**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



Nhóm 11 gồm các sinh viên:

1. Đỗ Duy Đức – 2151264650
2. Đào Nguyễn Gia Bảo - 2151261233
3. Nguyễn Đỗ Cường - 2151264647
4. Trịnh Đạt - 2151264649

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN PHÂN TÍCH CHUỖI THỜI GIAN**

ĐỀ TÀI: PHÂN TÍCH DỮ LIỆU NHIỆT ĐỘ TẠI NƯỚC TUSINIA

GVHD: TRẦN ANH ĐẠT

Hà Nội, ngày 15 tháng 6 năm 2024

# MỤC LỤC

Contents

[MỤC LỤC 2](#_Toc169447992)

[Phần I. Giới thiệu chung. 4](#_Toc169447993)

[Phần II. Một số mô hình chuỗi thời gian. 6](#_Toc169447994)

[1. Mô hình Prophet 6](#_Toc169447995)

[2. Mô hình LSTM 7](#_Toc169447996)

[3. Mô hình SARIMA 9](#_Toc169447997)

[Phần III. Xây dựng mô hình 10](#_Toc169447998)

[1. Mô tả dữ liệu 10](#_Toc169447999)

[2. HUẤN LUYỆN MÔ HÌNH 11](#_Toc169448000)

[2.1. Prophet 11](#_Toc169448001)

[2.2 SARIMA 13](#_Toc169448002)

[2.3. LSTM 15](#_Toc169448003)

[Phần IV: Đánh giá mô hình 20](#_Toc169448004)

[Phần V: Kết luận 21](#_Toc169448005)

**LỜI NÓI ĐẦU**

Trong bối cảnh biến đổi khí hậu và những thách thức môi trường toàn cầu, việc nghiên cứu và phân tích dữ liệu nhiệt độ trở nên cực kỳ quan trọng. Nước Tunisia, với vị trí địa lý đặc thù và khí hậu đặc trưng của khu vực Bắc Phi, không nằm ngoài tác động của các hiện tượng thời tiết và biến đổi khí hậu toàn cầu. Việc theo dõi và phân tích dữ liệu nhiệt độ tại Tunisia không chỉ giúp hiểu rõ hơn về các xu hướng thời tiết hiện tại mà còn cung cấp cơ sở để dự báo và ứng phó với những biến đổi trong tương lai.

Đề tài "Phân tích dữ liệu nhiệt độ tại nước Tunisia" nhằm mục đích khám phá và làm sáng tỏ các xu hướng nhiệt độ, những biến động và các yếu tố ảnh hưởng đến nhiệt độ tại quốc gia này. Bằng cách sử dụng các mô hình chuỗi thời gian hiện đại, chúng tôi hy vọng sẽ cung cấp được cái nhìn tổng quan về tình hình nhiệt độ tại Tunisia, đồng thời đưa ra những dự báo và khuyến nghị cho việc ứng phó với biến đổi khí hậu.

Đề tài này sẽ tập trung vào việc thu thập, xử lý và phân tích dữ liệu nhiệt độ từ nguồn đáng tin cậy, bao gồm cả dữ liệu lịch sử và hiện tại. Chúng em sẽ sử dụng các công cụ và phương pháp phân tích dữ liệu tiên tiến để khám phá các mẫu và xu hướng, từ đó đưa ra những kết luận và đề xuất thực tiễn cho các nhà hoạch định chính sách, các tổ chức môi trường và cộng đồng địa phương.

Mong rằng kết quả của nghiên cứu này sẽ góp phần vào việc dự đoán về biến đổi khí hậu tại Tunisia, đồng thời hỗ trợ trong việc xây dựng các chiến lược ứng phó hiệu quả, đảm bảo sự phát triển bền vững cho đất nước và khu vực.

# Phần I. Giới thiệu chung.

1. Dữ liệu chuỗi thời gian.

Chuỗi thời gian (time series) là một chuỗi các điểm dữ liệu xảy ra theo thứ tự liên tiếp trong một khoảng thời gian. Một chuỗi thời gian sẽ theo dõi chuyển động của các điểm dữ liệu đã chọn (chẳng hạn như giá của chứng khoán) trong một khoảng thời gian xác định.

Dữ liệu chuỗi thời gian được ứng dụng phổ biến trong đời sống (theo dõi lượng mưa,nhiệt độ,thị trường chứng khoán…) mà còn được sử dụng để theo dõi và tìm ra xu hướng trong các hoạt động trong danh nghiệp theo thời gian.

Việc phân tích dữ liệu chuỗi thời gian sẽ giúp bạn:

* Hiểu đặc điểm của tệp dữ liệu, sự thay đổi của tệp dữ liệu theo thời gian.
* Xác định được những yếu tố ảnh hưởng đến các biến tại các thời điểm khác nhau.
* Hỗ trợ đưa ra dự đoán giá trị tương lai của các biến trong chuỗi thời gian dựa vào xu hướng của dữ liệu trong quá khứ.

1. Các yếu tố cần lưu ý khi phân tích dữ liệu chuỗi thời gian

Một dữ liệu chuỗi thời gian thường bị ảnh hưởng bởi thành 4 thành phần sau:

* Trend(xu hướng): Đề cập đến hướng di chuyển của dữ liệu qua thời gian, có thể là tăng lên hoặc giảm đi ,thường được nhận biết dựa trên độ dốc của đồ thị biểu diễn dữ liệu.

Một số xu hướng phổ biến :

* Upward Trend:  Có xu hướng tăng theo thời gian.
* Downward Trend:  Có xu hướng giảm theo thời gian.
* Horizontal Trend:  Ổn định hoặc biến đổi rất ít theo thời gian.
* Damped Trend:  Giảm dần theo thời gian, nhưng tốc độ giảm đi chậm dần.
* Non-linear Trend:  Biến đổi không tuân theo một mô hình tuyến tính đơn giản mà có các biến đổi phức tạp hơn.
* Seasonality(tính mùa vụ): Đề cập đến các biến động tăng hoặc giảm mà diễn ra đều đặn và lặp lại trong một khoảng thời gian.

Một số tính mùa vụ phổ biến :

* Holiday Seasonality: Thường được gây ra bởi các sự kiện đặc biệt như ngày lễ, sự kiện đặc biệt .
* Weekly Seasonality: Sự thay đổi lặp lại trong khoảng thời gian 7 ngày.
* Monthly Seasonality: Sự thay đổi lặp lại trong khoảng thời gian 30 hoặc 31 ngày.
* Annual Seasonality: Sự thay đổi lặp lại trong khoảng một năm.
* Cycles(tính chu kỳ): Đề cập đến những biến động lên xuống lặp lại, hoặc những thay đổi định kỳ, có thể kéo dài trong nhiều năm và chuyển từ giai đoạn này qua giai đoạn khác.

Sự khác biệt giữa tính mùa vụ và tính chu kỳ :

* Tính mùa vụ đề cập đến sự thay đổi lặp lại , xảy ra trong một khoảng thời gian cố định,bị ảnh hưởng nhiều yếu tố , có thể dự đoán được.
* Tính chu kỳ không bị giới hạn trong một khoảng thời gian cố định và có thể có diễn ra với tần suất khác nhau,nên việc dự đoán sẽ trở nên khó khăn hơn.
* Iregularity (sự bất thường): Đề cập đến những sự thay đổi bất thường của dữ liệu, xảy ra một cách ngẫu nhiên, có thể trái ngược hoàn toàn với các dữ liệu trong quá khứ, khó có thể giải thích và không dự đoán được trước.

Sự bất thường này có thể do sự sai sót trong đo lường dữ liệu, hoặc những sự kiện bất ngờ diễn ra và có thể vô tình làm ảnh hưởng đến tính chính xác khi đánh giá dữ liệu chuỗi thời gian.

# Phần II. Một số mô hình chuỗi thời gian.

## Mô hình Prophet

Prophet hay còn được gọi là Facebook Prophet là một mô hình dự báo dữ liệu chuỗi thời gian dựa trên mô hình cộng (Additive model) trong đó các xu hướng phi tuyến tính phù hợp với thời vụ hàng năm, hàng tuần và hàng ngày, cộng với các ảnh hưởng ngày lễ. Cơ sở của mô hình Prophet là phân rã chuỗi thời gian thành 4 thành phần đại diện cho xu hướng, tính chu kỳ, ảnh hưởng của ngày lễ và sai số mô hình theo phương trình:

Trong đó:

- g(t) là hàm đại diện cho xu hướng, sự thay đổi không có tính chu kỳ của

chuỗi thời gian.

- s(t) là hàm đại diện cho sự thay đổi có tính chu kỳ của chuỗi thời gian (ví

dụ: hàng tuần, hàng tháng, hàng năm)

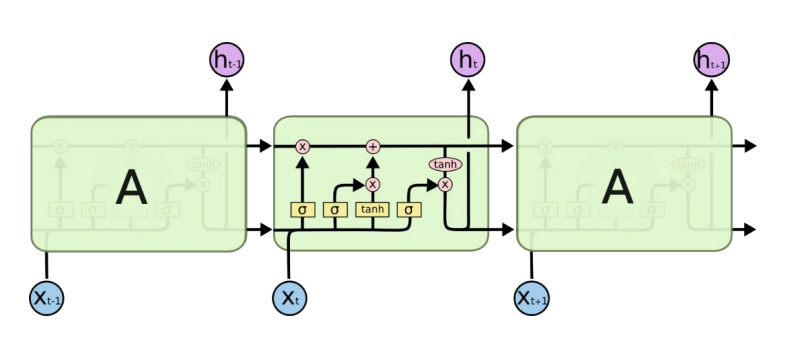
- h(t) là hàm đại diện cho sự ảnh hưởng của ngày nghỉ (do người dùng cung

cấp) xảy ra theo lịch một hoặc vài ngày.

- là sai số mang tính ngẫu nhiên không xác định được của mô hình.

## Mô hình LSTM

Mạng LSTM (Long Short Term Memory) là một biến thể của mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) được thiết kế để giải quyết các bài toán về phụ thuộc xa trong RNN do bị ảnh hưởng của vấn đề triệt tiêu gradient (gradient vanishing). LSTM lần đầu tiên được đề xuất bởi Hochreiter và Schmidhuber và được cải tiến bởi nhiều nhà nghiên cứu khác. Hình 2-4 biểu diễn cấu trúc của một mạng LSTM gồm nhiều tế bào liên kết với nhau thành một chuỗi. Mỗi tế bào LSTM có 4 tầng (tương ứng với 4 hàm kích hoạt sigmoid, tanh) tương tác với nhau để tạo ra trạng thái tế bào (cell state). Hình 2- 1 là cấu trúc bên trong của một tế bào LSTM.



Hình 2.1: Một mạng LSTM và cấu trúc bên trong của tế bào LSTM

Mỗi tế bào có thể ghi nhớ hoặc quên thông tin cần thiết thông qua cơ chế sàng lọc thông tin của các cổng (gate). Một tế bào LSTM có 3 cổng như vậy để duy trì và

điều hành trạng thái của tế bào:

- Cổng quên (forget gate): Có nhiệm vụ loại bỏ những thông tin không cần

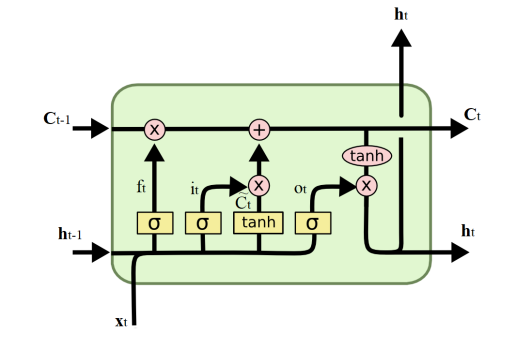
thiết nhận được khỏi trạng thái bên trong tế bào.

- Cổng vào (input gate): Có nhiệm vụ chọn lọc những thông tin cần thiết nào

được thêm vào trạng thái bên trong tế bào.

- Cổng ra (output gate): Có nhiệm vụ xác định những thông tin nào từ trạng

thái bên trong tế bào được sử dụng như đầu ra



Hình 2.2: Cấu trúc của một tế bào LSTM

Về mặt toán học, cơ chế hoạt động của một tế bào LSTM như sau:

- Đầu tiên, tế bào LSTM nhận trạng thái bên trong của bộ nhớ trước đó C𝑡−1 và xác định có quên trạng thái trước đó hay không qua phương trình:

C’𝑡 = Ct-1 ∗ ft (2.1)

Trong đó, cổng quên ft trả về giá trị trong đoạn [0,1], xác định mức độ quên trạng thái bên trong của tế bào trước.

- Song song đó, trạng thái bên trong tế bào hiện tại Ct được cập nhật thêm thông tin đầu vào qua :

Ct = C’t + (it \* Čt) (2.2)

Trong đó, it chứa giá trị kích hoạt cổng vào, xác định mức độ sàng lọc thông tin đầu vào và Čt là giá trị trạng thái tiềm năng của tế bào hiện tại.

- Như vậy, trạng thái bên trong của tế bào hiện tại truyền sang tế bào kế tiếp là:

Ct = Ct-1 \* ft + (it \* Čt) (2.3)

- Cuối cùng, trạng thái đầu ra của tế bào (ht) được tính dựa theo phương trình:

ht = ot \* tanh(Ct) (2.4)

Trong đó, ot là giá trị kích hoạt cổng ra, xác định mức độ sàng lọc thông tin đầu ra của trạng thái bên trong tế bào.

## Mô hình SARIMA

Mô hình trung bình trượt tích hợp tự hồi quy theo mùa (SARIMA hoặc ARIMA theo mùa), là một mô hình mở rộng của ARIMA hỗ trợ dữ liệu chuỗi thời gian đơn biến với thành phần theo mùa. SARIMA bổ sung thêm ba tham số mới là thành phần tự hồi quy (AR), tích hợp (I) và trung bình động (MA) cho thành phần theo mùa của chuỗi thời gian đơn biến, cũng như một tham số xác định khoảng thời gian theo mùa. Việc xác định bộ tham số của mô hình SARIMA yêu cầu chọn các tham số cho cả các yếu tố xu hướng và theo mùa của chuỗi.

Ba tham số xu hướng giống như mô hình ARIMA:

- p: Bậc tự hồi quy xu hướng.

- d: Bậc tích hợp xu hướng

- q: Bậc trung bình động xu hướng.

Bốn tham số theo mùa không phải là một phần của ARIMA phải được xác

định:

- P: Bậc tự hồi quy theo mùa.

- D: Bậc tích hợp theo mùa.

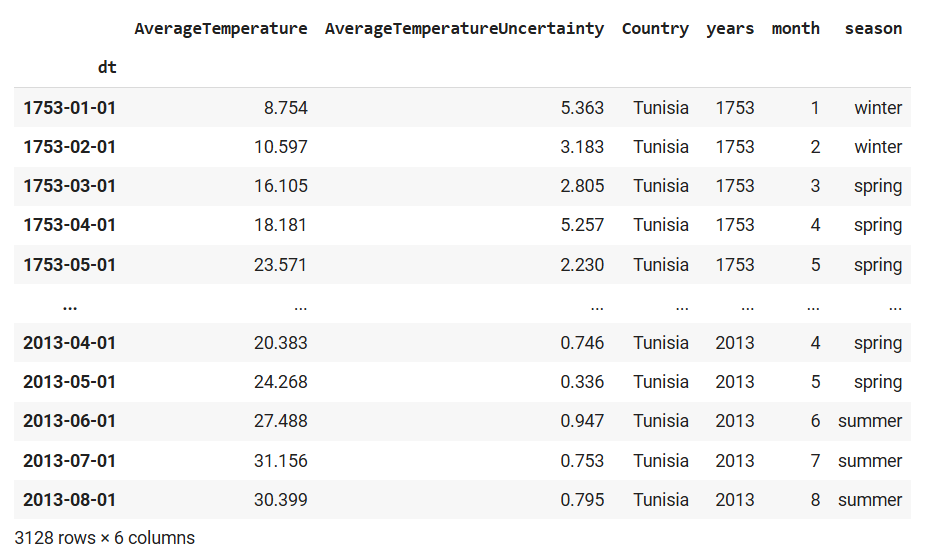
- Q: Bậc trung bình động theo mùa.

- m: Số bước thời gian (kỳ) trong một mùa.

Như vậy, tùy theo giá trị của các tham số mà mô hình SARIMA có thể là mô hình ARIMA, ARMA, AR hay là MA.

# Phần III. Xây dựng mô hình

## Mô tả dữ liệu



Bộ dữ liệu về nhiệt độ tại nước Tusinia được thu thập từ năm 1753 đến 2013.

Bộ dữ liệu của nhóm em gồm 7 cột:  
- dt: Cột dữ liệu về chuỗi thời gian

- AverageTemperature: Cột nhiệt độ trung bình

- AverageTemperatureUncertainty: Cột nhiệt độ trung bình không xác định

- Country: Quốc gia

- years: năm

- month: tháng

- season: mùa

## 2. HUẤN LUYỆN MÔ HÌNH

### 2.1. Prophet

# Khởi tạo mô hình Prophet với các tham số cần thiết

model = Prophet(

seasonality\_mode='multiplicative', # Chế độ mùa vụ

seasonality\_prior\_scale=10.0, # Điều chỉnh độ mượt mà của mùa vụ

yearly\_seasonality=True, # Bật mùa vụ hàng năm

weekly\_seasonality=False, # Tắt mùa vụ hàng tuần

daily\_seasonality=False, # Tắt mùa vụ hàng ngày

interval\_width=0.95 # Độ rộng khoảng tin cậy

)

model.fit(train)

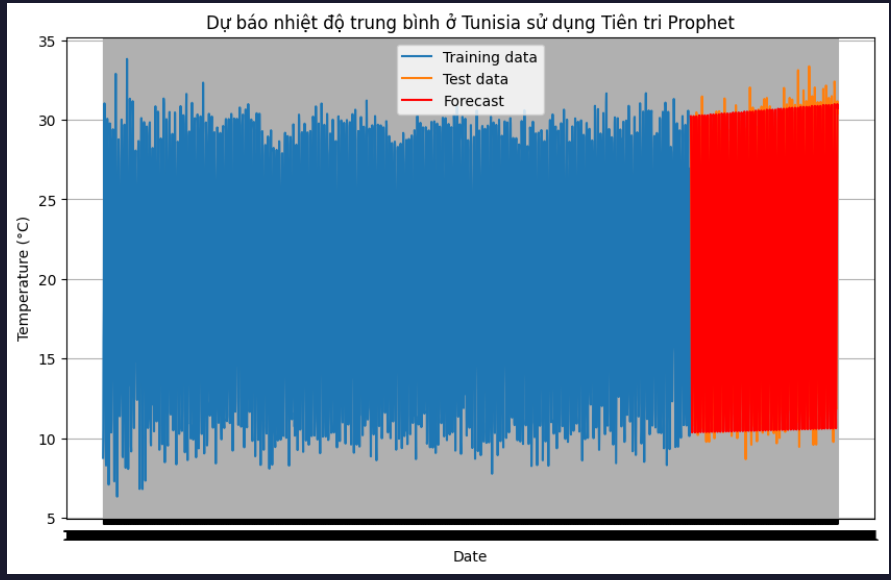
# Thực hiện dự báo trên tập kiểm tra

future = model.make\_future\_dataframe(periods=len(test), freq='MS')

forecast = model.predict(future)

# Lấy các giá trị dự báo cho tập kiểm tra

predicted = forecast['yhat'].iloc[-len(test):].values



seasonality\_mode='multiplicative': Sử dụng chế độ mùa vụ tỷ lệ thuận. Điều này có nghĩa là biến động theo mùa vụ sẽ tỷ lệ thuận với mức độ của chuỗi thời gian.

seasonality\_prior\_scale=10.0: Điều chỉnh độ mượt mà của thành phần mùa vụ. Giá trị 10.0 cho phép thành phần mùa vụ có nhiều biến động hơn.

yearly\_seasonality=True: Bật thành phần mùa vụ hàng năm, tức là mô hình sẽ tính đến chu kỳ biến động hàng năm.

weekly\_seasonality=False: Tắt thành phần mùa vụ hàng tuần, tức là mô hình sẽ không tính đến chu kỳ biến động hàng tuần.

daily\_seasonality=False: Tắt thành phần mùa vụ hàng ngày, tức là mô hình sẽ không tính đến chu kỳ biến động hàng ngày.

interval\_width=0.95: Đặt độ rộng khoảng tin cậy cho dự báo là 95%, nghĩa là khoảng tin cậy của dự báo sẽ bao gồm 95% khả năng giá trị thực sẽ nằm trong đó.

=> Đoạn code trên khởi tạo mô hình Prophet với các tham số cấu hình cụ thể, huấn luyện mô hình với dữ liệu lịch sử, tạo DataFrame cho các mốc thời gian tương lai, thực hiện dự báo, và cuối cùng lấy các giá trị dự báo cho tập kiểm tra. Kết quả này có thể được sử dụng để đánh giá hiệu suất của mô hình dự báo.

### 2.2 SARIMA

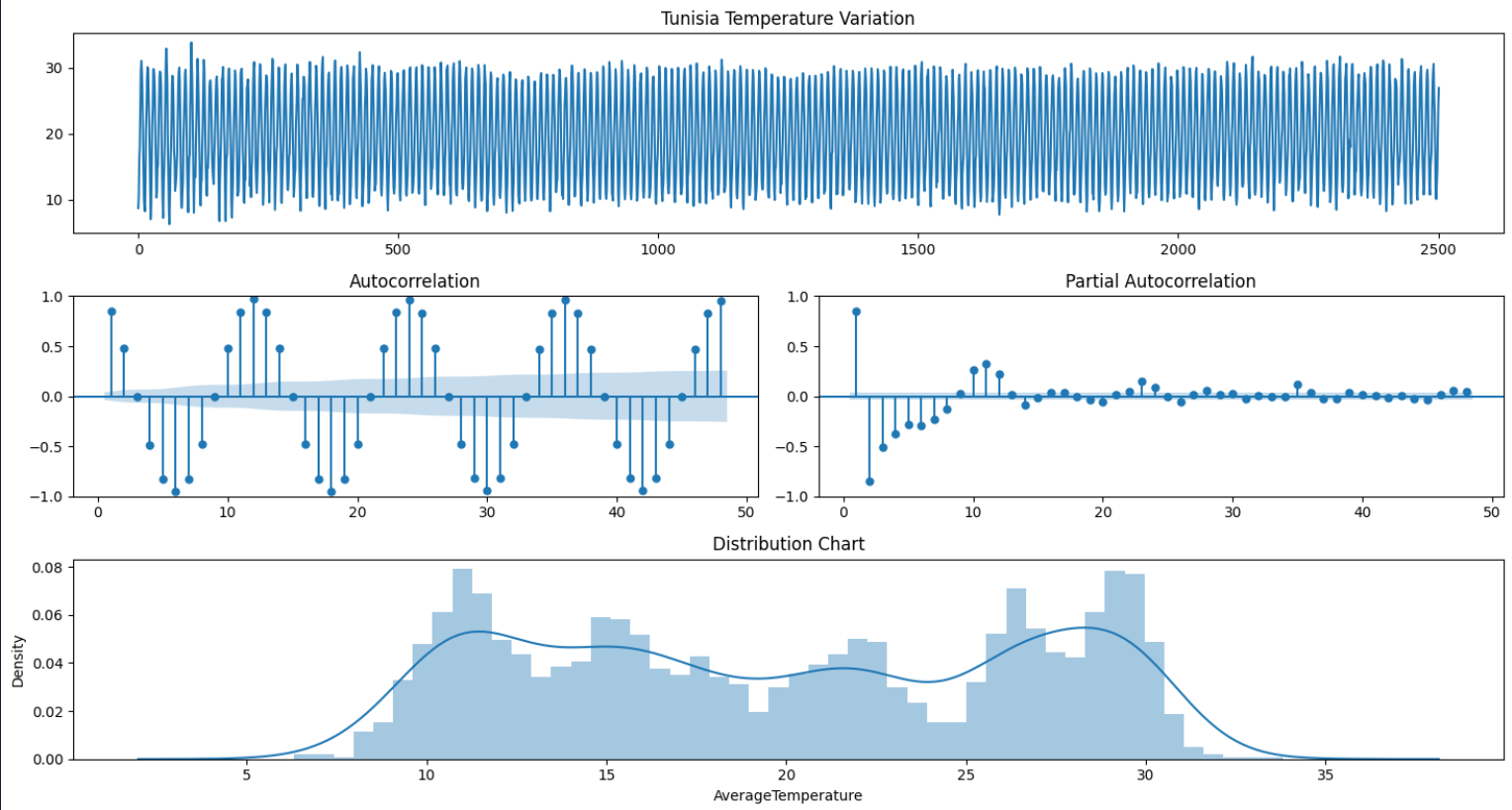
# Fit the ARIMA model on the training data (using SARIMAX for ARIMA functionality)

model = SARIMAX(train['AverageTemperature'], order=(1, 1, 1), seasonal\_order=(0, 1, 1, 12))

results = model.fit()

# Print the model summary

print(results.summary())



SARIMAX(train['AverageTemperature'], order=(1, 1, 1), seasonal\_order=(0, 1, 1, 12)):

train['AverageTemperature']: Dữ liệu huấn luyện, là chuỗi thời gian nhiệt độ trung bình.

order=(1, 1, 1): Các tham số của mô hình ARIMA, bao gồm:

p=1: Bậc của phần tự hồi quy (AR - AutoRegressive).

d=1: Bậc của phần sai phân (I - Integrated) để làm cho chuỗi thời gian trở nên tĩnh.

q=1: Bậc của phần trung bình trượt (MA - Moving Average).

seasonal\_order=(0, 1, 1, 12): Các tham số của thành phần mùa vụ, bao gồm:

P=0: Bậc của phần tự hồi quy theo mùa (Seasonal AR).

D=1: Bậc của phần sai phân theo mùa.

Q=1: Bậc của phần trung bình trượt theo mùa (Seasonal MA).

m=12: Chu kỳ mùa vụ (ở đây là 12, biểu thị chu kỳ hàng năm cho dữ liệu hàng tháng).

Tóm lại:

Khởi tạo mô hình ARIMA bằng cách sử dụng SARIMAX: Đặt các tham số cho phần tự hồi quy, sai phân, và trung bình trượt, cùng với các tham số mùa vụ.

Huấn luyện mô hình với dữ liệu nhiệt độ trung bình: Sử dụng phương pháp tối ưu hóa để tìm các tham số tốt nhất cho mô hình.

In ra bản tóm tắt mô hình: Cung cấp thông tin chi tiết về các tham số, thống kê và hiệu suất của mô hình sau khi huấn luyện.

=> Đoạn code này giúp hiểu rõ hơn về cấu trúc và hiệu suất của mô hình ARIMA khi áp dụng cho chuỗi thời gian nhiệt độ trung bình, đồng thời giúp đánh giá khả năng dự báo của mô hình.

### 2.3. LSTM

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import mean\_squared\_error

from keras.models import Sequential

from keras.layers import LSTM, Dense, Dropout

df = pd.read\_csv('temp\_Tunisia\_new.csv')

# Hàm chuẩn bị dữ liệu cho mô hình LSTM

def prepare\_data(data, look\_back):

X, Y = [], []

for i in range(len(data)-look\_back):

X.append(data[i:(i+look\_back), 0])

Y.append(data[i + look\_back, 0])

return np.array(X), np.array(Y)

# Chọn cột 'AverageTemperature' và chuyển đổi thành numpy array

data = df['AverageTemperature'].values.reshape(-1, 1)

# Chuẩn hóa dữ liệu sang khoảng [0, 1]

scaler = MinMaxScaler(feature\_range=(0, 1))

data\_normalized = scaler.fit\_transform(data)

# Số lượng bước quay lại (look back) cho mô hình LSTM

look\_back = 12 # Ví dụ: Sử dụng 12 tháng trước để dự báo tháng hiện tại

# Chuẩn bị dữ liệu cho mô hình LSTM

X, Y = prepare\_data(data\_normalized, look\_back)

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra

train\_size = int(len(X) \* 0.8)

X\_train, X\_test = X[:train\_size], X[train\_size:]

Y\_train, Y\_test = Y[:train\_size], Y[train\_size:]

# Reshape lại dữ liệu để phù hợp với đầu vào của mạng LSTM (số mẫu, số bước thời gian, số đặc trưng)

X\_train = np.reshape(X\_train, (X\_train.shape[0], X\_train.shape[1], 1))

X\_test = np.reshape(X\_test, (X\_test.shape[0], X\_test.shape[1], 1))

# Xây dựng mô hình LSTM

model = Sequential()

model.add(LSTM(units=50, return\_sequences=True, input\_shape=(look\_back, 1)))

model.add(Dropout(0.2))

model.add(LSTM(units=50))

model.add(Dropout(0.2))

model.add(Dense(units=1))

# Biên dịch mô hình

model.compile(optimizer='adam', loss='mean\_squared\_error')

# Huấn luyện mô hình

history = model.fit(X\_train, Y\_train, epochs=100, batch\_size=32, validation\_data=(X\_test, Y\_test), verbose=2)

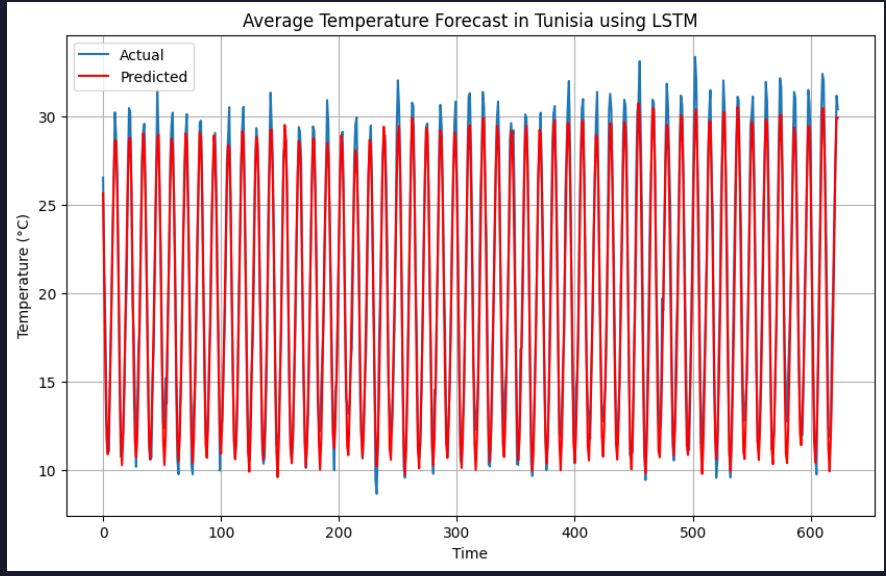
# Dự báo giá trị tiếp theo

predicted = model.predict(X\_test)

# Đưa dự báo về dạng ban đầu

predicted = scaler.inverse\_transform(predicted)

Y\_test = scaler.inverse\_transform([Y\_test])



Import thư viện và đọc dữ liệu:

import numpy as np: Thư viện numpy để xử lý mảng và ma trận.

import pandas as pd: Thư viện pandas để đọc và xử lý dữ liệu từ file CSV.

import matplotlib.pyplot as plt: Thư viện matplotlib để vẽ đồ thị.

from sklearn.preprocessing import MinMaxScaler: Sử dụng MinMaxScaler để chuẩn hóa dữ liệu về khoảng [0, 1].

from sklearn.metrics import mean\_squared\_error: Đánh giá sai số sử dụng MSE (Mean Squared Error).

from keras.models import Sequential: Sequential là mô hình mạng nơ-ron tuần tự trong Keras.

from keras.layers import LSTM, Dense, Dropout: Các lớp mạng nơ-ron cần thiết cho mạng LSTM.

Dữ liệu được đọc từ file 'temp\_Tunisia\_new.csv' vào DataFrame df.

Chuẩn bị dữ liệu cho mô hình LSTM:

prepare\_data(data, look\_back): Hàm này chia dữ liệu thành các cặp (input, output) cho mô hình LSTM. look\_back là số bước thời gian quay lại để dự đoán bước thời gian hiện tại.

Dữ liệu được chọn từ cột 'AverageTemperature' của DataFrame và chuyển thành mảng numpy.

Chuẩn hóa dữ liệu:

MinMaxScaler được áp dụng để chuẩn hóa dữ liệu thành khoảng [0, 1]. Dữ liệu chuẩn hóa là data\_normalized.

Chia dữ liệu thành tập huấn luyện và tập kiểm tra:

Dữ liệu được chia thành X (input) và Y (output) bằng hàm prepare\_data.

train\_size là kích thước của tập huấn luyện (80% dữ liệu), còn lại là tập kiểm tra.

Reshape lại dữ liệu cho mạng LSTM:

Dữ liệu X\_train và X\_test được reshape lại để phù hợp với đầu vào của mạng LSTM với số mẫu, số bước thời gian và số đặc trưng.

Xây dựng mô hình LSTM:

Mô hình Sequential với hai lớp LSTM có 50 đơn vị nơ-ron, và lớp dropout để tránh overfitting.

Lớp cuối cùng là một lớp Dense với 1 đơn vị nơ-ron để đưa ra dự đoán.

Biên dịch và huấn luyện mô hình:

Mô hình được biên dịch với optimizer là 'adam' và hàm loss là 'mean\_squared\_error'.

Mô hình được huấn luyện với X\_train và Y\_train trong 100 epochs và batch size là 32.

Dữ liệu X\_test và Y\_test được sử dụng làm tập validation để đánh giá mô hình trong quá trình huấn luyện.

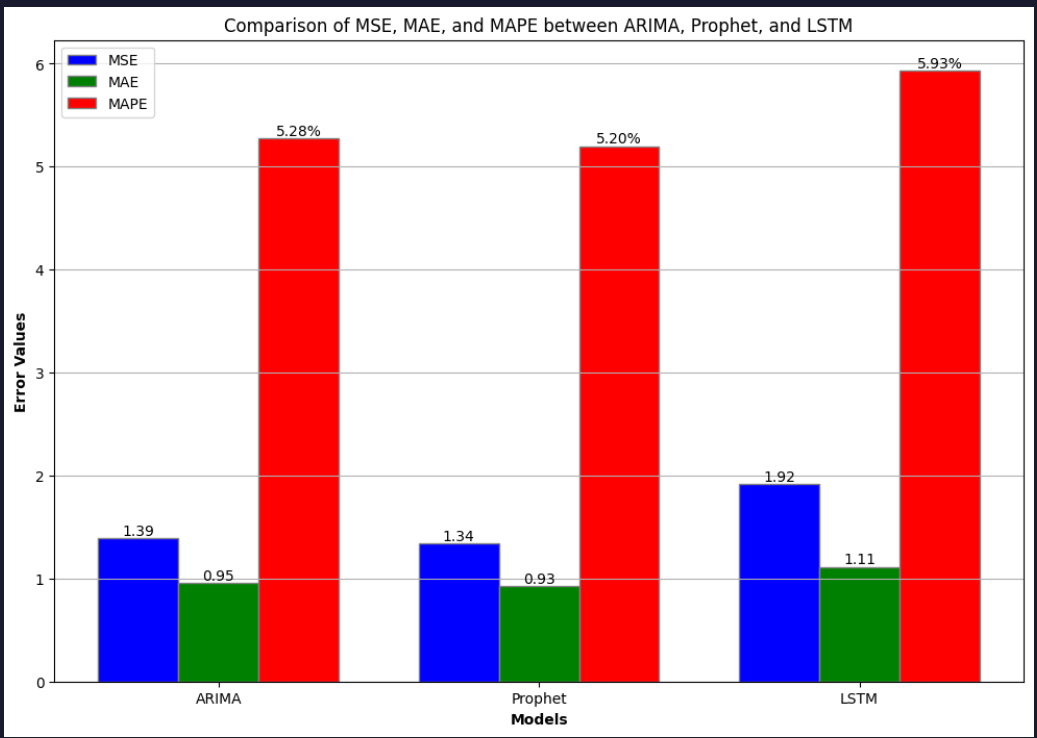
Dự đoán và đánh giá mô hình:

Dữ liệu được dự đoán trên X\_test và sau đó được đưa về dạng ban đầu bằng cách sử dụng scaler.inverse\_transform.

mean\_squared\_error có thể được tính để đánh giá hiệu suất dự đoán của mô hình.

Mã nguồn trên giúp bạn xây dựng và huấn luyện một mô hình LSTM để dự đoán nhiệt độ trung

# Phần IV: Đánh giá mô hình



Từ bảng độ đo đánh giá trên, ta có thể thấy mô hình Phrophet là mô hình có độ chính xác cao nhất

# Phần V: Kết luận

Phần V: Kết luận

Trong bài tập lớn này, chúng tôi đã thực hiện phân tích dữ liệu nhiệt độ tại đất nước Tunisia bằng cách sử dụng một loạt các công cụ và kỹ thuật phân tích dữ liệu. Đây là một tóm tắt về những điều chính mà chúng tôi đã thực hiện và những kết quả đạt được:

Thu thập và khám phá dữ liệu: Đầu tiên, chúng tôi đã thu thập dữ liệu về nhiệt độ trung bình tại Tunisia từ nguồn tin cậy. Sau đó, chúng tôi đã khám phá dữ liệu để hiểu cấu trúc và các đặc điểm quan trọng của nó như các giá trị thiếu, phân bố của dữ liệu, và xu hướng thời gian.

Tiền xử lý dữ liệu: Chúng tôi đã thực hiện các bước tiền xử lý dữ liệu như loại bỏ giá trị thiếu, chuẩn hóa dữ liệu để chuẩn bị cho các bước phân tích và mô hình hóa sau này.

Phân tích thống kê đơn giản: Chúng tôi đã sử dụng các kỹ thuật thống kê đơn giản như biểu đồ dạng đường để phân tích xu hướng, mùa vụ và các biến động trong dữ liệu nhiệt độ.

Xây dựng mô hình dự đoán bằng LSTM: Để dự đoán nhiệt độ trong tương lai, chúng tôi đã sử dụng mô hình LSTM (Long Short-Term Memory) trong Machine Learning. Mô hình này đã được huấn luyện trên dữ liệu lịch sử và sau đó được sử dụng để dự đoán nhiệt độ trong tương lai. Chúng tôi đã sử dụng các kỹ thuật như chuẩn hóa dữ liệu, chia dữ liệu thành tập huấn luyện và kiểm tra, và đánh giá hiệu suất của mô hình bằng các thước đo như Mean Squared Error (MSE).

Kết quả và nhận định: Mô hình LSTM đã đạt được kết quả khả quan trong việc dự đoán nhiệt độ tại Tunisia. Qua đó, chúng tôi có thể đưa ra những nhận định về xu hướng thời tiết, dự báo cho các hoạt động và các ứng dụng khác liên quan đến nhiệt độ.

Tóm lại, bài tập lớn này không chỉ giúp chúng tôi áp dụng và thực hành các kỹ năng phân tích dữ liệu mà còn cung cấp một cái nhìn sâu sắc hơn về biến động của nhiệt độ tại Tunisia và khả năng dự đoán thông qua các mô hình học máy tiên tiến như LSTM. Đây là một bước đầu tiên trong việc áp dụng khoa học dữ liệu vào việc hiểu và dự báo các hiện tượng tự nhiên quan trọng.