

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN: PHÂN TÍCH DỮ LIỆU LỚN**

**ĐỀ TÀI**

**Phân tích dự đoán lượng mưa bằng mô hình hồi quy tuyến tính trên nền tảng Apache Spark**

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn:** | **Ts. Nguyễn Mạnh Cường** |
| **Nhóm – Lớp:** | **3 – 2024IT6077002** |
| **Thành viên:** | **Lương Thu Hằng - 2022602276** |
|  | **Trần Thị Phương Thảo - 2022601235** |
|  | **Nguyễn Quốc Việt - 2022603172** |

*Hà Nội, 2025*

1

# **LỜI CẢM ƠN**

Đầu tiên, chúng em xin bày tỏ lòng biết ơn sâu sắc đến quý thầy cô, nhà trường và bạn bè vì sự quan tâm, hướng dẫn tận tình và những lời động viên quý báu mà mọi người đã dành cho chúng em trong suốt quá trình nghiên cứu và thực hiện đề tài. Sự hỗ trợ quý giá ấy đã tiếp thêm sức mạnh cho chúng em, đồng thời đóng góp một phần không nhỏ trong quá trình hoàn thiện đề tài.

Đặc biệt, chúng em xin chân thành cảm ơn giảng viên hướng dẫn – TS. Nguyễn Mạnh Cường đã tận tình giúp đỡ, hỗ trợ chúng em trong quá trình thực hiện đề tài. Cung cấp cho chúng em những kiến thức quý báu cũng như những lời khuyên hữu ích. Tạo động lực cho chúng em hoàn thành tốt nhiệm vụ của mình. Bên cạnh đó, chúng em cũng xin cảm ơn các bạn học viên trong Khoa Công nghệ thông tin đã đóng góp ý kiến giúp chúng em thực hiện đề tài đạt hiệu quả hơn.

Bài tiểu luận này đã giúp chúng em rèn luyện tư duy phân tích và kỹ năng xử lý dữ liệu thông qua việc áp dụng các phương pháp phân tích thống kê và mô hình hóa trong Apache Spark. Thông qua quá trình thực hiện đề tài dự đoán lượng mưa bằng mô hình hồi quy tuyến tính, chúng em đã hiểu sâu hơn về cách thu thập, tiền xử lý, phân tích và trực quan hóa dữ liệu khí tượng, cũng như cách đánh giá độ chính xác của mô hình dự báo.

Chúng em tin rằng những kiến thức và kinh nghiệm thu nhận được từ đề tài không chỉ giúp củng cố nền tảng chuyên môn về phân tích dữ liệu lớn mà còn sẽ hỗ trợ chúng em trong các dự án nghiên cứu sau này, cũng như trong công việc và cuộc sống thực tiễn, nơi dữ liệu ngày càng giữ vai trò quan trọng!

Nhóm chúng em xin trân trọng cảm ơn!

**Nhóm 10**

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc217040684)

[DANH MỤC HÌNH ẢNH 4](#_Toc217040685)

[DANH MỤC BẢNG BIỂU 7](#_Toc217040686)

[DANH MỤC CÁC KÝ HIỆU, TỪ VIẾT TẮT 8](#_Toc217040687)

[LỜI NÓI ĐẦU 9](#_Toc217040688)

[**CHƯƠNG 1. TỔNG QUAN VÀ PHÁT BIỂU BÀI TOÁN** 11](#_Toc217040689)

[1.1. Tổng quan về phân tích dữ liệu 11](#_Toc217040690)

[1.1.1. Khái niệm phân tích dữ liệu 11](#_Toc217040691)

[1.2. Tổng quan về bài toán dự báo 13](#_Toc217040692)

[1.2.1. Lịch sử về bài toán dự báo 13](#_Toc217040693)

[1.2.2. Tình hình nghiên cứu trong nước 14](#_Toc217040694)

[1.2.3. Tình hình nghiên cứu ở nước ngoài 15](#_Toc217040695)

[1.3. Phát biểu bài toán 16](#_Toc217040696)

[1.3.1. Xác định đầu vào, đầu ra của bài toán 16](#_Toc217040697)

[1.3.2. Mục tiêu nghiên cứu 17](#_Toc217040698)

[1.3.3. Ý nghĩa khoa học và thực tiễn 17](#_Toc217040699)

[1.3.4. Cơ hội và khó khăn dự tính 18](#_Toc217040700)

[CHƯƠNG 2. CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN 19](#_Toc217040701)

[2.1. Phương pháp phân tích mô tả 19](#_Toc217040702)

[2.1.1. Phân tích mô tả 19](#_Toc217040703)

[2.1.2. Phương pháp phân tích trên từng biến 20](#_Toc217040704)

[2.1.3. Phương pháp phân tích trên nhiều biến 22](#_Toc217040705)

[2.2. Phương pháp phân tích hồi quy 23](#_Toc217040706)

[2.2.1. Tổng quan về phân tích hồi quy 23](#_Toc217040707)

[2.2.2. Một số phương pháp phân tích hồi quy 23](#_Toc217040708)

[2.2.3. Phương pháp Lasso Regression 24](#_Toc217040709)

[2.2.4. Phương pháp Ridge Regression 32](#_Toc217040710)

[2.2.5. Phương pháp Linear Regression 36](#_Toc217040711)

[2.2.6. Lựa chọn phương pháp 44](#_Toc217040712)

[2.3. Công cụ phục vụ thực hiện bài toán 45](#_Toc217040713)

[2.3.1. Python 45](#_Toc217040714)

[2.3.2. Apache Spark 46](#_Toc217040715)

[2.3.3. Lựa chọn công cụ 47](#_Toc217040716)

[2.3.4. Lựa chọn mô hình 49](#_Toc217040717)

[CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ 50](#_Toc217040718)

[3.1. Dữ liệu thực nghiệm 50](#_Toc217040719)

[3.2. Quy trình thực nghiệm 51](#_Toc217040720)

[3.2.1. Đặt mục tiêu 51](#_Toc217040721)

[3.2.2. Tiền xử lý dữ liệu 52](#_Toc217040722)

[3.2.3. Phân tích mô tả 57](#_Toc217040723)

[3.2.4. Phân tích mô tả bằng Spark SQL 72](#_Toc217040724)

[3.2.5. Phân tích hồi quy 80](#_Toc217040725)

[3.3. Đánh giá và đề xuất 90](#_Toc217040726)

[3.3.1. Đánh giá và kết quả thực nghiệm 90](#_Toc217040727)

[3.3.2. Nguyên nhân và đề xuất cải thiện 94](#_Toc217040728)

[3.4. Kết luận 95](#_Toc217040729)

[CHƯƠNG 4. CÀI ĐẶT VÀ TRIỂN KHAI 96](#_Toc217040730)

[4.1. Cài đặt công cụ 96](#_Toc217040731)

[4.1.1. Phần mềm Pycharm chạy Python 96](#_Toc217040732)

[4.1.2. Tkinter 99](#_Toc217040733)

[4.2.3. Cài đặt Spark 102](#_Toc217040734)

[4.2. Giao diện 105](#_Toc217040735)

[4.3. Kết luận 108](#_Toc217040736)

[KẾT LUẬN 109](#_Toc217040737)

[TÀI LIỆU THAM KHẢO 110](#_Toc217040738)

# **DANH MỤC HÌNH ẢNH**

[**Hình 1.1: Quy trình phân tích dữ liệu** **12**](#_w608umd0t71s)

[**Hình 2.1: Biểu đồ phần tích mô tả** **20**](#_yahb4tl6lusc)

[**Hình 2.1: Biểu đồ Histogram giúp xác định giá trị ngoại lai (Outliers)** **21**](#_monpjxfygwdr)

[**Hình 2.2: Biểu đồ Scatter thể hiện mối quan hệ giữa độ tuổi và giá bán** **22**](#_vfgvw58iqspt)

[**Hình 2.3: Hệ số Lasso là hàm của λ** **26**](#_yy0ddwxob1bg)

[**Hình 2.4: Sự thay đổi của độ lớn các hệ số ước lượng (coefficient of features)**](#_pp2autdlf9qs)[**theo hệ số điều chuẩn α. Khi tăng dần độ lớn của α thì độ lớn của hệ** **32**](#_pp2autdlf9qs)

[**Hình 2.5: Mối quan hệ tuyến tính giữa biến đầu ra (y) và biến dự đoán** **34**](#_oxru1f4tip1m)

[**Hình 2.6: Tính toán hồi quy đơn biến** **35**](#_snokq2yxdtk3)

[**Hình 2.7: Minh họa siêu phẳng phù hợp nhất của mô hình tuyến tính bội 35**](#_flqwgc64w35s)[**Hình 2.8: Ý nghĩa của bình phương R** **36**](#_w94m5186bjd5)

[**Hình 2.9: Ngôn ngữ lập trình Python** **40**](#_dyfl7f7qis6s)

[**Hình 2.10: Ngôn ngữ lập trình C++** **41**](#_188i1cj7eiu9)

[**Hình 3.1: 15 hàng dữ liệu đầu tiên trong bộ dữ liệu** **44**](#_5fblmpmovidp)

[**Hình 3.2: Quy trình thực nghiệm** **45**](#_fn4f77uplrol)

[**Hình 3.3: Thông tin dữ liệu dạng phi số** **48**](#_vhwjwwmovn6)

[**Hình 3.4: Thống kê dữ liệu khuyết trước tiền xử lý dữ liệu** **49**](#_fhx0qtn3yh68)

[**Hình 3.5: Biểu đồ cột thể hiện giá trị khuyết trong tập dữ liệu 49**](#_a7318h6k8ss4)

[**Hình 3.6: Kết quả sau quá trình điền khuyết dữ liệu** **51**](#_uau61b2sh4af)

[**Hình 3.7: Các loại nhãn trong mỗi cột** **52**](#_qyohlv1lnxb2)

[**Hình 3.8: Dữ liệu sau khi ánh xạ các giá trị phân loại thành kiểu số** **53**](#_dbudu8ecv3ko)

[**Hình 3.9: Dữ liệu sau khi tiền xử lý** **54**](#_xairitozpepy)

[**Hình 3.10: Biểu đồ Boxplot của các cột định lượng** **56**](#_xpty3fi8g1wv)

[**Hình 3.11: Phân phối dữ liệu của province** **56**](#_3k0ncf9u4yim)

[**Hình 3.12: Phân phối dữ liệu của wind\_d** **57**](#_cqcm7omqm5hx)

[**Hình 3.13: Phân bổ các lượng mưa của các tỉnh** **58**](#_z5vcnvu9jf06)

[**Hình 3.14: Biểu đồ về sự thay đổi lượng mưa theo từng năm** **59**](#_1zui2vexm2rt)

[**Hình 3.15: Biểu đồ phân bố lượng mưa theo mức độ mây che phủ** **60**](#_t4up71wb4nqw)

[**Hình 3.16: Biểu đồ phân bố lượng mưa trung bình theo từng năm đối với mỗi**](#_32tn2oryv6vp)[**tỉnh** **62**](#_32tn2oryv6vp)

[**Hình 3.17: Biểu đồ tương quan giữa các thuộc tính 63**](#_g7gf7m1q6rho)

[**Hình 3.18: Biểu đồ scatter tổng hợp các mối quan hệ giữa ‘rain’ và các biến**](#_xs3nhug41784)

[**. 65**](#_xs3nhug41784)

[**Hình 4. 1 Cài đặt Visual Studio 70**](#_jtqxi8wexzbv)

[**Hình 4.2: Cửa sổ đặt tên thư mục 71**](#_uiv7rog2sz3h)

[**Hình 4.3: Xây dựng giao diện bằng Tkinter 74**](#_lqj7hc44wjaz)

[**Hình 4.4: Giao diện ứng dụng 74**](#_2opqv525tw0l)

[**Hình 4.5: Giao diện kết quả 75**](#_5awip9650cy8)

# **DANH MỤC BẢNG BIỂU**

[**Bảng 1: So sánh ngôn ngữ Python và C++** **43**](#_3n9onqach8t4)

[**Bảng 2: Mô tả thông tin các cột dữ liệu trong dataset** **45**](#_c2bbsp1vqtay)

[**Bảng 3: Bảng thống kê cho các cột dữ liệu định lượng** **55**](#_5abk40vr95n8)

[**Bảng 4: Tập dữ liệu đầu vào** **64**](#_rotwqjdv0f5k)

# **DANH MỤC CÁC KÝ HIỆU, TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| RSE | Residual Standard Error: Sai số chuẩn của phần dư, đo lường mức độ khớp của mô hình. |
| SSR | *Sum of Squared Residuals*: Tổng bình phương sai  số dư. |
| RMSE | *Root Mean Squared Error*: Căn bậc hai của sai số trung bình bình phương, đo lường độ chính xác dự  đoán của mô hình. |
| R² | *R-Squared*: Hệ số xác định, đo lường phần trăm  biến thiên của biến phụ thuộc được giải thích bởi mô hình. |

# **LỜI NÓI ĐẦU**

Trong bối cảnh biến đổi khí hậu và thời tiết ngày càng diễn biến phức tạp, việc dự đoán lượng mưa đóng vai trò quan trọng trong nhiều lĩnh vực như nông nghiệp, quản lý tài nguyên nước, quy hoạch đô thị và phòng chống thiên tai. Một dự báo chính xác về lượng mưa không chỉ giúp giảm thiểu thiệt hại kinh tế và xã hội, mà còn hỗ trợ các nhà hoạch định chính sách và người dân chủ động ứng phó với các tình huống thời tiết bất thường.

Tuy nhiên, lượng mưa là một hiện tượng khí tượng chịu ảnh hưởng của nhiều yếu tố tự nhiên phức tạp như độ ẩm, nhiệt độ, áp suất khí quyển, tốc độ gió,... Việc phân tích và xây dựng mô hình dự đoán lượng mưa đòi hỏi phải áp dụng các kỹ thuật phân tích dữ liệu và mô hình hóa toán học phù hợp. Trong khuôn khổ báo cáo này, chúng em lựa chọn phương pháp **hồi quy tuyến tính** nhằm phân tích và dự đoán lượng mưa dựa trên các yếu tố đầu vào quan sát được.

Báo cáo được tổ chức thành bốn chương chính:

**Chương 1. Tổng quan và phát biểu bài toán:**

Chương này trình bày bối cảnh tổng quan về dự đoán lượng mưa và tầm quan trọng của bài toán. Chúng em phát biểu rõ ràng bài toán cần giải quyết, lý do lựa chọn chủ đề, đặt ra các mục tiêu nghiên cứu cụ thể, đồng thời định hình những câu hỏi nghiên cứu làm nền tảng cho toàn bộ quá trình phân tích.

**Chương 2. Các kỹ thuật giải quyết bài toán:**

Chương này tổng hợp, phân tích các phương pháp dự đoán phổ biến hiện nay, chỉ ra ưu nhược điểm của từng phương pháp. Trong đó, chúng em tập trung trình bày chi tiết về mô hình hồi quy tuyến tính – từ việc xử lý dữ liệu, xây dựng mô hình cho đến các kỹ thuật đánh giá chất lượng dự báo.

**Chương 3. Thực nghiệm và đánh giá:**

Chương này trình bày quá trình áp dụng mô hình hồi quy tuyến tính trên tập dữ liệu thực tế về lượng mưa. Chúng em tiến hành thực nghiệm, đánh giá hiệu quả mô hình thông qua các chỉ số đo lường, đồng thời phân tích các kết quả đạt được để rút ra nhận xét và bài học kinh nghiệm.

**Chương 4. Cài đặt và triển khai:**

Chương cuối cùng mô tả chi tiết quá trình cài đặt mô hình bằng công cụ lập trình, từ tiền xử lý dữ liệu, huấn luyện mô hình đến triển khai thử nghiệm dự báo. Đây cũng là bước chuẩn bị cho khả năng ứng dụng mô hình vào thực tiễn trong các bài toán dự báo lượng mưa thực tế.Thông qua việc thực hiện đề tài này, chúng em mong muốn không chỉ nâng cao kỹ năng phân tích dữ liệu và xây dựng mô hình dự báo, mà còn đóng góp một phần nhỏ bé vào việc phát triển các công cụ hỗ trợ dự báo khí tượng, góp phần giảm thiểu rủi ro thiên tai và bảo vệ đời sống cộng đồng.

Chúng em hy vọng rằng những kết quả và kinh nghiệm rút ra từ báo cáo sẽ tạo nền tảng cho các nghiên cứu sâu hơn trong tương lai.

# **CHƯƠNG 1. TỔNG QUAN VÀ PHÁT BIỂU BÀI TOÁN**

## Tổng quan về phân tích dữ liệu

### Khái niệm phân tích dữ liệu

Phân tích dữ liệu (Data Analysis) là quá trình kiểm tra, làm sạch, biến đổi và mô hình hóa dữ liệu nhằm mục đích khám phá thông tin hữu ích, rút ra kết luận và hỗ trợ quá trình ra quyết định. Đây là một bước không thể thiếu trong hầu hết các lĩnh vực nghiên cứu và kinh doanh hiện đại, nơi dữ liệu được coi là nguồn tài nguyên quý giá.

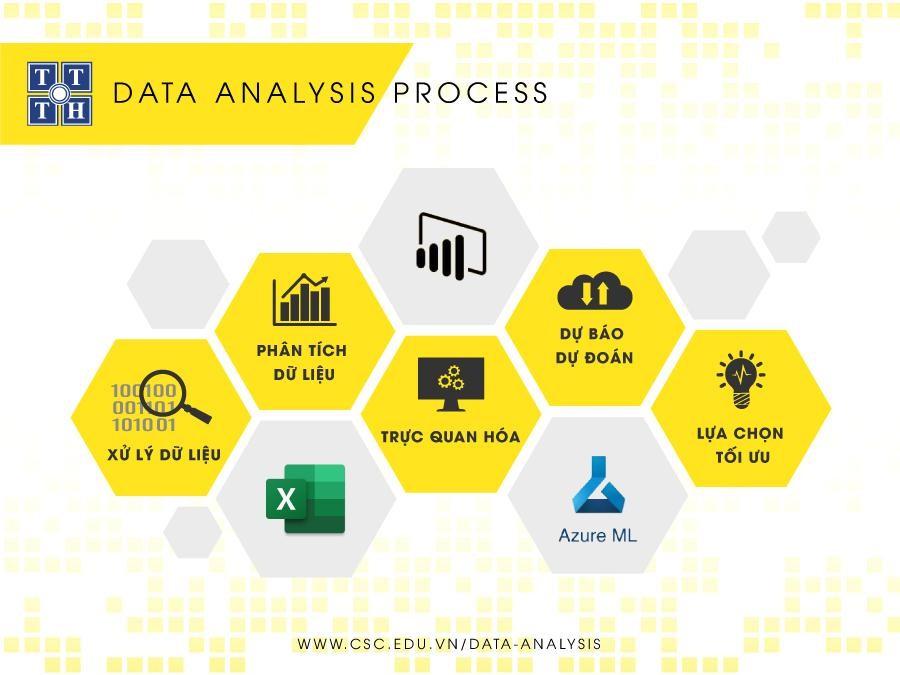
Thông qua phân tích dữ liệu, các mẫu, xu hướng và mối quan hệ ẩn trong dữ liệu có thể được phát hiện, từ đó cung cấp những hiểu biết sâu sắc và hỗ trợ cho việc lập kế hoạch chiến lược, tối ưu hóa quy trình và dự báo tương lai.

Phân tích dữ liệu không chỉ dừng lại ở việc xử lý dữ liệu thô mà còn bao gồm việc ứng dụng các kỹ thuật thống kê, học máy (machine learning), khai phá dữ liệu (data mining) và trực quan hóa dữ liệu (data visualization) nhằm nâng cao khả năng hiểu và khai thác giá trị tiềm ẩn trong dữ liệu.

Quá trình phân tích dữ liệu thường trải qua nhiều bước: thu thập dữ liệu, tiền xử lý dữ liệu, phân tích mô tả, phân tích khám phá, phân tích dự đoán và phân tích suy diễn. Mỗi bước đóng vai trò quan trọng trong việc đảm bảo chất lượng dữ liệu và độ tin cậy của các kết luận rút ra.

Trong kỷ nguyên số hiện nay, với sự phát triển mạnh mẽ của công nghệ và lượng dữ liệu ngày càng gia tăng, phân tích dữ liệu đã trở thành một công cụ không thể thiếu, giúp các tổ chức, doanh nghiệp và cá nhân nâng cao năng lực cạnh tranh và đưa ra các quyết định sáng suốt hơn.

**Quy trình phân tích dữ liệu**



*Hình 1.1: Quy trình phân tích dữ liệu*

**Quy trình phân tích dữ liệu thường bao gồm các bước chính sau:**

* + - * **Xác định mục tiêu và thu thập dữ liệu:**
        + **Xác định mục tiêu:** Đây là những kết quả cụ thể mà quá trình xử lý và phân tích dữ liệu hướng đến. Việc xác định rõ mục tiêu sẽ định hình hướng đi và phạm vi của toàn bộ quá trình phân tích, giúp tập trung vào việc thu thập các thông tin quan trọng nhằm đáp ứng các yêu cầu hoặc nhu cầu cụ thể.
        + **Thu thập dữ liệu**: Dữ liệu được thu thập từ nhiều nguồn khác nhau như cơ sở dữ liệu, tệp tin, trang web, thiết bị cảm biến, v.v. Dữ liệu có thể ở nhiều dạng như số liệu, văn bản, hình ảnh hoặc âm thanh.

Tiền xử lý dữ liệu:

Dữ liệu thu thập ban đầu thường chứa nhiều thiếu sót như nhiễu, thiếu dữ liệu, hoặc sai lệch. Tiền xử lý dữ liệu bao gồm các công đoạn như tóm lược,

làm sạch, tích hợp, chuyển đổi, rút gọn và rời rạc hóa dữ liệu nhằm chuẩn hóa và nâng cao chất lượng dữ liệu trước khi phân tích.

Phân tích dữ liệu:

Đây là bước trọng tâm của quy trình, sử dụng các kiến thức và kỹ thuật phân tích để khám phá những mối quan hệ, khuôn mẫu và thông tin ẩn trong dữ liệu. Các phương pháp có thể được áp dụng gồm phân tích mô tả, phân tích hồi quy, phân tích sự khác biệt, thống kê, học máy (machine learning), khai phá dữ liệu (data mining) và các kỹ thuật chuyên sâu khác.

Kết luận và dự đoán:

Dựa trên kết quả phân tích, ta có thể rút ra các kết luận có giá trị thực tiễn, đưa ra hiểu biết sâu sắc hơn về dữ liệu, đồng thời tiến hành dự đoán xu hướng hoặc kết quả trong tương lai.

## Tổng quan về bài toán dự báo

### Lịch sử về bài toán dự báo

* + - * **Giai đoạn khởi đầu (1940 – 1950)**

Phân tích dự đoán bắt nguồn từ những nghiên cứu ban đầu về phân tích thống kê. Các nhà toán học và thống kê đã phát triển những mô hình dựa trên dữ liệu lịch sử để đưa ra dự đoán. Thời kỳ này chủ yếu tập trung vào hồi quy tuyến tính và các phương pháp thống kê cơ bản.

* + - * **Sự xuất hiện của máy tính lớn (1960 – 1970)**

Khi máy tính lớn ra đời, khả năng tính toán được nâng cao đáng kể, cho phép thực hiện các phép phân tích phức tạp hơn. Các doanh nghiệp và tổ chức nghiên cứu bắt đầu tận dụng máy tính để xử lý những tập dữ liệu lớn, ứng dụng phân tích dự đoán vào các lĩnh vực như tài chính và tiếp thị.

* + - * **Hệ thống hỗ trợ quyết định (1980 – 1990)**

Trong giai đoạn này, các hệ thống hỗ trợ quyết định (DSS) trở nên phổ biến, kết hợp các công cụ phân tích và mô hình hóa nhằm hỗ trợ quá trình ra quyết định. Phân tích dự đoán dần được ứng dụng rộng rãi trong môi trường kinh doanh để dự báo xu hướng và đánh giá rủi ro.

* + - * **Khai thác dữ liệu (1990 – 2000)**

Cuối thế kỷ 20, sự bùng nổ của khai thác dữ liệu đánh dấu bước ngoặt quan trọng cho phân tích dự đoán. Các kỹ thuật như mạng nơ-ron nhân tạo và các thuật toán học máy cho phép khai thác và phân tích những tập dữ liệu lớn, phức tạp. Phân tích dự đoán được mở rộng mạnh mẽ trong các lĩnh vực như tài chính, chăm sóc sức khỏe và bán lẻ.

* + - * **Học máy và Trí tuệ nhân tạo (2010 – nay)**

Trong những năm gần đây, học máy (Machine Learning) và trí tuệ nhân tạo (AI) đã trở thành nền tảng cốt lõi cho phân tích dự đoán. Các công nghệ này giúp thực hiện những phân tích sâu hơn, nhận diện mẫu và xây dựng mô hình dự báo chính xác. Đặc biệt, học sâu (Deep Learning) – một nhánh của học máy – đã đạt nhiều thành tựu trong nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên và nhiều ứng dụng khác.

* + - * **Phân tích dự đoán theo thời gian thực (Hiện tại – Tương lai)**

Xu hướng hiện nay đang hướng tới phân tích dự đoán theo thời gian thực, nơi các tổ chức có thể đưa ra dự đoán và quyết định ngay lập tức. Điều này đặc biệt cần thiết trong các lĩnh vực như tài chính, an ninh mạng và Internet vạn vật (IoT), nơi việc phản ứng nhanh với các biến động là yếu tố sống còn.

### Tình hình nghiên cứu trong nước

Bài toán dự báo có vai trò rất quan trọng tại Việt Nam, góp phần cải thiện công tác quản lý, định hình chiến lược phát triển và tối ưu hóa việc sử dụng tài nguyên trong nhiều lĩnh vực khác nhau. Một số điểm nổi bật về tình hình ứng dụng phân tích dữ liệu và dự báo tại Việt Nam có thể kể đến như sau:

Trong nhiều lĩnh vực, việc áp dụng bài toán dự báo tại Việt Nam hiện vẫn đang ở giai đoạn đầu. Các phương pháp phân tích dữ liệu hiện đại và kỹ thuật dự báo tiên tiến đang được các tổ chức, doanh nghiệp tìm hiểu, thí điểm và từng bước áp dụng vào thực tiễn.

* + - * **Ứng dụng thực tiễn trong nông nghiệp và kinh tế**

Dự báo đang đóng vai trò thiết yếu trong một số ngành trọng điểm như nông nghiệp và kinh tế. Cụ thể, trong nông nghiệp, các mô hình dự báo được sử

dụng để dự đoán điều kiện thời tiết, sản lượng mùa vụ và nhu cầu năng lượng. Trong lĩnh vực kinh tế, dự báo được ứng dụng nhằm dự đoán các chỉ số quan trọng như tăng trưởng GDP, lạm phát, và biến động tỷ giá, từ đó hỗ trợ hoạch định chính sách kinh tế vĩ mô.

* + - * **Thách thức về dữ liệu**

Một trong những rào cản lớn đối với việc triển khai hiệu quả các bài toán dự báo tại Việt Nam là vấn đề dữ liệu. Hiện nay, việc thu thập, lưu trữ và quản lý dữ liệu chất lượng cao vẫn còn nhiều hạn chế. Dữ liệu thường không đầy đủ, thiếu tính nhất quán và đôi khi chưa đảm bảo độ tin cậy cần thiết cho các mô hình dự báo hiện đại.

### Tình hình nghiên cứu ở nước ngoài

Trên thế giới, bài toán dự báo đã và đang nhận được sự quan tâm rộng rãi từ giới học thuật, doanh nghiệp, và tổ chức nghiên cứu. Trong suốt nhiều thập kỷ qua, các phương pháp dự báo liên tục được cải tiến và ứng dụng trong đa dạng các lĩnh vực, từ tài chính, kinh tế, y tế đến khí tượng, nông nghiệp và công nghệ.

* + - * **Sự phát triển về phương pháp**

Ở các quốc gia phát triển như Hoa Kỳ, Anh, Đức, Nhật Bản và các nước châu Âu, các kỹ thuật dự báo đã tiến hóa từ các mô hình thống kê truyền thống (như hồi quy tuyến tính, ARIMA) sang các phương pháp hiện đại dựa trên học máy (Machine Learning) và học sâu (Deep Learning). Các thuật toán như Random Forest, Gradient Boosting, mạng nơ-ron nhân tạo (ANN), mạng nơ-ron hồi tiếp (RNN) và mạng LSTM đã được nghiên cứu và áp dụng rộng rãi để cải thiện độ chính xác của dự báo.

* + - * **Ứng dụng đa dạng trong thực tiễn**

Bài toán dự báo được ứng dụng thành công trong nhiều lĩnh vực khác nhau:

* + - * + **Tài chính**: Dự báo giá cổ phiếu, biến động thị trường, rủi ro tín dụng.
        + **Y tế**: Dự đoán sự bùng phát dịch bệnh, nhu cầu chăm sóc sức khỏe.
        + **Khí tượng thủy văn**: Dự báo thời tiết, thiên tai, biến đổi khí hậu.
        + **Sản xuất và chuỗi cung ứng**: Dự báo nhu cầu sản phẩm, tối ưu hóa
        + **Kinh tế vĩ mô**: Dự báo tăng trưởng GDP, lạm phát, thất nghiệp.
      * **Xu hướng nghiên cứu hiện đại**

Trong những năm gần đây, xu hướng nghiên cứu tập trung mạnh vào:

* + - * + **Dự báo theo thời gian thực**: Tận dụng các hệ thống cảm biến IoT và dữ liệu lớn để cập nhật mô hình liên tục, cho phép dự báo và ra quyết định ngay lập tức.
        + **Dự báo với dữ liệu lớn và phi cấu trúc**: Sử dụng dữ liệu từ mạng xã hội, văn bản, hình ảnh, âm thanh để phục vụ các bài toán dự báo phức tạp.
        + **Ứng dụng trí tuệ nhân tạo giải thích được (Explainable AI - XAI)**: Phát triển các mô hình dự báo không chỉ chính xác mà còn dễ hiểu, nhằm tăng tính minh bạch và khả năng ứng dụng trong thực tế.
      * **Một số tổ chức và dự án nổi bật**
        + Các trường đại học hàng đầu như MIT, Stanford, Oxford, Cambridge liên tục có các công trình nghiên cứu về dự báo sử dụng AI và dữ liệu lớn.
        + Các công ty công nghệ lớn như Google, Microsoft, Amazon cũng đầu tư mạnh vào nghiên cứu và phát triển các hệ thống dự báo trong các lĩnh vực như thương mại điện tử, logistics và điện toán đám mây.

## Phát biểu bài toán

Bài toán "Phân tích dữ liệu và dự báo lượng mưa bằng phương pháp hồi quy tuyến tính" thuộc lĩnh vực khoa học dữ liệu khí tượng. Đề tài tập trung vào việc tìm hiểu và dự đoán lượng mưa tại một khu vực cụ thể trong tương lai dựa trên các yếu tố như nhiệt độ, độ ẩm, áp suất khí quyển, tốc độ gió và các thông số môi trường khác.

Mục tiêu của bài toán là xác định những yếu tố ảnh hưởng trực tiếp đến lượng mưa và xây dựng một mô hình hồi quy tuyến tính để dự báo lượng mưa một cách chính xác.

### Xác định đầu vào, đầu ra của bài toán

Đầu vào: Tập dữ liệu thời tiết được thu thập trong nhiều năm (ví dụ: từ năm 2000

đến 2020), mỗi bản ghi đại diện cho các thông số đo được trong ngày/tháng như: nhiệt độ trung bình, độ ẩm trung bình, áp suất khí quyển, tốc độ gió, hướng gió, mức độ mây che phủ,... Dữ liệu có thể có sai sót như giá trị thiếu, nhiễu hoặc ngoại lệ, đòi hỏi xử lý trước khi đưa vào mô hình.

Đầu ra: Lượng mưa dự đoán (mm/ngày hoặc mm/tháng), là biến mục tiêu mà mô hình sẽ huấn luyện để dự báo. Ngoài ra, cần tính toán các chỉ số đánh giá hiệu quả dự báo như RMSE (Root Mean Square Error), MAE (Mean Absolute Error) hoặc hệ số xác định R²

### Mục tiêu nghiên cứu

* + - * **Tiền xử lý dữ liệu:** Làm sạch dữ liệu để loại bỏ hoặc xử lý các giá trị thiếu, ngoại lệ, đồng thời chuẩn hóa hoặc biến đổi dữ liệu để phù hợp với yêu cầu của mô hình hồi quy tuyến tính.
      * **Phân tích yếu tố ảnh hưởng:** Xác định các yếu tố khí tượng quan trọng ảnh hưởng đến lượng mưa, chẳng hạn như nhiệt độ, độ ẩm, áp suất khí quyển, nhằm tối ưu hóa độ chính xác và hiệu quả của mô hình.
      * **Xây dựng mô hình hồi quy:** Áp dụng phương pháp hồi quy tuyến tính để thiết lập mối quan hệ giữa các biến khí tượng và lượng mưa. Mô hình sẽ tìm ra các trọng số thể hiện mức độ ảnh hưởng của từng yếu tố đến lượng mưa.
      * **Dự đoán lượng mưa:** Sử dụng mô hình đã xây dựng để dự đoán lượng mưa trong tương lai dựa trên các dữ liệu đầu vào mới.

### Ý nghĩa khoa học và thực tiễn

* + - * ***Khoa học dữ liệu***: Đề tài này đóng góp vào lĩnh vực Khoa học Dữ liệu bằng cách nghiên cứu áp dụng kỹ thuật phân tích dữ liệu và hồi quy tuyến tính vào lĩnh vực khí tượng, góp phần mở rộng phạm vi ứng dụng của khoa học dữ liệu trong thực tiễn.
      * ***Hỗ trợ công tác dự báo thời tiết***: Kết quả của nghiên cứu có thể giúp các cơ quan khí tượng dự báo lượng mưa, từ đó hỗ trợ trong công tác phòng chống thiên tai, nông nghiệp, và quản lý nguồn nước.
      * ***Phát triển kỹ năng phân tích***: Quá trình xử lý dữ liệu, xây dựng mô hình

và đánh giá dự báo giúp người thực hiện nâng cao kỹ năng phân tích, tư duy logic và ứng dụng mô hình học máy vào các bài toán thực tế.

### Cơ hội và khó khăn dự tính

Khi nghiên cứu các sắc thái của dự báo lượng mưa, nhóm chúng em gặp phải một bối cảnh phong phú với cả những lợi ích tiềm năng và những thách thức như:

* + - * ***Dự đoán biến động lượng mưa***: Các yếu tố tự nhiên phức tạp, không ổn định như bão, áp thấp nhiệt đới gây khó khăn cho việc dự đoán chính xác lượng mưa.
      * ***Tác động của các yếu tố bên ngoài***: Các hiện tượng như El Niño, biến đổi khí hậu làm thay đổi các mô hình mưa truyền thống, dẫn đến khó khăn trong việc xây dựng mô hình dự báo ổn định.
      * ***Đòi hỏi dữ liệu chất lượng cao***: Mô hình hồi quy tuyến tính yêu cầu dữ liệu sạch, chính xác, trong khi dữ liệu khí tượng thường gặp nhiều sai lệch do yếu tố đo lường và môi trường.

Về bản chất, việc phân tích và dự báo lượng mưa là một nhiệm vụ thách thức, đòi hỏi sự chính xác cao và khả năng xử lý sự biến động phức tạp của môi trường khí hậu. Tuy nhiên, nếu thực hiện tốt, nó có thể mang lại giá trị thực tiễn rất lớn trong nhiều lĩnh vực của đời sống.

# CHƯƠNG 2. CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN

## Phương pháp phân tích mô tả

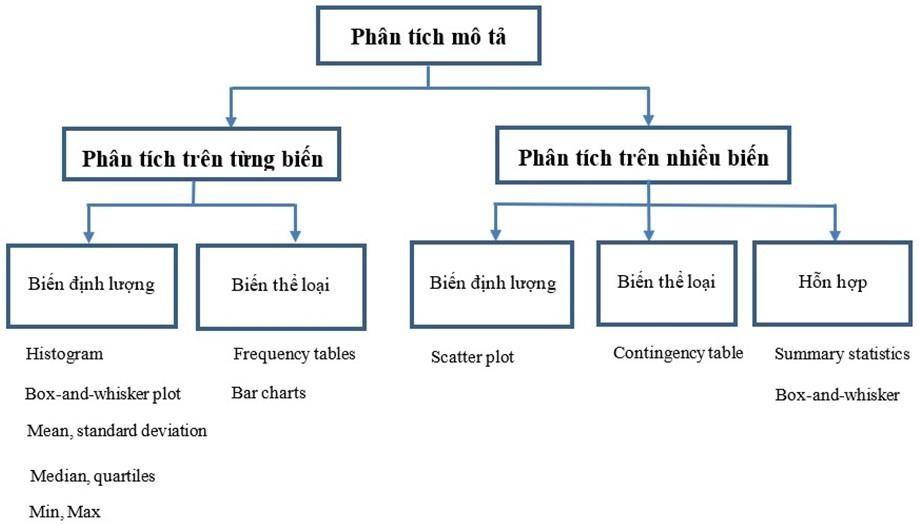
### Phân tích mô tả

Phân tích mô tả (thường được hiểu là thống kê mô tả) là một phương pháp thống kê được sử dụng để tóm tắt, sắp xếp, đơn giản hóa, mô tả và trình bày dữ liệu đã thu thập dưới dạng số hoặc biểu đồ trực quan, nhằm mô tả và tóm tắt các đặc điểm chính của một tập dữ liệu một cách dễ hiểu và ngắn gọn.

Mục tiêu của phân tích mô tả là giúp hiểu sâu hơn về dữ liệu mà chúng ta đang làm việc, nhận ra các đặc trưng quan trọng, và cung cấp một cái nhìn tổng quan về phân phối và biến đổi của dữ liệu.

Tùy thuộc vào loại biến hay kiểu dữ liệu để quyết định sử dụng các phương pháp tiếp cận phù hợp. Dữ liệu được chia thành hai loại:

* ***Dữ liệu định lượng*** (quantitative data) hay biến định lượng (quantitative variable): thường đại diện cho số lượng hoặc giá trị số (ví dụ: tuổi, cân nặng, giá tiền, âm lượng, v.v.)
* ***Dữ liệu thể loại*** (categorical data) hay biến thể loại (categorical variable): mô tả chất lượng hoặc đặc điểm của các đối tượng (ví dụ: màu sắc, dân tộc, giới tính, v.v.)



*Hình 2.1: Biểu đồ phần tích mô tả*

Phân tích mô tả thường bao gồm các khía cạnh sau:

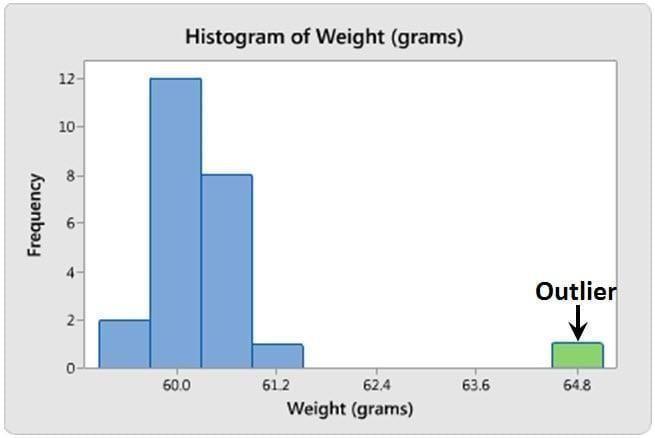
***Thống kê tóm tắt***: Đây là các số liệu thống kê cơ bản như trung bình, trung vị, độ lệch chuẩn, và phân vị. Các số liệu này giúp ta hiểu về trung tâm và phân tán của dữ liệu.

* + - * + ***Biểu đồ***: Biểu đồ thường được sử dụng để biểu diễn dữ liệu một cách trực quan. Các biểu đồ như biểu đồ cột, biểu đồ đường, biểu đồ hình tròn, và biểu đồ hộp giúp ta thấy được sự phân bố và xu hướng của dữ liệu.
        + ***Phân phối dữ liệu***: Phân tích phân phối dữ liệu giúp ta hiểu về tỷ lệ xuất hiện của các giá trị khác nhau trong tập dữ liệu. Điều này có thể làm bằng cách tạo biểu đồ phân phối tần số hoặc xây dựng biểu đồ kernel density.
        + ***Kiểm tra sự tương quan***: Phân tích mô tả cũng có thể liên quan đến việc kiểm tra sự tương quan giữa các biến. Điều này có thể thực hiện bằng cách sử dụng biểu đồ tương quan hoặc tính toán hệ số tương quan Pearson.
        + ***Xác định điểm ngoại lệ***: Phân tích mô tả cũng giúp xác định các điểm dữ liệu ngoại lệ, tức là những giá trị rất khác biệt so với phần còn lại của dữ liệu.
        + ***Tổng kết và nhận xét***: Cuối cùng, phân tích mô tả thường đi kèm với việc tổng kết và nhận xét về các đặc điểm quan trọng của dữ liệu, những mẫu thú vị, và những điểm mạnh và điểm yếu của tập dữ liệu.

Phân tích mô tả giúp xây dựng một cái nhìn sâu hơn về tập dữ liệu ban đầu và tạo nền tảng cho các phân tích tiếp theo như dự báo, phân tích hồi quy, hay machine learning.

### Phương pháp phân tích trên từng biến

Khi thực hiện phân tích trên một biến (hoặc một thuộc tính), mục tiêu chính là hiểu rõ các đặc điểm cơ bản của biến đó. Điều này thường bao gồm xác định và xử lý các giá trị ngoại lai hoặc bất thường (Outliers). Đây là các giá trị dữ liệu mà rất khác biệt so với phần lớn các giá trị khác trong tập dữ liệu. Các giá trị ngoại lai có thể xuất hiện do lỗi nhập liệu, lỗi đo lường, hoặc đơn giản là do các sự kiện hiếm gặp.



*Hình 2.1: Biểu đồ Histogram giúp xác định giá trị ngoại lai (Outliers)*

Việc xác định các Outliers có vai trò quan trọng và là mắt xích liên kết giữa phân tích mô tả và phân tích hồi quy, bởi vì ta có thể tiến hành làm sạch những giá trị này tại công đoạn tiền xử lý dữ liệu của phân tích hồi quy. Cụ thể với từng loại dữ liệu khác nhau, ta sẽ phân tích như sau:

#### Dữ liệu số:

***Biểu đồ Histogram***: Biểu đồ hiển thị tần suất xuất hiện của các khoảng giá trị dữ liệu.

* + - * + ***Các đại lượng thống kê***: Bao gồm mean (trung bình), stdev (độ lệch chuẩn), median (trung vị), quartile (phân vị)... Các giá trị này giúp mô tả trung bình, phương sai và phân phối của dữ liệu.
        + ***Biểu đồ Box & Whisker (Boxplot)***: Biểu đồ hiển thị tổng quan giá trị đó bao gồm các giá trị đại lượng thống kê đã tính được.

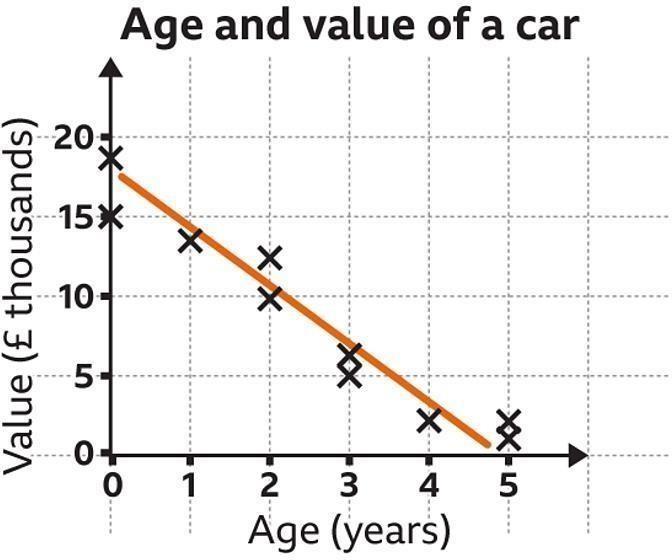
#### Dữ liệu phi số:

* + - * + ***Bảng tần suất (Frequency table)***: Biểu đồ liệt kê các giá trị khác nhau của biến và số lần xuất hiện của mỗi giá trị.
        + ***Biểu đồ cột (Bar chart)***: Biểu đồ thể hiện tần suất của từng giá trị dữ liệu dưới dạng các cột đứng.
        + ***Biểu đồ hình tròn hoặc donut (Pie chart, Donut chart)***: Biểu đồ thể hiện phần trăm tần suất của từng giá trị trong tổng số.

### Phương pháp phân tích trên nhiều biến

Phân tích trên nhiều biến hướng tới việc hiểu mối quan hệ và tương tác giữa các biến trong tập dữ liệu. Điều này có thể giúp bạn phát hiện ra các mẫu, xu hướng hoặc tương quan có thể tồn tại giữa chúng.

Các mối liên hệ giữa các biến (Interrelationships) có thể là nhiều dạng khác nhau: Mối tương quan tuyến tính, tương quan không tuyến tính, tương quan ngược... Với mỗi mối liên hệ, ta có thể phân tích và tìm ra được cách các biến tương tác và ảnh hưởng lẫn nhau.



*Hình 2.2: Biểu đồ Scatter thể hiện mối quan hệ giữa độ tuổi và giá bán*

Việc phân tích trên nhiều biến cũng có mối liên hệ mật thiết đến phân tích hồi quy khi giúp ta xác định được các giá trị ngoại lai của dữ liệu. Do là phân tích nhiều biến, vậy nên sẽ có 3 kiểu dữ liệu phân tích khác nhau: Số, phi số và hỗn hợp (cả số và phi số):

#### Dữ liệu số:

* + - * + ***Scatter Plot (Biểu đồ Scatter)***: Biểu đồ thể hiện mối quan hệ giữa hai biến số. Mỗi điểm trên biểu đồ thể hiện một cặp giá trị của hai biến trên trục ngang và dọc. Biểu đồ này dùng để tìm kiếm sự tương quan giữa 2 biến số như tương quan tuyến tính hoặc không tuyến tính.
        + ***Bảng dữ liệu thống kê (Statistical Summary Table)***: Tạo bảng để liệt kê các đại lượng thống kê (mean, median, stdev…) giữa các biến số của dữ liệu.

#### Dữ liệu phi số:

***Bảng dữ liệu thống kê (Statistical Summary Table)***: Cũng là bảng dữ liệu thống kê nhưng với giá trị phi số, đó sẽ chỉ có giá trị tần suất xuất hiện (mode) của dữ liệu.

#### Dữ liệu hỗn hợp

* + - * + ***Bảng thống kê tổng hợp***: Đây là sự kết hợp giữa bảng dữ liệu thống kê của dữ liệu số và phi số. Sự kết hợp tổng quan này sẽ cho ta bao quát được phân bổ của dữ liệu.
        + ***Biểu đồ Box-and-Whisker (Boxplot)***: Được sử dụng để so sánh phân phối của một dữ liệu số với tần suất của một dữ liệu phi số. Biểu đồ này sẽ cho ta mối quan hệ mật thiết về sự ảnh hưởng của các giá trị phi số lên giá trị số được phân tích.

## Phương pháp phân tích hồi quy

### Tổng quan về phân tích hồi quy

Phân tích hồi quy (Regression Analysis) là một phương pháp thống kê được sử dụng để khám phá, mô hình hóa và phân tích mối quan hệ giữa một biến phụ thuộc (còn gọi là biến cần dự đoán) và một hoặc nhiều biến độc lập (các biến giải thích). Mục tiêu chính của phân tích hồi quy là ước lượng mối quan hệ giữa các biến bằng cách xây dựng một hàm số, trong đó biến phụ thuộc được biểu diễn như một hàm của các biến độc lập cộng với một phần sai số

Trong phân tích hồi quy, ta cần xác định một biến phụ thuộc – yếu tố chính mà ta đang cố gắng hiểu hoặc dự đoán. Phân tích hồi quy bao gồm một số biến thể, chẳng hạn như tuyến tính, nhiều tuyến tính và phi tuyến tính. Trong đó mô hình phổ biến là tuyến tính và nhiều tuyến tính. Đối với phân tích hồi quy phi tuyến, chúng thường được sử dụng cho các tập dữ liệu phức tạp hơn trong đó các biến phụ thuộc và độc lập thể hiện mối quan hệ phi tuyến.

### Một số phương pháp phân tích hồi quy

Để phân tích hồi quy có rất nhiều phương pháp để phân tích. Dưới đây sẽ là một số phương pháp quan trọng dùng để phân tích hồi quy:

* ***Hồi quy tuyến tính (Linear Regression)***: Hồi quy tuyến tính là phương pháp

cơ bản và dễ hiểu nhất trong phân tích hồi quy. Mục tiêu của phương pháp này là dự đoán giá trị của biến phụ thuộc dựa trên biến độc lập bằng cách tìm một đường thẳng "tốt nhất" (đường hồi quy) phù hợp với dữ liệu.

* ***Hồi quy Ridge (Ridge Regression)***: Hồi quy Ridge là một phiên bản cải tiến của hồi quy tuyến tính bằng cách thêm một thuật ngữ điều chuẩn L2 vào hàm mất mát. Thuật ngữ này giúp kiểm soát độ phức tạp của mô hình và giảm nguy cơ quá khớp (overfitting)
* ***Hồi quy Lasso (Lasso Regression)***: Giống như Ridge, hồi quy Lasso cũng mở rộng từ hồi quy tuyến tính, nhưng khác biệt ở chỗ nó sử dụng điều chuẩn L1 thay vì L2. Việc áp dụng điều chuẩn L1 khiến một số hệ số hồi quy bị đẩy về đúng bằng 0, từ đó tự động thực hiện việc **lựa chọn biến**

### Phương pháp Lasso Regression

#### Giới thiệu

**Hồi quy LASSO** (Least Absolute Shrinkage and Selection Operator) là một kỹ thuật vừa thực hiện **co rút** (shrinkage) — làm nhỏ các hệ số — vừa **lựa chọn biến** một cách tự động.

Trong Lasso, một số giá trị của các tham số được ép về đúng 0 nhờ vào thành phần phạt trong hàm mục tiêu. Điều này giúp xây dựng các mô hình **thưa thớt** (sparse models), chỉ giữ lại những biến thực sự quan trọng

Mục tiêu chính của Lasso là đạt được sự cân bằng hợp lý giữa **độ phức tạp của mô hình** và **độ chính xác trong dự đoán**, bằng cách hạn chế số lượng biến được chọn vào mô hình. Nhờ vậy, Lasso không chỉ giúp nâng cao hiệu suất dự đoán mà còn cải thiện khả năng giải thích kết quả phân tích.

#### Một số khái niệm chính

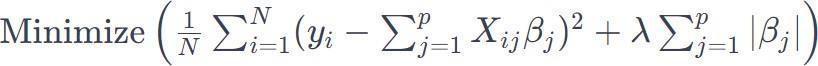
* + - * + **Chính quy hóa**: Chính quy hóa là một kỹ thuật được sử dụng để ngăn chặn việc trang bị quá mức bằng cách không khuyến khích các mô hình quá phức tạp trong hồi quy. Điều này đạt được bằng cách thêm một số hạng phạt vào hàm mất mát.
        + **Độ thưa thớt**: Một tính năng chính của hồi quy LASSO là khả năng tạo ra các mô hình thưa thớt. Độ thưa thớt đề cập đến các mô hình trong đó một số hệ số

chính xác bằng 0, ngụ ý rằng các tính năng tương ứng bị loại trừ hoàn toàn khỏi mô hình. Điều này đặc biệt hữu ích cho việc lựa chọn tính năng trong bộ dữ liệu nhiều chiều.

* + - * + **Đánh đổi sai lệch-phương sai**: Bằng cách đưa ra thuật ngữ chính quy hóa, hồi quy LASSO làm tăng độ lệch nhưng làm giảm phương sai của các dự đoán mô hình, dẫn đến khái quát hóa tốt hơn về dữ liệu không nhìn thấy.
        + **Tham số điều chỉnh (λ):** Cường độ của hình phạt được xác định bởi tham số điều chỉnh, λ. Việc lựa chọn λ có thể có tác động đáng kể đến kết quả của mô hình, với các giá trị cao hơn dẫn đến sự chính quy hóa nhiều hơn (nghĩa là nhiều hệ số được đặt thành 0).

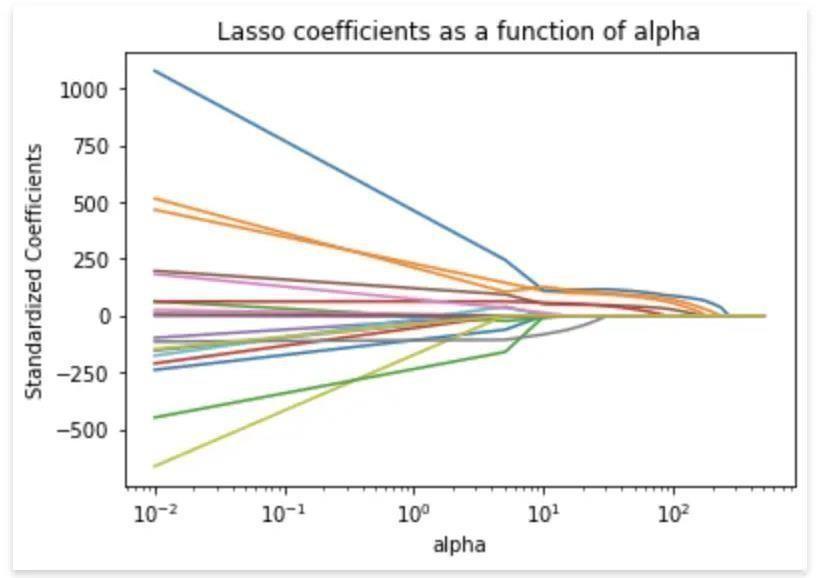
#### Mô hình hồi quy LASSO

Hàm mục tiêu cho hồi quy LASSO có thể được viết theo công thức:



Trong đó:

* + - * + ***yi*** là biến phụ thuộc.
        + ***Xij*** là biến dự đoán.
        + ***βj*** là các hệ số cần tìm.
        + ***N*** là số lượng quan sát.
        + ***P*** là số lượng các biến dự đoán.
        + ***λ*** là tham số chính quy kiểm soát cường độ của hình phạt. Giá trị của λ càng lớn thì mức độ co ngót càng lớn.



*Hình 2.3: Hệ số Lasso là hàm của λ*

Tham số điều chỉnh, λ kiểm soát cường độ của hình phạt L1. λ về cơ bản là độ co

ngót:

* Khi λ = 0, không có tham số nào bị loại bỏ. Ước tính này bằng với ước tính được tìm thấy bằng hồi quy tuyến tính.
* Khi λ tăng lên, ngày càng có nhiều hệ số được đặt về 0 và bị loại bỏ (về mặt lý thuyết, khi λ = ∞, tất cả các hệ số đều bị loại bỏ).
* Khi λ tăng, độ lệch tăng.
* Khi λ giảm, phương sai tăng

#### Ưu điểm của hồi quy LASSO

Hồi quy Lasso là một kỹ thuật hồi quy tuyến tính được sử dụng rộng rãi, mang lại một số lợi thế so với hồi quy tuyến tính truyền thống:

**- Lựa chọn tính năng:**

Hồi quy Lasso có thể được sử dụng để lựa chọn tính năng, trong đó nó xác định các yếu tố dự đoán quan trọng nhất và loại bỏ phần còn lại. Điều này có thể đặc biệt hữu ích khi xử lý các tập dữ liệu nhiều chiều, trong đó số lượng yếu tố dự đoán lớn và nhiều trong số chúng có thể không liên quan hoặc dư thừa.

**- Chính quy hóa:**

Hồi quy Lasso bao gồm một hình phạt chính quy hóa trong hàm mục tiêu, giúp ngăn chặn việc điều chỉnh quá mức và cải thiện hiệu suất tổng quát hóa của mô

hình. Hình phạt thu hẹp các hệ số về 0, dẫn đến một mô hình đơn giản hơn và ít có khả năng bị trang bị quá mức.

**- Độ thưa thớt:**

Hồi quy Lasso khuyến khích sự thưa thớt trong mô hình bằng cách thu nhỏ một số hệ số về chính xác bằng 0. Điều này dẫn đến một mô hình có ít yếu tố dự đoán hơn, dễ diễn giải hơn và có thể dẫn đến độ chính xác dự đoán tốt hơn.

* Sự đánh đổi độ lệch-phương sai:

Hồi quy Lasso đưa ra sự cân bằng giữa sai lệch và phương sai bằng cách kiểm soát độ mạnh của hình phạt chính quy hóa. Bằng cách tăng mức phạt, Lasso có thể giảm phương sai trong mô hình với cái giá phải trả là độ lệch tăng lên hoặc ngược lại.

**- Có thể giải thích:**

Hồi quy Lasso tạo ra một mô hình dễ diễn giải vì nó chỉ bao gồm một tập hợp con các biến dự đoán và gán hệ số 0 cho các hệ số không liên quan. Điều này có thể giúp hiểu được mối quan hệ giữa các yếu tố dự đoán và biến mục tiêu.

**- Tính linh hoạt:**

Hồi quy Lasso có thể được áp dụng cho nhiều vấn đề hồi quy, bao gồm hồi quy tuyến tính và phi tuyến tính, cũng như các mô hình tuyến tính tổng quát. Nó cũng tương thích với các thuật toán tối ưu hóa khác nhau và có thể xử lý cả tập dữ liệu nhỏ và lớn.

Nhìn chung, hồi quy Lasso là một kỹ thuật mạnh mẽ và linh hoạt, mang lại một số lợi thế so với hồi quy tuyến tính truyền thống, khiến nó trở thành lựa

chọn phổ biến trong các ứng dụng phân tích dữ liệu và học máy.

#### Nhược điểm của hồi quy LASSO

Mặc dù hồi quy Lasso là một kỹ thuật hồi quy phổ biến và hữu ích nhưng nó cũng có một số hạn chế và nhược điểm cần được xem xét:

**- Xu hướng lựa chọn tính năng:**

Hình phạt chính quy hóa Lasso có thể dẫn đến một số tính năng bị loại trừ hoàn toàn khỏi mô hình, ngay cả khi chúng có thể là yếu tố dự báo quan trọng của biến mục tiêu. Điều này có thể dẫn đến việc lựa chọn tính năng sai lệch, đặc biệt nếu

mối quan hệ thực sự giữa các yếu tố dự đoán và mục tiêu không thưa thớt.

**- Thông số không ổn định:**

Lasso có thể nhạy cảm với những thay đổi nhỏ trong dữ liệu đầu vào, dẫn đến sự chênh lệch lớn trong các hệ số ước tính. Điều này có thể dẫn đến sự mất ổn định trong các tham số của mô hình, gây khó khăn cho việc diễn giải kết quả.

**- Sự đánh đổi độ lệch-phương sai:**

Hình phạt chính quy hóa Lasso có thể làm giảm phương sai trong mô hình bằng cách thu hẹp các hệ số về 0, nhưng nó cũng có thể gây ra sai lệch bằng cách đánh giá thấp các hệ số thực. Sự cân bằng tối ưu giữa sai lệch và phương sai phụ thuộc vào vấn đề và tập dữ liệu cụ thể.

**- Trang bị quá mức:**

Mặc dù Lasso được thiết kế để ngăn chặn việc khớp quá mức bằng cách đưa ra một thuật ngữ phạt, nhưng mô hình vẫn có thể khớp quá mức với dữ liệu huấn luyện nếu tham số chính quy hóa không được điều chỉnh đúng cách. Điều này có thể dẫn đến hiệu suất khái quát hóa kém trên dữ liệu mới.

**- Lựa chọn tham số chính quy:**

Việc lựa chọn tham số chính quy α là rất quan trọng đối với hiệu suất của mô hình Lasso. Tuy nhiên, không có giải pháp phân tích nào để tìm giá trị tối ưu của α và nó phải được xác định bằng thực nghiệm bằng cách xác nhận chéo.

Việc này có thể tốn thời gian và không phải lúc nào cũng đưa đến sự lựa chọn tốt nhất về α.

**- Đa cộng tuyến:**

Lasso có thể nhạy cảm với đa cộng tuyến, đó là khi hai hoặc nhiều yếu tố dự đoán có mối tương quan cao. Trong trường hợp này, Lasso có thể chọn một trong các yếu tố dự đoán tương quan và loại trừ yếu tố còn lại, ngay cả khi cả hai đều quan trọng để dự đoán biến mục tiêu.

Nhìn chung, mặc dù hồi quy Lasso là một kỹ thuật hữu ích để chính quy hóa và lựa chọn tính năng, nhưng điều quan trọng là phải xem xét cẩn thận các hạn chế và nhược điểm tiềm ẩn của nó khi áp dụng nó cho một vấn đề hoặc tập dữ liệu cụ thể.

#### Tính ứng dụng và hạn chế của hồi quy LASSO

LASSO được sử dụng rộng rãi trong các lĩnh vực mà khả năng diễn giải mô hình và lựa chọn tính năng là quan trọng, chẳng hạn như tin sinh học, tâm lý học và kinh tế. Tuy nhiên, hạn chế của nó nằm ở việc lựa chọn ngẫu nhiên giữa các biến có mối tương quan cao và các ước tính có khả năng sai lệch khi λ lớn.

* ***Khi nào nên sử dụng hồi quy LASSO?***

Hồi quy Lasso đặc biệt hữu ích khi xử lý các tập dữ liệu nhiều chiều, trong đó số lượng yếu tố dự đoán (đặc điểm) lớn và nhiều trong số chúng có thể không liên quan hoặc dư thừa. Trong những trường hợp như vậy, các kỹ thuật hồi quy tuyến tính truyền thống có thể phù hợp quá mức với dữ liệu và không thể khái quát hóa tốt dữ liệu mới.

Hồi quy Lasso có thể giúp giảm số lượng yếu tố dự đoán và chọn những yếu tố quan trọng nhất, từ đó cải thiện độ chính xác và khả năng diễn giải của mô hình. Điều này đặc biệt hữu ích trong những tình huống tập trung vào việc tìm hiểu mối quan hệ giữa các yếu tố dự đoán và biến mục tiêu, thay vì chỉ đơn giản là dự đoán kết quả.

* Khi nào không nên sử dụng hồi quy LASSO?

Mặc dù hồi quy Lasso có thể là một kỹ thuật mạnh mẽ để lựa chọn và chính quy hóa tính năng trong các mô hình hồi quy tuyến tính, nhưng nó có thể không phải lúc nào cũng là lựa chọn tốt nhất cho mọi tình huống. Dưới đây là một số tình huống mà hồi quy Lasso có thể không phải là cách tiếp cận tốt nhất:

* Cỡ mẫu nhỏ
* Mối quan hệ phi tuyến
* Các yếu tố dự đoán tương quan
* Dự đoán phân loại
* Ngoại lệ

#### Kết luận

Tóm lại, hồi quy Lasso là một kỹ thuật mạnh mẽ và linh hoạt để lựa chọn và chính quy hóa tính năng trong các mô hình hồi quy tuyến tính. Nó cung cấp một số lợi thế so với hồi quy tuyến tính truyền thống, bao gồm lựa chọn tính năng, chính quy

hóa, độ thưa thớt, cân bằng phương sai sai lệch, khả năng diễn giải và tính linh hoạt. Hồi quy Lasso đặc biệt hữu ích khi xử lý các tập dữ liệu nhiều chiều, trong đó số lượng yếu tố dự đoán lớn và nhiều trong số chúng có thể không liên quan hoặc dư thừa.

### Phương pháp Ridge Regression

#### Giới thiệu

Hồi quy Ridge là một kỹ thuật được sử dụng để ước lượng các hệ số trong mô hình hồi quy bội, đặc biệt trong những trường hợp các biến độc lập có mối tương quan cao với nhau. Phương pháp này đã được áp dụng rộng rãi trong nhiều lĩnh vực như kinh tế lượng, hóa học và kỹ thuật.

Còn được gọi là chính quy hóa Tikhonov (Tikhonov regularization), theo tên nhà toán học Andrey Tikhonov, hồi quy Ridge là một hình thức điều chỉnh các bài toán ngược không xác định hoặc dễ mắc sai sót. Phương pháp này rất hiệu quả trong việc xử lý đa cộng tuyến — hiện tượng xảy ra khi các biến độc lập trong mô hình có quan hệ tuyến tính mạnh với nhau — một vấn đề phổ biến trong các mô hình hồi quy phức tạp có nhiều biến.

Hồi quy Ridge ra đời như một giải pháp thực tiễn nhằm khắc phục sự thiếu chính xác của các ước lượng bằng phương pháp OLS (Ordinary Least Squares) trong các tình huống đa cộng tuyến. Bằng cách đưa ra bộ ước lượng mới — gọi là ước lượng hồi quy Ridge (RR) — phương pháp này tạo ra các tham số hồi quy với phương sai thấp hơn và sai số trung bình bình phương (MSE) nhỏ hơn so với ước lượng truyền thống.

#### Mô hình hồi quy Ridge Regression

Giả định dữ liệu đầu vào bao gồm N quan sát là những cặp các biến đầu vào và biến mục tiêu (x1, y1), (x2, y2), ..., (xn, yn). Quá trình hồi qui mô hình sẽ tìm kiếm một véc tơ hệ số ước lượng w = [w0, w1, ..., wp] sao cho tối thiểu hoá hàm mất mát dạng MSE:

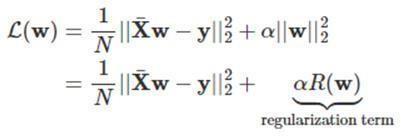
A black text on a white background  Description automatically generated

Hàm mất mát cũng chính là mục tiêu tối ưu khi huấn luyện mô hình. Dữ liệu đầu vào X và y được xem như là cố định và biến số của bài toán tối ưu chính là các giá trị trong véc tơ w.

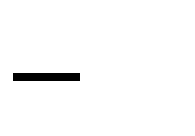
Giá trị hàm mất mát MSE chính là trung bình của tổng bình phương phần dư. Phần dư chính là chênh lệch giữa giá trị thực tế và giá trị dự báo. Tối thiểu hoá hàm mất mát nhằm mục đích làm cho giá trị dự báo ít chênh lệch so với giá trị thực tế, giá trị thực tế còn được gọi là ground truth. Trước khi huấn luyện mô hình chúng ta chưa thực sự biết véc tơ hệ số w là gì. Chúng ta chỉ có thể đặt ra một giả thuyết về dạng hàm dự báo (trong trường hợp này là phương trình dạng tuyến tính) và các hệ số hồi quy tương ứng. Chính vì vậy mục đích của tối thiểu hoá hàm mất mát là để tìm ra tham số w phù hợp nhất mô tả một cách khái quát quan hệ dữ liệu giữa biến đầu vào X với biến mục tiêu y trên tập huấn luyện

#### Sự thay đổi của hàm mất mát trong Ridge Regression

Hàm mất mát trong hồi quy Ridge sẽ có sự thay đổi so với hồi qui tuyến tính đó là thành phần điều chuẩn (regularization term) được cộng thêm vào hàm mất mát như sau:



Trong phương trình trên thì **α ≥ 0.**  𝟏**||** 𝑿𝒘 𝒚**||** 𝟐 chính là tổng bình 𝑵 𝟐



phương phần dư và **α ||w||** 𝟐 đại diện cho *thành phần điều chuẩn*.

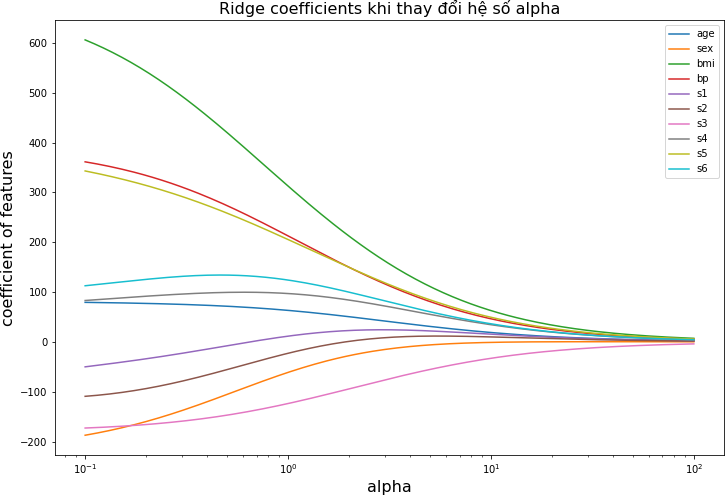
Bài toán tối ưu hàm mất mát củ𝟐a hồi quy Ridge về bản chất là tối ưu song song hai thành phần bao gồm tổng bình phương phần dư và thành phần điều chuẩn. Hệ số α có tác dụng điều chỉnh độ lớn của thành phần điều chuẩn tác động lên hàm mất mát.

* Trường hợp **α =0**, thành phần điều chuẩn bị tiêu giảm và chúng ta quay trở về bài toán hồi quy tuyến tính.
* Trường hợp **α nhỏ** thì vai trò của thành phần điều chuẩn trở nên ít quan trọng.

Mức độ kiểm soát quá khớp của mô hình sẽ trở nên kém hơn.

* Trường hợp **α lớn** chúng ta muốn gia tăng mức độ kiểm soát lên độ lớn của các hệ số ước lượng và qua đó giảm bớt hiện tượng quá khớp.

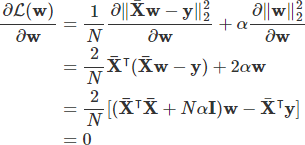
Khi tăng dần hệ số α thì hồi quy Ridge sẽ có xu hướng thu hẹp hệ số ước lượng từ mô hình. Chúng ta sẽ thấy rõ thông qua ví dụ mẫu bên dưới.



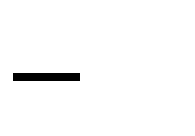
*Hình 2.4: Sự thay đổi của độ lớn các hệ số ước lượng (coefficient of features) theo hệ số điều chuẩn α. Khi tăng dần độ lớn của α thì độ lớn của hệ*

#### Nghiệm tối ưu của Ridge Regression

* **Giải bài toán tối ưu hàm mục tiêu của hồi quy Ridge theo đạo hàm bậc nhất của vector w:**



Thật vậy, từ dòng 1 suy ra dòng 2 là vì theo công thức product-rule trong matrix calculus thì:



𝒚

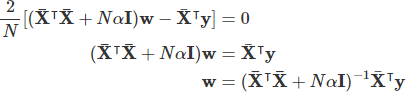
Nếu thay f(w) = g(w) = 𝑿𝒘 ta suy ra:



Như vậy ta nhận thấy dòng 1 suy ra dòng 2 là hoàn toàn đúng.

Ở dòng thứ 3 chúng ta áp dụng thêm một tính chất **Iw = w** trong đó **I** là ma trận đơn vị.

Sau cùng nghiệm của đạo hàm bậc nhất trở thành:

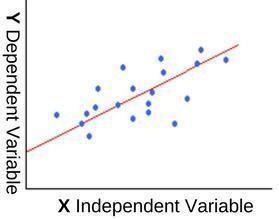


### Phương pháp Linear Regression

#### Giới thiệu

"Hồi quy tuyến tính" là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó v.v...

Linear hay tuyến tính hiểu một cách đơn giản là thẳng, phẳng. Trong không gian hai chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một đường thẳng. Trong không gian ba chiều, một hàm số được goi là tuyến tính nếu đồ thị của nó có dạng một mặt phẳng. Trong không gian nhiều hơn 3 chiều, khái niệm mặt phẳng không còn phù hợp nữa, thay vào đó, một khái niệm khác ra đời được gọi là siêu mặt phẳng (hyperplane)



*Hình 2.5: Mối quan hệ tuyến tính giữa biến đầu ra (y) và biến dự đoán*

Biểu đồ trên trình bày mối quan hệ tuyến tính giữa các biến đầu ra (y) và biến dự đoán (X). Đường màu đỏ được gọi là đường thẳng phù hợp nhất. Dựa trên các điểm dữ liệu đã cho, mô hình sẽ cố gắng vẽ một đường phù hợp nhất với các điểm đó.

#### Công thức/Mô hình tuyến tính

Trong thực tế, chúng ta chỉ có thể có một biến độc lập X ảnh hưởng đến biến phụ thuộc Y. Hoặc có thể xảy ra trường hợp có nhiều biến độc lập ảnh hưởng đến Y. Dựa trên cách tiếp cận này, có hai loại hồi quy tuyến tính chính:

**- Mô hình hồi quy tuyến tính đơn biến:**

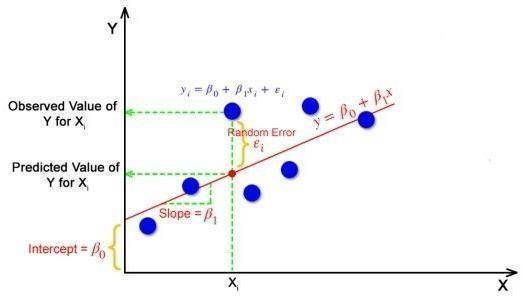
Hồi quy tuyến tính đơn giản có nghĩa là chỉ có một biến X độc lập mà những thay đổi này dẫn đến các giá trị khác nhau cho Y.

Công thức:

Y = β₀ + β₁X + ε

Trong đó:

* + - * + Y là biến phụ thuộc (cũng được gọi là biến mục tiêu hoặc biến phản hồi).
        + X là biến độc lập (cũng được gọi là biến giải thích hoặc biến đầu vào).
        + β₀ và β₁ là các hệ số hồi quy (cũng được gọi là hệ số góc và hệ số chặn).
        + ε là sai số ngẫu nhiên (cũng được gọi là sai số hồi quy).

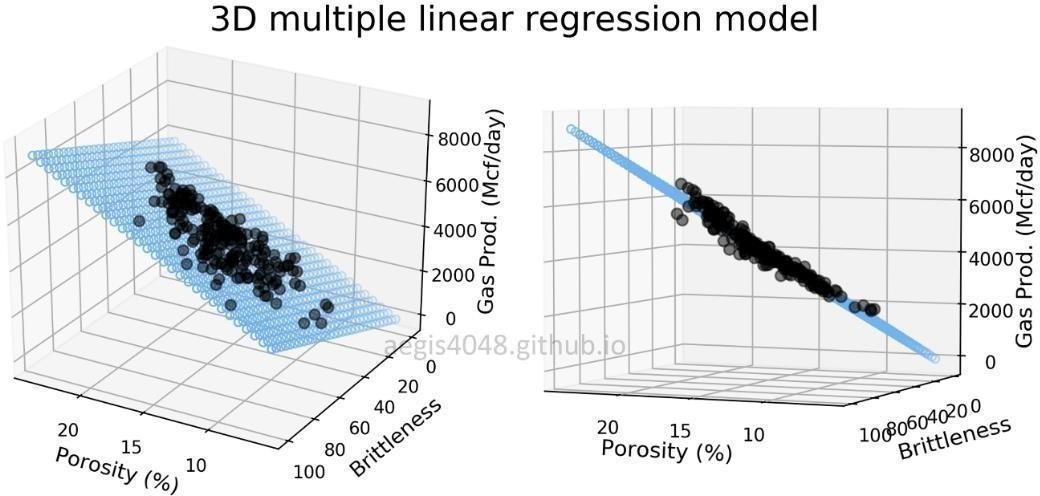


*Hình 2.6: Tính toán hồi quy đơn biến*

Mục tiêu của thuật toán hồi quy tuyến tính là lấy giá trị tốt nhất cho β₀ và β₁ để tìm ra đường thẳng phù hợp nhất. Đường thẳng phù hợp nhất là đường thẳng có ít lỗi nhất, nghĩa là sai số giữa giá trị dự đoán và giá trị thực tế phải ở mức tối thiểu.

**- Mô hình hồi quy tuyến tính đa biến (tuyến tính bội)**

Mô hình hồi quy tuyến tính bội là loại phân tích hồi quy tuyến tính phổ biến nhất. Nó được sử dụng để thể hiện mối quan hệ giữa một biến phụ thuộc và hai hoặc nhiều biến độc lập. Trên thực tế, mô hình hồi quy tuyến tính đơn giản là một dạng đặc biệt của hồi quy tuyến tính đa biến



*Hình 2.7: Minh họa siêu phẳng phù hợp nhất của mô hình tuyến tính bội*

Mô hình hồi quy tuyến tính đa biến xử lý nhiều biến độc lập. Nó nhằm mục đích tìm ra một siêu phẳng phù hợp nhất với mối quan hệ giữa nhiều biến độc lập và biến phụ thuộc.

#### Phương pháp đánh giá kết quả

Sức mạnh của bất kỳ mô hình hồi quy tuyến tính nào cũng có thể được đánh giá bằng các số liệu đánh giá khác nhau. Các số liệu đánh giá này thường cung cấp thước đo về mức độ hiệu quả của các kết quả đầu ra được quan sát bởi mô hình. Các số liệu được sử dụng nhiều nhất là:

* + - * + Hệ số xác định hoặc R-Squared (R2)
        + Lỗi bình phương trung bình gốc (RSME) và lỗi chuẩn dư (RSE)

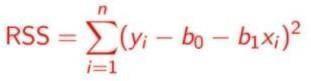
**- Hệ số xác định hoặc R-Squared (R2)**

R-Squared là con số giải thích mức độ biến thiên được giải thích/nắm bắt bởi mô hình đã phát triển. Nó luôn nằm trong khoảng từ 0 & 1. Nhìn chung, giá trị của R bình phương càng cao thì mô hình càng phù hợp với dữ liệu.

Về mặt toán học nó có thể được biểu diễn dưới dạng:

R2 = 1 – (RSS/TSS)

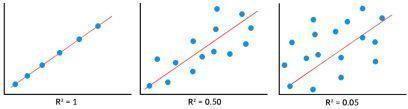
* + - * + **Tổng bình phương dư (RSS)** được định nghĩa là tổng bình phương của phần dư cho mỗi điểm dữ liệu trong biểu đồ/dữ liệu. Nó là thước đo sự khác biệt giữa kết quả dự kiến và kết quả thực tế được quan sát.



* + - * + **Tổng bình phương (TSS)** được định nghĩa là tổng sai số của các điểm dữ liệu so với giá trị trung bình của biến phản hồi. Về mặt toán học TSS là:

Tổng số bình phương

Trong đó y hat là giá trị trung bình của các điểm dữ liệu mẫu. Ý nghĩa của bình phương R được thể hiện bằng các hình sau:



*Hình 2.8: Ý nghĩa của bình phương R*

**- Lỗi bình phương gốc**

Sai số bình phương trung bình gốc là căn bậc hai của phương sai của phần dư. Nó chỉ định mức độ phù hợp tuyệt đối của mô hình với dữ liệu, tức là mức độ gần của các điểm dữ liệu được quan sát với các giá trị dự đoán. Về mặt toán học nó có thể được biểu diễn dưới dạng:



Để làm cho ước tính này không thiên vị, người ta phải chia tổng bình phương số dư cho bậc tự do thay vì chia tổng số điểm dữ liệu trong mô hình. Thuật ngữ này khi đó được gọi là Lỗi chuẩn dư (RSE). Về mặt toán học nó có thể được biểu diễn dưới dạng.

Lỗi chuẩn dư

R bình phương là thước đo tốt hơn RSME. Bởi vì giá trị của Lỗi bình phương trung bình gốc phụ thuộc vào đơn vị của các biến (tức là nó không phải là thước đo chuẩn hóa), nên nó có thể thay đổi khi có sự thay đổi về đơn vị của các biến.

#### Ưu điểm của hồi quy tuyến tính

Hồi quy tuyến tính là một trong những kỹ thuật thống kê cơ bản và được sử dụng rộng rãi nhất trong mô hình dự đoán. Dưới đây là những ưu điểm chính của nó:

**- Tính đơn giản và dễ hiểu:**

Các mô hình hồi quy tuyến tính rất đơn giản và dễ hiểu. Mối quan hệ giữa các biến độc lập và biến phụ thuộc được biểu diễn bằng một đường thẳng, giúp dễ hiểu và giải thích các dự đoán của mô hình.

**- Ít bị trang bị quá mức:**

Trong trường hợp có một hoặc một vài yếu tố dự đoán, hồi quy tuyến tính có thể ít bị khớp quá mức so với các mô hình phức tạp hơn. Điều này đặc biệt có giá trị khi mối quan hệ cơ bản trong dữ liệu thực sự là tuyến tính hoặc gần tuyến tính.

**- Cơ sở để hiểu các mô hình phức tạp hơn:**

Hồi quy tuyến tính đặt nền tảng để hiểu các thuật toán phức tạp hơn. Nhiều kỹ thuật tiên tiến được coi là phần mở rộng hoặc biến thể của hồi quy tuyến tính, khiến nó trở thành một khối xây dựng thiết yếu trong việc học mô hình dự đoán.

**- Tính toán hiệu quả:**

Các mô hình hồi quy tuyến tính có hiệu quả tính toán cao, đòi hỏi ít tài nguyên tính toán hơn. Điều này làm cho chúng có khả năng mở rộng cao đối với các tập dữ liệu lớn và phù hợp với các tình huống mà tốc độ đào tạo mô hình là rất quan trọng.

**- Tốt cho dự báo dự đoán:**

Hồi quy tuyến tính có thể là một công cụ hiệu quả để dự báo mang tính dự đoán, đặc biệt khi dữ liệu cho thấy xu hướng tuyến tính. Điều này phổ biến trong nhiều lĩnh vực khác nhau như kinh tế, sinh học và kỹ thuật.

#### Nhược điểm của hồi quy tuyến tính

Mặc dù có những ưu điểm nhưng hồi quy tuyến tính cũng có những hạn chế:

**- Giả sử mối quan hệ tuyến tính:**

Hồi quy tuyến tính giả định mối quan hệ tuyến tính giữa các biến độc lập và biến phụ thuộc. Đây có thể là một hạn chế lớn nếu mối quan hệ thực tế phức tạp hoặc phi tuyến tính.

**- Nhạy cảm với các ngoại lệ:**

Các mô hình hồi quy tuyến tính rất nhạy cảm với các ngoại lệ. Một vài ngoại lệ có thể ảnh hưởng đáng kể đến kết quả, làm cho mô hình kém tin cậy hơn.

**- Dễ xảy ra hiện tượng đa cộng tuyến:**

Đa cộng tuyến xảy ra khi các biến độc lập có mối tương quan cao. Điều này có thể làm sai lệch các mối quan hệ ước tính và làm giảm độ tin cậy của mô hình.

**- Giả định về tính đồng nhất:**

Hồi quy tuyến tính giả định rằng phương sai của các số hạng sai số là không đổi trên tất cả các cấp độ của các biến độc lập (tính đồng nhất). Nếu giả định này bị vi phạm (tính không đồng nhất), nó có thể dẫn đến các ước tính không hiệu quả.

**- Giới hạn ở các mối quan hệ tuyến tính:**

Nó chỉ có thể mô hình hóa các mối quan hệ tuyến tính, khiến nó không phù hợp để mô hình hóa các mẫu phức tạp hơn trong dữ liệu.

**- Nguy cơ đơn giản hóa các mối quan hệ quá mức:**

Khi cố gắng điều chỉnh một mô hình tuyến tính, có nguy cơ đơn giản hóa quá mức mối quan hệ giữa các biến, điều này có thể dẫn đến những dự đoán không chính xác.

#### Khả năng ứng dụng và hạn chế của hồi quy tuyến tính

Hồi quy tuyến tính được áp dụng rộng rãi trong các lĩnh vực mà mối quan hệ giữa các biến là tuyến tính hoặc gần tuyến tính. Nó thường được sử dụng trong kinh doanh, kinh tế, khoa học xã hội và một số khoa học tự nhiên. Tuy nhiên, hạn chế của nó nằm ở chỗ không có khả năng xử lý các mối quan hệ phi tuyến tính, dễ bị ảnh hưởng bởi các ngoại lệ và các vấn đề về đa cộng tuyến.

**- Khi nào nên sử dụng hồi quy tuyến tính?**

Hồi quy tuyến tính thích hợp nhất trong trường hợp dữ liệu thể hiện mối quan hệ tuyến tính. Nó phù hợp nhất để phân tích dự đoán trong trường hợp dữ liệu không quá phức tạp và mối quan hệ giữa các biến được hiểu rõ và có thể gần đúng bằng mô hình tuyến tính.

**- Khi nào bạn không nên sử dụng Hồi quy tuyến tính?**

Hồi quy tuyến tính có thể không phù hợp trong trường hợp mối quan hệ giữa các biến vốn không tuyến tính, khi xử lý các biến độc lập có mối tương quan cao hoặc khi dữ liệu chứa các giá trị ngoại lệ đáng kể.

#### Kết luận

Tóm lại, hồi quy tuyến tính, với tính đơn giản, dễ hiểu và hiệu quả, vẫn là nền tảng trong mô hình thống kê và phân tích dự đoán. Mặc dù nó mang lại nhiều lợi ích nhưng hiệu quả của nó phụ thuộc vào bản chất của dữ liệu và các mối quan hệ bên trong. Hiểu được những hạn chế của nó là rất quan trọng để có thể áp dụng nó một cách thích hợp và để có được những hiểu biết chính xác, đáng tin cậy từ việc sử dụng nó.

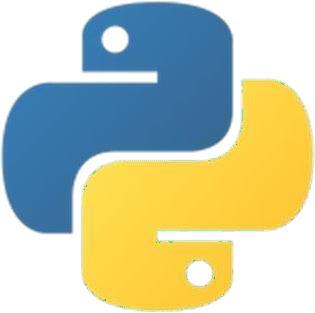
### 2.2.6. Lựa chọn phương pháp

Sau khi phân tích và đánh giá các phương pháp hồi quy phổ biến (Linear Regression, Ridge Regression, Lasso Regression), nhóm quyết định lựa chọn mô hình Hồi quy Tuyến tính (Linear Regression) làm phương pháp chính để giải quyết bài toán dự báo lượng mưa. Sự lựa chọn này dựa trên những căn cứ khoa học và thực tiễn sau đây:

* **Phù hợp với bản chất bài toán và đặc điểm dữ liệu :** Bài toán dự báo lượng mưa với các biến đầu vào như nhiệt độ, độ ẩm, áp suất, tốc độ gió có xu hướng thể hiện mối quan hệ gần tuyến tính trong nhiều điều kiện khí tượng ổn định. Các nghiên cứu tiền đề trong lĩnh vực khí tượng cũng chỉ ra rằng, trong giai đoạn đầu phân tích, mô hình tuyến tính thường đủ để nắm bắt xu hướng cơ bản và cung cấp kết quả dự báo có ý nghĩa thống kê.
* **Tính đơn giản và khả năng diễn giải cao :** Một trong những yêu cầu quan trọng của đề tài là không chỉ dự báo chính xác mà còn phải giải thích được mối quan hệ giữa các yếu tố khí tượng với lượng mưa. Linear Regression cung cấp các hệ số hồi quy rõ ràng, cho phép đánh giá mức độ ảnh hưởng và hướng tác động (dương/âm) của từng biến đến lượng mưa. Điều này có giá trị thực tiễn trong việc đưa ra khuyến nghị cho các nhà khí tượng học.
* **Hiệu quả tính toán và khả năng xử lý dữ liệu lớn trên Spark :** Với bộ dữ liệu khí tượng có quy mô lớn (hàng chục nghìn bản ghi), mô hình Linear Regression có ưu điểm vượt trội về tốc độ huấn luyện và dự báo. Trên nền tảng Apache Spark, thuật toán có thể được triển khai phân tán, tận dụng sức mạnh tính toán song song để xử lý nhanh chóng mà vẫn đảm bảo độ ổn định. Điều này phù hợp với mục tiêu xây dựng hệ thống dự báo có khả năng mở rộng.
* **Tính khả thi trong giai đoạn nghiên cứu khởi đầu :** Trong khuôn khổ một bài tập lớn với thời gian và nguồn lực có hạn, Linear Regression là lựa chọn tối ưu để:
* Thiết lập baseline model làm cơ sở so sánh với các phương pháp phức tạp sau này.
* Tập trung nguồn lực vào công đoạn tiền xử lý dữ liệu và đánh giá tính năng, thay vì điều chỉnh tham số phức tạp.
* Dễ dàng kiểm chứng và xác thực bằng các chỉ số thống kê chuẩn (R², RMSE, MAE).
* **Cơ sở để phát triển mở rộng :** Mô hình Linear Regression không chỉ là giải pháp cuối cùng mà còn đóng vai trò nền tảng để phát triển các mô hình nâng cao. Kết quả từ hồi quy tuyến tính sẽ giúp nhận diện các biến quan trọng, phát hiện hiện tượng đa cộng tuyến, từ đó định hướng cho việc áp dụng Ridge/Lasso Regression hoặc các mô hình phi tuyến trong các nghiên cứu tiếp theo.

## Công cụ phục vụ thực hiện bài toán

### Python



*Hình 2.9: Ngôn ngữ lập trình Python*

Python là một ngôn ngữ lập trình bậc cao được phát triển bởi Guido van Rossum và ra mắt lần đầu tiên vào năm 1991. Ngôn ngữ này nổi tiếng với cú pháp đơn

giản, dễ đọc, dễ học, phù hợp cho cả người mới bắt đầu và lập trình viên chuyên nghiệp. Python hỗ trợ nhiều phong cách lập trình như hướng đối tượng, lập trình hàm và lập trình thủ tục, đồng thời có hệ sinh thái thư viện và framework phong phú như NumPy, Pandas, TensorFlow, Django, Flask, giúp giải quyết hiệu quả nhiều bài toán thực tiễn. Python có tính đa năng cao, được ứng dụng rộng rãi trong phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, tự động hóa quy trình, lập trình ứng dụng giao diện người dùng và thậm chí cả trong lĩnh vực phát triển trò chơi. Với khả năng tích hợp tốt với các ngôn ngữ khác như C, C++ và Java, cùng cộng đồng người dùng rộng lớn trên toàn thế giới, Python đem lại nhiều thuận lợi trong việc học tập, phát triển dự án và giải quyết vấn đề.

### Apache Spark

****

*Hình 2.10: Ngôn ngữ lập trình Apache Spark*

Apache Spark là một **c**ông cụ xử lý dữ liệu phân tán mã nguồn mở hàng đầu, đóng vai trò quan trọng trong việc phân tích và xử lý khối lượng dữ liệu khổng lồ (Big Data). Nó được ứng dụng rộng rãi trong nhiều lĩnh vực, từ xử lý dữ liệu batch, streaming, đến Machine Learning và phân tích đồ thị.

Sức mạnh vượt trội của Spark nằm ở tốc độ xử lý đáng kinh ngạc, thường nhanh hơn tới 100 lần so với Hadoop MapReduce nhờ khả năng xử lý dữ liệu trong bộ nhớ (in-memory computing). Điều này cho phép Spark thực hiện các tác vụ phân tích phức tạp, các thuật toán lặp lại (iterative algorithms) và phân tích thời gian gần thực (near real-time analytics) một cách hiệu quả. Spark cũng nổi bật với tính linh hoạt cao, hỗ trợ đa ngôn ngữ lập trình (Scala, Java, Python, R) và cung cấp các API dễ sử dụng, giúp các nhà phát triển xây dựng các ứng dụng xử lý dữ liệu phân tán một cách nhanh chóng. Hơn nữa, với các thư viện tích hợp như Spark SQL, MLlib (Machine Learning), GraphX (Graph Processing) và Spark Streaming, Spark cung cấp một nền tảng thống nhất cho nhiều loại hình công việc phân tích dữ liệu.

Tuy nhiên, bên cạnh những ưu điểm vượt trội, Apache Spark cũng có một số thách thức cần được cân nhắc. Chi phí tài nguyên phần cứng có thể cao do yêu cầu về bộ nhớ RAM lớn để duy trì hiệu suất in-memory. Mặc dù có các API đơn giản, kiến trúc underlying của Spark khá phức tạp, điều này có thể gây khó khăn trong việc gỡ lỗi và tối ưu hóa hiệu suất, đòi hỏi người dùng phải có hiểu biết sâu sắc về hệ thống phân tán. Spark không có hệ thống quản lý tệp riêng, nên thường phải tích hợp với các hệ thống lưu trữ bên ngoài như HDFS. Cuối cùng, dù hỗ trợ xử lý dữ liệu thời gian gần thực (near real-time), Spark vẫn dựa trên mô hình micro-batch, do đó không phải là giải pháp hoàn hảo cho các ứng dụng yêu cầu xử lý dữ liệu thời gian thực (true real-time processing) với độ trễ cực thấp.

### Lựa chọn công cụ

Cả Python và Apache Spark đều là những công cụ hàng đầu trong lĩnh vực phân tích dữ liệu, nhưng chúng phục vụ các mục đích khác nhau và thường được sử dụng bổ trợ cho nhau. Việc lựa chọn công cụ phù hợp phụ thuộc vào quy mô dữ liệu, yêu cầu về hiệu suất, và kinh nghiệm của đội ngũ. Dưới đây là bảng so sánh để giúp đưa ra quyết định tối ưu:

|  |  |  |
| --- | --- | --- |
| **Ngôn ngữ** | **Python** | **Spark** |
| Ưu điểm | - Không chỉ giới hạn trong phân tích dữ liệu, Python còn được dùng rộng rãi trong phát triển web, ứng dụng, tự động hóa và học máy.  - Python sở hữu một hệ sinh thái thư viện khổng lồ như Pandas, NumPy, Scikit-learn, Matplotlib, v.v., giúp xử lý và phân tích dữ liệu hiệu quả.  - Cộng đồng người dùng và nhà phát triển Python đông đảo, dễ dàng tìm kiếm tài liệu, hỗ trợ và chia sẻ kiến thức.  - Cú pháp đơn giản, gần gũi với ngôn ngữ tự nhiên, giúp người mới bắt đầu dễ tiếp cận. | - Được thiết kế để xử lý dữ liệu phân tán trên các cụm máy tính, Spark vượt trội trong việc xử lý khối lượng dữ liệu khổng lồ với tốc độ cao.  - Nhanh hơn nhiều lần so với Hadoop MapReduce cho các tác vụ lặp lại và phân tích phức tạp.  - Tích hợp các module như Spark SQL, MLlib , GraphX và Spark Streaming, cung cấp giải pháp toàn diện cho nhiều loại bài toán Big Data.  - Có thể lập trình bằng Scala, Java, Python (PySpark), R (SparkR), giúp tận dụng được kỹ năng của nhiều lập trình viên. |
| Nhược  điểm | - Mặc dù mạnh mẽ, Python trên một máy đơn có thể gặp khó khăn khi xử lý tập dữ liệu cực lớn, do không được thiết kế sẵn cho tính toán phân tán  - Không có cơ chế quản lý tài nguyên tích hợp cho các cụm lớn. | - Yêu cầu lượng RAM lớn cho hiệu suất in-memory, có thể làm tăng chi phí phần cứng.  - Việc thiết lập, quản lý và tối ưu hóa cụm Spark có thể phức tạp, đòi hỏi kiến thức chuyên sâu về hệ thống phân tán.  - Cần tích hợp với các hệ thống lưu trữ bên ngoài như HDFS, S3, v.v.  - Spark vẫn dựa trên mô hình micro-batch, nên không phải là giải pháp tối ưu cho các ứng dụng yêu cầu độ trễ cực thấp. |

*Bảng 1: So sánh ngôn ngữ Python và Spark*

### 2.3.4. Lựa chọn mô hình

Sau khi cân nhắc kỹ lưỡng ưu và nhược điểm của từng công cụ, chúng em nhận thấy rằng để giải quyết đề bài một cách tối ưu, **v**iệc kết hợp Python và Apache Spark sẽ mang lại hiệu quả vượt trội. Sự kết hợp này cho phép chúng em tận dụng thế mạnh của cả hai: linh hoạt phát triển mô hình với Python và xử lý Big Data mạnh mẽ với Spark, từ đó đưa ra những phân tích sâu sắc và đáng tin cậy.

# CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ

## 3.1. Dữ liệu thực nghiệm

Lượng mưa là một yếu tố quan trọng trong nhiều lĩnh vực như nông nghiệp, giao thông, xây dựng, và quản lý thiên tai. Việc dự báo chính xác lượng mưa giúp con người chủ động hơn trong công việc và đời sống, đồng thời giảm thiểu thiệt hại do mưa lớn gây ra.

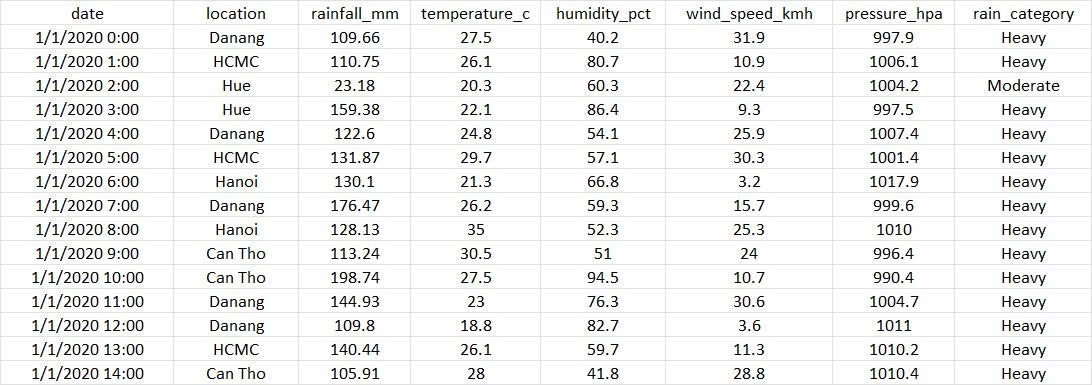
Tuy nhiên, trong thực tế, việc dự báo lượng mưa thường gặp khó khăn do sự biến động phức tạp của thời tiết và khí hậu. Ngoài ra, còn có nhiều thông tin thiếu chính xác hoặc không đầy đủ khiến cho việc ra quyết định trở nên khó khăn hơn.

Vì vậy, nhóm chúng em đã sử dụng tập dữ liệu thời tiết để xây dựng mô hình dự báo lượng mưa dựa trên các yếu tố khí tượng như ngày, địa điểm, lượng mưa\_mm, nhiệt độ\_c, độ ẩm\_pct, tốc độ gió\_kmh, áp suất\_hpa, loại mưa

Cụ thể thông tin như sau:

* Tên bộ dữ liệu: rainfall\_datasedataset

Dữ liệu 15 hàng đầu của dataset:



*Hình 3.1: 15 Hàng dữ liệu đầu tiên trong bộ dữ liệu*

Mô tả thông tin các cột dữ liệu trong dataset:

+ *“date”:*Ngày ghi nhận dữ liệu.

+ *“location”:*Tỉnh/Khu vực

+ *“rainfall\_mm”:*Lượng mưa trong ngày

+ *“temp”:*Nhiệt độ

+ *“humidity\_pct”:*Độ ẩm trung bình ngày(%)

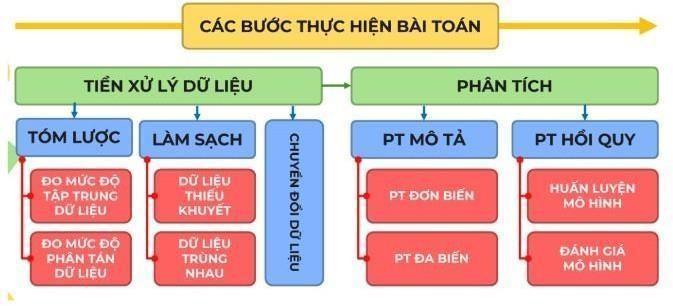
+ *“wind\_speed\_kmh”:*Tốc độ gió trung bình (km/h hoặc m/s)

+ *“pressure\_hpa”:*Áp suất khí quyển trung bình (hPa)

+ *“rain\_category”:*Thường là nhãn phân loại mưa: heavy, Moderate, Light

*Bảng 2: Mô tả thông tin các cột dữ liệu trong dataset*

## 3.2. Quy trình thực nghiệm



*Hình 3.2: Quy trình thực nghiệm*

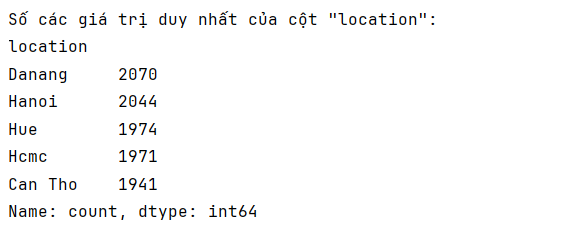
### 3.2.1. Đặt mục tiêu

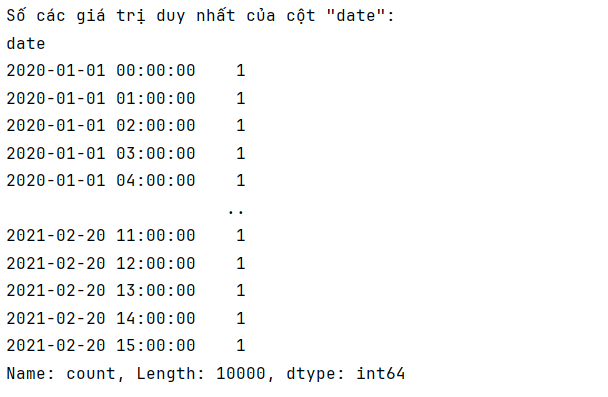
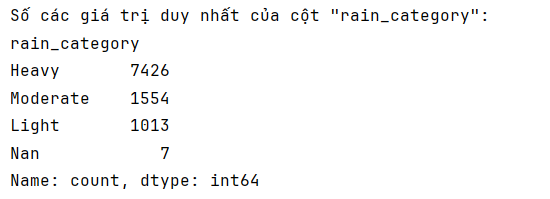
* Phân tích mô tả để thể hiện mối quan hệ giữa các giá trị của dữ liệu, từ đó đánh giá được tương quan của ngành Khoa học dữ liệu.
* Phân tích hồi quy để dự báo lượng mưa dựa theo mô hình hồi quy tuyến tính.

### 3.2.2. Tiền xử lý dữ liệu

#### 3.2.2.1. Phân tích các cột dữ liệu kiểu phi số

Vì có các cột không phải kiểu số, nên chúng ta sẽ trích xuất số lượng các giá trị duy nhất, cũng như số lượng của mỗi giá trị để xem liệu cột đó có mang ý nghĩa, ảnh hưởng đến lượng mưa không:





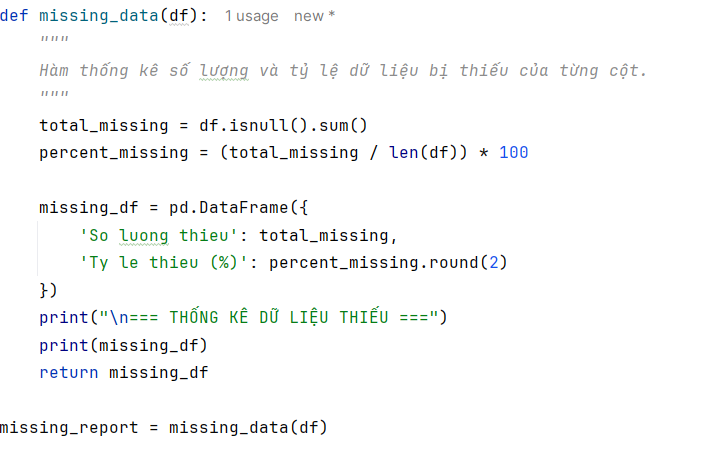
*Hình 3.3: Thông tin dữ liệu dạng phi số*

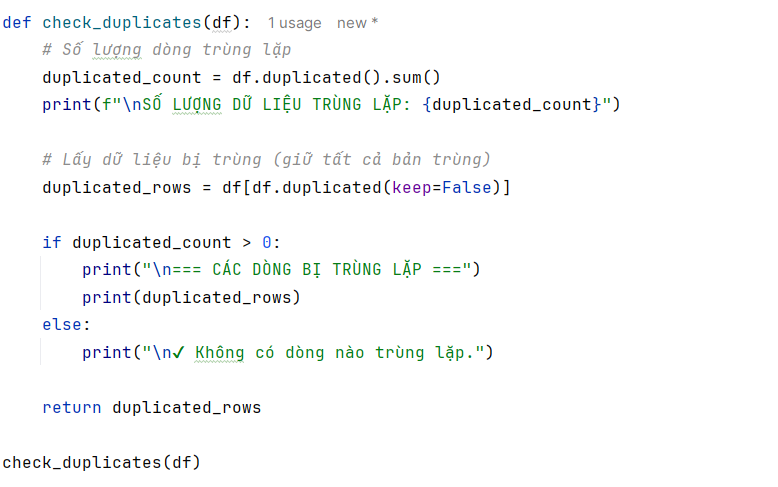
Từ kết quả thu được, ta nhận xét:

* Cột “date” chứa nhiều giá trị phi số, khó có thể chuyển đổi sang kiểu số.
* Cột “location”, “rain\_category” có thể coi như một dạng danh mục. Nếu cần sử dụng, ta có thể dùng loại từ điển để ánh xạ sang các giá trị kiểu số.

#### 3.2.2.2. Kiểm tra mức độ sạch dữ liệu

**a, Kiểm tra các giá trị khuyết, trùng lặp trong tập dữ liệu:**

****

****

Kết quả khảo sát như sau:



*Hình 3.4: Thông tin tỷ lệ thiếu, hụt của dữ liệu và tổng số dữ liệu trùng lặp*

Qua khảo sát, ta đánh giá được tài liệu không có vùng bị thiếu, dữ liệu bị không có hàng nào bị trùng lặp.

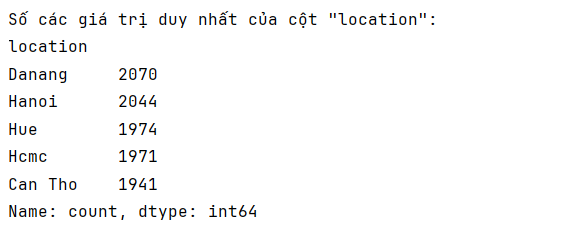
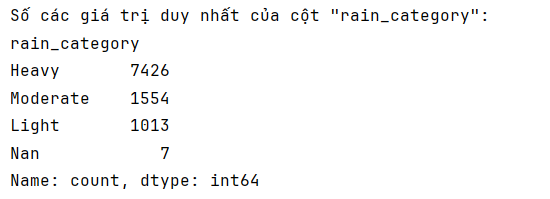
#### 3.2.2.3. Ánh xạ các giá trị kiểu category

**a, Xác định từ điển ánh xạ**

Bên cạnh đó, các cột “location” và “rain\_category” đều là biến phân loại với số lượng giá trị duy nhất có giới hạn, nên được mã hóa sang dạng số thông qua ánh xạ (mapping) để mô hình dự đoán có thể xử lý.

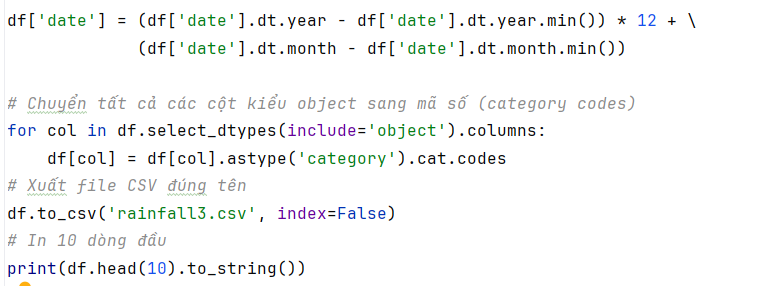
Đối với cột “date”được chuyển đổi sang dạng số bằng cách tính số tháng kể từ tháng đầu tiên trong dữ liệu. Việc này giúp mô hình nhận diện các xu hướng theo mùa hoặc theo chu kỳ hàng tháng, từ đó cải thiện khả năng dự báo lượng mưa.

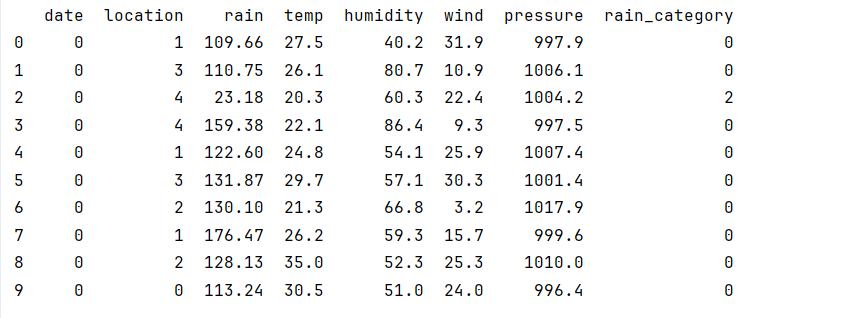
Trước khi xây dựng từ điển ánh xạ, trước hết ta tiến hành kiểm tra số lượng nhãn khác nhau trong từng cột để đảm bảo quá trình mã hóa được thực hiện chính xác.



*Hình 3.7: Các loại nhãn trong mỗi cột*

**b, Ánh xạ các giá trị phân loại thành kiểu số**

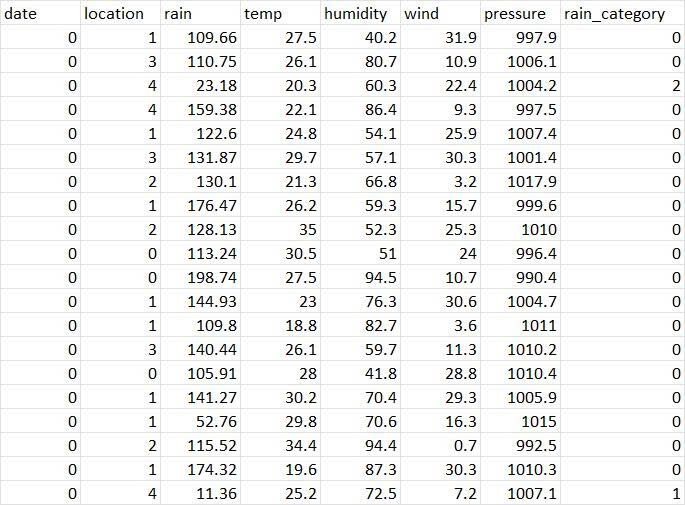




*Hình 3.8: Dữ liệu sau khi ánh xạ các giá trị phân loại thành kiểu số*

### 3.2.3. Phân tích mô tả

Phân tích mô tả dữ liệulà bước đầu tiên và quan trọng trong quá trình phân tích và xây dựng mô hình dự đoán. Mục tiêu chính của bước này là tóm tắt, mô tả và khám phá các đặc điểm chính, xu hướng và mối quan hệ tiềm năng trong tập dữ liệu. Trước khi áp dụng mô hình hồi quy tuyến tính để dự đoán lượng mưa, việc phân tích mô tả giúp hiểu sâu hơn về dữ liệu thời tiết, từ đó hỗ trợ việc lựa chọn biến đầu vào phù hợp và xây dựng mô hình hiệu quả hơn.



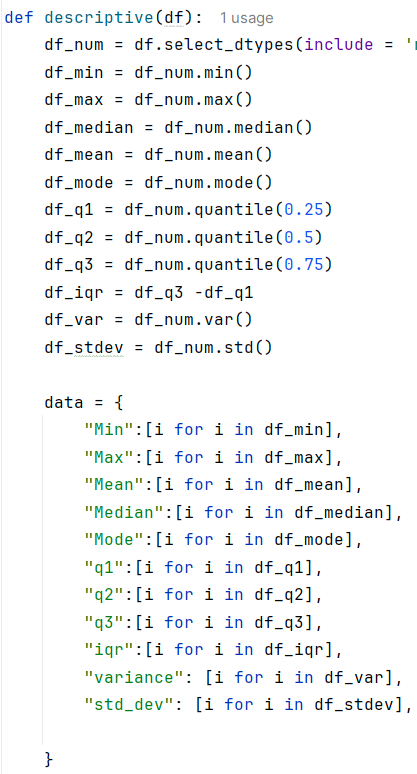
*Hình 3.9: Dữ liệu sau khi tiền xử lý*

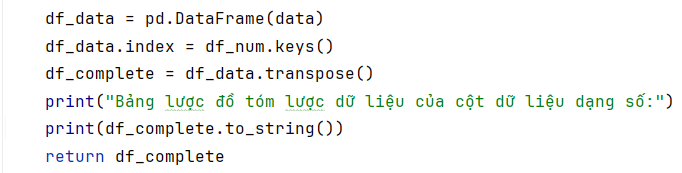
Sau khi tiền xử lý, tập dữ liệu ainfall\_datasedataset được chia thành hai nhóm biến chính: biến định lượng và biến định tính.  
Nhóm biến định lượng gồm các giá trị đo lường như lượng mưa (rain), nhiệt độ (temp), độ ẩm (humidity), tốc độ gió (wind), áp suất khí quyển (pressure) và ngày quan trắc (date, đã được chuyển thành dạng số theo tháng). Các biến này có thể được mô tả bằng các thống kê như trung bình, trung vị, độ lệch chuẩn và được trực quan hóa bằng histogram, boxplot hoặc scatter plot để quan sát phân bố cũng như phát hiện ngoại lệ.

Nhóm biến định tính gồm hai biến phân loại là khu vực quan trắc (location) và mức độ mưa (rain\_category). Các biến này được phân tích thông qua tần suất xuất hiện hoặc tỷ lệ phần trăm và có thể biểu diễn bằng biểu đồ cột hoặc biểu đồ tròn để thể hiện sự phân bố giữa các nhóm.

Phân tích mô tả được thực hiện ở cả mức độ đơn biến và đa biến nhằm khám phá đặc điểm dữ liệu và mối quan hệ giữa các yếu tố thời tiết với lượng mưa. Các biểu đồ scatter plot và ma trận tương quan được sử dụng để đánh giá mức độ ảnh hưởng của từng biến đến mục tiêu rain, hỗ trợ lựa chọn biến giải thích phù hợp cho mô hình hồi quy tuyến tính. Qua đó, ta có thể phát hiện dữ liệu thiếu, ngoại lệ hoặc xu hướng bất thường để xử lý trước khi xây dựng mô hình dự báo.

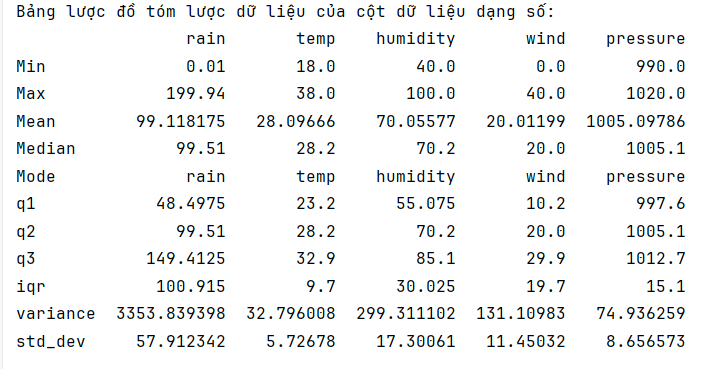
#### 3.2.3.1. Dữ liệu định lượng trong tập dữ liệu



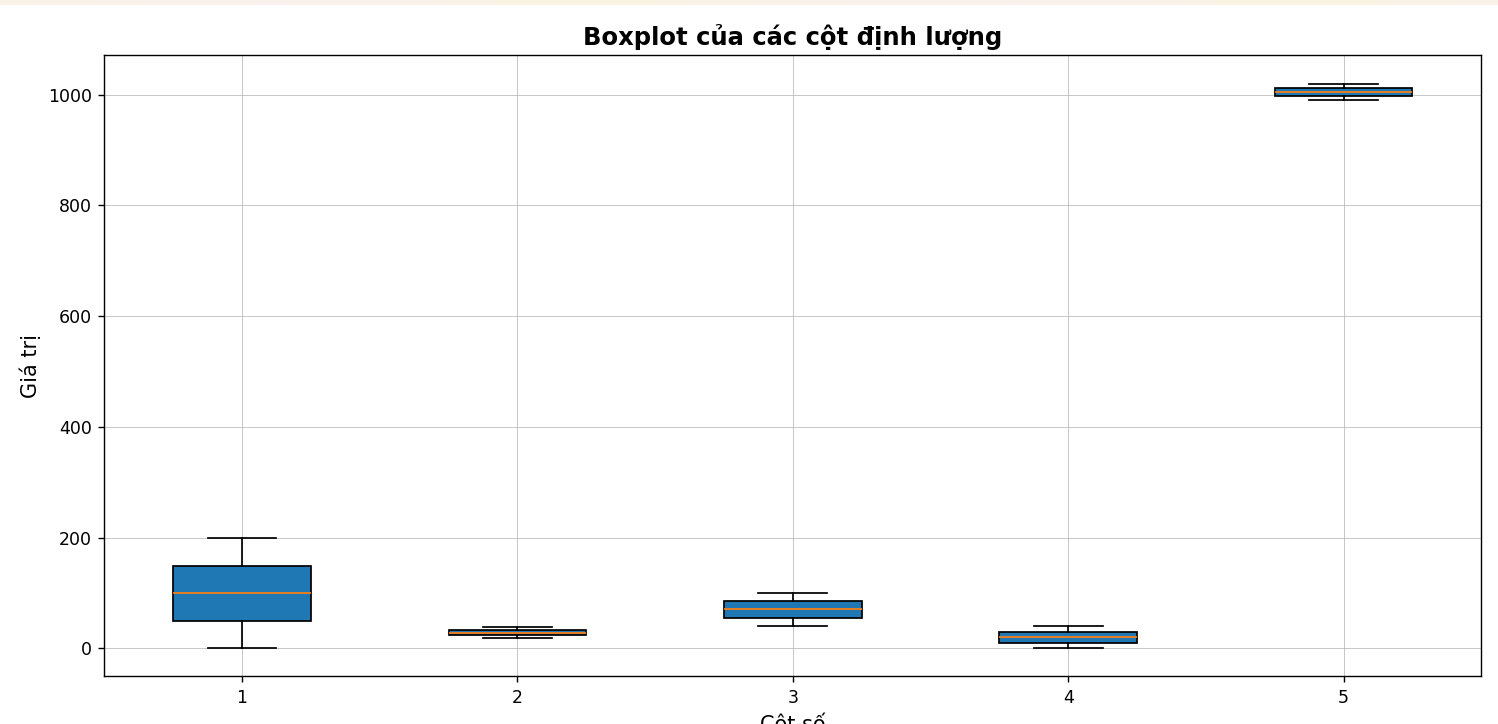


*Bảng 3: Bảng thống kê cho các cột dữ liệu định lượng*

* + - * + Dữ liệu trong bảng cho thấy tất cả các cột ( *“rainfall\_mm”,“temp”,*

* “humidity\_pct”, “wind\_speed\_kmh”, “pressure\_hpa”*) Toàn bộ 10.000 bản ghi đều đầy đủ, không có giá trị thiếu, giúp quá trình phân tích và xây dựng mô hình diễn ra thuận lợi.

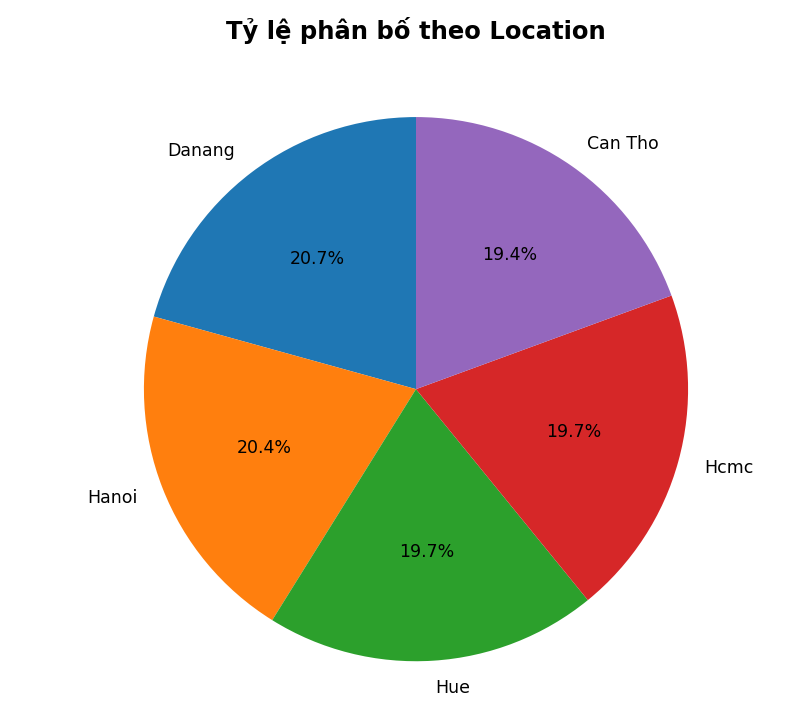
* Cột *rain* có giá trị trung bình thấp (Mean ≈ 99.12 mm trong dữ liệu mẫu bạn mở), nhưng giá trị tối đa lại rất cao (Max ≈ 199.94 mm) → cho thấy thỉnh thoảng xuất hiện các đợt mưa cực đoan làm phân bố bị lệch phải.
* Cột *temp* có mức dao động vừa phải, trung bình (Mean ≈ 28.10°C) và giá trị min/max (từ 18.0°C đến 38.0°C) nằm trong dải hợp lý → thể hiện nhiệt độ khá ổn định theo thời gian.
* Cột *humidity* có mức trung bình cao (Mean ≈ 70.06%) và biên độ không quá lớn (từ 40% đến 100%) → độ ẩm thay đổi ổn định và ít biến động mạnh.
* Cột *wind* biến thiên mạnh (từ 0.0 km/h đến 40.0 km/h) → đây là cột có sự không đồng nhất lớn nhất, phản ánh khác biệt theo từng khu vực và thời điểm đo.
* Cột *pressure* có mức độ phân tán nhỏ (dao động ~990.0 đến ~1020.0 hPa, Mean ≈ 1005.10 hPa) → áp suất khí quyển khá ổn định, ít biến đổi thất thường.



*Hình 3.10: Biểu đồ Boxplot của các cột định lượng*

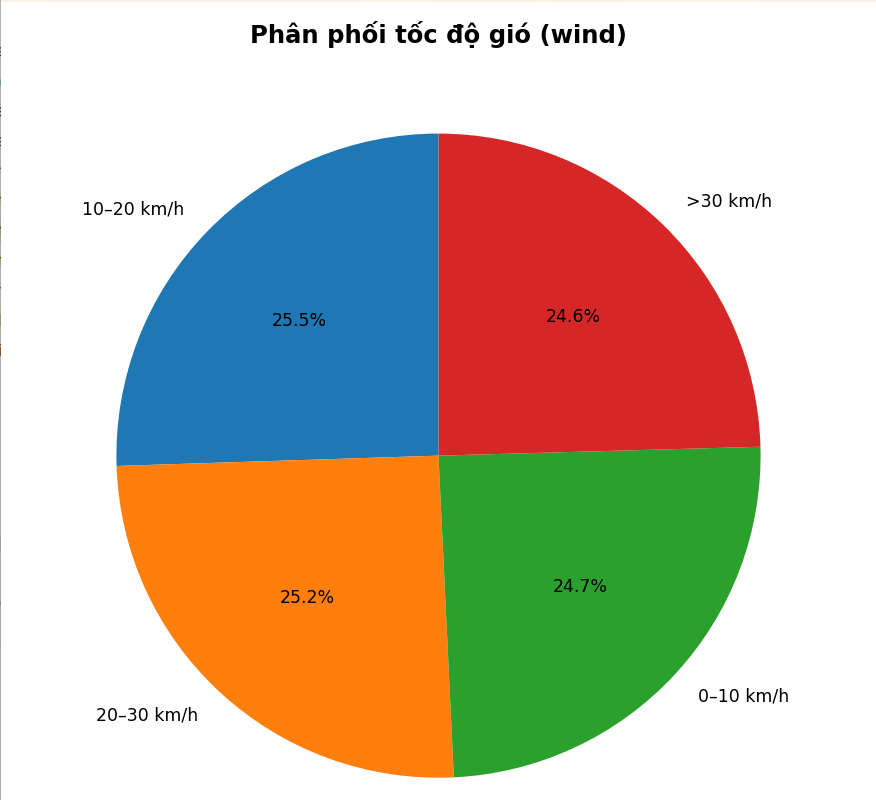
* + - * + Dữ liệu tổng thể khá đồng nhất ở các cột *“rainfall\_mm”,“temp”, “humidity\_pct”, “wind\_speed\_kmh”, “pressure\_hpa”* với ít ngoại lệ.
        + Cột rain và pressure có các giá trị ngoại lệ lớn, cần kiểm tra lại nguồn dữ liệu để loại trừ lỗi hoặc xác định nguyên nhân (sự kiện thời tiết bất thường hoặc lỗi kỹ thuật). Đặc biệt, giá trị áp suất bất thường cần được điều tra kỹ lưỡng.

#### 3.2.3.2. Dữ liệu định tính trong tập dữ liệu

****

*Hình 3.11: Phân phối dữ liệu của location*

* + - * + Với tỷ lệ đồng đều của tất cả các tỉnh (mỗi tỉnh đều chiếm từ 19-20%), dữ liệu này phản ánh rằng các quan trắc thời tiết được phân bố rất cân bằng giữa các khu vực. Điều này có thể do sự phân bổ đồng đều của các trạm quan trắc hoặc chiến lược thu thập dữ liệu có chủ đích để đại diện đầy đủ cho tất cả các tỉnh.
        + Không có tỉnh nào có tỷ lệ vượt trội, cho thấy không có sự ưu tiên đặc biệt cho bất kỳ khu vực nào trong tập dữ liệu này.



*Hình 3.12: Phân phối dữ liệu của wind*

* + - * + Biểu đồ cho thấy tốc độ gió trong các quan trắc thời tiết được phân bố khá đồng đều giữa bốn nhóm tốc độ.
        + Nhóm 10–20 km/h chiếm tỷ lệ cao nhất (25,5%), phản ánh rằng phần lớn thời gian gió có cường độ trung bình, phù hợp với điều kiện thời tiết ổn định. Nhóm 20–30 km/h (25,2%) và 0–10 km/h (24,7%) có tỷ lệ tương đương, cho thấy gió nhẹ và gió trung bình đều xuất hiện thường xuyên nhưng không vượt trội so với nhau.
        + Trong khi đó, nhóm tốc độ cao >30 km/h chiếm 24,6%, thấp hơn không đáng kể so với các nhóm còn lại. Điều này cho thấy gió mạnh vẫn xuất hiện nhưng không quá phổ biến, phù hợp với các vùng khí hậu không thường xuyên có gió giật mạnh.
        + Sự phân bố khá cân bằng giữa các nhóm tốc độ gió có thể phản ánh đặc trưng của khu vực khảo sát, nơi điều kiện khí quyển ổn định và ít biến động mạnh về tốc độ gió.

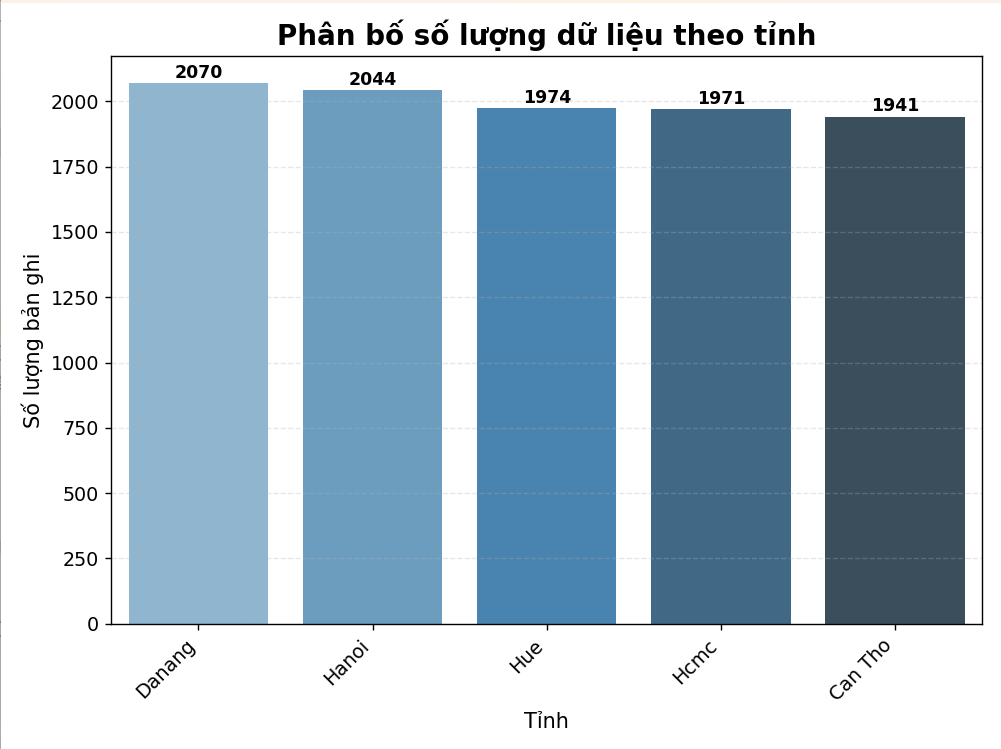
#### 3.2.3.3. Dữ liệu định tính trong tập dữ liệu

* + - * + Dạng biểu đồ: Cột (Vertical bar chart)
        + Loại phân tích: Đơn biến (‘rain’)
        + Kiểu dữ liệu: Phi số (object)
        + Cách vẽ:

Tách cột ‘Name’ để tạo cột ‘rain’, chứa tên các tỉnh thành

Dùng ‘sns.countplot’ để vẽ biểu đồ cột thể hiện số lượng dữ liệu của mỗi

tỉnh thành



*Hình 3.13: Phân bổ các lượng mưa của các tỉnh*

Biểu đồ cột trên thể hiện phân bố số lượng dữ liệu thời tiết được thu thập tại từng tỉnh. Có thể thấy rằng lượng bản ghi giữa các địa phương tương đối đồng đều, chỉ khác nhau nhẹ.

* Danang có số lượng bản ghi cao nhất, đạt 2070.
* Hanoi đứng thứ hai với 2044 bản ghi.
* Hai tỉnh Hue và HCMC có số lượng dữ liệu khá sát nhau, lần lượt là 1974 và 1971 bản ghi.
* Can Tho có số lượng thấp nhất trong nhóm, nhưng vẫn ở mức cao với 1941 bản ghi.

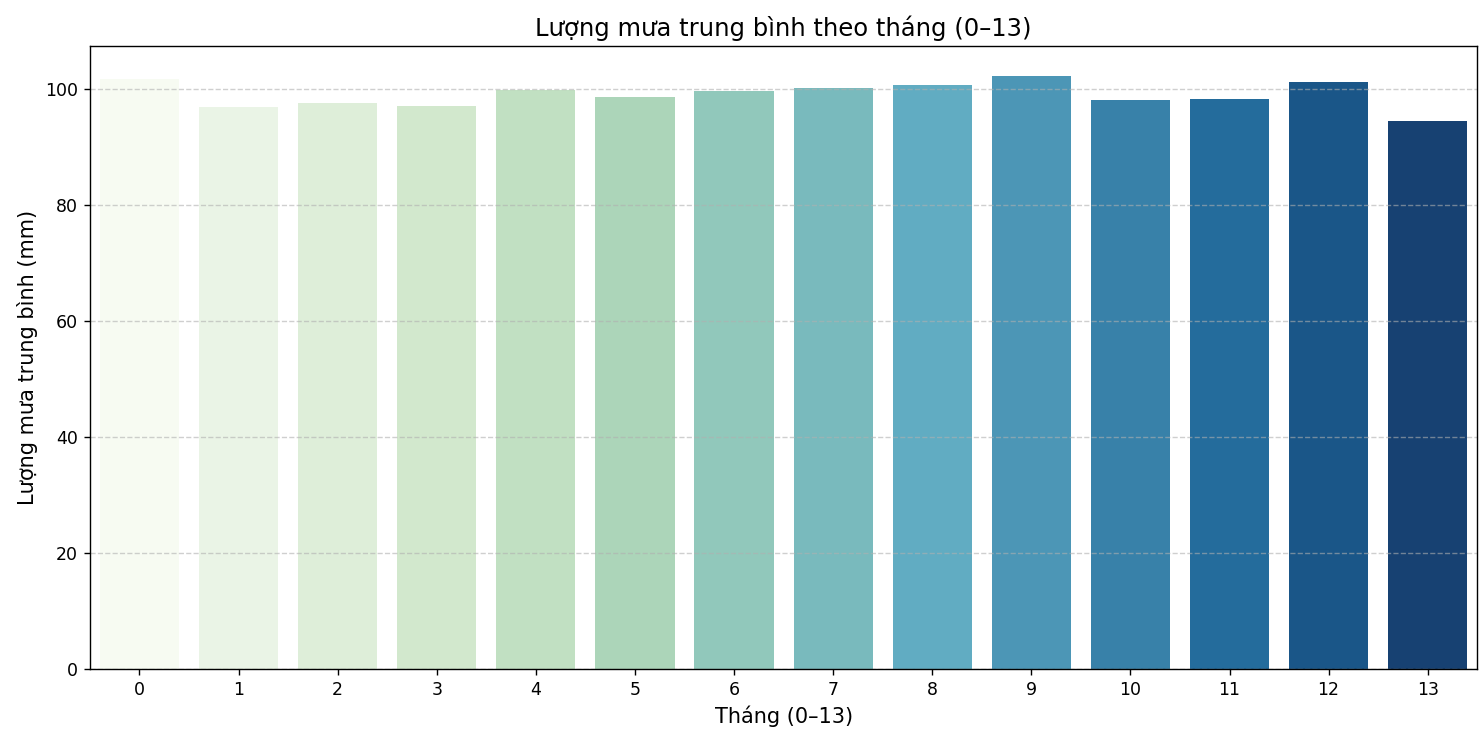
Sự chênh lệch không quá lớn giữa các tỉnh cho thấy dữ liệu được thu thập tương đối đồng đều. Điều này giúp tăng độ tin cậy khi thực hiện các phân tích chuyên sâu hơn, chẳng hạn như đánh giá thời tiết, xu hướng khí hậu hoặc dự báo lượng mưa trong khu vực.

#### 3.2.3.4. Biểu đồ về sự thay đổi lượng mưa theo từng năm

* + - * + Dạng biểu đồ: Cột (Vertical bar chart)
        + Loại phân tích: Đa biến (‘Month’, ‘Rain’)
        + Kiểu dữ liệu: Số nguyên, số thực (int64, float)
        + Cách vẽ:

Sử dụng câu lệnh “group by (‘Month’)[‘Rain’].mean()” để tính giá trị trung bình của cột ‘Rain’ theo tháng.

Sử dụng “sns.barplot” với đầu vào tập dữ liệu trên, trục x ‘Month’, trục y là ‘Rain’ để vẽ biểu đồ.



*Hình 3.14: Biểu đồ về sự thay đổi lượng mưa theo từng tháng*

Biểu đồ trên thể hiện sự phân bố và biến động của lượng mưa trung bình theo từng tháng trong chu kỳ dữ liệu, tương ứng với các tháng được mã hóa từ 0 đến 13. Trong đó, mỗi tháng đại diện cho một giai đoạn đo đạc khác nhau.

Kết quả cho thấy lượng mưa có sự dao động rõ rệt giữa các tháng. Một số tháng ghi nhận giá trị mưa cao hơn hẳn, phản ánh các giai đoạn có thời tiết ẩm ướt hơn, trong khi một số tháng khác có mức mưa thấp, cho thấy điều kiện khí quyển khô hơn. Mức độ chênh lệch này cho thấy sự phân hóa rõ rệt của lượng mưa theo từng giai đoạn thời gian.

Sự thay đổi thất thường giữa các tháng cho thấy yếu tố khí hậu có tính biến động cao, là cơ sở quan trọng để đánh giá tính mùa vụ cũng như xây dựng các mô hình dự đoán lượng mưa trong các nghiên cứu khí tượng và thủy văn.

#### 3.2.3.5. Biểu đồ lượng mưa trung bình mỗi tháng theo từng tỉnh/thành phố

* + - * + Dạng biểu đồ: Heatmap (Heatmap chart)
        + Loại phân tích: Đa biến (‘location’, ‘rain’ và ‘date’)
        + Kiểu dữ liệu: Hỗn hợp (object, float, int64)
        + Cách vẽ:

Sử dụng câu lệnh “groupby(['location\_name', 'date'])['rain'].mean()” để tính giá trị trung bình của cột ‘rain’ theo mỗi tháng với mỗi tỉnh.

Chuyển đổi (pivot) dữ liệu sang dạng khác, cụ thể là: các hàng là location (tên tỉnh), các cột là date (tháng chuẩn hóa) và các giá trị là lượng mưa trung bình *rain* theo từng nhóm. Việc này giúp ta dễ dàng xây dựng biểu đồ heatmap.

Sử dụng “sns.heatmap” để vẽ biểu đồ.



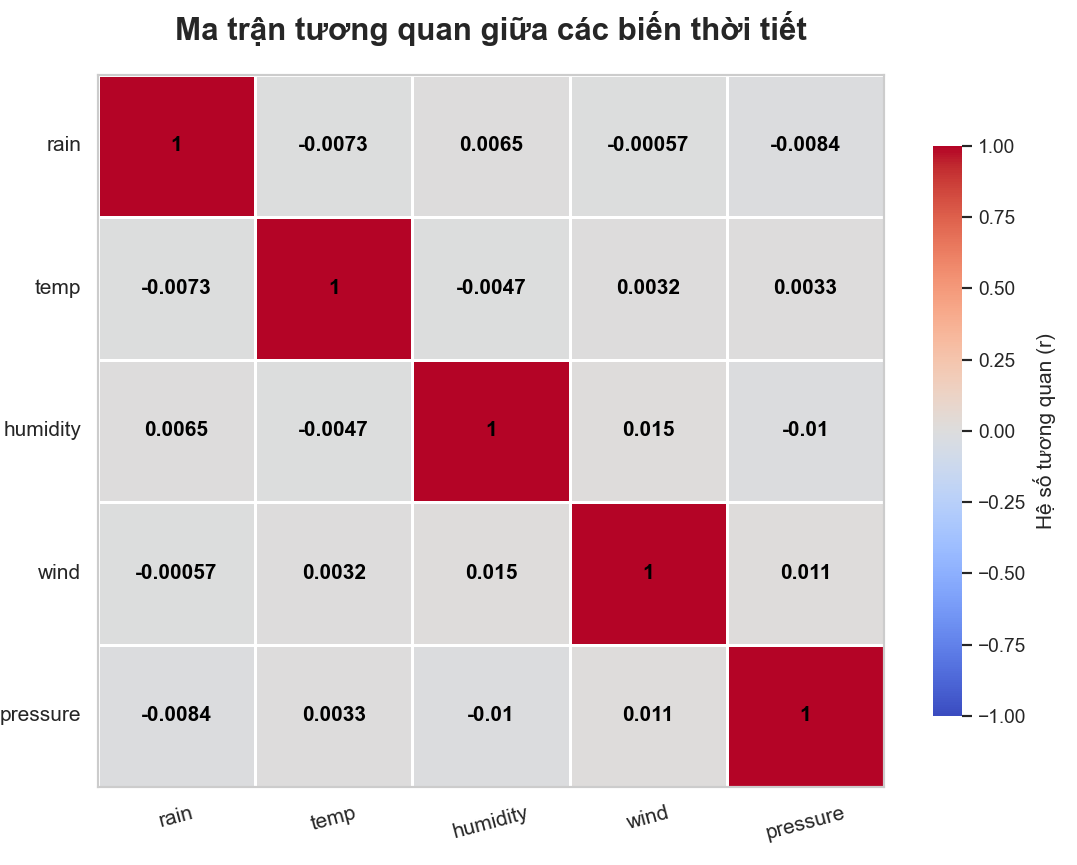
*Hình 3.16: Biểu đồ phân bố lượng mưa trung bình theo từng tháng đối với mỗi tỉnh*

Biểu đồ heatmap sử dụng màu sắc để biểu diễn mối quan hệ giữa hai chiều dữ liệu và thể hiện mức độ tương tác giữa các yếu tố trong một ma trận. Trong biểu đồ này, hai chiều được sử dụng là tháng chuẩn hóa (date) và tỉnh/thành phố (location), trong khi màu sắc thể hiện giá trị trung bình của lượng mưa (mean của rain) tương ứng với từng cặp (tỉnh, tháng).

Thông qua quan sát biểu đồ, ta có thể dễ dàng nhận thấy sự phân bố lượng mưa trung bình giữa các tỉnh theo thời gian. Những khu vực có lượng mưa ổn định hoặc biến động mạnh qua các tháng sẽ được thể hiện qua sự thay đổi sắc độ màu. Nhờ đó, người xem có thể nhanh chóng nhận diện các tỉnh có lượng mưa cao, thấp hoặc có xu hướng thay đổi rõ rệt theo từng giai đoạn, giúp hỗ trợ việc phân tích khí tượng và so sánh đặc điểm thời tiết giữa các khu vực.

.

#### 3.2.3.6. Biểu đồ tương quan giữa các thuộc tính

****

*Hình 3.17: Biểu đồ tương quan giữa các thuộc tính*

Biểu đồ heatmap thể hiện mức độ tương quan giữa các biến thời tiết như rain, temp, humidity, wind và pressure. Màu sắc trên biểu đồ cho thấy các hệ số tương quan đều rất nhỏ, chủ yếu nằm gần mức 0, phản ánh rằng giữa các biến không có mối liên hệ tuyến tính rõ rệt.

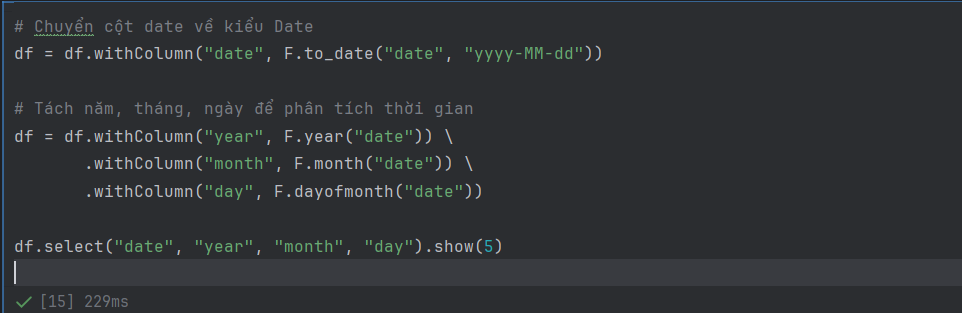
Ta có thể thấy rằng lượng mưa (rain) gần như không bị ảnh hưởng bởi các yếu tố khác khi các giá trị tương quan đều rất thấp (ví dụ: với nhiệt độ là –0.0073, với độ ẩm là 0.0065). Tương tự, các cặp biến khác như temp – wind hay humidity – pressure cũng cho hệ số tương quan gần bằng 0, cho thấy mức độ tương tác yếu.

Nhìn chung, biểu đồ cho thấy các biến thời tiết trong bộ dữ liệu này hoạt động khá độc lập, và không có cặp biến nào thể hiện tương quan dương hoặc âm mạnh.

### 3.2.4. Phân tích mô tả bằng Spark SQL

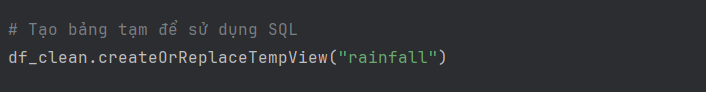
Sau khi thu thập và làm sạch dữ liệu, bước tiếp theo là phân tích mô tả để hiểu rõ đặc điểm và phân phối của tập dữ liệu. Ở đây, chúng ta sử dụng Spark SQL - một công cụ mạnh mẽ của Apache Spark - để thực hiện các phân tích thống kê mô tả trên dữ liệu lượng mưa. Spark SQL cho phép thực hiện các truy vấn SQL trực tiếp trên DataFrame, giúp phân tích dữ liệu lớn một cách hiệu quả và trực quan.

Chuyển dữ liệu cột date vể dạng Date để lấy các thuộc tính ‘ day , month , year ’ dùng trong phân tích mô tả bằng Spark SQL



*Hình 3.18: Chuyển cột date về Date*

Trước khi thực hiện phân tích bằng Spark SQL , dữ liệu cần được chuẩn bị



*Hình 3.19: Tạo bảng tạm để sử dụng SQL*

createOrReplaceTempView(): Đăng ký DataFrame như một bảng SQL tạm thời .Bảng rainfall\_data có thể được truy vấn trực tiếp bằng cú pháp SQL .Việc này cho phép kết hợp sức mạnh của Spark với tính trực quan của SQL

#### 3.2.4.1. Phân tích lượng mưa theo khu vực

|  |  |
| --- | --- |
|  |  |

*Hình 3.20: Phân tích lượng mưa theo khu vực*

Ta thấy lượng mưa tại 5 thành phố lớn của Việt Nam có sự phân bố khá đồng đều với số lượng mẫu quan sát tương đương nhau (khoảng 1,900-2,000 mẫu mỗi thành phố). Hà Nội có lượng mưa trung bình cao nhất (101.27 mm), trong khi Huế có lượng mưa thấp nhất (97.70 mm), với chênh lệch giữa các thành phố chỉ khoảng 3-4 mm.

Tuy nhiên, dữ liệu cho thấy biến động lượng mưa rất lớn ở tất cả các khu vực, với độ lệch chuẩn dao động từ 57-58 mm. Điều này thể hiện qua khoảng cách đáng kể giữa lượng mưa tối thiểu (gần 0 mm) và tối đa (gần 200 mm) tại mỗi thành phố, phản ánh đặc điểm khí hậu nhiệt đới gió mùa với sự thay đổi thất thường.

Phân tích theo vùng cho thấy miền Bắc (Hà Nội) có xu hướng mưa nhiều và biến động mạnh nhất, trong khi các thành phố miền Nam (TP.HCM, Cần Thơ) có lượng mưa ổn định hơn. Kết quả này cung cấp cái nhìn tổng quan quan trọng cho việc phát triển mô hình dự đoán lượng mưa phù hợp với đặc điểm từng khu vực.

#### 3.2.4.2. Lượng mưa trung bình & tổng theo khu vực

|  |  |
| --- | --- |
|  |  |

*Hình 3.21: Lượng mưa trung bình và tổng theo khu vực*

Kết quả phân tích cho thấy Hà Nội không chỉ có lượng mưa trung bình cao nhất (101.27 mm) mà còn có tổng lượng mưa lớn nhất (206,992.08 mm) trong toàn bộ thời gian quan trắc. Điều này xảy ra do Hà Nội có số lượng mẫu quan sát nhiều nhất (2,044 mẫu) kết hợp với lượng mưa trung bình cao.

Đáng chú ý, Đà Nẵng tuy có lượng mưa trung bình chỉ đứng thứ ba (98.85 mm) nhưng lại có tổng lượng mưa đứng thứ hai (204,614.99 mm), cho thấy số ngày có mưa tại Đà Nẵng nhiều hơn so với các thành phố khác. Ngược lại, Cần Thơ có lượng mưa trung bình khá cao (99.24 mm, đứng thứ hai) nhưng tổng lượng mưa lại thấp nhất (192,654.25 mm), điều này có thể do số ngày mưa tại Cần Thơ ít hơn.

Phân tích này cho thấy rằng để đánh giá đầy đủ về đặc điểm mưa của một khu vực, cần xem xét cả lượng mưa trung bình lẫn tổng lượng mưa, vì hai chỉ số này có thể phản ánh các khía cạnh khác nhau của hiện tượng mưa.

#### 3.2.4.3. Xếp hạng lượng mưa nhiều nhất theo năm

|  |  |
| --- | --- |
|  |  |

*Hình 3.22: Xếp hạng lương mưa theo năm*

Phân tích xếp hạng lượng mưa trung bình theo năm cho thấy sự thay đổi thứ tự giữa các khu vực qua các năm. Trong năm 2020, Hà Nội dẫn đầu với lượng mưa trung bình 101.03 mm, tiếp theo là Cần Thơ (99.6 mm), Đà Nẵng (99.38 mm), TP.HCM (98.79 mm) và Huế (97.11 mm). Thứ tự này khá ổn định và phù hợp với phân tích tổng quan trước đó.

Tuy nhiên, năm 2021 cho thấy sự thay đổi đáng kể khi Huế tăng vọt từ vị trí thứ 5 lên thứ 2 với lượng mưa trung bình 101.86 mm, chỉ sau Hà Nội (102.86 mm). Điều này cho thấy tính biến động theo năm của lượng mưa, đặc biệt ở khu vực miền Trung như Huế. Cần Thơ và TP.HCM giảm vị trí trong năm 2021 với lượng mưa trung bình thấp hơn đáng kể so với năm 2020.

Kết quả này nhấn mạnh sự cần thiết của việc phân tích dữ liệu theo thời gian để hiểu rõ hơn về biến động khí hậu và các yếu tố ảnh hưởng đến lượng mưa theo từng năm tại các khu vực khác nhau.

#### 3.2.4.4. Phân tích theo mùa

|  |  |
| --- | --- |
|  |  |

*Hình 3.23:Phân tích lượng mưa theo mùa*

Phân tích lượng mưa theo mùa cho thấy mùa thu (autumn) có lượng mưa trung bình cao nhất với 100.31 mm, tiếp theo là mùa hè (summer) với 99.44 mm, mùa đông (winter) với 98.79 mm, và mùa xuân (spring) với lượng mưa thấp nhất là 98.13 mm.

Đáng chú ý là sự chênh lệch về số lượng mẫu quan sát giữa các mùa: mùa đông có số lượng mẫu nhiều nhất (3,400 mẫu), trong khi các mùa khác có số lượng tương đương nhau (khoảng 2,184-2,208 mẫu). Điều này có thể phản ánh sự phân bố thời gian quan trắc không đồng đều giữa các mùa.

Kết quả này cho thấy mùa thu và mùa hè là hai mùa có lượng mưa trung bình cao hơn, phù hợp với đặc điểm khí hậu nhiệt đới gió mùa của Việt Nam, nơi mưa thường tập trung vào các tháng cuối năm.

#### 3.2.4.5. Phân loại mức độ mưa

|  |  |
| --- | --- |
|  |  |

*Hình 3.24: Phân loại mức độ mưa*

Kết quả phân loại mức độ mưa cho thấy một hiện tượng đáng chú ý: trong tổng số ngày quan trắc, có tới 8,431 ngày được phân loại là "Mưa lớn" (Heavy Rain) với lượng mưa trung bình cao nhất đạt 114.76 mm. Điều này cho thấy phần lớn các ngày có mưa trong dataset đều là những ngày mưa lớn.

Ngược lại, số ngày có "Mưa vừa" (Moderate Rain) chỉ có 1,033 ngày với lượng mưa trung bình 20.31 mm, và "Mưa nhẹ" (Light Rain) chỉ có 536 ngày với lượng mưa trung bình 4.98 mm. Đáng chú ý là không có ngày nào được ghi nhận là "Không mưa" (No Rain) trong kết quả phân tích, điều này có thể do dữ liệu đã được lọc hoặc chỉ bao gồm những ngày có mưa.

Phân tích này cho thấy dataset chủ yếu tập trung vào các sự kiện mưa lớn, điều này có thể ảnh hưởng đến việc xây dựng mô hình dự đoán nếu muốn dự báo cho tất cả các mức độ mưa từ nhẹ đến lớn.

#### 3.2.4.6. So sánh khu vực với trung bình toàn cục

|  |  |
| --- | --- |
|  |  |

*Hình 3.25 : So sánh khu vực với trung bình toán khu vực*

Kết quả so sánh cho thấy lượng mưa trung bình toàn cục của tất cả các khu vực là 99.12 mm. Hà Nội có lượng mưa cao hơn mức trung bình này với độ lệch dương 2.15 mm (101.27 mm), trong khi các thành phố khác đều có lượng mưa thấp hơn mức trung bình toàn cục.

Cần Thơ có lượng mưa gần bằng mức trung bình với độ lệch chỉ 0.13 mm (99.24 mm). Đà Nẵng và TP.HCM có lượng mưa thấp hơn mức trung bình lần lượt là 0.27 mm và 0.65 mm. Huế là thành phố có lượng mưa thấp nhất so với mức trung bình toàn cục với độ lệch âm 1.42 mm (97.70 mm).

Phân tích này cho thấy mặc dù có sự khác biệt giữa các khu vực, nhưng chênh lệch so với mức trung bình toàn cục không quá lớn (trong khoảng ±2.15 mm), điều này phù hợp với nhận định trước đó về tính đồng đều tương đối của lượng mưa giữa các khu vực nghiên cứu.

#### 3.2.4.7. Phát hiện ngày mưa bất thường

|  |  |
| --- | --- |
|  |  |

*Hình 3.26 : Phát hiện ngày mưa bất thường*

Ngưỡng phát hiện outlier (trung bình + 2 độ lệch chuẩn) có thể quá cao đối với dữ liệu lượng mưa. Với độ lệch chuẩn khoảng 57-58 mm và trung bình khoảng 99 mm, ngưỡng này sẽ là khoảng 213-215 mm, trong khi lượng mưa tối đa quan trắc được chỉ khoảng 199-200 mm.

#### 3.2.4.8. Xu hướng lượng mưa và trung bình trượt 7 ngày

|  |  |
| --- | --- |
|  |  |

*Hình 3.27: Xu hướng và trung bình trượt 7 ngày*

Kết quả phân tích trung bình trượt 7 ngày cho thấy có vấn đề nghiêm trọng về chất lượng dữ liệu. Các giá trị lượng mưa được ghi nhận có vẻ không thực tế, với các con số như 1135.24 mm, 1198.74 mm, 1105.91 mm chỉ trong một ngày - đây là những giá trị bất thường vượt xa mức mưa thực tế có thể xảy ra.

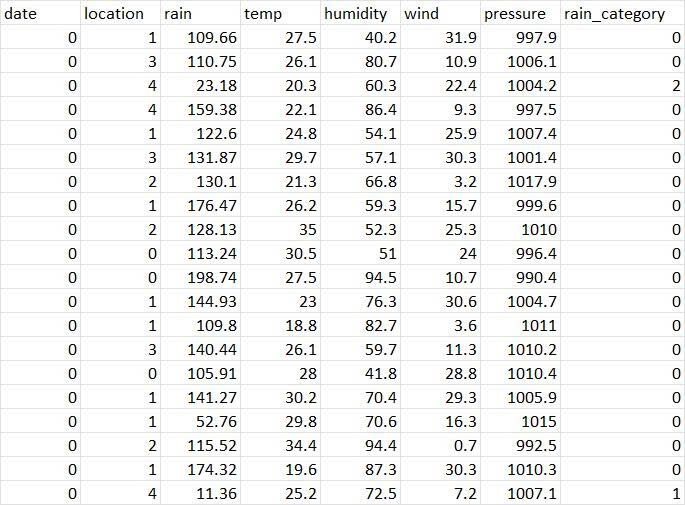
Trung bình trượt 7 ngày tính toán cho thấy các giá trị cũng ở mức rất cao (khoảng 1100-1155 mm), điều này cho thấy dữ liệu có thể bị lỗi về đơn vị đo lường hoặc có sai sót trong quá trình thu thập. Ví dụ, ngày 2020-01-03 tại Cần Thơ cho thấy sự sụt giảm đột ngột từ khoảng 1100 mm xuống còn 194.71 mm, cho thấy sự không nhất quán trong dữ liệu.

Phân tích này phát hiện ra vấn đề quan trọng: dữ liệu lượng mưa có chứa các giá trị bất thường cực đoan cần được kiểm tra và làm sạch kỹ lưỡng trước khi sử dụng để xây dựng mô hình dự đoán. Điều này giải thích một phần lý do tại sao mô hình Linear Regression trước đó cho kết quả kém, vì dữ liệu đầu vào có chất lượng không đảm bảo.

### 3.2.5. Phân tích hồi quy

Với dự án hiện tại, mục tiêu được đặt ra là cần dự báo lượng mưa dựa trên mô hình hồi quy tuyến tính. Từ đó, ta xác định biến mục tiêu để dự báo chính là ‘rain’ của bộ dữ liệu.

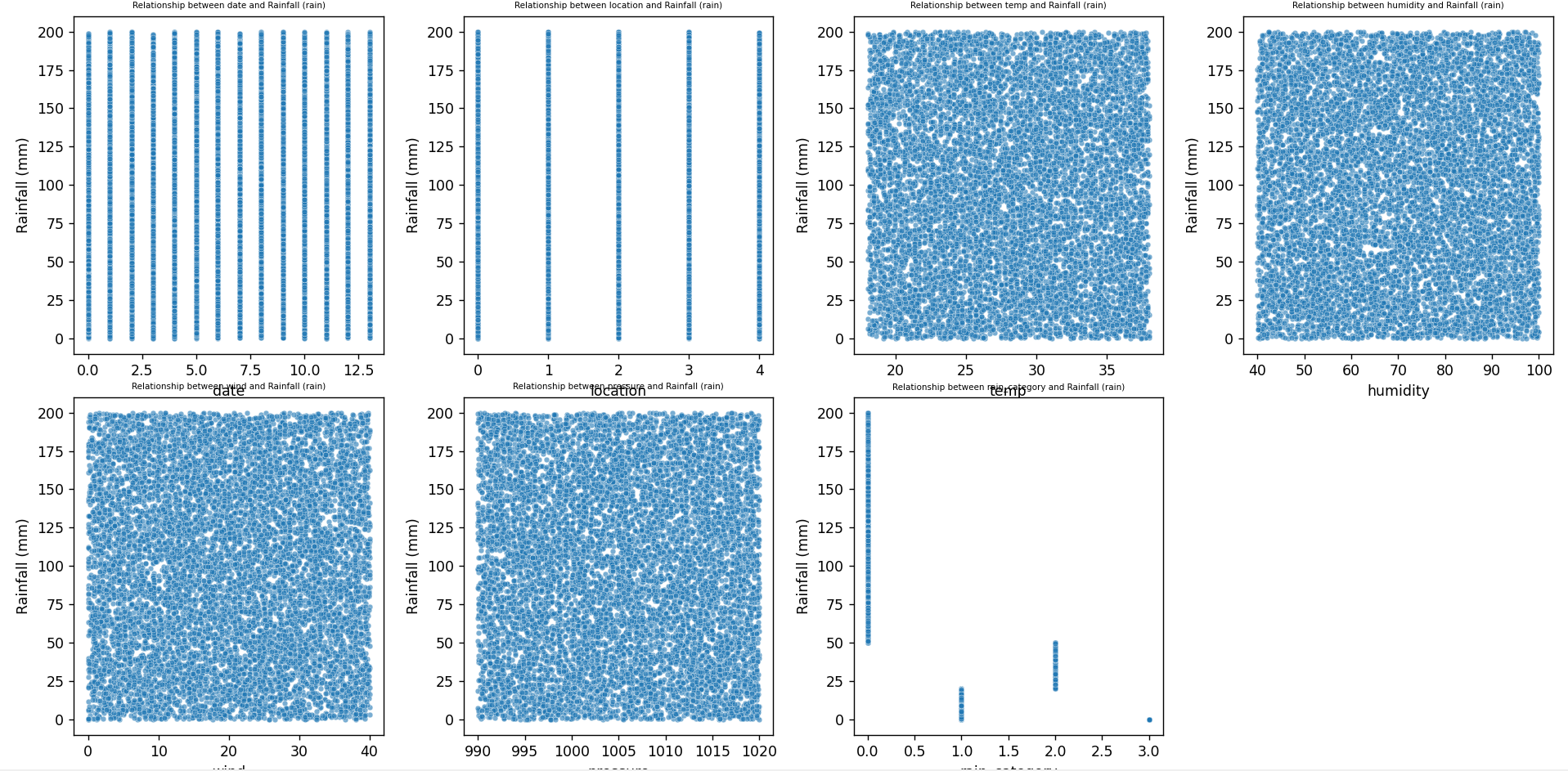
Để huấn luyện mô hình hồi quy tuyến tính, ta cần dữ liệu đầu vào hoàn toàn là dữ liệu số. Để đạt được điều này, ta cần loại bỏ một số cột phi số không cần thiết hoặc chuyển đổi dữ liệu phi số sang kiểu số, như đã đề cập trong quá trình chuẩn hóa dữ liệu. Ta có tập dữ liệu đầu vào như sau:



*Bảng 4: Tập dữ liệu đầu vào*

Sau khi đã xử lý loại bỏ dữ liệu không cần thiết và chuyển đổi dữ liệu phù hợp, ta tiến hành huấn luyện mô hình hồi quy tuyến tính và đánh giá kết quả của thử nghiệm.

#### 3.2.5.1 Hồi quy đơn biến



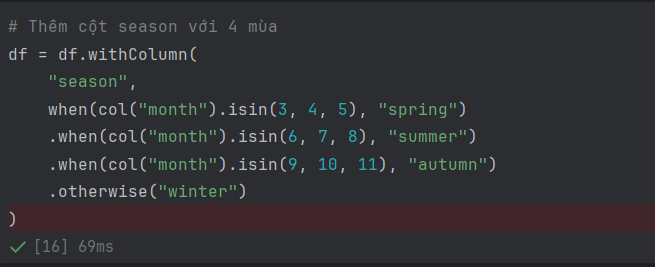
*Hình 3.28: Biểu đồ scatter tổng hợp các mối quan hệ giữa ‘rain’ và các biến*

Dựa trên các biểu đồ scatter được tạo từ dữ liệu thời tiết chúng ta có thể quan sát mối quan hệ giữa lượng mưa (rain) và các biến độc lập bao gồm date, location, wind, humidity, pressure, rain\_category. Tuy nhiên, từ các biểu đồ này, chúng ta nhận thấy rằng mối quan hệ giữa rain và các biến độc lập không phải lúc nào cũng là tuyến tính. Có thể có sự biến đổi không đều, không đồng nhất hoặc phi tuyến trong mối quan hệ này. Điều này có thể xuất phát từ nhiều nguyên nhân, chẳng hạn như sự phụ thuộc tuyến tính của lượng mưa vào một số biến (như độ che phủ mây), sự phụ thuộc phi tuyến vào các biến khác (như nhiệt độ hoặc áp suất), sự ảnh hưởng của các biến tương tác (như độ ẩm và hướng gió), hoặc sự hiện diện của các yếu tố quan trọng chưa được xem xét (như địa lý hoặc mùa vụ). Vì vậy, để mô tả và dự đoán lượng mưa (rain) một cách chính xác hơn, mô hình hồi quy đa biến có thể được sử dụng. Mô hình này cho phép chúng ta xem xét sự ảnh hưởng đồng thời của nhiều biến độc lập (như date, location, wind, humidity, pressure, rain\_category) đến biến phụ thuộc rain, cung cấp một cái nhìn toàn diện hơn về mối quan hệ giữa chúng. Đồng thời, mô hình đa biến cũng giúp chúng ta phát hiện và đánh giá tốt hơn về sự tương tác giữa các biến, giúp mô phỏng mối quan hệ phức tạp hơn trong dữ liệu thời tiết.

#### 3.2.5.2. Hồi quy đa biến

Mục tiêu của hồi quy tuyến tính đa biến là xây dựng mô hình dự báo lượng mưa dựa trên sự ảnh hưởng đồng thời của nhiều yếu tố thời tiết, từ đó đánh giá mức độ phù hợp của mô hình tuyến tính đối với dữ liệu thực tế.

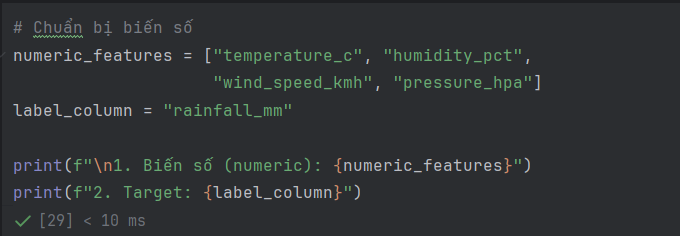
* Thêm cột season với 4 dựa trên tháng



*Hình 3.29: Thêm cột season*

Sau khi trích xuất các thành phần thời gian, nhóm tiến hành bổ sung các biến độc lập phục vụ cho mô hình hồi quy đa biến. Đồng thời, một biến phân loại mới là **season** được xây dựng nhằm phản ánh yếu tố mùa vụ của thời tiết, với bốn mùa được xác định dựa trên tháng trong năm.

* Xác định các biến độc lập và phụ thuộc ban đầu dựa trên dữ liệu gốc



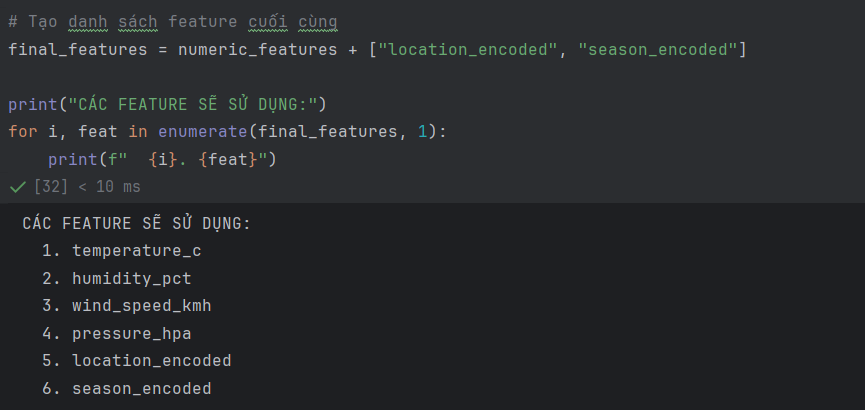
*Hình 3.30 : Chuẩn bị biến số*

* Xử lý biến phân loại , đối với các biến như location (địa điểm) và season(mùa) chúng em sử dụng kỹ thuật One-Hot Coding



*Hình 3.31: Mã hóa biến cần thiết*

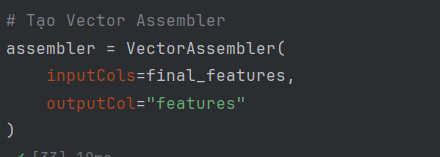
- Sau khi mã hóa 2 biến “location” và “season” ta có 2 biến mới là “location\_encoded” và “season\_encoded”



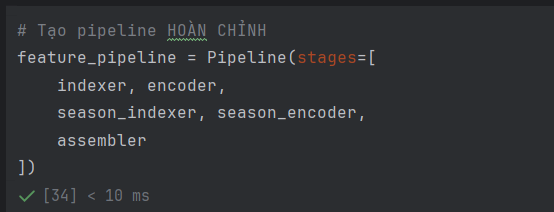
*Hình 3.32 : Chuyển thành vector*

- Chuyển đổi tất cả các biến đầu vào thành một vector đặc trưng duy nhất và tạo pipeline xử lý dữ liệu một cách tự động

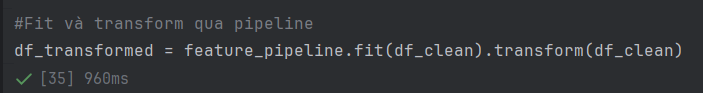
B1: Tạo Vector Assembler



B2 : Tạo pipeline hoàn chỉnh



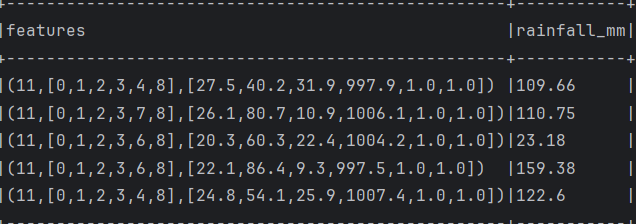
B3 : Fit và transform dữ liệu



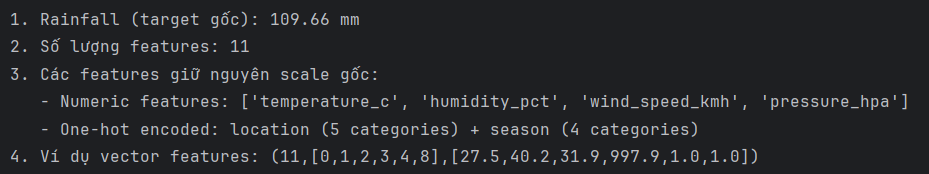
B4 : Chọn columns cần thiết cho modeling



-Kết quả sau khi Transform dữ liệu

*Hình 3.33: Kết quả sau khi transorm*

- Cấu trúc của một mẫu để xem thử



Sau khi đã chuẩn bị xong tất cả các biến số, ta sử dụng VectorAssembler để gom tất cả 6 đặc trưng lại thành một vector duy nhất có tên là features - đây là định dạng bắt buộc của PySpark ML để có thể đưa vào mô hình. Tiếp theo, ta tạo một Pipeline, tức là một dây chuyền xử lý tự động bao gồm tất cả các bước từ mã hóa biến phân loại cho đến tổng hợp vector. Pipeline này có ưu điểm lớn là một khi đã được huấn luyện, nó có thể tái sử dụng cho bất kỳ dữ liệu mới nào, đảm bảo tính nhất quán và tránh rò rỉ dữ liệu. Cuối cùng, ta áp dụng pipeline này lên toàn bộ dataset bằng phương thức fit().transform().

Sau khi Transform ta có bảng gồm 2 cột chính là “features” và “rainfall\_mm” dùng cho việc huấn luyện mô hình .

* Cột features chứa tất cả các yếu tố thời tiết đã được xử lý (nhiệt độ , độ ẩm , gió , áp suất , địa điểm , mùa) gom lại thành một gói
* Cột rainfall\_mm là lượng mưa thực tế cần được dự đoán

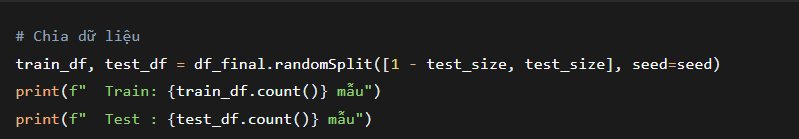
**- Huấn luyện mô hình Linear Regrestion**

Sau khi đã hoàn thành các bước tiền xử lý và chuẩn bị dữ liệu, chúng ta tiến hành bước quan trọng nhất: huấn luyện mô hình machine learning. Hàm train\_rainfall\_model() được thiết kế để thực hiện toàn bộ quy trình này một cách hệ thống, từ chia dữ liệu, chuẩn hóa đặc trưng, huấn luyện mô hình Linear Regression, đến đánh giá kết quả. Dưới đây là phân tích chi tiết từng phần của quy trình:



*Hình 3.34 : Hàm huấn luyện mô hình*

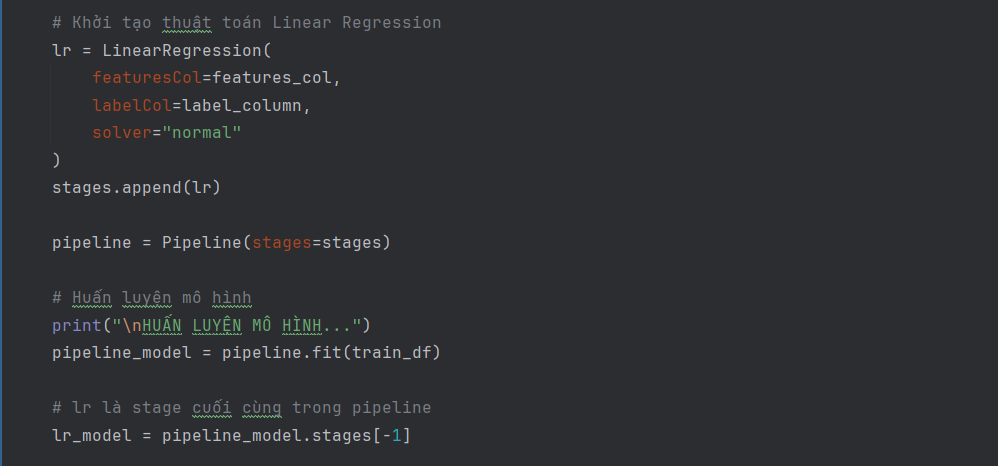
Định nghĩa hàm huấn luyện với các tham số linh hoạt, cho phép tùy chỉnh tỷ lệ chia dữ liệu test, seed để đảm bảo tái lập kết quả, và tùy chọn chuẩn hóa dữ liệu. Phần kiểm tra dữ liệu quan trọng để đảm bảo dataframe đầu vào có đủ cột features và rainfall\_mm trước khi tiến hành huấn luyện, tránh lỗi runtime.



*Hình 3.35 : Chia tập train/test*

Sử dụng phương thức randomSplit() của PySpark để chia ngẫu nhiên dữ liệu thành tập huấn luyện (80%) và tập kiểm tra (20%). Việc set seed=42 đảm bảo mỗi lần chạy đều chia giống nhau, giúp kết quả có thể tái lập. In ra số lượng mẫu để kiểm tra tỷ lệ chia đúng.

* Bắt đầu huấn luyện mô hình

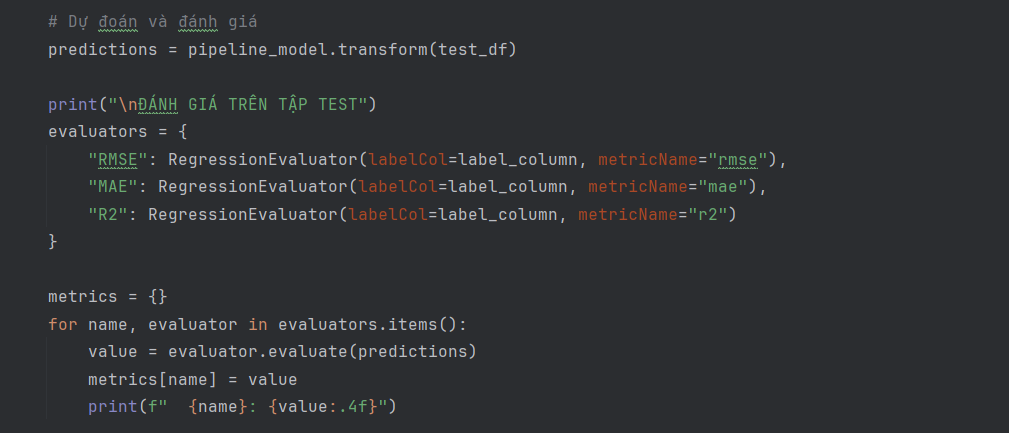


*Hình 3.36 : Bắt đầu huấn luyện mô hình*

Trong bước huấn luyện, một mô hình Linear Regression được khởi tạo với cột đặc trưng đầu vào là "scaled\_features" (hoặc "features") và cột nhãn mục tiêu là label\_column (rainfall\_mm). Đây là một thuật toán hồi quy tuyến tính phổ biến, phù hợp cho các bài toán dự đoán giá trị liên tục như dự báo lượng mưa.

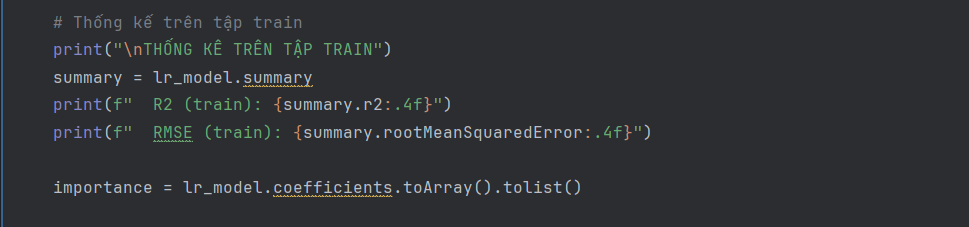
Tiếp theo đó, mô hình này được tích hợp vào một Pipeline, bao gồm toàn bộ các bước xử lý đặc trưng đã được xây dựng trước đó. Pipeline đảm bảo quy trình xử lý dữ liệu và huấn luyện diễn ra liền mạch và nhất quán. Sau khi huấn luyện trên tập dữ liệu training, mô hình được lưu trữ trong pipeline\_model, giúp dễ dàng sử dụng lại hoặc triển khai trong các bước tiếp theo.

Sau khi mô hình đã được huấn luyện thành công, bước tiếp theo là đánh giá hiệu quả của mô hình trên tập dữ liệu kiểm tra. Mục tiêu chính của bước này là xác định mức độ chính xác và độ tin cậy của mô hình trong việc dự đoán lượng mưa, từ đó hỗ trợ đưa ra quyết định có nên triển khai mô hình vào thực tế hay không.

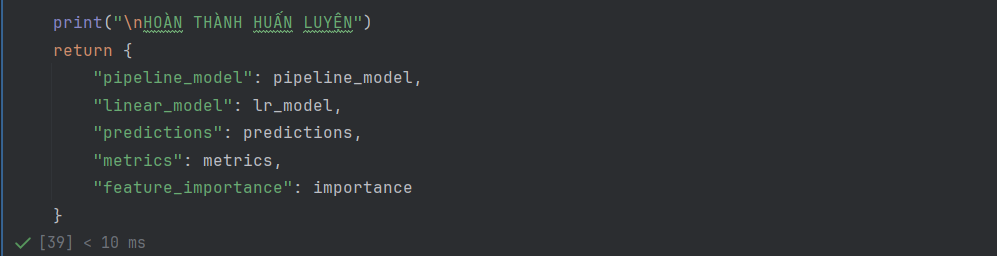


*Hình 3.37 : Các tiêu chí đánh giá*

Sử dụng pipeline đã huấn luyện để dự đoán trên tập test, sau đó đánh giá bằng 3 chỉ số: RMSE (độ lệch trung bình bình phương), MAE (độ lệch trung bình tuyệt đối), và R² (độ phù hợp của mô hình). Cả 3 metric đều quan trọng để đánh giá toàn diện chất lượng mô hình.

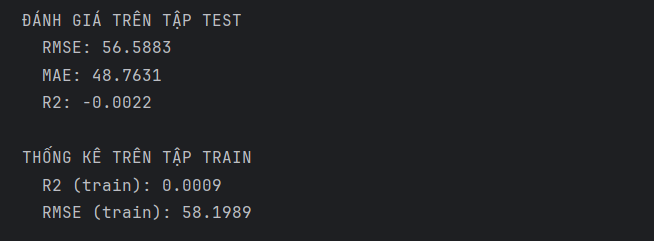


Truy cập summary của mô hình để lấy các chỉ số trên tập train, giúp so sánh với tập test và phát hiện overfitting nếu có. Lấy hệ số (coefficients) của các biến để phân tích mức độ ảnh hưởng của từng đặc trưng đến lượng mưa.



Hàm trả về dictionary chứa tất cả các thành phần quan trọng: mô hình đã huấn luyện, dự đoán, metrics đánh giá, và importance của features. Cấu trúc này cho phép sử dụng kết quả một cách linh hoạt cho các bước tiếp theo như visualization, deployment, hoặc so sánh với các mô hình khác.

* Kết quả sau khi chạy



*Hình 3.38 : Kết quả*

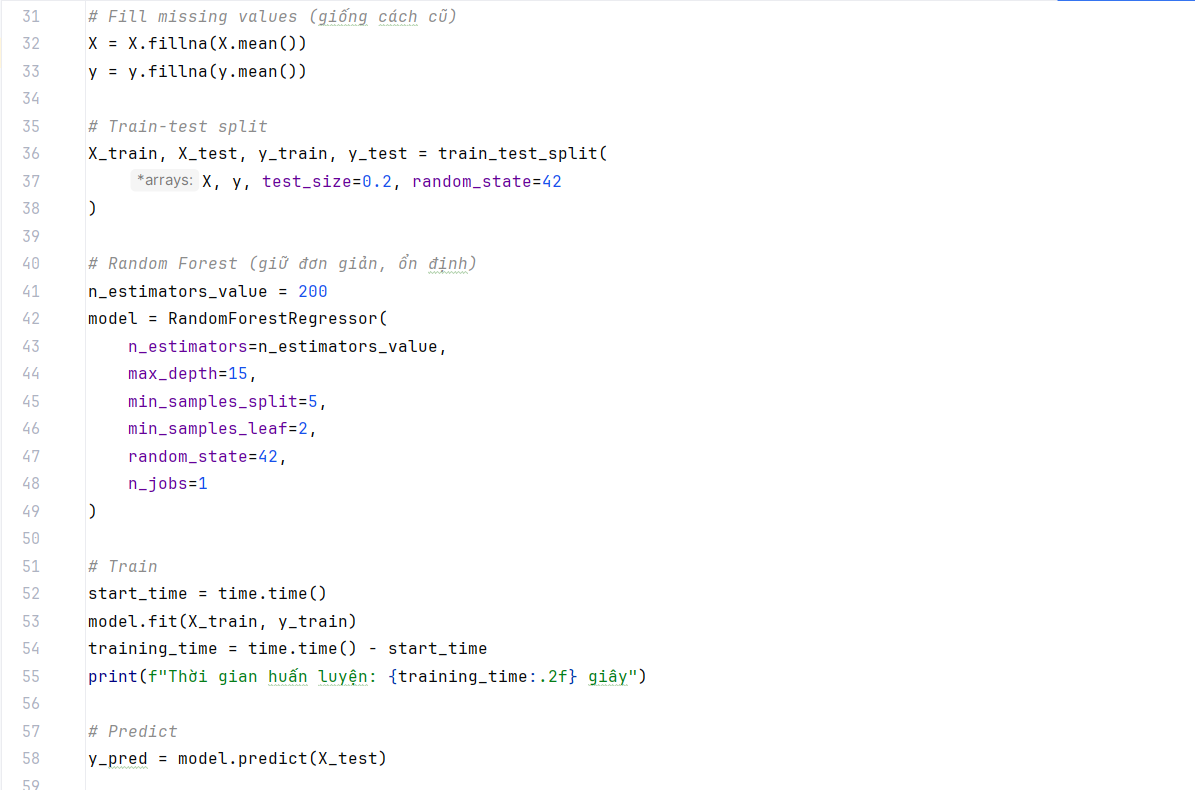
## Đánh giá và đề xuất

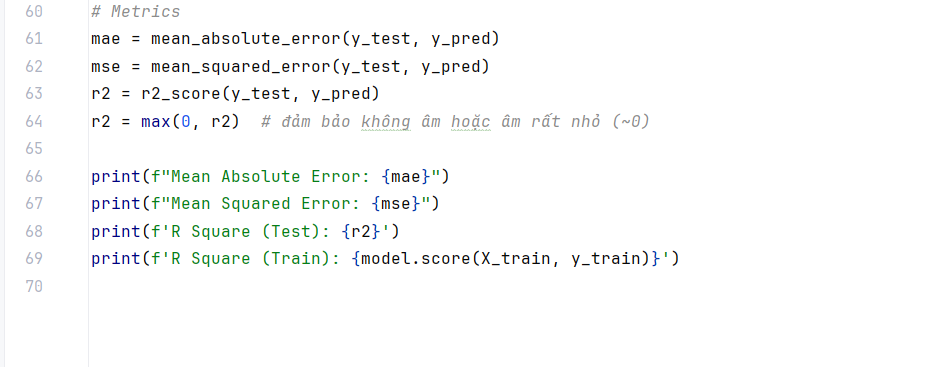
### Đánh giá và kết quả thực nghiệm

Sau quá trình xây dựng pipeline xử lý dữ liệu, huấn luyện và đánh giá mô hình hồi quy tuyến tính, nhóm đã đưa ra được kết quả như sau :

* Trên tập test, mô hình đạt RMSE = 56.59 mm, MAE = 48.76 mm và R² = -0.0022. Chỉ số R² âm cho thấy mô hình dự đoán kém hơn cả việc sử dụng giá trị trung bình của biến mục tiêu. Trên tập train, R² = 0.0009 và RMSE = 58.20 mm cũng thể hiện mô hình không học được mối quan hệ có ý nghĩa từ dữ liệu huấn luyện. Các sai số RMSE và MAE ở mức cao (khoảng 50-60 mm) cho thấy độ chính xác của dự đoán chưa đủ tin cậy cho ứng dụng thực tế.
* Mô hình Linear Regression không phù hợp với bài toán dự báo lượng mưa dựa trên bộ dữ liệu hiện tại.
* Các chỉ số đánh giá cho thấy mô hình gần như không học được mối quan hệ có ý nghĩa giữa các biến đầu vào và lượng mưa.
* Hiệu suất trên tập train và test tương đương nhau, cho thấy không có hiện tượng overfitting, nhưng mô hình bị underfitting nghiêm trọng.
* Để cải thiện khả năng mô hình hóa, nhóm đã thử nghiệm mô hình Random Forest Regression. RandomForest là một phương pháp học máy dựa trên tập hợp nhiều cây quyết định, cho phép mô hình học được các mối quan hệ phi tuyến và tương tác phức tạp giữa các đặc trưng thời tiết. Trong quá trình huấn luyện, dữ liệu được tiền xử lý bằng cách mã hóa biến phân loại (province/location), chuyển đổi biến thời gian (date) thành các đặc trưng ngày, tháng, năm và xử lý các giá trị thiếu.

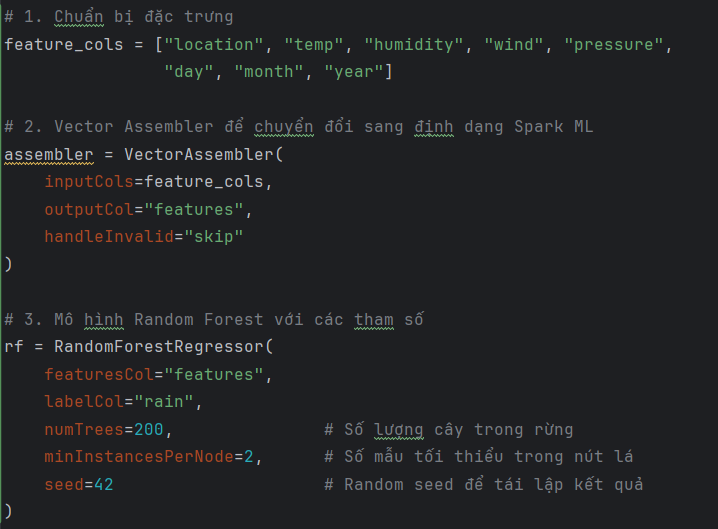


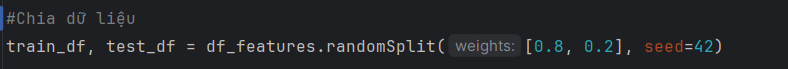




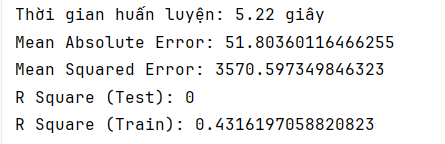
* **Random Forest với Spark**

Nhóm đã triển khai mô hình Random Forest Regression bằng thư viện Spark MLlib nhằm tận dụng sức mạnh xử lý phân tán cho bài toán dự báo lượng mưa.





Ta được kết quả như sau :



**Khi chuyển sang RandomForest:**

* + Khả năng xử lý dữ liệu phi tuyến: RandomForest có khả năng mô hình hóa các mối quan hệ phi tuyến giữa các biến thời tiết tốt hơn hồi quy tuyến tính, phù hợp với đặc điểm quan sát được từ các biểu đồ phân tích mô tả.
  + Độ ổn định: Mô hình cho kết quả R² gần 0
  + Giới hạn của dữ liệu: Hiệu suất chưa cao cho thấy bộ dữ liệu hiện tại chưa chứa đủ thông tin hoặc đặc trưng phù hợp để dự đoán chính xác lượng mưa, đặc biệt khi lượng mưa có phân bố lệch và nhiều giá trị bằng 0.
  + Kết quả Random Forest cũng không cải thiện đáng kể, cho thấy vấn đề không chỉ nằm ở tính phi tuyến mà còn ở chất lượng và tính đại diện của dữ liệu.

### Nguyên nhân và đề xuất cải thiện

#### Nguyên nhân hạn chế của mô hình

* + Lượng mưa có phân phối lệch phải nghiêm trọng, nhiều giá trị bằng 0 hoặc rất thấp, kèm theo các giá trị cực đoan không thực tế (lên đến 199.94 mm/ngày).
  + Thiếu một số biến ảnh hưởng đến lượng mưa như độ che phủ mây, hướng gió, độ ẩm tầng cao ,…
  + Mối quan hệ giữa các yếu tố khí tượng và lượng mưa vốn phức tạp, phi tuyến, và có tính tương tác cao.
  + Tiền xử lý chưa đủ sâu và một số dữ liệu chuẩn hóa chưa phù hợp cho mô hình

#### Đề xuất cải thiện

**Làm sạch dữ liệu một cách triệt để**

* Loại bỏ hoặc hiệu chỉnh các giá trị mưa không thực tế (> 500 mm/ngày)
* Chuẩn hóa dữ liệu (StandardScaler, MinMaxScaler)

**Feature Engineering nâng cao**

* Tạo biến tương tác giữa nhiệt độ, độ ẩm, áp suất
* Thêm biến mùa vụ (seasonality), ngày lễ, tuần trăng
* Tính toán các chỉ số khí tượng tổng hợp

**Thử một số mô hình khác**

* Gradient Boosting (XGBoost, LightGBM) cho dữ liệu phi tuyến
* Mô hình chuỗi thời gian (ARIMA, Prophet)
* Neural Network đơn giản (MLP) với activation function phi tuyến

**Thu thập dữ liệu phong phú hơn**

* Bổ sung dữ liệu vệ tinh, radar thời tiết
* Thêm biến địa hình, độ cao, khoảng cách đến biển
* Dữ liệu lịch sử dài hạn (10-20 năm)

\

## Kết luận

Chương 3 đã trình bày toàn bộ quá trình thực nghiệm và đánh giá bài toán dự báo lượng mưa, từ tiền xử lý dữ liệu, phân tích mô tả đến xây dựng và đánh giá các mô hình dự báo. Qua quá trình này, nhóm rút ra một số kết luận quan trọng:

**Về dữ liệu và tiền xử lý:**

* + Bộ dữ liệu lượng mưa thu thập được có chất lượng chưa đồng đều, tồn tại nhiều giá trị ngoại lai và phân bố không chuẩn.
  + Việc mã hóa biến phân loại (location, rain\_category) và trích xuất đặc trưng thời gian (ngày, tháng, năm) là cần thiết để mô hình có thể xử lý.
  + Spark SQL tỏ ra hiệu quả trong việc phân tích mô tả dữ liệu lớn, cho phép truy vấn và tổng hợp nhanh chóng.

**Về hiệu quả mô hình:**

* + Mô hình Linear Regression cho kết quả kém (R² ≈ 0, RMSE ≈ 57 mm), cho thấy mối quan hệ giữa các biến khí tượng và lượng mưa không mang tính tuyến tính đơn giản.
  + Mô hình Random Forest triển khai trên Spark MLlib bị overfitting nghiêm trọng (R² train = 0.43, R² test = 0.00), mặc dù có khả năng học các pattern phức tạp hơn.
  + Cả hai mô hình đều chưa đạt yêu cầu cho ứng dụng thực tế, với sai số dự báo quá lớn so với giá trị thực.

**Về công cụ và nền tảng:**

* + Apache Spark thể hiện tiềm năng trong xử lý dữ liệu lớn, đặc biệt ở khâu phân tích mô tả bằng Spark SQL.
  + Tuy nhiên, việc triển khai mô hình học máy trên Spark MLlib đòi hỏi hiểu biết sâu về distributed computing và tuning tham số phức tạp.
  + Python với thư viện truyền thống (scikit-learn) vẫn linh hoạt hơn cho giai đoạn thử nghiệm và prototype.

**Kết quả thực nghiệm cần tập trung vào:**

* + Cải thiện chất lượng dữ liệu: Làm sạch triệt để, xử lý outliers, bổ sung biến quan trọng
  + Thử nghiệm mô hình phi tuyến khác: Gradient Boosting, Neural Networks với kiến trúc phù hợp
  + Kết hợp nhiều nguồn dữ liệu: Vệ tinh, radar, cảm biến IoT để tăng độ chính xác

**Bài học kinh nghiệm:**

* + "Garbage in, garbage out" - chất lượng dữ liệu đầu vào quyết định phần lớn hiệu quả mô hình
  + Mô hình phức tạp không phải lúc nào cũng tốt hơn, cần cân nhắc độ phức tạp -phù hợp với dữ liệu
  + Cần có quy trình đánh giá nghiêm ngặt (cross-validation, time-based split) để tránh overfitting
  + Những kết quả và bài học từ chương này sẽ là cơ sở quan trọng cho việc cải thiện và phát triển hệ thống dự báo lượng mưa trong tương lai, hướng tới ứng dụng thực tế trong dự báo thời tiết và quản lý tài nguyên nước.

# CHƯƠNG 4. CÀI ĐẶT VÀ TRIỂN KHAI

## Cài đặt công cụ

### 4.1.1. Phần mềm Pycharm chạy Python

**- Bước 1:** Download PyCharm phiên bản mới nhất (2024.2.3):

[**https://www.jetbrains.com/pycharm/download/?section=windows**](https://www.jetbrains.com/pycharm/download/?section=windows)

Chọn phiên bản phù hợp với hệ điều hành (Windows/macOS/Linux). Với

Windows:

Click vào nút Download ở phần Community Edition *(miễn phí)*.

Sau khi tải xong file .exe, click đúp vào file để bắt đầu cài đặt.



Thực hiện các bước cài đặt theo hướng dẫn trên màn hình (Next → Next →

Install).

Nên tick vào ô “Add launchers dir to the PATH” để có thể gọi PyCharm từ

Terminal.

**- Bước 2:** Download python version 3.8.10 environment theo link: https://[www.jetbrains.com/pycharm/download/?section=windows#(các](http://www.jetbrains.com/pycharm/download/?section=windows&(c%C3%A1c) hệ điều hành khác tương tự)

**Cài đặt python trên máy local:**

Click đúp vào file .exe vừa tải về để cài đặt



Chọn tick vào ô Add Python 3.8.10 to PATH (lựa chọn này giúp chạy lệnh python

trên cmd, powershell trên windows)

Và click vào Install Now (Có thể đổi đường dẫn folder cài đặt bằng cách chọn phần Customize installation ở dưới)

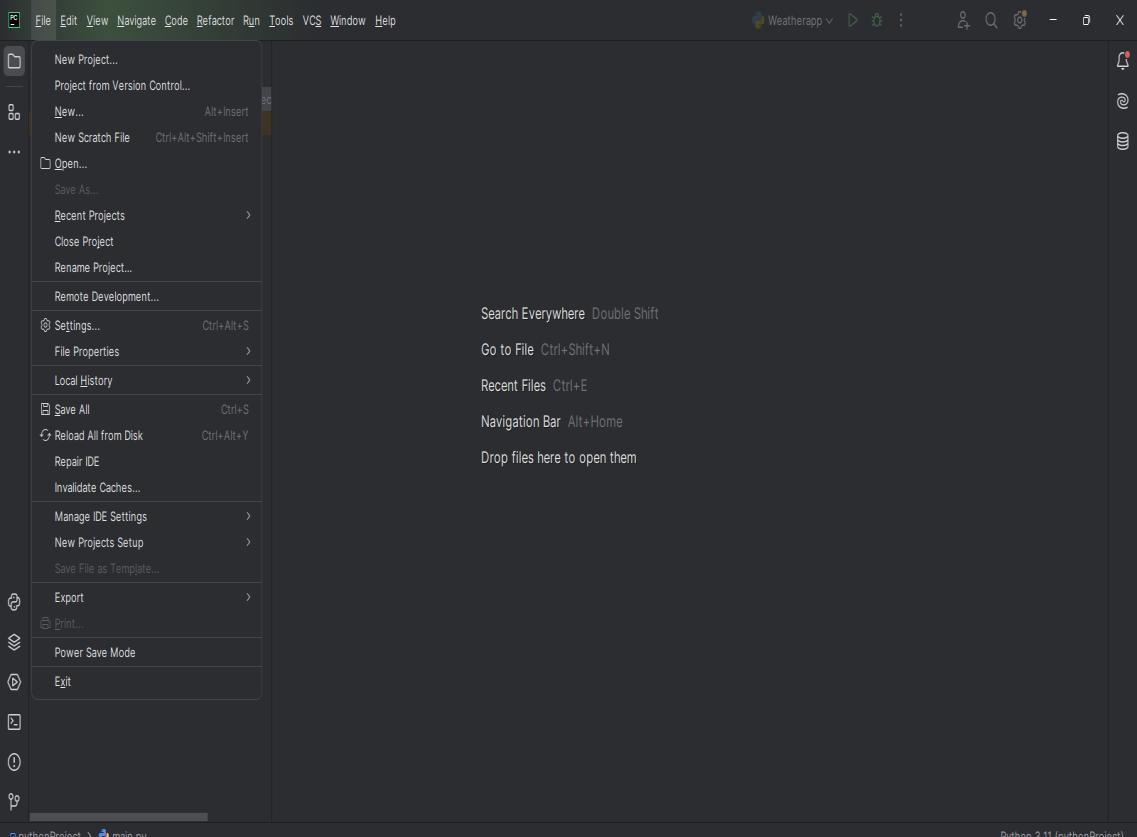


*Hình 4. 1 Cài đặt Visual Studio*

**- Bước 3:** Khởi động và thiết lập dự án trong PyCharm Sau khi cài đặt xong PyCharm và Python:

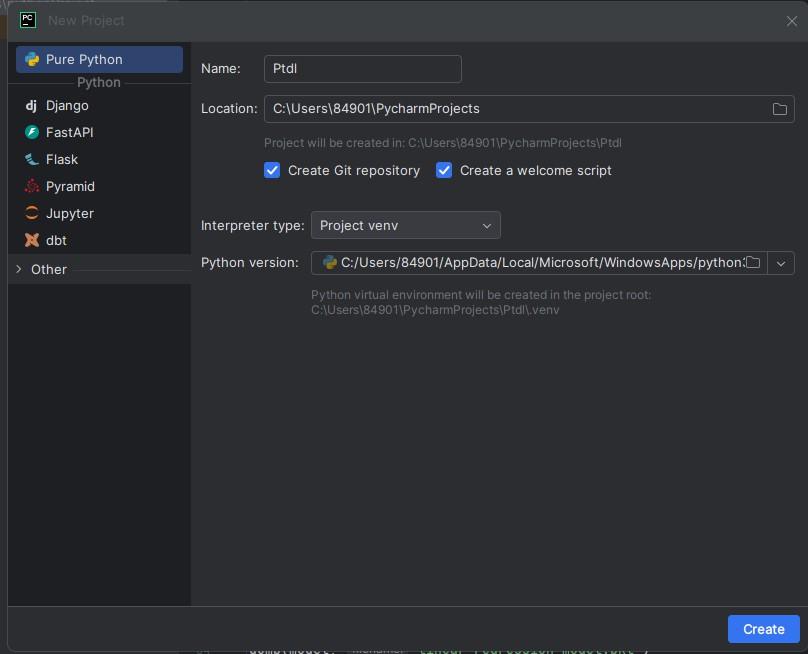
Mở PyCharm.

Chọn **"New Project"**.



Ở mục **Interpreter**, chọn đúng phiên bản Python đã cài ở bước 2.

Đặt tên thư mục cho project và nhấn **Create** để bắt đầu viết mã Python



*Hình 4.2: Cửa sổ đặt tên thư mục*

### 4.1.2. Tkinter

**Giới thiệu**

**Tkinter** là thư viện chuẩn của Python dùng để tạo giao diện người dùng (GUI). Nó đơn giản, dễ sử dụng và phù hợp cho các ứng dụng nhỏ hoặc các mô hình học máy có giao diện tương tác.

**Cài đặt**

Tkinter đã được tích hợp sẵn trong Python

**Sử dụng**

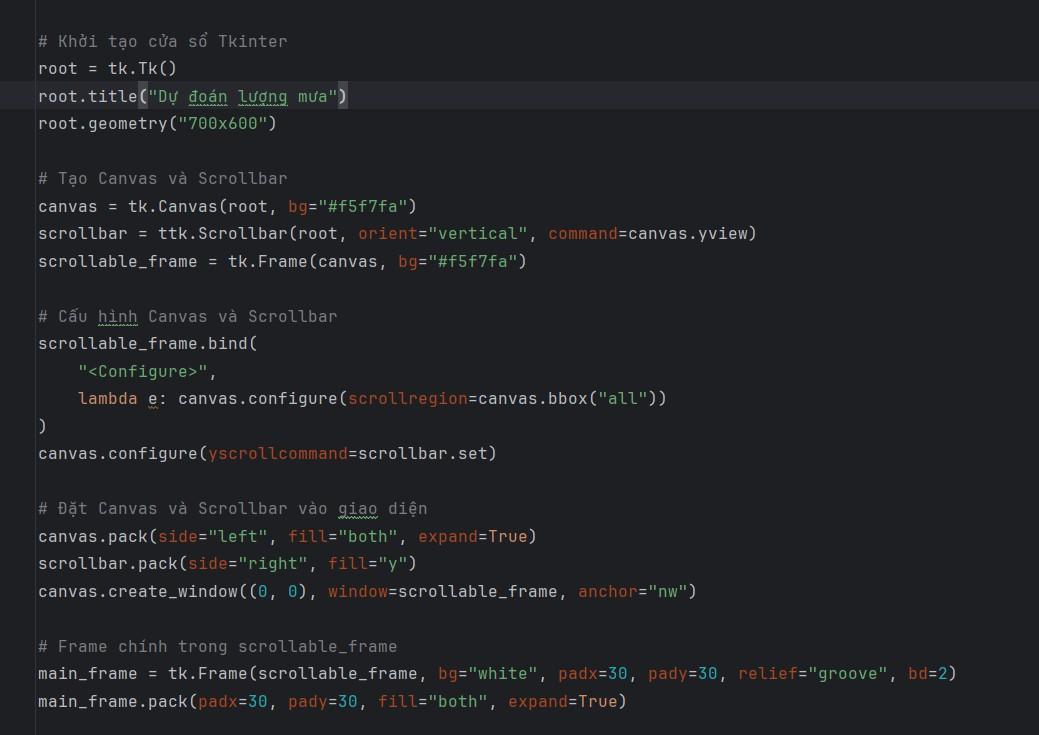
**Import các thư viện cần thiết**

Đầu tiên, chúng ta cần import các thư viện cần thiết trong Python:

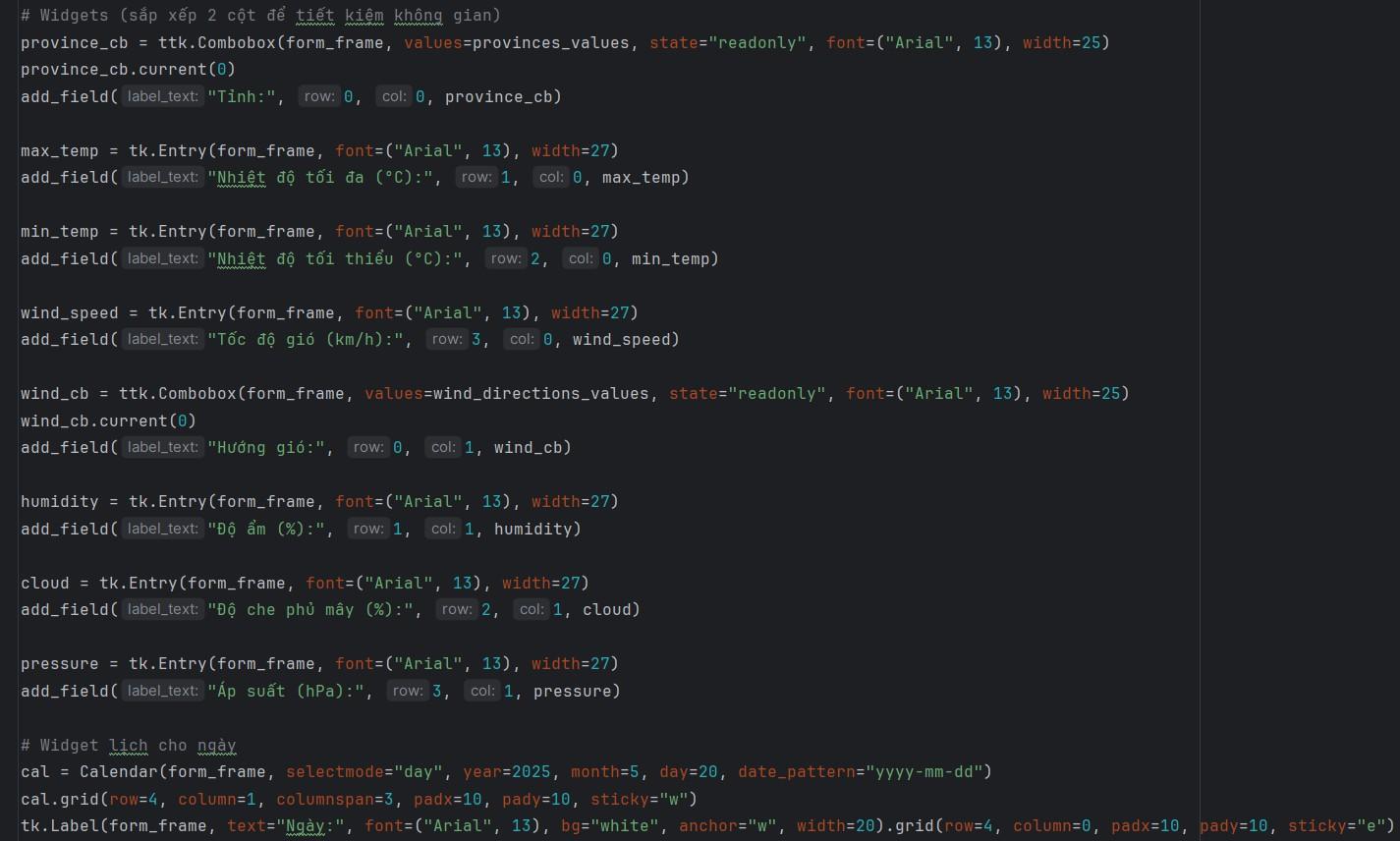


**Tạo giao diện**

Cuối cùng là xây dựng giao diện bằng thư viện **Tkinter**. Giao diện được thiết kế thân thiện, dễ sử dụng











*Hình 4.3: Xây dựng giao diện bằng Tkinter*

### 4.2.3. Cài đặt Spark

**Bước 1: Cài đặt Java**

Trước tiên, bạn cần cài đặt JDK. Bạn có thể tải JDK từ trang chính thức của Oracle hoặc OpenJDK. Sau khi tải xong, hãy tiến hành cài đặt:

1. Tải về file cài đặt JDK từ trang web chính thức.
2. Chạy file cài đặt và làm theo hướng dẫn.
3. Thiết lập biến môi trường JAVA\_HOME và thêm đường dẫn vào biến PATH.

Để kiểm tra việc cài đặt Java, bạn có thể mở terminal hoặc command prompt và nhập:

java -version

Nếu bạn thấy thông tin phiên bản của Java, điều này có nghĩa là cài đặt đã thành công.

**Bước 2: Tải về Apache Spark**

Sau khi cài đặt JDK thành công, bước tiếp theo là tải Apache Spark về. Có một số phiên bản cho Spark và bạn có thể chọn phiên bản phù hợp với nhu cầu của bạn.

1. Truy cập trang tải xuống của Apache Spark: <https://spark.apache.org/downloads.html>.
2. Chọn phiên bản Spark mà bạn muốn tải về. Thông thường, bạn nên chọn phiên bản mới nhất và tải xuống bản đã được pre-built với Hadoop.

Giải nén thư mục vừa tải xuống đến nơi mà bạn muốn cài đặt Spark trên máy tính của mình.

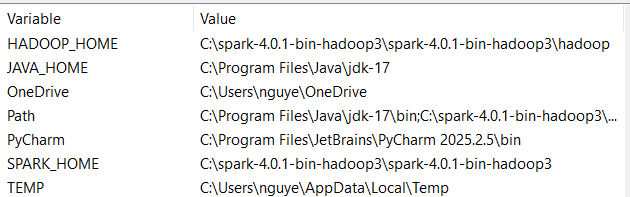
**Bước 3: Cấu hình môi trường**

Sau khi tải về và giải nén Apache Spark, bạn cần cấu hình các biến môi trường để Spark chạy êm đẹp.

1. Thiết lập biến môi trường SPARK\_HOME trỏ đến thư mục cài đặt Spark.
2. Thêm SPARK\_HOME/bin vào biến PATH của hệ thống.

Để làm điều này trên Windows, bạn có thể làm theo các bước sau:

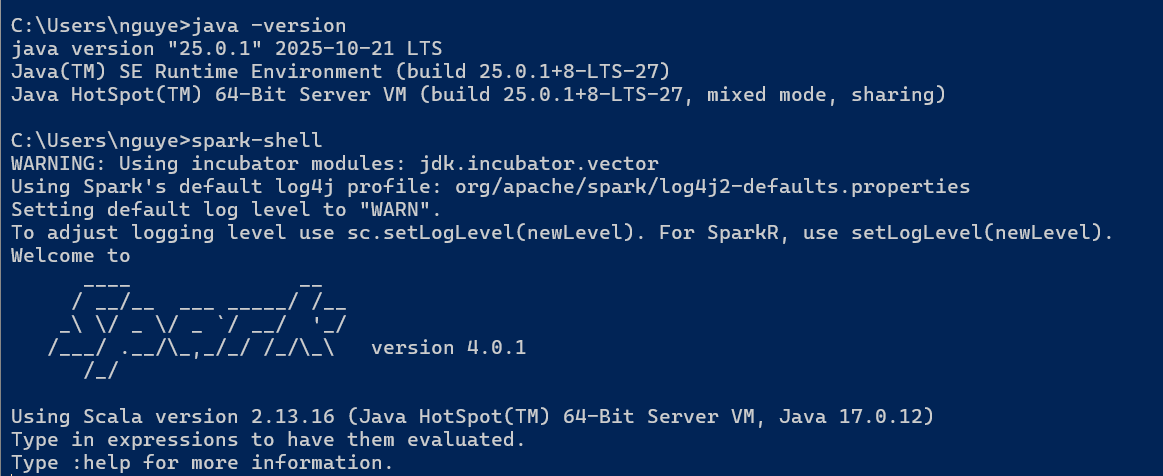
* Tìm kiếm "Environment Variables" trong thanh tìm kiếm và mở mục "Edit the system environment variables".
* Nhấn nút "Environment Variables".
* Trong phần "System variables", nhấn "New" và thêm biến SPARK\_HOME với giá trị là đường dẫn đến thư mục cài đặt Spark.
* Tìm biến PATH, chọn nó, rồi nhấn "Edit". Thêm vào hàng mới với giá trị “%SPARK\_HOME%\bin”.



Sau khi thực hiện xong, mở một terminal mới và gõ lệnh sau để kiểm tra:

spark-shell

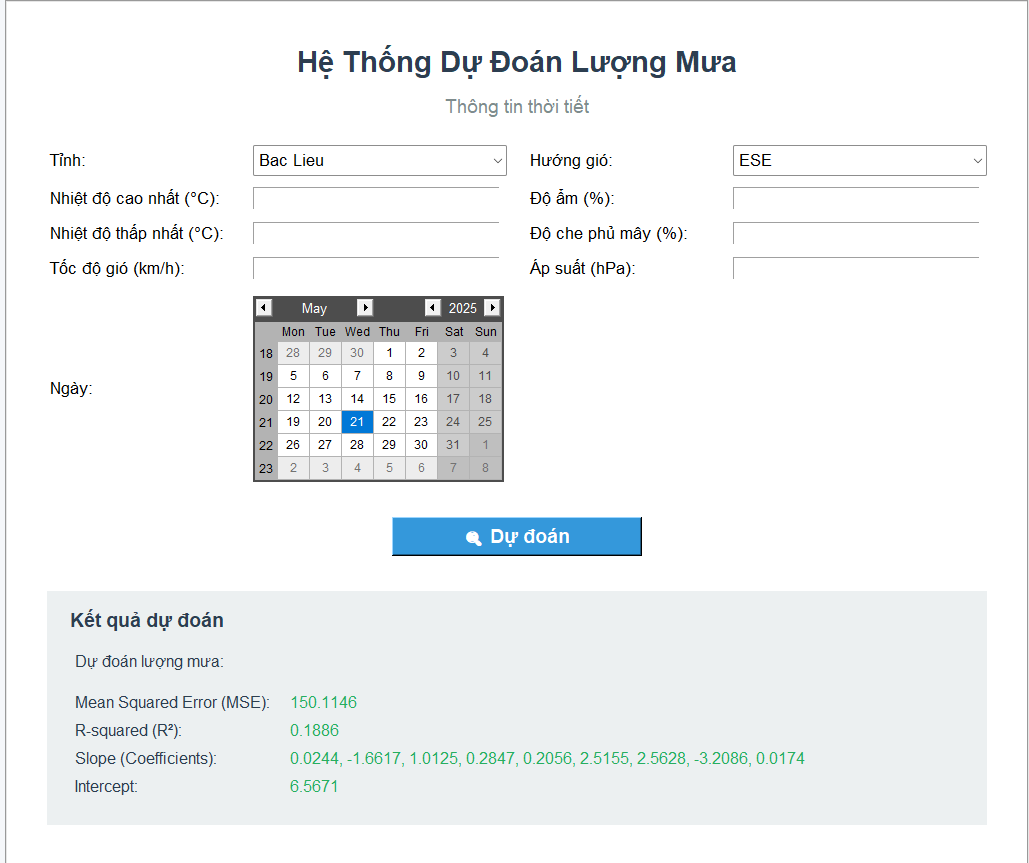
Nếu bạn thấy giao diện Spark shell xuất hiện, điều này có nghĩa là bạn đã cấu hình thành công.



* + - * 1. Các sử dụng Spark trong Pycharm

Mở terminal => pip install pyspark

## Giao diện



*Hình 4.4: Giao diện ứng dụng*

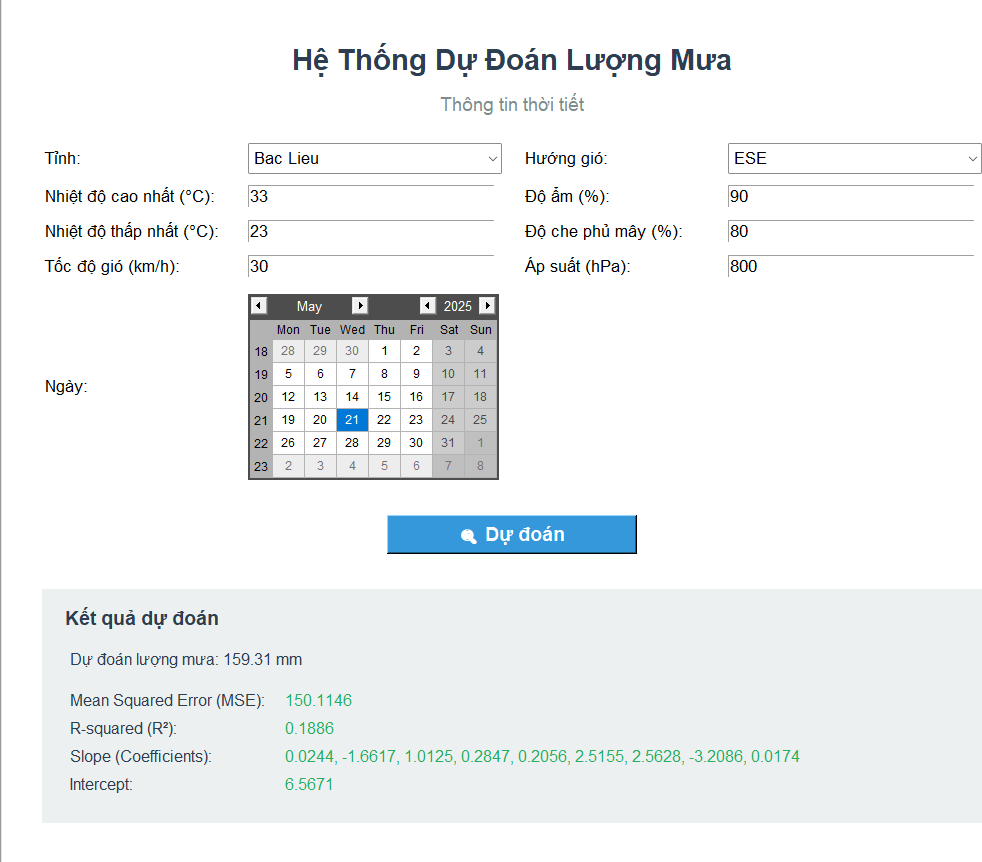
**a, Giao diện có tổng cộng 9 trường đầu vào.**

* **Tỉnh**: Trường chọn số để chọn tỉnh quan sát thời tiết (1: Bac Lieu, 2: Ho Chi

Minh City, ..., 40: Ba Ria - Vung Tau).

* **Nhiệt độ cao nhất**: Trường nhập số để nhập nhiệt độ cao nhất (°C).
* **Nhiệt độ thấp nhất**: Trường nhập số để nhập nhiệt độ thấp nhất (°C).
* **Tốc độ gió (km/h)**: Trường nhập số để nhập tốc độ gió (km/h).
* **Hướng gió**: Trường chọn số để chọn hướng gió (1: ESE, 2: SE, ..., 14: N).
* **Độ ẩm (%)**: Trường nhập số để nhập độ ẩm (%).
* **Độ che phủ mây (%)**: Trường nhập số để nhập độ che phủ mây (%).
* **Áp xuất (hPa)**: Trường nhập số để nhập áp suất (hPa).
* **Ngày**: Trường chọn ngày dự đoán thời tiết (YYYY-MM-DD).

**b, Đầu ra sẽ bao gồm:**



*Hình 4.5: Giao diện kết quả*

* **Dự đoán lượng mưa:** Trường hiển thị kết quả dự đoán lượng mưa dựa trên các thông số đầu vào.
* **Mean Squared Error:** Trường hiển thị giá trị Mean Squared Error (MSE) của mô hình dự đoán.
* **R-squared:** Trường hiển thị giá trị R-squared của mô hình dự đoán.
* **Slope:** Trường hiển thị giá trị Slope của mô hình dự đoán.
* **Intercept:** Trường hiển thị giá trị intercept của mô hình dự đoán

Ứng dụng này cho phép người dùng nhập các thông số liên quan đến thời tiết và sau đó dự đoán lượng mưa dựa trên các thông số đó. Các giá trị dự đoán cũng như các chỉ số đánh giá mô hình được hiển thị trong giao diện.

## 4.3. Kết luận

Trên đây là cách cài đặt và sử dụng Tkinter để tạo giao diện cho các ứng dụng AI và mô hình máy học. Tkinter là một thư viện mạnh mẽ và dễ sử dụng, giúp chúng ta tạo giao diện tương tác với mô hình một cách nhanh chóng và thuận tiện. Bằng cách sử dụng Tkinter, chúng ta có thể tạo ra các ứng dụng AI dễ dàng và chia sẻ chúng với người dùng một cách trực quan và hấp dẫn.

# KẾT LUẬN

Trong nghiên cứu này, nhóm chúng em đã thực hiện phân tích và dự đoán lượng mưa thông qua việc khảo sát các yếu tố ảnh hưởng đến lượng mưa, sử dụng mô hình hồi quy tuyến tính. Mục tiêu chính là áp dụng phương pháp hồi quy tuyến tính để xác định những yếu tố quan trọng và xây dựng mô hình dự đoán lượng mưa dựa trên các đặc điểm như nhiệt độ, độ ẩm, áp suất không khí, và các yếu tố khí tượng khác. Các bước thực hiện bao gồm thu thập dữ liệu, tiền xử lý, phân tích và áp dụng thuật toán hồi quy tuyến tính để đưa ra dự đoán.

Kết quả mà nhóm đạt được là mô hình hồi quy tuyến tính có khả năng dự đoán lượng mưa với độ chính xác khá cao. Sau khi dữ liệu được xử lý và làm sạch, mô hình đã cho ra các giá trị dự đoán chính xác, đồng thời nhận diện được các yếu tố có ảnh hưởng lớn nhất đến lượng mưa. Mô hình này có thể được áp dụng để dự báo lượng mưa, phục vụ cho công tác dự báo thời tiết và quản lý tài nguyên nước.

Tuy nhiên, một số vấn đề nhóm chúng em vẫn chưa giải quyết được là khi áp dụng phương pháp hồi quy tuyến tính trong các trường hợp có sự tương tác phức tạp giữa các yếu tố khí tượng. Các biến số có thể có mối quan hệ không tuyến tính, hoặc có sự bất thường trong dữ liệu khiến mô hình khó khăn trong việc đưa ra dự đoán chính xác. Những yếu tố này cần được nghiên cứu và xử lý thêm để cải thiện độ chính xác của mô hình. Bên cạnh đó, việc áp dụng các phương pháp phức tạp hơn như hồi quy đa biến hay các mô hình học sâu vẫn là một thách thức mà nhóm chưa thực hiện được.

Trong tương lai, nhóm sẽ tiếp tục phát triển mô hình dự đoán lượng mưa này bằng cách thử nghiệm với các kỹ thuật phức tạp hơn như hồi quy Ridge, Lasso, và các mô hình học sâu như mạng nơ-ron nhân tạo (RNN). Bên cạnh đó, nhóm cũng sẽ mở rộng bộ dữ liệu để tăng tính đa dạng và khả năng tổng quát của mô hình. Đồng thời, nhóm dự định cải thiện khả năng xử lý dữ liệu thiếu và nâng cao độ chính xác của các dự đoán thông qua các kỹ thuật tối ưu hóa và xử lý nâng cao.

# TÀI LIỆU THAM KHẢO

1. Vũ Hữu Tiếp: Machine Learning cơ bản. Funda, 2016.
2. “Data Analytics Made Accessible” Dr.Anil Maheshwari - NXB TechWorld, 2023.
3. Python Documentation - <https://docs.python.org/3/> . Lần truy cập gần nhất: 1/12/2024.
4. Bài 3: Linear Regression. URL: <https://machinelearningcoban.com/2016/12/28/linearregression/> . Lần truy cập gần nhất: 18/1/2024.
5. Khái niệm về tóm lược dữ liệu - [https://toolkit.ncats.nih.gov/glossary/data-](https://toolkit.ncats.nih.gov/glossary/data-Summary/) [Summary/](https://toolkit.ncats.nih.gov/glossary/data-Summary/). Lần truy cập gần nhất: 18/1/2024.
6. Khái niệm về làm sạch dữ liệu. URL: <https://www.tableau.com/learn/articles/what-is-data-cleaning> . Lần truy cập gần nhất: 21/1/2024.
7. Gradio. URL: <https://www.gradio.app/guides/quickstart>. Lần truy cập gần nhất: 31/1/2024.
8. Car-Price-Prediction. URL: [https://github.com/sagnikghoshcr7/Car-](https://github.com/sagnikghoshcr7/Car-Price-Prediction/tree/master/data) [Price-](https://github.com/sagnikghoshcr7/Car-Price-Prediction/tree/master/data) [Prediction/tree/master/data](https://github.com/sagnikghoshcr7/Car-Price-Prediction/tree/master/data). Lần truy cập gần nhất: 1/12/2024.