**A REVIEW OF LIVER PATIENT ANALYSIS**

**METHODS USING MACHINE LEARNING**

**1.INTRODUCTION**

Nonalcoholic fatty liver disease (NAFLD) is one of the most common chronic liver diseases worldwide and has become a signifcant public health concern [1, 2]. Te spectrum of NAFLD ranges from simple steatosis and nonalcoholic steatohepatitis (NASH) to fbrosis. Simple steatosis is considered to have a benign progression, while NASH may progress to fbrosis, cirrhosis, and even hepatocellular carcinoma [3, 4]. Furthermore, NAFLD is a disease signifcantly associated with metabolic syndrome, cardiovascular disease, and type 2 diabetes [5–7]. For these reasons, it is critically important to obtain an early diagnosis that would enable improved prevention and management of NAFLD.

**Over view:**

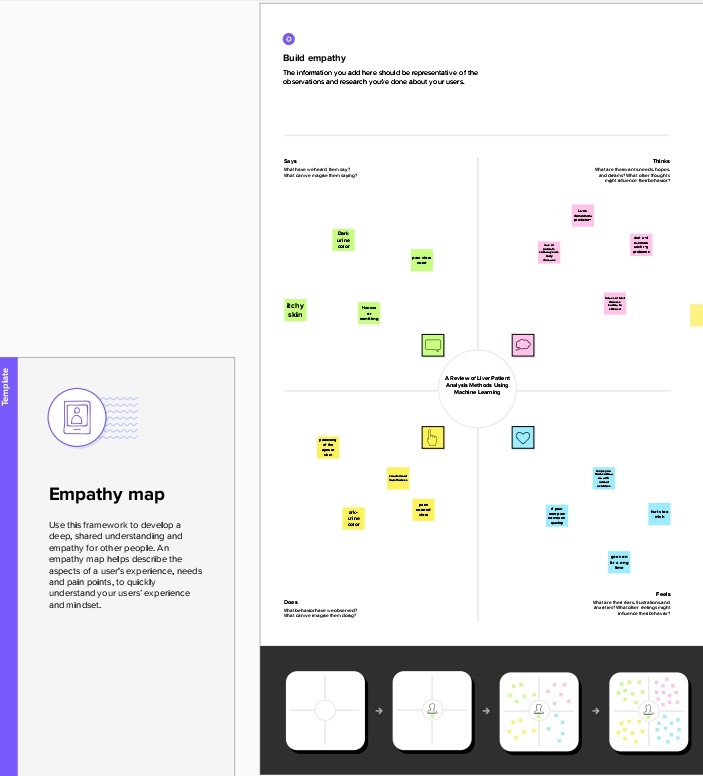
Abstract: Liver Disease is the leading cause of global death that impacts the massive quantity of humans around the world. This disease is caused by an assortment of elements that harm the liver. For example, obesity, an undiagnosed hepatitis infection, alcohol misuse which is responsible for abnormal nerve function, coughing up or vomiting blood, kidney failure, liver failure, jaundice, liver encephalopathy and there are many more. Diagnosis of liver infection at 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 chronic liver disease diagnosis by prediction. In this work, we used five algorithms Logistic Regression, Decision Tree, Support Vector Machine, Naïve Bayes, and Random Forest. The performance of different classification techniques was evaluated on different measurement techniques such as accuracy, precision, recall, and specificity. We found the accuracy 74%, 72%, 72%, 71%, and 57% for SVM,DT,RF,LR and NB. The analysis result shown the SVM achieved the highest accuracy. Moreover, 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.

**Purpose:**

The purpose of the present study was to employ a computer-aided diagnosis system that classifies [chronic liver disease](https://www.sciencedirect.com/topics/medicine-and-dentistry/chronic-liver-disease) (CLD) using ultrasound shear wave [elastography](https://www.sciencedirect.com/topics/physics-and-astronomy/elastography" \o "Learn more about elastography from ScienceDirect's AI-generated Topic Pages) (SWE) imaging, with a stiffness value-clustering and machine-learning algorithm. A clinical data set of 126 patients (56 healthy controls, 70 with CLD) was analyzed. First, an RGB-to-stiffness inverse mapping technique was employed. A five-cluster segmentation was then performed associating corresponding different-color regions with certain stiffness value ranges acquired from the SWE manufacturer-provided color bar. Subsequently, 35 features (7 for each cluster), indicative of physical characteristics existing within the SWE image, were extracted. A stepwise regression analysis toward feature reduction was used to derive a reduced feature subset that was fed into the support vector machine classification algorithm to classify CLD from healthy cases. The highest accuracy in classification of healthy to CLD subject discrimination from the support vector machine model was 87.3% with sensitivity and specificity values of 93.5% and 81.2%, respectively. Receiver operating characteristic curve analysis gave an area under the curve value of 0.87 (confidence interval: 0.77–0.92). A machine-learning algorithm that quantifies color information in terms of stiffness values from SWE images and discriminates CLD from healthy cases is introduced. New objective parameters and criteria for CLD diagnosis employing SWE images provided by the present study can be considered an important step toward color-based interpretation, and could assist radiologists' [diagnostic performance](https://www.sciencedirect.com/topics/medicine-and-dentistry/diagnostic-performance) on a daily basis after being installed in a PC and employed retrospectively, immediately after the examination

**2.Problem Definition & Design Thinking**

**Empathy Map:**

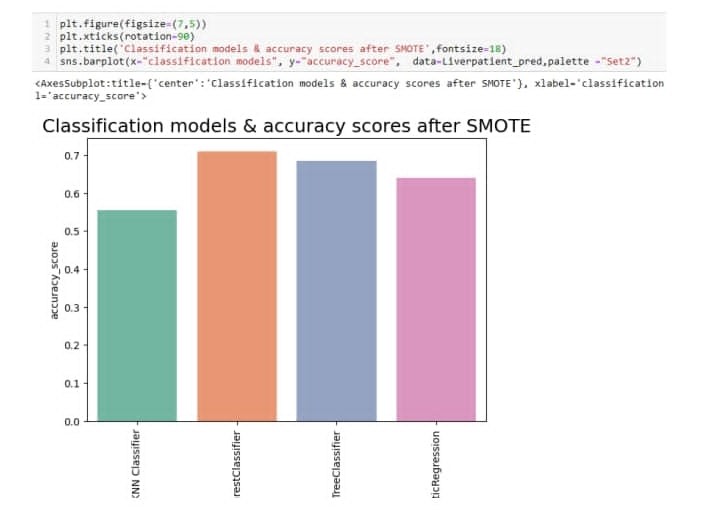
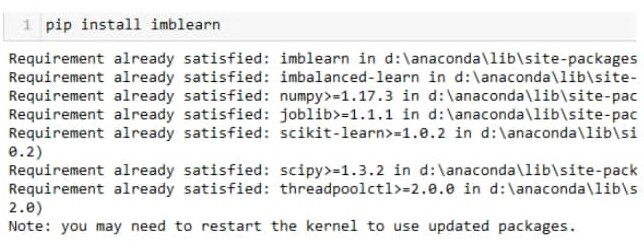
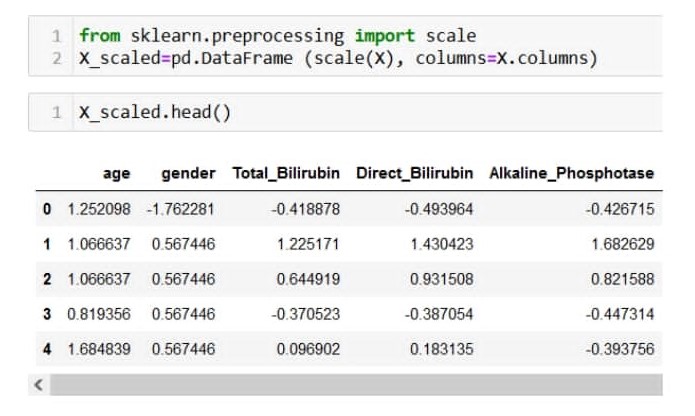
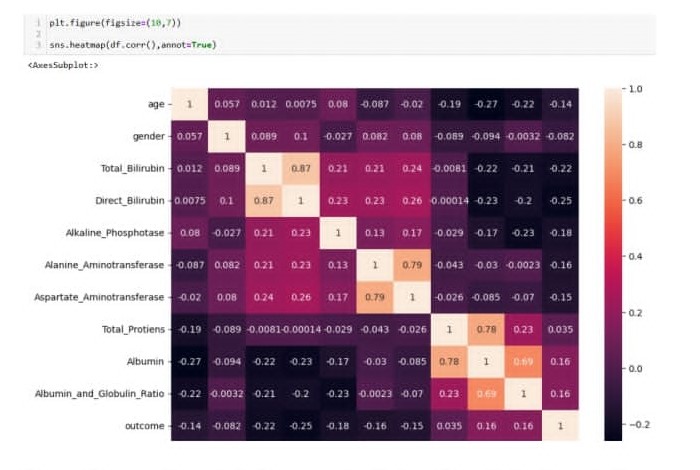
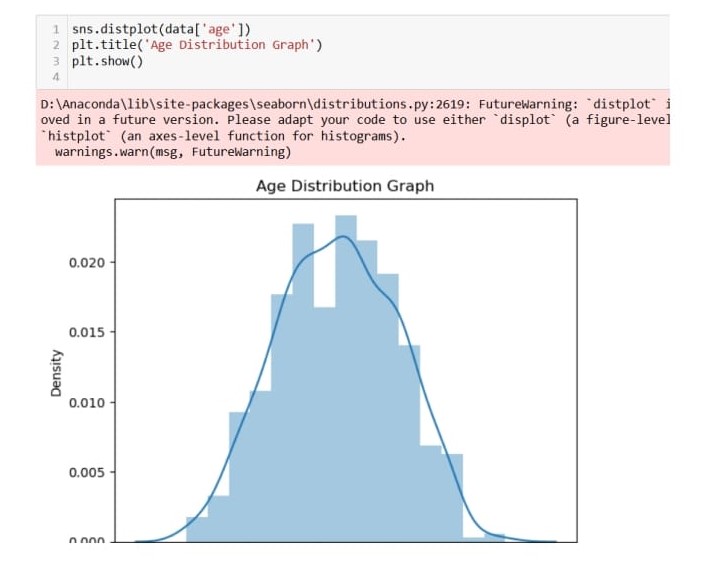
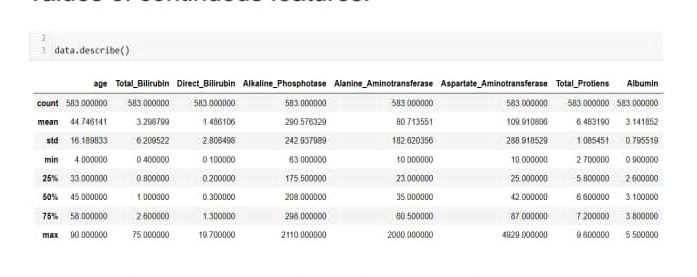
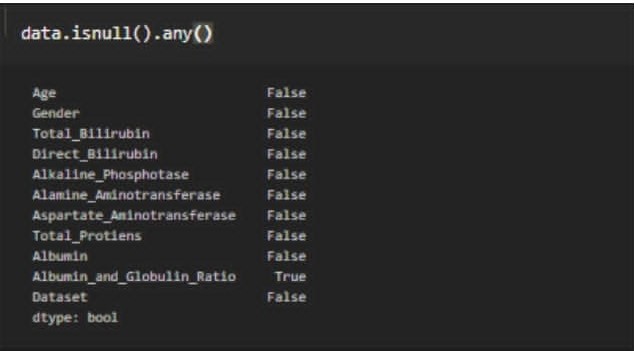
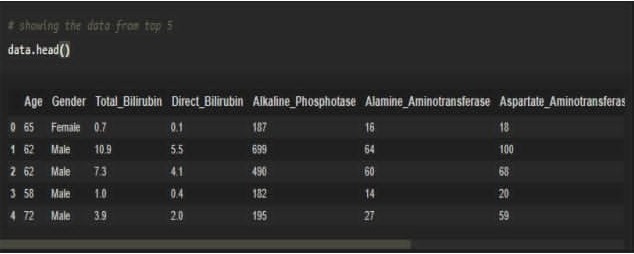


**Ideation & Brainstorming Map:**

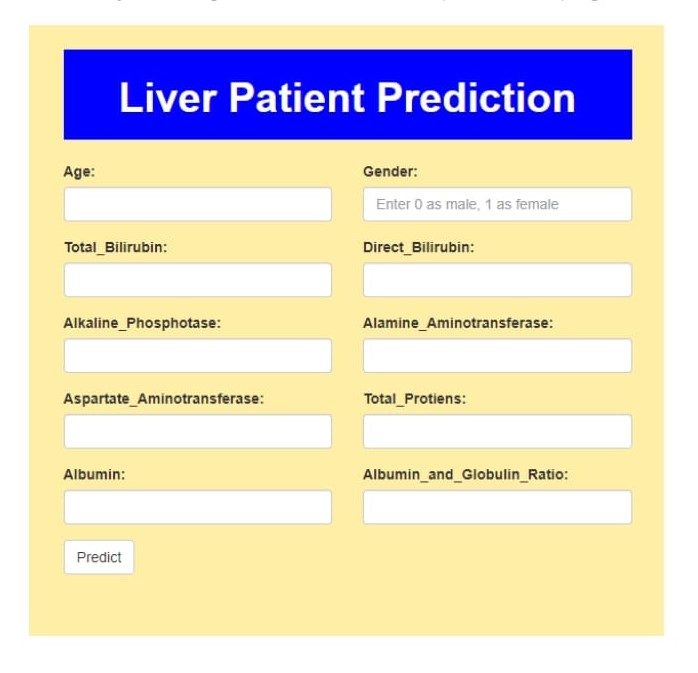


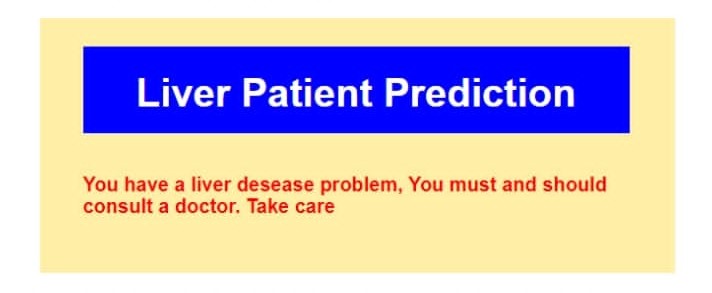
**3.Result**:

**Colab Result**



**Website Result**





**4. Advantages & Disadvantages**

**Advantages:**

Liver disease is a serious health condition that affects millions of people worldwide. Early detection and accurate diagnosis of liver diseases are crucial for effective treatment and management. In recent years, machine learning algorithms have been widely used to analyze liver patient data to assist clinicians in making more accurate diagnoses and predicting disease progression. Here are some advantages of using machine learning methods for liver patient analysis:

* Accuracy: Machine learning algorithms can analyze large amounts of patient data with high accuracy, which can help clinicians to make more accurate diagnoses and treatment decisions.
* Efficiency: Machine learning algorithms can analyze patient data much faster than traditional methods, which can help clinicians to make more timely diagnoses and treatment decisions.
* Personalized Medicine: Machine learning algorithms can analyze patient data to identify patterns that are unique to each patient, which can help clinicians to develop personalized treatment plans that are tailored to each patient's individual need.
* Predictive Analytics: Machine learning algorithms can analyze patient data to predict disease progression and identify patients who are at high risk for developing complications.
* Cost-Effective: Machine learning algorithms can help reduce healthcare costs by enabling clinicians to make more accurate diagnoses and treatment decisions, which can lead to fewer hospitalizations and lower healthcare costs.
* Some common machine learning algorithms used for liver patient analysis include decision trees, logistic regression, support vector machines, random forests, and neural networks. These algorithms can be trained using different types of data, such as medical images, laboratory test results, and patient demographics.
* By analyzing this data, these algorithms can identify patterns that are indicative of liver disease, which can help clinicians to make more accurate diagnoses and treatment decisions.
* In conclusion, machine learning methods have many advantages when it comes to analyzing liver patient data.
* These methods can improve the accuracy and efficiency of diagnosis and treatment decisions, enable personalized medicine, provide predictive analytics, and reduce healthcare costs.
* As technology continues to evolve, machine learning algorithms will likely play an increasingly important role in liver patient analysis and care.

**Disadvantages**:

While there are many advantages to using machine learning algorithms for liver patient analysis, there are also some potential disadvantages that should be considered: Limited Data Availability: Machine learning algorithms require large amounts of high-quality data to be trained effectively. However, there may be limited data available for liver disease patients, especially for rare or complex diseases.

* Bias and Overfitting: Machine learning algorithms can be susceptible to bias and overfitting if they are not properly trained and validated. This can lead to inaccurate predictions and diagnoses, especially if the algorithm is trained on a biased dataset.
* Lack of Interpretability: Some machine learning algorithms can be difficult to interpret, which can make it challenging for clinicians to understand the underlying factors contributing to a particular diagnosis or prediction.
* Ethical Concerns: There may be ethical concerns related to the use of machine learning algorithms for liver patient analysis, such as data privacy and security, and potential biases in the algorithm that could lead to discrimination.
* Expertise Requirements: Machine learning algorithms require specialized expertise to develop and deploy effectively, which may be a barrier for healthcare providers who do not have the necessary skills or resources.
* Technical Limitations: Machine learning algorithms may require significant computing power and storage capacity, which can be costly and challenging to implement in resource-limited settings.
* In conclusion, while machine learning algorithms have many potential benefits for liver patient analysis, there are also several potential limitations and challenges that need to be considered.
* These include limited data availability, bias and overfitting, lack of interpretability, ethical concerns, expertise requirements, and technical limitations. Addressing these challenges will be critical to ensuring that machine learning algorithms can be used effectively and responsibly to improve liver patient care.

**5. Applications**

* LD is a common clinical disorder; it is also associated with high morbidity and mortality.
* Additionally, LD has been increasing in parallel with the prevalence of diabetes, metabolic syndrome, alcohol and obesity . Higher prevalence of LD has appeared as a greater economic burden.
* Therefore, accurate identification of individuals at risk and early recognition of LD could offer immense benefits for diagnosis, prevention, or even proper treatment.
* Subsequently, reliance on a single diagnostics test is not sufficient to evaluate liver function . A wide variety of biochemical measures are therefore used to determine the general condition of the liver.
* Different biochemical tests commonly referred to as Liver Function Tests (LFT) provide secondary evidence for hepatic diseases .
* Metrical record, physical examination along with diagnostic test (LFTs) results entail to recognize patients with liver disease; diagnosis of differential jaundice; monitor the severity (*i*.*e*., course and response of the disease); and detect hepatotoxicity caused by various agents .
* In addition, commonly used LFTs are mainly used to determine liver damage instead of monitoring hepatic functions which can make the identification of disease complicated . Certainly, these biochemical tests can also detect problems such as hemolysis (high bilirubin), higher alkaline phosphatase level (bone disease).
* Abnormal LFTs often suggest that the liver may not function properly and indicate the severity of the problem. But still, the correctness and accuracy to predict liver disease remain uncertain.

**6. Conclusion**

* We used machine learning to develop software to rapidly and objectively analyze liver biopsy specimens for histologic features of NAFLD.
* The results from the software correlate with those from histo pathologists, with high levels of inter observer and intra observer agreement.Findings were validated in a separate group of patients.
* This tool might be used for objective assessment of response to therapy for NAFLD in practice and clinical trials.

**7. Future Scope**

* There has been a rapid growth in the use of automatic decision-making systems and tools in the medical domain.
* By using the concepts of big data, deep learning, and machine learning, these systems extract useful information from large medical datasets and help physicians in making accurate and timely decisions regarding predictions and diagnosis of diseases.
* In this regard, this study provides an extensive review of the progress of applying Artificial Intelligence in forecasting and detecting liver diseases and then summarizes related limitations of the studies followed by future research.

**8.Appendix**

* Appendicitis is a common disease that occurs particularly often in childhood and adolescence.
* The accurate diagnosis of acute appendicitis is the most significant precaution to avoid severe unnecessary surgery.
* In this paper, the author presents a machine learning (ML) technique to predict appendix illness whether it is acute or subacute, especially between 10 and 30 years and whether it requires an operation or just taking medication for treatment.
* The dataset has been collected from public hospital-based citizens between 2016 and 2019. The predictive results of the models achieved by different ML techniques (Logistic Regression, Naïve Bayes, Generalized Linear, Decision Tree, Support Vector Machine, Gradient Boosted Tree, Random Forest) are compared.
* The covered dataset are 625 specimens and the total of the medical records that are applied in this paper include 371 males (60.22%) and 254 females (40.12%). According to the dataset, the records consist of 318 (50.88%) operated and 307 (49.12%) unoperated patients.
* It is observed that the random forest algorithm obtains the optimal result with an accurately predicted result of 83.75%, precision of 84.11%, sensitivity of 81.08%, and the specificity of 81.01%. Moreover, an estimation method based on ML techniques is improved and enhanced to detect individuals with acute appendicitis.

**A. Source code**:

**DATA COLLECTION & PREPARATION**

**Collect the dataset**

**Importing the libraries**

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

**1.2: Read the Dataset**

data = pd.read\_csv(‘E:/Datascience/Datasets/indian\_liver\_patient.csv ‘)

data.head()

**Data Preparation**

**Handling missing values**

data.info()

data.isnull().any()

data.isnull().sum()

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

data.isull().sum()

**Handling Categorical Values**

from sklearn.preprocessing import LabelEncoder

lc = LabelEncoder()

data = [‘gender’]= lc.fit\_transform(data[‘gender’])

**EXPLORATORY STATISTICAL**

Descriptive Statistical

data.describe()

**Visual analysis**

**Univariate analysis**

sns.displot(data[‘age’]

plt.title(‘Age Distribution Graph’)

plt.show()

**Brivariate analysis**

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

**Multivariate analysis**

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

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

**Scalling the Data**

From sklearn.preprocessing import scale

X\_scaled=pd.DataFrame (scale(X), columns=X, columns)

X\_scaled.head()

**Splitting data into train and test**

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)

**Handling Imbalance Data**

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()

**MODEL BUILDING**

**Training the model in multiple algorithms**

**Random Forest model**

from sklearn.ensemble import RandomForestClassifier

model1=RandomForestClassifier()

model1.fit(X\_train\_smote, Y\_train\_smote)

Y\_predict=model1.predict(X\_test)

rfc1=accurancy\_score(Y\_test, Y\_predit)

rfc1

pd.crosstab(Y\_test, Y\_predict)

print(classification\_report(Y\_test, Y\_predit))

**Decision tree model**

from sklearn.tree import DescisionTreeclassifier

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)

**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\_prdict)

**Logistic Regression model**

from sklearn.linear \_model import LogisticRegression

model5=LogisticRegression()

model5.fit(X\_train\_smote, y\_train\_smote)

y\_predict=model5.predict(X\_test)

logil=accuracy\_score(y\_test, y\_predict)

logil

pd.crosstab(y\_test, y\_predict)

print(classification\_report(y-test, y\_predict)

**ANN model**

import tensorflow.keras

from tensorflow .keras.models import Sequential

from tensorflow.keras.layers import Dense

classifier=sequential()

classifier.add(Dense(units=100, activation=’relu’, imput\_dim=10))

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

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

classifier.compile(optimizer=’adam’, loss=’binary\_crossentropy’, metrics=[‘accracy’])

model\_history=classifier.fit(x\_train, y\_train, batch\_size=100, validation\_split=0.2, epochs=100)

**Testing the model**

model4.predict([[50,1,1.2,0.3,150,70,80,7.2,3.4,0.8]])

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

y\_pred=(y\_pred>0.5)

y\_pred05

def predict\_exit(sample\_value):

sample\_value=np.array(sample\_value)

sample\_value=sample\_value.reshape(1,-1)

sample\_value-scale(sample\_value)

return calssificer.predict(sample\_value)

sample\_value[[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]

if predit\_exit(sample\_value)>0.5:

print(‘Prediction: Liver Patient’)

else:

print(‘Prediction: Healthy’)

**Performance Testing & Hyperparameter Tuning**

**Compare the model**

Acc\_smote=[[‘KNN Classifier’,knn1], [‘RandomForestClassifier’, rfc1], [‘DecisionTreelassifier’, dtc1], [‘LogisticRegression’, logil]]

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 model5 & accuracy scores after SMOTE’, fontsize=18)

Sns.barplot(x=”classificationmodels”, y=”accuracy\_score”, data=liverpatient\_pred, palette=”set2”)

from sklearn.ensemble import ExtraTreeClassifier

model=ExtraTreeClassifier()

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)

**Model Deployment**

**Save the best model**

Import joblib

Joblib.dump(model1,’ETC.pkl’)