# Import necessary libraries

import pandas as pd

import numpy as np

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

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

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

from tensorflow import keras

from tensorflow.keras import layers

# Step 1: Load the data

data = pd.read\_csv('student\_data.csv') # Load your dataset

# Step 2: Data preprocessing

# Handling missing values

data.fillna(data.mean(), inplace=True)

# Step 3: Feature Engineering

# Let's assume we have features such as attendance, grades, and behavior scores

features = data[['attendance', 'grades', 'behavior\_score']].values

labels = data['behavior\_change'].values # This should be your target variable

# Step 4: Split the dataset

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

# Step 5: Feature Scaling

scaler = StandardScaler()

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

X\_test = scaler.transform(X\_test)

# Step 6: Model Training with Random Forest

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

rf\_model.fit(X\_train, y\_train)

# Step 7: Model Evaluation

y\_pred\_rf = rf\_model.predict(X\_test)

print(confusion\_matrix(y\_test, y\_pred\_rf))

print(classification\_report(y\_test, y\_pred\_rf))

# Step 8: (Optional) Neural Network Model

nn\_model = keras.Sequential([

layers.Dense(32, activation='relu', input\_shape=(X\_train.shape[1],)),

layers.Dense(16, activation='relu'),

layers.Dense(1, activation='sigmoid') # Binary classification

])

nn\_model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

nn\_model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)

# Evaluate Neural Network

y\_pred\_nn = (nn\_model.predict(X\_test) > 0.5).astype("int32")

print(confusion\_matrix(y\_test, y\_pred\_nn))

print(classification\_report(y\_test, y\_pred\_nn))