**Khai báo các thư viện để sử dụng.**

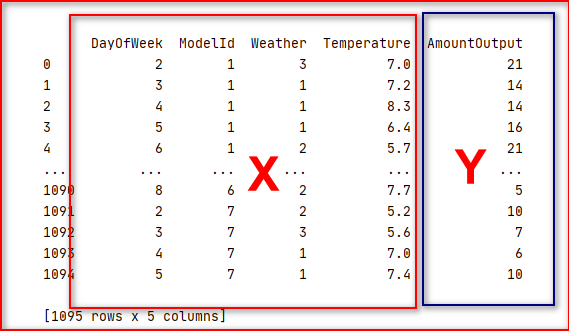
import pandas as pd  
import tensorflow as tf  
from tensorflow import keras  
from tensorflow.keras import layers  
import tensorflow\_docs as tfdocs  
import tensorflow\_docs.modeling  
import os

1. **Khối 1**

**''' Importing the dataset '''**1. xlsx\_file = **'Restaurant\_Data.xlsx'**2. sheet = **"Sheet1"**3. columns = [**'DayOfWeek'**, **'ModelId'**, **'Weather'**, **'Temperature'**, **'AmountOutput'**]  
4. x\_column = [**'DayOfWeek'**, **'ModelId'**, **'Weather'**, **'Temperature'**]  
5. y\_column = [**'AmountOutput'**]  
6. data = pd.read\_excel(xlsx\_file, sheet\_name=sheet)  
7. dataset = data[columns]

**''' Get x and y data'''**8. x = dataset[x\_column]  
9 .y = dataset[y\_column]

1. Dòng 1: đường dẫn chứa file dữ liệu (.xlsx là file excel). Trường hợp file dữ liệu nằm cùng thư mục với file code thì đường dẫn chỉ cần là tên file thì chương trình nó sẽ đọc được. Nếu file dữ liệu không nằm cùng thư mục với file code thì cần thi rõ địa chỉ chứa file dữ liệu thì nó mới nhận được. Ví dụ: C:/User/Restaurant\_Data.xlsx.
2. Dòng 2: Sheet trong file dữ liệu. 1 file excel có thể có nhiều sheet, mỗi sheet là 1 bảng tính khác nhau. Do đó cần khai báo rõ tên sheet chứa data để chương trình đọc được.
3. Dòng 3: các cột dữ liệu dữ liệu cần đọc trong file excel.
4. Dòng 4: các cột dữ liệu đầu vào để dự đoán (DayOfWeek: thứ từ 2 đến 8, ModelId: Mã sản phẩm từ 1 đến 10, Weather: thời tiết, mưa,, nắng, mây, mưa,… kí hiệu từ 1 đến 7, Temperature: nhiệt độ.
5. Dòng 5: cột dự liệu dự đoán (AmountOutput: Dự đoán số lượng sản phẩm bán ra).
6. Dòng 6:, 7 Đọc dữ liệu, lưu vào biến dataset.
7. Dòng 8, 9: Gán dữ liệu vào 2 biến x, y (x chứa 4 cột để dự đoán, y chứa 1 cột là số lượng hàng bán).



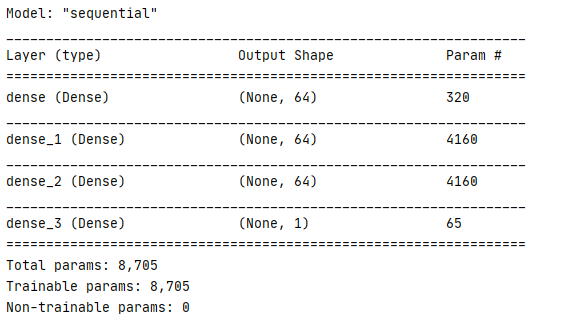
1. **Khối 2**

1. def build\_model():  
2. model = keras.Sequential([  
3. layers.Dense(64, activation=**'relu'**, input\_shape=[len(train\_dataset.keys())]),  
4. layers.Dense(64, activation=**'relu'**),  
5. layers.Dense(64, activation=**'sigmoid'**),  
6. layers.Dense(1)  
 ])  
  
 7. optimizer = tf.keras.optimizers.RMSprop(0.001)  
  
 8. model.compile(loss=**'mse'**,  
 optimizer=optimizer,  
 metrics=[**'mae'**, **'mse'**])  
 9. return model

1. Dòng 1: Khai báo hàm tạo model, tên là build\_model.
2. Dòng 2: tạo biến model, keras.Sequential được coi như một đường ống, sau đó mình sẽ khai báo các layer vào trong đường ống đó theo thứ tự, thì model sẽ huấn luyện dữ liệu theo đúng thứ tự đó.
3. Dòng 3: (Layer 1), Đây là layer đầu tiên trong đường ống, nhận dữ liệu đầu vào (4 cột dữ liệu của biến x). Input\_shape là kích thước của dữ liệu đầu vào, ở đây là 4, (thứ, mã sản phẩm, thời tiết, nhiệt độ.) => train\_dataset.keys()=4. Hàm được sử dụng ở đây là hàm ‘relu’, đây là hàm được sử dụng phổ biến nhất hiện nay.
4. Dòng 4: (Layer 2). Đây là layer thứ 2, cũng là ‘relu’, nói chung không biết dùng hàm gì thì cứ dùng relu cho lành. Số 64 kia chính là số lượng neroun của layer.
5. Dòng 5: (Layer 3). Đây là layer thứ 3, sử dụng ‘sigmoid’ cho mới lạ thôi chứ cũng không quan trọng lắm.
6. Dòng 6: (Layer 4). Đây là layer thứ 4, layer cuối cùng, chính là layer chứ dữ liệu đầu ra, là một số duy nhất (chính là dữ liệu dự báo số lượng hàng bán). Do đó số lượng neuron của nó chỉ là 1.
7. Dòng 7: Optimizer: dùng để chọn thuật toán training (SGD, RMSprop hoặc Adam). Ở đây đang dùng RMSprop.
8. Dòng 8: Truyền các tham số đã vào model để chuẩn bị huấn luyện.
9. Dòng 9: Hàm trả ra model.
10. **Khối 3.**

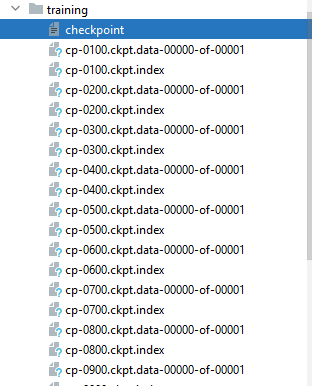
**''' Build model '''**1. model = build\_model()  
  
**''' Print model summary '''**2. print(model.summary())  
  
**''' Include the epoch in the file name (uses `str.format`) '''**3. checkpoint\_path = **"training/cp-{epoch:04d}.ckpt"**4. checkpoint\_dir = os.path.dirname(checkpoint\_path)  
  
**''' Create a callback that saves the model's weights every 5 epochs '''**5. cp\_callback = tf.keras.callbacks.ModelCheckpoint(  
 filepath=checkpoint\_path,  
 verbose=1,  
 save\_weights\_only=True,  
 period=100)  
  
6. EPOCHS = 1000

1. Dòng 1: Tạo model, gọi lại hàm build\_model mình đã khai báo ở trên.
2. Dòng 2: In ra thông tin của model

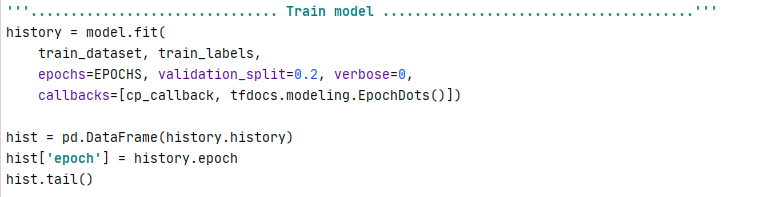


1. Dòng 3, 4: Đường dẫn để lưu model.
2. Dòng 5, 6. Callback, giúp lưu lại các model sau mỗi lần huấn luyện. Ở đây mình đang khai báo epochs=1000 tức là model sẽ train data 1000 lần, sau mỗi lần nó sẽ học thêm được các thông tin, và sẽ cải thiện dần. một vài lần đầu kết quả sẽ khá kém, càng về sau càng tốt đến một ngưỡng nào đó.

(Ví dụ thư mục lưu model)

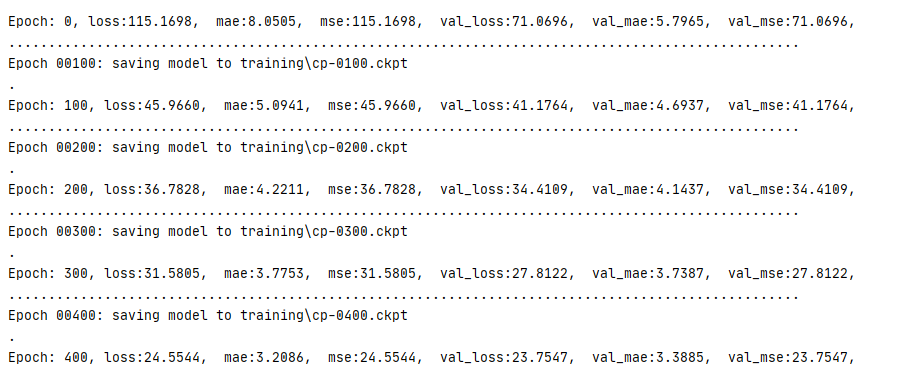


1. **Khối 4**

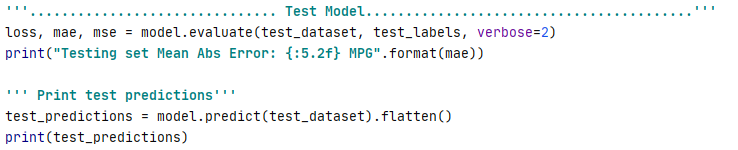


Gọi hàm phương thức fit để bắt đầu huấn luyện model, các tham số đã được khai báo và giải thích bên trên.

Ví dụ trong lúc model huấn luyện như hình bên dưới sẽ thấy loss và mae sẽ giảm dần, mae là mean absolute error, sai số trung bình, sai số càng huấn luyện về sau sẽ càng nhỏ.



1. **Khối 5**



Test model, tính loss, mae, mse, nói chung đây là các độ đo để xem model chuẩn hay không, càng nhỏ càng tốt.



Kết quả đang sai số khoảng 2 – 2.5 sản phẩm, chấp nhận được.