GOVERNMENT OF TAMILNADU

Naan Mudhalvan - Project-Based Experiential Learning

A Review of Liver Patient Analysis Methods Using Machine Learning

Submitted by

Team ID : NM2023TMID22017

AAFRIN. A (20326ER001)

ABINAYA. K (20326ER002)

BOOMIKA. A (20326ER003)

DEEPA. K (20326ER004)

Under the guidance of

Mrs. J. SUKANYA, MCA., M.Phil.,

Assistant Professor

PG and Research Department of Computer Science

M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR WOMEN

(Affiliated to Mother Teresa Women's University, Kodaikanal)
Reaccredited with 'A' Grade by NAAC
DINDIGUL-624001.

APRIL - 2023

M.V. MUTHIAH GOVERNMENT ARTS COLLEG FOR WOMEN

(Affiliated to Mother Teresa Women's University, Kodaikanal)
Reaccredited with 'A' Grade by NAAC
Dindigul - 624 001

PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of the project entitled, **A REVIEW OF LIVER PATIENT ANALYSIS METHODS USING MACHINE LEARNING** done by **AAFRIN. A (20326ER001), ABINAYA. K (20326ER002), BOOMIKA. A (20326ER003), DEEPA. K (20326ER004)**. This is submitted in partial fulfillment for the award of the degree of **Bachelor of Science in Computer Science in M.V.MUTHIAH GOVERNMENT ARTS COLLEGE FOR WOMEN,DINDIGUL** during the period of December 2022 to April 2023.

J.Sukany

Project Mentor(s)

Head of the Department

Submitted for Viva-Voce Examination held on <u>11.04.2023</u>

TABLE OF CONTENTS

CHAPTER - 1

INTRODUCTION

1.1 OVERVIEW

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, Decision Tree, 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.

1.2 PURPOSE

The Purpose of this Project includes,

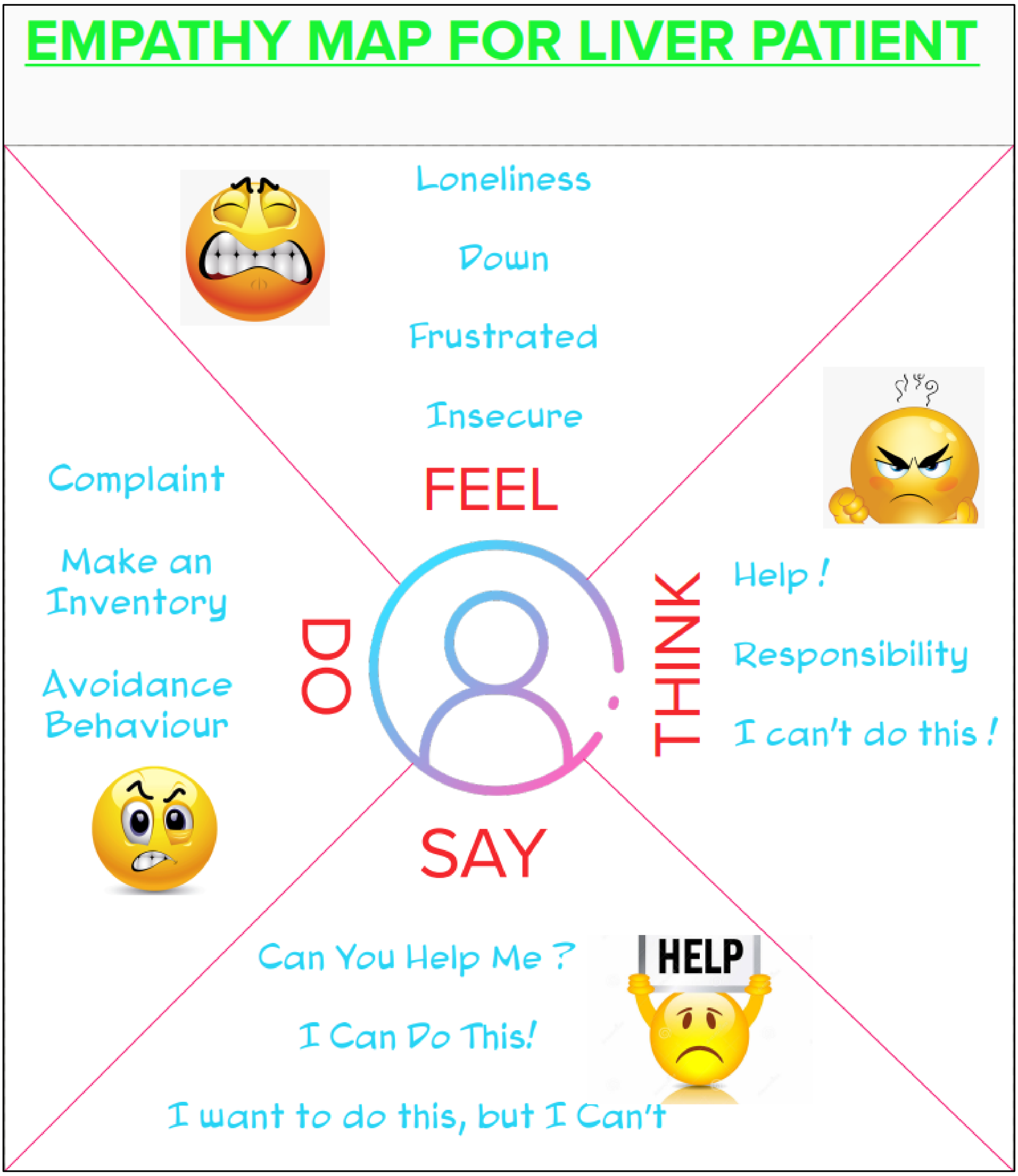
- Reduce the high cost of liver disease diagnosis
- It detects patterns of specific Liver diseases
- It can alert clinicians to any irregularities to assess a patient's health based on knowledge collected from large data sets.
- It shows improved accuracy, efficiency, and decision-making.
- It helps physicians to identify best treatment for particular Liver disease.
- It can produce fast analysis report, operational efficiency and reduce operational cost.

CHAPTER - 2

PROBLEM DEFINITION & DESIGN THINKING

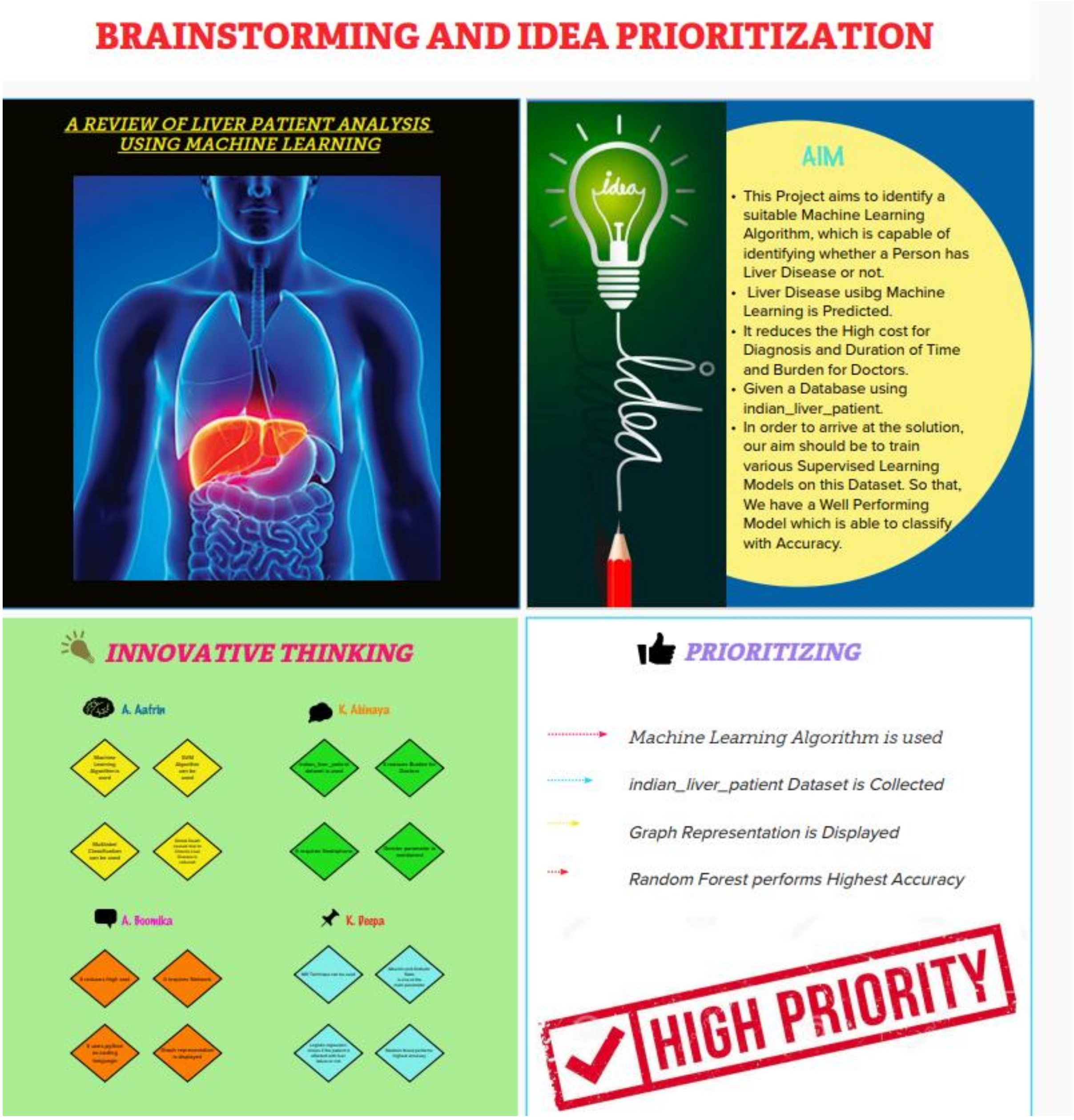
2.1 EMPATHY MAP

A Template that organizes User’s Behaviors and feelings to create sense of empathy between the User and Admin.



2.2 IDEATION & BRAINSTORMING MAP

- ➢ Ideation is the process of forming ideas from Conception to Implementation in a Business Testing
- ➢ Brainstorming Map is a technique that can be used to organize Thoughts and Idea



BRAINSTORMING AND IDEA PRIORITIZATION

A REVIEW OF LIVER PATIENT ANALYSIS USING MACHINE LEARNING

AIM

- This Project aims to identify a suitable Machine Learning Algorithm, which is capable of identifying whether a Person has Liver Disease or not.
- Liver Disease usibg Machine Learning is Predicted.
- It reduces the High cost for Diagnosis and Duration of Time and Burden for Doctors.
- Given a Database using indian_liver_patient.
- In order to arrive at the solution, our aim should be to train various Supervised Learning Models on this Dataset. So that, We have a Well Performing Model which is able to classify with Accuracy.

INNOVATIVE THINKING

Idea
Idea
A. Aafrin
Machine Learning Algorithm is used
SVM Algorithm can be used
Multiclass Classification can be used
K. Abinaya
It reduces Burden for Doctors
Gender parameter is considered
A. Boomika
It reduces High cost
It requires Network
It uses python as coding language
Graph representation is displayed
K. Deepa
ML Techniques can be used
Random forest performs highest accuracy

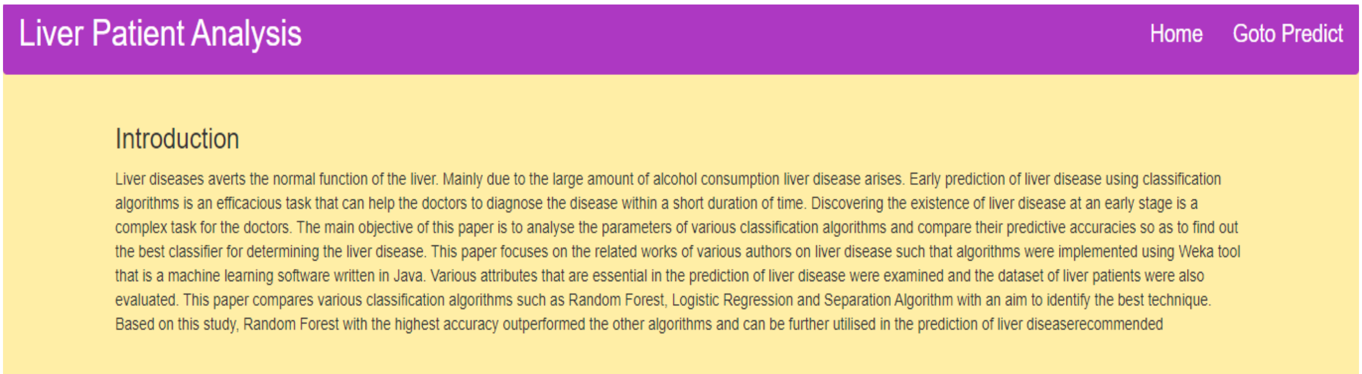
PRIORITIZING

- Machine Learning Algorithm is used
- indian_liver_patient Dataset is Collected
- Graph Representation is Displayed
- Random Forest performs Highest Accuracy

HIGH PRIORITY

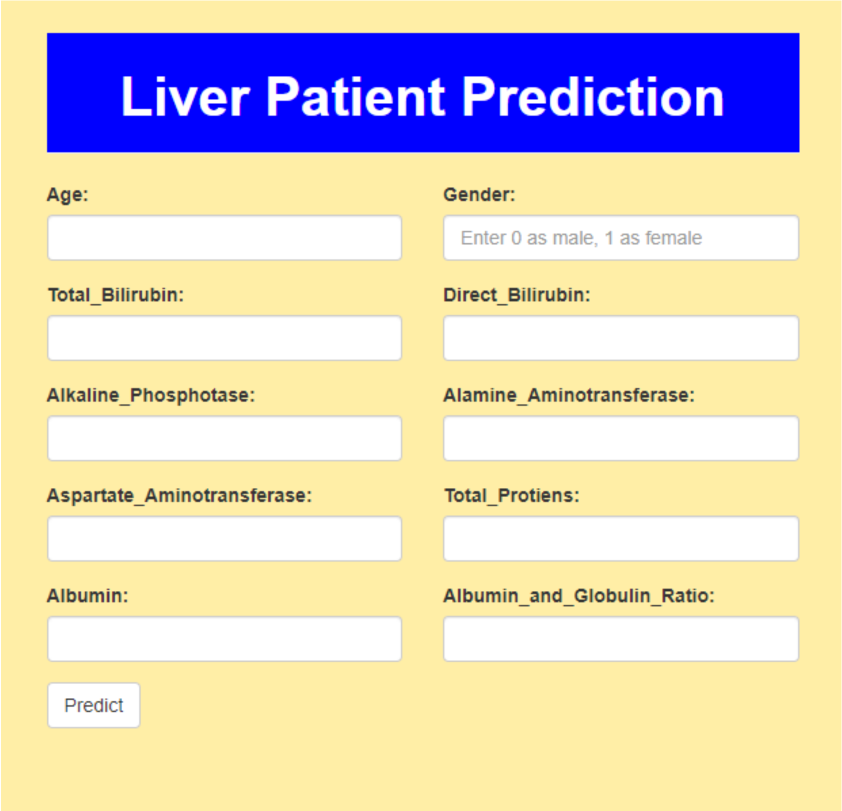
CHAPTER - 3

RESULT



Introduction

Liver diseases averts the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of liver disease at an early stage is a complex task for the doctors. The main objective of this paper is to analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This paper focuses on the related works of various authors on liver disease such that algorithms were implemented using Weka tool that is a machine learning software written in Java. Various attributes that are essential in the prediction of liver disease were examined and the dataset of liver patients were also evaluated. This paper compares various classification algorithms such as Random Forest, Logistic Regression and Separation 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 diseaserecommended



Liver Patient Prediction

Age:

Gender: Enter 0 as male, 1 as female

Total_Bilirubin:

Direct_Bilirubin:

Alkaline_Phosphotase:

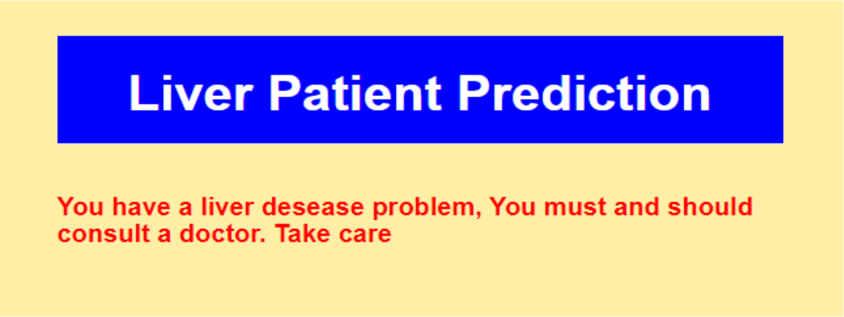
Alamine_Aminotransferase:

Aspartate_Aminotransferase:

Total_Protiens:

Albumin:

Albumin_and_Globulin_Ratio:



Liver Patient Prediction

You have a liver desease problem, You must and should consult a doctor. Take care

CHAPTER - 4

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Reduce the high cost of liver disease diagnosis
- It detects patterns of specific Liver diseases
- It can alert clinicians to any irregularities to assess a patient's health based on knowledge collected from large data sets.
- It shows improved accuracy, efficiency, and decision-making.
- It helps physicians to identify best treatment for particular Liver disease.
- It can produce fast analysis report, operational efficiency and reduce operational cost.

DISADVANTAGES

- Acquisition in Data
- Human Interface is Eliminated
- Nature of the Jobs in being replaced by Machines

CHAPTER - 5

APPLICATIONS

- Liver Disease Monitoring
- Liver Disease Risk Prediction
- Discoveries in Disease
- Diagnosis and Detection of Disease
- Epidemic Outbreak Prediction
- Medical Research
- Image Analysis
- EMR Scans

. CHAPTER - 6

CONCLUSION

Liver disease is a serious condition that threatens human life and requires urgent medical attention. Health professionals are based on pathological methods to make a medical report concerning a patient’s condition. ML models, such as Random Forest, Decision Tree, Logistic Regression, KNN and ANN, were evaluated in terms of Accuracy, Precision, Recall, F-Measure and AUC, in order to predict liver disease occurrence. From the experimental results, 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.

CHAPTER - 7

FUTURE SCOPE

In future work, we aim to re-consider the liver disease manifestation methodology following, first, the ROPE (region of practical equivalence) analysis to test whether a feature is significant (in the sense of important enough for the liver disease risk prediction) and, secondly, extend the machine learning framework by using deep learning methods and comparing the results on the aforementioned metrics.

CHAPTER - 8

APPENDIX

8.1 SOURCE CODE

```
import pandas as pd
import numpy as np
import seaborn as sns
import sklearn
import matplotlib.pyplot as plt
from matplotlib import rcParams
from scipy import stats
from pandas import DataFrame
import pip
data=pd.read_csv('/indian_liver_patient.csv')
data.head()
data.info()
data.isnull().any()
data.isnull().sum()
#checking for the missing data after cleansing data(
data['Albumin_and_Globulin_Ratio'].fillna(data['Albumin_and_Globulin_Ratio'].median(),inplace=True)
data.isnull().sum()
from sklearn.preprocessing import LabelEncoder
lc = LabelEncoder()
data['Gender']=lc.fit_transform(data['Gender'])
data.describe()
sns.distplot(data['Age'])
plt.title('Age distribution Graph')
plt.show()
sns.countplot(data ['Dataset'],
hue=data['Gender'])
plt.figure(figsize=(10,7))
sns.heatmap(data.corr(),annot=True)
from sklearn.preprocessing import scale
X_scaled=pd.DataFrame(scale(X),columns=X.columns)
X_scaled.head()
X=data.iloc[:,:-1]
y=data.Dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test= train_test_split(X_scaled,y,test_size=0.2,
random_state=42)
pip install imblearn
from imblearn.over_sampling import SMOTE
smote = SMOTE()
y_train.value_counts()
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
y_train_smote.value_counts()
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fit(X_train_smote, y_train_smote)
rf_predict=model1.predict(X_test)
rfc1=accuracy_score(y_test,y_predict)
rfc1
pd.crosstab(y_test,y_predict)
```

```
print(classification_report(y_test,y_predict))
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
model4=DecisionTreeClassifier()
model4.fit(X_train_smote, y_train_smote)
y_predict=model4.predict(X_test)
dtc1=accuracy_score(y_test,y_predict)
dtc1
pd.crosstab(y_test,y_predict)
print(classification_report(y_test, y_predict))
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
model2=KNeighborsClassifier()
model2.fit(X_train_smote, y_train_smote)
y_predict = model2.predict(X_test)
knn1=accuracy_score(y_test, y_predict)
knn1
pd.crosstab(y_test, y_predict)
print(classification_report(y_test, y_predict))
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
model5=LogisticRegression()
model5.fit(X_train_smote, y_train_smote)
y_predict=model5.predict(X_test)
```

```
logi1=accuracy_score(y_test, y_predict)

logi1

pd.crosstab(y_test,y_predict)

print (classification_report (y_test, y_predict))

import tensorflow.keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

#intialising the ANN

classifier = Sequential()

#adding the input layer and the first hidden layer

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

#adding the second hidden layer

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

#adding the output layer

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

#compiling the ANN

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

#fitting the ANN to the training set

model_history = classifier.fit(X_train,y_train,
batch_size=100,validation_split=0.2, epochs=100)

model4.predict([[50,1,1,2,0.8,150,70,80,7.2,3.4,0.8]])

model1.predict([50,1,1.2,0.8,150,70,80,7.2,3.4,0.8])

classifier.save("liver.h5")

y_pred = classifier.predict(X_test)

y_pred

def predict_exit(sample_value):

 sample_value = np.array(sample_value)
 sample_value = sample_value.reshape(1,-1)
 sample_value = scale(sample_value)
 return classifier.predict(sample_value)
sample_value =[[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]
if predict_exit(sample_value )>0.5:
print('prediction:liver patient')
else:
print('prediction:healthy')
acc_smote= [['KNN classifier',knn1],['RandomForestClassifier',rfc1],
['DecisionTreeClassifier',dtc1],['LogisticRegression',logi1]]
Liverpatient_pred= pd.DataFrame(acc_smote, columns=['classification
models','accuracy_score']
Liverpatient_pred
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.metrics import accuracy_score
plt.figure(figsize=(7,5))
plt.xticks(rotation=90)
plt.title('classificaton models & accuracy scores after SMOTE',fontsize=18)
sns.barplot(x="classification models", y="accuracy_score",
data=Liverpatient_pred)
from sklearn.ensemble import ExtraTreesClassifier
model=ExtraTreesClassifier()
model.fit(X,y)
model.feature_importances_
dd=pd.DataFrame(model.feature_importances_,index=X.columns).sort_values(
0,ascending=False)
dd
dd.plot(kind='barh',figsize=(7,6))
```

```
plt.title("FEATURE IMPORTANCE",fontsize=14)
import joblib
joblib.dump(model1, 'ETC.pk1')
```