TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

**KHOA ĐIỆN TỬ**

**Bộ môn: Công nghệ Thông tin**.

**BÀI TẬP KẾT THÚC MÔN HỌC**

MÔN HỌC

**KHOA HỌC DỮ LIỆU**

Sinh viên: Giáp Quốc An

Lớp: 57KMT.01

Giáo viên GIẢNG DẠY: T.s Nguyễn Văn Huy

Link GitHub:

**Thái Nguyên – 2025**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP KẾT THÚC MÔN HỌC**

**MÔN HỌC: KHOA HỌC DỮ LIỆU**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

*Sinh viên:* Giáp Quốc An

*Lớp*: 57KMT.01

*Ngành:* Kỹ thuật máy tính

*Giáo viên hướng dẫn:* T.s Nguyễn Văn Huy

*Ngày giao đề: 20/05/2025 Ngày hoàn thành: 26/05/2025*

*Tên đề tài :*Xây dựng web app phân tích và dự báo nhiệt độ, lượng mưa tại một thành phố.

*Yêu cầu :* Biểu đồ thời tiết và kết quả dự báo cho 7 ngày kế tiếp.

|  |
| --- |
| **GIÁO VIÊN HƯỚNG DẪN** |
| *(Ký và ghi rõ họ tên)* |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

### Thái Nguyên, ngày….tháng…..năm 20....

## GIÁO VIÊN HƯỚNG DẪN

*(Ký ghi rõ họ tên)*

Mục Lục

[TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP 1](#_Toc18752)

[KHOA ĐIỆN TỬ 1](#_Toc12601)

[Bộ môn: Công nghệ Thông tin. 1](#_Toc32147)

[Thái Nguyên – 2025 1](#_Toc4451)

[Thái Nguyên, ngày….tháng…..năm 20.... 3](#_Toc6726)

[GIÁO VIÊN HƯỚNG DẪN 3](#_Toc26068)

[CHƯƠNG 1: MỞ ĐẦU 5](#_Toc30454)

[1.1. Lý do chọn đề tài 5](#_Toc13534)

[1.2. Mục tiêu nghiên cứu 6](#_Toc302)

[1.3. Phạm vi nghiên cứu 6](#_Toc15448)

[1.4. Phương pháp nghiên cứu 6](#_Toc21497)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ 7](#_Toc30062)

[2.1. Tổng quan về dự báo thời tiết 7](#_Toc19700)

[2.2. Chuỗi thời gian (Time Series) 8](#_Toc29665)

[2.3. Mô hình Prophet 8](#_Toc4884)

[2.4. Streamlit 8](#_Toc25656)

[CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU THỜI TIẾT 9](#_Toc25727)

[3.1. Nguồn dữ liệu 9](#_Toc18628)

[3.2. Tiền xử lý dữ liệu 10](#_Toc27675)

[3.3. Khám phá dữ liệu (EDA) 10](#_Toc21335)

[CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG DỰ BÁO 11](#_Toc9419)

[4.1. Kiến trúc tổng thể 11](#_Toc27170)

[4.2. Chi tiết chức năng 12](#_Toc16248)

[4.3. Biểu đồ minh họa 12](#_Toc6167)

[CHƯƠNG 5: KẾT LUẬN VÀ ĐỀ XUẤT PHÁT TRIỂN 13](#_Toc27207)

[5.1. Kết luận 13](#_Toc28093)

[5.2. Hạn chế 14](#_Toc2743)

[5.3. Hướng phát triển 14](#_Toc15716)

### 

### CHƯƠNG 1: MỞ ĐẦU

#### 1.1. Lý do chọn đề tài

Biến đổi khí hậu ngày càng làm gia tăng các hiện tượng thời tiết cực đoan, như nhiệt độ tăng đột biến, lũ lụt, hoặc hạn hán kéo dài. Những hiện tượng này không chỉ ảnh hưởng đến sản xuất nông nghiệp mà còn gây ra các vấn đề lớn trong giao thông, y tế công cộng, và các ngành kinh tế khác. Dự báo thời tiết chính xác, dễ tiếp cận là yếu tố then chốt để giảm thiểu rủi ro và hỗ trợ ra quyết định. Sự phát triển của trí tuệ nhân tạo (AI) và học máy, kết hợp với các công cụ lập trình web hiện đại, tạo cơ hội xây dựng các ứng dụng dự báo thời tiết thân thiện, trực quan. Đề tài này tận dụng các công nghệ như mô hình Prophet và Streamlit để cung cấp giải pháp dự báo thời tiết ngắn hạn hiệu quả, phù hợp với người dùng phổ thông.

#### 1.2. Mục tiêu nghiên cứu

* **Xây dựng ứng dụng web**: Tạo một ứng dụng web dựa trên Streamlit, dễ sử dụng, cho phép dự báo thời tiết ngắn hạn (1–14 ngày) tại một thành phố cụ thể.
* **Tiền xử lý dữ liệu**: Phân tích và xử lý tập dữ liệu thời tiết lịch sử để đảm bảo dữ liệu đủ chất lượng cho dự báo.
* **Áp dụng mô hình Prophet**: Sử dụng mô hình Prophet để dự báo nhiệt độ (°C) và lượng mưa (mm) với độ chính xác cao.
* **Trực quan hóa kết quả**: Hiển thị kết quả dự báo qua biểu đồ (Matplotlib) và bảng dữ liệu, dễ hiểu cho người dùng không chuyên.

#### 1.3. Phạm vi nghiên cứu

* **Dữ liệu**: Sử dụng tập dữ liệu thời tiết lịch sử (weatherHistory.csv) từ một thành phố cụ thể, tập trung vào nhiệt độ và lượng mưa.
* **Công cụ**: Framework Streamlit để xây dựng giao diện web, thư viện Prophet để dự báo, và Matplotlib để trực quan hóa.
* **Hạn chế phạm vi**: Không dự báo thời tiết toàn cầu hoặc nhiều thành phố cùng lúc; chỉ tập trung vào các yếu tố cơ bản (nhiệt độ, lượng mưa).

#### 1.4. Phương pháp nghiên cứu

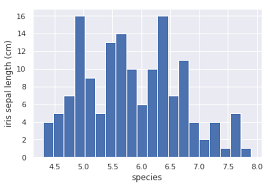
* **Phân tích dữ liệu khám phá (EDA)**: Hiểu cấu trúc và đặc điểm của dữ liệu thời tiết, xác định xu hướng, mùa vụ, và các bất thường.
* **Tiền xử lý dữ liệu**: Chuẩn hóa định dạng thời gian, xử lý giá trị thiếu, và tạo cột dữ liệu phù hợp cho Prophet.
* **Dự báo chuỗi thời gian**: Sử dụng mô hình Prophet để huấn luyện và dự đoán dựa trên dữ liệu lịch sử.
* **Triển khai ứng dụng**: Tích hợp mô hình vào giao diện Streamlit với các biểu đồ và bảng dữ liệu trực quan.

### CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

#### 2.1. Tổng quan về dự báo thời tiết

Dự báo thời tiết sử dụng dữ liệu khí tượng (nhiệt độ, độ ẩm, áp suất, lượng mưa, v.v.) và các mô hình toán học để dự đoán trạng thái thời tiết tương lai. Các yếu tố dự báo quan trọng bao gồm:

* Nhiệt độ (°C): Biến động theo ngày và mùa.
* Lượng mưa (mm): Quan trọng trong nông nghiệp, quản lý tài nguyên nước.
* Các yếu tố khác: Độ ẩm, áp suất khí quyển, tốc độ gió.



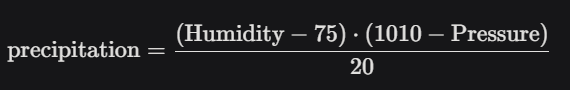
#### 2.2. Chuỗi thời gian (Time Series)

Chuỗi thời gian là tập hợp dữ liệu được ghi nhận theo thứ tự thời gian (hàng giờ, hàng ngày, v.v.). Trong dự báo thời tiết, dữ liệu nhiệt độ và lượng mưa là chuỗi thời gian, có đặc điểm:

* **Xu hướng (trend)**: Thay đổi dài hạn, ví dụ, nhiệt độ tăng do biến đổi khí hậu.
* **Mùa vụ (seasonality)**: Biến động định kỳ theo ngày, tuần, hoặc năm.
* **Nhiễu (noise)**: Các biến động ngẫu nhiên không dự đoán được.

#### 2.3. Mô hình Prophet

Prophet là một mô hình dự báo chuỗi thời gian mã nguồn mở do Facebook phát triển, phù hợp với dữ liệu có mùa vụ và xu hướng thay đổi. Công thức mô hình:



* Ưu điểm của Prophet:
* Xử lý dữ liệu thiếu và dị thường (outliers) hiệu quả.
* Tự động phát hiện mùa vụ và xu hướng.
* Dễ tích hợp vào các ứng dụng thực tế.

#### 2.4. Streamlit

Streamlit là một framework Python mã nguồn mở, giúp xây dựng ứng dụng web tương tác mà không cần kiến thức về HTML/CSS/JavaScript. Các tính năng nổi bật:

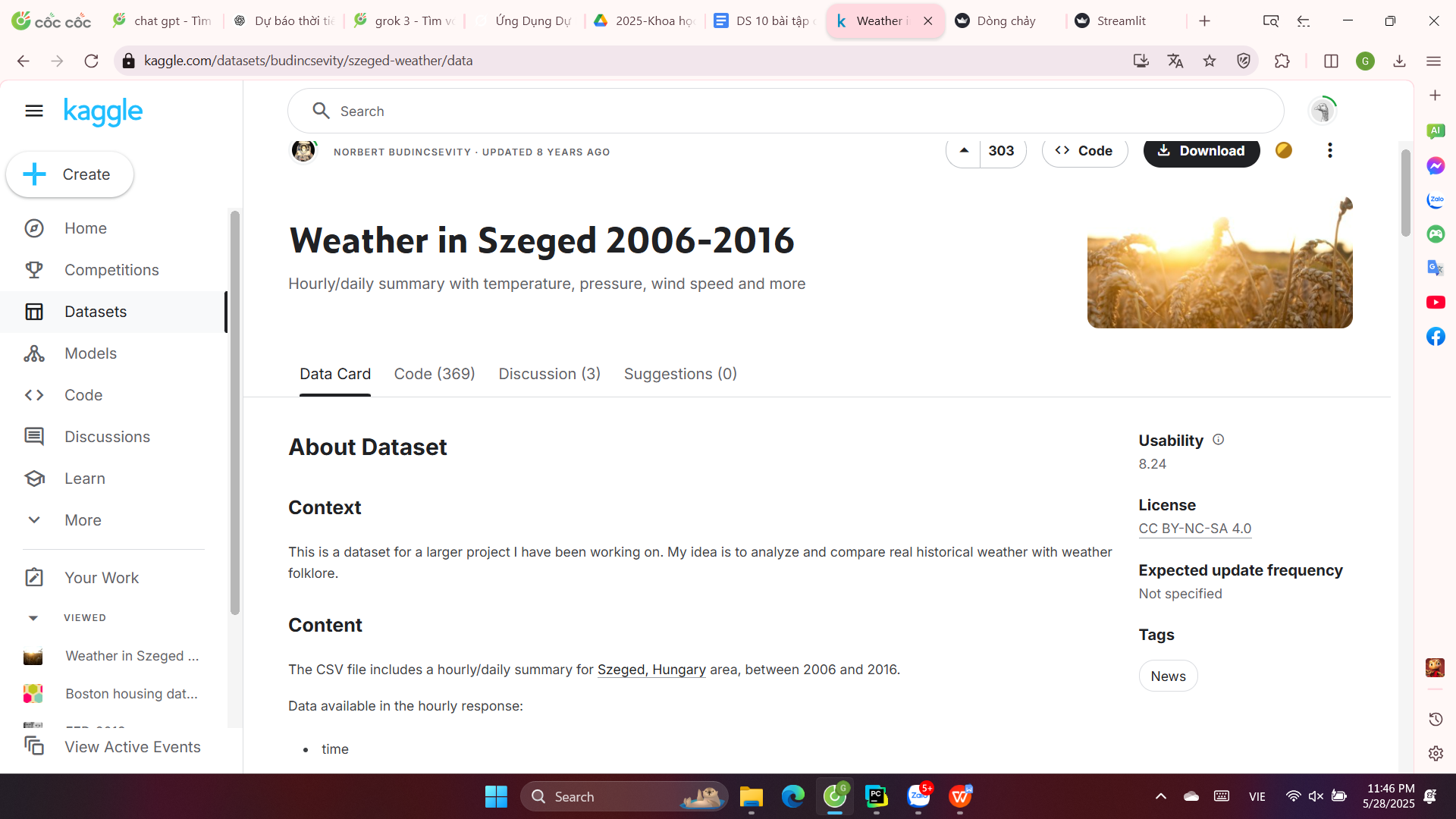
* Tích hợp với các thư viện Python như pandas, Matplotlib, và Prophet.
* Hỗ trợ các widget như selectbox, slider, và bảng dữ liệu.
* Tăng tốc độ phát triển ứng dụng phân tích dữ liệu.

### CHƯƠNG 3: PHÂN TÍCH DỮ LIỆU THỜI TIẾT

#### 3.1. Nguồn dữ liệu

Tập dữ liệu **weatherHistory.csv** từ Kaggle chứa thông tin thời tiết tại một thành phố trong nhiều năm, với các cột chính:

* **Formatted Date**: Thời gian ghi nhận (bao gồm múi giờ).
* **Temperature (C)**: Nhiệt độ trung bình.
* **Humidity**: Độ ẩm (%).
* **Pressure (millibars)**: Áp suất khí quyển.
* **Precip Type**: Loại mưa (mưa, tuyết, v.v.).
* **Precipitation Sum (mm)**: Tổng lượng mưa (nếu có).



#### 3.2. Tiền xử lý dữ liệu

Các bước tiền xử lý:

1. **Chuyển đổi định dạng thời gian**: Chuyển cột Formatted Date sang kiểu datetime, loại bỏ múi giờ để đồng nhất dữ liệu.
2. **Chuẩn hóa tên cột**: Thay khoảng trắng bằng dấu \_ (ví dụ: Temperature (C) → Temperature\_C).
3. **Tạo cột cho Prophet**: Tạo cột ds (ngày) và y (giá trị dự báo, ví dụ: nhiệt độ hoặc lượng mưa).
4. **Xử lý dữ liệu thiếu**:
5. **Loại bỏ dị thường**: Loại bỏ các giá trị bất thường (outliers) dựa trên ngưỡng thống kê (ví dụ: IQR).

#### 3.3. Khám phá dữ liệu (EDA)

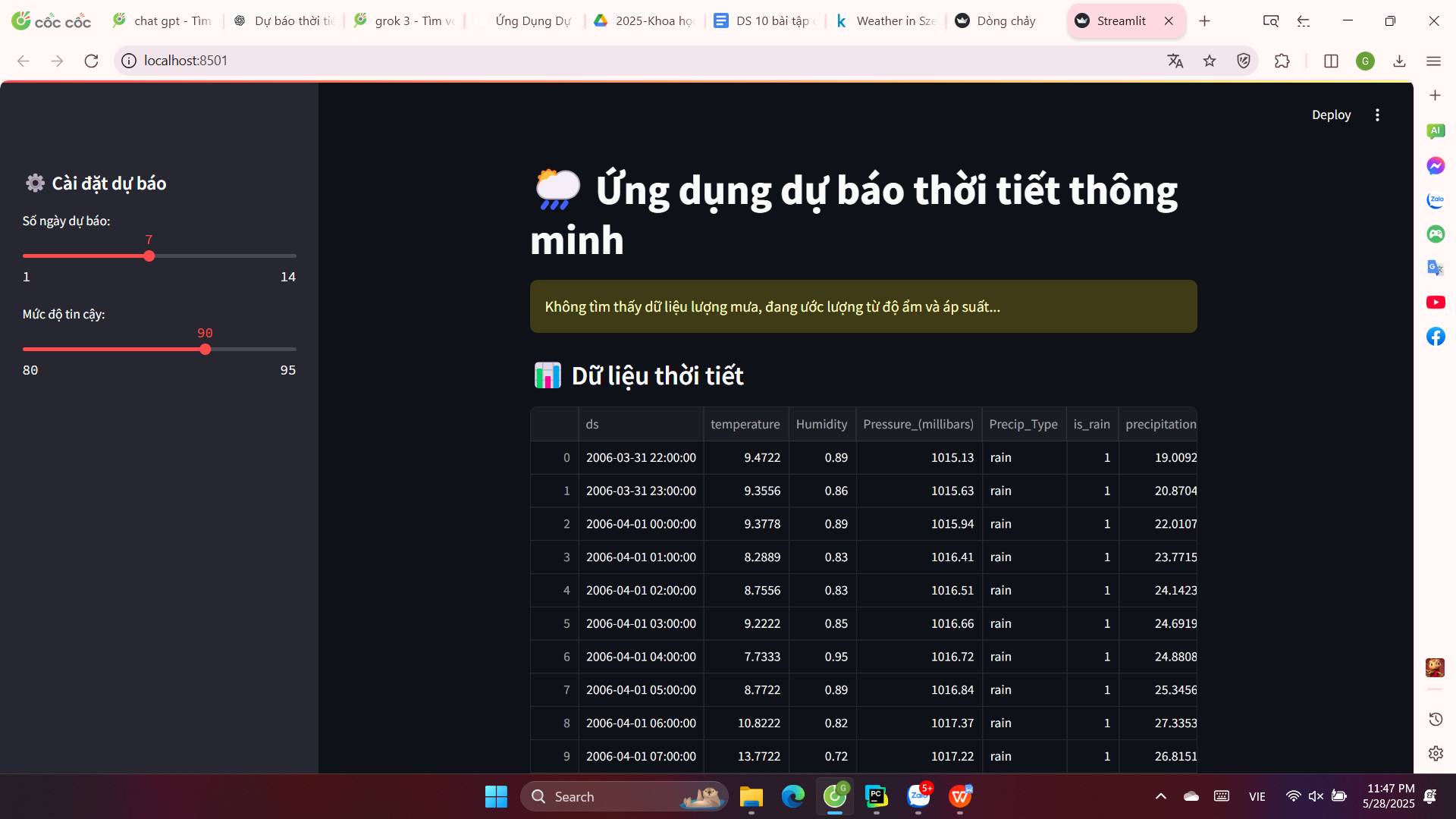
* **Biểu đồ phân bố**:
  + Nhiệt độ: Thể hiện xu hướng mùa vụ rõ ràng, dao động theo ngày và năm.
  + Lượng mưa: Phân bố lệch (skewed), phần lớn giá trị bằng 0 (ngày không mưa).
* **Xu hướng và mùa vụ**:
  + Nhiệt độ tăng vào mùa hè, giảm vào mùa đông.
  + Lượng mưa có xu hướng tập trung vào các tháng mùa mưa.
* **Tương quan**:
  + Độ ẩm và áp suất có ảnh hưởng đến lượng mưa, được sử dụng trong công thức heuristic.

### CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG DỰ BÁO

#### 4.1. Kiến trúc tổng thể

Ứng dụng bao gồm các module chính:

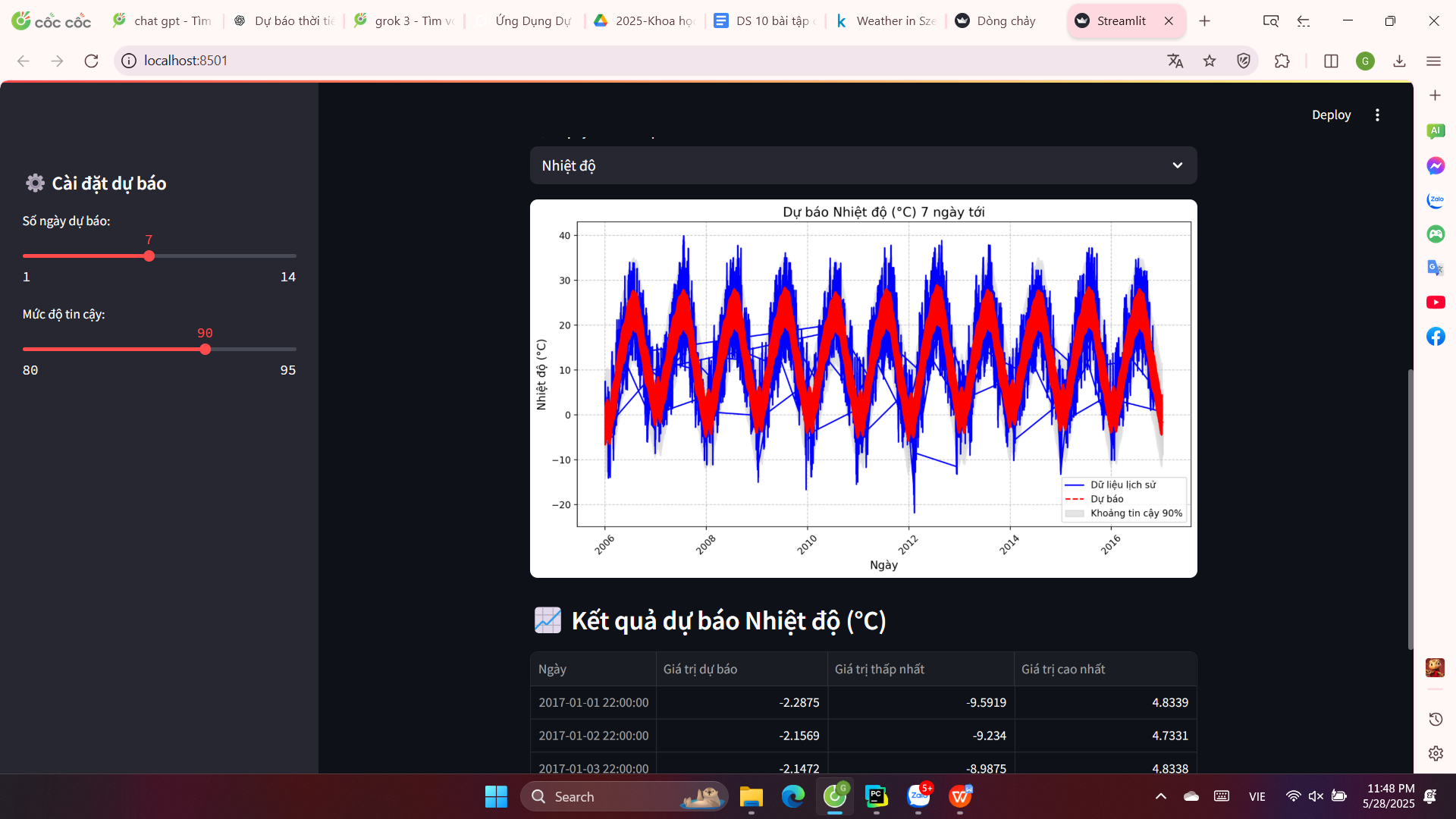
1. **Load dữ liệu**: Kiểm tra và đọc file weatherHistory.csv.
2. **Tiền xử lý**: Chuẩn hóa dữ liệu, xử lý giá trị thiếu, tính toán lượng mưa nếu cần.
3. **Giao diện người dùng**:
   * Chọn yếu tố dự báo (nhiệt độ/lượng mưa).
   * Điều chỉnh tham số (số ngày dự báo, mức độ tin cậy).
4. **Dự báo**: Huấn luyện mô hình Prophet và tạo dự đoán.
5. **Hiển thị kết quả**: Biểu đồ (Matplotlib) và bảng dữ liệu (Streamlit).

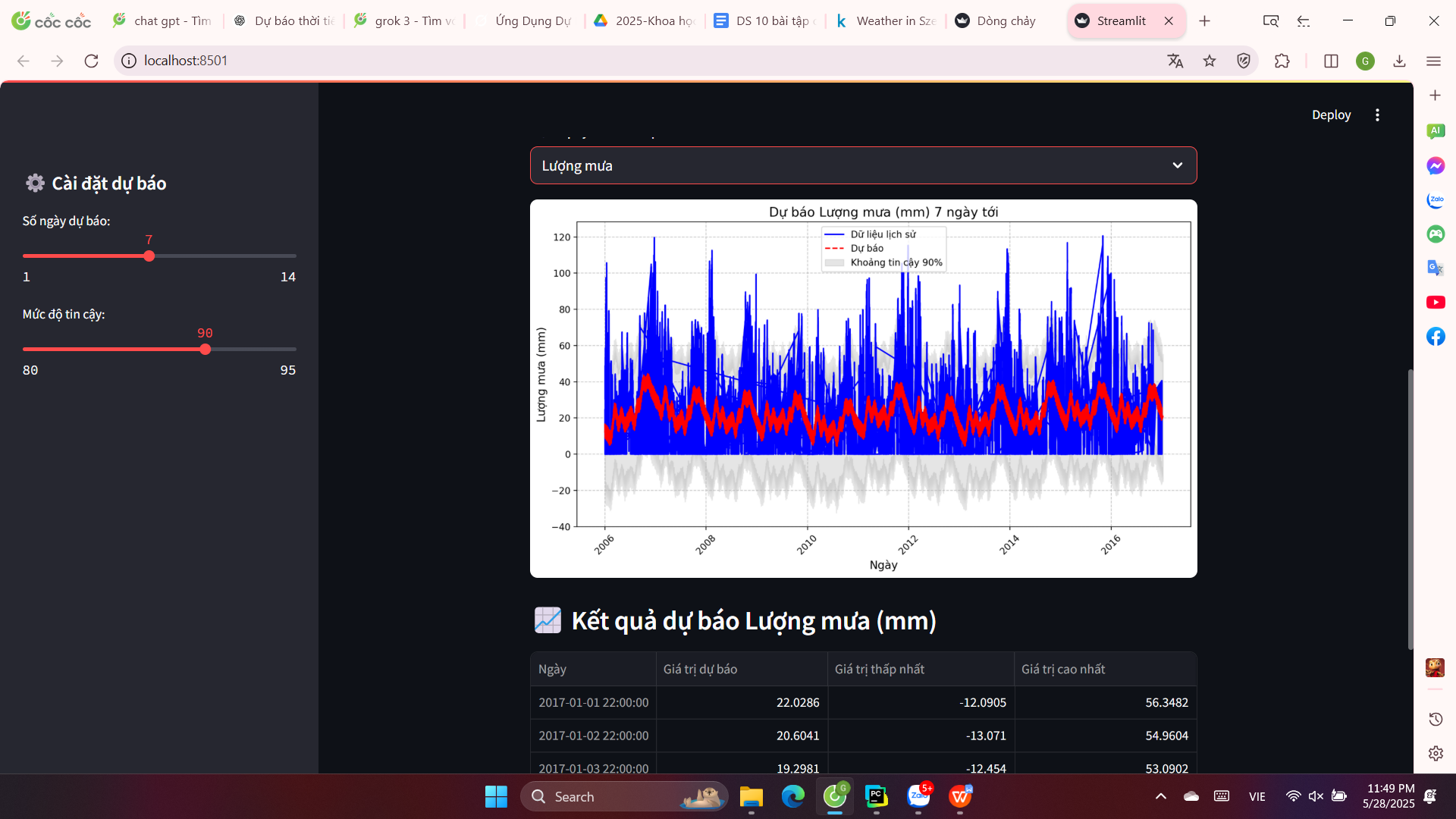


#### 4.2. Chi tiết chức năng

* @st.cache\_data: Lưu trữ dữ liệu đã xử lý để tăng tốc độ tải ứng dụng.
* st.selectbox: Cho phép người dùng chọn giữa “Nhiệt độ” và “Lượng mưa”.
* st.sidebar.slider: Điều chỉnh số ngày dự báo (1–14) và mức độ tin cậy (confidence interval, ví dụ: 95%).
* st.pyplot: Hiển thị biểu đồ với:
  + Đường lịch sử (xanh).
  + Đường dự báo (đỏ).
  + Vùng tin cậy (xám).
* st.dataframe: Hiển thị bảng chi tiết với các cột: ngày, giá trị dự báo, giá trị thấp nhất, giá trị cao nhất.

#### 4.3. Biểu đồ minh họa





### 

### CHƯƠNG 5: KẾT LUẬN VÀ ĐỀ XUẤT PHÁT TRIỂN

#### 5.1. Kết luận

Đề tài đã đạt được các mục tiêu đề ra:

* **Ứng dụng web**: Xây dựng thành công ứng dụng Streamlit trực quan, dễ sử dụng, cho phép người dùng dự báo nhiệt độ và lượng mưa ngắn hạn.
* **Mô hình Prophet**: Áp dụng hiệu quả để dự báo chuỗi thời gian, tự động xử lý dữ liệu thiếu và dị thường.
* **Trực quan hóa**: Biểu đồ và bảng dữ liệu rõ ràng, dễ hiểu, phù hợp với người dùng không chuyên.
* **Xử lý dữ liệu**: Tự động tính toán lượng mưa khi thiếu dữ liệu, đảm bảo tính liên tục của dự báo.

#### 5.2. Hạn chế

* **Phạm vi địa lý**: Chỉ hỗ trợ dự báo cho một thành phố cố định, không cho phép chọn địa điểm khác.
* **Dữ liệu tĩnh**: Dựa trên tập dữ liệu lịch sử (weatherHistory.csv), không cập nhật theo thời gian thực.
* **Mô hình đơn giản**: Chỉ sử dụng Prophet, chưa khai thác các mô hình học sâu phức tạp như LSTM hay Transformer.
* **Yếu tố hạn chế**: Chỉ dự báo nhiệt độ và lượng mưa, chưa tích hợp các yếu tố khác như tốc độ gió, độ che phủ mây, hoặc áp suất.

#### 5.3. Hướng phát triển

Để nâng cao giá trị và tính ứng dụng của hệ thống, các hướng phát triển sau được đề xuất:

1. **Tích hợp dữ liệu thời gian thực**:
   * Sử dụng API thời tiết như **OpenWeatherMap** ([https://openweathermap.org/api](https://openweathermap.org/api" \t "_blank)) hoặc **WeatherStack** ([https://weatherstack.com/](https://weatherstack.com/" \t "_blank)) để lấy dữ liệu thời tiết theo thời gian thực.
   * Cập nhật dữ liệu hàng giờ hoặc hàng ngày, đảm bảo dự báo sát với thực tế.
2. **Hỗ trợ đa địa điểm**:
   * Cho phép người dùng nhập tên thành phố hoặc tọa độ địa lý để dự báo thời tiết tại nhiều khu vực.
   * Xây dựng cơ sở dữ liệu địa phương hoặc sử dụng API để truy xuất thông tin thời tiết toàn cầu.
3. **Áp dụng mô hình học sâu**:
   * Tích hợp các mô hình như **LSTM** (Long Short-Term Memory) hoặc **GRU** (Gated Recurrent Unit) để dự báo chính xác hơn với dữ liệu phức tạp.
   * Kết hợp Prophet với **XGBoost** để cải thiện độ chính xác trong các trường hợp có nhiều biến phụ thuộc.
   * Thử nghiệm **Transformer** (như Temporal Fusion Transformer) để xử lý chuỗi thời gian với nhiều yếu tố khí tượng.
4. **Phân tích xu hướng dài hạn**:
   * Phân tích tác động của biến đổi khí hậu đến nhiệt độ và lượng mưa theo mùa hoặc năm.
   * Xây dựng các biểu đồ xu hướng dài hạn, ví dụ: tăng nhiệt độ trung bình qua các thập kỷ.
5. **Tích hợp các yếu tố thời tiết bổ sung**:
   * Dự báo thêm các yếu tố như tốc độ gió, độ che phủ mây, hoặc chỉ số UV.
   * Tích hợp các yếu tố địa lý (độ cao, khoảng cách đến biển) để cải thiện độ chính xác.
6. **Ứng dụng thực tế**:
   * **Nông nghiệp**: Cung cấp dự báo lượng mưa và nhiệt độ để hỗ trợ nông dân lập kế hoạch gieo trồng, tưới tiêu, hoặc thu hoạch.
   * **Logistics**: Dự báo thời tiết để tối ưu hóa lộ trình vận chuyển, tránh các điều kiện thời tiết xấu.
   * **Du lịch**: Tích hợp ứng dụng vào các nền tảng du lịch, cung cấp thông tin thời tiết theo điểm đến.
   * **Quản lý thiên tai**: Hỗ trợ cảnh báo sớm lũ lụt, hạn hán, hoặc sóng nhiệt dựa trên dự báo lượng mưa và nhiệt độ.
7. **Phiên bản mobile**:
   * Phát triển ứng dụng di động (iOS/Android) sử dụng các framework như Flutter hoặc React Native, tích hợp API thời tiết.
   * Hỗ trợ thông báo đẩy (push notifications) để cảnh báo thời tiết bất thường.
8. **Tăng cường giao diện người dùng**:
   * Thêm các tính năng như bản đồ thời tiết tương tác, hiển thị nhiệt độ hoặc lượng mưa theo khu vực.
   * Hỗ trợ đa ngôn ngữ (bao gồm tiếng Việt) để mở rộng đối tượng người dùng.
9. **Triển khai sản xuất**:
   * Đưa ứng dụng lên cloud (AWS, Google Cloud, hoặc Heroku) để đảm bảo khả năng mở rộng và truy cập liên tục.
   * Tối ưu hóa hiệu suất xử lý dữ liệu lớn và tích hợp hệ thống giám sát hiệu suất ứng dụng.