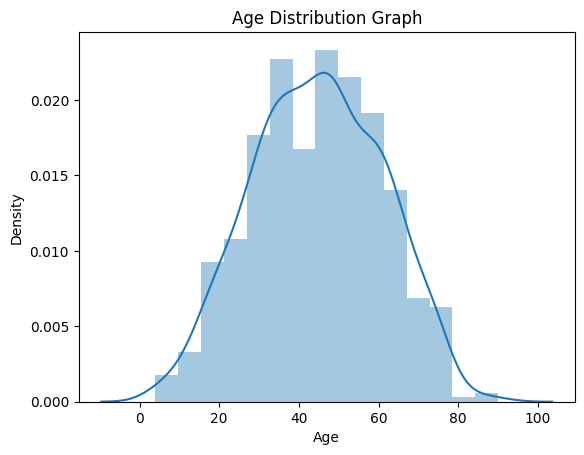
580 1.00

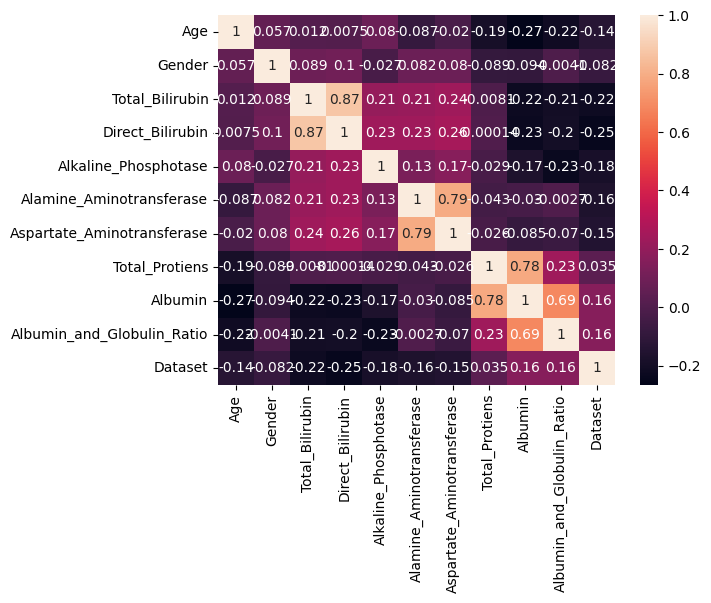
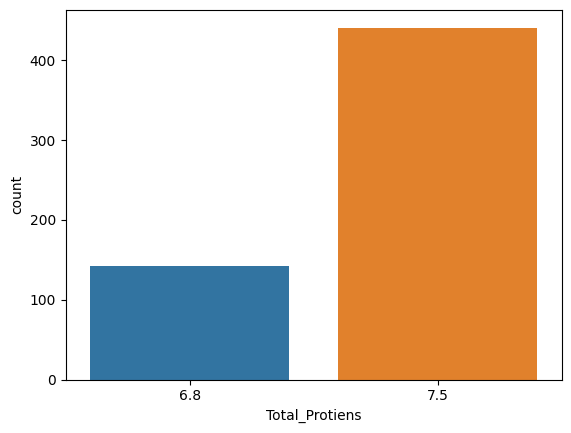
581 1.00

582 1.50

Name: Albumin\_and\_Globulin\_Ratio, Length: 583, dtype: float64

|  | **Age** | **Gender** | **Total\_Bilirubin** | **Direct\_Bilirubin** | **Alkaline\_Phosphotase** | **Alamine\_Aminotransferase** | **Aspartate\_Aminotransferase** | **Total\_Protiens** | **Albumin** | **Albumin\_and\_Globulin\_Ratio** | **Dataset** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 | 583.000000 |
| **mean** | 44.746141 | 0.756432 | 3.298799 | 1.486106 | 290.576329 | 80.713551 | 109.910806 | 6.483190 | 3.141852 | 0.947427 | 1.286449 |
| **std** | 16.189833 | 0.429603 | 6.209522 | 2.808498 | 242.937989 | 182.620356 | 288.918529 | 1.085451 | 0.795519 | 0.318522 | 0.452490 |
| **min** | 4.000000 | 0.000000 | 0.400000 | 0.100000 | 63.000000 | 10.000000 | 10.000000 | 2.700000 | 0.900000 | 0.300000 | 1.000000 |
| **25%** | 33.000000 | 1.000000 | 0.800000 | 0.200000 | 175.500000 | 23.000000 | 25.000000 | 5.800000 | 2.600000 | 0.700000 | 1.000000 |
| **50%** | 45.000000 | 1.000000 | 1.000000 | 0.300000 | 208.000000 | 35.000000 | 42.000000 | 6.600000 | 3.100000 | 0.950000 | 1.000000 |
| **75%** | 58.000000 | 1.000000 | 2.600000 | 1.300000 | 298.000000 | 60.500000 | 87.000000 | 7.200000 | 3.800000 | 1.100000 | 2.000000 |
| **max** | 90.000000 | 1.000000 | 75.000000 | 19.700000 | 2110.000000 | 2000.000000 | 4929.000000 | 9.600000 | 5.500000 | 2.800000 | 2.000000 |





|  | **Age** | **Gender** | **Total\_Bilirubin** | **Direct\_Bilirubin** | **Alkaline\_Phosphotase** | **Alamine\_Aminotransferase** | **Aspartate\_Aminotransferase** | **Total\_Protiens** | **Albumin** | **Albumin\_and\_Globulin\_Ratio** | **Dataset** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1.252098 | -1.762281 | -0.418878 | -0.493964 | -0.426715 | -0.354665 | -0.318393 | 0.292120 | 0.198969 | -0.149025 | -0.633595 |
| **1** | 1.066637 | 0.567446 | 1.225171 | 1.430423 | 1.682629 | -0.091599 | -0.034333 | 0.937566 | 0.073157 | -0.651777 | -0.633595 |
| **2** | 1.066637 | 0.567446 | 0.644919 | 0.931508 | 0.821588 | -0.113522 | -0.145186 | 0.476533 | 0.198969 | -0.180447 | -0.633595 |
| **3** | 0.819356 | 0.567446 | -0.370523 | -0.387054 | -0.447314 | -0.365626 | -0.311465 | 0.292120 | 0.324781 | 0.165194 | -0.633595 |
| **4** | 1.684839 | 0.567446 | 0.096902 | 0.183135 | -0.393756 | -0.294379 | -0.176363 | 0.753153 | -0.933340 | -1.720124 | -0.633595 |

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>

Collecting imblearn

Downloading imblearn-0.0-py2.py3-none-any.whl (1.9 kB)

Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.9/dist-packages (from imblearn) (0.10.1)

Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.2.0)

Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.22.4)

Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.2.2)

Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.10.1)

Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (3.1.0)

Installing collected packages: imblearn

Successfully installed imblearn-0.0

1 329

2 137

Name: Dataset, dtype: int64

1 329

2 329

Name: Dataset, dtype: int64

precision recall f1-score support

1 1.00 1.00 1.00 87

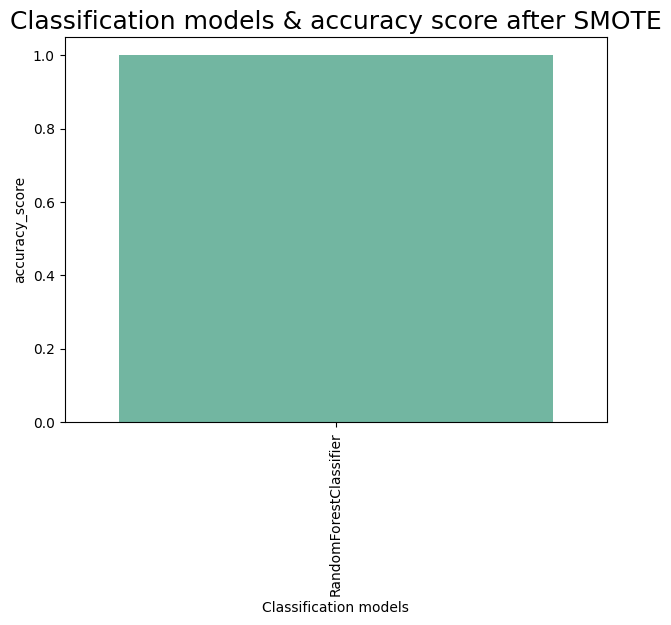
2 1.00 1.00 1.00 30

accuracy 1.00 117

macro avg 1.00 1.00 1.00 117

weighted avg 1.00 1.00 1.00 117

|  | **Classification models** | **accuracy\_score** |
| --- | --- | --- |
| **0** | RandomForestClassifier | 1.0 |

****

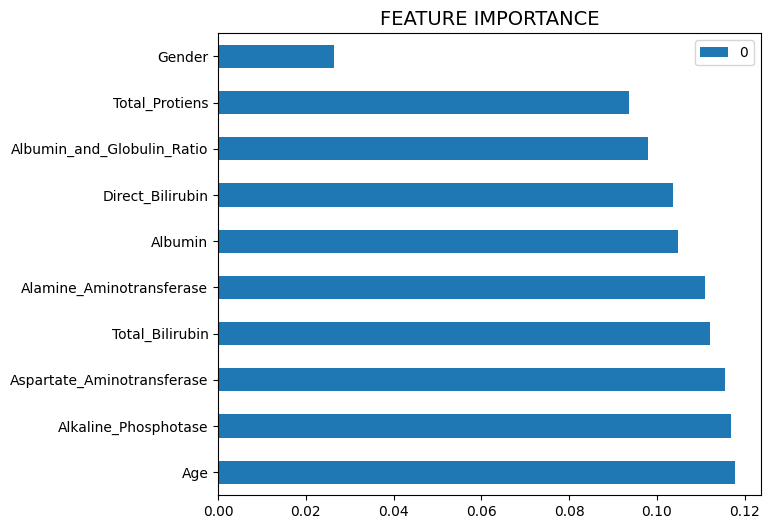
ExtraTreesClassifier

ExtraTreesClassifier()

array([0.11777117, 0.02645932, 0.11215644, 0.103584 , 0.11695258,

0.11107897, 0.11545029, 0.09372862, 0.10488599, 0.09793263])

|  | **0** |
| --- | --- |
| **Age** | 0.117771 |
| **Alkaline\_Phosphotase** | 0.116953 |
| **Aspartate\_Aminotransferase** | 0.115450 |
| **Total\_Bilirubin** | 0.112156 |
| **Alamine\_Aminotransferase** | 0.111079 |
| **Albumin** | 0.104886 |
| **Direct\_Bilirubin** | 0.103584 |
| **Albumin\_and\_Globulin\_Ratio** | 0.097933 |
| **Total\_Protiens** | 0.093729 |
| **Gender** | 0.026459 |

****

['ETC.pk1']

**CHAPTER 4**

**ADVANTAGES & DISADVANTAGES**

**ADVANTAGES:**

Improved accuracy: Machine learning algorithms can improve the accuracy of liver disease diagnosis and analysis, allowing for earlier detection and better treatment outcomes.

Faster analysis: Machine learning algorithms can analyze large amounts of patient data quickly and accurately, reducing the time it takes for doctors to make a diagnosis and determine the best treatment options.

Customized treatment plans: Machine learning algorithms can analyze patient data to create personalized treatment plans based on a patient's unique medical history and risk factors.

Better resource allocation: Machine learning algorithms can help healthcare providers allocate resources more efficiently by identifying patients who are at higher risk of liver disease and targeting interventions accordingly.

**DISADVANTAGE:**

Need for high-quality data: Machine learning algorithms require high-quality, accurate data to function effectively. If the data used for analysis is incomplete or inaccurate, the algorithm may produce incorrect results.

Limited generalizability: Machine learning algorithms are trained on specific datasets and may not be generalizable to other patient populations. This can limit their usefulness in certain contexts.

Limited interpretability: Machine learning algorithms can produce accurate results, but they may not be easily interpretable by clinicians who are unfamiliar with the underlying statistical models.

Ethical concerns: Machine learning algorithms may be used to predict a patient's risk of developing liver disease or other medical conditions, raising ethical concerns about privacy and potential discrimination against certain patient populations.

**CHAPTER 5**

**APPLICATIONS**

Machine learning algorithms can be applied to various aspects of liver patient analysis, including:

Early detection of liver disease: Machine learning algorithms can analyze patient data to identify early signs of liver disease, allowing for earlier interventions and better treatment outcomes.

Risk assessment: Machine learning algorithms can predict a patient's risk of developing liver disease based on their medical history, lifestyle factors, and other risk factors.

Treatment optimization: Machine learning algorithms can analyze patient data to determine the most effective treatment plan for a patient based on their unique medical history and risk factors.

Monitoring disease progression: Machine learning algorithms can monitor a patient's liver function over time to identify changes in disease progression and adjust treatment plans accordingly.

Medical imaging analysis: Machine learning algorithms can analyze medical images such as CT scans and MRI to identify liver lesions and other abnormalities.

Drug toxicity prediction: Machine learning algorithms can predict the risk of drug toxicity in patients with liver disease, allowing for more personalized dosing and improved treatment outcomes.

Clinical decision support: Machine learning algorithms can provide clinicians with decision support tools based on patient data to aid in diagnosis, treatment planning, and monitoring.

**CHAPTER 6**

**CONCLUSION**

The review of liver patient analysis methods using machine learning suggests that machine learning algorithms can improve the accuracy and speed of liver disease diagnosis and analysis, allowing for earlier detection, more personalized treatment plans, and better treatment outcomes. Machine learning algorithms can be applied to various aspects of liver patient analysis, including early detection, risk assessment, treatment optimization, monitoring disease progression, medical imaging analysis, drug toxicity prediction, and clinical decision support. However, the effectiveness of machine learning algorithms depends on the quality of the data used for analysis and the ability of clinicians to interpret the results. Therefore, future research should focus on improving the quality and accuracy of data used for machine learning analysis and developing interpretable machine learning models that can be easily understood and used by clinicians. Overall, machine learning has the potential to revolutionize the field of liver patient analysis and improve patient outcomes.

**CHAPTER 7**

**FUTURE SCOPE:**

The future scope of review of liver patient analysis methods using machine learning is promising, with several potential areas for future research and development.

Integration with Electronic Health Records (EHR): Machine learning algorithms can be integrated with EHR systems to provide real-time analysis and decision support tools to clinicians, improving the efficiency and quality of care.

Multi-modal data analysis: Machine learning algorithms can be trained on multiple types of data, including genetic, imaging, and clinical data, to develop more comprehensive models for liver disease diagnosis and analysis.

Transfer learning: Transfer learning techniques can be used to adapt pre-trained machine learning models to new datasets, reducing the need for large amounts of labeled data and accelerating the development of new models.

Interpretable machine learning: Developing interpretable machine learning models that can be easily understood and used by clinicians is an important area for future research, as it will increase the adoption and effectiveness of machine learning in clinical practice.

Patient stratification: Machine learning algorithms can be used to stratify patients based on their risk of developing liver disease, allowing for more targeted interventions and better outcomes.

Explainable AI: Developing explainable AI models that provide clinicians with insights into how the model arrived at its decision, increasing trust and transparency of the models.

Overall, the future scope of review of liver patient analysis methods using machine learning is vast, with opportunities for innovation and development in a variety of areas that can ultimately lead to improved patient outcomes

**CHAPTER 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('/content/indian\_liver\_patient.csv')

data.head()

data.info()

data.isnull().any()

data.isnull().sum()

data['Albumin\_and\_Globulin\_Ratio']

data['Albumin\_and\_Globulin\_Ratio'] = data['Albumin\_and\_Globulin\_Ratio']

data['Albumin\_and\_Globulin\_Ratio']= data['Albumin\_and\_Globulin\_Ratio'].fillna(data['Albumin\_and\_Globulin\_Ratio'].mode()[0])

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['Total\_Protiens'], x = data['Gender'])

df = data[['Age','Gender','Total\_Bilirubin','Direct\_Bilirubin','Alkaline\_Phosphotase','Alamine\_Aminotransferase','Aspartate\_Aminotransferase','Total\_Protiens','Albumin','Albumin\_and\_Globulin\_Ratio','Dataset']]

sns.heatmap(df.corr(), annot=True)

from sklearn.preprocessing import scale

X= data

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, radom\_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 accuracy\_score

from sklearn.metrics import classification\_report

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))

acc\_smote = [['RandomForestClassifier', rfc1]]

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 score 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)

model.feature\_importances\_

dd=pd.DataFrame(model.feature\_importances\_, index=X.columns).sort\_values(0,ascending=False)

dd

dd.plot(kind='barh', figsize=(7,6))

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

import joblib

joblib.dump(model1, 'ETC.pk1')