PUBLIC HEALTHCARE AWARNESS CAMPAIGN

3rd – AI&DS

M.A.M COLLEGE OF ENGINEERING

TEAM MEMBERS,

A.REJOE BLESY

S.SIVA SANKARI

N.YASMEEN NILOFAR

M.SEEMA

A.AYISHA

PROGRAM:

random

import time

# Simulate a smartwatch import measuring pulse rate

def measure\_pulse\_rate():

return random.randint(60, 120) # Simulated pulse rate data

def alert\_high\_pulse\_rate(threshold):

while True:

pulse\_rate = measure\_pulse\_rate()

print(f"Current Pulse Rate: {pulse\_rate} BPM")

if pulse\_rate > threshold:

print(f"High Pulse Rate Alert! Pulse rate is above {threshold} BPM.")

# Add code here to send an alert to the user (e.g., notifications)

time.sleep(10) # Adjust the interval as needed

if \_name\_ == "\_main\_":

threshold = 100 # Set the pulse rate threshold for alerting

alert\_high\_pulse\_rate(threshold)

PROGRAM:

# Import necessary libraries

import pandas as pd

import numpy as np

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

from sklearn.ensemble import RandomForestClassifier

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

# Load your medical dataset

data = pd.read\_csv('medical\_data.csv') # Replace 'medical\_data.csv' with your dataset file

# Data preprocessing

# Handle missing values, encode categorical variables, etc.

# Define features (X) and target (y)

X = data.drop('target\_variable', axis=1) # Adjust 'target\_variable' to the name of your target column

y = data['target\_variable']

# Split data into training and testing sets

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

# Build a machine learning model (Random Forest in this example)

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Train the model

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

# Make predictions on the test set

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

# Evaluate the model

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

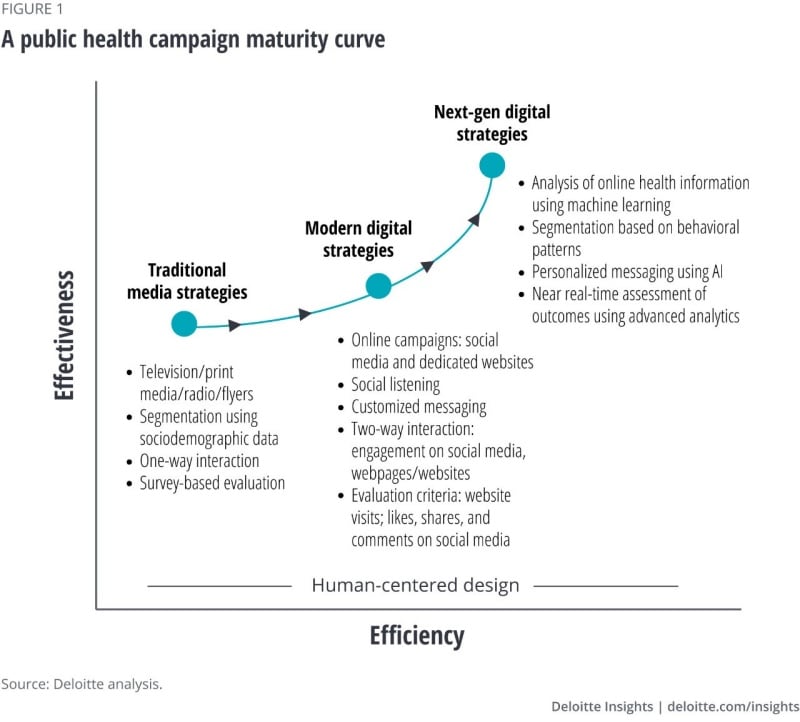
report = classification\_report(y\_test, y\_pred)

# Print results

print(f'Accuracy: {accuracy}')

print(report)

OUTPUT:



OUTPUT:

