**#from tabulate import tabulate**

**import pandas as pd**

**a=pd.read\_excel(r"C:\Users\vennela\Documents\miniproject2/traindata1.xlsx",header=0)**

**a["Gender of the patient"]=a["Gender of the patient"].replace("Female",0)**

**a["Gender of the patient"]=a["Gender of the patient"].replace("Male",1)**

**print(a.isnull().sum())**

**a=a.dropna()**

**#col=a.columns**

**#print(col)**

**#a["Albumin\_and\_Globulin\_Ratio"]=a["Albumin\_and\_Globulin\_Ratio"].fillna(0)**

**print(a)**

**a.hist(figsize=(14,11))**

**print(a.isnull().sum())**

**import numpy as np**

**#from tabulate import tabulate**

**import matplotlib.pyplot as plt**

**from sklearn import preprocessing**

**x=a.iloc[:,2:-1]**

**print(x)**

**y=np.array(a.iloc[:,10:])**

**y=np.ravel(y)**

**print(y)**

**from sklearn.linear\_model import LogisticRegression**

**from sklearn.metrics import confusion\_matrix,accuracy\_score,classification\_report**

**classifier = LogisticRegression(solver='lbfgs',random\_state=0,max\_iter=10000)**

**#print(classifier)**

**classifier.fit(X\_train, Y\_train)**

**predicted\_y = classifier.predict(X\_test)**

**print(predicted\_y)**

**y\_test\_h=np.array([0.7,0.1,187,16,18,6.8,3.3,0.9])**

**y\_test\_h=y\_test\_h.reshape(1,8)**

**print(y\_test\_h)**

**h=classifier.predict(y\_test\_h)**

**print(h)**

**cm = confusion\_matrix(Y\_test, predicted\_y)**

**print(cm)**

**print(accuracy\_score(predicted\_y,Y\_test))**

**print(classification\_report(predicted\_y,Y\_test))**

**output**

**[1 1 1 ... 1 1 1]**

**[[7.00e-01 1.00e-01 1.87e+02 1.60e+01 1.80e+01 6.80e+00 3.30e+00 9.00e-01]]**

**[1]**

**[[4595 231]**

**[1667 297]]**

**0.7204712812960236**

**precision recall f1-score support**

**1 0.95 0.73 0.83 6262**

**2 0.15 0.56 0.24 528**

**accuracy 0.72 6790**

**macro avg 0.55 0.65 0.53 6790**

**weighted avg 0.89 0.72 0.78 6790**

**# Decision Tree**

**from sklearn import tree**

**from sklearn.tree import DecisionTreeClassifier**

**clf\_d = DecisionTreeClassifier()**

**clf\_d.fit(X\_train, Y\_train)**

**y\_predict=clf\_d.predict(X\_test)**

**print(confusion\_matrix(Y\_test,y\_predict))**

**print(accuracy\_score(y\_predict,Y\_test))**

**print(classification\_report(y\_predict,Y\_test))**

**[[4824 2]**

**[ 1 1963]]**

**0.9995581737849779**

**precision recall f1-score support**

**1 1.00 1.00 1.00 4825**

**2 1.00 1.00 1.00 1965**

**accuracy 1.00 6790**

**macro avg 1.00 1.00 1.00 6790**

**weighted avg 1.00 1.00 1.00 6790**

**from sklearn.ensemble import RandomForestClassifier**

**from sklearn.metrics import confusion\_matrix,accuracy\_score,classification\_report**

**clf\_r=RandomForestClassifier(n\_estimators=500)**

**clf\_r.fit(X\_train,Y\_train)**

**y\_pred\_r=clf\_r.predict(X\_test)**

**y\_test\_h=np.array([0.7,0.1,187,16,18,6.8,3.3,0.9])**

**y\_test\_h=y\_test\_h.reshape(1,8)**

**h=clf\_r.predict(y\_test\_h)**

**print(h)**

**print(accuracy\_score(y\_pred\_r,Y\_test))**

**print(classification\_report(y\_pred\_r,Y\_test))**

**[1]**

**0.9998527245949926**

**precision recall f1-score support**

**1 1.00 1.00 1.00 4827**

**2 1.00 1.00 1.00 1963**

**accuracy 1.00 6790**

**macro avg 1.00 1.00 1.00 6790**

**weighted avg 1.00 1.00 1.00 6790**

**from sklearn.svm import SVC**

**from sklearn.metrics import confusion\_matrix,accuracy\_score,classification\_report**

**clf\_s = SVC(random\_state=7,gamma="auto")**

**clf\_s.fit(X\_train,Y\_train)**

**y\_pred\_s=clf\_s.predict(X\_test)**

**print(accuracy\_score(y\_pred\_s,Y\_test))**

**print(classification\_report(y\_pred\_s,Y\_test))**

**0.9998527245949926**

**precision recall f1-score support**

**1 1.00 1.00 1.00 4827**

**2 1.00 1.00 1.00 1963**

**accuracy 1.00 6790**

**macro avg 1.00 1.00 1.00 6790**

**weighted avg 1.00 1.00 1.00 6790**