import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear\_model import LogisticRegression

from xgboost import XGBClassifier

from sklearn.metrics import accuracy\_score, roc\_auc\_score, classification\_report

import shap

import matplotlib.pyplot as plt

import seaborn as sns

# Load Dataset

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00225/indian\_liver\_patient.csv"

df = pd.read\_csv(url)

# Handle missing values

df = df.dropna()

# Convert gender to binary

df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})

# Define target and features

X = df.drop('Dataset', axis=1) # Dataset: 1=Liver disease, 2=No liver disease

y = df['Dataset'].apply(lambda x: 1 if x == 1 else 0) # Binary classification

# Train/Test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scale features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Initialize models

models = {

'Logistic Regression': LogisticRegression(),

'Random Forest': RandomForestClassifier(n\_estimators=100, random\_state=42),

'XGBoost': XGBClassifier(use\_label\_encoder=False, eval\_metric='logloss')

}

# Train and evaluate

for name, model in models.items():

model.fit(X\_train\_scaled, y\_train)

y\_pred = model.predict(X\_test\_scaled)

acc = accuracy\_score(y\_test, y\_pred)

auc = roc\_auc\_score(y\_test, model.predict\_proba(X\_test\_scaled)[:, 1])

print(f"\n{name}")

print("Accuracy:", round(acc, 4))

print("ROC-AUC:", round(auc, 4))

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

# SHAP for XGBoost

explainer = shap.Explainer(models['XGBoost'], X\_train\_scaled)

shap\_values = explainer(X\_test\_scaled)

output

# Plot SHAP summary

shap.summary\_plot(shap\_values, X\_test, feature\_names=X.columns)

Logistic Regression

Accuracy: 0.7273

ROC-AUC: 0.7723

Classification Report:

precision recall f1-score support

0 0.67 0.60 0.63 30

1 0.75 0.81 0.78 51

accuracy 0.73 81

macro avg 0.71 0.71 0.71 81

weighted avg 0.72 0.73 0.72 81

Random Forest

Accuracy: 0.7901

ROC-AUC: 0.8446

Classification Report:

precision recall f1-score support

0 0.79 0.67 0.73 30

1 0.79 0.88 0.83 51

accuracy 0.79 81

macro avg 0.79 0.77 0.78 81

weighted avg 0.79 0.79 0.79 81

XGBoost

Accuracy: 0.8025

ROC-AUC: 0.8607

Classification Report:

precision recall f1-score support

0 0.79 0.67 0.72 30

1 0.81 0.88 0.84 51

accuracy 0.80 81

macro avg 0.80 0.78 0.78 81

weighted avg 0.80 0.80 0.80 81

final link

https://github.com/aswini10thclass-gif/Predicting-Liver-Care/upload/main