TITLE 4

NAME: Y.YAMINI REDDY

REG NO: 192211153

SUBJECT CODE: SPIC409

GUIDE NAME: DR.D.JAYALAKSHMI

APPROVAL ID: SSE\_26\_11\_153\_4 CODE

import pandas as pd

import numpy as np

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

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

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

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder, StandardScaler

from sklearn.pipeline import Pipeline

csv\_path = "C:/Users/Arshad/Desktop/yamini main.csv"

df = pd.read\_csv(csv\_path)

selected\_features = ['glaucoma', 'age', 'ocular\_pressure', 'Mean\_deviation',

'PSD', 'GHT', 'cornea\_thickness', 'RNFL4.mean']

features = df[selected\_features]

labels = df['glaucoma']

categorical\_features = ['GHT']

preprocessor = ColumnTransformer(

transformers=[

('num', StandardScaler(), features.select\_dtypes(include='number').columns),

('cat', OneHotEncoder(handle\_unknown='ignore'), categorical\_features)

])

noise = np.random.normal(0, 0.65, size=features.select\_dtypes(include='number').shape)

features\_noisy = features.select\_dtypes(include='number') + noise

svm\_pipeline = Pipeline([

('preprocessor', preprocessor),

('classifier', SVC(C=0.15, kernel='rbf', gamma='scale', random\_state=92))

])

knn\_pipeline = Pipeline([

('preprocessor', preprocessor),

('classifier', KNeighborsClassifier(n\_neighbors=5))

])

X\_train\_svm, X\_test\_svm, y\_train\_svm, y\_test\_svm = train\_test\_split(features\_noisy, labels, test\_size=0.85, random\_state=42)

svm\_pipeline.fit(X\_train\_svm, y\_train\_svm)

svm\_predictions = svm\_pipeline.predict(X\_test\_svm)

svm\_accuracy = accuracy\_score(y\_test\_svm, svm\_predictions)

print(f"SVM Accuracy: {svm\_accuracy}")

print("Classification Report (SVM):")

print(classification\_report(y\_test\_svm, svm\_predictions))

print("Confusion Matrix (SVM):")

print(confusion\_matrix(y\_test\_svm, svm\_predictions))

X\_train\_knn, X\_test\_knn, y\_train\_knn, y\_test\_knn = train\_test\_split(features\_noisy, labels, test\_size=0.85, random\_state=42)

knn\_pipeline.fit(X\_train\_knn, y\_train\_knn)

knn\_predictions = knn\_pipeline.predict(X\_test\_knn)

knn\_accuracy = accuracy\_score(y\_test\_knn, knn\_predictions)

print(f"KNN Accuracy: {knn\_accuracy}")

print("Classification Report (KNN):")

print(classification\_report(y\_test\_knn, knn\_predictions))

print("Confusion Matrix (KNN):")

print(confusion\_matrix(y\_test\_knn, knn\_predictions))