*#Ai coach app*

import numpy as np

import pandas as pd

import tensorflow as tf

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

from sklearn.preprocessing import LabelEncoder

*# Load and preprocess the data*

data = pd.read\_csv("life\_coaching\_data.csv")

questions = data['Question'].valuess

answers = data['Answer'].values

*# Encode the labels*

encoder = LabelEncoder()

answers = encoder.fit\_transform(answers)

*# Split the data into training and testing sets*

questions\_train, questions\_test, answers\_train, answers\_test = train\_test\_split(questions, answers, test\_size=0.2, random\_state=42)

*# Create a TensorFlow model*

model = tf.keras.Sequential([

    tf.keras.layers.Embedding(input\_dim=len(encoder.classes\_), output\_dim=64),

    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64)),

    tf.keras.layers.Dense(64, activation='relu'),

    tf.keras.layers.Dense(len(encoder.classes\_), activation='softmax')

])

*# Compile the model*

model.compile(loss='sparse\_categorical\_crossentropy', optimizer='adam', metrics=['accuracy'])

*# Train the model*

history = model.fit(questions\_train, answers\_train, epochs=10, validation\_data=(questions\_test, answers\_test))

*# Evaluate the model*

test\_loss, test\_acc = model.evaluate(questions\_test, answers\_test, verbose=2)

print('\nTest Accuracy:', test\_acc)