```
#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