

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC CNTT & TT VIỆT – HÀN**

**Khoa Khoa Học Máy Tính**



**ĐỒ ÁN CHUYÊN NGÀNH 3**

# **XÂY DỰNG HỆ THỐNG PHÂN TÍCH VÀ DỰ BÁO THỜI TIẾT**

**Sinh viên thực hiện** : 1. Lê Hồng Anh – 21AD002  
2. Dương Tấn Huy – 21AD025

**Lớp** : 21AD

**Giảng viên hướng dẫn** : ThS. Hà Thị Minh Phương

**Đà Nẵng – 05/2025**

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC CNTT & TT VIỆT – HÀN**

**Khoa Khoa Học Máy Tính**



**ĐỒ ÁN CHUYÊN NGÀNH 3**

# **XÂY DỰNG HỆ THỐNG PHÂN TÍCH VÀ DỰ BÁO THỜI TIẾT**

**Sinh viên thực hiện** : 1. Lê Hồng Anh – 21AD002  
2. Dương Tấn Huy – 21AD025  
**Lớp** : 21AD  
**Giảng viên hướng dẫn** : ThS. Hà Thị Minh Phương

**Đà Nẵng – 05/2025**

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Đà Nẵng, ngày ... tháng ... năm 2025*

**Cán bộ hướng dẫn**

*(họ tên và chữ ký)*

## LỜI CẢM ƠN

Đối với một sinh viên tại trường đại học CNTT & TT Việt Hàn, đồ án chuyên ngành là một minh chứng cho những kiến thức đã có được sau một năm học tập tại trường.

Để thực hiện và hoàn thành tốt đồ án này, chúng em đã nhận được sự giúp đỡ và hướng dẫn rất tận tình của các thầy cô thuộc Khoa Khoa học máy tính, Trường Đại học Công Nghệ Thông Tin và Truyền Thông Việt – Hàn. Chúng em xin cảm ơn các thầy cô thuộc bộ môn chuyên ngành đã cung cấp cho chúng em các thông tin, kiến thức vô cùng quý báu và cần thiết trong suốt thời gian qua để chúng em có thể thực hiện và hoàn thành đồ án của mình. Đặc biệt chúng em xin chân thành cảm ơn cô Hà Thị Minh Phương người đã trực tiếp hướng dẫn chúng em trong thời gian thực hiện đồ án này. Chúng em xin được gửi lời cảm ơn chân thành nhất với cô.

Sau đó, xin cảm ơn các bạn trong Ngành Công nghệ thông tin đã ủng hộ, giúp đỡ, chia sẻ kiến thức, kinh nghiệm và tài liệu có được giúp chúng tôi trong quá trình nghiên cứu và thực hiện đề tài.

Cuối cùng, do giới hạn về mặt thời gian và kiến thức cũng như kinh nghiệm thực tiễn nên đề tài không tránh khỏi những sai sót. Chúng em rất mong nhận được sự thông cảm của quý thầy cô và mong đón nhận những góp ý của thầy cô và các bạn.

Chúng em xin chân thành cảm ơn!

# MỤC LỤC

<b>MỤC LỤC .....</b>	<b>5</b>
<b>DANH MỤC CÁC KÝ HIỆU, CÁC TỪ VIẾT TẮT .....</b>	<b>7</b>
<b>DANH MỤC CÁC BẢNG.....</b>	<b>8</b>
<b>DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ.....</b>	<b>9</b>
<b>MỞ ĐẦU.....</b>	<b>11</b>
1. Giới thiệu.....	11
2. Mục tiêu và nhiệm vụ nghiên cứu .....	11
3. Đối tượng và phạm vi nghiên cứu .....	12
4. Phương pháp nghiên cứu.....	12
5. Nội dung và kế hoạch thực hiện .....	13
6. Bố cục báo cáo.....	13
<b>Chương 1 – GIỚI THIỆU CHUNG.....</b>	<b>1</b>
1.1 Mô tả Bài toán Dự báo Thời tiết.....	1
1.1.1. Định nghĩa. ....	1
1.1.2. Lịch sử phát triển. ....	1
1.1.3. Tầm quan trọng. ....	4
1.1.4. Các thành phần của Bài toán Dự báo Thời tiết.....	4
1.1.5. Thách thức trong Dự báo Thời tiết.....	5
1.2 Mô tả về cách thức Dự báo Thời tiết.....	5
1.2.1 Các phương pháp dự báo thời tiết.....	5
1.2.2 Phương pháp lựa chọn.....	6
<b>Chương 2 – TỔNG QUAN KIẾN THỨC.....</b>	<b>7</b>
2.1 Giới thiệu chung về Machine Learning.....	7
2.1.1 Định nghĩa. ....	7
2.1.2 Các loại học máy.....	7
2.1.3 Quy trình chung của Machine Learning.....	8
2.1.4 Ứng dụng của Machine Learning.....	8
2.2 Phương pháp học Ensemble .....	9
2.2.1 Bagging Classifier .....	9
2.2.2 AdaBoost Classifier .....	10
2.2.3 Gradient Boosting Classifier.....	12
2.2.4 Hist Gradient Boosting Classifier .....	14
2.2.5 CatBoost Classifier.....	15
2.2.6 Tổng Quan Mối Quan Hệ Giữa Các Thuật Toán.....	16
2.3 Mô hình Deep Learning .....	17
2.3.1 Định nghĩa .....	17
2.3.2 Cấu trúc của Deep Learning .....	18
2.3.3 Phân loại .....	18
2.3.4 Quá trình huấn luyện Deep Learning .....	18
2.3.5 Ứng dụng của Deep Learning.....	19
2.3.6 Ưu điểm, nhược điểm.....	19
<b>Chương 3 – TRIỂN KHAI XÂY DỰNG.....</b>	<b>20</b>
3.1 Giới thiệu chung về Dataset .....	20
3.2 Triển khai xây dựng .....	21
3.2.1 Tổng quan về tập dữ liệu. ....	21
3.2.2 Phân tích dữ liệu thăm dò (EDA).....	25
3.2.3 Kỹ thuật đặc trưng (Feature Engineering).....	29
3.2.4 Đào tạo & Đánh giá Model. ....	36
3.2.5 So sánh hiệu suất của các mô hình cơ sở. ....	37
3.3 Models Điều chỉnh siêu tham số và xác thực chéo .....	38

3.4 Đào tạo và đánh giá mô hình học sâu .....	39
3.5 Triển khai Website Forecast với Streamlit .....	42
<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>47</b>
1. Kết quả đạt được.....	47
2. Hạn chế.....	47
3. Hướng phát triển .....	47
4. Kết luận. ....	48
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>49</b>

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

Từ viết tắt	Tiếng Việt	Tiếng Anh
EDA	Phân tích dữ liệu thăm dò	Exploratory Data Analysis
ML	Học máy	Machine Learning
DL	Học sâu	Deep Learning
V-H-C	Tầm nhìn – Độ ẩm – Độ bao phủ mây	Visibility – Humidity – Cloudcover

## DANH MỤC CÁC BẢNG

<i>Bảng 1. Bảng so sánh các phương pháp dự báo .....</i>	<i>5</i>
<i>Bảng 2. Ứng dụng của Machine Learning .....</i>	<i>9</i>
<i>Bảng 3. So Sánh Nguyên lý hoạt động &amp; Mỗi quan hệ .....</i>	<i>17</i>
<i>Bảng 4. Định nghĩa Dataset .....</i>	<i>21</i>



## DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 1.1: Tác phẩm "Meteorologica" của Nhà triết học Aristotle .....	1
Hình 1.2: Evangelista Torricelli phát minh ra khí áp kế.....	2
Hình 1.3: Nhiệt kế đo độ C đầu tiên .....	2
Hình 1.4: Bản vẽ mô hình ẩm kế tóc của Nhà vật lý Horace Bénédict de Saussure ..	2
Hình 1.5: TIROS-1 - Vệ tinh thời tiết đầu tiên của NASA .....	3
Hình 1.6: Tổ chức Khí tượng Thế giới .....	WMO4
Hình 2.1: Ensemble Learning .....	9
Hình 2.2: Bagging Classifier .....	10
Hình 2.3: AdaBoost Classifier .....	11
Hình 2.4: Gradient Boosting Classifier .....	12
Hình 2.5: Hist Gradient Boosting Classifier.....	14
Hình 2.6: CatBoost Classifier .....	15
Hình 2.7: Tổng Quan Mọi Quan Hệ .....	16
Hình 2.8: Sự khác biệt giữa machine learning và deep learning.....	17
Hình 3.1: Một số dòng đầu tiên của tập dữ liệu .....	21
Hình 3.2: Thông tin tóm tắt về dữ liệu .....	22
Hình 3.3: Thống kê tóm tắt.....	22
Hình 3.4: Kiểm tra giá trị null.....	23
Hình 3.5: Xử lý null với đặc trưng preciptype, severerisk .....	24
Hình 3.6: Xử lý null với đặc trưng solarradiation, solarenergy, snowdepth .....	24
Hình 3.7: Xử lý null với đặc trưng visibility .....	24
Hình 3.8: Xử lý null với đặc trưng uvindex.....	25
Hình 3.9: Biểu đồ phân phối, biểu đồ hộp, và biểu đồ xác suất .....	25
Hình 3.10: Biểu đồ cột Tần suất các loại thời tiết .....	26
Hình 3.11: Tần suất xuất hiện của các loại thời tiết .....	26
Hình 3.12: Multi-label classification .....	27
Hình 3.13: Số lượng mẫu cho từng nhãn thời tiết .....	t27
Hình 3.14: Biểu đồ cột Tần suất xuất hiện của 7 nhãn thời tiết .....	27
Hình 3.15: Biểu đồ heatmap.....	28
Hình 3.16: Nhóm biểu đồ phân tán .....	28
Hình 3.17: Biểu đồ phân tán cho Group 3 .....	28
Hình 3.18: Xử lý cột datetime, sunrise, sunset .....	29
Hình 3.19: Tạo biến đặc trưng và biến mục tiêu .....	29
Hình 3.20: Cân bằng dữ liệu .....	30
Hình 3.21: Biểu đồ cột (countplot) các nhãn sau khi cân bằng .....	30
Hình 3.22: Chia dữ liệu thành tập huấn luyện và kiểm tra.....	30
Hình 3.23: Mã hóa nhãn tập huấn luyện và kiểm tra cho preciptype .....	31
Hình 3.24: Áp dụng pipeline với 2 tập dữ liệu .....	31
Hình 3.25: Lựa chọn đặc trưng bằng feature score .....	32
Hình 3.26: Chọn đặc trưng tự động dựa trên RandomForestClassifier.....	32

Hình 3.27: Hàm vẽ countplot của các đặc trưng dựa trên loại tầm quan trọng.....	33
Hình 3.28: hàm plot_feature_importances và truyền 'gain' làm tham số .....	33
Hình 3.29: hàm plot_feature_importances và truyền 'weight' làm tham số.....	34
Hình 3.30: hàm plot_feature_importances và truyền 'cover' làm tham số .....	34
Hình 3.31: hàm plot_feature_importances và truyền 'total_gain' làm tham số.....	35
Hình 3.32: hàm plot_feature_importances và truyền 'total_cover' làm tham số .....	35
Hình 3.33: Lựa chọn các đặc trưng cuối cùng để huấn luyện.....	36
Hình 3.34: Chuẩn hóa dữ liệu chuẩn bị cho vào mô hình huấn luyện.....	36
Hình 3.35: Mảng lưu các giá trị và Hàm huấn luyện mô hình.....	36
Hình 3.36: Kết quả mô hình GradientBoostingClassifier .....	37
Hình 3.37: So sánh hiệu suất của các mô hình cơ sở.....	37
Hình 3.38: Định nghĩa không gian siêu tham số cho ExtraTreesClassifier.....	38
Hình 3.39: Kết quả best_score và best_params sau khi huấn luyện.....	39
Hình 3.40: Tóm tắt mô hình .....	40
Hình 3.41: Biểu đồ tổn thất và biểu đồ độ chính xác .....	40
Hình 3.42: So sánh lại sau khi huấn luyện model Deep learning.....	41
Hình 3.43: Tính điểm độ chính xác của mô hình cuối cùng được chọn.....	41
Hình 3.44: Lưu mô hình hoạt động tốt nhất để triển khai.....	41
Hình 3.45: Giao diện chính.....	42
Hình 3.46: Tìm kiếm và dự báo thời tiết khu vực đã chọn .....	43
Hình 3.47: Biểu đồ nhiệt theo giờ trong ngày .....	43
Hình 3.48: Biểu đồ mưa theo giờ trong ngày.....	44
Hình 3.49: Biểu đồ tuyết theo giờ trong ngày.....	44
Hình 3.50: Biểu đồ V-H-C theo giờ trong ngày.....	45
Hình 3.51: Biểu đồ gió theo giờ trong ngày .....	45
Hình 3.52: Giao diện nhập thủ công.....	46
Hình 3.53: Nhập thông tin thủ công và dự đoán thời tiết .....	46

# MỞ ĐẦU

## 1. Giới thiệu

Đề tài này tập trung vào việc xây dựng một hệ thống phân tích và dự báo thời tiết nhằm cung cấp thông tin dự báo chính xác và đáng tin cậy về điều kiện thời tiết trong tương lai. Mục tiêu của đề tài là phát triển một hệ thống có khả năng xử lý và phân tích dữ liệu từ nhiều nguồn khác nhau như các trạm khí tượng, vệ tinh, và các nguồn dữ liệu mở. Sau đó, sử dụng các mô hình và thuật toán học máy (Machine Learning) cũng như học sâu (Deep Learning) để dự đoán dựa trên các yếu tố thời tiết như ngày giờ, nhiệt độ, nhiệt độ điểm sương, phần trăm độ ẩm tương đối, tốc độ gió, tầm nhìn, áp suất khí quyển.

Hệ thống dự báo thời tiết này sẽ bao gồm các bước chính: thu thập dữ liệu thời tiết từ nhiều nguồn khác nhau, xử lý và phân tích dữ liệu để đảm bảo chất lượng và tính nhất quán, và áp dụng các mô hình học máy và học sâu tiên tiến để dự báo thời tiết. Các mô hình sẽ được sử dụng để xây dựng các dự báo chính xác. Ngoài ra, hệ thống sẽ sử dụng các công cụ trực quan hóa dữ liệu để tạo ra các biểu đồ và đồ thị minh họa, giúp người dùng dễ dàng hiểu và sử dụng thông tin dự báo.

Kết quả dự kiến của đề tài là một hệ thống dự báo thời tiết hiệu quả, cung cấp thông tin dự báo chính xác và đáng tin cậy cho người dùng thông qua giao diện thân thiện và dễ sử dụng. Hệ thống này sẽ giúp người dân lập kế hoạch cho các hoạt động ngoài trời một cách hiệu quả và an toàn, đồng thời hỗ trợ các tổ chức và doanh nghiệp trong việc tối ưu hóa hoạt động dựa trên dự báo thời tiết. Bằng cách cung cấp thông tin thời tiết kịp thời và hữu ích, hệ thống sẽ góp phần nâng cao chất lượng cuộc sống và giảm thiểu các rủi ro liên quan đến thời tiết.

## 2. Mục tiêu và nhiệm vụ nghiên cứu

### - Mục tiêu:

Mục tiêu của đề tài là xây dựng một hệ thống phân tích và dự báo thời tiết chính xác và đáng tin cậy. Hệ thống này sẽ giúp người dùng lập kế hoạch cho các hoạt động ngoài trời một cách hiệu quả và an toàn, dựa trên thông tin thời tiết được cung cấp. Cụ thể, các mục tiêu bao gồm:

- + Phát triển hệ thống có khả năng thu thập, xử lý và phân tích dữ liệu thời tiết từ nhiều nguồn khác nhau.

- + Áp dụng các mô hình và thuật toán học máy (Machine Learning) và học sâu (Deep Learning) để dự báo các điều kiện thời tiết tương lai.

- + Phát triển giao diện người dùng thân thiện để cung cấp thông tin dự báo thời tiết chính xác và đáng tin cậy.

### - Nhiệm vụ:

**Thu thập dữ liệu:** Xác định và thu thập dữ liệu thời tiết từ nhiều nguồn khác nhau như các trạm khí tượng, vệ tinh, các tổ chức cung cấp dữ liệu thời tiết và các nguồn dữ liệu mở.

**Xử lý và tổ chức dữ liệu:** Làm sạch, biến đổi và lưu trữ dữ liệu một cách hiệu quả, đảm bảo dữ liệu có chất lượng cao và sẵn sàng cho quá trình phân tích.

**Phân tích dữ liệu:** Sử dụng các phương pháp phân tích thống kê và trực quan hóa để khám phá các đặc điểm quan trọng của dữ liệu thời tiết.

**Áp dụng các mô hình học máy, học sâu:** Lựa chọn, huấn luyện và đánh giá các mô hình học máy để dự báo thời tiết, tối ưu hóa hiệu suất của các mô hình.

**Phát triển hệ thống giao diện:** Tích hợp các thành phần hệ thống và phát triển giao diện người dùng thân thiện để cung cấp thông tin dự báo thời tiết cho người dùng.

### 3. Đối tượng và phạm vi nghiên cứu

#### - Đối tượng nghiên cứu:

**Dữ liệu thời tiết:** Dữ liệu từ các trạm khí tượng, vệ tinh, các tổ chức cung cấp dữ liệu thời tiết và các nguồn dữ liệu mở.

**Mô hình dự báo:** Các mô hình và thuật toán học máy và học sâu sử dụng để dự báo thời tiết

**Người dùng:** Các cá nhân và tổ chức có nhu cầu dự báo thời tiết để lập kế hoạch cho các hoạt động ngoài trời.

#### - Phạm vi nghiên cứu:

**Phạm vi dữ liệu:** Dữ liệu thời tiết được thu thập từ nhiều nguồn khác nhau, bao gồm dữ liệu lịch sử và dữ liệu hiện tại.

**Phạm vi không gian:** Hệ thống sẽ tập trung vào dự báo thời tiết cho một hoặc nhiều khu vực cụ thể, có thể là một thành phố, một vùng hoặc cả quốc gia.

**Phạm vi thời gian:** Dự báo thời tiết sẽ bao gồm các khoảng thời gian ngắn hạn (vài giờ đến vài ngày) và dài hạn (vài tuần đến vài tháng).

### 4. Phương pháp nghiên cứu

#### - Phương pháp xử lý và phân tích dữ liệu:

**Làm sạch dữ liệu:** Sử dụng các kỹ thuật làm sạch dữ liệu để loại bỏ các dữ liệu nhiễu và xử lý các giá trị thiếu.

**Phân tích thống kê:** Sử dụng các phương pháp phân tích thống kê để khám phá và hiểu rõ các đặc điểm của dữ liệu.

**Trực quan hóa dữ liệu:** Sử dụng các công cụ trực quan hóa như Matplotlib, Seaborn, Tableau để minh họa dữ liệu và phát hiện các xu hướng.

#### - Phương pháp xây dựng và đánh giá mô hình:

**Chọn mô hình:** Lựa chọn các mô hình học máy và học sâu phù hợp.

**Huấn luyện mô hình:** Sử dụng tập dữ liệu lịch sử để huấn luyện các mô hình học máy và học sâu, tối ưu hóa các tham số để cải thiện độ chính xác.

**Đánh giá mô hình:** Sử dụng các tập dữ liệu kiểm tra và các chỉ số đánh giá như Accuracy, Precision, Recall, F1-score để đánh giá hiệu suất của các mô hình.

## 5. Nội dung và kế hoạch thực hiện

Thời gian	Nội dung thực hiện
Tuần thứ 1 – 2	<ul style="list-style-type: none"><li>- Thu thập phác thảo các ý tưởng, chức năng, yêu cầu hệ thống và dự đoán các vấn đề trong quá trình thiết kế hệ thống.</li><li>- Viết đề cương Đồ án Chuyên ngành 3.</li></ul>
Tuần thứ 3 – 4	<ul style="list-style-type: none"><li>- Thu thập, xử lý và tổ chức được dữ liệu.</li><li>- Phân tích và Trực quan hóa dữ liệu.</li></ul>
Tuần thứ 5 – 8	<ul style="list-style-type: none"><li>- Áp dụng ML và DL để xây dựng hệ thống cho phép xử lý, tổ chức và phân tích dữ liệu nhằm dự báo, hỗ trợ ra quyết định.</li><li>- Đánh giá kết quả các mô hình.</li><li>- Triển khai hệ thống website bằng Streamlit.</li></ul>
Tuần thứ 9 – 10	<ul style="list-style-type: none"><li>- Dùng thử sản phẩm và kiểm tra các ngoại lệ có trong hệ thống.</li><li>- Khắc phục và hoàn thiện sản phẩm.</li></ul>
Tuần thứ 11 – 12	<ul style="list-style-type: none"><li>- Chuẩn bị báo cáo và slide Đồ án Chuyên ngành 3.</li></ul>

## 6. Bố cục báo cáo

Sau phần *Mở đầu*, báo cáo được trình bày trong ba chương, cụ thể như sau:

Chương 1. *Giới thiệu chung*

Chương 2. *Tổng quan kiến thức*

Chương 3. *Triển khai xây dựng*

Cuối cùng là *Kết luận, Tài liệu tham khảo* liên quan đến đề tài.

# Chương 1 – GIỚI THIỆU CHUNG

## 1.1 Mô tả Bài toán Dự báo Thời tiết

### 1.1.1. Định nghĩa.

Dự báo thời tiết là quá trình dự đoán trạng thái tương lai của khí quyển tại một địa điểm và thời gian cụ thể. Việc này sử dụng các mô hình toán học, thuật toán và dữ liệu hiện tại về điều kiện khí quyển.

### 1.1.2. Lịch sử phát triển.

#### ▪ Thời Kỳ Cổ Đại và Trung Cổ:

- Thời cổ đại: Người cổ đại dựa vào quan sát các hiện tượng tự nhiên và kinh nghiệm để dự đoán thời tiết. Các nền văn minh như người Babylon, Hy Lạp, và Trung Quốc sử dụng các mẫu mực đơn giản như hình dạng của mây, màu sắc của bầu trời để dự đoán thời tiết.

- Aristotle (384–322 TCN): Nhà triết học Hy Lạp này đã viết tác phẩm "Meteorologica", một trong những công trình đầu tiên về hiện tượng thời tiết và khí tượng.



Hình 1.1: Tác phẩm "Meteorologica" của Nhà triết học Aristotle

#### ▪ Thế Kỷ 17 và 18: Bước Đầu của Khoa Học Khí Tượng:

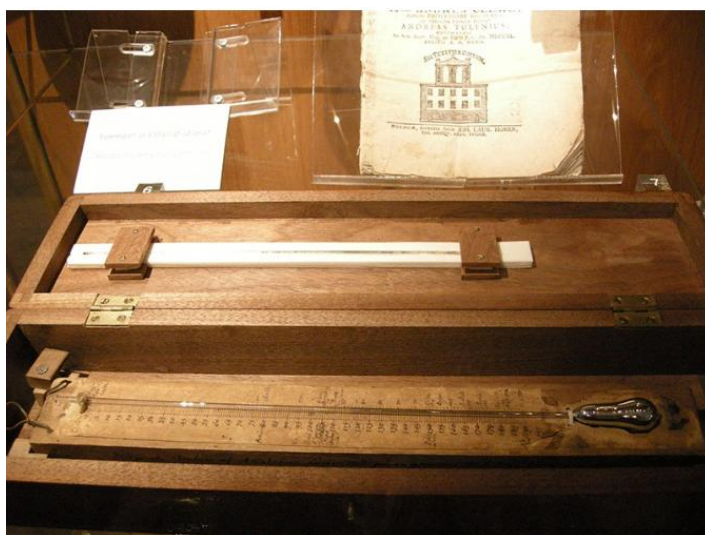
- Phát minh các thiết bị đo lường:

+ Barometer (1643): Evangelista Torricelli phát minh ra khí áp kế, thiết bị đầu tiên đo áp suất khí quyển.



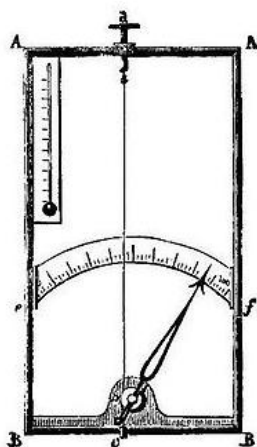
*Hình 1.2: Evangelista Torricelli phát minh ra khí áp kế*

+ Thermometer (1593): Galileo Galilei phát minh ra nhiệt kế, cải tiến bởi Daniel Gabriel Fahrenheit và Anders Celsius sau này.



*Hình 1.3: Nhiệt kế đo độ C đầu tiên*

+ Hygrometer (1783): Horace-Bénédict de Saussure phát minh ra ẩm kế, thiết bị đo độ ẩm không khí.



*Hình 1.4: Bản vẽ mô hình ẩm kế tóc của Nhà vật lý Horace Bénédict de Saussure*

- Blaise Pascal (1648): Chứng minh rằng áp suất khí quyển giảm theo độ cao, mở ra nghiên cứu về tầng khí quyển.

▪ ***Thế Kỷ 19: Sự Hình Thành của Khí Tượng Học Hiện Đại:***

- Telegraph (1837): Sự ra đời của điện báo cho phép truyền tải thông tin thời tiết nhanh chóng, giúp cải thiện dự báo thời tiết.

- Robert FitzRoy (1854): Được coi là người sáng lập dự báo thời tiết hiện đại. Ông đã thiết lập hệ thống báo cáo thời tiết và đưa ra các dự báo hàng ngày ở Anh.

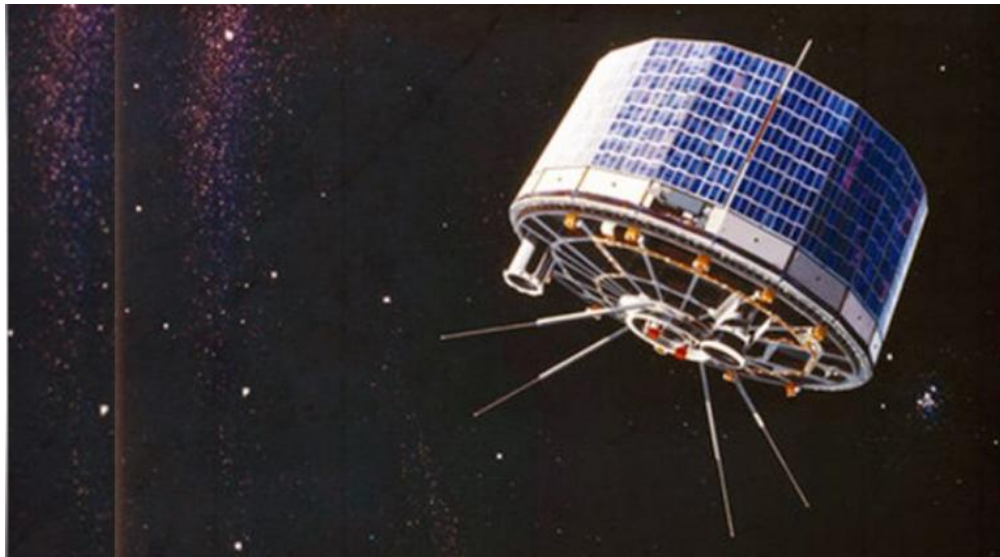
- William Ferrel (1856): Đóng góp lý thuyết quan trọng về sự lưu thông của khí quyển, giải thích cách mà các khối không khí di chuyển và tương tác với nhau.

▪ ***Thế Kỷ 20: Sự Phát Triển Mạnh Mẽ và Tinh Vi Hóa:***

- Sự phát triển của máy tính (1940s): Các máy tính điện tử đầu tiên cho phép thực hiện các mô hình toán học phức tạp để dự báo thời tiết.

- Numerical Weather Prediction (NWP): John von Neumann và nhóm của ông tại Princeton phát triển các mô hình số dự báo thời tiết đầu tiên, sử dụng các phương trình động lực học khí quyển.

- Sự ra đời của vệ tinh thời tiết (1960): TIROS-1, vệ tinh thời tiết đầu tiên của NASA, cung cấp các hình ảnh đầu tiên về mây từ không gian, mở ra kỷ nguyên mới cho dự báo thời tiết.



*Hình 1.5: TIROS-1 - Vệ tinh thời tiết đầu tiên của NASA*

▪ ***Thế Kỷ 21: Công Nghệ Cao và Học Máy:***

- Học máy và trí tuệ nhân tạo: Sử dụng các thuật toán học máy để phân tích và dự báo thời tiết dựa trên dữ liệu lớn, cải thiện đáng kể độ chính xác và tốc độ dự báo.

- Mạng lưới toàn cầu: Các tổ chức như Tổ chức Khí tượng Thế giới (WMO) phối hợp và chia sẻ dữ liệu thời tiết toàn cầu, tạo ra một mạng lưới thông tin phong phú và kịp thời.





*Hình 1.6: Tổ chức Khí tượng Thế giới WMO*

- Công nghệ di động và Internet: Ứng dụng di động và các dịch vụ trực tuyến cung cấp dự báo thời tiết tức thì cho người dùng trên khắp thế giới.

### **1.1.3. Tầm quan trọng.**

- An toàn công cộng: Cảnh báo sớm về các hiện tượng thời tiết nguy hiểm như bão, lũ, và sóng nhiệt có thể cứu sống nhiều người và giảm thiểu thiệt hại tài sản.
- Kinh tế: Các ngành nông nghiệp, hàng không, vận tải, và du lịch phụ thuộc nhiều vào dự báo thời tiết chính xác để lên kế hoạch và thực hiện hoạt động hàng ngày.
- Đời sống hàng ngày: Giúp mọi người lên kế hoạch cho các hoạt động ngoài trời, đảm bảo sức khỏe và sự thoải mái.

### **1.1.4. Các thành phần của Bài toán Dự báo Thời tiết.**

#### **▪ Thu thập dữ liệu:**

- Vệ tinh: Cung cấp dữ liệu về mây, bề mặt trái đất và khí quyển từ không gian.
- Radar: Theo dõi các hiện tượng thời tiết như mưa, bão.
- Trạm khí tượng: Ghi nhận dữ liệu tại mặt đất về nhiệt độ, độ ẩm, áp suất khí quyển, tốc độ và hướng gió.
- Phao biển: Thu thập dữ liệu thời tiết và hải dương học từ các đại dương.

#### **▪ Xử lý và phân tích dữ liệu:**

- Xử lý dữ liệu thô: Dữ liệu từ các nguồn khác nhau cần được làm sạch và chuẩn hóa để đảm bảo tính chính xác.
- Phân tích dữ liệu: Sử dụng các công cụ phân tích để hiểu rõ xu hướng và mô hình thời tiết hiện tại.

#### **▪ Thu thập dữ liệu:**

- Mô hình toán học: Sử dụng các phương trình toán học mô phỏng sự thay đổi của khí quyển.
- Mô hình thống kê: Áp dụng các phương pháp thống kê để dự đoán dựa trên dữ liệu quá khứ.
- Mô hình học máy: Sử dụng thuật toán học máy để phát hiện mẫu và xu hướng

trong dữ liệu lớn.

▪ **Đánh giá và cải tiến:**

- Kiểm tra độ chính xác: So sánh dự báo với kết quả thực tế để đánh giá độ chính xác của mô hình.
- Cải tiến mô hình: Dựa trên kết quả đánh giá, điều chỉnh và cải tiến mô hình để nâng cao độ chính xác của dự báo.

**1.1.5. Thách thức trong Dự báo Thời tiết.**

- Độ phức tạp của khí quyển: Khí quyển là một hệ thống động lực phức tạp với nhiều biến số và tương tác phi tuyến, làm cho việc dự báo trở nên khó khăn.
- Thiếu dữ liệu: Một số khu vực, đặc biệt là các vùng biển và các vùng xa xôi, thiếu các trạm quan trắc, gây ra lỗ hổng dữ liệu.
- Biến đổi khí hậu: Sự thay đổi khí hậu toàn cầu làm cho các mẫu thời tiết trở nên khó dự đoán hơn và có thể làm giảm độ chính xác của các mô hình dự báo hiện tại.

**1.2 Mô tả về cách thức Dự báo Thời tiết**

Có nhiều phương pháp dự báo thời tiết khác nhau, mỗi phương pháp đều có những ưu điểm và nhược điểm riêng.

**1.2.1 Các phương pháp dự báo thời tiết.**

**Phương pháp truyền thống (Climatology):** Dự báo dựa trên các mẫu khí hậu lịch sử. Ưu điểm là đơn giản, nhược điểm là không chính xác cho các tình huống thời tiết đột xuất.

**Phương pháp thống kê (Statistical Methods):** Sử dụng các mô hình thống kê dựa trên dữ liệu quá khứ. Ưu điểm là có thể xử lý nhiều dữ liệu, nhược điểm là cần dữ liệu lịch sử phong phú và không phản ứng nhanh với biến đổi khí hậu.

**Phương pháp số (Numerical Weather Prediction - NWP):** Sử dụng các mô hình toán học và máy tính mạnh để mô phỏng khí quyển. Ưu điểm là chính xác hơn và có thể dự báo dài hạn, nhược điểm là phức tạp và tốn kém về tài nguyên máy tính.

**Phương pháp học máy và học sâu (Machine Learning & Deep Learning):** Sử dụng các thuật toán để phân tích dữ liệu và dự báo. Ưu điểm là khả năng học từ dữ liệu lớn, nhược điểm là cần dữ liệu lớn và phức tạp trong việc thiết lập và đào tạo mô hình.

Phương pháp	Ưu điểm	Nhược điểm
Truyền thống (Climatology)	Đơn giản	Không chính xác với thời tiết đột xuất
Thống kê (Statistical)	Xử lý nhiều dữ liệu	Cần dữ liệu lịch sử phong phú
Số (NWP)	Chính xác hơn, dự báo dài hạn	Phức tạp, tốn kém về tài nguyên máy tính
Học máy & Học sâu (ML & DL)	Học từ dữ liệu lớn	Cần dữ liệu lớn, thiết lập và đào tạo phức tạp

*Bảng 1. Bảng so sánh các phương pháp dự báo*

### **1.2.2 Phương pháp lựa chọn.**

Dựa trên các ưu điểm và nhược điểm của từng phương pháp, bài toán này sẽ sử dụng Phương pháp học máy và học sâu (Machine Learning & Deep learning). Phương pháp này cho phép sử dụng các thuật toán hiện đại để học từ dữ liệu lớn, phát hiện các mẫu phức tạp và cung cấp dự báo chính xác và linh hoạt hơn so với các phương pháp truyền thống và thống kê.

#### **▪ *Ưu Điểm Của Phương Pháp Học Máy Và Học Sâu.***

- Khả năng học từ dữ liệu lớn: Các mô hình có thể xử lý và học từ khối lượng lớn dữ liệu thời tiết, giúp cải thiện độ chính xác của dự báo.
- Phát hiện các mẫu phức tạp: Các thuật toán có khả năng phát hiện các mẫu và quan hệ phức tạp trong dữ liệu mà các phương pháp truyền thống có thể bỏ qua.
- Tự động cải tiến: Hệ thống học máy có thể tự động cải thiện mô hình dự báo theo thời gian khi có thêm dữ liệu mới.
- Linh hoạt và đa dạng: Có nhiều loại mô hình và thuật toán khác nhau, cho phép lựa chọn và kết hợp các phương pháp phù hợp nhất cho từng loại dữ liệu và điều kiện khí hậu cụ thể.

#### **▪ *Nhược Điểm Của Phương Pháp Học Máy Và Học Sâu.***

- Cần nhiều dữ liệu: Để đạt được hiệu suất tốt, các mô hình cần một lượng lớn dữ liệu huấn luyện.
- Phức tạp trong thiết lập và huấn luyện: Quá trình xây dựng và huấn luyện mô hình đòi hỏi kiến thức sâu về cả khoa học dữ liệu và khí tượng học.
- Yêu cầu tài nguyên tính toán: Huấn luyện các mô hình phức tạp thường tốn nhiều tài nguyên tính toán và thời gian.
- Khả năng giải thích: Một số mô hình có thể hoạt động như "hộp đen", khó khăn trong việc giải thích và hiểu rõ cách thức mà mô hình đưa ra dự báo.

#### **▪ *Tại Sao Chọn Phương Pháp Học Máy Và Học Sâu Cho Bài Toán Đây?***

- Độ chính xác cao: Nhờ khả năng xử lý và học từ dữ liệu lớn, các mô hình có thể cung cấp dự báo với độ chính xác cao hơn.
- Khả năng tự động cải thiện: Các mô hình cho phép hệ thống dự báo tự động học từ các dữ liệu mới và cải thiện chất lượng dự báo theo thời gian.
- Linh hoạt: Có thể áp dụng cho nhiều loại dữ liệu và điều kiện thời tiết khác nhau, phù hợp với tính phức tạp và biến đổi của thời tiết.

## Chương 2 – TỔNG QUAN KIẾN THỨC

### 2.1 Giới thiệu chung về Machine Learning

#### 2.1.1 Định nghĩa.

Machine Learning là quá trình cho phép máy tính sử dụng dữ liệu để tự cải thiện hiệu suất của mình. Không giống như các hệ thống được lập trình cứng nhắc để thực hiện các nhiệm vụ cụ thể, các mô hình học máy có khả năng tự điều chỉnh và tối ưu hóa dựa trên thông tin đầu vào mà chúng nhận được. Điều này giúp cho các hệ thống trở nên linh hoạt hơn và có khả năng ứng dụng trong nhiều tình huống phức tạp.

#### 2.1.2 Các loại học máy.

##### ▪ *Supervised Learning (Học có giám sát).*

Đây là phương pháp học mà trong đó mô hình được huấn luyện trên một tập dữ liệu có nhãn. Điều này có nghĩa là dữ liệu đầu vào đi kèm với các đầu ra mong muốn. Các bài toán phổ biến:

- Phân loại (Classification): Dự đoán nhãn của các quan sát mới dựa trên các đặc trưng đầu vào. Ví dụ: phân loại email thành thư rác hoặc không thư rác.
- Hồi quy (Regression): Dự đoán một giá trị số liên tục dựa trên các đặc trưng đầu vào. Ví dụ: dự đoán giá nhà dựa trên diện tích và số phòng.
- Các thuật toán phổ biến: Hồi quy tuyến tính (Linear Regression), Hồi quy logistic (Logistic Regression), Cây quyết định (Decision Trees), Rừng ngẫu nhiên (Random Forest), Máy vector hỗ trợ (Support Vector Machines - SVM).

##### ▪ *Unsupervised Learning (Học không giám sát).*

Trong học không giám sát, mô hình được huấn luyện trên một tập dữ liệu không có nhãn, nghĩa là chỉ có các đầu vào mà không có đầu ra mong muốn. Các bài toán phổ biến:

- Phân cụm (Clustering): Nhóm các quan sát thành các nhóm (clusters) sao cho các quan sát trong cùng một nhóm tương tự nhau hơn so với các quan sát ở nhóm khác. Ví dụ: phân nhóm khách hàng có hành vi mua sắm tương tự.
- Giảm chiều (Dimensionality Reduction): Giảm số lượng đặc trưng trong dữ liệu để đơn giản hóa mô hình mà vẫn giữ được các thông tin quan trọng. Ví dụ: giảm số chiều của dữ liệu hình ảnh.

Các thuật toán phổ biến: K-means, Hierarchical Clustering, Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE).

##### ▪ *Reinforcement Learning (Học tăng cường).*

- Học tăng cường là phương pháp học mà trong đó mô hình học bằng cách tương tác với môi trường và nhận phản hồi dưới dạng phần thưởng hoặc hình phạt.
- Mô hình học thông qua việc thử nghiệm và lỗi để tối ưu hóa hành động nhằm đạt được phần thưởng cao nhất.

- Các ứng dụng phổ biến: Huấn luyện robot, chơi các trò chơi như cờ vua hoặc cờ vây, điều khiển tự động.
- Các thuật toán phổ biến: Q-learning, Deep Q-Networks (DQN), Policy Gradient Methods, Actor-Critic Methods.

### 2.1.3 Quy trình chung của Machine Learning.

**Thu thập dữ liệu (Data Collection):** Đây là bước đầu tiên và quan trọng nhất. Chất lượng và số lượng dữ liệu sẽ ảnh hưởng trực tiếp đến hiệu suất của mô hình. Dữ liệu có thể được thu thập từ nhiều nguồn khác nhau như cơ sở dữ liệu, cảm biến, trang web, hoặc nhập liệu bằng tay.

**Tiền xử lý dữ liệu (Data Preprocessing):** Xử lý dữ liệu thiếu (Missing Data Handling), Chuẩn hóa dữ liệu (Normalization/Standardization), Biến đổi và tạo đặc trưng (Feature Engineering), Phân chia dữ liệu (Train/Test Split)

**Chọn mô hình (Model Selection):** Lựa chọn thuật toán hoặc mô hình phù hợp với bài toán và dữ liệu. Các mô hình khác nhau có ưu điểm và nhược điểm riêng, và không có một mô hình nào là tốt nhất cho tất cả các bài toán.

**Huấn luyện mô hình (Model Training):** Sử dụng tập dữ liệu huấn luyện để xây dựng mô hình. Quá trình huấn luyện liên quan đến việc tối ưu hóa các tham số của mô hình để giảm thiểu lỗi dự đoán trên tập huấn luyện.

**Đánh giá mô hình (Model Evaluation):** Sử dụng tập dữ liệu kiểm tra để đánh giá hiệu suất của mô hình. Các chỉ số đánh giá phổ biến bao gồm độ chính xác (accuracy), độ nhạy (recall), độ chính xác (precision), F1-score, và AUC-ROC.

**Điều chỉnh siêu tham số (Hyperparameter Tuning):** Tìm kiếm các giá trị tối ưu cho các siêu tham số của mô hình để cải thiện hiệu suất. Các phương pháp điều chỉnh siêu tham số phổ biến bao gồm Grid Search và Random Search.

**Triển khai mô hình (Model Deployment):** Đưa mô hình vào sử dụng trong môi trường thực tế, nơi nó có thể đưa ra các dự đoán hoặc quyết định dựa trên dữ liệu mới.

**Giám sát và bảo trì (Monitoring and Maintenance):** Theo dõi hiệu suất của mô hình sau khi triển khai và cập nhật mô hình khi cần thiết để duy trì hiệu suất cao.

### 2.1.4 Ứng dụng của Machine Learning.

Lĩnh vực	Ứng dụng thực tiễn
Y tế (Healthcare)	Chẩn đoán bệnh từ hình ảnh y khoa như MRI và X-ray. Dự đoán sự bùng phát dịch bệnh. Phát triển thuốc mới và cá nhân hóa liệu pháp điều trị.
Tài chính (Finance)	Dự đoán rủi ro tín dụng và xác định các khoản vay có khả năng vỡ nợ. Phát hiện gian lận trong giao dịch. Giao dịch tự động và quản lý danh mục đầu tư.
Thương mại điện tử (E-commerce)	Gợi ý sản phẩm dựa trên hành vi mua sắm của khách hàng. Phân tích hành vi khách hàng để tối ưu hóa chiến lược marketing. Quản lý chuỗi cung ứng và tối ưu hóa kho hàng.

Ô tô (Automotive)	Xe tự lái và hệ thống hỗ trợ lái xe. Bảo trì dự đoán để ngăn ngừa hỏng hóc. Tối ưu hóa lộ trình và quản lý đội xe.
Giải trí (Entertainment)	Gợi ý phim, nhạc dựa trên sở thích người dùng. Tạo nội dung tự động như kịch bản phim, âm nhạc. Phân tích cảm xúc và thị hiếu của khán giả.

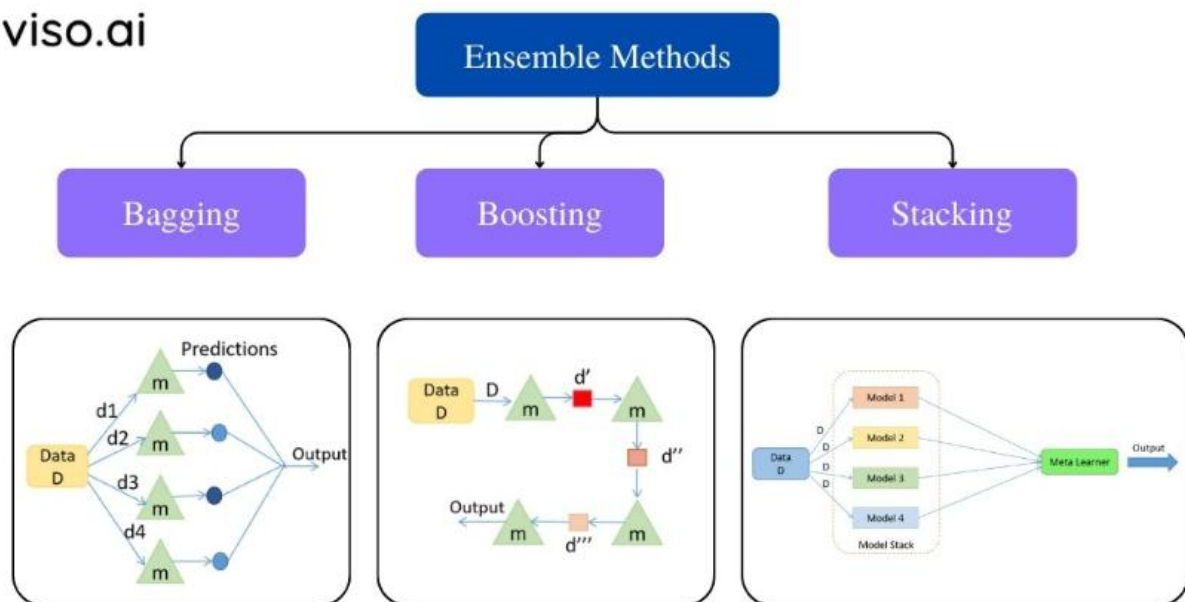
*Bảng 2. Ứng dụng của Machine Learning*

## 2.2 Phương pháp học Ensemble

Học ensemble (ensemble learning) là kỹ thuật kết hợp nhiều mô hình học máy (gọi là mô hình con) để tạo ra một mô hình tổng hợp có hiệu suất tốt hơn từng mô hình riêng lẻ. Hai hướng tiếp cận phổ biến là:

- Bagging (Bootstrap Aggregating): giảm phương sai (variance).
- Boosting: giảm độ lệch (bias).

viso.ai

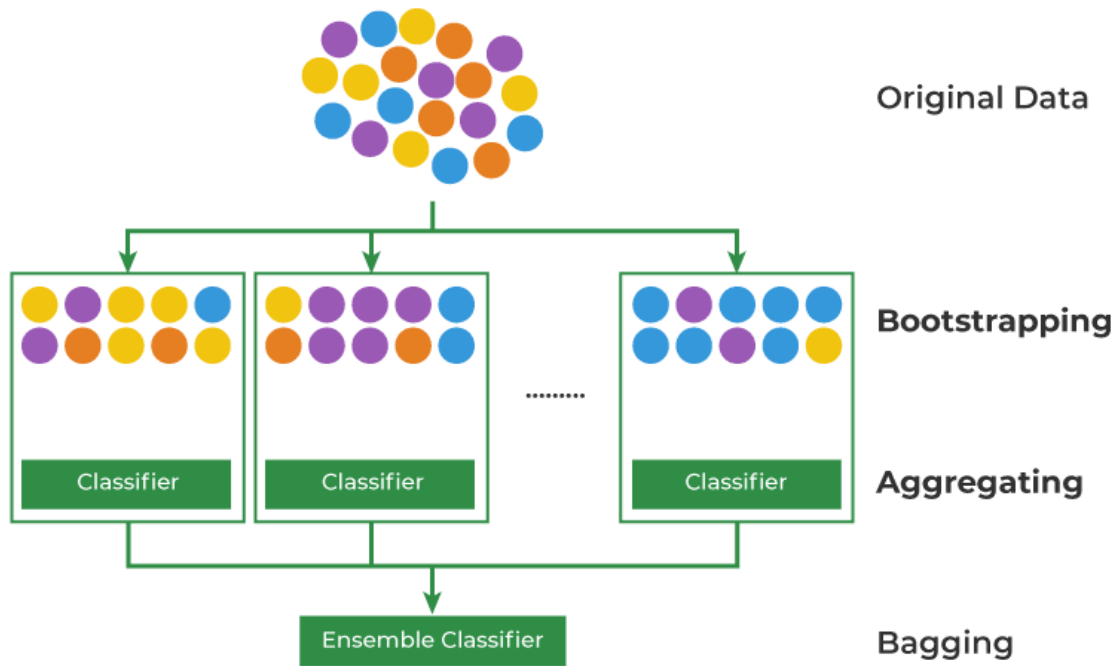


*Hình 2.1: Ensemble Learning*

### 2.2.1 Bagging Classifier

#### a. Định nghĩa.

Bagging (viết tắt của Bootstrap Aggregating) là kỹ thuật học ensemble nhằm cải thiện độ ổn định và độ chính xác của các mô hình học máy bằng cách kết hợp nhiều mô hình con được huấn luyện độc lập trên các tập dữ liệu con được tạo ngẫu nhiên từ dữ liệu gốc.



Hình 2.2: Bagging Classifier

#### b. Nguyên lý hoạt động.

Giả sử tập huấn luyện ban đầu có NNN mẫu:  $D = \{(x_1, y_1), \dots, (x_N, y_N)\}$

Bagging tạo MMM tập con  $D_1, D_2, \dots, D_N$  bằng lấy mẫu có hoàn lại (bootstrap sampling).

Với mỗi tập  $D_i$ , huấn luyện một mô hình  $h_i(x)$ .

Dự đoán đầu ra của mô hình tổng hợp:

- Phân loại (bỏ phiếu đa số):

$$H(x) = \text{mode}(h_1(x), h_2(x), \dots, h_M(x))$$

- Hồi quy (trung bình):

$$H(x) = \frac{1}{M} \sum_{i=1}^M h_i(x)$$

#### c. Ưu nhược điểm.

##### ▪ Ưu điểm:

- Giảm phương sai của mô hình
- Giảm khả năng overfitting.

##### ▪ Nhược điểm:

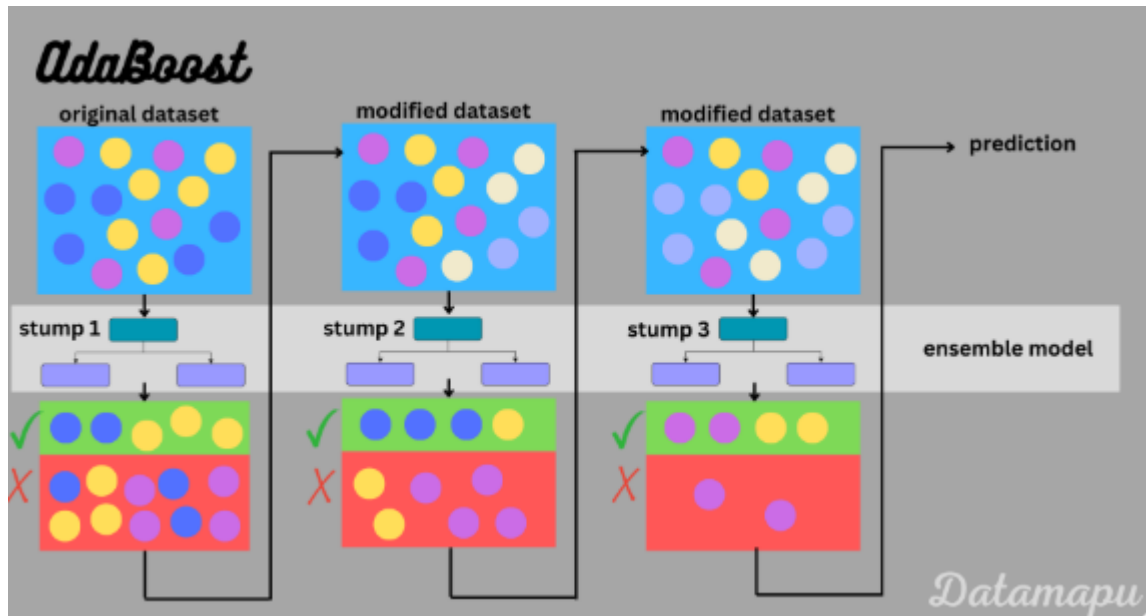
- Không giảm được độ lệch (bias) nếu mô hình con vốn dĩ kém.
- Tốn tài nguyên tính toán khi số lượng mô hình con lớn.

### 2.2.2 AdaBoost Classifier

#### a. Định nghĩa.

AdaBoost (Adaptive Boosting) là thuật toán boosting đầu tiên phổ biến, kết hợp

hiệu quả của mô hình yếu thành một mô hình mạnh bằng cách huấn luyện tuần tự và điều chỉnh trọng số cho các mẫu bị phân loại sai.



Hình 2.3: AdaBoost Classifier

#### b. Nguyên lý hoạt động.

Khởi tạo trọng số  $w_i = \frac{1}{N}$  cho mỗi mẫu  $(x_i, y_i)$ .

Lặp lại trong  $T$  vòng (số mô hình con):

- Huấn luyện mô hình  $h_t(x)$  để tối thiểu hóa sai số có trọng số:

$$\epsilon_t = \sum_{i=1}^N w_i \cdot II(h_t(x_i) \neq y_i)$$

- Tính trọng số của mô hình:

$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)$$

- Cập nhật trọng số mẫu:

$$w_i \leftarrow w_i \cdot \exp(-\alpha_t y_t h_t(x_i))$$

- Sau đó chuẩn hóa lại  $w_i$  để tổng bằng 1.

Dự đoán cuối cùng:

$$H(x) = \text{sign} \left( \sum_{t=1}^T -\alpha_t h_t(x) \right)$$

#### c. Ưu nhược điểm.

##### ▪ Ưu điểm:

- Hiệu quả với các mô hình con đơn giản (như cây nông).
- Cải thiện độ chính xác mô hình tổng thể.

##### ▪ Nhược điểm:

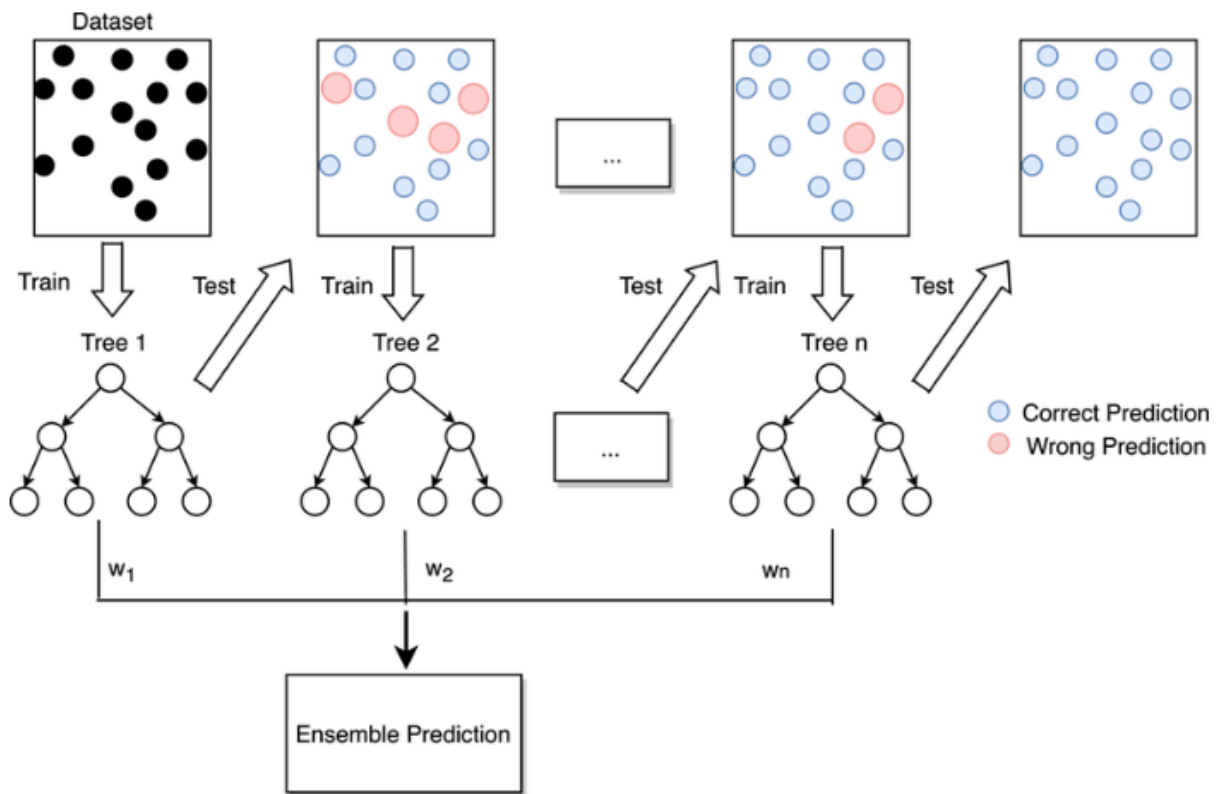


- Nhạy cảm với nhiễu và outliers.
- Không phù hợp khi dữ liệu quá nhiều nhiễu hoặc mất cân bằng nghiêm trọng.

### 2.2.3 Gradient Boosting Classifier

#### a. Định nghĩa.

Gradient Boosting Classifier là một thuật toán học tăng cường (boosting) thuộc họ ensemble learning, kết hợp tuần tự nhiều mô hình học yếu (thường là cây quyết định nông) để tạo thành một mô hình mạnh hơn. Mỗi mô hình sau được huấn luyện nhằm giảm lỗi (residual) còn lại của toàn bộ mô hình trước đó, thông qua việc tối ưu gradient của hàm mất mát.



Hình 2.4: Gradient Boosting Classifier

#### b. Nguyên lý hoạt động.

##### Giả định:

- Tập dữ liệu huấn luyện:  $D = \{(x_i, y_i)\}_{i=1}^N$
- Mô hình dự đoán tại vòng lặp thứ  $t$ :  $F_t(x)$
- Hàm mất mát:  $L(y, F(x))$  (ví dụ: log loss cho phân loại)

##### Bước 1 – Khởi tạo mô hình đầu tiên:

- Chọn một giá trị khởi tạo:

$$F_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$$

- Đối với bài toán phân loại nhị phân, đây thường là log-odds:

$$F_0(x) = \frac{1}{2} \log \left( \frac{p}{1-p} \right), \text{ với } p = \frac{1}{N} \sum_{i=1}^N y_i$$

**Bước 2 – Với mỗi vòng boosting  $t = 1, 2, \dots, T$ :**

- Tính gradient âm (residual):

$$r_i^{(t)} = - \left[ \frac{\partial L(y_i, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)} \right]$$

Đây là giá trị mà mô hình mới cần học tại điểm  $x_i$ .

- Huấn luyện mô hình học yếu  $h_t(x)$  để khớp với  $r_i^{(t)}$ :

$$h_t(x) \approx r_i^{(t)}$$

- Tìm hệ số  $\gamma_t$  tốt nhất (optional – line search):

$$\gamma_t = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, F_{t-1}(x_i) + \gamma h_t(x_i))$$

- Cập nhật mô hình:

$$F_t(x) = F_{t-1}(x) + \eta \cdot \gamma_t h_t(x)$$

Trong đó  $\eta$  là learning rate (tốc độ học, thường  $0 < \eta \leq 0.1$ )

**Bước 3 – Dự đoán:**

- Với bài toán phân loại nhị phân:

$$\hat{y} = \text{sign}(F_T(x)) \text{ hoặc } P(y = 1|x) = \frac{1}{1 + e^{-2F_T(x)}}$$

- Với phân loại đa lớp, áp dụng nguyên lý tương tự với *softmax*.

### c. Ưu nhược điểm.

#### ▪ **Ưu điểm:**

- Hiệu quả cao với các tập dữ liệu vừa và lớn.
- Tùy biến linh hoạt: có thể dùng nhiều loại hàm mất mát (log-loss, exponential loss, MSE...).
- Có khả năng giảm bias và variance nếu tinh chỉnh tốt.
- Là nền tảng cho nhiều mô hình mạnh như XGBoost, LightGBM, CatBoost.

#### ▪ **Nhược điểm:**

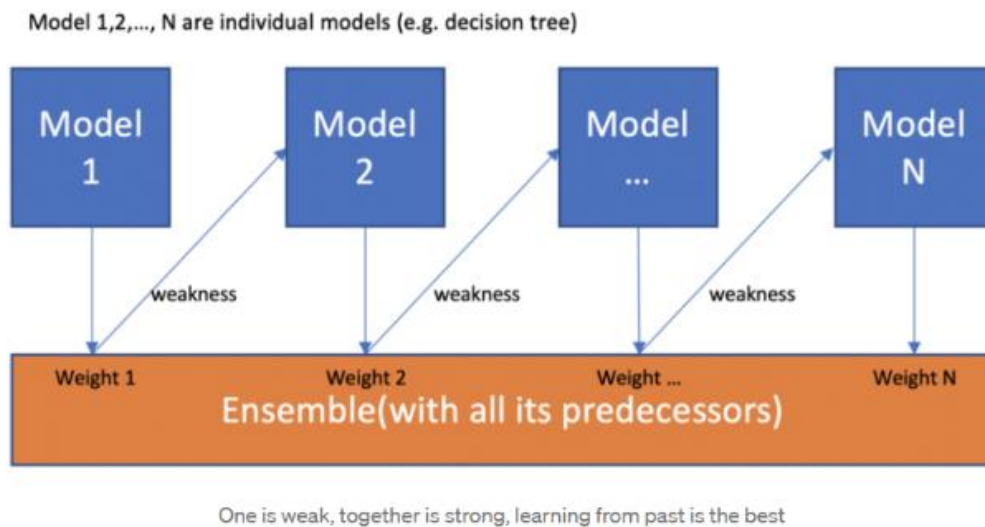
- Dễ overfitting nếu số vòng boosting quá nhiều hoặc cây quá sâu.

- Huấn luyện chậm, do bản chất tuần tự (không thể huấn luyện song song các mô hình con).
- Nhạy cảm với nhiễu trong dữ liệu.
- Cần nhiều công sức để tinh chỉnh siêu tham số (số cây, learning rate, max\_depth...)

## 2.2.4 Hist Gradient Boosting Classifier

### a. Định nghĩa.

HistGradientBoostingClassifier là một phiên bản tối ưu hóa của Gradient Boosting trong thư viện scikit-learn, sử dụng kỹ thuật băm dữ liệu thành các histogram để tăng tốc độ huấn luyện và giảm bộ nhớ.



Hình 2.5: Hist Gradient Boosting Classifier

### b. Nguyên lý hoạt động (so với Gradient Boosting).

Thay vì sử dụng toàn bộ các giá trị đặc trưng, HistGradientBoosting sử dụng **binned features**:

- Mỗi đặc trưng được chia thành  $k$  bins (thường 255).
- Các mẫu được gán vào bin tương ứng → giảm độ phức tạp khi tìm điểm chia tốt nhất trong cây.

Sau khi binning:

- Dữ liệu được ánh xạ:  $x_i \rightarrow b(x_i) \in \{1, 2, \dots, k\}$
- Xây dựng cây nhị phân tối ưu hóa trên histogram, giữ nguyên nguyên lý của Gradient Boosting ở bước tính residual và cập nhật mô hình.

Các bước còn lại tương tự Gradient Boosting, nhưng tính toán nhanh hơn nhờ vào histogram.

### c. Ưu nhược điểm.

#### ▪ Ưu điểm:

- Nhanh hơn và tiết kiệm bộ nhớ hơn so với GradientBoosting truyền thống.

- Tích hợp tốt trong scikit-learn, hỗ trợ early stopping và xử lý missing values.

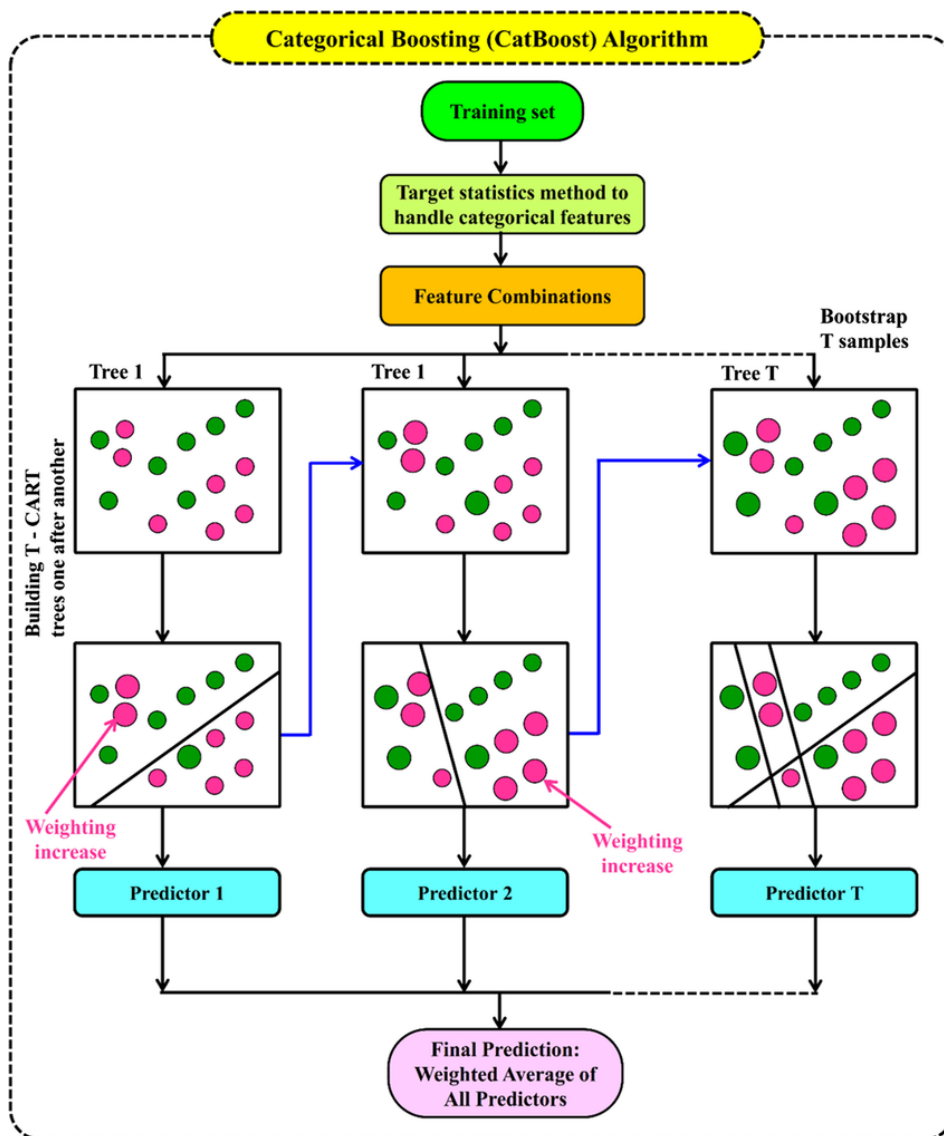
▪ **Nhược điểm:**

- Giảm độ chính xác nếu số lượng bucket quá ít.
- Không xử lý tốt dữ liệu phân loại (categorical) mà không cần mã hóa.

## 2.2.5 CatBoost Classifier

### a. Định nghĩa.

CatBoost là một thuật toán boosting hiện đại được phát triển bởi Yandex, thiết kế để xử lý hiệu quả dữ liệu có nhiều đặc trưng phân loại (categorical features) và tránh overfitting.



Hình 2.6: CatBoost Classifier

### b. Nguyên lý hoạt động (so với Gradient Boosting).

**Xử lý đặc trưng phân loại (categorical features):**

- Sử dụng Target Statistics có trật tự (Ordered Target Encoding) để tránh rò rỉ thông tin:

$$\text{Encoded}(x_i) = \frac{\sum_{j < i, x_j = x_i} y_j + a \cdot p}{\sum_{j < i, x_j = x_i} 1 + a}$$

- Trong đó:

$a$  : hệ số điều chỉnh.

$p$  : trung bình toàn cục.

$j < i$  : tránh sử dụng thông tin tương lai .

**Ordered Boosting (khác với boosting truyền thống):**

- Tách dữ liệu huấn luyện thành nhiều tập con ngẫu nhiên.

- Với mỗi tập, mô hình con được huấn luyện mà không nhìn thấy các mẫu nó sẽ đánh giá, giúp giảm overfitting.

**Cập nhật mô hình tổng giống Gradient Boosting:**

$$F_t(x) = F_{t-1}(x) + \eta h_t(x)$$

Với  $\eta$  là learning rate.

### c. Ưu nhược điểm.

#### ▪ **Ưu điểm:**

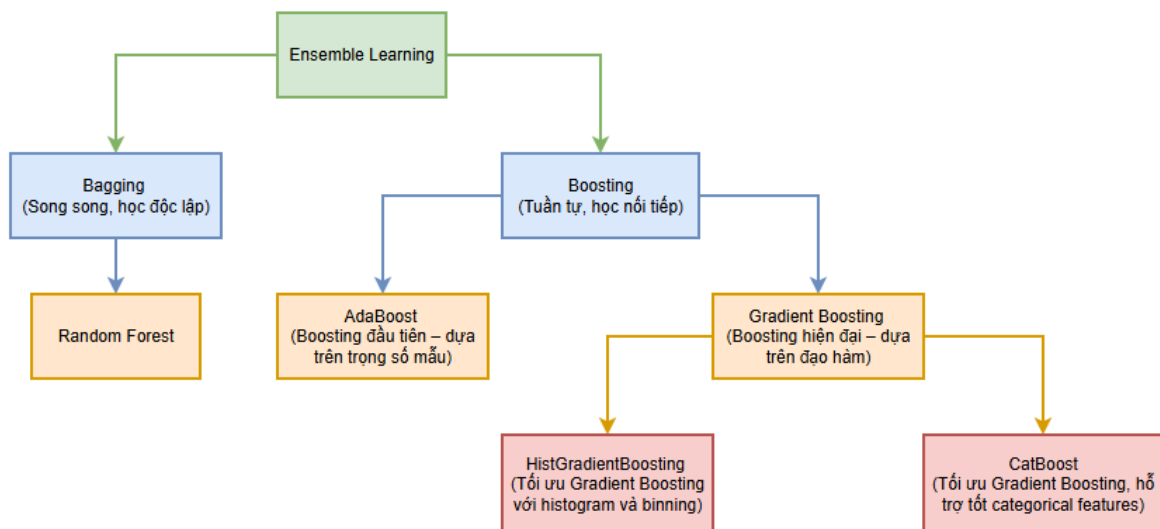
- Hiệu suất cao, ít cần tinh chỉnh
- Hỗ trợ trực tiếp dữ liệu phân loại.
- Giảm overfitting nhờ cơ chế boosting có trật tự.

#### ▪ **Nhược điểm:**

- Phức tạp hơn khi debug và phân tích.
- Dung lượng mô hình có thể lớn nếu không kiểm soát.

## 2.2.6 Tổng Quan Mối Quan Hệ Giữa Các Thuật Toán

### a. Tổng Quan Mối Quan Hệ.



Hình 2.7: Tổng Quan Mối Quan Hệ

## b. So sánh Nguyên lý hoạt động & Môi quan hệ.

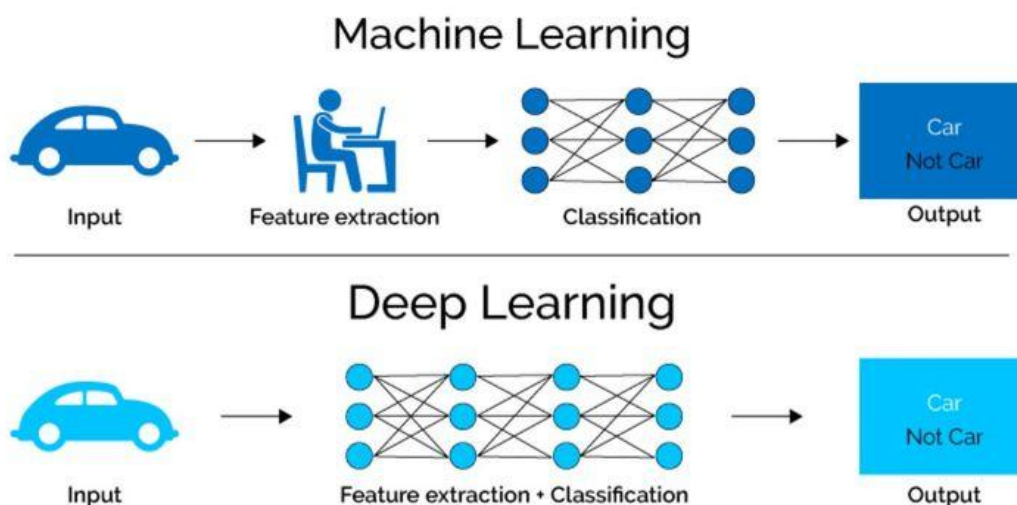
Thuật toán	Dựa trên	Mô hình con	Cách kết hợp	Ưu điểm chính	Là biến thể của
Bagging	Lấy mẫu ngẫu nhiên (bootstrap)	Mạnh như cây quyết định	Trung bình / Bỏ phiếu	Giảm phương sai	-
AdaBoost	Cập nhật trọng số mẫu	Mô hình yếu (stump)	Tổ hợp tuyến tính có trọng số	Giảm bias, đơn giản	Boosting
Gradient Boosting	Tối ưu gradient của hàm mất mát	Mô hình yếu (thường cây nông)	Cộng tích lũy theo gradient	Linh hoạt, mạnh	Boosting
HistGradient Boosting	Như GB nhưng dùng histogram & binning	Mô hình yếu (cây)	Cộng tích lũy	Rất nhanh, tối ưu cho dữ liệu lớn	Gradient Boosting
CatBoost	GB cải tiến + encoding đặc biệt	Cây	Cộng tích lũy	Xử lý categorical tốt, chống overfit	Gradient Boosting

Bảng 3. So Sánh Nguyên lý hoạt động & Môi quan hệ

## 2.3 Mô hình Deep Learning

### 2.3.1 Định nghĩa

Deep Learning (học sâu) là một nhánh của học máy (Machine Learning), sử dụng các mạng nơ-ron nhân tạo (Artificial Neural Networks) với nhiều lớp (layers) để tự động học và trích xuất đặc trưng từ dữ liệu. Nó đã trở thành một trong những phương pháp mạnh mẽ và phổ biến nhất hiện nay, đặc biệt trong các lĩnh vực như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, và hệ thống gợi ý.



Hình 2.8: Sự khác biệt giữa machine learning và deep learning

### 2.3.2 Cấu trúc của Deep Learning

Deep Learning (học sâu) là một nhánh của học máy (Machine Learning), sử dụng các mạng nơ-ron nhân tạo (Artificial Neural Networks) với nhiều lớp (layers) để tự động học và trích xuất đặc trưng từ dữ liệu. Nó đã trở thành một trong những phương pháp mạnh mẽ và phổ biến nhất hiện nay, đặc biệt trong các lĩnh vực như nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, và hệ thống gợi ý.

Deep Learning dựa vào các mạng nơ-ron nhân tạo với nhiều lớp ẩn (hidden layers), nơi mỗi lớp học các đặc trưng phức tạp hơn từ dữ liệu. Mạng nơ-ron nhân tạo bắt chước cách thức hoạt động của não bộ con người, với các nơ-ron và các kết nối giữa chúng. Các lớp trong mạng học các đặc trưng từ dữ liệu ban đầu và chuyển tiếp đến lớp tiếp theo cho đến khi mô hình có thể đưa ra quyết định hoặc dự đoán.

- Đầu vào (Input layer): Nhận dữ liệu thô, chẳng hạn như hình ảnh, văn bản, âm thanh.

- Các lớp ẩn (Hidden layers): Các lớp giữa đầu vào và đầu ra, nơi các nơ-ron xử lý dữ liệu thông qua các phép toán, trích xuất các đặc trưng từ dữ liệu.

- Lớp đầu ra (Output layer): Cung cấp kết quả cuối cùng, ví dụ như phân loại hình ảnh, dự đoán giá trị, v.v.

### 2.3.3 Phân loại

Các loại Mạng Nơ-Ron trong Deep Learning:

**Artificial Neural Networks (ANN):** Là loại mạng nơ-ron đơn giản nhất, bao gồm một lớp đầu vào, một hoặc nhiều lớp ẩn, và một lớp đầu ra. ANN có thể giải quyết các bài toán cơ bản trong phân loại và hồi quy.

**Convolutional Neural Networks (CNN):** Là loại mạng nơ-ron mạnh mẽ cho các bài toán nhận dạng hình ảnh và xử lý ảnh. CNN sử dụng các lớp convolutional để phát hiện các đặc trưng như cạnh, góc, hình dạng trong hình ảnh, từ đó giúp nhận diện các đối tượng.

**Recurrent Neural Networks (RNN):** Được sử dụng chủ yếu cho các bài toán xử lý chuỗi dữ liệu, như phân tích văn bản, nhận dạng giọng nói, và dự báo chuỗi thời gian. RNN có khả năng lưu trữ thông tin từ các bước trước để đưa ra quyết định cho các bước tiếp theo.

**Generative Adversarial Networks (GAN):** Là một mô hình học sâu gồm hai mạng nơ-ron (generator và discriminator) tương tác với nhau. GAN có thể tạo ra dữ liệu mới rất giống với dữ liệu thật, chẳng hạn như tạo ảnh, nhạc hoặc video.

**Transformers:** Mạng nơ-ron hiện đại nhất trong xử lý ngôn ngữ tự nhiên (NLP), được ứng dụng rộng rãi trong các bài toán như dịch máy, phân tích cảm xúc và chatbot. Các mô hình nổi tiếng như BERT và GPT dựa trên kiến trúc Transformer.

### 2.3.4 Quá trình huấn luyện Deep Learning

Deep Learning yêu cầu một lượng lớn dữ liệu và tài nguyên tính toán để huấn luyện mô hình. Quá trình huấn luyện bao gồm các bước cơ bản như:

- Tiền xử lý dữ liệu: Chuẩn hóa, làm sạch và biến đổi dữ liệu để nó phù hợp với mô hình.
- Chọn mô hình: Lựa chọn kiến trúc mạng nơ-ron phù hợp với bài toán (CNN, RNN, v.v.).
- Huấn luyện mô hình: Sử dụng các thuật toán tối ưu (như Gradient Descent) để cập nhật trọng số của mạng nơ-ron.
- Đánh giá mô hình: Kiểm tra hiệu suất của mô hình bằng các chỉ số như accuracy, precision, recall, v.v.

### 2.3.5 Ứng dụng của Deep Learning

**Nhận dạng hình ảnh và video:** Deep Learning đã làm cách mạng trong nhận dạng hình ảnh, cho phép nhận diện vật thể, phân loại hình ảnh, và thậm chí tạo ảnh mới.

**Xử lý ngôn ngữ tự nhiên (NLP):** Các mô hình deep learning như GPT và BERT giúp cải thiện các hệ thống dịch tự động, phân tích cảm xúc, và tạo nội dung tự động.

**Nhận dạng giọng nói và âm thanh:** Các mô hình deep learning có thể được sử dụng trong các ứng dụng nhận dạng giọng nói (ví dụ: trợ lý ảo như Siri, Google Assistant).

**Hệ thống gợi ý:** Deep learning giúp tối ưu hóa các hệ thống gợi ý, ví dụ như trong các dịch vụ như Netflix, YouTube, hoặc Amazon, để dự đoán và đề xuất các nội dung cho người dùng.

**Y tế:** Deep learning có thể được sử dụng để phân tích hình ảnh y tế, chẩn đoán bệnh tự động, và dự đoán nguy cơ mắc bệnh.

**Ô tô tự lái:** Các xe tự lái sử dụng deep learning để nhận dạng các đối tượng trên đường, dự đoán hành vi của người tham gia giao thông và điều khiển xe.

### 2.3.6 Ưu điểm, nhược điểm

#### ▪ **Ưu điểm.**

- Khả năng tự học: Deep Learning có thể học các đặc trưng từ dữ liệu mà không cần sự can thiệp của con người.
- Tính linh hoạt: Deep Learning có thể giải quyết nhiều loại bài toán khác nhau từ nhận dạng hình ảnh đến phân tích ngôn ngữ.
- Hiệu quả trên dữ liệu lớn: Mô hình deep learning có thể hoạt động hiệu quả với lượng dữ liệu rất lớn, điều này làm cho nó rất mạnh trong các ứng dụng thực tế.

#### ▪ **Nhược điểm.**

- Cần nhiều dữ liệu và tài nguyên tính toán: Deep Learning yêu cầu lượng dữ liệu lớn và phần cứng mạnh mẽ (như GPU).
- Khó giải thích: Các mô hình deep learning thường được coi là "hộp đen" vì rất khó giải thích lý do tại sao mô hình lại đưa ra một dự đoán cụ thể.
- Thời gian huấn luyện dài: Việc huấn luyện các mô hình deep learning có thể mất nhiều thời gian, đặc biệt đối với các mạng sâu và dữ liệu lớn.



## Chương 3 – TRIỂN KHAI XÂY DỰNG

### 3.1 Giới thiệu chung về Dataset

Bộ dữ liệu được sử dụng trong đề tài là tập dữ liệu dự báo thời tiết được thu thập từ nền tảng *Visual Crossing Weather*. Dữ liệu chứa các thông tin chi tiết về điều kiện thời tiết theo thời gian, được tổ chức theo từng địa điểm và ngày, phù hợp để phục vụ các bài toán phân tích, dự báo và phân loại thời tiết.

Bộ dữ liệu bao gồm khoảng 33 đặc trưng (features) mô tả các yếu tố khí tượng và môi trường. Cụ thể như:

- Thông tin định danh & thời gian: name, datetime, sunrise, sunset, moonphase, stations.
- Nhiệt độ & cảm nhận: tempmax, tempmin, temp, feelslikemax, feelslikemin, feelslike.
- Độ ẩm & điểm sương: dew, humidity.
- Mưa & tuyết: precip, precipprob, precipcover, preciptype, snow, snowdepth.
- Gió & áp suất: windgust, windspeed, winddir, sealevelpressure.
- Mây & tầm nhìn: cloudcover, visibility.
- Bức xạ & tia cực tím: solarradiation, solarenergy, uvindex.
- Nguy cơ thời tiết cực đoan: severerisk.
- Mô tả điều kiện thời tiết: conditions, description, icon.

Nhóm đã thu thập dữ liệu thời tiết tập trung ở khu vực Châu Á với hơn 50 quốc gia bao gồm 160 thành phố, từ ngày 1/1/2023 tới ngày 31/12/2024.

Dưới đây là những thành phần chính trong dataset và định nghĩa của chúng:

STT	Tên cột	Định nghĩa	Đơn vị
1	name	Tên thành phố	-
2	datetime	Thời gian	-
3	tempmax	Nhiệt độ tối đa	°C
4	tempmin	Nhiệt độ tối thiểu	°C
5	temp	Nhiệt độ (hoặc nhiệt độ trung bình)	°C
6	feelslikemax	Nhiệt độ cảm nhận tối đa	°C
7	feelslikemin	Nhiệt độ cảm nhận tối thiểu	°C
8	feelslike	Nhiệt độ cảm nhận	°C
9	dew	Điểm sương	°C
10	humidity	Độ ẩm tương đối	%
11	precip	Lượng mưa	mm
12	precipprob	Khả năng mưa	%
13	precipcover	Lượng mưa bao phủ	%
14	preciptype	Loại mưa	-

15	snow	Tuyết	cm
16	snowdepth	Độ sâu của tuyết	cm
17	windgust	Tốc độ gió	km/h
18	windspeed	Gió giật	km/h
19	winddir	Hướng gió	Độ
20	sealevelpressure	Áp suất mực nước biển	mb
21	cloudcover	Mây che phủ	%
22	visibility	Khả năng hiển thị (tầm nhìn)	km
23	solarradiation	Bức xạ mặt trời	W/m <sup>2</sup>
24	solarenergy	Năng lượng mặt trời	MJ/m <sup>2</sup>
25	uvindex	Chỉ số UV	-
26	severerisk	Rủi ro nghiêm trọng	-
27	sunrise	Thời gian mặt trời mọc	-
28	sunset	Thời gian mặt trời lặn	-
29	moonphase	Giai đoạn mặt trăng	-
30	conditions	Văn bản ngắn về thời tiết	-
31	description	Mô tả thời tiết trong ngày	-
32	icon	Biểu tượng thời tiết	-
33	stations	Danh sách các nguồn trạm thời tiết	-

Bảng 4. Định nghĩa Dataset

## 3.2 Triển khai xây dựng

### 3.2.1 Tổng quan về tập dữ liệu.

Trong phần này, chúng ta sẽ tải tập dữ liệu và xem xét sơ bộ cấu trúc cũng như nội dung của nó.

Mounted at /content/drive

	name	datetime	tempmax	tempmin	temp	feelslikemax	feelslikemin	feelslike	dew	humidity	...	solarenergy	uvindex	severerisk	sunrise	sunset	moonphase	conditions	description	icon
0	Manama	2023-01-01	21.0	17.9	19.5	21.0	17.9	19.5	12.5	64.2	...	13.6	6.0	10.0	2023-01-01T06:25:21	2023-01-01T16:56:58	0.31	Partially cloudy	Partly cloudy throughout the day.	partly-cloudy-day
1	Manama	2023-01-02	18.6	14.9	16.5	18.6	14.9	16.5	12.9	79.7	...	6.5	3.0	10.0	2023-01-02T06:25:37	2023-01-02T16:57:38	0.34	Rain. Partially cloudy	Partly cloudy throughout the day with rain.	rain
2	Manama	2023-01-03	18.9	14.9	17.1	18.9	14.9	17.1	12.0	71.9	...	10.0	5.0	10.0	2023-01-03T06:25:52	2023-01-03T16:58:19	0.38	Rain. Partially cloudy	Partly cloudy throughout the day with a chance...	rain
3	Manama	2023-01-04	22.2	15.9	18.5	22.2	15.9	18.5	15.3	81.6	...	6.6	3.0	10.0	2023-01-04T06:26:06	2023-01-04T16:59:01	0.41	Rain. Partially cloudy	Partly cloudy throughout the day with rain in ...	rain
4	Manama	2023-01-05	19.9	15.9	18.1	19.9	15.9	18.1	15.9	86.8	...	4.2	2.0	10.0	2023-01-05T06:26:18	2023-01-05T16:59:43	0.45	Rain. Partially cloudy	Partly cloudy throughout the day with a chance...	rain

5 rows × 33 columns

Hình 3.1: Một số dòng đầu tiên của tập dữ liệu

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 116960 entries, 0 to 116959
Data columns (total 33 columns):
#   Column                Non-Null Count  Dtype
---  -
0   name                   116960 non-null object
1   datetime               116960 non-null object
2   tempmax                116960 non-null float64
3   tempmin                116960 non-null float64
4   temp                   116960 non-null float64
5   feelslikemax           116960 non-null float64
6   feelslikemin           116960 non-null float64
7   feelslike               116960 non-null float64
8   dew                    116960 non-null float64
9   humidity                116960 non-null float64
10  precip                  116960 non-null float64
11  precipprob              116960 non-null int64
12  precipcover             116960 non-null float64
13  preciptype              56717 non-null object
14  snow                    116960 non-null float64
15  snowdepth               116952 non-null float64
16  windgust                116960 non-null float64
17  windspeed               116960 non-null float64
18  winddir                 116960 non-null float64
19  sealevelpressure        116960 non-null float64
...
31  icon                    116960 non-null object
32  stations                116960 non-null object
dtypes: float64(23), int64(1), object(9)
memory usage: 29.4+ MB

```

Hình 3.2: Thông tin tóm tắt về dữ liệu

	tempmax	tempmin	temp	feelslikemax	feelslikemin	feelslike	dew	humidity	precip	precipprob	...	windspeed	winddir	sealevelpressure	cloudcover	visibility
count	116960.000000	116960.000000	116960.000000	116960.000000	116960.000000	116960.000000	116960.000000	116960.000000	116960.000000	116960.000000	...	116960.000000	116960.000000	116960.000000	116960.000000	114289.000000
mean	24.725276	14.519738	19.387650	26.288150	14.029444	19.930371	10.867398	64.095253	2.943316	42.391416	...	19.714357	184.775951	1013.204796	46.497929	13.313623
std	12.057044	12.690504	12.156139	14.334943	14.658146	14.257778	12.375333	20.336120	10.522630	49.417915	...	9.583537	104.591586	7.574794	30.369345	7.133651
min	-48.400000	-53.900000	-51.500000	-48.400000	-60.700000	-51.600000	-55.800000	3.500000	0.000000	0.000000	...	0.000000	0.000000	971.800000	0.000000	0.000000
25%	19.000000	7.600000	13.000000	19.000000	6.200000	12.875000	3.100000	51.100000	0.000000	0.000000	...	13.000000	90.500000	1008.400000	20.200000	9.300000
50%	28.200000	18.000000	23.200000	28.500000	18.000000	23.100000	12.000000	68.500000	0.000000	0.000000	...	18.400000	193.300000	1012.300000	46.300000	10.400000
75%	32.800000	24.200000	28.000000	36.700000	24.200000	29.900000	22.000000	80.100000	1.000000	100.000000	...	25.200000	275.300000	1017.600000	71.900000	18.300000
max	51.000000	38.800000	44.700000	70.200000	50.400000	54.600000	29.900000	100.000000	428.578000	100.000000	...	302.000000	360.000000	1061.600000	100.000000	50.000000

8 rows x 24 columns

Hình 3.3: Thống kê tóm tắt

Sau khi kiểm tra dữ liệu xong, phát hiện có 3 đặc trưng có số lượng null rất lớn (precipdtype, visibility, severerisk) và 4 đặc trưng chỉ có rất ít giá trị null (solarradiation, solarenergy, uvindex, snowdepth).

	0
name	0
datetime	0
tempmax	0
tempmin	0
temp	0
feelslikemax	0
feelslikemin	0
feelslike	0
dew	0
humidity	0
precip	0
precipprob	0
precipcover	0
preciptype	60243
snow	0
snowdepth	8
windgust	0
windspeed	0
winddir	0
sealevelpressure	0
cloudcover	0
visibility	2671
solarradiation	1
solarenergy	1
uvindex	1
severerisk	22884
sunrise	0
sunset	0
moonphase	0
conditions	0
description	0
icon	0
stations	0

**dtype:** int64

Hình 3.4: Kiểm tra giá trị null

Đầu tiên với đặc trưng ***preciptype***, ta thay thế giá trị null bằng “***no precipitation***”. Còn với đặc trưng ***severerisk***, ta thay thế giá trị null bằng “***0***”.

```
df['preciptype'].fillna('no precipitation', inplace=True)
df['severerisk'].fillna(0, inplace=True)
```

Hình 3.5: Xử lý null với đặc trưng *preciptype*, *severerisk*

Tiếp theo xử lý giá trị null cho 3 đặc trưng *solarradiation*, *solarenergy*, *snowdepth*. Đối với cả 3 đặc trưng, ta thay thế giá trị null bằng giá trị trung bình trong 3 tháng tại vị trí giá trị null.

```
df['datetime'] = pd.to_datetime(df['datetime'], format='mixed')
df['quarter'] = df['datetime'].dt.to_period('Q')

# Với solarradiation:
df['solarradiation'] = df.groupby(['name', 'quarter'])['solarradiation']\
    .transform(lambda x: x.fillna(round(x.mean(), 1)))

# Với solarenergy:
df['solarenergy'] = df.groupby(['name', 'quarter'])['solarenergy']\
    .transform(lambda x: x.fillna(round(x.mean(), 1)))

# Với snowdepth:
df['snowdepth'] = df.groupby(['name', 'quarter'])['snowdepth']\
    .transform(lambda x: x.fillna(round(x.mean(), 1)))

df.drop(columns='quarter', inplace=True)
```

Hình 3.6: Xử lý null với đặc trưng *solarradiation*, *solarenergy*, *snowdepth*

Tiếp theo xử lý giá trị null cho đặc trưng *visibility*. Đối với đặc trưng này, ta thay thế giá trị null bằng giá trị trung bình trong 3 tháng tại vị trí giá trị null, nếu vẫn còn xuất hiện giá trị null tiếp tục thay thế bằng giá trị trung bình trong 1 năm, nếu vẫn còn null thì thay thế bằng giá trị trung bình theo tên thành phố (tức 2 năm). Cuối cùng null vẫn còn thì ta thay thế bằng giá trị trung bình toàn cột *visibility*.

```
df['datetime'] = pd.to_datetime(df['datetime'], format='mixed')

df['quarter'] = df['datetime'].dt.to_period('Q') # quý
df['year_month'] = df['datetime'].dt.to_period('M') # tháng

# Bước 1: Fill theo name + quarter (3 tháng)
df['visibility'] = df.groupby(['name', 'quarter'])['visibility']\
    .transform(lambda x: x.fillna(round(x.mean(), 1)))

# Bước 2: Fill theo name + year_month (12 tháng)
df['visibility'] = df.groupby(['name', 'year_month'])['visibility']\
    .transform(lambda x: x.fillna(round(x.mean(), 1)))

# Nếu còn null, fill theo name
df['visibility'] = df.groupby('name')['visibility']\
    .transform(lambda x: x.fillna(round(x.mean(), 1)))

# Nếu vẫn còn, fill bằng mean toàn dataset
df['visibility'] = df['visibility'].fillna(round(df['visibility'].mean(), 1))

df.drop(columns=['quarter', 'year_month'], inplace=True)
```

Hình 3.7: Xử lý null với đặc trưng *visibility*

Tiếp đến, ta xử lý giá trị null cho đặc trưng uvindex. Đối với đặc trưng này, ta thay thế giá trị null bằng giá trị phổ biến nhất được tìm thấy, được duyệt theo tên thành phố và trong thời gian khoảng 1 tháng.

```
df['datetime'] = pd.to_datetime(df['datetime'], format='mixed')

df['year_month'] = df['datetime'].dt.to_period('M')

# Xử lý uvindex
def fill_mode(series):
    if series.isnull().all():
        return series # Nếu toàn bộ là null thì để nguyên
    mode_value = series.mode()
    if not mode_value.empty:
        return series.fillna(mode_value.iloc[0]) # Lấy giá trị mode đầu tiên
    else:
        return series

df['uvindex'] = df.groupby(['name', 'year_month'])['uvindex'].transform(fill_mode)

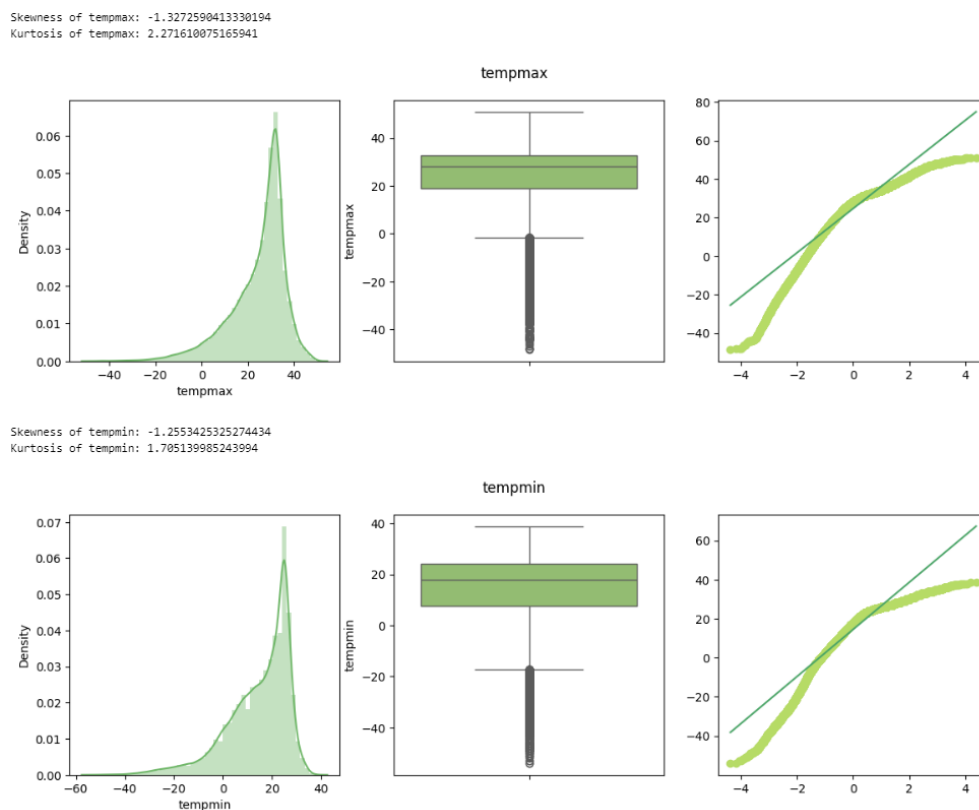
df.drop(columns='year_month', inplace=True)
```

Hình 3.8: Xử lý null với đặc trưng uvindex

### 3.2.2 Phân tích dữ liệu thăm dò (EDA).

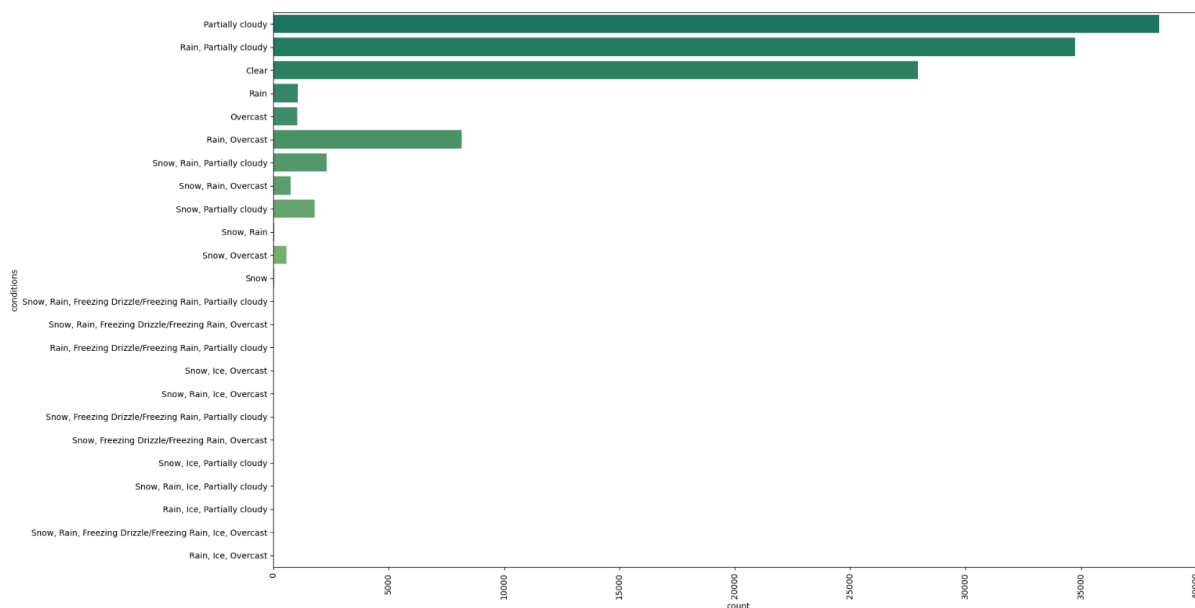
Phân tích dữ liệu khám phá giúp chúng tôi hiểu sự phân bố của các tính năng, mối quan hệ giữa các biến và xác định các mẫu hoặc điểm bất thường. Trong phần này, chúng ta sẽ trực quan hóa và phân tích tập dữ liệu.

Biểu đồ phân phối, biểu đồ hộp (boxplot), và biểu đồ xác suất (probability plot). Với mỗi cột, tính toán skewness và kurtosis và in ra các giá trị này.



Hình 3.9: Biểu đồ phân phối, biểu đồ hộp, và biểu đồ xác suất

Biểu đồ cột Tần suất các loại thời tiết dự đoán được trong đặc trưng *conditions*.



Hình 3.10: Biểu đồ cột Tần suất các loại thời tiết

Trong đặc trưng *conditions* có khoảng hơn 24 kiểu thời tiết được mô tả chi tiết trong dữ liệu, từ các trạng thái cơ bản như "Partially cloudy", "Clear" đến các kiểu phức tạp hơn như "Snow, Rain, Freezing Drizzle/Freezing Rain, Overcast", "Rain, Freezing Drizzle/Freezing Rain, Partially cloudy", hay "Snow, Rain, Freezing Drizzle/Freezing Rain, Partially cloudy".

	count
conditions	
Partially cloudy	38402
Rain, Partially cloudy	34749
Clear	27928
Rain, Overcast	8167
Snow, Rain, Partially cloudy	2311
Snow, Partially cloudy	1798
Rain	1069
Overcast	1049
Snow, Rain, Overcast	756
Snow, Overcast	572
Snow, Rain	53
Snow	53
Snow, Rain, Freezing Drizzle/Freezing Rain, Partially cloudy	13
Snow, Rain, Freezing Drizzle/Freezing Rain, Overcast	8
Snow, Ice, Partially cloudy	6
Snow, Freezing Drizzle/Freezing Rain, Overcast	5
Snow, Rain, Ice, Overcast	4
Snow, Ice, Overcast	4
Snow, Rain, Ice, Partially cloudy	3
Snow, Freezing Drizzle/Freezing Rain, Partially cloudy	3
Rain, Ice, Partially cloudy	3
Rain, Freezing Drizzle/Freezing Rain, Partially cloudy	2
Snow, Rain, Freezing Drizzle/Freezing Rain, Ice, Overcast	1
Rain, Ice, Overcast	1

Hình 3.11: Tần suất xuất hiện của các loại thời tiết

Hướng giải quyết được đề ra ở đây là thực hiện bài toán phân loại nhiều nhãn (multi-label classification), sử dụng 7 thành phần cơ bản nhất để dự đoán, cụ thể là **'Partially cloudy', 'Rain', 'Clear', 'Overcast', 'Snow', 'Freezing Drizzle/Freezing Rain', 'Freezing Drizzle/Freezing Rain'**. Sau đó, đánh nhị phân (0/1) cho từng nhãn, tức là chuyển cột **conditions** từ dạng văn bản (text) thành nhiều cột nhị phân (multi-label binary format) .

```
# Các thành phần cơ bản cần dự đoán
labels = ['Partially cloudy', 'Rain', 'Clear', 'Overcast',
         'Snow', 'Freezing Drizzle/Freezing Rain', 'Ice']

# Với mỗi label, tạo cột mới trong dataframe
for label in labels:
    df[label] = df['conditions'].apply(lambda x: 1 if label in x else 0)
```

Hình 3.12: Multi-label classification

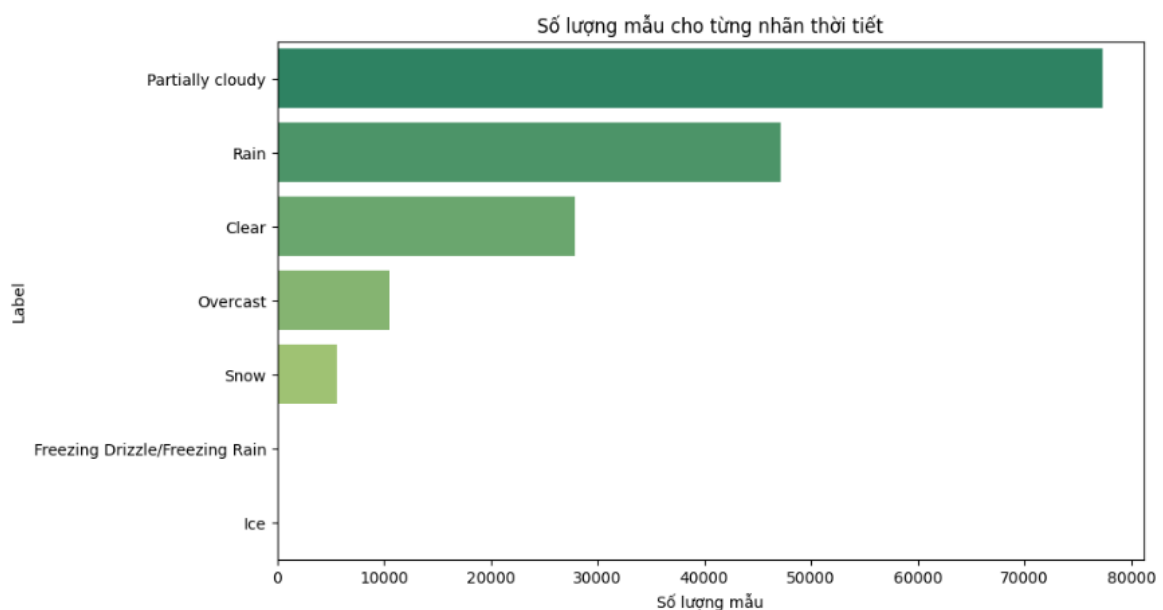
Tính số lượng mẫu cho các nhãn thời tiết bằng cách tính tổng số của từng nhãn.

```
# Tính tổng số lượng 1 cho từng label
label_counts = df[labels].sum()

# In ra
print(label_counts)
```

Partially cloudy	77290
Rain	47148
Clear	27928
Overcast	10567
Snow	5590
Freezing Drizzle/Freezing Rain	32
Ice	22
dtype:	int64

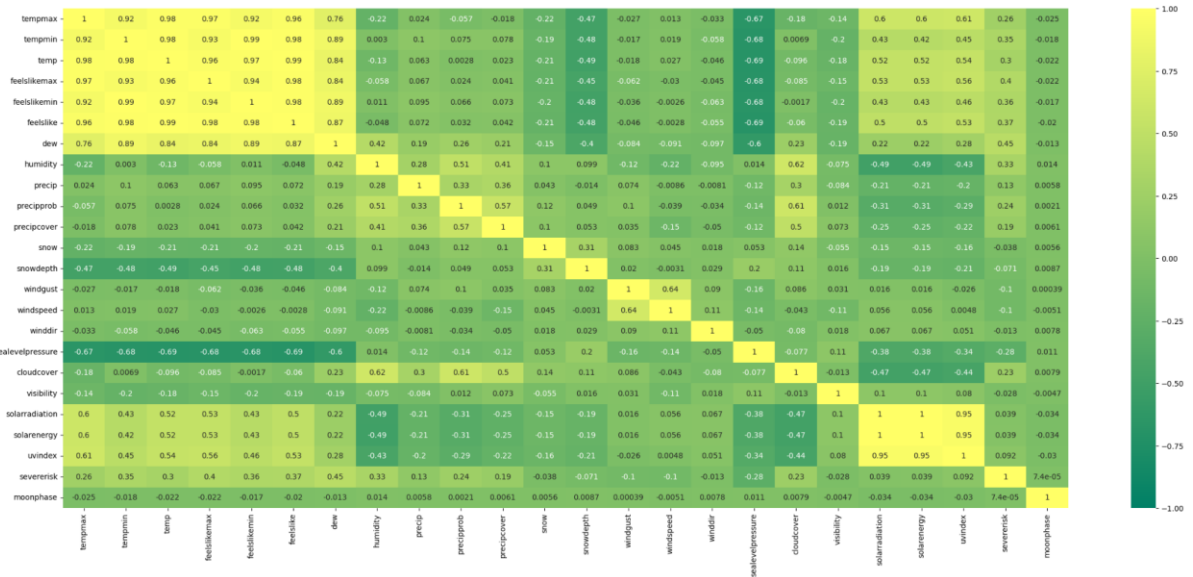
Hình 3.13: Số lượng mẫu cho từng nhãn thời tiết



Hình 3.14: Biểu đồ cột Tần suất xuất hiện của 7 nhãn thời tiết



### Biểu đồ heatmap cho ma trận tương quan.

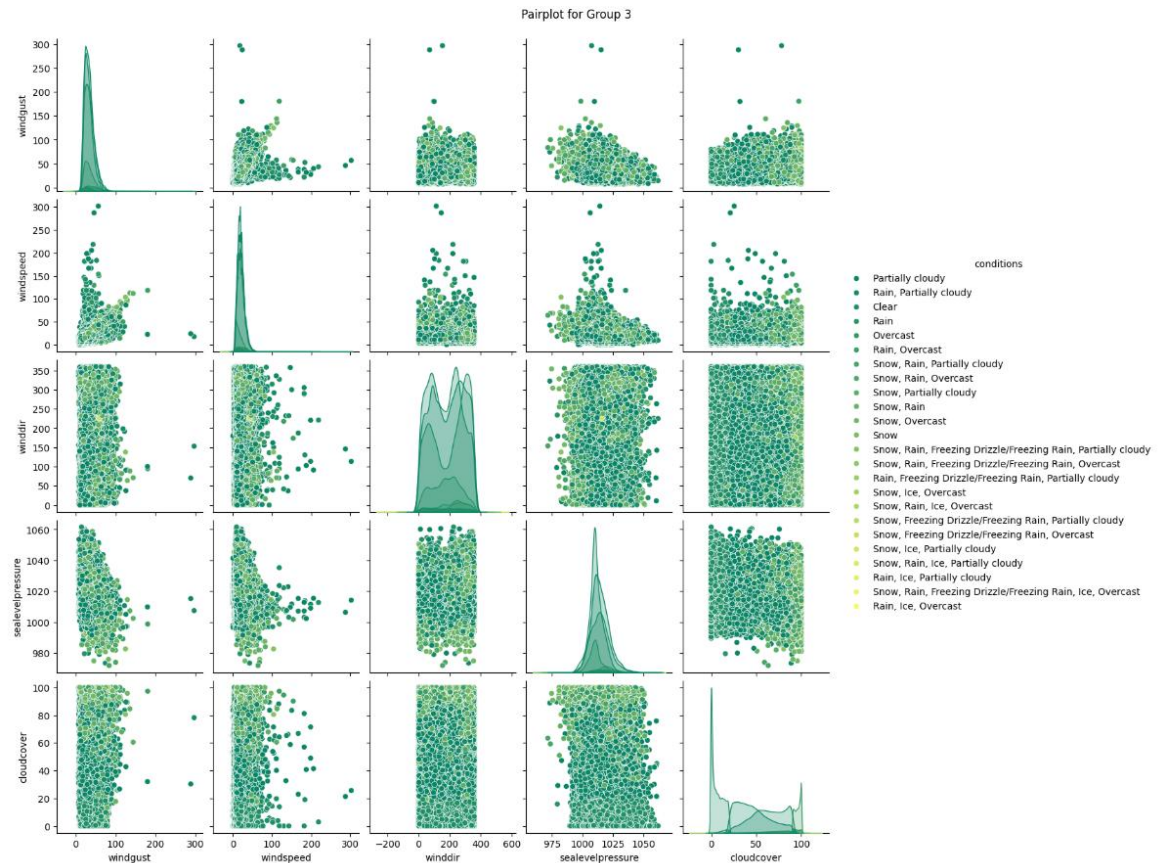


Hình 3.15: Biểu đồ heatmap

Chia 4 nhóm nhỏ để vẽ biểu đồ phân tán giữa các biến và biến mục tiêu **conditions**.

```
group1 = ['tempmax', 'tempmin', 'temp', 'feelslikemax', 'feelslikemin', 'feelslike']
group2 = ['humidity', 'dew', 'precip', 'precipprob', 'precipcover']
group3 = ['windgust', 'windspeed', 'winddir', 'sealevelpressure', 'cloudcover']
group4 = ['visibility', 'solarradiation', 'solarenergy', 'uvindex', 'severerisk', 'moonphase']
```

Hình 3.16: Nhóm biểu đồ phân tán



Hình 3.17: Biểu đồ phân tán cho Group 3

### 3.2.3 Kỹ thuật đặc trưng (Feature Engineering).

#### ▪ *Feature Extraction*

Trích xuất các thành phần thời gian từ đặc trưng *datetime* để tạo ra các cột mới và loại bỏ đặc trưng *datetime*, tương tự với đặc trưng *sunrise*, *sunset*.

#### Feature Extraction

```
df['datetime'] = pd.to_datetime(df['datetime'], format='mixed')

df['year'] = df['datetime'].dt.year
df['month'] = df['datetime'].dt.month
df['day'] = df['datetime'].dt.day

df.drop('datetime',axis=1,inplace=True)
```

```
df['sunrise'] = pd.to_datetime(df['sunrise'],errors='coerce')

df['hour_sunrise'] = df['sunrise'].dt.hour
df['minute_sunrise'] = df['sunrise'].dt.minute
df['second_sunrise'] = df['sunrise'].dt.second

df.drop('sunrise',axis=1,inplace=True)
```

```
df['sunset'] = pd.to_datetime(df['sunset'],errors='coerce')

df['hour_sunset'] = df['sunset'].dt.hour
df['minute_sunset'] = df['sunset'].dt.minute
df['second_sunset'] = df['sunset'].dt.second

df.drop('sunset',axis=1,inplace=True)
```

Hình 3.18: Xử lý cột *datetime*, *sunrise*, *sunset*

#### ▪ *Feature Splitting*

Chọn biến mục tiêu là *conditions* và biến đặc trưng là các đặc trưng khác.

```
features = [
    'tempmax', 'tempmin', 'temp', 'feelslikemax', 'feelslikemin',
    'feelslike', 'dew', 'humidity', 'precip', 'precipprob', 'precipcover',
    'preciptype', 'snow', 'snowdepth', 'windgust', 'windspeed', 'winddir',
    'sealevelpressure', 'cloudcover', 'visibility', 'solarradiation',
    'solarenergy', 'uvindex', 'severerisk', 'moonphase',
    'year', 'month', 'day', 'hour_sunrise', 'minute_sunrise',
    'second_sunrise', 'hour_sunset', 'minute_sunset', 'second_sunset'
]
target_columns = ['Partially cloudy', 'Rain', 'Clean', 'Overcast', 'Snow', 'Freezing Drizzle/Freezing Rain', 'Ice']
```

```
X = df[features]
y = df[target_columns]
```

Hình 3.19: Tạo biến đặc trưng và biến mục tiêu

- **Balancing the target classes**

Cân bằng dữ liệu bằng cách tăng cường tăng cường mẫu cho 2 nhãn hiếm (ít mẫu nhất) là **Freezing Drizzle/Freezing Rain** và **Ice** với mục tiêu ít nhất là 5000 mẫu.

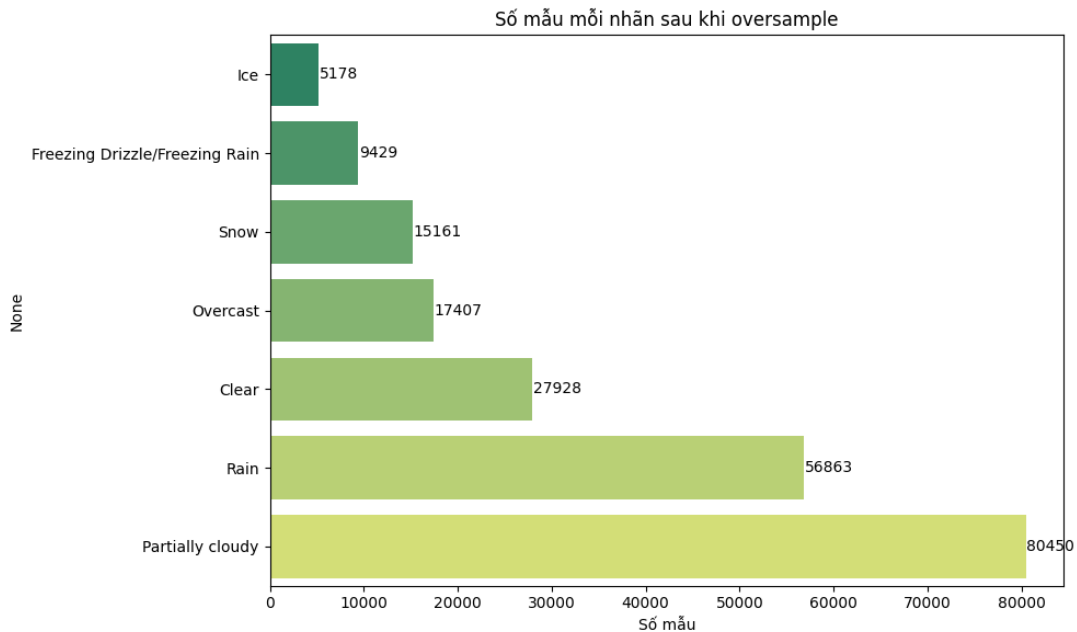
```
def oversample_label(X, y, label_name, target_increase):
    mask = y[label_name] == 1
    X_label = X[mask]
    y_label = y[mask]

    repeats = target_increase // len(X_label) + 1
    X_oversampled = pd.concat([X_label] * repeats, ignore_index=True).iloc[:target_increase]
    y_oversampled = pd.concat([y_label] * repeats, ignore_index=True).iloc[:target_increase]

    X = pd.concat([X, X_oversampled], ignore_index=True)
    y = pd.concat([y, y_oversampled], ignore_index=True)
    return X, y

# Cân bằng dữ liệu cho 2 nhãn hiếm
X, y = oversample_label(X, y, "Freezing Drizzle/Freezing Rain", 5000)
X, y = oversample_label(X, y, "Ice", 5000)
```

Hình 3.20: Cân bằng dữ liệu



Hình 3.21: Biểu đồ cột (countplot) các nhãn sau khi cân bằng

- **Train Test Split**

Chia dữ liệu thành tập huấn luyện và kiểm tra.

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

```
print("Shape of the training set:",X_train.shape)
print("Shape of the testing set:",X_test.shape)
```

```
Shape of the training set: (101568, 34)
Shape of the testing set: (25392, 34)
```

Hình 3.22: Chia dữ liệu thành tập huấn luyện và kiểm tra

### ▪ *Label Encoding*

Mã hóa nhãn trong tập huấn luyện và tập kiểm tra của đặc trưng *preciptype*.

```
encoder = LabelEncoder()
X_train['preciptype'] = encoder.fit_transform(X_train['preciptype'])
X_test['preciptype'] = encoder.transform(X_test['preciptype'])
```

```
encoder.classes_
```

```
array(['freezingrain', 'freezingrain,snow', 'no precipitation', 'rain',
      'rain,freezingrain', 'rain,freezingrain,snow',
      'rain,freezingrain,snow,ice', 'rain,ice', 'rain,snow',
      'rain,snow,ice', 'snow', 'snow,ice'], dtype=object)
```

```
label_mapping = {label: idx for idx, label in enumerate(encoder.classes_)}
label_mapping
```

```
{'freezingrain': 0,
 'freezingrain,snow': 1,
 'no precipitation': 2,
 'rain': 3,
 'rain,freezingrain': 4,
 'rain,freezingrain,snow': 5,
 'rain,freezingrain,snow,ice': 6,
 'rain,ice': 7,
 'rain,snow': 8,
 'rain,snow,ice': 9,
 'snow': 10,
 'snow,ice': 11}
```

Hình 3.23: Mã hóa nhãn tập huấn luyện và kiểm tra cho *preciptype*

### ▪ *Feature Selection*

Kiểm tra các đặc trưng nào quan trọng nhất để từ đó chọn ra các đặc trưng tối ưu cho việc huấn luyện.

```
pipeline = Pipeline(steps=[
    ('constant', DropConstantFeatures()),
    ('duplicate', DropDuplicateFeatures())
])
```

```
X_train = pipeline.fit_transform(X_train)
X_test = pipeline.transform(X_test)
```

```
print("Shape of the training set:", X_train.shape)
print("Shape of the testing set:", X_test.shape)
```

```
Shape of the training set: (101568, 34)
Shape of the testing set: (25392, 34)
```

Hình 3.24: Áp dụng pipeline với 2 tập dữ liệu

Lựa chọn đặc trưng (feature selection) cho bài toán phân loại nhiều nhãn (multi-label classification) bằng cách tính điểm quan trọng (feature score) của từng đặc trưng đối với từng nhãn, sau đó chọn ra top 10 đặc trưng có ảnh hưởng trung bình cao nhất.

```
# Lưu điểm số đặc trưng theo từng nhãn
feature_scores = {}

for column in y_train.columns:
    selector = SelectKBest(score_func=f_classif, k='all') # chọn tất cả để tính điểm
    selector.fit(X_train, y_train[column])
    feature_scores[column] = selector.scores_

# Tạo DataFrame chứa điểm số
score_df = pd.DataFrame(feature_scores, index=X_train.columns)

# Tính điểm trung bình của mỗi đặc trưng qua tất cả các nhãn
score_df['mean_score'] = score_df.mean(axis=1)

# Lấy top 10 đặc trưng có điểm trung bình cao nhất
top_10_features = score_df['mean_score'].sort_values(ascending=False).head(10).index.tolist()

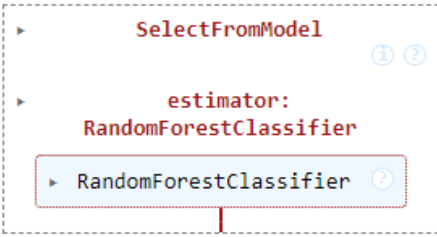
print("Top 10 đặc trưng được chọn:", top_10_features)
```

Top 10 đặc trưng được chọn: ['precipprob', 'cloudcover', 'preciptype', 'snow', 'precipcover', 'humidity', 'tempmax', 'feelslike', 'feelslikemax', 'temp']

Hình 3.25: Lựa chọn đặc trưng bằng feature score

Lựa chọn đặc trưng (feature selection) cho bài toán phân loại nhiều nhãn (multi-label classification) bằng cách sử dụng phương pháp chọn đặc trưng tự động dựa trên mô hình học máy - cụ thể là RandomForestClassifier - để chọn ra tối đa 10 đặc trưng quan trọng nhất.

```
sfm = SelectFromModel(estimator=RandomForestClassifier(), max_features=10)
sfm.fit(X_train, y_train)
```



```
selected_features = sfm.get_feature_names_out()
selected_features
```

```
array(['precip', 'precipprob', 'precipcover', 'preciptype', 'snow',
      'cloudcover'], dtype=object)
```

Hình 3.26: Chọn đặc trưng tự động dựa trên RandomForestClassifier

Lựa chọn đặc trưng (feature selection) cho bài toán phân loại nhiều nhãn (multi-label classification) bằng cách Huấn luyện mô hình XGBClassifier và trực quan hóa mức độ quan trọng của các đặc trưng (feature importances).

```

• xgb = XGBClassifier()
  xgb.fit(X_train,y_train)

```

```

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

```

```

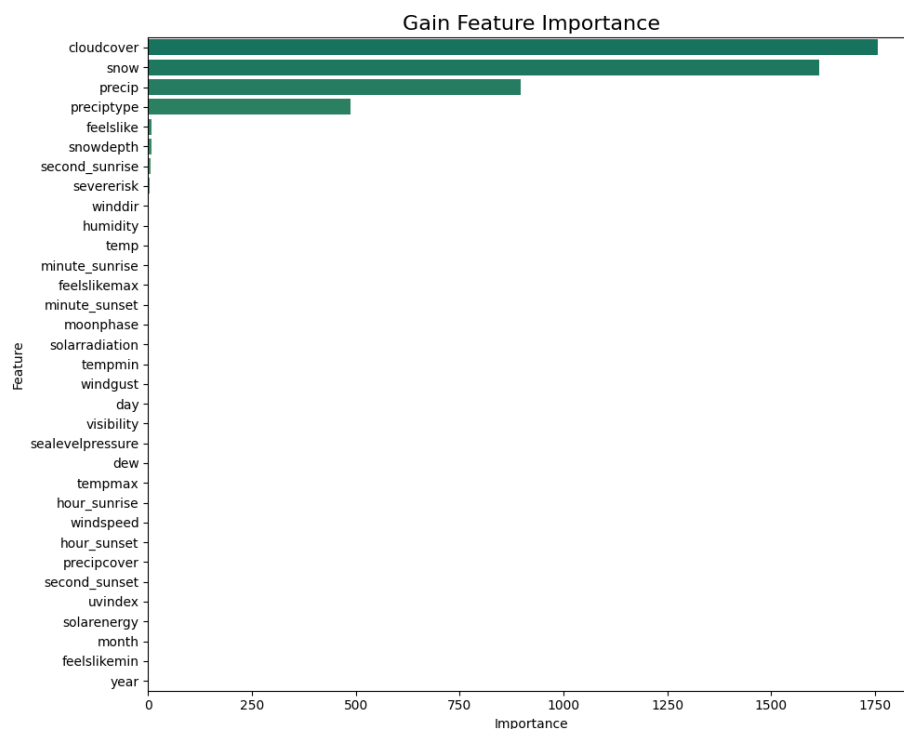
def plot_feature_importances(feat_imp_type, figsize=(10, 8)):
    featimps = xgb.get_booster().get_score(importance_type=feat_imp_type)
    keys = list(featimps.keys())
    values = list(featimps.values())

    featimps_df = pd.DataFrame(data=values, index=keys, columns=["Importance"])
    featimps_df.sort_values(by="Importance", ascending=False).reset_index()
    featimps_df.rename({'index': 'Feature'}, axis=1, inplace=True)

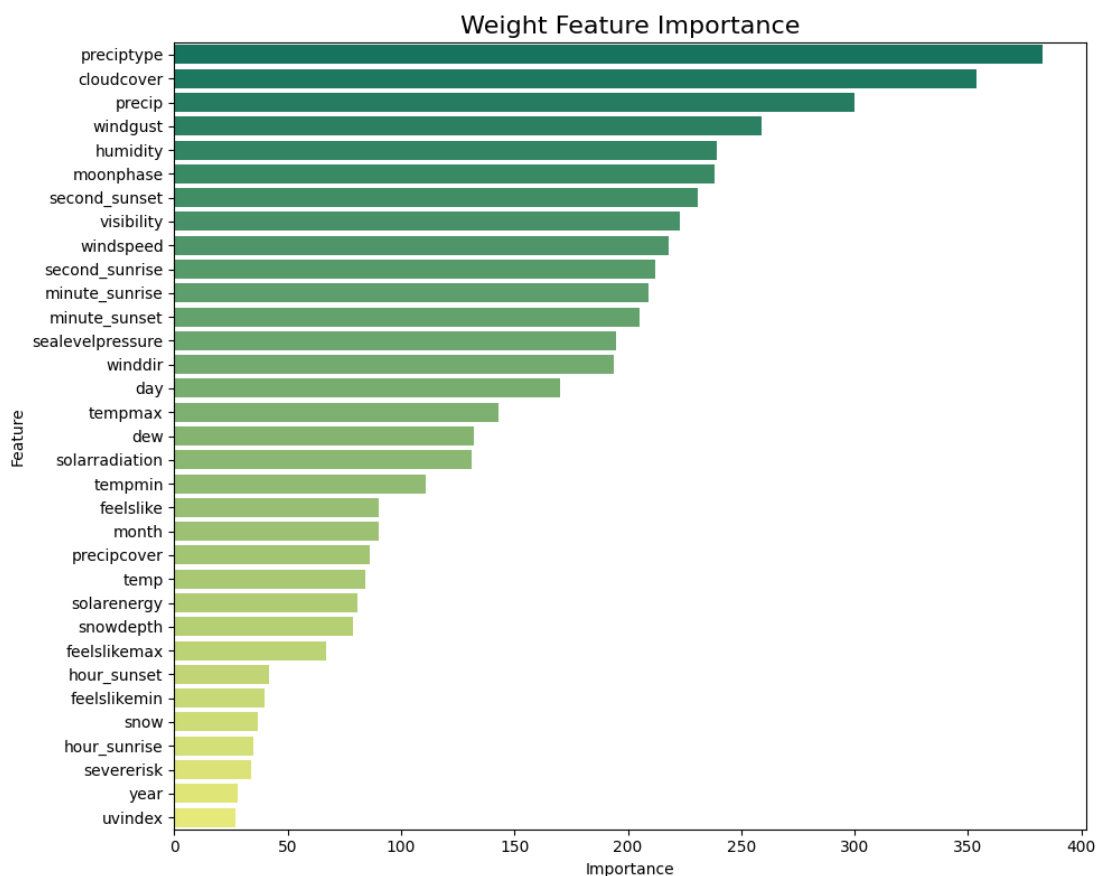
    plt.figure(figsize=figsize)
    fig = sns.barplot(
        x='Importance',
        y='Feature',
        data=featimps_df,
        orient='h',
        palette='summer'
    )
    plt.title(f"{feat_imp_type.title()} Feature Importance", fontsize=16)
    plt.tight_layout()
    plt.show()
    plt.close('all')
    del fig
    gc.collect()

```

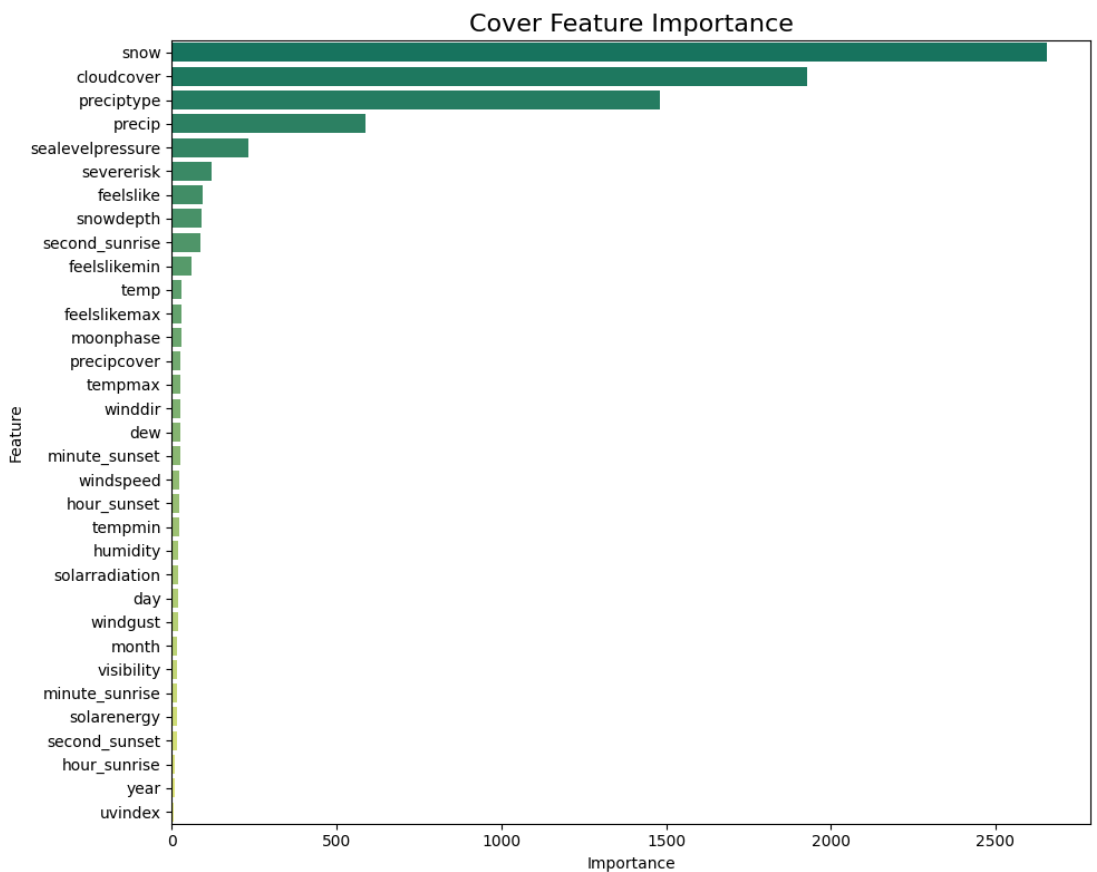
Hình 3.27: Hàm vẽ countplot của các đặc trưng dựa trên loại tầm quan trọng



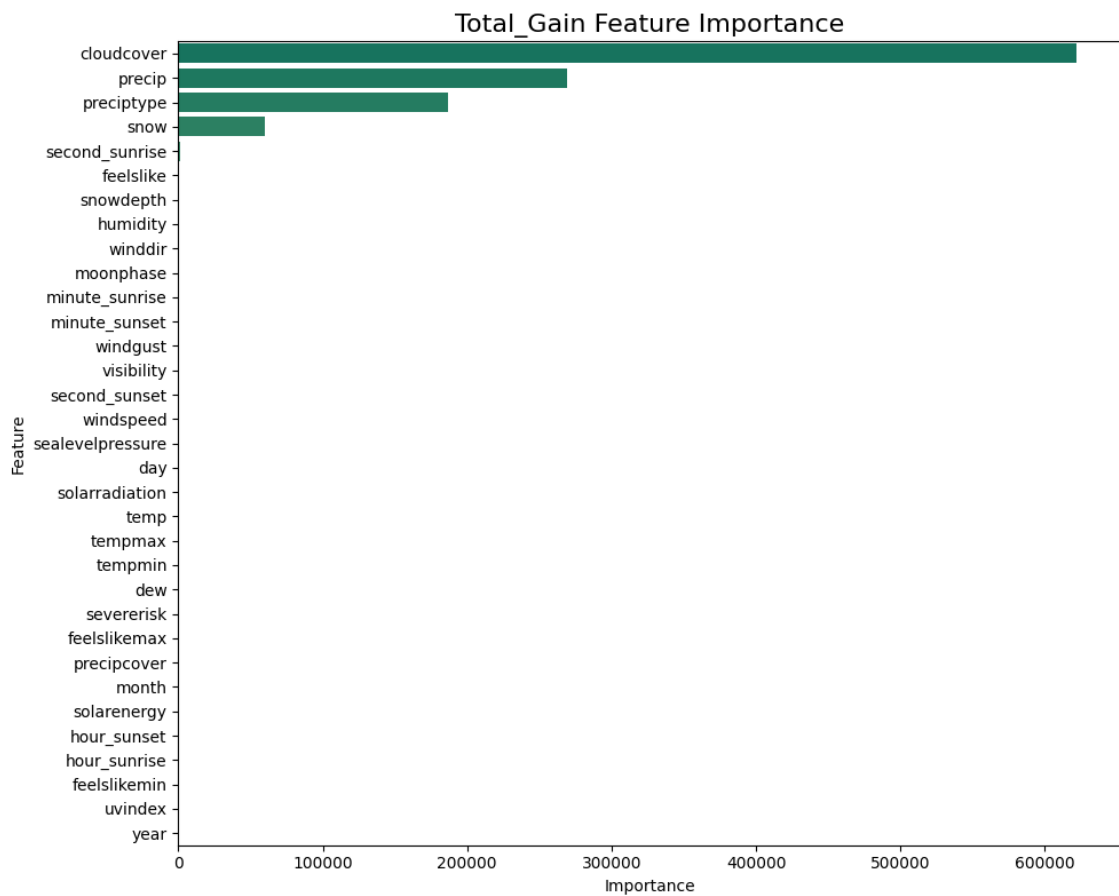
Hình 3.28: hàm plot\_feature\_importances và truyền 'gain' làm tham số



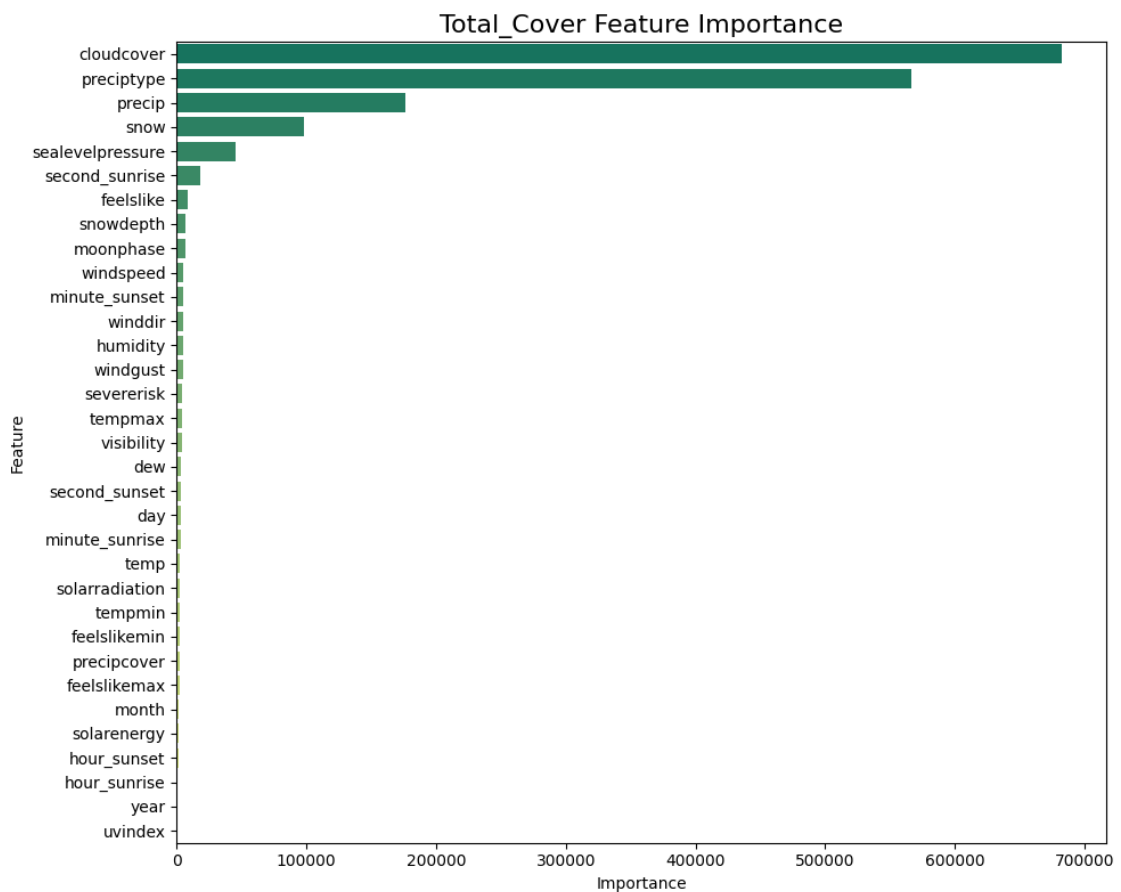
Hình 3.29: hàm `plot_feature_importances` và truyền 'weight' làm tham số



Hình 3.30: hàm `plot_feature_importances` và truyền 'cover' làm tham số



Hình 3.31: hàm `plot_feature_importances` và truyền `'total_gain'` làm tham số



Hình 3.32: hàm `plot_feature_importances` và truyền `'total_cover'` làm tham số



```

final_selected_features = [
    'tempmax', 'tempmin', 'temp', 'feelslikemax', 'feelslikemin', 'feelslike', 'dew', 'humidity',
    'precip', 'precipprob', 'precipcover', 'preciptype', 'snow', 'snowdepth',
    'windgust', 'windspeed', 'winddir', 'sealevelpressure', 'cloudcover', 'visibility',
    'solarradiation', 'solarenergy', 'uvindex', 'severerisk', 'moonphase', 'year', 'month', 'day',
    'hour_sunrise', 'minute_sunrise', 'second_sunrise', 'hour_sunset', 'minute_sunset', 'second_sunset'
]

final_X_train = X_train[final_selected_features]
final_X_test = X_test[final_selected_features]

```

Hình 3.33: Lựa chọn các đặc trưng cuối cùng để huấn luyện

## ■ Feature Scaling

Chuẩn hóa dữ liệu chuẩn bị cho vào mô hình huấn luyện.

```

scaler = StandardScaler()
features = final_X_train.columns
final_X_train = scaler.fit_transform(final_X_train)
final_X_train = pd.DataFrame(final_X_train, columns=features)
final_X_test = scaler.transform(final_X_test)
final_X_test = pd.DataFrame(final_X_test, columns=features)

final_X_train.head()

```

	tempmax	tempmin	temp	feelslikemax	feelslikemin	feelslike	dew	humidity	precip	precipprob	...	moonphase
0	0.470832	0.518624	0.450696	0.432068	0.497378	0.394584	0.845608	0.852653	0.032987	1.064560	...	1.090676
1	-1.485019	-1.229966	-1.502169	-1.511342	-1.274267	-1.536972	-1.073477	1.004883	0.295190	1.064560	...	-0.781359
2	0.796807	1.043202	0.907914	1.299428	1.183808	1.202683	1.317384	0.877206	0.101702	1.064560	...	0.085323
3	-0.530378	-0.614158	-0.541235	-0.526594	-0.522462	-0.479215	-0.401797	0.057130	-0.301547	-0.939355	...	-0.192015
4	-0.359629	-0.234030	-0.177010	-0.383121	-0.149828	-0.163859	0.365837	1.338806	-0.251186	1.064560	...	-0.642690

5 rows × 34 columns

Hình 3.34: Chuẩn hóa dữ liệu chuẩn bị cho vào mô hình huấn luyện

### 3.2.4 Đào tạo & Đánh giá Model.

Tạo các mảng để lưu các giá trị như: tên mô hình đưa vào huấn luyện và các chỉ số như Accuracy, Precision, Recall, F1.

Viết hàm `train_and_evaluate_model` để huấn luyện các mô hình và in báo cáo phân loại của mô hình tương ứng.

```

models = []
accuracy_scores = []
precision_scores = []
recall_scores = []
f1_scores = []
hamming_losses = []

def train_and_evaluate_model(model):
    model.fit(final_X_train, y_train)
    y_pred = model.predict(final_X_test)

    print("Classification Report:")
    print(classification_report(y_test, y_pred, zero_division=0))

    acc = accuracy_score(y_test, y_pred) # subset accuracy
    hamming = hamming_loss(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='macro', zero_division=0)
    recall = recall_score(y_test, y_pred, average='macro', zero_division=0)
    f1 = f1_score(y_test, y_pred, average='macro', zero_division=0)

    accuracy_scores.append(acc)
    hamming_losses.append(hamming)
    precision_scores.append(precision)
    recall_scores.append(recall)
    f1_scores.append(f1)
    models.append(model)
    gc.collect()

```

Hình 3.35: Mảng lưu các giá trị và Hàm huấn luyện mô hình

Huấn luyện và đánh giá mô hình GradientBoostingClassifier và hiển thị báo cáo phân loại sau khi training mô hình.

```
gbc = OneVsRestClassifier(GradientBoostingClassifier(random_state=42))
train_and_evaluate_model(gbc)
```

```
Classification Report:
              precision    recall  f1-score   support

     0           1.00        1.00        1.00     16243
     1           1.00        1.00        1.00     11407
     2           1.00        1.00        1.00      5484
     3           1.00        1.00        1.00      3429
     4           1.00        1.00        1.00      2988
     5           1.00        1.00        1.00      1843
     6           1.00        1.00        1.00      1006

 micro avg       1.00        1.00        1.00    42400
 macro avg       1.00        1.00        1.00    42400
 weighted avg    1.00        1.00        1.00    42400
 samples avg     1.00        1.00        1.00    42400
```

Hình 3.36: Kết quả mô hình GradientBoostingClassifier

### 3.2.5 So sánh hiệu suất của các mô hình cơ sở.

	Model	Accuracy	Precision	Recall	F1
15	OneVsRestClassifier(estimator=StackingClassifi...	1.000000	1.000000	1.000000	1.000000
14	OneVsRestClassifier(estimator=VotingClassifier...	1.000000	1.000000	1.000000	1.000000
8	OneVsRestClassifier(estimator=GradientBoosting...	1.000000	1.000000	1.000000	1.000000
9	OneVsRestClassifier(estimator=AdaBoostClassifi...	0.999961	0.999858	1.000000	0.999929
10	OneVsRestClassifier(estimator=BaggingClassifie...	0.999921	0.999781	1.000000	0.999890
6	OneVsRestClassifier(estimator=DecisionTreeClas...	0.999803	0.999357	1.000000	0.999678
7	OneVsRestClassifier(estimator=RandomForestClas...	0.998385	0.998885	0.999599	0.999240
13	OneVsRestClassifier(estimator=<catboost.core.C...	0.997598	0.999088	0.998303	0.998693
12	OneVsRestClassifier(estimator=XGBClassifier(ba...	0.997204	0.998912	0.999081	0.998996
2	OneVsRestClassifier(estimator=SVC())	0.966525	0.988014	0.987883	0.987926
11	OneVsRestClassifier(estimator=ExtraTreesClassi...	0.953017	0.985430	0.968297	0.975834
1	OneVsRestClassifier(estimator=KNeighborsClassi...	0.850504	0.929519	0.920034	0.923802
0	OneVsRestClassifier(estimator=LogisticRegressi...	0.797889	0.948400	0.958275	0.953053
3	OneVsRestClassifier(estimator=LinearSVC())	0.795959	0.949728	0.958252	0.953652
4	OneVsRestClassifier(estimator=GaussianNB())	0.530167	0.743146	0.933523	0.816152
5	OneVsRestClassifier(estimator=BernoulliNB())	0.376063	0.647693	0.868233	0.732111

Hình 3.37: So sánh hiệu suất của các mô hình cơ sở

Dựa trên bảng kết quả, ta có thể so sánh hiệu suất của 16 mô hình như sau:

**Độ chính xác (Accuracy)** cho ta biết tỷ lệ dự đoán chính xác của mô hình.

**Độ chính xác (Precision)** cho ta biết tỷ lệ dự đoán là đúng trong số các dự đoán mà mô hình đưa ra..

**Độ thu hồi (Recall)** cho ta biết tỷ lệ các trường hợp thực tế được dự đoán chính xác.

**Điểm F1 (F1 Score)** là thước đo tổng hợp cho cả độ chính xác và độ thu hồi.

Dựa trên cả bốn chỉ số hiệu suất, mô hình 8, 14, 15 (GradientBoostingClassifier, VotingClassifier, StackingClassifier) có hiệu suất tốt nhất trong việc dự đoán các biến mục tiêu. Mô hình 5 (BernoulliNB) có hiệu suất thấp nhất và cần được cải thiện thêm.

### 3.3 Models Điều chỉnh siêu tham số và xác thực chéo

Định nghĩa không gian siêu tham số để thực hiện tối ưu hóa mô hình Gradient Boosting trong bài toán phân loại đa nhãn (multi-label classification) bằng cách sử dụng RandomizedSearchCV. Cụ thể:

- Mô hình chính là GradientBoostingClassifier, được bao bọc trong OneVsRestClassifier để xử lý từng nhãn riêng biệt trong bài toán đa nhãn.

- Tập siêu tham số (param\_grid) gồm các yếu tố quan trọng như:

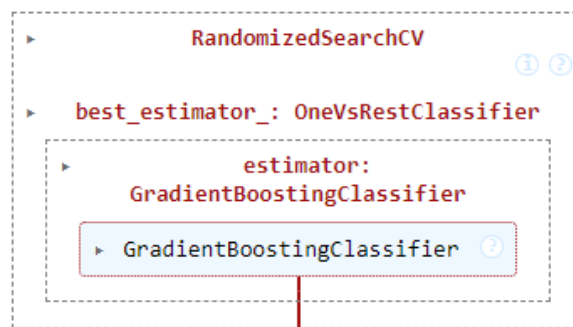
- + Loại hàm mất mát (loss),
- + Số lượng cây (n\_estimators),
- + Tốc độ học (learning\_rate),
- + Tiêu chí chia cây (criterion),
- + Số đặc trưng được xét khi chia (max\_features).

- Dùng RandomizedSearchCV để thử ngẫu nhiên 10 tổ hợp siêu tham số và chọn tổ hợp tốt nhất thông qua cross-validation (3 lần chia dữ liệu).

```
param_grid = {
    'estimator__loss': ['log_loss', 'deviance', 'exponential'],
    'estimator__n_estimators': [100, 400, 800, 1000],
    'estimator__learning_rate': [0.2, 0.4, 0.7, 1],
    'estimator__criterion': ['friedman_mse', 'squared_error'],
    'estimator__max_features': ['sqrt', 'log2']
}
base_model = OneVsRestClassifier(GradientBoostingClassifier())

grid_gb = RandomizedSearchCV(base_model, param_grid, n_iter=10, cv=3, verbose=3)
grid_gb.fit(final_X_train, y_train)
```

Hình 3.38: Định nghĩa không gian siêu tham số cho ExtraTreesClassifier



```
grid_gb.best_score_
```

```
np.float64(1.0)
```

```
grid_gb.best_params_
```

```
{'estimator__n_estimators': 800,
 'estimator__max_features': 'sqrt',
 'estimator__loss': 'exponential',
 'estimator__learning_rate': 0.7,
 'estimator__criterion': 'squared_error'}
```

Hình 3.39: Kết quả *best\_score* và *best\_params* sau khi huấn luyện

Ngoài ra, còn 7 mô hình khác cũng được điều chỉnh siêu tham số và xác thực chéo.

### 3.4 Đào tạo và đánh giá mô hình học sâu

Trong dự án dự báo thời tiết, việc sử dụng mô hình học sâu đóng vai trò quan trọng trong việc nâng cao độ chính xác của dự đoán. Học sâu (Deep Learning) là một nhánh của học máy, nổi bật với khả năng học và xử lý các đặc trưng phức tạp từ dữ liệu lớn. Bằng cách áp dụng các mạng nơ-ron sâu, mô hình có thể khai thác mối quan hệ phi tuyến tính giữa các yếu tố thời tiết, từ đó đưa ra dự đoán chính xác hơn so với các phương pháp truyền thống.

Dự án này có khởi tạo và cấu hình một mô hình mạng nơ-ron sâu (Deep Neural Network) bằng cách sử dụng thư viện Keras trong Python. Mô hình này được thiết kế để giải quyết bài toán phân loại đa lớp (multi-class classification), trong trường hợp này là dự báo thời tiết dựa trên các thông số đầu vào.

Model: "sequential"

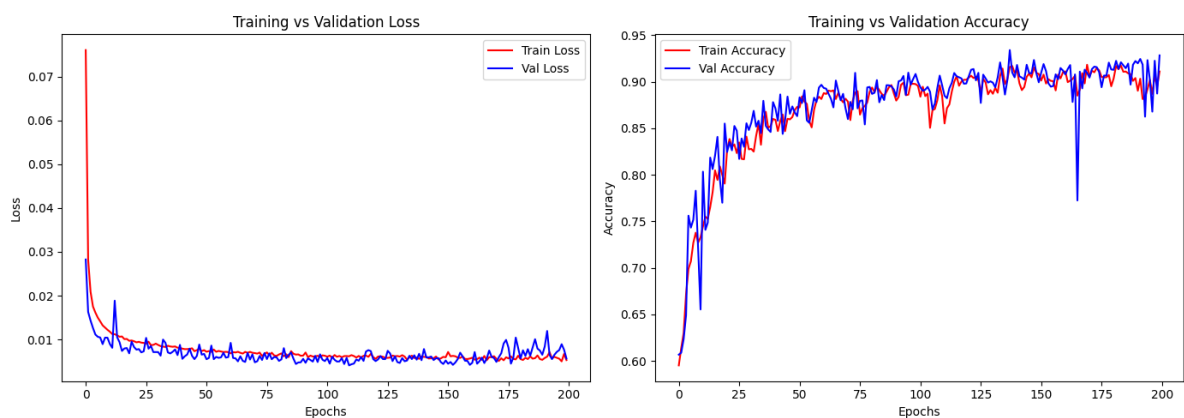
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	1,120
dense_1 (Dense)	(None, 64)	2,112
dropout (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 128)	8,320
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 256)	33,024
dropout_2 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 7)	1,799

Total params: 46,375 (181.15 KB)

Trainable params: 46,375 (181.15 KB)

Non-trainable params: 0 (0.00 B)

Hình 3.40: Tóm tắt mô hình



Hình 3.41: Biểu đồ tổn thất và biểu đồ độ chính xác

	STT	Model	Accuracy	Precision	Recall	F1
15	1	OneVsRestClassifier(estimator=StackingClassifi...	1.000000	1.000000	1.000000	1.000000
14	2	OneVsRestClassifier(estimator=VotingClassifier...	1.000000	1.000000	1.000000	1.000000
8	3	OneVsRestClassifier(estimator=GradientBoosting...	1.000000	1.000000	1.000000	1.000000
9	4	OneVsRestClassifier(estimator=AdaBoostClassifi...	0.999961	0.999858	1.000000	0.999929
10	5	OneVsRestClassifier(estimator=BaggingClassifie...	0.999921	0.999781	1.000000	0.999890
6	6	OneVsRestClassifier(estimator=DecisionTreeClas...	0.999803	0.999357	1.000000	0.999678
7	7	OneVsRestClassifier(estimator=RandomForestClas...	0.998385	0.998885	0.999599	0.999240
13	8	OneVsRestClassifier(estimator=<catboost.core.C...	0.997598	0.999088	0.998303	0.998693
12	9	OneVsRestClassifier(estimator=XGBClassifier(ba...	0.997204	0.998912	0.999081	0.998996
16	10	Deep Learning	0.992832	0.996994	0.995857	0.996421
2	11	OneVsRestClassifier(estimator=SVC())	0.966525	0.988014	0.987883	0.987926
11	12	OneVsRestClassifier(estimator=ExtraTreesClassi...	0.953017	0.985430	0.968297	0.975834
1	13	OneVsRestClassifier(estimator=KNeighborsClassi...	0.850504	0.929519	0.920034	0.923802
0	14	OneVsRestClassifier(estimator=LogisticRegressi...	0.797889	0.948400	0.958275	0.953053
3	15	OneVsRestClassifier(estimator=LinearSVC())	0.795959	0.949728	0.958252	0.953652
4	16	OneVsRestClassifier(estimator=GaussianNB())	0.530167	0.743146	0.933523	0.816152
5	17	OneVsRestClassifier(estimator=BernoulliNB())	0.376063	0.647693	0.868233	0.732111

Hình 3.42: So sánh lại sau khi huấn luyện model Deep learning

```

avg_cv_scores = cross_val_score(best_model, final_X_test, y_test, scoring='accuracy', cv=RepeatedKFold(n_repeats=3, n_splits=4), verbose=2)
mean_score = round(np.mean(avg_cv_scores), 4)
print(f'Mean Cross Validation Performance of GradientBoostingClassifier: {mean_score*100}%')

[CV] END ..... total time= 54.1s
[CV] END ..... total time= 53.8s
[CV] END ..... total time= 52.7s
[CV] END ..... total time= 53.2s
[CV] END ..... total time= 55.5s
[CV] END ..... total time= 53.0s
[CV] END ..... total time= 53.5s
[CV] END ..... total time= 55.1s
[CV] END ..... total time= 1.0min
[CV] END ..... total time= 56.4s
[CV] END ..... total time= 56.8s
[CV] END ..... total time= 54.0s
Mean Cross Validation Performance of StackingClassifier: 99.96000000000001%

```

Hình 3.43: Tính điểm độ chính xác của mô hình cuối cùng được chọn

```

with open('/content/drive/MyDrive/DACN3/source/gbC.pkl', 'wb') as f:
    pickle.dump(best_model, f, protocol=4)

with open('/content/drive/MyDrive/DACN3/source/scaler.pkl', 'wb') as f:
    pickle.dump(scaler, f, protocol=4)

with open('/content/drive/MyDrive/DACN3/source/label_encoder.pkl', 'wb') as f:
    pickle.dump(encoder, f, protocol=4)

model.save('/content/drive/My Drive/DACN3/source/deep.keras')

with open('/content/drive/MyDrive/DACN3/source/best_grid_gb.pkl', 'wb') as f:
    pickle.dump(grid_gb.best_estimator_, f, protocol=4)

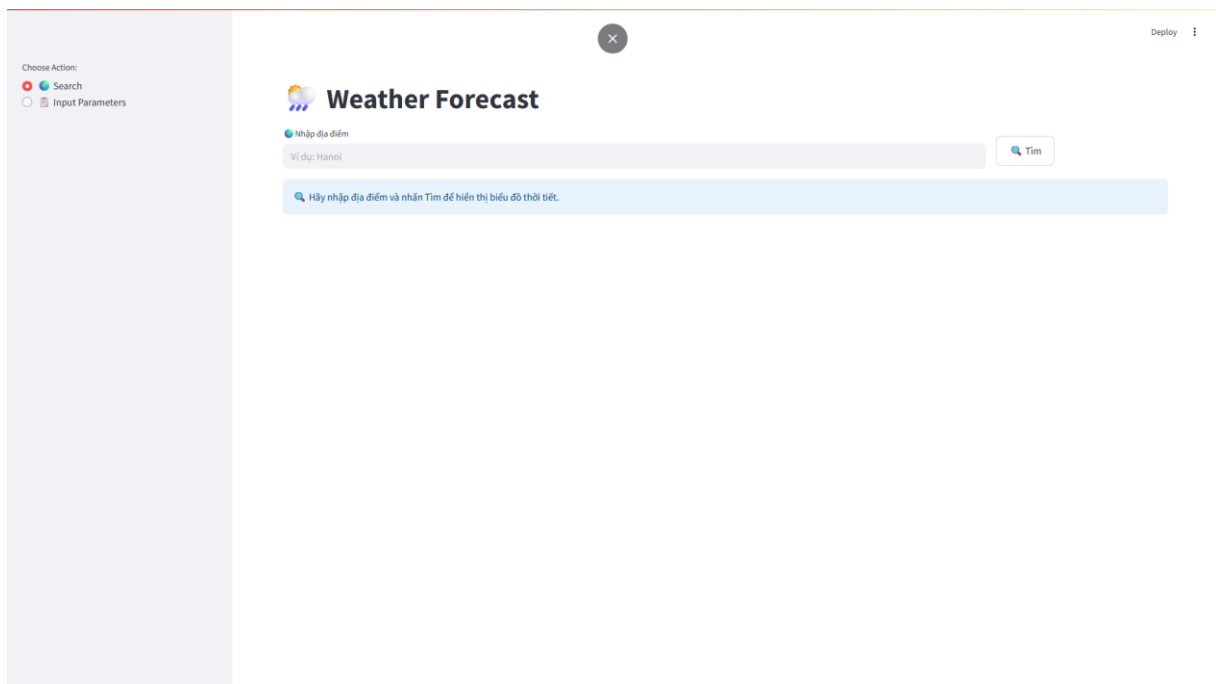
```

Hình 3.44: Lưu mô hình hoạt động tốt nhất để triển khai

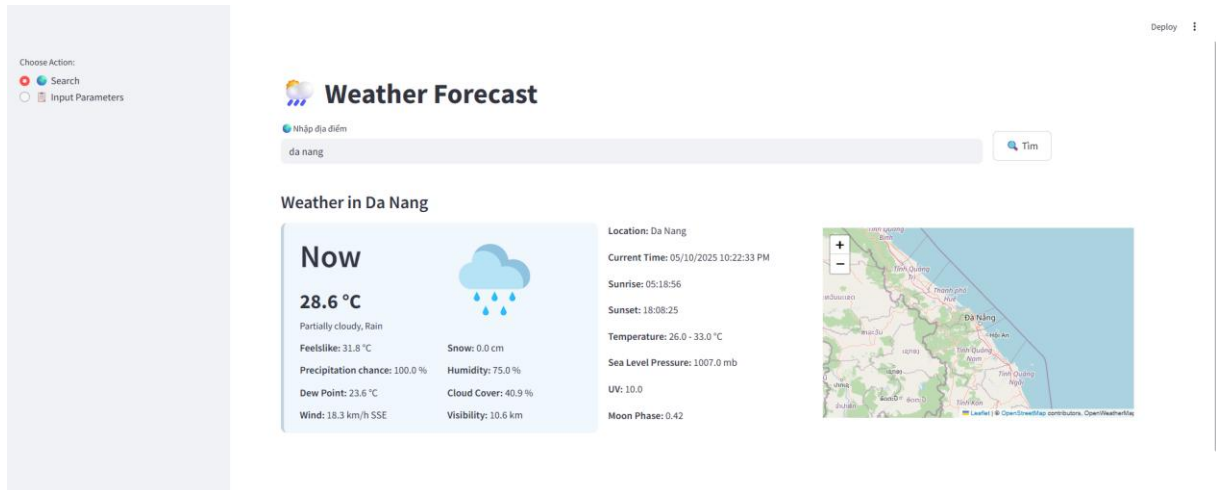
### 3.5 Triển khai Website Forecast với Streamlit

Streamlit là một thư viện Python mã nguồn mở, giúp tạo các ứng dụng web tương tác nhanh chóng và đơn giản, đặc biệt hữu ích trong lĩnh vực khoa học dữ liệu và học máy. Điểm nổi bật của Streamlit là người dùng chỉ cần viết mã Python mà không cần phải biết HTML, CSS hay JavaScript.

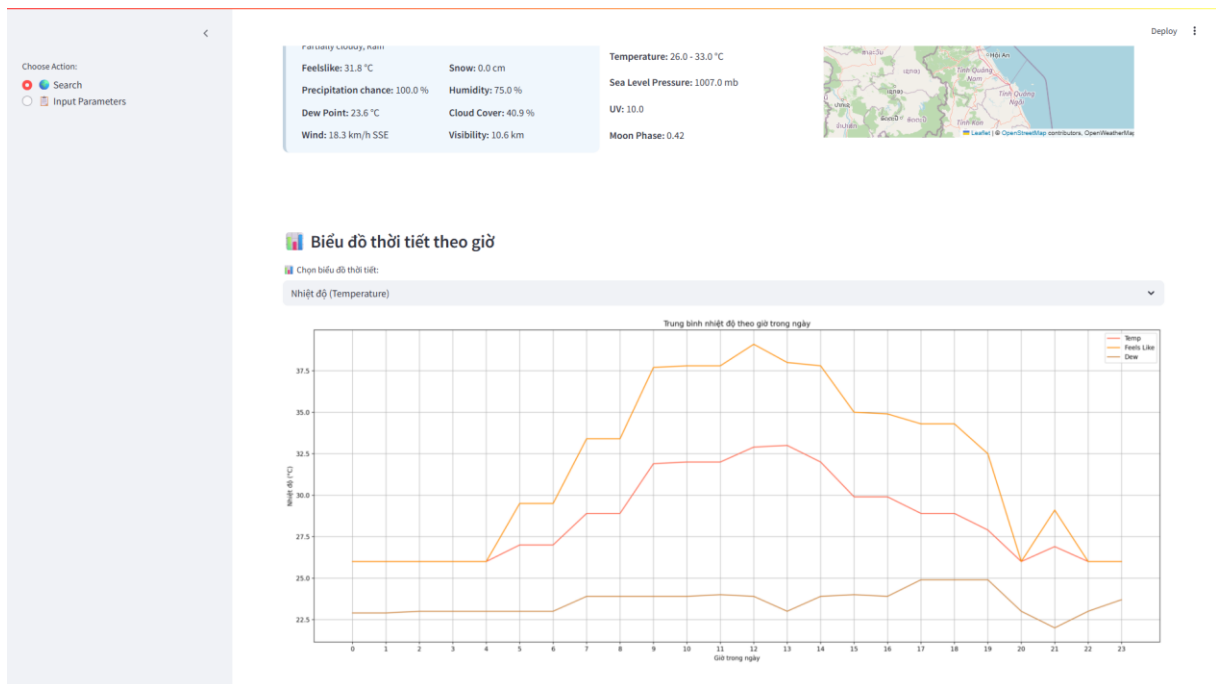
Một ứng dụng cụ thể của Streamlit là trong dự đoán và mô hình hóa, ví dụ như dự báo thời tiết (Weather Forecast). Người dùng có thể tìm kiếm khu vực muốn dự báo tại phần Tìm kiếm. Sau đó nhấn nút Tìm sẽ dự đoán được kiểu thời tiết mà bạn đã nhập, bên cạnh đó sẽ hiện các thông tin các về thời tiết. Bên cạnh đó sẽ có bản đồ tại vị trí và các biểu đồ thời tiết theo giờ trong ngày. Dựa trên mô hình chính xác nhất đang có là GradientBoostingClassifier sau khi đã thực hiện tối ưu hóa mô hình. Người dùng cũng có thể tự nhập thủ công các thông số thời tiết vào mục Input Parameters. Dựa trên các thông số được cung cấp, mô hình sẽ thực hiện dự báo thời tiết, đảm bảo sự linh hoạt và đáp ứng nhu cầu cá nhân hóa trong quá trình sử dụng.



Hình 3.45: Giao diện chính

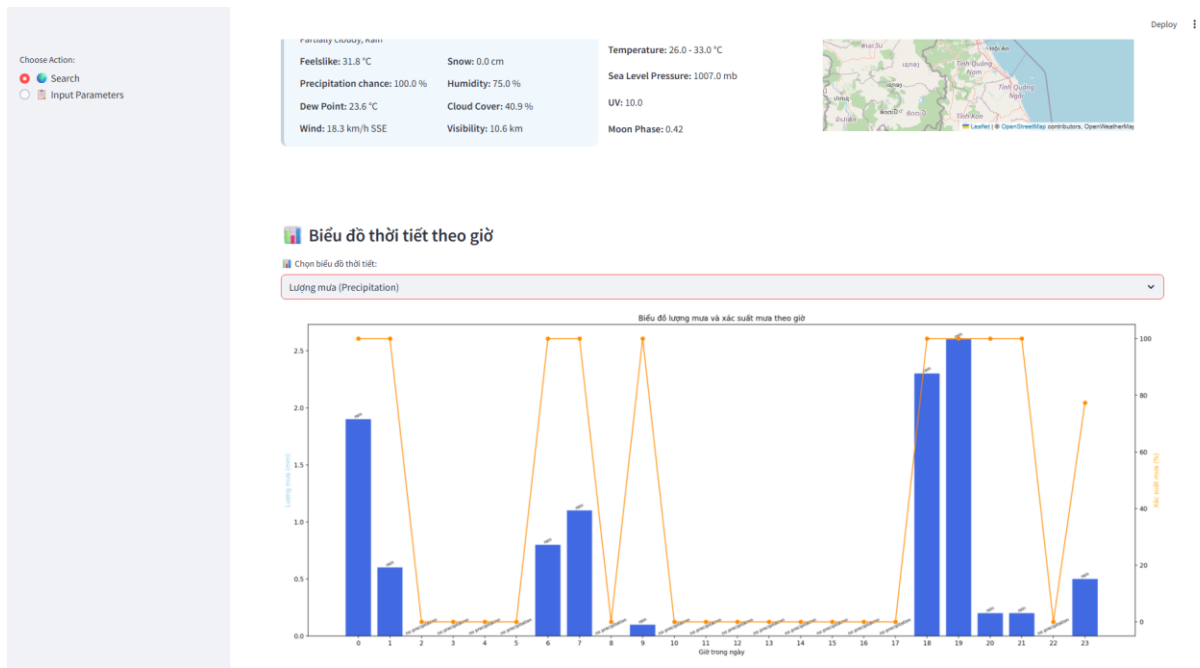


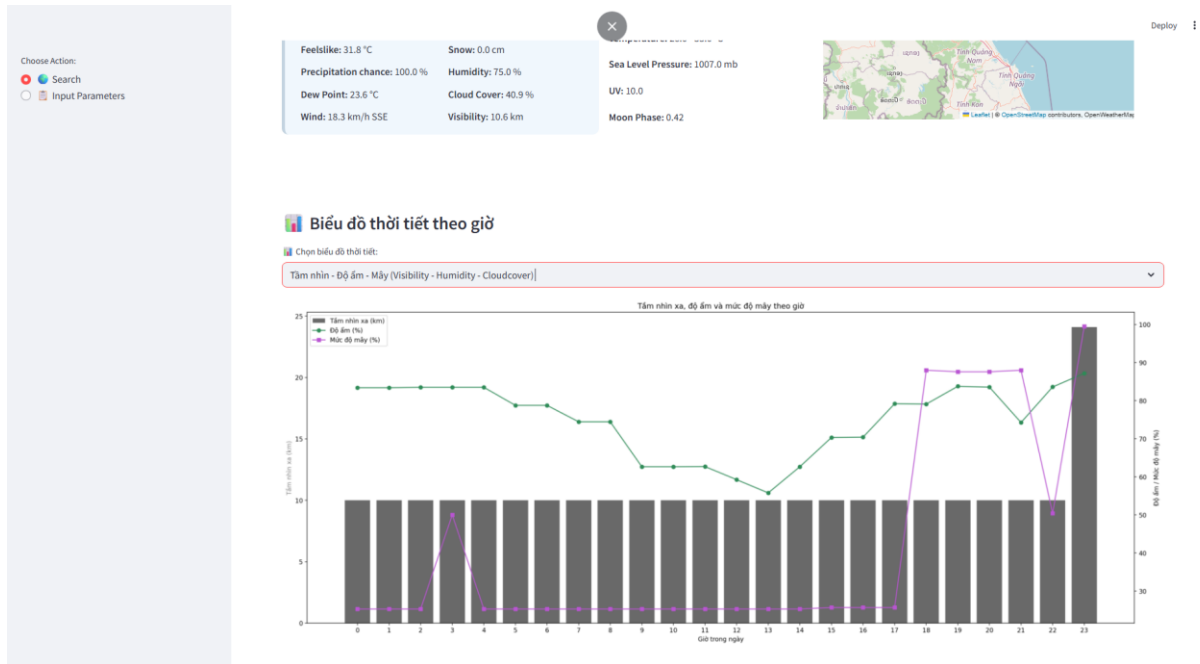
Hình 3.46: Tìm kiếm và dự báo thời tiết khu vực đã chọn



Hình 3.47: Biểu đồ nhiệt theo giờ trong ngày



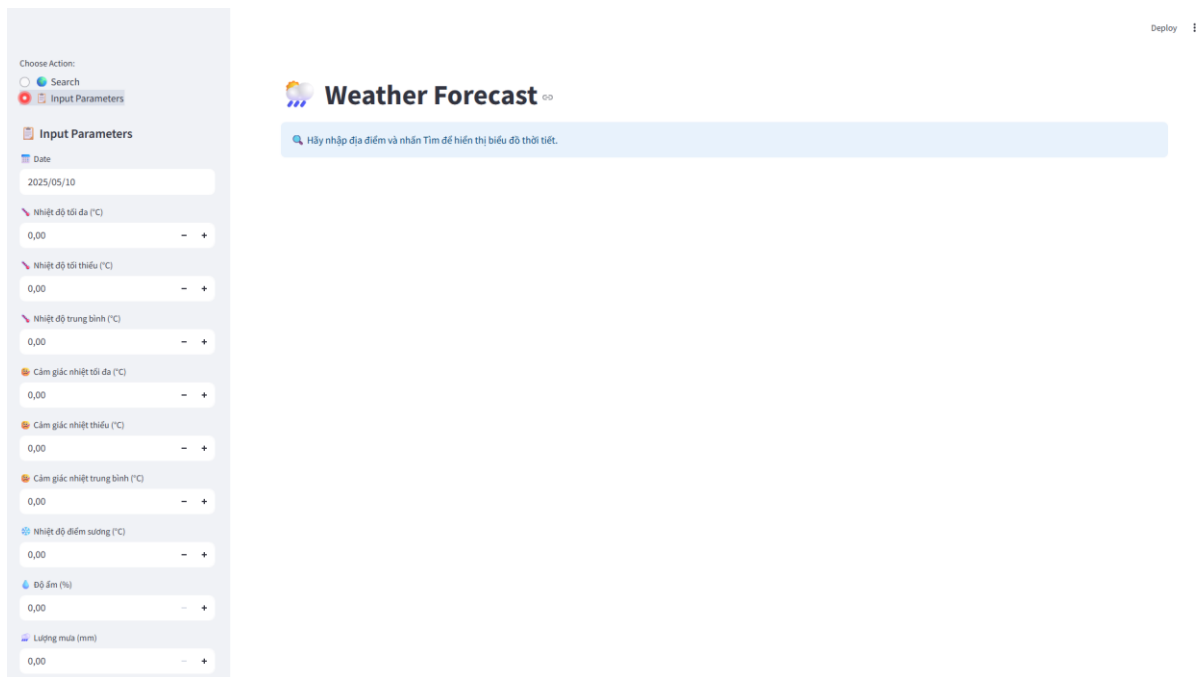




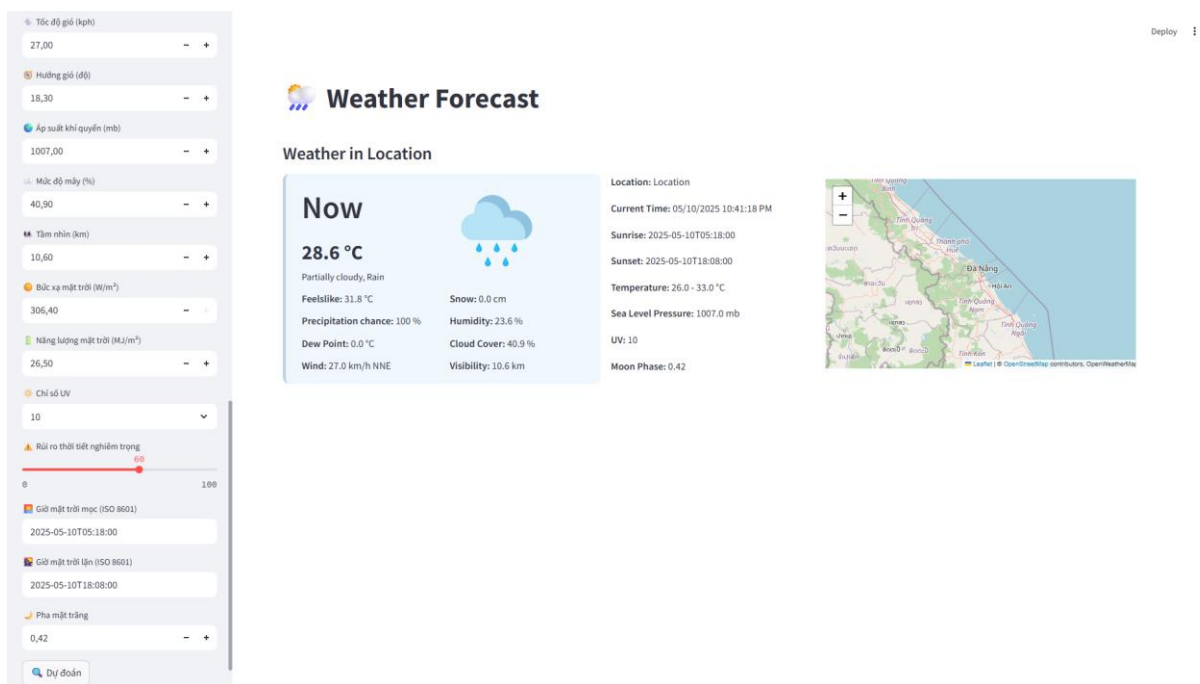
Hình 3.50: Biểu đồ V-H-C theo giờ trong ngày



Hình 3.51: Biểu đồ gió theo giờ trong ngày



Hình 3.52: Giao diện nhập thủ công



Hình 3.53: Nhập thông tin thủ công và dự đoán thời tiết

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. Kết quả đạt được.

**Phân tích dữ liệu một cách chi tiết:** Thực hiện các mô hình dự báo dựa trên dữ liệu lịch sử và các yếu tố thời tiết hiện tại để dự đoán thời tiết trong tương lai.

**Cung cấp thông tin hữu ích cho người sử dụng:** Cung cấp dịch vụ dự báo thời tiết cụ thể và chính xác cho các khu vực cụ thể, cũng như cảnh báo sớm về các sự kiện thời tiết bất thường.

**Xây dựng một kho dữ liệu:** Có cấu trúc rõ ràng và đáng tin cậy, bao gồm thông tin về nhiệt độ, độ ẩm, áp suất không khí, tốc độ gió, và lượng mưa, v.v.

## 2. Hạn chế.

Mặc dù việc phân tích dữ liệu thời tiết và dự báo thời tiết mang lại nhiều lợi ích, nhưng cũng tồn tại một số hạn chế nhất định:

**Độ chính xác chưa tuyệt đối:** Dự báo thời tiết, bao gồm dự báo thời tiết, vẫn còn sai số do sự phức tạp của các hiện tượng khí tượng và hạn chế của các mô hình dự báo.

**Giới hạn về dữ liệu và công nghệ:** Chất lượng và độ phủ của dữ liệu thời tiết có thể bị hạn chế do thiếu các trạm quan trắc ở một số khu vực, đặc biệt là vùng sâu vùng xa. Ngoài ra, công nghệ phân tích dữ liệu và dự báo cũng có thể gặp khó khăn khi xử lý lượng dữ liệu lớn và phức tạp.

**Tác động của biến đổi khí hậu:** Biến đổi khí hậu có thể làm thay đổi các mẫu thời tiết và gây khó khăn cho việc dự báo. Những thay đổi đột ngột và bất thường trong khí hậu có thể làm giảm độ tin cậy của các mô hình dự báo hiện tại.

**Hạn chế về thời gian dự báo:** Dự báo thời tiết có độ tin cậy cao nhất trong ngắn hạn (vài giờ đến vài ngày). Dự báo dài hạn (vài tuần trở lên) thường kém chính xác hơn và gặp nhiều thách thức trong việc duy trì độ chính xác.

**Ảnh hưởng từ các yếu tố bên ngoài:** Các yếu tố như địa hình, thảm thực vật, và hoạt động của con người cũng có thể ảnh hưởng đến thời tiết và khó dự báo chính xác. Ví dụ, các thành phố lớn có thể tạo ra hiệu ứng đảo nhiệt đô thị, ảnh hưởng thời tiết cục bộ.

## 3. Hướng phát triển.

**Nâng cao độ chính xác của mô hình dự báo:**

- Cải tiến mô hình số: Tích hợp các mô hình khí tượng tiên tiến hơn, như mô hình kết hợp đa quy mô (multi-scale models), để cải thiện độ chính xác và khả năng dự đoán chi tiết.

- Tích hợp dự báo thời tiết tương lai (theo ngày hoặc theo giờ)

- Phát triển các thuật toán mới: Sử dụng trí tuệ nhân tạo (AI) và học máy (machine learning) để phân tích các mẫu thời tiết phức tạp và cải thiện độ chính xác của dự báo ngắn hạn và dài hạn.

### ***Tăng cường thu thập dữ liệu:***

- Tìm kiếm và thu thập thêm dữ liệu từ nhiều vùng khác nhau để tăng độ đa dạng dữ liệu
- Tìm kiếm và thu thập thêm dữ liệu thời tiết theo giờ
- Tăng thêm các yếu tố thời tiết trong dữ liệu để tăng cường độ chính xác

### ***Cải thiện hạ tầng công nghệ:***

- Tăng cường sức mạnh tính toán: Đầu tư vào các siêu máy tính và hệ thống tính toán hiệu năng cao để xử lý lượng dữ liệu lớn và phức tạp một cách nhanh chóng và hiệu quả.
- Phát triển nền tảng dữ liệu mở: Tạo ra các nền tảng dữ liệu mở, cho phép chia sẻ và truy cập dữ liệu thời tiết dễ dàng hơn giữa các cơ quan, tổ chức và các nhà nghiên cứu.

### ***Ứng dụng công nghệ AI và dữ liệu lớn:***

- Phân tích dữ liệu lớn: Sử dụng công nghệ dữ liệu lớn (big data) để phân tích và xử lý các tập dữ liệu thời tiết khổng lồ, từ đó rút ra các kết luận và mô hình dự báo chính xác hơn.
- Trí tuệ nhân tạo: Sử dụng AI để tự động hóa quá trình phân tích dữ liệu và đưa ra dự báo thời tiết, đồng thời học hỏi từ các dữ liệu mới để cải thiện liên tục.

### ***Phát triển các ứng dụng di động và web:***

- Ứng dụng thân thiện với người dùng: Phát triển các ứng dụng di động và web thân thiện với người dùng, cung cấp thông tin thời tiết chi tiết, dễ hiểu và có tính tương tác cao.
- Tính năng tiên tiến: Tích hợp các tính năng tiên tiến như cảnh báo thời gian thực, dự báo tùy chỉnh theo vị trí, và các công cụ phân tích thời tiết cá nhân hóa.

## **4. Kết luận.**

Tóm lại, hệ thống không chỉ cung cấp thông tin thời tiết chính xác và đáng tin cậy mà còn hỗ trợ người dùng trong việc chuẩn bị và đối phó với các tình huống khẩn cấp có liên quan đến thời tiết, từ đó nâng cao sự an toàn và hiệu quả trong các hoạt động hàng ngày và kinh doanh.

## TÀI LIỆU THAM KHẢO

- [1] <https://www.kaggle.com/code/sayamkumar/weather-prediction/notebook>
- [2] <https://www.visualcrossing.com/weather-query-builder/>
- [3] <https://www.visualcrossing.com/resources/documentation/weather-data/weather-data-documentation/>
- [4] <https://scikit-learn.org/stable/api/sklearn.ensemble.html>
- [5] <https://viblo.asia/p/multi-label-classification-cho-bai-toan-tag-predictions-oOVIY2Lr58W>
- [6] <https://openweathermap.org/>
- [7] <https://openweathermap.org/api/weather-map-1h>
- [8] <https://www.timeanddate.com/weather/vietnam/da-nang>
- [9] <https://meteostat.net/en/place/vn/ngu-hanh-son?s=48855&t=2025-04-26/2025-05-03>