

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN**



KHÓA LUẬN TỐT NGHIỆP

**XÂY DỰNG MÔ HÌNH DỰ ĐOÁN NHIỆT ĐỘ CHO BÀI
TOÁN PHÂN TÍCH VÀ DỰ ĐOÁN TRÊN CHUỖI THỜI GIAN**

Giảng viên hướng dẫn: TS.VÕ THỊ HỒNG THẨM

Sinh viên thực hiện: LÊ GIA BẢO

MSSV: 2000003466

Khoá: 2020 - 2024

Ngành/ chuyên ngành: KHOA HỌC DỮ LIỆU

Tp HCM, tháng 1 năm 2024

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN**



KHÓA LUẬN TỐT NGHIỆP

**XÂY DỰNG MÔ HÌNH DỰ ĐOÁN NHIỆT ĐỘ CHO BÀI
TOÁN PHÂN TÍCH VÀ DỰ ĐOÁN TRÊN CHUỖI THỜI GIAN**

Giảng viên hướng dẫn: TS.VÕ THỊ HỒNG THẨM

Sinh viên thực hiện: LÊ GIA BẢO

MSSV: 2000003466

Khoá: 2020 - 2024

Ngành/ chuyên ngành: KHOA HỌC DỮ LIỆU

Tp HCM, tháng 1 năm 2024

LỜI CẢM ƠN

Lời đầu tiên em xin chân thành gửi lời cảm ơn đến các thầy, cô trong khoa Công nghệ thông tin trường Đại học Nguyễn Tất Thành đã tạo điều kiện thuận lợi cho em trong quá trình học tập tại trường cũng như trong thời gian thực hiện đồ án tốt nghiệp. Đặc biệt, em gửi lời cảm ơn tới Tiến sỹ Võ Thị Hồng Thắm - giảng viên trực tiếp hướng dẫn và chỉ bảo giúp em khắc phục những khó khăn, thiếu sót để có thể hoàn thành tốt các phần trong đồ án tốt nghiệp từ phần lý thuyết cho tới phần thực hành.

Mặc dù đã cố gắng với tất cả nỗ lực của bản thân để hoàn thành đồ án. Do thời gian có hạn, năng lực và kinh nghiệm còn hạn chế nên đồ án lần này không tránh khỏi những thiếu sót. Kính mong nhận được sự góp ý từ phía thầy cô để em có thể nâng cao kiến thức của bản thân và hoàn thành đồ án được tốt nhất.

Em xin chân thành cảm ơn

LÊ GIA BẢO

LỜI MỞ ĐẦU

Em tiến hành nghiên cứu sâu rộng, bắt đầu từ việc thu thập và tiền xử lý dữ liệu, phân tích mô hình, đến việc đánh giá và tối ưu hóa để đảm bảo mô hình hoạt động hiệu quả. Mục tiêu của em là cung cấp một công cụ dự đoán mạnh mẽ, có khả năng đối mặt với sự phức tạp của dữ liệu chuỗi thời gian.

Em giới thiệu một loạt các mô hình học máy, từ các mô hình cơ bản đến những mô hình phức tạp như Random Forest, XGBoost, LSTM, và ARIMA. Qua quá trình này, em muốn đánh giá hiệu suất của từng mô hình và xác định mô hình nào phù hợp nhất với bài toán dự đoán nhiệt độ trên chuỗi thời gian.

Phần lớn công việc của em tập trung vào việc tối ưu hóa và đánh giá mô hình. Em sử dụng các phương pháp đánh giá chất lượng dự đoán như Root Mean Squared Error (RMSE) để đảm bảo rằng mô hình của em có khả năng dự đoán chính xác và đáng tin cậy.

Bên cạnh đó, em cũng thực hiện phân tích thông tin thời tiết để hiểu rõ hơn về biến động của dữ liệu. Các biểu đồ và đồ thị sẽ giúp em định rõ các đặc điểm quan trọng trong dữ liệu, từ đó tối ưu hóa mô hình để phản ánh đúng xu hướng thực tế.

Cuối cùng, em hy vọng rằng công việc nghiên cứu này không chỉ là một đóng góp vào lĩnh vực dự báo thời tiết mà còn mang lại những hiểu biết mới về mối liên hệ giữa thời tiết và môi trường. Em tin rằng việc xây dựng một mô hình dự đoán mạnh mẽ có thể đóng góp tích cực cho cả cộng đồng nghiên cứu và ứng dụng trong thực tế.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

1. Hình thức (Bố cục, trình bày, lỗi, các mục, hình, bảng, công thức, phụ lục,)

.....

.....

.....

.....

.....

.....

.....

.....

2. Nội dung (mục tiêu, phương pháp, kết quả, sao chép, các chương, tài liệu,..).....

.....

.....

.....

.....

.....

.....

.....

3. Kết luận.....

.....

TPHCM, Ngày tháng năm 2018

Giáo viên hướng dẫn
(Ký tên, ghi rõ họ tên)

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

1. Hình thức (Bố cục, trình bày, lỗi, các mục, hình, bảng, công thức, phụ lục,)

.....

.....

.....

.....

.....

.....

.....

.....

2. Nội dung (mục tiêu, phương pháp, kết quả, sao chép, các chương, tài liệu,..).....

.....

.....

.....

.....

.....

.....

.....

.....

3. Kết luận.....

.....

TPHCM, Ngày tháng năm 2018

Giáo viên phản biện

(Ký tên, ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN	1
LỜI MỞ ĐẦU	2
NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN.....	3
NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN.....	4
DANH MỤC CÁC TỪ VIẾT TẮT	7
DANH MỤC CÁC HÌNH ẢNH.....	8
CHƯƠNG I:TỔNG QUAN ĐỀ TÀI.....	10
1.1. Giới thiệu đề tài.....	10
1.2. Lý do chọn đề tài.....	10
1.3. Mục tiêu của đề tài	11
1.4. Công nghệ áp dụng	12
CHƯƠNG II:CƠ SỞ LÝ LUẬN VỀ VẤN ĐỀ NGHIÊN CỨU	13
2.1. Lý thuyết về dự báo nhiệt độ	13
2.2. Giới thiệu về dự báo nhiệt độ	13
2.3. Khái niệm về dự báo.....	13
2.4. Dữ liệu chuỗi thời gian.	14
2.5. Công cụ sử dụng	15
2.6. Kỹ thuật sử dụng.....	17
CHƯƠNG III:MÔ HÌNH LÝ THUYẾT	19
3.1. Lý thuyết về Time Series Decomposition	19
3.2. Simple Moving Average.....	19
3.3. Tổng quan về LSTM(Long Short-Term Memory) :.....	21
3.3.1 Khái niệm	21
3.3.2. Cổng quên (Forget gate).....	22
3.3.3. Cổng đầu vào (Input gate)	23
3.3.4. Cổng đầu ra (Output gate)	24
3.3.5. Sơ đồ của mô hình LSTM	25
3.4. Hồi quy tuyến tính (Linear Regression)	26
3.4.1 Khái niệm	26
3.4.2 Sơ đồ của Linear Regression	27
3.5. Cây quyết định (Decision Tree)	27
3.5.1.Khái niệm	27

3.5.2. Sơ đồ của Decision Tree.....	28
3.6. Random Forest.....	28
3.6.1.Khái niệm	28
3.7. XGBoost	29
3.7.1.Khái niệm	29
3.8. ARIMA	30
3.8.1.Khái niệm	30
3.8.2.Sơ đồ mô hình Arima	31
3.9. Root Mean Squared Error (RMSE)	31
CHƯƠNG IV:MÔ HÌNH THỰC NGHIỆM	33
4.1. Mô tả bộ dữ liệu.....	33
4.1.1. Thông tin cung cấp	33
4.1.2. Tại sao nhiệt độ trung bình cửa quan trọng.....	34
4.2. Trực quan hóa dữ liệu.....	35
4.2.1.Thông tin về bộ dữ liệu	36
4.2.2. Tiền xử lý dữ liệu	39
4.3.3.Trực quan hoá dữ liệu.....	43
4.3.Xây dựng mô hình dự đoán	46
CHƯƠNG V:KẾT LUẬN VÀ KIẾN NGHỊ	63
1.Kết luận.....	63
2. Các mục tiêu đưa ra đã hoàn thành	63
3. Kiến nghị	64
DANH MỤC TÀI LIỆU THAM KHẢO	65

DANH MỤC CÁC TỪ VIẾT TẮT

CHỮ VIẾT TẮT	Ý NGHĨA
LSTM	Long Short-Term Memory
STD	Standard Deviation
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
MSE	Mean Squared Error

DANH MỤC CÁC HÌNH ẢNH

Hình 1. Google Colaboratory	15
Hình 2. Kaggle.....	16
Hình 3. Mô hình LSTM.....	22
Hình 4. Cổng quên (Forget gate).....	23
Hình 5. Cổng đầu vào (Input gate)	24
Hình 6. Cổng đầu ra (Output gate)	25
Hình 7. Thông tin về bộ dữ liệu	36
Hình 8. Thông tin về bộ dữ liệu	37
Hình 9. Số lượng các mẫu trong cột Region	37
Hình 10. Số lượng các mẫu trong cột Country và City	38
Hình 11. Xử lý dữ liệu.....	39
Hình 12. Xử lý dữ liệu State.....	39
Hình 13. Xử lý dữ liệu.....	40
Hình 14. Xử lý dữ liệu datetime	41
Hình 15. Biểu đồ boxplot	42
Hình 16. Biểu đồ choropleth	43
Hình 17. Biểu đồ cột về nhiệt độ trung bình	44
Hình 18. Biểu đồ nhiệt độ trung bình từng khu vực.....	45
Hình 19. Biểu đồ Linear Fit.....	46
Hình 20. Chọn dữ liệu	46
Hình 21. Kiểm tra dữ liệu ngày trong tháng.....	46
Hình 22. Rolling mean	47
Hình 23. Rolling STD.....	48
Hình 24. Ma trận tương quan	49
Hình 25. Moving Average	50
Hình 26. Biểu đồ phân tích mùa vụ (1 ngày)	51
Hình 27. Biểu đồ phân tích mùa vụ(365 ngày)	52
Hình 28. Mô hình LSTM.....	53

Hình 29.Mô hình LinearRegression	54
Hình 30.Mô hình Decision Tree	55
Hình 31.Mô hình Random Forest	56
Hình 32.Mô hình XGBoost	57
Hình 33.Mô hình ARIMA	58
Hình 34.So sánh RMSE các mô hình	59
Hình 35.Dự báo 10 ngày tiếp theo bằng LSTM	60
Hình 36.Kết quả dự đoán.....	61

CHƯƠNG I: TỔNG QUAN ĐỀ TÀI

1.1. Giới thiệu đề tài

Nhiệt độ là một trong những yếu tố quan trọng nhất của khí hậu. Nó có ảnh hưởng đến nhiều khía cạnh của đời sống, từ nông nghiệp đến sức khỏe con người. Trong bối cảnh cuộc cách mạng khoa học - công nghệ (cuộc cách mạng công nghiệp lần thứ 4) đang diễn ra mạnh mẽ, việc ứng dụng trí tuệ nhân tạo (AI) và các kỹ thuật tiên tiến vào lĩnh vực dự báo nhiệt độ là xu hướng tất yếu.

Sạt lở, lũ quét, hạn hán... luôn là những vấn đề nhức nhối, là dấu chấm hỏi lớn đặt ra cho nhà nước, chính phủ cũng như các doanh nghiệp tư nhân cần phải đưa ra các giải pháp phù hợp và những cảnh báo cấp thiết tới người dân địa phương để kịp thời ứng phó.

Có thể nói thời gian qua, thị trường công nghệ phục vụ trong lĩnh vực dự báo phần lớn là sự tham gia của các trung tâm, cơ sở nghiên cứu của Nhà nước. Tuy nhiên, nắm bắt được tiềm năng phân khúc của thị trường này, một số đơn vị tư nhân cũng bắt đầu nghiên cứu, cung cấp các ứng dụng công nghệ hữu ích và hiệu quả.

Việc dự báo nhiệt độ chính xác có thể giúp các cơ quan chức năng có những biện pháp ứng phó kịp thời và hiệu quả với các hiện tượng thời tiết cực đoan như bão, lũ lụt, hạn hán,... Từ đó, giúp giảm thiểu thiệt hại về người và tài sản. Giúp người dân và doanh nghiệp có kế hoạch sản xuất, kinh doanh phù hợp với thời tiết qua đó tăng năng xuất lao động. Đồng thời giúp người dân có kế hoạch sinh hoạt, nghỉ ngơi phù hợp với thời tiết qua đó thể giúp bảo vệ sức khỏe và giảm thiểu nguy cơ mắc các bệnh liên quan đến thời tiết.

Dự báo nhiệt độ là một công cụ quan trọng có tác động đến nhiều khía cạnh của cuộc sống. Việc nâng cao độ chính xác của dự báo nhiệt độ là một nhiệm vụ quan trọng cần được thực hiện để giảm thiểu tác động của thiên tai và nâng cao chất lượng cuộc sống.

1.2. Lý do chọn đề tài

Việc lựa chọn đề tài "**XÂY DỰNG MÔ HÌNH DỰ ĐOÁN NHIỆT ĐỘ CHO BÀI TOÁN PHÂN TÍCH VÀ DỰ ĐOÁN TRÊN CHUỖI THỜI GIAN**" là một quyết định mang lại nhiều lợi ích và có ý nghĩa lớn trong nghiên cứu. Một trong những lý do quan trọng là tính ứng dụng và thực tế của đề tài này. Dự đoán nhiệt độ không chỉ quan trọng

trong việc cung cấp thông tin thời tiết chính xác mà còn ảnh hưởng đến nhiều lĩnh vực như nông nghiệp, quản lý nguồn nước, và dự báo thời tiết. Hiểu biết sâu sắc về nhiệt độ có thể đóng vai trò quan trọng trong việc đưa ra quyết định chiến lược trong kế hoạch phát triển và quản lý các nguồn lực tự nhiên.

Đặc biệt, đề tài này đưa ra một thách thức nghiên cứu đầy hứng thú. Dự đoán nhiệt độ đòi hỏi sự hiểu biết sâu sắc về các yếu tố thời tiết, tương tác phức tạp giữa chúng, và khả năng ứng dụng các mô hình học máy hiện đại. Quá trình này không chỉ làm tăng cường kiến thức về dự báo thời tiết mà còn giúp phát triển các phương pháp học máy trong lĩnh vực này.

Mặt khác, đề tài còn đặt ra một tầm quan trọng trong bối cảnh biến đổi khí hậu. Khả năng dự đoán nhiệt độ đồng thời giúp xác định và đối mặt với tác động của thay đổi khí hậu, từ đó hỗ trợ trong việc xây dựng các chiến lược ứng phó và đối phó.

Cuối cùng, kết quả từ nghiên cứu này không chỉ đem lại lợi ích cho việc dự báo thời tiết mà còn mang lại ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Điều này chứng tỏ tầm quan trọng và tiềm năng của đề tài này giúp cung cấp thông tin hữu ích cho quyết định chiến lược và phát triển bền vững.

1.3. Mục tiêu của đề tài

Đề tài " **XÂY DỰNG MÔ HÌNH DỰ ĐOÁN NHIỆT ĐỘ CHO BÀI TOÁN PHÂN TÍCH VÀ DỰ ĐOÁN TRÊN CHUỖI THỜI GIAN** " được lựa chọn với mục tiêu chính là áp dụng và phát triển các mô hình học máy để cải thiện khả năng dự đoán nhiệt độ, một biến số quan trọng trong dự báo thời tiết. Qua quá trình nghiên cứu, em đặt ra một số mục tiêu cụ thể để đảm bảo tính hiệu quả và ứng dụng thực tế của đề tài.

Mục tiêu đầu tiên là xây dựng một loạt các mô hình dự đoán, từ các mô hình cơ bản như Linear Regression và Decision Trees đến các mô hình phức tạp như Random Forests, XGBoost, LSTM, và ARIMA. Việc này giúp kiểm soát và so sánh hiệu suất của từng mô hình trong việc dự đoán nhiệt độ.

Tiếp theo, em tập trung vào tối ưu hóa và đánh giá mô hình, điều chỉnh các tham số để đạt được độ chính xác cao nhất. Sử dụng các phương pháp đánh giá như RMSE, em đảm bảo rằng mô hình đáp ứng được yêu cầu độ chính xác và độ tin cậy trong dự đoán.

Phân tích thông tin thời tiết là bước quan trọng khác trong đề tài. Em sử dụng các biểu đồ và đồ thị để hiểu rõ hơn về xu hướng và mẫu biến động trong dữ liệu, từ đó có thêm thông tin hữu ích trong việc xây dựng mô hình.

Bên cạnh đó, em đặt ra mục tiêu đối mặt với thách thức của chuỗi thời gian, điều này bao gồm sự phụ thuộc lịch sử và mẫu biến động đặc trưng của dữ liệu. Điều này giúp mô hình hiểu rõ hơn về quá khứ để có dự đoán tốt hơn về tương lai.

Cuối cùng, em hướng đến ứng dụng thực tế của mô hình, giúp cung cấp dự báo thời tiết chính xác và hữu ích. Đề xuất sử dụng mô hình để đối mặt với biến đổi khí hậu, làm nền tảng cho quyết định chiến lược và dự báo thị trường trong các điều kiện biến động. Điều này sẽ mang lại những kiến thức quan trọng về mối quan hệ giữa thời tiết và thị trường, cũng như ứng dụng của học máy trong lĩnh vực dự báo thời tiết.

1.4. Công nghệ áp dụng

- Dữ liệu sử dụng: Trang Kaggle
- Code : Google Colab

CHƯƠNG II: CƠ SỞ LÝ LUẬN VỀ VẤN ĐỀ NGHIÊN CỨU

2.1. Lý thuyết về dự báo nhiệt độ

Lý thuyết dự báo nhiệt độ đóng vai trò quan trọng trong lĩnh vực khí tượng học và thời tiết, không chỉ là yếu tố chính cho các tổ chức khí tượng học mà còn ảnh hưởng sâu rộng đến nhiều ngành khác nhau như năng lượng, nông nghiệp, du lịch, và các ngành khác. Để dự báo nhiệt độ, các nhà nghiên cứu thường áp dụng các phương pháp thống kê và mô hình hóa để dự đoán sự biến động của nhiệt độ trong một khoảng thời gian cụ thể. Dữ liệu từ quá khứ, bao gồm thông tin về áp suất không khí, độ ẩm, gió, và các yếu tố khác ảnh hưởng đến nhiệt độ, được sử dụng để xây dựng các mô hình dự báo chính xác.

2.2. Giới thiệu về dự báo nhiệt độ

Dự báo nhiệt độ là quá trình ước lượng và dự đoán giá trị nhiệt độ không khí trong một vùng địa lý cụ thể và trong khoảng thời gian nhất định trong tương lai. Đây là một lĩnh vực quan trọng thuộc lĩnh vực khí tượng học và thời tiết, có ảnh hưởng lớn đến nhiều khía cạnh của cuộc sống hàng ngày và các ngành công nghiệp.

2.3. Khái niệm về dự báo

Luôn đóng vai trò quan trọng trong việc dự đoán và đánh giá các vấn đề về nhiệt độ ở tương lai. Quá trình này chủ yếu dựa trên việc sử dụng dữ liệu hiện tại và quá khứ để tạo ra các ước tính về những thay đổi có thể xảy ra trong tương lai. Trong ngữ cảnh này, dự báo tập trung vào việc dự đoán xu hướng nhiệt độ, biến đổi khí hậu và các yếu tố khác liên quan đến thời tiết.

Quy trình được thể hiện như sau:

Bước 1. Thu thập dữ liệu về nhiệt độ từ các nguồn đáng tin cậy ví dụ như là Kaggle.

Bước 2. Tiền xử lý dữ liệu để xử lý các giá trị thiếu, sử dụng các kỹ thuật như điền giá trị trung bình hoặc giá trị tối đa và trực quan hoá dữ liệu.

Bước 3. Phân chia dữ liệu thành hai phần tập huấn luyện và tập kiểm tra. Tập huấn luyện sẽ dùng cho công việc giúp cho mô hình học tập, trong khi tập kiểm tra dùng trong việc đưa ra đánh giá cho hiệu suất mô hình.

Bước 4. Dùng Sử dụng nhiều mô hình máy học để huấn luyện cho việc dự báo nhiệt độ

Bước 5. Sử dụng tập kiểm tra để tiến hành đánh giá về hiệu suất của dự báo. Kết quả thu được dùng so sánh để xem xét xác định mô hình nào có hiệu suất tốt nhất.

Bước 6. Tinh chỉnh mô hình để cải thiện độ chính xác trong dự đoán.

Bước 7. Chọn mô hình tốt nhất đã được huấn luyện để dự báo lợi nhuận trong tương lai.

2.4. Dữ liệu chuỗi thời gian.

Dữ liệu chuỗi thời gian đóng một vai trò quan trọng và phổ biến trong nhiều lĩnh vực như thống kê, kinh tế học, đặc biệt là trong lĩnh vực khoa học dữ liệu. Loại dữ liệu này ghi lại thông tin qua các khoảng thời gian cố định, giúp theo dõi sự biến động và thay đổi của giá trị quan tâm theo thời gian. Thông qua việc phân tích dữ liệu chuỗi thời gian, chúng ta có cơ hội dự đoán xu hướng, nhận biết chu kỳ, và hiểu rõ biến động trong tương lai, từ đó hỗ trợ quyết định một cách hiệu quả.

Ngoài ra, việc tích hợp yếu tố mùa vụ và các yếu tố thời tiết khác như độ ẩm, áp suất không khí, và gió cũng đóng vai trò quan trọng trong quá trình mô hình hóa. Mô hình dự báo thường cần đảm bảo rằng những yếu tố này được tính đến để cung cấp dự đoán chính xác.

Sau khi xây dựng mô hình, việc kiểm định và đánh giá là bước không thể thiếu để đảm bảo độ chính xác của mô hình trên dữ liệu mới. Các chỉ số đánh giá như Mean Absolute Error, Root Mean Squared Error thường được sử dụng để đánh giá hiệu suất.

Cuối cùng, dữ liệu chuỗi thời gian được áp dụng để dự báo nhiệt độ trong tương lai và hỗ trợ quyết định và lập kế hoạch trong nhiều lĩnh vực như năng lượng, nông nghiệp và quản lý tài nguyên. Việc tối ưu hóa mô hình thông qua tinh chỉnh siêu tham số là một bước quan trọng để cải thiện độ chính xác và ứng dụng hiệu suất cao của mô hình dự báo nhiệt độ.

2.5. Công cụ sử dụng



Hình 1. Google Colaboratory

Giới thiệu:

Google Colab là một dịch vụ cung cấp môi trường lập trình Python trực tuyến, không yêu cầu cài đặt phần mềm ngoại trừ trình duyệt web. Được thiết kế chủ yếu để thực hiện và chia sẻ các tệp Jupyter Notebook, Colab mang lại nhiều ưu điểm đặc sắc.

Đặc điểm chính:

- Colab là dịch vụ đám mây miễn phí từ Google, giúp người dùng thực hiện các dự án học máy và thống kê mà không cần quan tâm đến cấu hình máy tính cá nhân. Môi trường Colab được cung cấp trực tuyến, loại bỏ bất kỳ rắc rối cài đặt nào.
- Colab cung cấp truy cập vào GPU miễn phí, giúp gia tăng tốc quá trình đào tạo mô hình học máy và xử lý dữ liệu lớn. Điều này làm cho việc thực hiện các tác vụ có độ phức tạp cao trở nên dễ dàng và nhanh chóng.
- Colab sử dụng định dạng Jupyter Notebook, cung cấp môi trường linh hoạt để tạo, chỉnh sửa và chia sẻ các tệp notebook. Điều này giúp tương tác mạnh mẽ với mã và kết quả.
- Tích hợp mạnh mẽ với Google Drive, Colab cho phép người dùng lưu trữ và chia sẻ các notebook một cách thuận tiện. Dữ liệu được lưu trữ trực tuyến, giúp quản lý và truy cập dễ dàng.
- Colab hỗ trợ nhiều thư viện và framework phổ biến trong cộng đồng học máy như TensorFlow, PyTorch, và nhiều thư viện khác. Điều này mở rộng khả năng phát triển và nghiên cứu cho người dùng.

kaggle



Hình 2. Kaggle

Giới thiệu:

Kaggle là một nền tảng trực tuyến dành cho cộng đồng Machine Learning (ML) và Khoa học Dữ liệu, được thành lập vào năm 2010 bởi Anthony Goldbloom và Ben Hamner. Nhanh chóng trở thành một trong những nền tảng ML và khoa học dữ liệu hàng đầu trên thế giới, Kaggle cung cấp một loạt các dịch vụ và tài nguyên để hỗ trợ người dùng trong việc học hỏi, phát triển kỹ năng và ứng dụng ML vào thực tế.

Kaggle cung cấp một loạt các dịch vụ và tài nguyên đa dạng như sau:

- **Bộ Dữ Liệu Đa Dạng:** Kaggle sở hữu một thư viện lớn các bộ dữ liệu ML từ nhiều lĩnh vực khác nhau như tài chính, y tế và khí tượng học.
- **Cuộc Thi ML:** Nền tảng tổ chức các cuộc thi ML thường xuyên, tạo cơ hội cho người dùng thử nghiệm kỹ năng, cạnh tranh với người khác trên toàn thế giới.
- **Cộng Đồng Chia Sẻ:** Kaggle có các cộng đồng trực tuyến tích cực, nơi người dùng chia sẻ kiến thức và kinh nghiệm.

Kaggle mang lại nhiều lợi ích cho người dùng:

- **Học Hỏi và Phát Triển:** Kaggle là nguồn tài nguyên tuyệt vời cho việc học hỏi và phát triển kỹ năng ML. Người dùng có cơ hội thực hành và cải thiện kỹ năng thông qua bộ dữ liệu và cuộc thi.
- **Kết Nối Với Cộng Đồng:** Kaggle là môi trường lý tưởng để kết nối với những người khác quan tâm đến ML. Cộng đồng Kaggle cung cấp cơ hội chia sẻ kiến thức, kinh nghiệm và tìm kiếm hỗ trợ.
- **Ứng Dụng Thực Tế:** Kaggle cung cấp cơ hội để người dùng áp dụng ML vào thực tế thông qua các cuộc thi, giúp giải quyết vấn đề thực tế và tạo ra tác động tích cực đối với thế giới.

2.6. Kỹ thuật sử dụng

Phương pháp hồi quy tuyến tính:

- **Mô hình:** Mô hình quan hệ tuyến tính giữa một biến phụ thuộc và một hoặc nhiều biến độc lập.
- **Các điểm chính:** Đơn giản, dễ hiểu, nhanh chóng để đào tạo, nhạy cảm với các giá trị ngoại lai và mối quan hệ phi tuyến.
- **Các trường hợp sử dụng điển hình:** Dự đoán các giá trị liên tục (ví dụ: doanh số, giá cả), phân tích xu hướng, hiểu mối quan hệ giữa các biến.

Cây quyết định:

- **Mô hình:** Cấu trúc giống cây chia dữ liệu dựa trên các tính năng để đưa ra dự đoán.
- **Các điểm chính:** Xử lý các mối quan hệ phi tuyến, dễ hiểu, có thể quá khớp.
- **Các trường hợp sử dụng điển hình:** Phân loại, hồi quy, chọn tính năng, hiểu các quy tắc quyết định.

Rừng ngẫu nhiên:

- **Mô hình:** Một tập hợp các cây quyết định, lấy trung bình các dự đoán của chúng.
- **Các điểm chính:** Giảm quá khớp, thường có độ chính xác cao, ít dễ hiểu hơn các cây riêng lẻ.

- Các trường hợp sử dụng điển hình: Phân loại, hồi quy, xử lý các mối quan hệ phức tạp, phân tích tầm quan trọng của tính năng.

LSTM (Long Short-Term Memory):

- Mô hình: Một loại mạng nơ-ron hồi quy (RNN) có khả năng nắm bắt các phụ thuộc lâu dài trong dữ liệu tuần tự.
- Các điểm chính: Mạnh mẽ cho phân tích chuỗi thời gian, xử lý ngôn ngữ tự nhiên, các nhiệm vụ tuần tự phức tạp, có thể tồn kém về mặt tính toán.
- Các trường hợp sử dụng điển hình: Dự đoán chuỗi thời gian (ví dụ: giá cổ phiếu, thời tiết), tạo văn bản, dịch ngôn ngữ, phát hiện bất thường trong chuỗi.

ARIMA (Autoregressive Integrated Moving Average):

- Mô hình: Mô hình thống kê dành riêng cho phân tích chuỗi thời gian, mô hình hóa xu hướng, theo mùa và tự tương quan.
- Các điểm chính: Hiệu quả đối với chuỗi thời gian ổn định, yêu cầu dữ liệu ổn định, xử lý các mẫu phức tạp hạn chế.
- Các trường hợp sử dụng điển hình: Dự đoán chuỗi thời gian với các mẫu rõ ràng (ví dụ: doanh số, nhu cầu, lưu lượng truy cập web), hiểu các thành phần của chuỗi thời gian.

XGBoost (Extreme Gradient Boosting):

- Mô hình: Một triển khai hiệu quả cao của gradient boosting, kết hợp nhiều cây quyết định.
- Các điểm chính: Thường đạt được độ chính xác hàng đầu, xử lý các giá trị thiếu, có thể ít dễ hiểu hơn.
- Các trường hợp sử dụng điển hình: Phân loại, hồi quy, các vấn đề xếp hạng, chọn tính năng, xử lý các tập dữ liệu lớn.

CHƯƠNG III:MÔ HÌNH LÝ THUYẾT

3.1. Lý thuyết về Time Series Decomposition

Phương pháp phân rã chuỗi thời gian, hay Time Series Decomposition, là một kỹ thuật giúp tách một dãy số thời gian thành những thành phần cơ bản để hiểu rõ hơn về xu hướng, chu kỳ và thành phần ngẫu nhiên. Thông thường, chuỗi thời gian được phân rã thành ba thành phần chính: xu hướng dữ liệu với chu kỳ và thành phần được cho là ngẫu nhiên.

Xu hướng thể hiện hướng chung của chuỗi thời gian, có thể là tăng, giảm hoặc ổn định theo thời gian. Chu kỳ đại diện cho các biến động lặp lại trong chuỗi thời gian, ví dụ như sự tăng mạnh trong doanh số bán hàng vào mùa giáng sinh. Thành phần ngẫu nhiên là sự biến động không thể giải thích bằng xu hướng hoặc chu kỳ.

Có hai phương pháp chính để phân rã chuỗi thời gian: phân rã Additive và phân rã Multiplicative.

- Phân rã Additive theo công thức :

$$Y_t = Trend_t + Seasonal_t + Residual_t$$

- Phân rã Multiplicative thì nó nhân với nhau

$$Y_t = Trend_t * Seasonal_t * Residual_t$$

Chú thích:

$Trend_t$ cho là xu hướng mua bán tại t

$Seasonal_t$ thành phần của chu kỳ tại t

$Residual_t$ là thành phần là ngẫu nhiên tại t

Phương pháp này giúp nắm bắt cấu trúc của dữ liệu, hỗ trợ trong dự đoán và phân tích theo thời gian, làm cho việc hiểu rõ hơn về xu hướng và biến động của chuỗi thời gian trở nên thuận tiện.

3.2. Simple Moving Average

Khái niệm:

Simple Moving Average (SMA) là một phương pháp thống kê được sử dụng rộng rãi trong dự báo và phân tích chuỗi số liệu, bao gồm cả dự báo nhiệt độ. SMA tính trung bình của một số lượng quy định các giá trị liên tiếp trong chuỗi dữ liệu và di chuyển qua cửa sổ dữ liệu theo thời gian.

Trong việc dự báo nhiệt độ, việc sử dụng giá trị Simple Moving Average (SMA) có thể giúp nhận diện xu hướng dài hạn của thời tiết. SMA là một công cụ mạnh mẽ để làm mịn dữ liệu và làm nổi bật xu hướng chung của nhiệt độ theo thời gian. Nếu nhiệt độ vượt lên trên đường SMA, điều này có thể được coi là dấu hiệu của một xu hướng ấm lên trong tương lai. Ngược lại, nếu nhiệt độ xuống dưới đường SMA, điều này có thể là dấu hiệu của thời tiết lạnh hơn sắp đến.

Thêm vào đó, việc theo dõi các tín hiệu "Golden Cross" và "Death Cross" cũng là một phương pháp quan trọng để nhận biết sự thay đổi trong xu hướng nhiệt độ. Khi có sự kết hợp giữa giá trị nhiệt độ và đường SMA tạo ra "Golden Cross" (khi nhiệt độ vượt lên trên đường SMA), đây có thể là dấu hiệu mạnh mẽ cho việc dự báo thời tiết ấm hơn. Ngược lại, khi xuất hiện "Death Cross" (khi nhiệt độ xuống dưới đường SMA), có thể là dấu hiệu cho thời tiết lạnh hơn trong thời gian sắp tới.

Tuy nhiên, cần lưu ý rằng SMA cũng có nhược điểm, đặc biệt là khả năng tạo ra tín hiệu trễ do sự trễ trong việc phản ánh biến động nhanh chóng của nhiệt độ. Vì vậy, việc kết hợp SMA với các phương pháp và công cụ khác có thể giúp tăng cường chính xác và đồng thời cung cấp cái nhìn toàn diện hơn về dự báo nhiệt độ.

Công thức SMA:

$$SMA = \frac{P_1 + P_2 + P_3 + \dots + P_n}{n}$$

Trong đó:

- *SMA*: là giá trị trung bình đơn giản (Simple Moving Average)
- $P_1 + P_2 + P_3 + \dots + P_n$: các giá trị cần tính trung bình (trong ngữ cảnh dự báo nhiệt độ, có thể là các giá trị nhiệt độ tại các điểm thời gian khác nhau)
- n là số lượng giá trị được sử dụng để tính trung bình

3.3. Tổng quan về LSTM(Long Short-Term Memory) :

3.3.1 Khái niệm

LSTM (viết tắt của Long Short-Term Memory) là một kiến trúc mạng nơ-ron thần kinh sử dụng để xử lý và phân tích dữ liệu tuần tự, chẳng hạn như văn bản tự nhiên, âm thanh hoặc dữ liệu chuỗi thời gian.

Để sử dụng LSTM cho bài toán dự đoán, chúng ta cần huấn luyện mạng nơ-ron trên một tập dữ liệu lịch sử, sau đó sử dụng mô hình đã huấn luyện để dự đoán giá trị tiếp theo của chuỗi thời gian hoặc từ tiếp theo của văn bản.

Một cách tiếp cận phổ biến là sử dụng LSTM kết hợp với các mô hình khác, chẳng hạn như mô hình mạng hồi quy tuyến tính (linear regression), để dự đoán giá trị của chuỗi thời gian hoặc từ tiếp theo của văn bản. Các kỹ thuật như đánh giá độ chính xác và điều chỉnh siêu tham số cũng được sử dụng để cải thiện chất lượng dự đoán của mô hình.

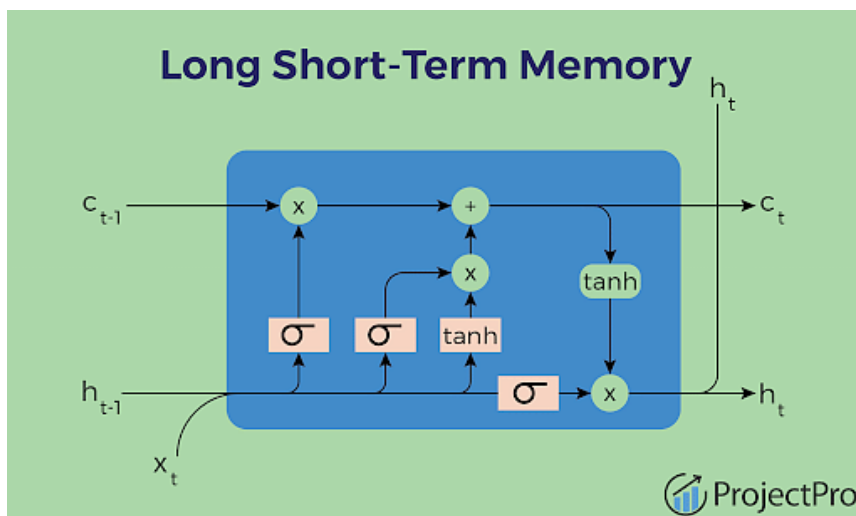
LSTM sử dụng các cổng thông tin để kiểm soát thông tin được lưu trữ và truyền qua các cell state. Kiến trúc này cho phép mạng nơ-ron học được cách lưu trữ và truyền thông tin dài hạn và ngắn hạn, giúp mô hình hiểu được mối quan hệ giữa các thành phần của dữ liệu đầu vào.

LSTM đã được sử dụng rộng rãi trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, dịch máy và dự đoán chuỗi thời gian.

Mô hình LSTM (Long Short-Term Memory) là một kiến trúc mạng nơ-ron sử dụng để xử lý và phân tích dữ liệu tuần tự. Nó được thiết kế để giải quyết vấn đề của việc xử lý thông tin trong dữ liệu dài hạn, trong đó các thông tin có thể bị mất đi theo thời gian. Kiến trúc của LSTM bao gồm ba cổng thông tin (gates):

- Cổng quên (Forget gate): quyết định thông tin nào cần bị xóa hoặc quên đi từ cell state (trạng thái của cell) của mô hình.
- Cổng đầu vào (Input gate): quyết định thông tin nào cần được cập nhật vào cell state.
- Cổng đầu ra (Output gate): quyết định thông tin nào cần được đưa ra từ cell state để đưa ra dự đoán hoặc sử dụng cho một mục đích khác.

Các cổng thông tin này giúp cho mô hình LSTM có khả năng học và lưu trữ thông tin dài hạn cũng như lưu trữ thông tin ngắn hạn. Một số ứng dụng của LSTM bao gồm dịch máy, nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên.



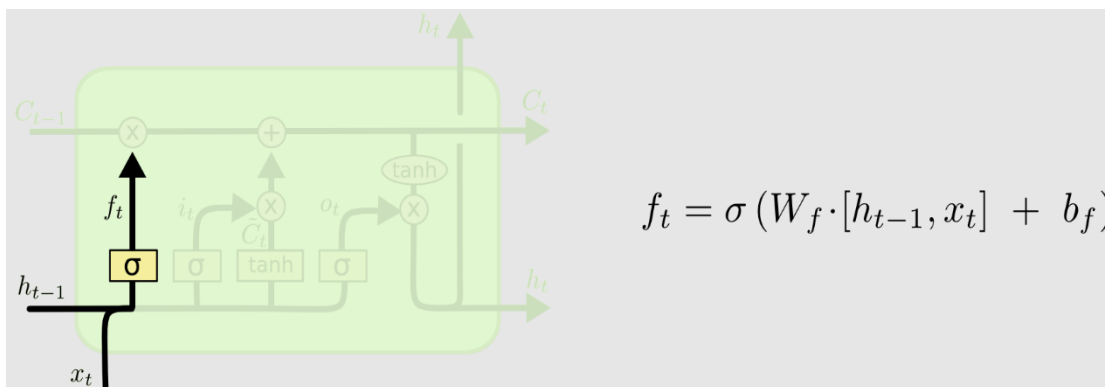
Hình 3. Mô hình LSTM

3.3.2. Cổng quên (Forget gate)

Cổng quên (Forget gate) là một trong ba cổng thông tin (gates) trong kiến trúc mô hình LSTM (Long Short-Term Memory). Cổng này quyết định thông tin nào cần bị xóa hoặc quên đi từ cell state (trạng thái của cell) của mô hình.

Trong quá trình huấn luyện, mô hình LSTM học cách xác định thông tin quan trọng để giữ lại và thông tin không quan trọng để xóa bỏ. Các thông tin không cần thiết được xóa bỏ thông qua cổng quên bằng cách nhân một vector cổng quên với trạng thái trước đó của cell (state), sau đó truyền kết quả tới cell state mới.

Cổng quên được tính toán bằng cách sử dụng hàm sigmoid để đưa ra các giá trị trong khoảng từ 0 đến 1, trong đó giá trị 1 có nghĩa là "giữ lại hoàn toàn" và giá trị 0 có nghĩa là "bỏ qua hoàn toàn". Các giá trị trung bình có nghĩa là "giữ lại một phần và xóa bỏ một phần". Ví dụ, giả sử rằng mô hình cần xác định thông tin quan trọng để giữ lại về một từ cụ thể trong câu. Nếu từ đó không còn quan trọng trong bối cảnh mới, cổng quên sẽ được kích hoạt và thông tin về từ đó sẽ được xóa bỏ khỏi trạng thái cell state của mô hình.



Hình 4.Cổng quên (Forget gate)

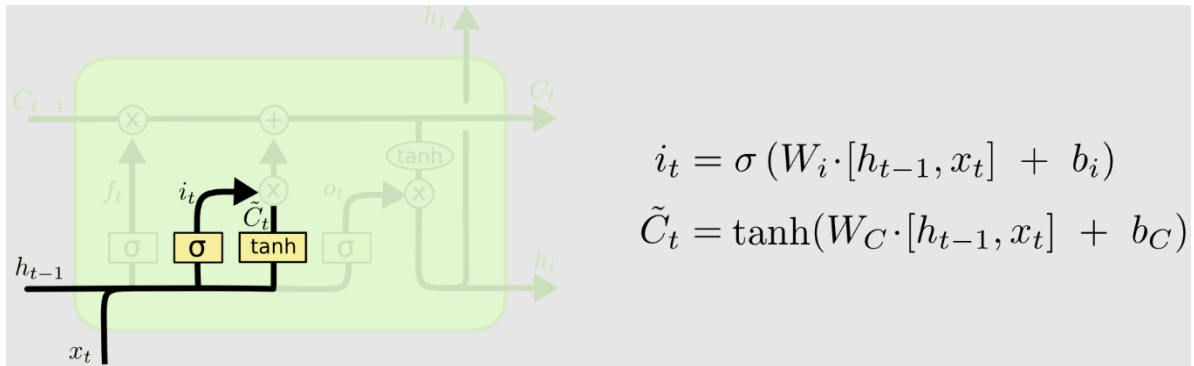
3.3.3. Cổng đầu vào (Input gate)

Cổng đầu vào (Input gate) là một trong ba cổng thông tin (gates) trong kiến trúc mô hình LSTM (Long Short-Term Memory). Cổng này quyết định thông tin nào sẽ được thêm vào cell state (trạng thái của cell) của mô hình.

Trong quá trình huấn luyện, mô hình LSTM học cách xác định thông tin mới nào cần được thêm vào để cập nhật trạng thái cell state. Cổng đầu vào thực hiện điều này bằng cách tính toán một vector đầu vào mới, sau đó đưa vào trạng thái cell state của mô hình.

Cổng đầu vào được tính toán bằng cách sử dụng hàm sigmoid để đưa ra các giá trị trong khoảng từ 0 đến 1, trong đó giá trị 1 có nghĩa là "giữ lại hoàn toàn" và giá trị 0 có nghĩa là "không giữ lại gì cả". Các giá trị trung bình có nghĩa là "giữ lại một phần và bỏ qua một phần". Sau đó, các thông tin mới sẽ được tính toán bằng cách sử dụng hàm tanh để đưa ra các giá trị trong khoảng từ -1 đến 1, sau đó nhân với vector đầu vào để tạo ra vector mới. Vector này sẽ được cộng vào trạng thái cell state hiện tại để tạo ra trạng thái cell state mới của mô hình.

Ví dụ, trong một bài toán dịch máy, cổng đầu vào sẽ quyết định thông tin mới nào từ câu nguồn sẽ được thêm vào để tạo ra câu đích.



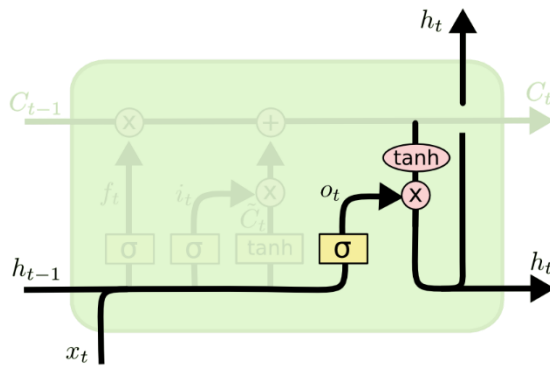
Hình 5. Cổng đầu vào (Input gate)

3.3.4. Cổng đầu ra (Output gate)

Cổng đầu ra (Output gate) là một trong ba cổng thông tin (gates) trong kiến trúc mô hình LSTM (Long Short-Term Memory). Cổng này quyết định thông tin nào cần được đưa ra từ cell state để đưa ra dự đoán hoặc sử dụng cho một mục đích khác.

Trong quá trình huấn luyện, mô hình LSTM học cách xác định thông tin quan trọng để đưa ra dự đoán hoặc sử dụng cho một mục đích khác. Các thông tin được chọn sẽ được truyền qua cổng đầu ra, sau đó được đưa vào hàm kích hoạt để tạo ra kết quả dự đoán hoặc được sử dụng cho mục đích khác.

Sau khi tính toán các đầu vào hiện tại và trạng thái bộ nhớ, cổng đầu ra quyết định giá trị nào sẽ được truyền sang bước thời gian tiếp theo. Trong cổng đầu ra, giá trị được phân tích và đánh giá tầm quan trọng từ -1 đến 1. Điều này điều chỉnh dữ liệu trước khi chúng được chuyển đến tính toán cho bước thời gian tiếp theo. Cuối cùng, nhiệm vụ của cổng quên là loại bỏ thông tin mà mô hình coi là không cần thiết để đưa ra quyết định về bản chất của các giá trị đầu vào. Cổng quên sử dụng hàm sigmoid để xuất ra các số trong khoảng từ 0 (quên giá trị này) đến 1 (giữ giá trị này).



$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

Hình 6.Cổng đầu ra (Output gate)

3.3.5. Sơ đồ của mô hình LSTM

Bước 1: Bắt đầu quá trình với việc thu thập dữ liệu từ các nguồn đáng tin cậy, chọn các thông tin liên quan đến mục tiêu cụ thể, ví dụ như dữ liệu nhiệt độ của một đất nước.

Bước 2: Tiến hành tiền xử lý dữ liệu cho mô hình Long Short-Term Memory (LSTM) bằng cách làm sạch dữ liệu và xử lý các giá trị bị thiếu. Đồng thời, thực hiện phân tích để hiểu rõ các đặc điểm và mối quan hệ giữa chúng.

Bước 3: Thiết kế và xây dựng kiến trúc cho mô hình LSTM, quyết định số lớp, số nơ-ron và các tham số khác của mô hình. Mô hình LSTM sẽ tự học từ dữ liệu để dự đoán hoặc phân tích các mẫu và xu hướng.

Bước 4: Sử dụng dữ liệu đã được tiền xử lý để đào tạo mô hình LSTM. Cung cấp dữ liệu và điều chỉnh các trọng số và tham số của mô hình dựa trên quá trình đào tạo.

Bước 5: Sau khi mô hình đã được học, thực hiện dự đoán hoặc phân tích trên tập dữ liệu. Kết quả từ mô hình LSTM giúp đưa ra quyết định hoặc phân tích các xu hướng và dự đoán trong dữ liệu, dựa trên khả năng học của mô hình từ các mẫu thời gian trước đó.

3.4. Hồi quy tuyến tính (Linear Regression)

3.4.1 Khái niệm

Hồi quy tuyến tính (Linear Regression) là một phương pháp thống kê để mô hình hóa mối quan hệ tuyến tính giữa một biến phụ thuộc (đối tượng cần dự đoán) và một hoặc nhiều biến độc lập (đối tượng mô tả).

Hồi quy tuyến tính, trong dự báo nhiệt độ là một phương pháp quan trọng giúp mô hình hóa mối quan hệ giữa nhiệt độ và các yếu tố độc lập như độ ẩm, áp suất không khí, và tốc độ gió. Trong quá trình này, ta thu thập dữ liệu chính xác và đa dạng từ các nguồn đáng tin cậy, sau đó thực hiện tiền xử lý để xử lý giá trị thiếu và loại bỏ ngoại lệ.

Sau khi có dữ liệu chuẩn bị, ta chọn các biến quan trọng và xây dựng mô hình hồi quy tuyến tính, trong đó nhiệt độ được xem xét là biến phụ thuộc, và các yếu tố khác là các biến độc lập. Mô hình này có thể được biểu diễn dưới dạng công thức toán học, với các hệ số tương ứng với ảnh hưởng của mỗi yếu tố.

Sau khi xây dựng mô hình, ta sử dụng các độ đo như R-squared, MSE, và RMSE để đánh giá hiệu suất của nó. Quá trình này không chỉ giúp ta định lượng mối quan hệ giữa nhiệt độ và các yếu tố, mà còn cho phép dự đoán nhiệt độ trong tương lai và đưa ra những dự báo hữu ích. Đồng thời, ta có thể kiểm định tính tổng quát của mô hình bằng cách sử dụng kiểm định trên dữ liệu kiểm định hoặc cross-validation.

Cuối cùng, hiểu rõ ý nghĩa của các hệ số hồi quy giúp ta diễn giải kết quả một cách chính xác, từ đó hỗ trợ quyết định và đưa ra dự báo thời tiết một cách đáng tin cậy. Hồi quy tuyến tính, mặc dù có giới hạn trong việc mô hình hóa mối quan hệ tuyến tính, nhưng vẫn là một công cụ quan trọng và linh hoạt trong dự báo nhiệt độ.

Công thức

$$y = B_0 + B_1X_1 + \dots + B_nX_n + \epsilon$$

Trong đó:

- y là biến phụ thuộc (đối tượng cần dự đoán).
- X_1, X_2, \dots, X_n : là các biến độc lập
- B_0 hệ số gốc, B_1, B_2, \dots, B_n là hệ số góc tương ứng

- ϵ là sai số ngẫu nhiên, biểu thị sự biến động chưa được mô hình hóa

3.4.2 Sơ đồ của Linear Regression

Bước 1: Thu thập dữ liệu liên quan đến biến phụ thuộc và các biến độc lập mà bạn nghiên cứu.

Bước 2: Tiến hành tiền xử lý dữ liệu cho mô hình Linear Regression bằng cách làm sạch dữ liệu và xử lý các giá trị bị thiếu. Đồng thời, thực hiện phân tích để hiểu rõ các đặc điểm và mối quan hệ giữa chúng, kiểm tra dữ liệu để xác định và xử lý các giá trị thiếu, ngoại lệ, hoặc dữ liệu không hợp lý

Bước 3: Chọn các biến độc lập có thể ảnh hưởng đến biến phụ thuộc. Sự chọn lựa này có thể dựa trên kiến thức chuyên gia hoặc phân tích thống kê và phân chia dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá mô hình

Bước 4: Sử dụng dữ liệu huấn luyện để xây dựng mô hình Linear Regression. Điều này bao gồm ước lượng các hệ số (intercept và các hệ số của biến độc lập) thông qua phương pháp bình phương tối thiểu.

Bước 5: RMSE để đánh giá hiệu suất của mô hình trên tập kiểm tra..

3.5. Cây quyết định (Decision Tree)

3.5.1. Khái niệm

Cây quyết định (Decision Tree) là một mô hình máy học phổ biến được sử dụng để dự đoán và phân loại dữ liệu. Trong ngữ cảnh dự đoán nhiệt độ, cây quyết định có thể được áp dụng để tìm ra các quy luật quyết định dựa trên các yếu tố như độ ẩm, áp suất không khí, và tốc độ gió để dự đoán nhiệt độ.

Cấu Trúc Cây Quyết Định:

- Nút gốc của cây đại diện cho toàn bộ tập dữ liệu và chứa quyết định về việc chia tập dữ liệu thành các nhánh dựa trên một thuộc tính nào đó.
- Các nút quyết định là các nút nội bộ của cây, mỗi nút đại diện cho một quyết định về cách phân loại hoặc dự đoán dữ liệu.
- Nút lá là nút cuối cùng của cây và chứa kết quả cuối cùng, tức là dự đoán nhiệt độ hoặc một giá trị cụ thể.

3.5.2. Sơ đồ của Decision Tree

Bước 1: Thu thập dữ liệu liên quan về nhiệt độ bạn nghiên cứu.

Bước 2: Kiểm tra dữ liệu để xác định và xử lý các giá trị thiếu, ngoại lệ, hoặc dữ liệu không hợp lý

Bước 3: Chọn thuộc tính dựa trên độ quan trọng của nó trong việc giải thích biến độc lập. Các thuộc tính như độ ẩm, áp suất không khí có thể là những lựa chọn phổ biến. Chia tập dữ liệu thành các nhóm dựa trên giá trị của thuộc tính chọn lựa.

Bước 4: Đưa ra quyết định cho mỗi nhánh, có thể là giá trị thường trực (trong trường hợp hồi quy) hoặc lớp/phân loại chủ yếu (trong trường hợp phân loại). Lặp lại quá trình xây dựng cây trên mỗi nhóm con cho đến khi đạt được điều kiện dừng, chẳng hạn như đạt đến một chiều sâu tối đa, số lượng quan sát tối thiểu cho mỗi lá, hoặc không còn có thể chia nhánh được.

Bước 5: RMSE để đánh giá hiệu suất của mô hình trên tập kiểm tra..

3.6. Random Forest

3.6.1. Khái niệm

Random Forest là một phương pháp ensemble learning, nơi mà nhiều mô hình học máy (thường là cây quyết định) được kết hợp để tạo ra một mô hình mạnh mẽ và ổn định hơn so với việc sử dụng một mô hình duy nhất.

RandomForestRegressor là một mô hình trong thư viện scikit-learn được sử dụng cho bài toán hồi quy (dự đoán giá trị số). RandomForestRegressor không có một công thức cụ thể như các mô hình toán học truyền thống, nhưng nó dựa trên ý tưởng của ensemble learning và sử dụng nhiều cây quyết định để tạo ra dự đoán tổng hợp.

Dưới đây là một số tham số quan trọng cho RandomForestRegressor:

- `n_estimators`: Số lượng cây quyết định trong "rừng". Điều này là một tham số quan trọng, quyết định số lượng cây mà mô hình sẽ sử dụng.
- `max_depth`: Chiều sâu tối đa của mỗi cây quyết định. Điều này kiểm soát độ sâu của cây, ảnh hưởng đến khả năng mô hình học các mối quan hệ phức tạp trong dữ liệu.

- `min_samples_split`:Số lượng mẫu tối thiểu được yêu cầu để chia một node trong quá trình xây dựng cây.
- `min_samples_leaf`:Số lượng mẫu tối thiểu được yêu cầu để tạo một lá trong cây.
- `max_features`:Số lượng thuộc tính được chọn để cân nhắc khi chia mỗi node. Giá trị này giúp tạo ra sự đa dạng giữa các cây.
- `random_state`:Điều này là một giá trị nguyên để đảm bảo rằng mô hình sẽ đưa ra kết quả nhất quán mỗi khi được đào tạo.

RandomForestRegressor sử dụng thuật toán bagging (Bootstrap Aggregating) để xây dựng nhiều cây quyết định độc lập và sau đó kết hợp chúng để giảm overfitting và tăng tính ổn định của mô hình. Tính đa dạng của các cây được đảm bảo bằng cách chọn một số lượng ngẫu nhiên các thuộc tính cho mỗi node trong quá trình xây dựng cây

3.7. XGBoost

3.7.1.Khái niệm

XGBoost, hay Extreme Gradient Boosting, là một mô hình học máy độc đáo và hiệu quả trong việc dự đoán nhiệt độ. XGBoost kết hợp những ưu điểm của các mô hình Gradient Boosting và các kỹ thuật tinh chỉnh mô hình, tạo ra một công cụ mạnh mẽ cho các ứng dụng dự đoán và dự báo thời tiết.

Thuật toán XGBoost hoạt động dựa trên cơ sở của Gradient Boosting, xây dựng một chuỗi các cây quyết định để liên tục sửa lỗi từng cây trước đó. Điểm độc đáo của XGBoost là sự kết hợp linh hoạt của hàm mất mát (loss function) và regularization term trong hàm mục tiêu, giúp kiểm soát độ phức tạp của mô hình và ngăn chặn hiện tượng quá mức điều chỉnh (overfitting).

$$Object(\theta) = L(\theta) + \Omega(\theta)$$

Trong đó

- $L(\theta)$ là hàm mất mát đo lường sai số dự đoán và giá trị thực tế, còn
- $\Omega(\theta)$ là regularization term. Hàm mục tiêu của mô hình để kiểm soát độ phức tạp của nó và ngăn chặn hiện tượng quá mức điều chỉnh.

3.8. ARIMA

3.8.1. Khái niệm

Dựa trên giả thuyết chuỗi dừng và phương sai sai số không đổi. Mô hình sử dụng đầu vào chính là những tín hiệu quá khứ của chuỗi được dự báo để dự báo nó. Các tín hiệu đó bao gồm: chuỗi tự hồi quy AR (auto regression) và chuỗi trung bình trượt MA (moving average). Hầu hết các chuỗi thời gian sẽ có xu hướng tăng hoặc giảm theo thời gian, do đó yếu tố chuỗi dừng thường không đạt được. Trong trường hợp chuỗi không dừng thì ta sẽ cần biến đổi sang chuỗi dừng bằng sai phân. Khi đó tham số đặc trưng của mô hình sẽ có thêm thành phần bậc của sai phân d và mô hình được đặc tả bởi 3 tham số ARIMA(p, d, q).

ARIMA (Autoregressive Integrated Moving Average) là một mô hình dự báo chuỗi thời gian phổ biến được sử dụng để mô hình hóa dữ liệu chuỗi thời gian và dự báo giá trị trong tương lai. Mô hình ARIMA kết hợp các thành phần autoregression (AR), integration (I) và moving average (MA) để mô hình hóa xu hướng, yếu tố chu kỳ và dao động ngẫu nhiên trong dữ liệu chuỗi thời gian.

Thành phần autoregression (AR) mô hình hóa mối quan hệ tuyến tính giữa giá trị hiện tại và các giá trị trước đó trong chuỗi thời gian. Thành phần integration (I) thực hiện việc chuyển đổi dữ liệu chuỗi thời gian để loại bỏ xu hướng phi tuyến tính, còn thành phần moving average (MA) mô hình hóa sự dao động ngẫu nhiên của dữ liệu.

Mô hình ARIMA được xác định bằng cách chọn các tham số p , d và q . Tham số p đại diện cho số lượng giá trị autoregressive được sử dụng trong mô hình, tham số q đại diện cho số lượng giá trị moving average và tham số d đại diện cho số lần chuyển đổi dữ liệu chuỗi thời gian.

Arima có dạng:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + u_t + \theta_1 u_{t-1} + \dots + \theta_q u_{t-q} + e_t$$

Trong đó

- Y_t là chuỗi dừng bậc d của chuỗi ban đầu (chuỗi khảo sát)
- ϕ là tham số tự hồi quy.
- θ là tham số trung bình di động.

- u_t là nhiễu trắng.
- e_t là sai số dự báo.
- p là bậc tự hồi qui.
- q là bậc trung bình trượt.

3.8.2. Sơ đồ mô hình Arima

Bước 1: Thu thập chuỗi thời gian cần dự đoán..

Bước 2: Kiểm tra dữ liệu để xác định và xử lý các giá trị thiếu, ngoại lệ, hoặc dữ liệu không hợp lý, Kiểm tra chuỗi thời gian để đảm bảo tính dừng (stationarity).

Nếu không có tính dừng, thực hiện bước chuyển đổi (differencing)

Bước 3: Sử dụng đồ thị ACF (Autocorrelation Function) và PACF (Partial Autocorrelation Function) để xác định bậc của MA và AR. ACF là đo độ tương quan giữa một quan sát và các quan sát trước đó. PACF chỉ đo tương quan giữa một quan sát và các quan sát trước đó loại trừ tác động của các quan sát trung gian..

Bước 4: Dựa vào kết quả từ bước trước để xây dựng mô hình ARIMA. Chia dữ liệu thành tập huấn luyện và tập kiểm tra. Sử dụng tập huấn luyện để ước lượng tham số và xây dựng mô hình ARIMA.

Bước 5: RMSE để đánh giá hiệu suất của mô hình trên tập kiểm tra..

3.9. Root Mean Squared Error (RMSE)

Trong quá trình thực hiện dự báo và Machine Learning, Root Mean Squared Error (RMSE) là một thước đo phổ biến được áp dụng để đánh giá sự chính xác của dự đoán. RMSE được ưa chuộng bởi vì nó trả về kết quả có đơn vị đo giống với dữ liệu gốc, giúp đơn giản hóa quá trình đánh giá mức độ sai số dự đoán và hiểu rõ ý nghĩa thực tế của nó

Công thức tính RMSE được mô tả như sau:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- n là số lượng quan sát trong tập dữ liệu.
- y_i là giá trị thực tế của biến phụ thuộc cho quan sát thứ i
- \hat{y}_i là giá trị dự đoán tương ứng cho quan sát thứ i

CHƯƠNG IV:MÔ HÌNH THỰC NGHIỆM

4.1. Mô tả bộ dữ liệu

Dữ liệu này có những thông tin về nhiệt độ trung bình của hệ thống khí hậu Trái đất, thời gian kéo dài từ 1995-2020. Thu thập từ trang Kaggle nhờ Đại học Dayton, tập dữ liệu có sẵn dưới dạng tệp txt riêng biệt về nhiệt độ trung bình trong suốt khoảng thời gian này. Các cột trong bộ dữ liệu bao gồm 8 trường dữ liệu và 2906327 dòng:

- **Region:** Đây là tên khu vực địa lý lớn hoặc miền địa lý mà dữ liệu đang tham chiếu. Ví dụ: Châu Á, Châu Âu, Bắc Mỹ.
- **Country:** Đây là tên quốc gia mà dữ liệu được thu thập hoặc đo lường. Mỗi quốc gia có thể thuộc vào một khu vực cụ thể.
- **State:** Nếu dữ liệu liên quan đến các quốc gia có cấu trúc liên bang hoặc tương tự, cột này có thể chứa tên của các bang, tỉnh hoặc đơn vị hành chính cấp dưới.
- **City:** Đây là tên của thành phố hoặc địa điểm cụ thể mà dữ liệu được thu thập. Đối với các quốc gia có nhiều thành phố, đây có thể là tên của thành phố cụ thể.
- **Month:** Cột này cho biết tháng tương ứng với dữ liệu nhiệt độ. Nó có thể là giá trị số hoặc tên của tháng.
- **Day:** Cột này chứa giá trị số đại diện cho ngày trong tháng tương ứng với dữ liệu.
- **Year:** Cột này chứa giá trị số đại diện cho năm của dữ liệu nhiệt độ.
- **AvgTemperature:** Đây là cột chứa giá trị nhiệt độ trung bình trong tháng và ngày cụ thể cho địa điểm, quốc gia hoặc khu vực được mô tả

4.1.1. Thông tin cung cấp

Dữ liệu này có thể được sử dụng để thực hiện các phân tích thống kê, xây dựng mô hình dự đoán nhiệt độ, hoặc theo dõi biến động nhiệt độ theo thời gian và địa điểm. Thông qua các cột này, người ta có thể hiểu về xu hướng thời tiết, mùa vụ, và biến động nhiệt độ tại các địa điểm cụ thể hoặc trong quốc gia, khu vực lớn.

4.1.2. Tại sao nhiệt độ trung bình của quan trọng

Cột "AvgTemperature" (Nhiệt độ trung bình) quan trọng trong dự báo nhiệt độ vì nó là yếu tố chính đo lường mức độ nhiệt độ trung bình tại một địa điểm và thời điểm cụ thể. Dưới đây là một số lý do:

- **Thành Phần Quan Trọng Của Dữ Liệu Nhiệt Độ:** Nhiệt độ trung bình là thông tin quan trọng nhất khi nói về dự báo nhiệt độ. Đó là giá trị chính mà người ta quan tâm để hiểu về điều kiện thời tiết hàng ngày hoặc hàng tháng.
- **Điều Tiết Thời Tiết Hàng Ngày và Hàng Tháng:** Nhiệt độ trung bình là một đại diện cho mức nhiệt độ trung bình trong một khoảng thời gian cụ thể. Điều này giúp người ta đánh giá được xu hướng và biến động của thời tiết theo thời gian.
- **Dự Báo Hạn Chế Các Biến Động Ngắn Hạn:** Trong dự báo nhiệt độ, sự biến động ngắn hạn có thể làm cho dự báo trở nên khó khăn. Sử dụng nhiệt độ trung bình giúp giảm ảnh hưởng của các yếu tố ngắn hạn và tạo ra dự báo ổn định hơn.
- **Mối Quan Hệ với Các Yếu Tố Khác:** Nhiệt độ trung bình thường có mối quan hệ với các yếu tố khác như áp suất không khí, độ ẩm, và gió. Việc hiểu mối quan hệ này có thể giúp cải thiện dự báo và đánh giá sự ảnh hưởng của các yếu tố khác lên nhiệt độ.
- **Thành Phần Quan Trọng Cho Mô Hình Dự Báo:** Trong các mô hình dự báo nhiệt độ, nhiệt độ trung bình thường là một trong những biến động quan trọng được sử dụng để đào tạo mô hình. Nó đóng vai trò quan trọng trong việc xác định xu hướng và biến động.
- **Đánh Giá Mức Độ Thay Đổi Khí Hậu:** Nhiệt độ trung bình cũng cung cấp thông tin quan trọng cho việc đánh giá thay đổi khí hậu. Việc theo dõi sự biến động của nhiệt độ trung bình giúp xác định xu hướng và thay đổi lâu dài trong khí hậu.

Cột "AvgTemperature" không chỉ là một dữ liệu thống kê mà còn là một yếu tố chính trong dự báo nhiệt độ, đóng vai trò quan trọng trong việc hiểu và mô hình hóa thời tiết.

4.2. Trực quan hóa dữ liệu

Việc trực quan hóa theo thời gian giúp chúng ta nhanh chóng nhận thức xu hướng thời tiết, biến động và thậm chí là sự thay đổi theo mùa vụ. Các biểu đồ có thể phân tích và hiển thị các chu kỳ thời tiết, từ biến động ngày đến các mô hình thay đổi hàng tuần hoặc theo mùa. Điều này làm cho quá trình dự báo trở nên linh hoạt và có thể được điều chỉnh dựa trên những mô hình này.

Ngoài ra, việc trực quan hóa cũng giúp dự báo tình hình thời tiết tương lai. Mô hình dự báo có thể được biểu diễn trực quan để so sánh với dữ liệu quá khứ và đánh giá sự biến động dự kiến. Những biểu đồ này giúp ta dễ dàng nhận biết và ứng phó với sự thay đổi thời tiết đối với các kịch bản dự kiến.

Việc trực quan hóa nhiệt độ kết hợp với các yếu tố khác như độ ẩm, áp suất không khí và gió giúp phân tích tương quan giữa chúng, mở rộng sự hiểu biết về các yếu tố gây ảnh hưởng lẫn nhau. Điều này cung cấp cái nhìn toàn diện hơn về môi trường thời tiết.

Cuối cùng, việc trực quan hóa không chỉ là một công cụ mạnh mẽ cho các nhà khoa học và dự báo viên, mà còn là một phương tiện giao tiếp hiệu quả với cộng đồng. Các biểu đồ và đồ thị giúp chia sẻ thông tin về thời tiết một cách dễ hiểu và tương tác, hỗ trợ người dùng tự chuẩn bị và ứng phó với biến động thời tiết.

4.2.1. Thông tin về bộ dữ liệu

```
[198] # in ra số dòng số cột
      print("Số dòng, số cột:", df.shape)

Số dòng, số cột: (2906327, 8)

df.columns

Index(['Region', 'Country', 'State', 'City', 'Month', 'Day', 'Year',
       'AvgTemperature'],
      dtype='object')

[200] # in ra thông tin tập dữ liệu
      df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2906327 entries, 0 to 2906326
Data columns (total 8 columns):
#   Column          Dtype
---  -----  ---
0   Region          object
1   Country         object
2   State          object
3   City           object
4   Month          int64
5   Day            int64
6   Year           int64
7   AvgTemperature float64
dtypes: float64(1), int64(3), object(4)
memory usage: 177.4+ MB
```

Hình 7. Thông tin về bộ dữ liệu

Tập dữ liệu mà bạn đã cung cấp bao gồm 2,906,327 dòng và 8 cột, đại diện cho một tập hợp lớn dữ liệu về thời tiết. Các cột chính trong tập dữ liệu này bao gồm 'Region', 'Country', 'State', 'City', 'Month', 'Day', 'Year', và 'AvgTemperature'. Các thông số này cung cấp cái nhìn tổng quan về đặc điểm và quy mô của dữ liệu.

Các cột 'Region', 'Country', 'State', và 'City' thường chứa thông tin địa lý, đặc biệt là về khu vực, quốc gia, bang/tỉnh và thành phố. Cột 'Month' biểu diễn tháng trong năm dưới dạng số nguyên, trong khi 'Day' và 'Year' lần lượt đại diện cho ngày trong tháng và năm với kiểu dữ liệu là số nguyên. Cột 'AvgTemperature' chứa thông tin về nhiệt độ trung bình, được biểu diễn dưới dạng số thực.

```
# Đếm số lượng giá trị duy nhất trong mỗi cột và sắp xếp theo thứ tự tăng dần
unique_counts = df.nunique().sort_values(ascending=True)
unique_counts
```

Region	7
Month	12
Year	28
Day	32
State	52
Country	125
City	321
AvgTemperature	1517

Hình 8. Thông tin về bộ dữ liệu

- **Region:** Có 7 giá trị duy nhất trong cột "Region". Điều này cho biết có 7 khu vực địa lý khác nhau trong tập dữ liệu.
- **Month:** Có 12 giá trị duy nhất trong cột "Month". Điều này phản ánh sự đa dạng của dữ liệu theo tháng trong năm.
- **Year:** Có 28 giá trị duy nhất trong cột "Year". Điều này cho thấy dữ liệu được thu thập trong suốt 26 năm.
- **Day:** Có 32 giá trị duy nhất trong cột "Day". Điều này có thể đại diện cho số ngày trong một tháng.
- **State:** Có 52 giá trị duy nhất trong cột "State". Điều này có thể là các bang/tỉnh khác nhau.
- **Country:** Có 125 giá trị duy nhất trong cột "Country". Điều này thể hiện sự đa dạng về quốc gia trong dữ liệu.
- **City:** Có 321 giá trị duy nhất trong cột "City". Điều này cho biết có 321 thành phố khác nhau trong tập dữ liệu.
- **AvgTemperature:** Có 1,517 giá trị duy nhất trong cột "AvgTemperature". Điều này làm thấy hiểu rằng có 1,517 mức độ nhiệt độ trung bình khác nhau trong dữ liệu.

```
df["Region"].value_counts()
```

North America	1556681
Europe	381990
Asia	316663
Africa	251118
South/Central America & Carribean	219530
Middle East	124749
Australia/South Pacific	55596

Hình 9. Số lượng các mẫu trong cột Region

- North America (Bắc Mỹ): Có 1,556,681 mẫu dữ liệu thuộc khu vực Bắc Mỹ.
- Europe (Châu Âu): Có 381,990 mẫu dữ liệu thuộc khu vực Châu Âu.

- Asia (Châu Á): Có 316,663 mẫu dữ liệu thuộc khu vực Châu Á.
- Africa (Châu Phi): Có 251,118 mẫu dữ liệu thuộc khu vực Châu Phi.
- South/Central America & Carribean (Nam/Central Mỹ & Carribean): Có 219,530 mẫu dữ liệu thuộc khu vực Nam/Central Mỹ & Carribean.
- Middle East (Trung Đông): Có 124,749 mẫu dữ liệu thuộc khu vực Trung Đông.
- Australia/South Pacific (Úc/Thái Bình Dương): Có 55,596 mẫu dữ liệu thuộc khu vực Úc/Thái Bình Dương.

```
df["Country"].value_counts()

US          1455337
Canada      74245
Australia   46330
China       46329
India       37063
...
Guyana      5065
Israel      4641
Burundi     4543
Georgia     4378
Serbia-Montenegro 3427
Name: Country, Length: 125, dtype: int64

[206] df['City'].value_counts()

Springfield 18530
Columbus    18530
Portland    18530
Washington DC 18530
Washington  18530
...
Frankfurt   4136
Flagstaff   3574
Pristina    3427
Yerevan     3226
Bonn        3133
Name: City, Length: 321, dtype: int64
```

Hình 10. Số lượng các mẫu trong cột Country và City

Trong cột "City", có tổng cộng 321 giá trị duy nhất, và số lần xuất hiện của mỗi giá trị chỉ ra tần suất của các thành phố khác nhau. Điều này giúp ta hiểu về sự đa dạng của các đơn vị địa lý cụ thể và tần suất xuất hiện của chúng trong dữ liệu.

Tương tự, trong cột "Country", có 125 giá trị duy nhất và số lần xuất hiện của mỗi giá trị chỉ ra tần suất xuất hiện của các quốc gia khác nhau. Thông qua các con số này, chúng ta có thể đánh giá mức độ phổ biến và đa dạng của các quốc gia trong tập dữ liệu.

4.2.2. Tiền xử lý dữ liệu

```
# Kiểm tra số lượng giá trị thiếu trong từng cột của DataFrame
missing_values = df.isnull().sum()
missing_values
```

Region	0
Country	0
State	1450990
City	0
Month	0
Day	0
Year	0
AvgTemperature	0
dtype:	int64

Hình 11. Xử lý dữ liệu

Cột "State" có 1,450,990 giá trị thiếu, có thể đề xuất rằng một số thông tin về bang/tỉnh có thể bị thiếu trong một số trường hợp. Quyết định xử lý giá trị thiếu trong cột này sẽ phụ thuộc vào mục tiêu cụ thể của phân tích, có thể bao gồm việc điền giá trị thiếu hoặc loại bỏ các mẫu dữ liệu tương ứng.

Các cột khác như "City", "Month", "Day", "Year", và "AvgTemperature" đều không có giá trị thiếu, giúp bảo đảm tính đầy đủ của thông tin thời tiết cũng như các thông tin thời gian liên quan.

```
(df['State'].isna().sum() / df['State'].shape[0])*100
```

```
49.92521488462929
```

```
[209] df = df.drop('State', axis=1)
```

Hình 12. Xử lý dữ liệu State

Kết quả là 49.93%, nghĩa là gần 50% các mẫu dữ liệu trong cột "State" có giá trị thiếu. Điều này chỉ ra rằng gần một nửa các mẫu dữ liệu không có thông tin về bang/tỉnh. Với việc thiếu quá nhiều thì em dùng phương pháp xóa cột 'State' ra khỏi bộ dữ liệu.

```
print(df.Day.unique())  
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24  
 25 26 27 28 29 30 31  0]  
  
print(df.Year.unique())  
[1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008  
 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020  201  200]  
  
[214] df =df[ (df['Year'] != 200) & (df['Year'] != 201) & (df['Day'] != 0) ]
```

Hình 13.Xử lý dữ liệu

Đoạn mã thực hiện quá trình lọc dữ liệu trong DataFrame dựa trên một số điều kiện cụ thể. Mục tiêu của việc này có thể là loại bỏ các hàng không phù hợp hoặc chứa giá trị ngoại lệ để chuẩn bị dữ liệu cho các phân tích tiếp theo.

Trong quá trình này, dữ liệu được lọc để chỉ giữ lại những hàng thỏa mãn các điều kiện sau:

- Giá trị trong cột 'Year' không phải là 200 và không phải là 201.
- Giá trị trong cột 'Day' không phải là 0.

Điều này giúp làm sạch dữ liệu và loại bỏ những giá trị ngoại lệ hoặc không phù hợp, đồng thời tăng tính chính xác và độ tin cậy của dữ liệu trong quá trình phân tích. Việc lọc dữ liệu như vậy thường được thực hiện trong quá trình tiền xử lý để chuẩn bị dữ liệu cho các bước phân tích và mô hình hóa sau này.

```
df['Date'] = pd.to_datetime(df[['Year', 'Month', 'Day']])
```

```
df.sample(10)
```

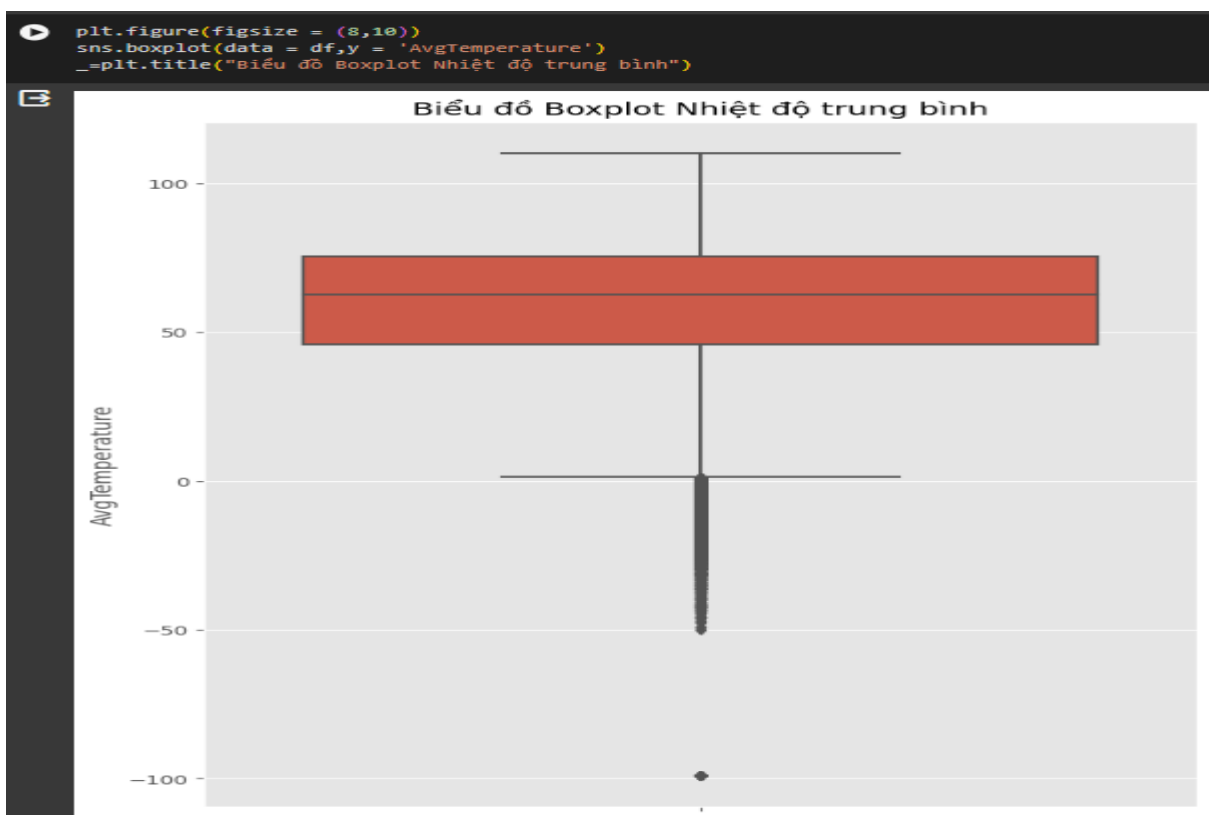
	Region	Country	City	Month	Day	Year	AvgTemperature	Date
2635479	North America	US	Abilene	8	31	2007	76.6	2007-08-31
1854230	North America	US	Rockford	9	24	2005	64.6	2005-09-24
2487649	North America	US	Salem	3	2	1999	40.8	1999-03-02
2540718	North America	US	Wilkes Barre	11	19	2001	45.1	2001-11-19
336287	Asia	India	Chennai (Madras)	4	29	2009	89.0	2009-04-29
2175389	North America	US	Great Falls	9	10	2006	69.3	2006-09-10
1861039	North America	US	Springfield	1	3	1999	15.7	1999-01-03
1816647	North America	US	Boise	4	19	2004	49.2	2004-04-19
2540693	North America	US	Wilkes Barre	10	25	2001	65.3	2001-10-25
1855607	North America	US	Rockford	7	2	2009	62.5	2009-07-02

Hình 14. Xử lý dữ liệu datetime

Dòng mã `df['Date'] = pd.to_datetime(df[['Year', 'Month', 'Day']])` có tác dụng thêm một cột mới vào DataFrame `df` mang tên 'Date', trong đó chứa thông tin về ngày tháng năm được tạo ra từ cột 'Year', 'Month', và 'Day'. Quá trình này sử dụng hàm `pd.to_datetime` của thư viện Pandas để chuyển đổi thông tin thời gian thành đối tượng datetime.

Đối với mỗi hàng trong DataFrame, giá trị của cột 'Date' sẽ được tạo ra bằng cách kết hợp thông tin từ cột 'Year', 'Month', và 'Day', và sau đó chuyển đổi thành định dạng datetime. Việc này giúp biến đổi dữ liệu thời gian thành một dạng có thể được sử dụng dễ dàng trong các phân tích và mô hình hóa dự báo về nhiệt độ.

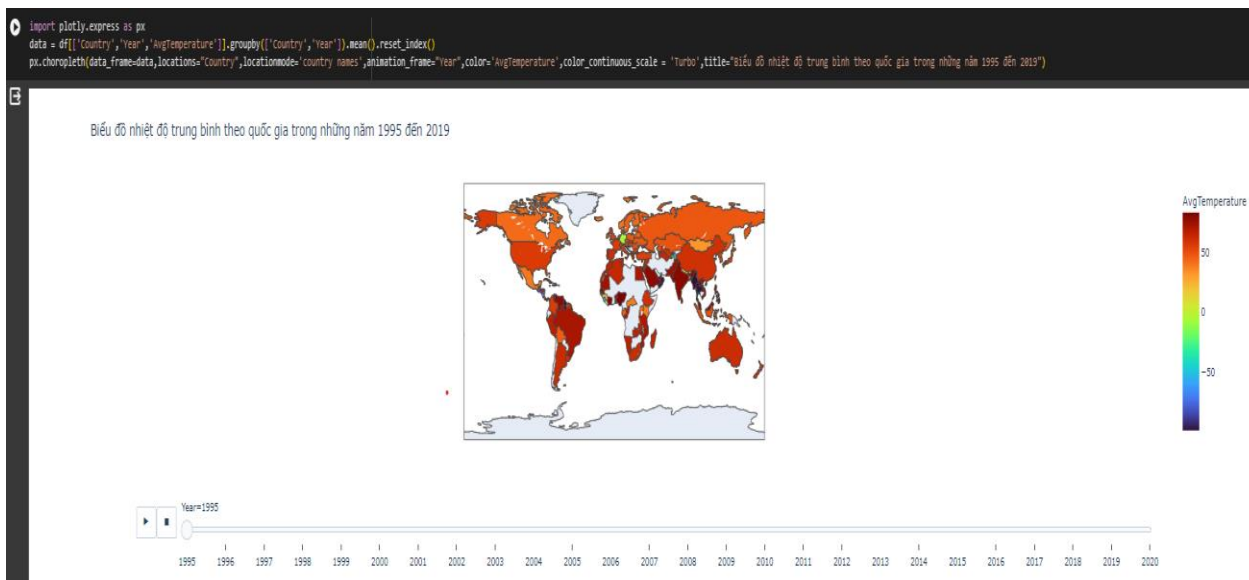
Thêm cột 'Date' này có thể giúp trong việc thực hiện các phân tích dựa trên dữ liệu thời gian như tìm hiểu xu hướng thay đổi nhiệt độ theo thời gian, xác định mùa vụ, hoặc thực hiện các mô hình dự đoán dựa trên dữ liệu chuỗi thời gian.



Hình 15. Biểu đồ boxplot

Biểu đồ boxplot được sử dụng để hiển thị phân phối và các đặc tính thống kê cơ bản của một biến số. Trong trường hợp này, biểu đồ boxplot cho cột 'AvgTemperature' sẽ thể hiện các thông tin như giá trị trung bình, phạm vi giá trị, và phân phối nhiệt độ trung bình. Qua đó giúp chọn các giá trị 'AvgTemperature' phù hợp cho các mô hình học máy.

4.3.3.Trực quan hoá dữ liệu



Hình 16. Biểu đồ choropleth

Đoạn mã trên là một công cụ mạnh mẽ để thể hiện biến động của nhiệt độ trung bình theo quốc gia trong khoảng thời gian từ 1995 đến 2020. Mỗi frame của biểu đồ tương ứng với một năm cụ thể, cung cấp cái nhìn đồng thời về sự biến thiên của nhiệt độ trung bình và mối quan hệ địa lý.

Bằng cách sử dụng màu sắc để đại diện cho mức độ nhiệt độ trung bình, biểu đồ cho phép so sánh giữa các quốc gia và nhận diện rõ ràng các khu vực có biến động nhiệt độ đặc biệt. Điều này giúp xác định xu hướng thay đổi khí hậu và tác động của chúng đối với từng quốc gia.

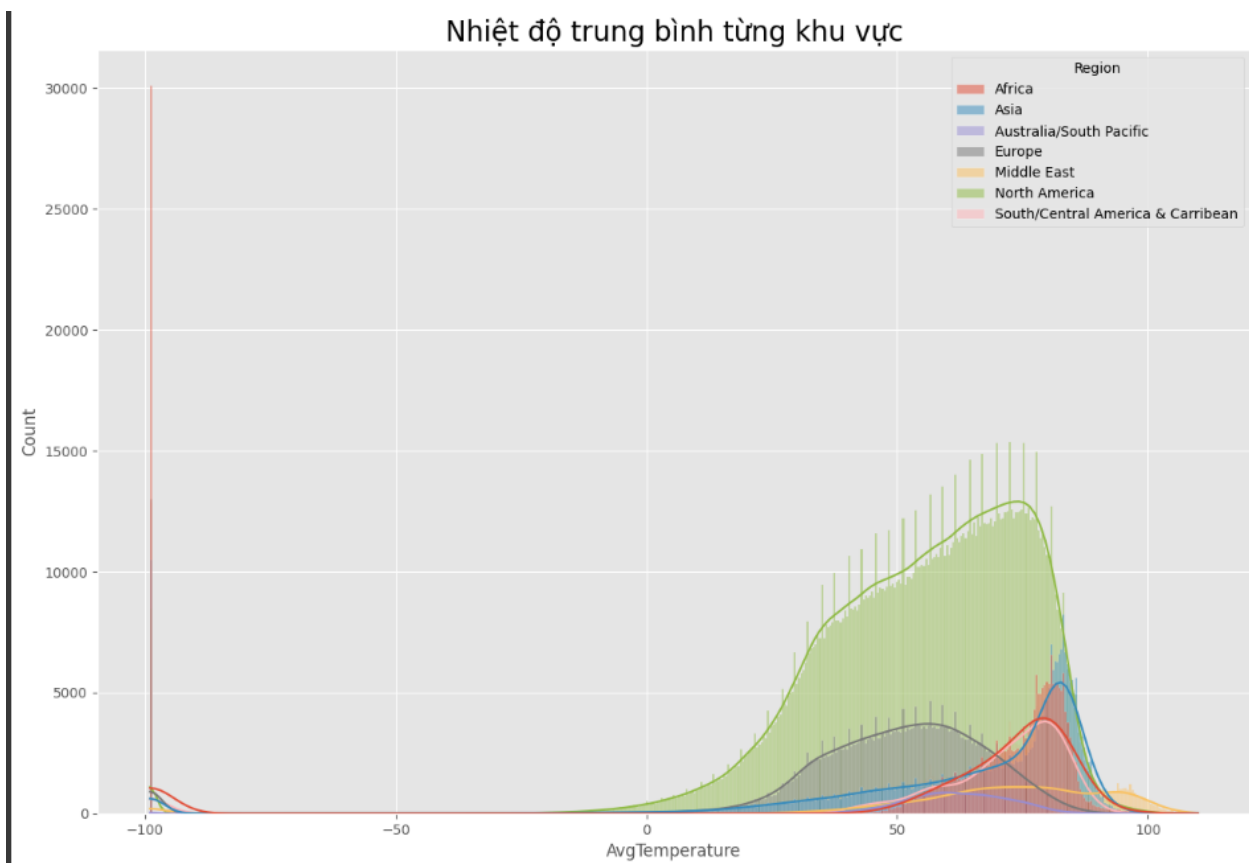
Ngoài ra, khả năng theo dõi sự thay đổi theo thời gian và phân tích chi tiết từng frame cho phép nhận biết các sự kiện đặc biệt hoặc biến động nhiệt độ đặc thù trong các khu vực địa lý cụ thể. Tổng cộng, biểu đồ choropleth động là một công cụ hiệu quả, trực quan hóa sự biến động nhiệt độ trung bình và giúp chúng ta hiểu rõ hơn về tình hình khí hậu toàn cầu.



Hình 17. Biểu đồ cột về nhiệt độ trung bình

Biểu đồ cột này cho thấy mức độ nhiệt độ trung bình của mỗi khu vực và giúp tạo ra một cái nhìn tổng quan về sự chênh lệch nhiệt độ giữa các khu vực khác nhau trên thế giới. Dựa trên mức độ nhiệt độ trung bình, chúng ta có thể xác định được những khu vực nào có xu hướng nóng hơn hoặc lạnh hơn so với các khu vực khác.

Cụ thể, nếu nhiệt độ trung bình được sắp xếp theo thứ tự tăng dần, biểu đồ có thể cung cấp thông tin về sự biến động của khí hậu và tác động của nó đối với từng khu vực trên thế giới.

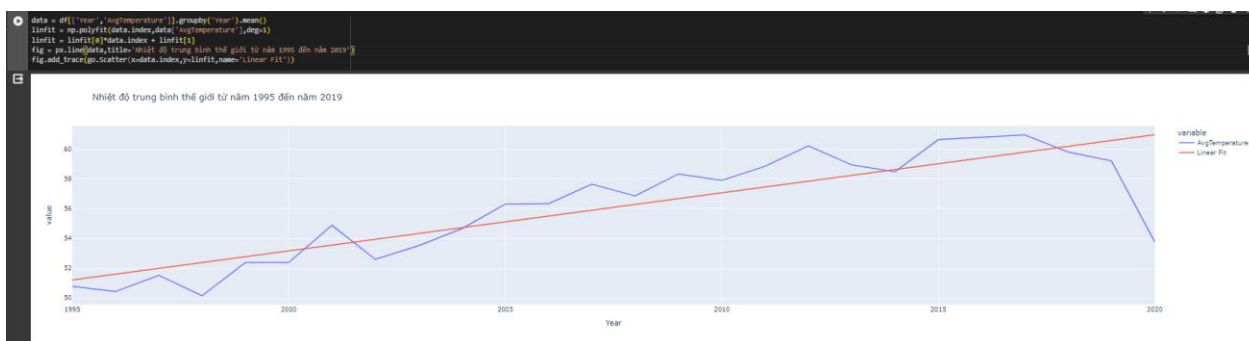


Hình 18. Biểu đồ nhiệt độ trung bình từng khu vực

Biểu đồ nhiệt độ trung bình từng khu vực là một công cụ quan trọng để hiểu sự chênh lệch về mức độ nhiệt độ giữa các khu vực trên thế giới. Thông qua biểu đồ, ta có thể đánh giá và so sánh mức độ nhiệt độ trung bình của mỗi khu vực, điều này mang lại nhiều thông tin hữu ích về khí hậu và đặc điểm thời tiết của từng vùng địa lý.

Biểu đồ cung cấp cái nhìn tổng quan về sự đa dạng của khí hậu trên thế giới, từ những khu vực nhiệt đới có nhiệt độ cao đến các vùng lạnh có nhiệt độ thấp. Nếu biểu đồ được thiết kế để theo dõi thay đổi theo thời gian, nó còn giúp định lượng và theo dõi xu hướng biến động nhiệt độ trong từng khu vực.

Bằng cách nhìn vào biểu đồ, ta có thể xác định những khu vực có nhiệt độ trung bình đặc biệt cao hoặc thấp, đồng thời hiểu rõ về đặc điểm khí hậu của từng khu vực. Biểu đồ còn giúp liên kết sự biến động nhiệt độ với thay đổi khí hậu toàn cầu, mang lại cái nhìn tổng thể về tác động của biến đổi khí hậu đối với các khu vực địa lý trên thế giới.



Hình 19. Biểu đồ Linear Fit

Biểu đồ này có công dụng là trình bày và hiển thị sự biến động của nhiệt độ trung bình thế giới từ năm 1995 đến 2019. Đồng thời, đường thẳng hồi quy tuyến tính được thêm vào giúp thể hiện xu hướng chung của dữ liệu theo thời gian, cho phép nhận biết sự thay đổi dài hạn của nhiệt độ trung bình toàn cầu. Phương pháp hồi quy tuyến tính giúp xác định xu hướng và dự đoán giá trị tiếp theo dựa trên dữ liệu chuỗi thời gian.

4.3. Xây dựng mô hình dự đoán

```
df = df[df['AvgTemperature'] > -80]
[227] data = df[df['Country'] == 'China']
[228] bj = data[data['City'] == 'Beijing']
```

Hình 20. Chọn dữ liệu

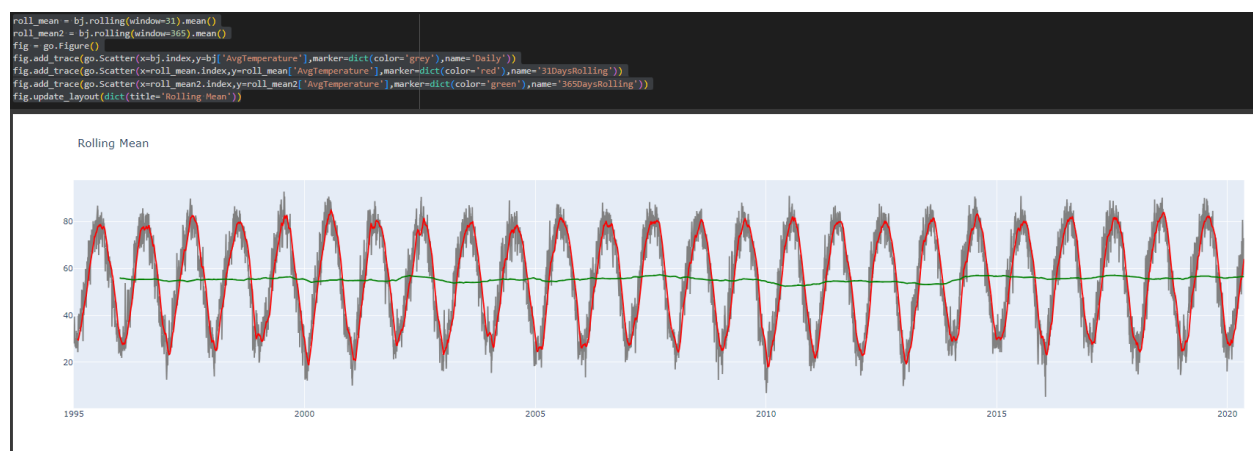
Nhờ biểu đồ BoxPlot trên em đã giúp em chọn giá trị thích hợp và đất nước Trung Quốc và thành phố Beijing được chọn để xây dựng mô hình máy học.

```
size = bj.groupby(['Year', 'Month']).size().reset_index()
size_max = size[0].max()
size_min = size[0].min()
n = size_max - size_min + 1
cmap = sns.color_palette("deep", n)
size = size.pivot(index='Year', columns='Month', values=0)
size.head(3)
```

Month	1	2	3	4	5	6	7	8	9	10	11	12
Year												
1995	31.0	28.0	31.0	30.0	31.0	30.0	31.0	31.0	30.0	31.0	30.0	31.0
1996	31.0	29.0	31.0	30.0	31.0	30.0	31.0	31.0	30.0	31.0	30.0	31.0
1997	31.0	28.0	31.0	30.0	31.0	30.0	31.0	31.0	30.0	31.0	30.0	31.0

Hình 21. Kiểm tra dữ liệu ngày trong tháng

Đoạn mã giúp kiểm tra dữ liệu có bị thiếu trong từng tháng để giúp khắc phục bằng xử lý dữ liệu thông qua đó tăng độ chính xác mô hình.



Hình 22. Rolling mean

Trong đoạn mã trên, sử dụng rolling mean giúp làm mịn và làm giảm nhiễu dữ liệu nhiệt độ hàng ngày, tạo ra một biểu đồ trực quan về xu hướng và biến động dài hạn. Đoạn mã tạo ra ba đường biểu đồ để thể hiện các khía cạnh khác nhau của dữ liệu:

Đầu tiên, đường biểu đồ hàng ngày (màu xám) thể hiện biến động tức thì của nhiệt độ. Sau đó, đường biểu đồ với rolling mean 31 ngày (màu đỏ) giúp làm mịn dữ liệu, làm giảm đột ngột và nhấn mạnh xu hướng trung bình trong khoảng ngắn hạn. Cuối cùng, đường biểu đồ với rolling mean 365 ngày (màu xanh) tạo ra một cái nhìn tổng quan về mô hình nhiệt độ theo mùa và giảm nhiễu ngắn hạn.

Công dụng chính của rolling mean trong trường hợp này là tạo ra các biểu đồ có thể giúp nhìn ra xu hướng dài hạn, phân biệt giữa ngày và đêm, xác định mô hình nhiệt độ theo mùa, và làm mềm đường biểu đồ để dễ dàng so sánh và đánh giá sự biến động của dữ liệu theo thời gian.

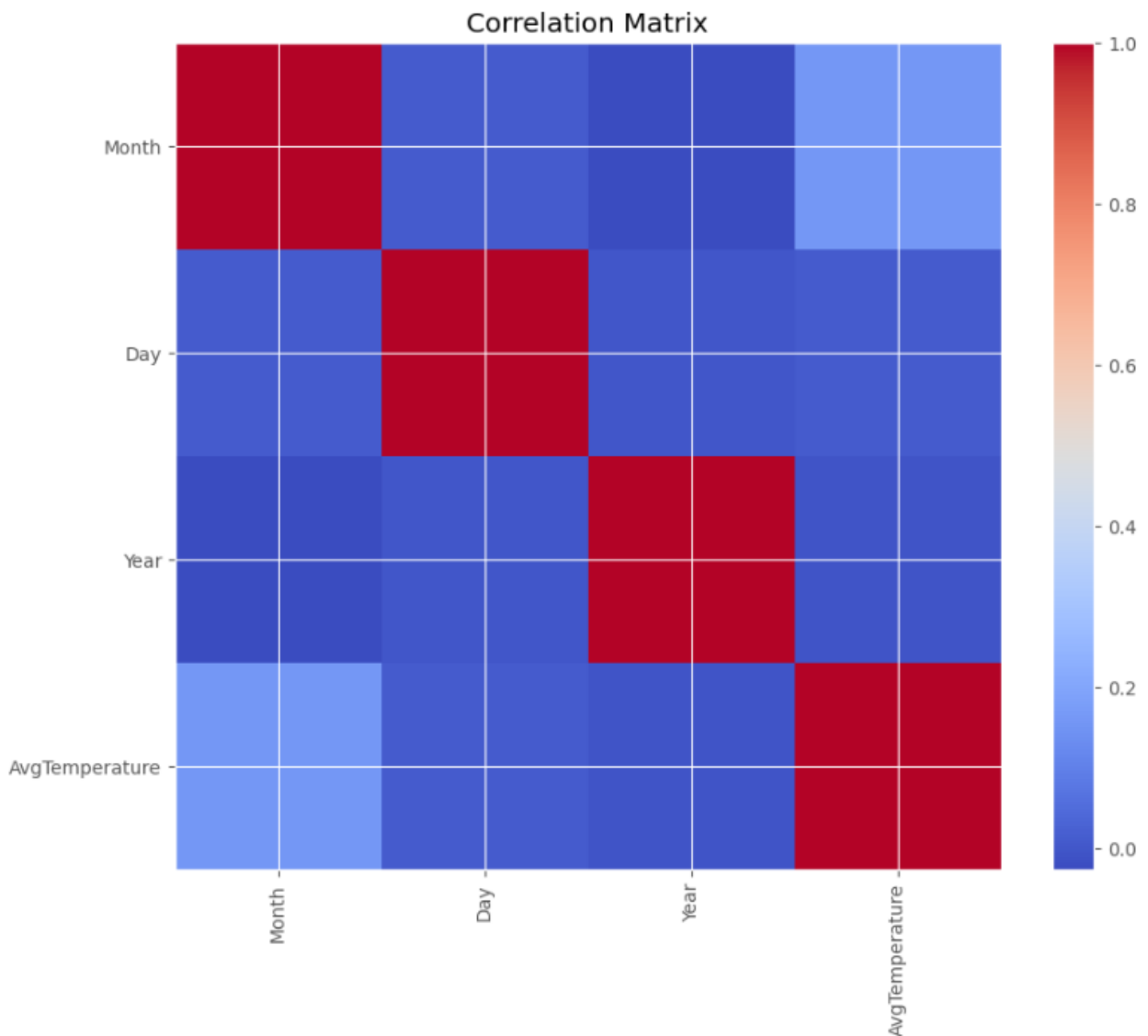


Hình 23. Rolling STD

Trong đoạn mã trên, việc sử dụng rolling standard deviation (độ lệch chuẩn trượt) đóng vai trò quan trọng trong việc đánh giá và đo lường sự biến động của dữ liệu nhiệt độ hàng ngày. Điều này giúp ta nhìn nhận rõ hơn về cả sự biến động ngắn hạn và dài hạn của nhiệt độ theo thời gian.

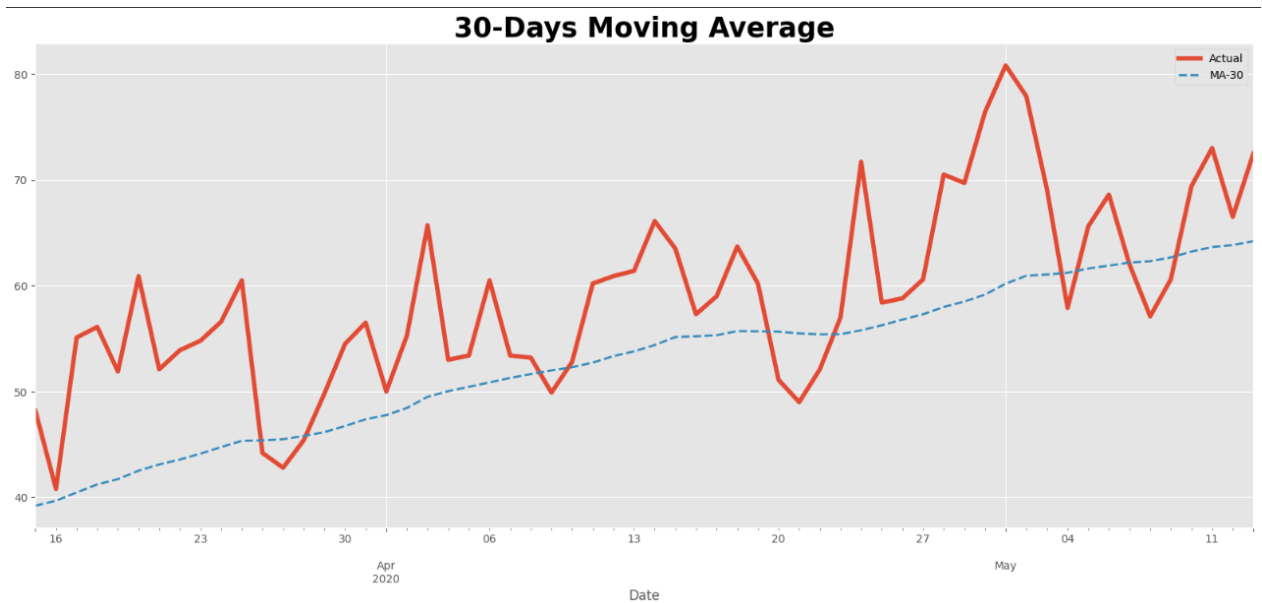
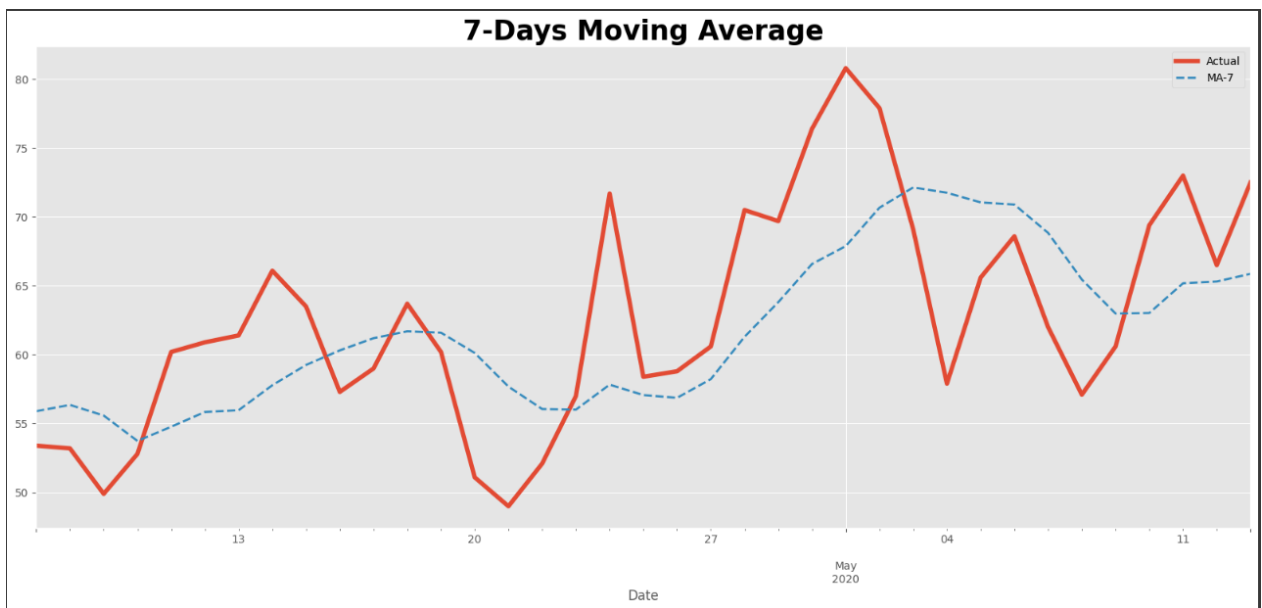
Đường biểu đồ với rolling standard deviation 31 ngày (màu đỏ) thường phản ánh sự biến động ngắn hạn, có thể liên quan đến các biến động thời tiết ngày và đêm. Trong khi đó, đường biểu đồ với rolling standard deviation 365 ngày (màu xanh) làm nổi bật sự biến động dài hạn, có thể phản ánh các mô hình thay đổi mùa vụ hoặc xu hướng dài hạn của dữ liệu.

Công dụng chính của rolling standard deviation là giúp ta phát hiện và đánh giá sự biến động, rủi ro và tính ổn định của dữ liệu nhiệt độ. Việc này có thể hữu ích trong việc đưa ra dự báo và hiểu rõ hơn về đặc điểm thời tiết trong quá khứ và tương lai.



Hình 24. Ma trận tương quan

Ma trận tương quan (Correlation Matrix) là một công cụ quan trọng trong dự báo nhiệt độ và nghiên cứu về thời tiết. Ma trận này giúp đánh giá mối quan hệ tuyến tính giữa các biến số liên quan đến nhiệt độ, giúp nhận biết xu hướng và mức độ ảnh hưởng của chúng lên nhau.

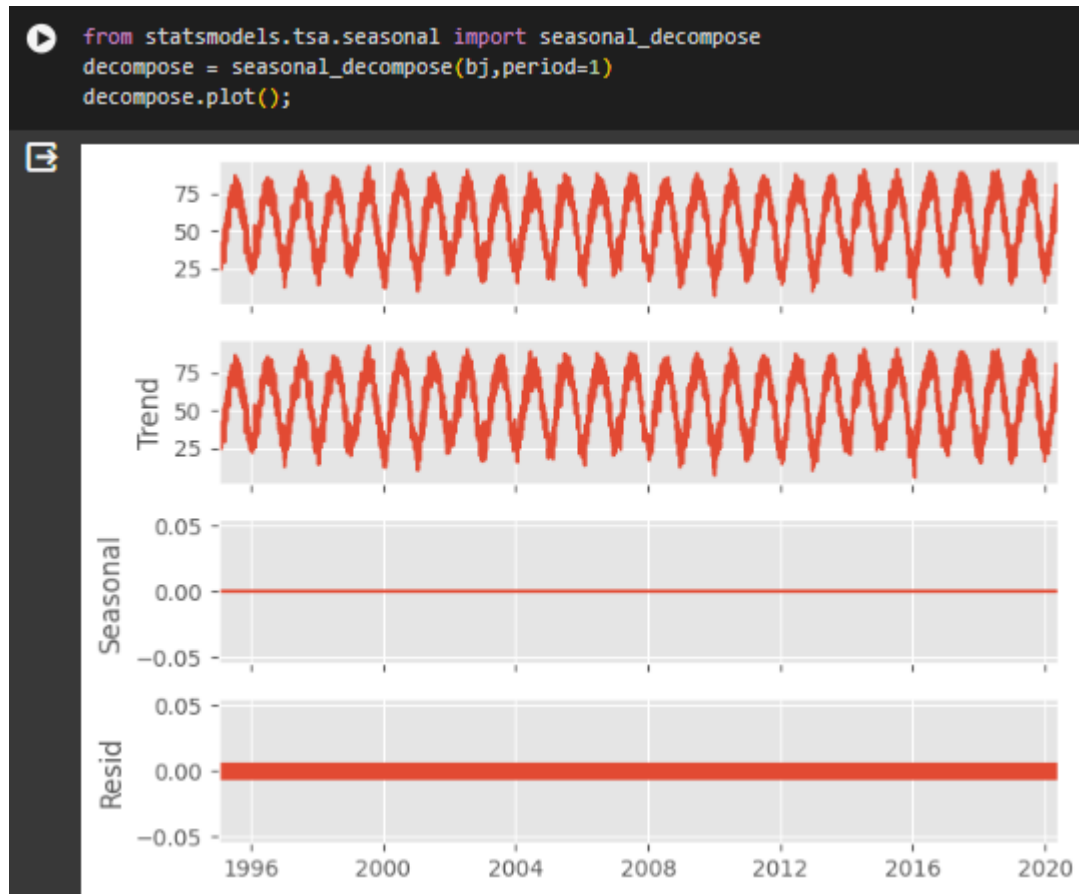


Hình 25. Moving Average

Trên biểu đồ, đường thực tế của nhiệt độ được biểu diễn bằng đường dày với đường trung bình chạy (MA) có thể làm sáng tỏ xu hướng chung của dữ liệu. Các công dụng quan trọng của biểu đồ này bao gồm:

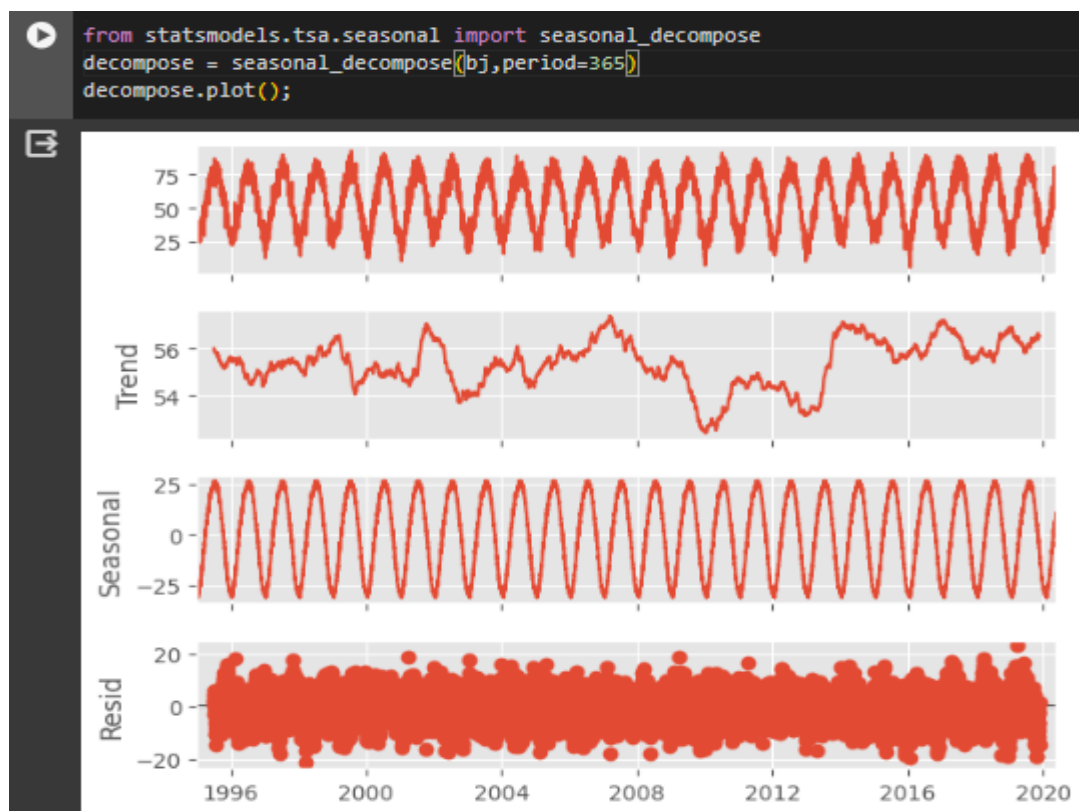
- Giúp chúng ta đánh giá xu hướng của nhiệt độ theo thời gian, từ đó có thể dự đoán mô hình chu kỳ và nhận diện sự biến động theo mùa vụ. Điều này giúp chúng ta hiểu rõ hơn về các thay đổi thời tiết và ảnh hưởng của chúng lên nhiệt độ.

- Biểu đồ giúp phân tích mối quan hệ giữa dữ liệu thực tế và đường MA. Bằng cách so sánh sự biến động giữa hai đường này, chúng ta có thể đánh giá mức độ biến động và xác định sự không ổn định trong thời tiết.
- Chú thích rõ ràng trên biểu đồ giúp người xem dễ dàng hiểu thông tin và nhận diện các điểm quan trọng, như sự tăng đột ngột hoặc giảm đột ngột của nhiệt độ.



Hình 26. Biểu đồ phân tích mùa vụ (1 ngày)

Biểu đồ chuỗi thời gian cung cấp một cái nhìn tổng quan về dữ liệu theo dõi hàng ngày. Phần ‘AvgTemperature’ thể hiện biến động hàng ngày dường như có một xu hướng tăng nhẹ giao động trong khoảng 25 độ F – 75 độ F, mặc dù có những biến động ngắn hạn. Trend làm nổi bật xu hướng dài hạn của dữ liệu, loại bỏ ảnh hưởng của biến động ngắn hạn và yếu tố mùa vụ.



Hình 27. Biểu đồ phân tích mùa vụ (365 ngày)

Đây là một phân tích tổng quan về dữ liệu được ghi nhận hàng ngày, với chu kỳ mùa vụ là 365 ngày. Các thành phần chính của biểu đồ đã được phân tích chi tiết: ‘AvgTemperature’ biểu diễn giá trị thực tế của chuỗi thời gian từ năm 1995 đến 2020. Xu hướng giao động đều theo thời gian là rõ ràng trong dữ liệu thô này. Đường xu hướng chỉ ra sự tăng trưởng dài hạn của dữ liệu, loại bỏ biến động ngắn hạn và ảnh hưởng mùa vụ. Có xu hướng không đổi qua các năm. Biểu đồ mùa vụ cho thấy biến động định kỳ trong dữ liệu, có thể phản ánh các yếu tố ảnh hưởng đến nhiệt độ trung bình. Mô hình định kỳ hàng năm là rõ ràng, với các đỉnh và đáy xảy ra vào cùng thời điểm mỗi năm. Resid phần dư thể hiện sự biến động ngẫu nhiên hoặc không giải thích được sau khi loại bỏ xu hướng và mùa vụ. Điều này có thể là do những yếu tố ngoại cảnh hoặc biến động ngẫu nhiên trong dữ liệu.

```
test_bj = bj[bj.index>'2019']
train_bj = bj[bj.index<'2019']

[243] scaler = MinMaxScaler()
train_bj = scaler.fit_transform(train_bj)
test_bj = scaler.transform(test_bj)

[244] from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM,Dense
from tensorflow.keras.preprocessing.sequence import TimeseriesGenerator
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.callbacks import EarlyStopping
time_steps = 20
features = 1

train_gen = TimeseriesGenerator(train_bj,train_bj,time_steps,batch_size=32)
test_gen = TimeseriesGenerator(test_bj,test_bj,time_steps,batch_size=32)

[245] model = Sequential()
model.add(LSTM(64,activation='relu',input_shape=(time_steps,features),return_sequences=True))
model.add(LSTM(32,activation='relu'))
model.add(Dense(1,activation='relu'))
model.compile(optimizer='adam',loss='mse')

model.summary()

Model: "sequential_2"

Layer (type)                 Output Shape              Param #
=====
lstm_4 (LSTM)                 (None, 20, 64)           16896
lstm_5 (LSTM)                 (None, 32)               12416
dense_2 (Dense)               (None, 1)                33
=====
Total params: 29345 (114.63 KB)
Trainable params: 29345 (114.63 KB)
Non-trainable params: 0 (0.00 Byte)
```

Hình 28.Mô hình LSTM

Sử dụng lớp TimeseriesGenerator từ thư viện để chuẩn bị dữ liệu đầu vào cho mô hình chuỗi thời gian. Bước này là quan trọng để mô hình có thể học được mối quan hệ và dự đoán các giá trị tiếp theo trong chuỗi thời gian.

Lớp TimeseriesGenerator thường được sử dụng để chia dữ liệu chuỗi thời gian thành các cặp (đầu vào, đầu ra) phù hợp cho mô hình học máy. Đối với mỗi bước thời gian, nó sẽ

xây dựng một "cửa sổ trượt" có kích thước `time_steps=20`, sử dụng các điểm dữ liệu liên kế để dự đoán điểm dữ liệu tiếp theo.

Lớp LSTM thứ nhất (`lstm_4`):

- Loại: LSTM
- Số lượng đơn vị LSTM: 64
- Đầu ra của lớp: (None, 20, 64)
- Số lượng tham số: 16,896
- Ghi chú: Lớp này nhận đầu vào là một dãy thời gian với kích thước (None, 20, features), trong đó 20 là số bước thời gian và 64 là số đơn vị LSTM.

Lớp LSTM thứ hai (`lstm_5`):

- Loại: LSTM
- Số lượng đơn vị LSTM: 32
- Đầu ra của lớp: (None, 32)
- Số lượng tham số: 12,416
- Ghi chú: Lớp này nhận đầu vào từ lớp LSTM trước đó và trả về một đầu ra không phải là dãy thời gian.

Lớp Dense thứ ba (`dense_2`):

- Loại: Dense
- Số lượng đơn vị: 1
- Đầu ra của lớp: (None, 1)
- Số lượng tham số: 33

```
[256] x = selected_data[features]
      y = selected_data[target]

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

[257] print("Kích thước tập huấn luyện:", X_train.shape)
      print("Kích thước tập kiểm tra:", X_test.shape)

      Kích thước tập huấn luyện: (7389, 3)
      Kích thước tập kiểm tra: (1848, 3)

      linear_model = LinearRegression()

[259] linear_model.fit(X_train, y_train)

      LinearRegression
      LinearRegression()
```

Hình 29. Mô hình LinearRegression

Giá trị của biến mục tiêu (target) dựa trên các biến độc lập (features) đã được thực hiện thông qua một mô hình hồi quy tuyến tính. Đầu tiên, dữ liệu được chia thành hai phần chính: tập huấn luyện và tập kiểm tra với tỷ lệ 80-20. Sau đó, một mô hình hồi quy tuyến tính được tạo và huấn luyện trên tập huấn luyện.

Kích thước của tập huấn luyện và tập kiểm tra được hiển thị để kiểm tra độ lớn của dữ liệu trong từng phần. Mô hình được huấn luyện trên tập huấn luyện để hiểu mối quan hệ giữa các biến độc lập và biến mục tiêu. Sau khi mô hình được huấn luyện, nó được sử dụng để dự đoán giá trị trên tập kiểm tra.

Kết quả dự đoán được lưu trong biến `y_pred_linear`. Các giá trị dự đoán này có thể được sử dụng để đánh giá hiệu suất của mô hình bằng cách so sánh chúng với giá trị thực tế từ tập kiểm tra. Điều này cung cấp cái nhìn về khả năng dự đoán của mô hình và mức độ chính xác của nó trong việc ước lượng giá trị của biến mục tiêu dựa trên các biến độc lập đã chọn

```
[262] # Khởi tạo mô hình
      tree_model = DecisionTreeRegressor(random_state=42)

[263] # Huấn luyện mô hình
      tree_model.fit(X_train, y_train)

      DecisionTreeRegressor
      DecisionTreeRegressor(random_state=42)

[264] # Dự đoán trên tập kiểm tra
      y_pred_tree = tree_model.predict(X_test)

[265] print("Giá trị dự đoán trên tập kiểm tra (Decision Tree):", y_pred_tree)

      Giá trị dự đoán trên tập kiểm tra (Decision Tree): [34.1 49.1 29. ... 61.9 68.9 20.9]
```

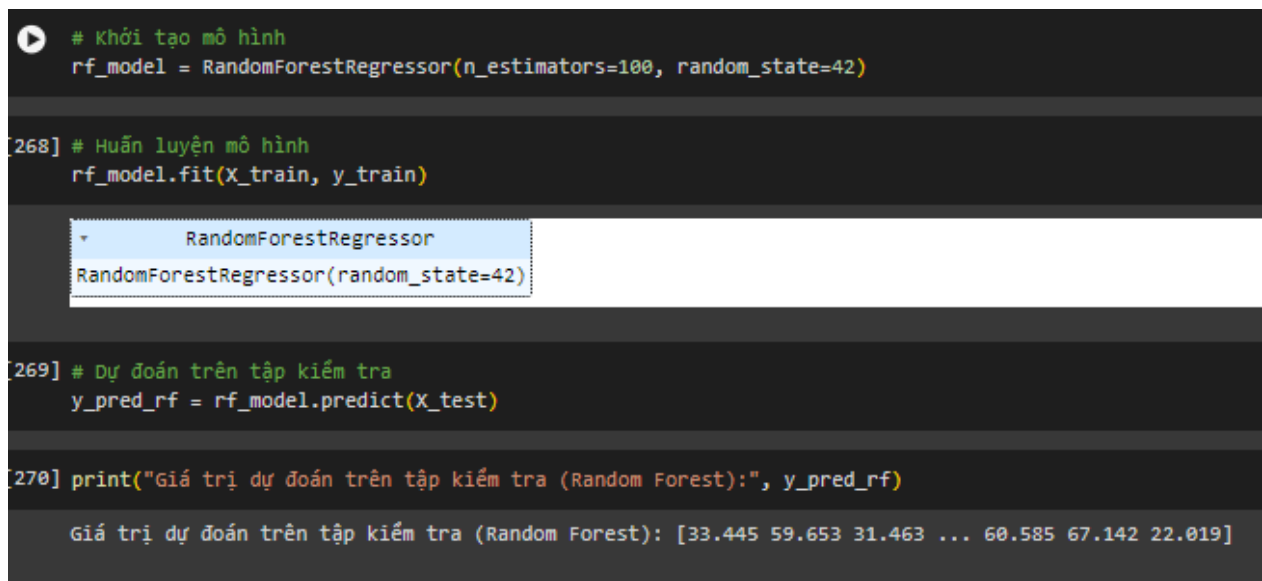
Hình 30. Mô hình Decision Tree

Mô hình cây quyết định (Decision Tree). Mô hình này được tạo ra để dự đoán giá trị của biến mục tiêu dựa trên các biến độc lập từ tập dữ liệu huấn luyện.

Đầu tiên, một đối tượng mô hình cây quyết định được khởi tạo với tham số `random_state` để đảm bảo sự tái tạo của kết quả. Sau đó, mô hình được huấn luyện trên tập

dữ liệu huấn luyện để học cách đưa ra dự đoán dựa trên các quy tắc quyết định được áp dụng cho các biến độc lập.

Sau quá trình huấn luyện, mô hình được sử dụng để dự đoán giá trị trên tập kiểm tra. Các giá trị dự đoán được lưu trữ trong biến `y_pred_tree`. Bằng cách so sánh giá trị dự đoán này với giá trị thực tế từ tập kiểm tra, chúng ta có thể đánh giá chất lượng và hiệu suất của mô hình cây quyết định trong việc dự đoán giá trị của biến mục tiêu.



```
# Khởi tạo mô hình
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

[268] # Huấn luyện mô hình
rf_model.fit(X_train, y_train)

RandomForestRegressor
RandomForestRegressor(random_state=42)

[269] # Dự đoán trên tập kiểm tra
y_pred_rf = rf_model.predict(X_test)

[270] print("Giá trị dự đoán trên tập kiểm tra (Random Forest):", y_pred_rf)

Giá trị dự đoán trên tập kiểm tra (Random Forest): [33.445 59.653 31.463 ... 60.585 67.142 22.019]
```

Hình 31. Mô hình Random Forest

Đầu tiên, một đối tượng mô hình Random Forest được khởi tạo với 100 cây quyết định (`n_estimators=100`) và `random_state=42` để đảm bảo sự tái tạo kết quả. Sau đó, mô hình được huấn luyện trên tập dữ liệu huấn luyện để học cách kết hợp thông tin từ nhiều cây quyết định khác nhau.

Sau quá trình huấn luyện, mô hình Random Forest được sử dụng để dự đoán giá trị trên tập kiểm tra. Các giá trị dự đoán được lưu trong biến `y_pred_rf`. Bằng cách so sánh giá trị dự đoán này với giá trị thực tế từ tập kiểm tra, chúng ta có thể đánh giá chất lượng và hiệu suất của mô hình Random Forest trong việc dự đoán giá trị của biến mục tiêu.

```
[272] # Khởi tạo mô hình
xgb_model = XGBRegressor(objective='reg:squarederror', colsample_bytree = 0.3, learning_rate = 0.1,
                        max_depth = 5, alpha = 10, n_estimators = 100)

[273] # Huấn luyện mô hình
xgb_model.fit(X_train, y_train)

XGBRegressor(alpha=10, base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=0.3, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.1, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=5, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=100, n_jobs=None,
             num_parallel_tree=None, ...)

[274] # Dự đoán trên tập kiểm tra
y_pred_xgb = xgb_model.predict(X_test)

[275] print("Giá trị dự đoán trên tập kiểm tra (XGBoost):", y_pred_xgb)

Giá trị dự đoán trên tập kiểm tra (XGBoost): [30.002424 69.631584 45.77297 ... 57.88432 69.514435 27.515879]
```

Hình 32. Mô hình XGBoost

Chúng ta đã triển khai một mô hình dự đoán nhiệt độ trung bình sử dụng thuật toán XGBoost. Đầu tiên, chúng ta đã khởi tạo mô hình XGBoost với các tham số quan trọng như mục tiêu là hồi quy ('reg:squarederror'), tỷ lệ mẫu được sử dụng khi xây dựng mỗi cây (colsample_bytree), tỷ lệ học (learning_rate), độ sâu tối đa của mỗi cây (max_depth), hệ số alpha để kiểm soát độ lớn của các nút lá (alpha), và số cây con (n_estimators).

Sau đó, chúng ta đã huấn luyện mô hình trên tập dữ liệu huấn luyện và tiến hành dự đoán trên tập kiểm tra. Kết quả dự đoán từ mô hình đã được lưu trong biến y_pred_xgb. Mô hình XGBoost thường được ưa chuộng trong các ứng dụng hồi quy do khả năng xử lý tốt các tình huống phức tạp và khả năng tinh chỉnh tham số linh hoạt để đạt được hiệu suất cao. Đánh giá hiệu suất của mô hình có thể được thực hiện bằng cách so sánh giá trị dự đoán với giá trị thực tế sử dụng các độ đo như Mean Absolute Error (MAE) và Root Mean Squared Error (RMSE).

```

import math
X =preprocessed_df[features + [target]].copy()
y = preprocessed_df['AvgTemperature'].copy()

train_size = int(0.80 * len(preprocessed_df))
test_size = len(preprocessed_df) - train_size

X_train, y_train = pd.DataFrame(X.iloc[:train_size]), pd.DataFrame(y.iloc[:train_size])
X_test, y_test = pd.DataFrame(X.iloc[train_size:]), pd.DataFrame(y.iloc[train_size:])

print(len(X_train), len(X_test))

model = ARIMA(y_train, order=(0,1,1))
model_fit = model.fit()

y_pred = model_fit.forecast(len(X_test))

```

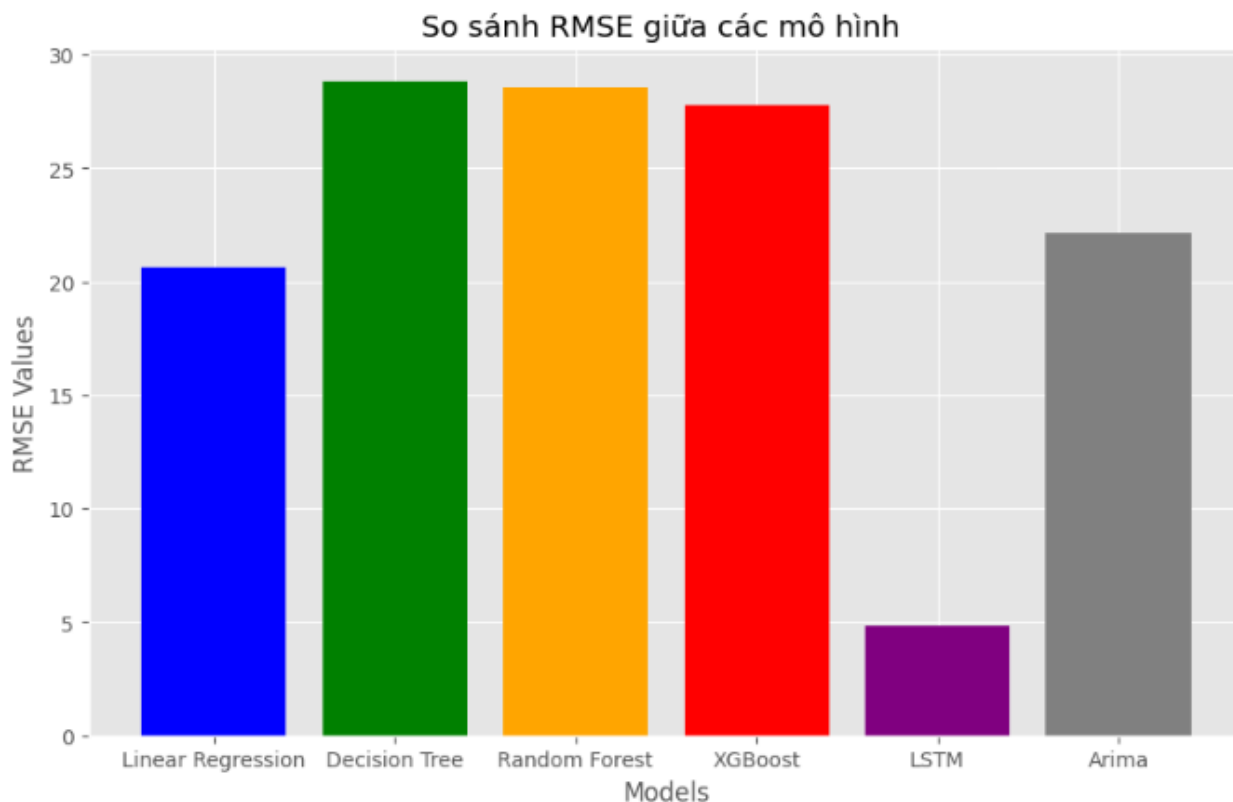
Hình 33. Mô hình ARIMA

Trong đoạn mã trên, chúng ta đã sử dụng mô hình ARIMA để dự đoán nhiệt độ trung bình. Trước hết, dữ liệu được chuẩn bị bằng cách chọn các biến độc lập và biến mục tiêu từ DataFrame `preprocessed_df`. Sau đó, dữ liệu được chia thành tập huấn luyện và tập kiểm tra với tỉ lệ 80-20%.

Mô hình ARIMA được xây dựng và huấn luyện trên tập huấn luyện, sử dụng `order=(0,1,1)` để áp dụng mô hình autoregressive (AR) với độ trễ 1 và moving average (MA) với độ trễ 1. Sau quá trình huấn luyện, mô hình được sử dụng để dự đoán giá trị trên tập kiểm tra.

Để đánh giá hiệu suất của mô hình, chúng ta sử dụng hai độ đo chính là Mean Absolute Error (MAE) và Root Mean Squared Error (RMSE). MAE đo lường trung bình của độ lệch tuyệt đối giữa giá trị dự đoán và giá trị thực tế. Trong khi đó, RMSE tính căn bậc hai của trung bình của bình phương của các độ lệch giữa giá trị dự đoán và giá trị thực tế.

Các giá trị dự đoán từ mô hình ARIMA được so sánh với giá trị thực tế để đánh giá chất lượng dự đoán của mô hình trong việc ước lượng nhiệt độ trung bình.



Hình 34. So sánh RMSE các mô hình

- Linear Regression: RMSE: 20.63
- Decision Tree: RMSE: 28.85
- Random Forest: RMSE: 28.55
- XGBoost: RMSE: 27.80
- LSTM (Long Short-Term Memory): RMSE: 5.03
- ARIMA (AutoRegressive Integrated Moving Average): RMSE: 22.14

Kết quả trên cho thấy mô hình LSTM đạt được RMSE thấp nhất, chỉ 5.03, đồng thời làm cho nó trở thành mô hình có khả năng dự đoán tốt nhất trong số các mô hình được đánh giá. Trong khi đó, Decision Tree và Random Forest có RMSE cao hơn, và Linear Regression cũng cho thấy độ chính xác không cao bằng so với các mô hình dự đoán phức tạp hơn như XGBoost và LSTM. ARIMA có kết quả khá ổn định, nhưng cũng không thể vượt qua hiệu suất của LSTM trong trường hợp này. Qua đó chúng ta sử dụng LSTM để dự đoán kết quả tương lai.

```

import datetime
data = bj.iloc[-time_steps:].to_numpy() #2D Array
data = scaler.transform(data)

data = np.expand_dims(data,0)

date = bj.index[-1]

date_store = bj.iloc[-time_steps:].index.to_list()

forecasts=10
for i in range(forecasts):
    predicted = model.predict(data[:,-20,:])
    date = date+datetime.timedelta(days=1)
    data = np.append(data,[predicted],axis=1)
    date_store.append(date)

data = scaler.inverse_transform(data.reshape(1,-1))
forecast_df = pd.DataFrame(index=date_store[time_steps-1:],data={'AvgTemperature':data.ravel()[time_steps-1:]})

```

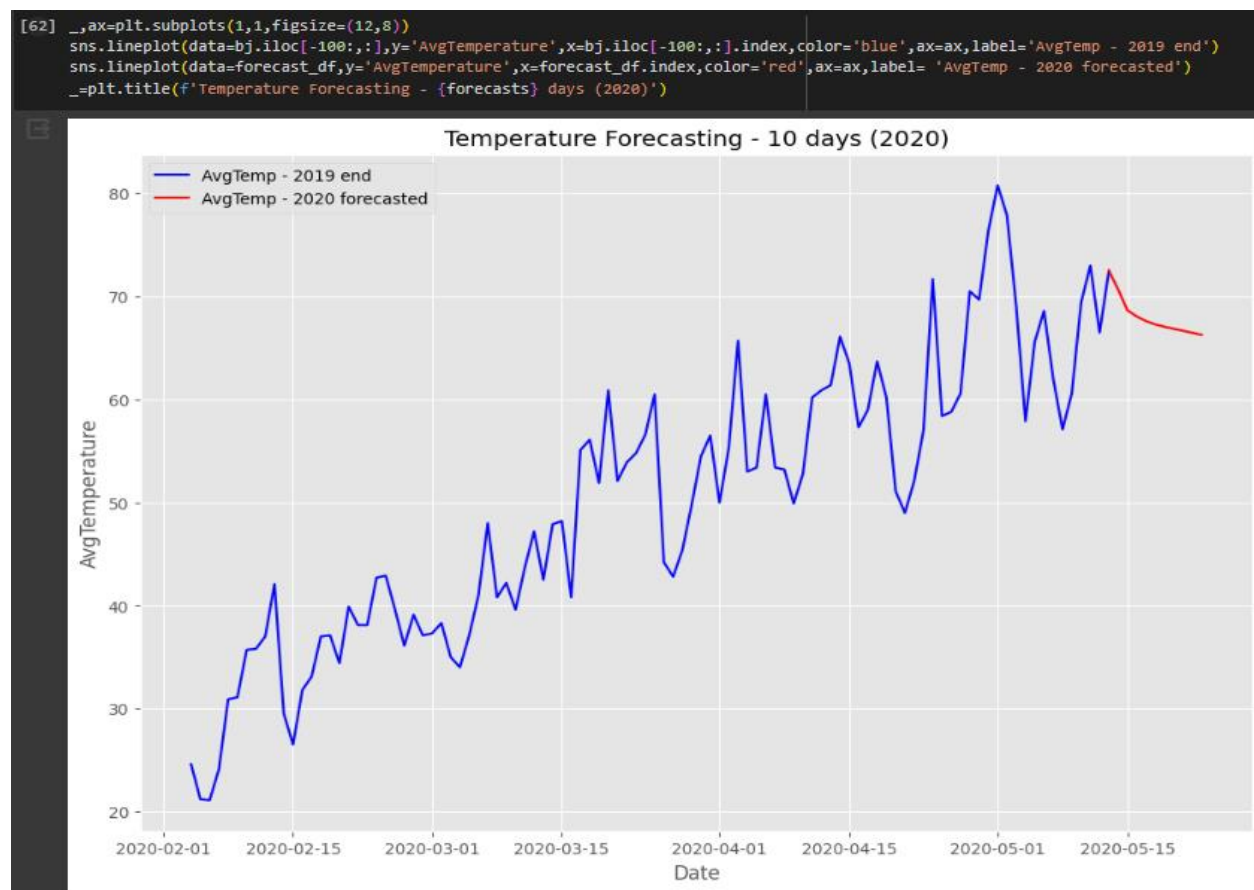
Hình 35. Dự báo 10 ngày tiếp theo bằng LSTM

Đầu tiên, một tập dữ liệu con được lấy từ dataframe bj, bao gồm các điểm dữ liệu cuối cùng với số lượng điểm dữ liệu xác định bởi biến time_steps. Dữ liệu này sau đó được chuẩn hóa sử dụng trình chuẩn hóa đã được sử dụng trong quá trình huấn luyện mô hình.

Một mảng numpy mới được tạo ra bằng cách thêm một chiều mới, phù hợp với đầu vào của mô hình LSTM. Sau đó, một vòng lặp được sử dụng để dự đoán nhiệt độ cho mỗi ngày tiếp theo. Mỗi lần lặp, mô hình LSTM được sử dụng để dự đoán nhiệt độ dựa trên dữ liệu hiện tại. Kết quả dự đoán được thêm vào mảng dữ liệu và ngày tương ứng cũng được cập nhật.

Cuối cùng, mảng dữ liệu sau khi đã được dự đoán được chuyển ngược chuẩn hóa để có giá trị thực tế. Một DataFrame mới được tạo với các ngày dự đoán và giá trị nhiệt độ dự đoán tương ứng.

Date	AvgTemperature
2020-05-13	72.500000
2020-05-14	72.826185
2020-05-15	72.452420
2020-05-16	72.877036
2020-05-17	73.424996
2020-05-18	73.994796
2020-05-19	74.584459
2020-05-20	75.182267
2020-05-21	75.761549
2020-05-22	76.321071
2020-05-23	76.882047



Hình 36. Kết quả dự đoán

Các giá trị trong cột "AvgTemperature" của DataFrame hiển thị nhiệt độ trung bình dự đoán cho các ngày từ 2020-05-13 đến 2020-05-23. Các giá trị này được tính toán thông qua quá trình dự đoán của mô hình LSTM sử dụng dữ liệu đầu vào gần đây và được hiển thị dưới dạng DataFrame để thể hiện sự biến động của nhiệt độ trung bình qua các ngày tiếp theo.

Tóm lại, dãy số dự đoán này cung cấp một cái nhìn tích cực về tính ổn định và khả năng dự đoán của mô hình LSTM, đặc biệt khi áp dụng thông tin từ 10 bước thời gian trước đó.

CHƯƠNG V: KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Trong quá trình nghiên cứu này, em đã tiến hành một quá trình tỉ mỉ và kỹ lưỡng từ việc thu thập dữ liệu đến tiền xử lý, phân tích trực quan, và xây dựng cũng như đánh giá các mô hình học máy. Việc thu thập dữ liệu được thực hiện thông qua lựa chọn và tập hợp các loại dữ liệu tài chính từ nguồn <https://www.kaggle.com/datasets/sudalairajkumar/daily-temperature-of-major-cities/data>, đảm bảo tính đa dạng và độ chính xác cao của dữ liệu. Sau đó, em đã tiến hành tiền xử lý dữ liệu, bao gồm các bước như làm sạch, chuẩn hóa và chuyển đổi dữ liệu thành định dạng phù hợp cho việc phân tích. Quá trình này đòi hỏi sự tỉ mỉ và kỹ lưỡng để đảm bảo dữ liệu sau khi xử lý là chính xác và đáp ứng yêu cầu của mô hình học máy.

Tiếp theo, em đã thực hiện phân tích trực quan dữ liệu để hiểu rõ hơn về các xu hướng và mẫu biến động. Các công đoạn này giúp em xác định những thông số quan trọng cho mô hình LSTM, bao gồm số lượng tầng ẩn, số nơ-ron mỗi tầng và số bước thời gian.

2. Các mục tiêu đưa ra đã hoàn thành

Mục tiêu chính của nghiên cứu là cải thiện độ chính xác và độ tin cậy trong dự đoán chuỗi thời gian tài chính, và em rất hài lòng với sự đạt được của mình trong việc đối mặt với thách thức này. Qua quá trình nghiên cứu, em đã thực hiện một loạt các bước quan trọng, bao gồm tích hợp dữ liệu từ nhiều bước thời gian khác nhau và tinh chỉnh kỹ thuật của mô hình học máy để phản ánh chính xác nhất các biến động trong dữ liệu dự báo thời gian.

Mục tiêu đã thành công trong việc xây dựng một mô hình có khả năng dự đoán xu hướng thị trường với độ chính xác và độ tin cậy đáng kể. Việc này không chỉ giúp nâng cao sự hiểu biết về tình hình thị trường mà còn tạo ra cơ hội mới trong việc áp dụng các phương pháp học máy hiện đại vào lĩnh vực tài chính. Bổ sung trực quan hoá dữ liệu và kết quả dự đoán đã thêm sức mạnh và minh họa rõ ràng, cung cấp cái nhìn chi tiết và trực quan hoá bằng các biểu đồ.

3. Kiến nghị

Dựa trên kết quả tích cực của nghiên cứu, việc áp dụng mô hình LSTM trong dự báo nhiệt độ với chuỗi thời gian dài hơn là một tiến triển quan trọng. Sự hiệu quả của mô hình đã được chứng minh, đặc biệt là khi sử dụng thông tin từ nhiều bước thời gian trước đó. Điều này mang lại tiềm năng cải thiện độ chính xác và độ tin cậy trong dự đoán nhiệt độ, giúp cung cấp thông tin chính xác và hữu ích cho các quyết định liên quan đến thời tiết và khí hậu.

Kiến nghị sử dụng mô hình LSTM không chỉ hỗ trợ trong việc đưa ra dự đoán chính xác về nhiệt độ mà còn mang lại những lợi ích trong việc quản lý và ứng phó với các biến động thời tiết. Điều này có thể hỗ trợ ngành công nghiệp dự báo thời tiết và các lĩnh vực ứng dụng khác, như nông nghiệp, đối phó với thách thức của biến đổi khí hậu.

Bổ sung mô hình LSTM vào công cụ dự báo nhiệt độ không chỉ tăng cường sự hiểu biết về biến động thời tiết mà còn tạo ra cơ hội mới trong việc áp dụng công nghệ tiên tiến để cải thiện khả năng dự báo và đáp ứng mọi ngóc ngách của thị trường nhiệt độ.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Types of RNN (Recurrent Neural Network) (opengenius.org).
- [2] 9.2. Bộ nhớ Ngắn hạn Dài (LSTM) — Đắm mình vào Học Sâu 0.14.4 documentation (aivivn.com).
- [3] Box G. E. P., Jenkins G. M., Reinsel G. C., & Ljung G. M., "Time Series Analysis: Forecasting and Control" John Wiley & Sons, 2015.
- [4] Introduction to Time Series and Forecasting, Peter J. Brockwell & Richard A. Davis.
- [5] Box G.E.P.; Jenkins G.M., "Time Series Analysis: Forecasting and Control", Holden-Day: San Francisco, CA, USA, 1976.
- [6] Hochreiter S. & Schmidhuber, J., "Long short-term memory", Neural computation, 9(8), 1735-1780, 1997.
- [7] Siامي-Namini S., Tavakoli N. & Namin A.S., "A comparison of ARIMA and LSTM in forecasting time series", In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, pp. 1394–1401, 2018.