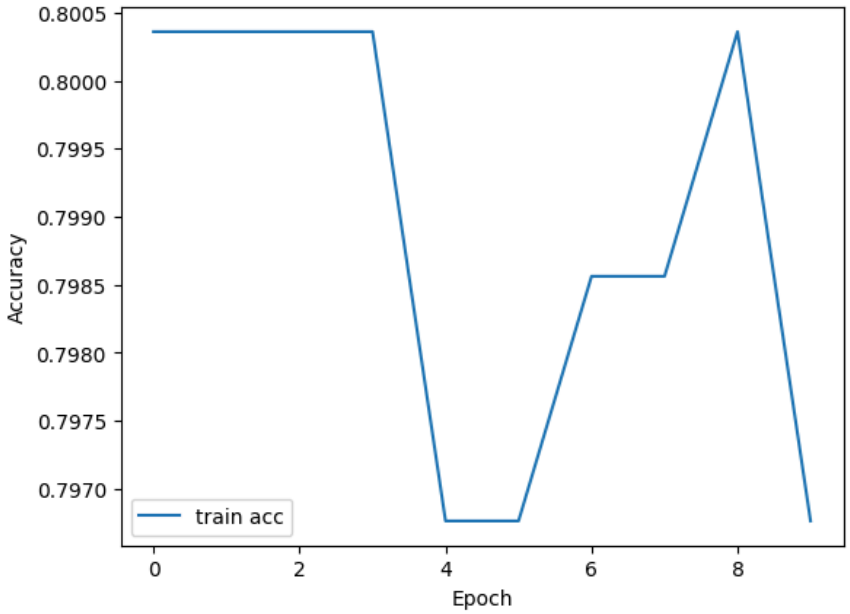
AI Model for Neural Network

```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
df= pd.read_csv('travel_data.csv');

# String colums dataset 80% train and 20% test data
def prepare_data_for_model(df, target_column='booking_made'):

    y = df[target_column]
    X = df.drop(columns=[target_column])

    X = pd.get_dummies(X, drop_first=True)

    # Scale numeric features
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    X_train, X_test, y_train, y_test = train_test_split(
        X_scaled, y, test_size=0.2, random_state=42
    )

    return X_train, X_test, y_train, y_test

def prepare_data_for_model_x_y(df, target_column='booking_made'):


    y = df[target_column]
    X = df.drop(columns=[target_column])

    X = pd.get_dummies(X, drop_first=True)

    # Scale numeric features
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    return X_scaled, y



import tensorflow as tf

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

# Prepare data
X_train, X_test, y_train, y_test = prepare_data_for_model(df,target_column='booking_made')

# Define model
model = Sequential([
    Dense(256, activation='sigmoid', input_shape=(X_train.shape[1],)),
    Dropout(0.3),
    Dense(256, activation='sigmoid'),
    Dropout(0.2),
    Dense(1, activation='sigmoid')  # binary classification
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

print('\n \n accuracy and lost results on train dataset using nurel network algorithim \n')

history = model.fit(X_train,y_train,epochs=10)

import matplotlib.pyplot as plt
plt.plot(history.history['accuracy'], label='train acc')
plt.xlabel('Epoch')
```

```
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

print('\n \n accuracy and lost results on test dataset using nurel network algorithim \n')
model.evaluate(X_test, y_test)
```

```
 accuracy and lost results on train dataset using nurel network algorithim

Epoch 1/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.8271 - loss: 0.4938
Epoch 2/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8123 - loss: 0.5037
Epoch 3/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.8232 - loss: 0.4792
Epoch 4/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.7975 - loss: 0.5321
Epoch 5/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7808 - loss: 0.5239
Epoch 6/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.8182 - loss: 0.4960
Epoch 7/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7653 - loss: 0.5288
Epoch 8/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8043 - loss: 0.4760
Epoch 9/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.8012 - loss: 0.4972
Epoch 10/10
18/18 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.7891 - loss: 0.4886
```




```
 accuracy and lost results on test dataset using nurel network algorithim

5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.7431 - loss: 0.5880
[0.5412213206291199, 0.769784152507782]
```