A Review of Liver Patient Analysis Methods Using Machine Learning

Introduction

Liver diseases averts the normal function of the liver. This disease is caused by an assortment of elements that harm the liver.

Overview

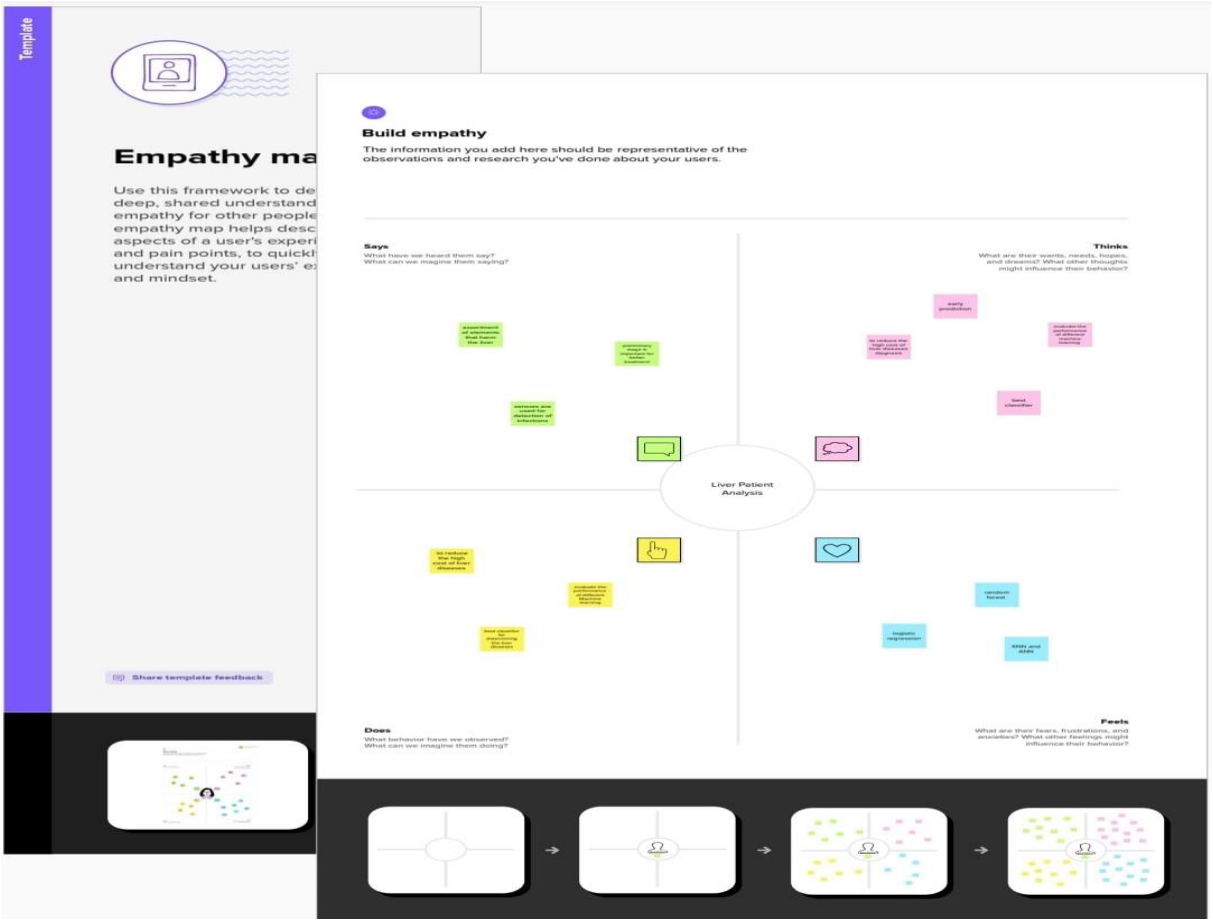
Diagnosis of liver infection at the preliminary stage is important for better treatment. In today's scenario devices like sensors are used for detection of infections. Accurate classification techniques are required for automatic identification of disease samples. This disease diagnosis is very costly 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

Purpose

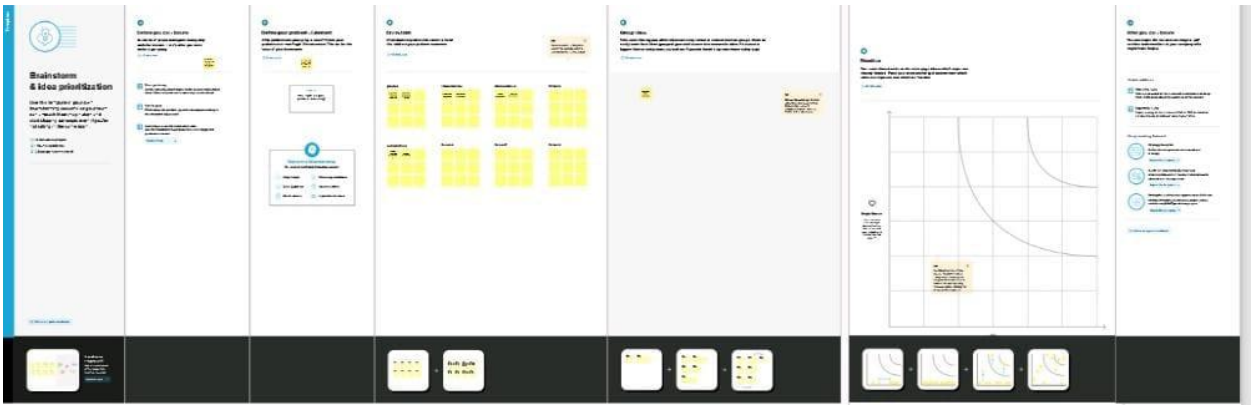
In this project we will analyze the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This project compares various classification algorithms such as Random Forest, Logistic Regression, KNN 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 utilized in the prediction of liver disease and can be recommended to the user.

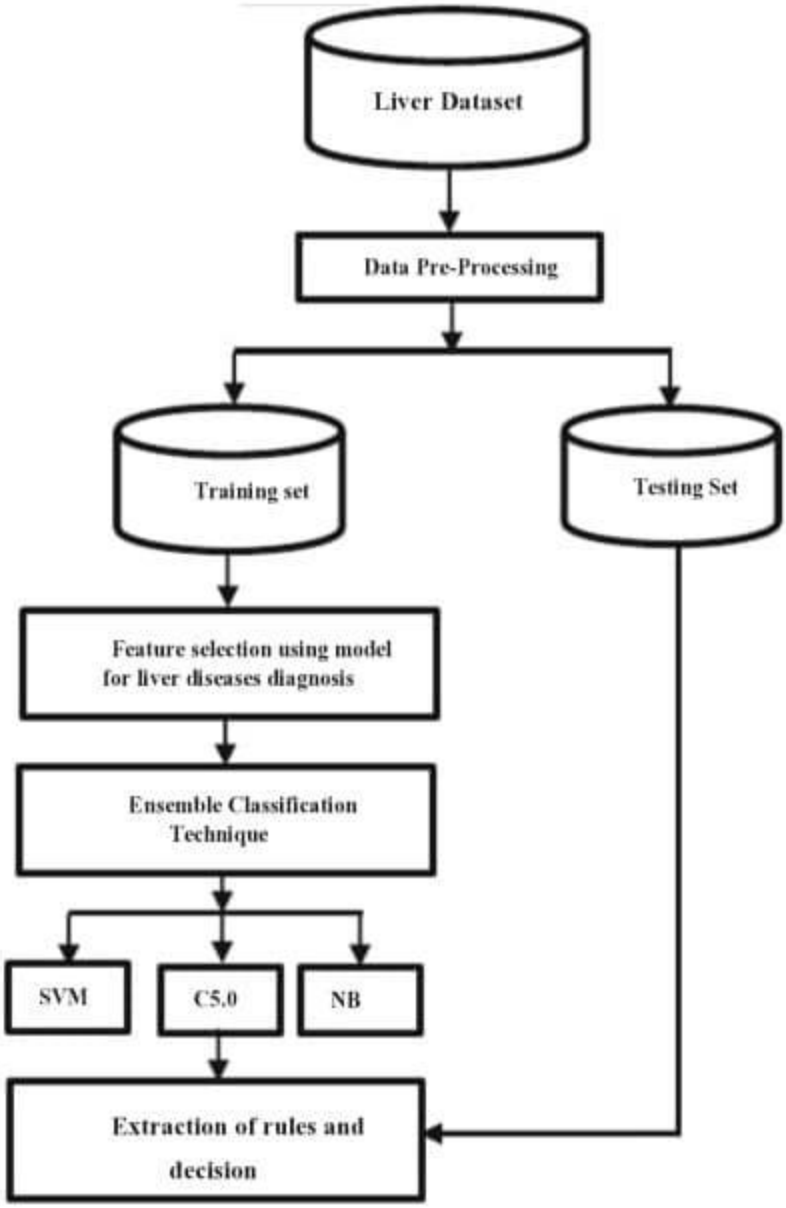
Problem Definition & Design Thinking

Empathy Map



2.2 Ideation & Brainstorming Map


Brainstorm
& idea prioritization


Liver Dataset
Data Pre-Processing
Training set
Testing Set
Feature selection using model for liver diseases diagnosis
Ensemble Classification Technique
SVM
C5.0
NB
Extraction of rules and decision

RESULT

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferas
0	65	Female	0.7	0.1	187	16	18
1	62	Male	10.9	5.5	699	64	100
2	62	Male	7.3	4.1	490	60	68
3	58	Male	1.0	0.4	182	14	20
4	72	Male	3.9	2.0	195	27	59

```
data.info()
```

```
<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
```

```
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
```

```
#checking for the missing data after cleaning data
data['Albumin_and_Globulin_Ratio'] = data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
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  0
Dataset                     0
dtype: int64
```

	age	gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase
0	1.252098	-1.762281	-0.418878	-0.493964	-0.426715
1	1.066637	0.567446	1.225171	1.430423	1.682629
2	1.066637	0.567446	0.644919	0.931508	0.821588
3	0.819356	0.567446	-0.370523	-0.387054	-0.447314
4	1.684839	0.567446	0.096902	0.183135	-0.393756

Advantages and Disadvantages

Diagnostic criterion standard Confirmed diagnostic value Highly invasive test Etiologic suggestion Differential diagnosis. The potential complications include death Grade and stage evaluation Therapeutic decision Significant sampling error (eligibility) High costTreatment evaluation Inter-observer variation (effectiveness)Follow-up comparison of treated and untreated patients.

Application

The gold standard for the diagnosis of liver fibrosis and nonalcoholic fatty liver disease (NAFLD) is liver biopsy. Various noninvasive modalities, e.g., ultrasonography, elastography and clinical predictive scores, have been used as alternatives to liver biopsy, with limited performance. Recently, artificial intelligence (AI) models have been developed and integrated into noninvasive diagnostic tools to improve their performance.

Methods

We systematically searched for studies on AI-assisted diagnosis of liver fibrosis and NAFLD on MEDLINE, Scopus, Web of Science and Google Scholar. The pooled sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV) and diagnostic odds ratio (DOR) with their 95% confidence intervals (95% CIs) were calculated using a random effects model. A summary receiver operating characteristic curve and the area under the curve was generated to determine the diagnostic accuracy of the AI-

assisted system. Subgroup analyses by diagnostic modalities, population and AI classifiers were performed.

CONCLUSION

Conclusion and Feature WorkIn this paper, we proposed and built a machine learning based on a hybrid classifier tobe used as a classification model for liver diseases diagnosis to improve performanceand experts to identify the chances of disease and conscious prescription of furthertreatment healthcare and examinations.In future work, the use of fast datasets technique like Apache Hadoop or Spark canbe incorporated with this technique. In addition to this, we can use distributed refinedalgorithms like Forest Tree implemented in Apache Hadoop to increase scalability and efficiency.

FUTURE SCOPE

Liver diseases have produced a big data such as metabolomics analyses, electronic health records, and report including patient medical information, and disorders. However, these data must be analyzed

and integrated if they are to produce models about physiological mechanisms of pathogenesis. We use machine learning based on classifier for big datasets in the fields of liver to Predict and therapeutic discovery. A dataset was developed with twenty three attributes that include the records of 7000 patients in which 5295 patients were male and rests were female. Support Vector Machine (SVM), Boosted C5.0, and Naive Bayes (NB), data mining techniques are used with the proposed model for the prediction of liver diseases. The performance of these classifier techniques are evaluated with accuracy, sensitivity, specificity.

APPENDIX

```
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

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['outcome'],hue=data['gender'])

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

sns.heatmap(df.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.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,rando

m_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()

from sklearn.ensemble import RandomForestClassifier

model1=RandomForestClassifier()

model1.fit(X_train_smote,Y_train_smote)

y_predit=model1.predict(X_test)
```

rfc1=accurcy_score(y_test,y_predit)

rfc1

pd.crosstab(y_test,y_predit)

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))

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.linear_model import LogisticRegression model5 Logistic Regression() models.fit(x_train_smote, y_train_smote) y_predict

```
model5.predict(x_test) logit accuracy_score(y_test, y_predict) logil
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
 # Initialising 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=0.7, epochs=106)

#Age Gender Total Bilrabin-Direct Lrubin Alkaline_ Phosphatase Alant _AminotransferoseAsparaty_Aminot

mode14.predict([[50,1,1.2,0.8,150,79,00,7.2.3.4,0.8]))

D:\Anaconda\lib\site-packages\sklearn\base.py:450: Userlarning: x does not have valid feature names, but DecisionTreeCla ifier was fitted with feature nanes warnings.warn

(array([1], dtype-Int64)

Age Gender Total Bilrabin-Direct Lrubin Alkaline_ Phosphatase Alant_ Aminotransferose Asparaty_ Aminot

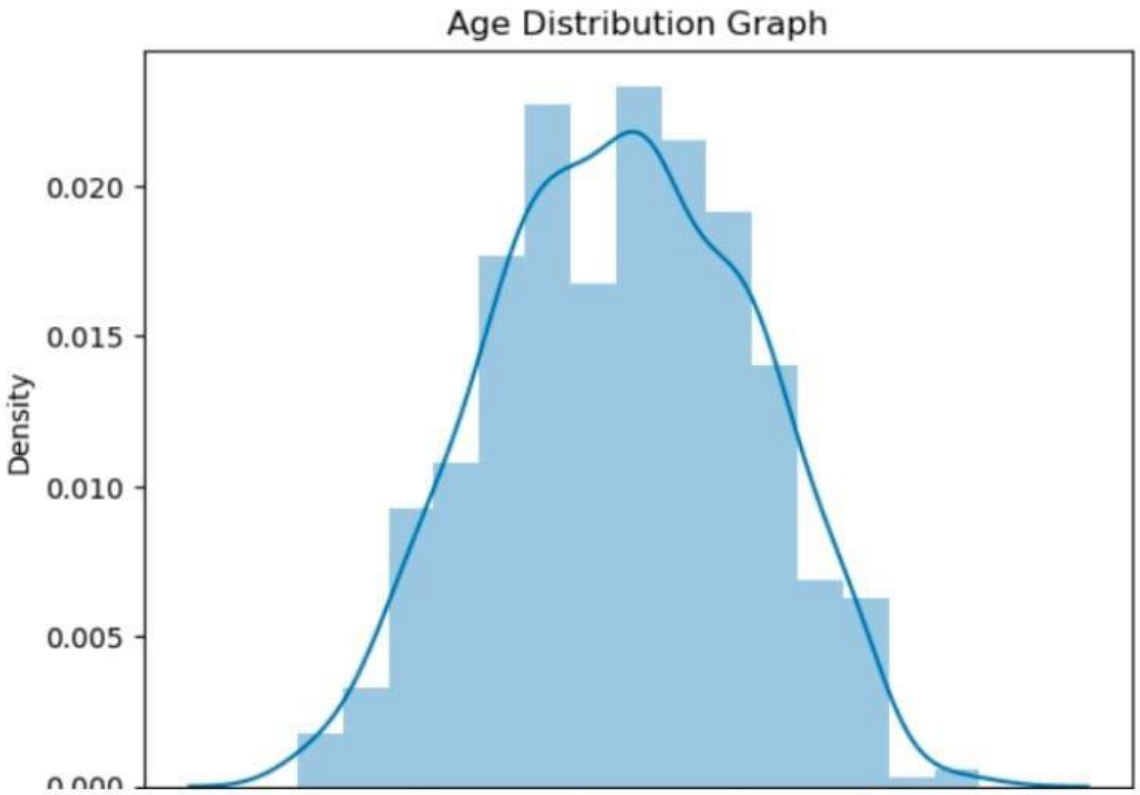
model1.predict([[50,1,1.2,0.8,150,79,00,7.2.3.4,0.8]))

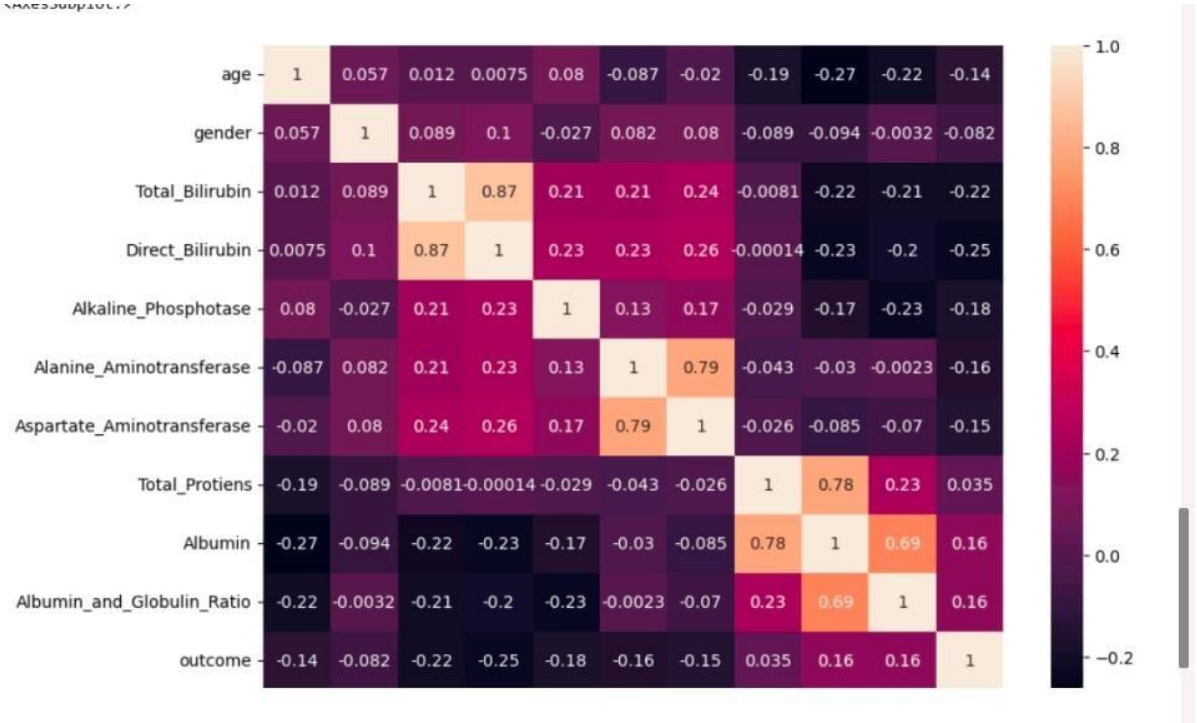
```
classifier.save("liver.h5")

 y_pred= classifier.predict(X_test)

4/4 [======================] –n 0s 2ms/step

 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)

 #Age Gender Total Bilrubin Direct Bilrubin Alkaline Phosphotase
```

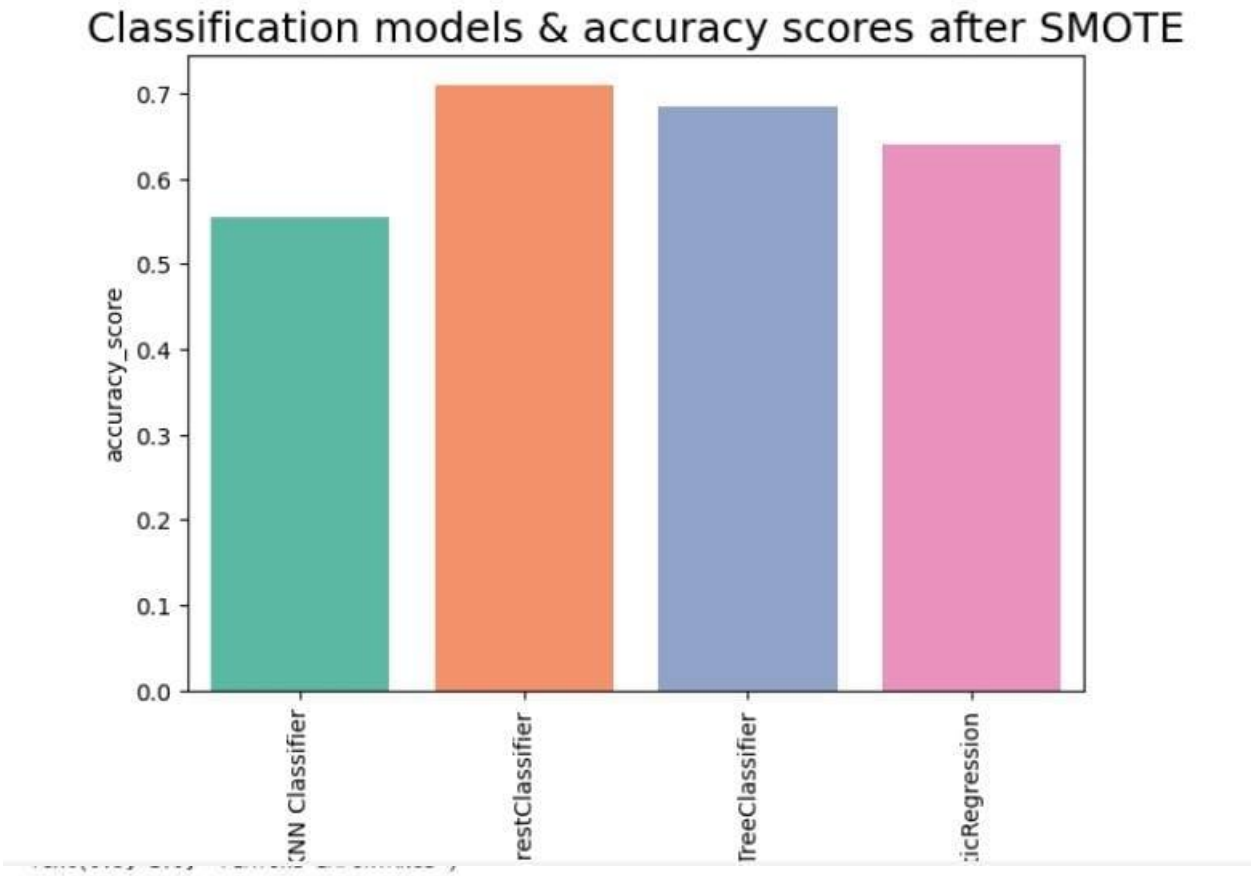
```
sample_value = [[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]

if predict_exit(sample_value)>0.5:

print("Prediction: Liver Patient')

else:

print("Prediction: Healthy ')

acc_smote= [["KNN Classifier', knn1], ['RandomForestClassifier', rfc1], 2
['DecisionTreeClassifier', 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 models & accuracy scores after SHOTE, fontsize=18)

sns.barplot(x="classification models", y accuracy score", data-Liverpatient
pred, palette "Set2")

from sklearn.ensemble import ExtraTreesClassifier
```

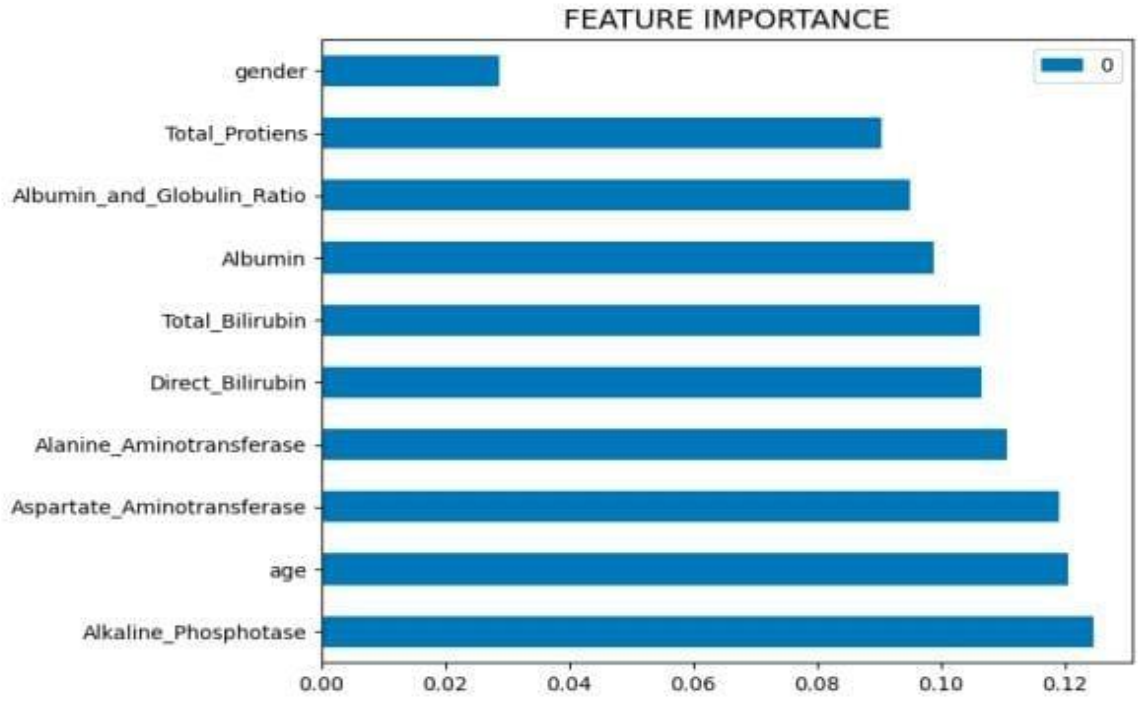
```
model-ExtraTreesClassifier()

model.fit(x,y)

ExtraTreesClassifier()

model.feature_importances_

dd.plot(kind-"barh', figsize-(7,6))

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

import Joblib.

 joblib.dump (model1, ETC.pk1')
```

OUTPUT


Age Distribution Graph
Density
0.020
0.015
0.010
0.005
0.000


age 1 0.057 0.012 0.0075 0.08 -0.087 -0.02 -0.19 -0.27 -0.22 -0.14
gender 0.057 1 0.089 0.1 -0.027 0.082 0.08 -0.089 -0.094 -0.0032 -0.082
Total_Bilirubin 0.012 0.089 1 0.87 0.21 0.21 0.24 -0.0081 -0.22 -0.21 -0.22
Direct_Bilirubin 0.0075 0.1 0.87 1 0.23 0.23 0.26 -0.00014 -0.23 -0.2 -0.25
Alkaline_Phosphotase 0.08 -0.027 0.21 0.23 1 0.13 0.17 -0.029 -0.17 -0.23 -0.18
Alanine_Aminotransferase -0.087 0.082 0.21 0.23 0.13 1 0.79 -0.043 -0.03 -0.0023 -0.16
Aspartate_Aminotransferase -0.02 0.08 0.24 0.26 0.17 0.79 1 -0.026 -0.085 -0.07 -0.15
Total_Protiens -0.19 -0.089 -0.0081 -0.00014 -0.029 -0.043 -0.026 1 0.78 0.23 0.035
Albumin -0.27 -0.094 -0.22 -0.23 -0.17 -0.03 -0.085 0.78 1 0.69 0.16
Albumin_and_Globulin_Ratio -0.22 -0.0032 -0.21 -0.2 -0.23 -0.0023 -0.07 0.23 0.69 1 0.16
outcome -0.14 -0.082 -0.22 -0.25 -0.18 -0.16 -0.15 0.035 0.16 0.16 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Classification models & accuracy scores after SMOTE
accuracy_score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
NN Classifier
restClassifier
TreeClassifier
icRegression


FEATURE IMPORTANCE
0
gender
Total_Protiens
Albumin_and_Globulin_Ratio
Albumin
Total_Bilirubin
Direct_Bilirubin
Alanine_Aminotransferase
Aspartate_Aminotransferase
age
Alkaline_Phosphotase
0.00
0.02
0.04
0.06
0.08
0.10
0.12

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:

Predict

Liver Patient Prediction

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