**Bahria University,**

Karachi Campus



## LAB EXPERIMENT NO.

## 5

## LIST OF TASKS

|  |  |
| --- | --- |
| **TASK NO** | **OBJECTIVE** |
| **1** | Using python implement Naïve Bayes with two different splitting ratios on Heart Attack Analysis & prediction dataset to predict the chances of heart failure in a person and performed the following steps: ▪ Data Pre-processing step ▪ Fitting Naive Bayes to the Training set ▪ Predicting the test result ▪ Test accuracy of the result(Creation of Confusion matrix) ▪ Visualizing the test set result. ▪ Compare the accuracies |
| **2** | Design a workflow with the help of Knime to predict whether a user buys a product by clicking the ad on the site based on their salary, age, and gender dataset provided in the lab (i.e. Social network ad dataset). |

**Submitted On:**

18 feb 2024

(Date: DD/MM/YY)

**TASK 1: Using python implement Naïve Bayes with two different splitting ratios on Heart Attack Analysis & prediction dataset to predict the chances of heart failure in a person and performed the following steps: ▪ Data Pre-processing step ▪ Fitting Naive Bayes to the Training set ▪ Predicting the test result ▪ Test accuracy of the result(Creation of Confusion matrix) ▪ Visualizing the test set result. ▪ Compare the accuracies**

**SOLUTION:**

import pandas as pd

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

from sklearn.preprocessing import StandardScaler

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import confusion\_matrix, accuracy\_score

import matplotlib.pyplot as plt

import numpy as np

file\_path = 'heart\_failure\_clinical\_records\_dataset.csv'

data = pd.read\_csv(file\_path)

data.head(), data.info(), data.describe()

X = data.drop("DEATH\_EVENT", axis=1)

y = data["DEATH\_EVENT"]

scaler = StandardScaler()

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

X\_train\_70, X\_test\_70, y\_train\_70, y\_test\_70 = train\_test\_split(X\_scaled, y, test\_size=0.3, random\_state=42)

X\_train\_80, X\_test\_80, y\_train\_80, y\_test\_80 = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

def train\_evaluate\_nb(X\_train, X\_test, y\_train, y\_test):

classifier = GaussianNB()

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

cm = confusion\_matrix(y\_test, y\_pred)

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

return cm, accuracy, y\_pred

cm\_70, accuracy\_70, \_ = train\_evaluate\_nb(X\_train\_70, X\_test\_70, y\_train\_70, y\_test\_70)

cm\_80, accuracy\_80, \_ = train\_evaluate\_nb(X\_train\_80, X\_test\_80, y\_train\_80, y\_test\_80)

cm\_70, accuracy\_70, cm\_80, accuracy\_80

def plot\_confusion\_matrix(cm, title='Confusion Matrix'):

plt.figure(figsize=(5,4))

plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)

plt.title(title)

plt.colorbar()

tick\_marks = np.arange(2)

plt.xticks(tick\_marks, ['No Heart Failure', 'Heart Failure'], rotation=45)

plt.yticks(tick\_marks, ['No Heart Failure', 'Heart Failure'])

plt.tight\_layout()

plt.ylabel('True label')

plt.xlabel('Predicted label')

for i in range(cm.shape[0]):

for j in range(cm.shape[1]):

plt.text(j, i, cm[i, j], horizontalalignment="center", color="white" if cm[i, j] > cm.max() / 2. else "black")

plt.show()

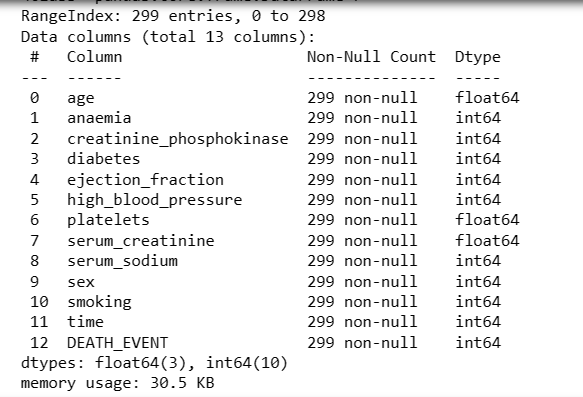
plot\_confusion\_matrix(cm\_70, title='Confusion Matrix (70:30 Split)')

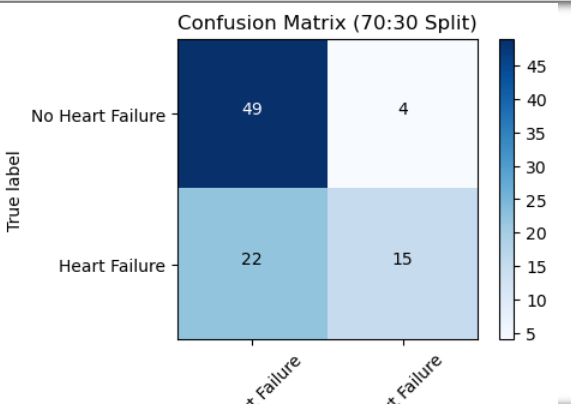
plot\_confusion\_matrix(cm\_80, title='Confusion Matrix (80:20 Split)')

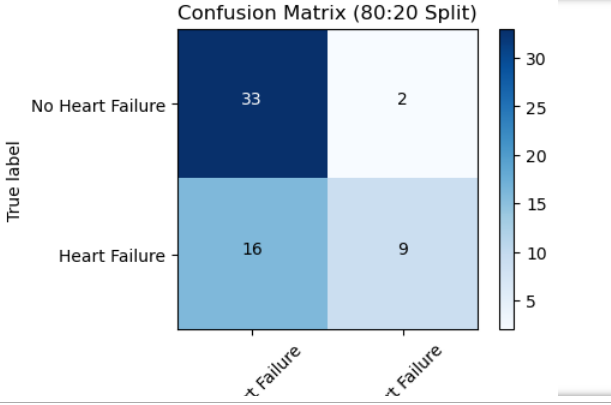
plot\_confusion\_matrix(cm\_70, title='Confusion Matrix (70:30 Split)')

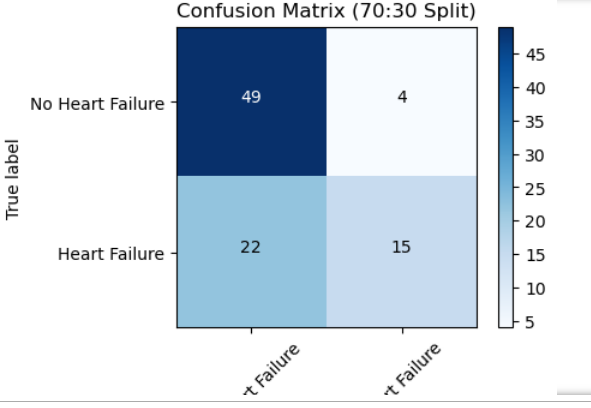
plot\_confusion\_matrix(cm\_80, title='Confusion Matrix (80:20 Split)')

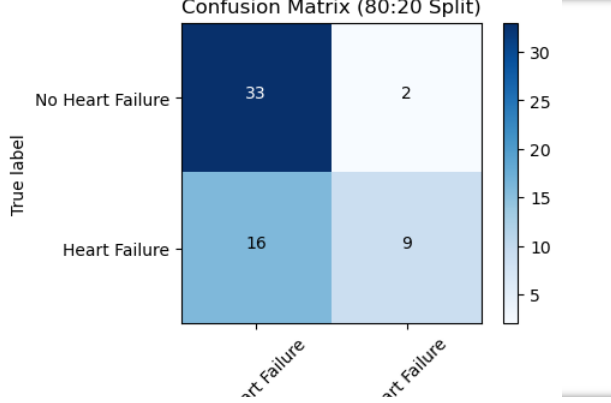
**OUTPUT:**

****

****

****

****

****

**TASK 2:** **Design a workflow with the help of Knime to predict whether a user buys a product by clicking the ad on the site based on their salary, age, and gender dataset provided in the lab (i.e. Social network ad dataset).**

