ĐẠI HỌC QUỐC GIA, THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

HỌC MÁY (CO3117)

BÁO CÁO BTL

Weather-Type Classification với SVM & XGBoost

GVHD: Huỳnh Văn Thống
SV thực hiện: Hồ Anh Dũng 2310543
Đào Quang Dương 2310579
Hà Bảo Nhi 2312496
Mai Doãn Chiến 2110060

TP Hồ Chí Minh, Tháng 11 Năm 2025

Mục lục

List of Figures

List of Tables

Abstract

Bài báo cáo trình bày quy trình cho bài toán phân loại thời tiết trên dữ liệu 13,200 bản ghi với 11 thuộc tính. Quy trình gồm: (i) mô tả dữ liệu; (ii) tiền xử lý với Encode biến classification cho *Cloud Cover*, *Season*, *Location* và chuẩn hoá z-score cho các thuộc tính số; (iii) chia dữ liệu theo *stratified split* thành 8/1/1 (80%/10%/10%) cho *train/validation/test*. Ba mô hình được nhóm sử dụng bao gồm Support Vector Machine (SVM), Logistic Regression và XGBoost. Việc lựa chọn mô hình và tinh chỉnh siêu tham số được thực hiện dựa trên tập *validation*; đánh giá cuối cùng trên tập *test* bằng Accuracy, Macro-F1 và ma trận nhầm lẫn (Confusion Matrix).

1 Giới thiệu chung

Bài báo cáo trình bày quy trình sử dụng các mô hình học máy cơ bản cho bài toán *phân loại thời tiết* trên dữ liệu dạng bảng. Ba mô hình được xem xét gồm Support Vector Machine (SVM), Logistic Regression và XGBoost. Quy trình bao gồm mô tả dữ liệu, tiền xử lý (Encode thuộc tính phân loại và chuẩn hoá z-score cho các thuộc tính số), lựa chọn chiến lược đánh giá, tinh chỉnh hyperparameter và phân tích kết quả theo các thước đo xác định trước.

1.1 Bối cảnh

Bài toán đặt ra là dự đoán nhãn thời tiết (*Cloudy*, *Rainy*, *Snowy*, *Sunny*) từ các thuộc tính số và thuộc tính phân loại. Cách tiếp cận bảo đảm tính chặt chẽ trong đánh giá: các bước tiền xử lý được thực hiện rõ ràng quy trình huấn luyện/đánh giá để tránh rò rỉ dữ liệu; việc chia dữ liệu đảm bảo phân phối nhãn theo *stratified split*.

1.2 Dữ liệu

Bộ dữ liệu sử dụng là *Weather Type Classification* của tác giả Nikhil Narayan trên Kaggle. Dữ liệu được tạo tổng hợp (synthetic) có chủ đích đưa vào các giá trị ngoại lai để người học rèn luyện phát hiện và xử lý outlier trong bài toán classification. Tập dữ liệu gồm 13,200 bản ghi với 11 thuộc tính; nhãn thời tiết cần dự đoán *Weather Type* gồm bốn giá trị: *Rainy*, *Sunny*, *Cloudy*, *Snowy*. Các thuộc tính gồm [6]:

- **Thuộc tính số**: *Temperature* (đo bằng thang Celcius), *Humidity* (gồm cả giá trị $> 100\%$ để tạo ngoại lai), *Wind Speed* (có dải giá trị rất lớn, bao gồm mức không thực tế), *Precipitation (%)*, *Atmospheric Pressure* (hPa), *UV Index*, *Visibility (km)*.
- **Thuộc tính phân loại**: *Cloud Cover* (mức độ mây), *Season*, *Location*.

Theo mô tả của tác giả, dữ liệu không đại diện cho điều kiện thời tiết thực tế mà được thiết kế phục vụ mục đích học tập: luyện tập tiền xử lý, chọn thuộc tính, đánh giá mô hình và thử nghiệm phương pháp phát hiện/ứng xử với ngoại lai.

1.3 Mục tiêu

Bài báo cáo này trình bày:

- Xây dựng quy trình tiền xử lý rõ ràng: Encode thuộc tính phân loại (*Cloud Cover*, *Season*, *Location*); chuẩn hoá z-score cho các thuộc tính số; khảo sát đặc trưng dữ liệu (phân bố nhãn, phân bố thuộc tính, tương quan và dấu hiệu ngoại lai).
- Thiết lập đánh giá theo tỉ lệ 8/1/1 cho *train/validation/test* với *stratified split*; lựa chọn hyperparameter trên *validation*, sau đó đánh giá cuối cùng trên *test*.
- So sánh SVM (kernel RBF), Logistic Regression (quy chuẩn hoá L2) và XGBoost dưới cùng điều kiện tiền xử lý và bộ hyperparameter xác định trước riêng cho từng loại; kết quả đánh giá bằng Accuracy, Macro-F1 và ma trận nhầm lẫn (Confusion matrix).

1.4 Cấu trúc bài báo

- Phần 1 giới thiệu mục tiêu và phạm vi.
- Phần 2 trình bày dữ liệu, khám phá và tiền xử lý.
- Phần 3 mô tả các mô hình học máy đơn giản được nhóm sử dụng: SVM (RBF), Logistic Regression (L2, softmax) và XGBoost.
- Phần 4 thiết lập thực nghiệm và tiêu chí đánh giá.
- Phần 5 báo cáo kết quả và phân tích.
- Phần 6 tóm tắt kết luận và hướng phát triển.

Source Code: Link Github

2 Dữ liệu và tiền xử lý dữ liệu

2.1 Tổng quan bộ dữ liệu

Nghiên cứu sử dụng bộ dữ liệu *Weather Type Classification* trên Kaggle bao gồm 13.200 bản ghi được tạo lập tổng hợp (synthetic) nhằm mô phỏng các điều kiện khí tượng đa dạng. Bộ dữ liệu bao gồm 11 thuộc tính đầu vào và 1 biến mục tiêu (*Weather Type*) với bốn trạng thái *Cloudy*, *Rainy*, *Snowy*, *Sunny*.
Các đặc điểm thống kê cơ bản của dữ liệu bao gồm:

- **Kích thước:** 13.200 mẫu. Quá trình kiểm tra sơ bộ xác nhận dữ liệu sạch, không chứa giá trị khuyết (missing values) và không có các bản ghi trùng lặp.
- **Biến mục tiêu (Target):** *Weather Type* là biến định danh gồm 4 nhãn: *Cloudy, Rainy, Snowy, Sunny.*
- **Biến định lượng (Numerical):** Bao gồm 7 thuộc tính: *Temperature, Humidity, Wind Speed, Precipitation (%), Atmospheric Pressure, UV Index, Visibility (km).*
- **Biến định danh (Categorical):** Bao gồm 3 thuộc tính: *Cloud Cover, Season, Location.*

2.2 Phân tích dữ liệu (EDA)

2.2.1 Phân bố nhãn (Target Distribution)

Biểu đồ tần suất tại Hình 1 cho thấy biến mục tiêu có sự phân bố đồng đều lý tưởng. Mỗi lớp thời tiết (*Rainy, Cloudy, Sunny, Snowy*) chiếm xấp xỉ 25% tổng số mẫu (khoảng 3.300 mẫu/lớp). Sự cân bằng này là một lợi thế lớn, giúp loại bỏ nguy cơ mô hình bị thiên kiến về phía lớp đa số (majority class) và cho phép sử dụng tin cậy các độ đo như *Accuracy* và *Macro-F1* trong quá trình đánh giá mà không cần áp dụng các kỹ thuật tái lấy mẫu (resampling).

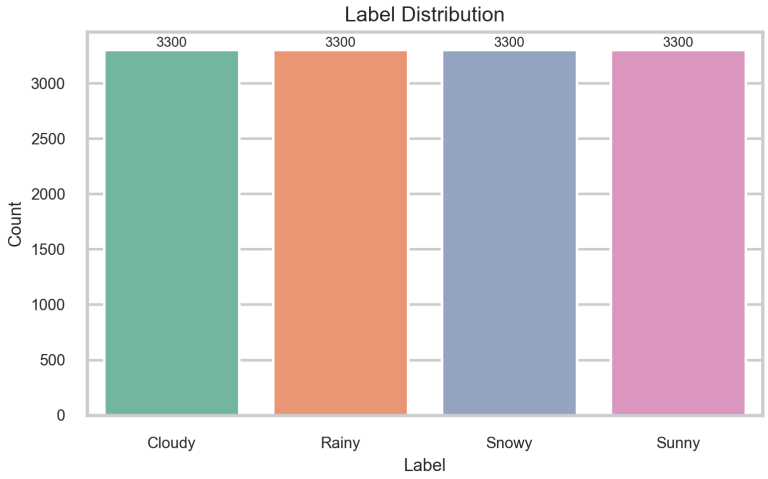

Figure 1: Phân bố số lượng mẫu giữa 4 nhãn thời tiết.

2.2.2 Phân tích thuộc tính số và độ lệch (Skewness)

Việc phân tích phân phối xác suất của các biến định lượng thông qua biểu đồ Histogram (Hình 2) và hệ số độ lệch (Skewness) chỉ ra các đặc điểm quan trọng sau:

- **Hiện tượng lệch phải (Right-skewed):** Hai thuộc tính *Wind Speed* (skewness ≈ 1.36) và *Visibility (km)* (skewness ≈ 1.23) có hệ số độ lệch lớn hơn 1. Biểu đồ cho thấy mật độ dữ liệu tập trung cao ở giá trị thấp và kéo dài về phía giá trị cao. Điều này ám chỉ sự tồn tại của các giá trị ngoại lai (outliers) hoặc các sự kiện cực đoan (gió rất mạnh, tầm nhìn rất xa).
- **Phân phối đa đỉnh (Multimodal):** Thuộc tính *Precipitation (%)* thể hiện phân phối đa đỉnh, phản ánh tính chất biến động phức tạp của lượng mưa, không tuân theo phân phối chuẩn.
- **Xấp xỉ chuẩn:** *Atmospheric Pressure* có phân phối tập trung mạnh quanh giá trị trung bình (1000 hPa) với hình dáng chuông (Bell-curve), tuy nhiên xuất hiện các giá trị ngoại lai ở hai cực của phân phối.

Quyết định xử lý: Để giảm thiểu tác động của độ lệch và nén dải giá trị của các biến đuôi dài, phương pháp biến đổi logarit tự nhiên $ln(1+x)$ (`np.log1p`) được áp dụng cho *Wind Speed* và *Visibility*. Các biến còn lại được giữ nguyên phân phối gốc trước khi đưa vào chuẩn hóa.

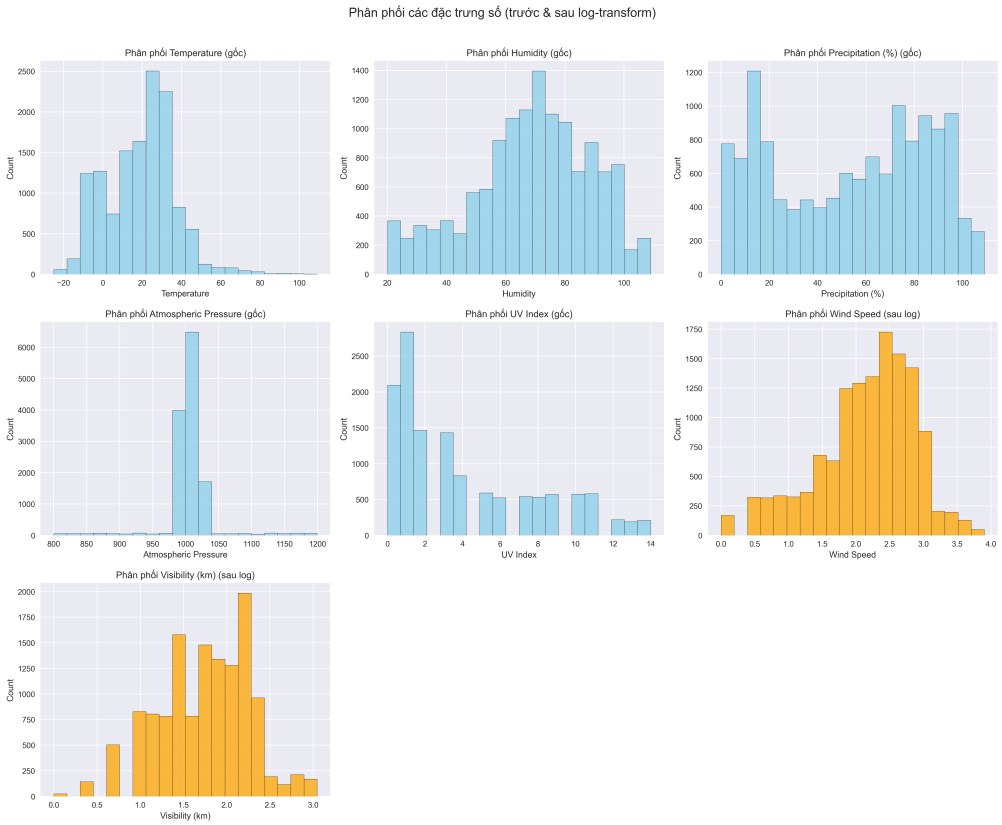

Figure 2: Phân phối của các thuộc tính số. Hàng trên thể hiện dữ liệu gốc, hàng dưới thể hiện dữ liệu sau khi biến đổi Logarit (đối với các biến có độ lệch cao).

2.2.3 Phân tích thuộc tính phân loại và mối quan hệ với nhãn

Biểu đồ phân phối chéo (Cross-tabulation) tại Hình 3 làm rõ mối tương quan giữa các thuộc tính định danh và loại thời tiết:

- **Cloud Cover (Độ che phủ mây):** Đây là đặc trưng có khả năng phân loại mạnh. Trạng thái *Clear* (quang mây) tương ứng gần như tuyệt đối với thời tiết *Sunny*. Ngược lại, trạng thái *Overcast* (u ám) chủ yếu dẫn đến *Rainy* hoặc *Cloudy*.
- **Season (Mùa) và Location (Vị trí):** Yếu tố mùa vụ và địa lý tác động rõ rệt đến thời tiết *Snowy*. Cụ thể, *Snowy* xuất hiện chủ yếu vào mùa *Winter* và tại khu vực *Mountain*. Trong khi đó, khu vực *Coastal* (ven biển) có xu hướng ít ghi nhận trường hợp *Snowy*.

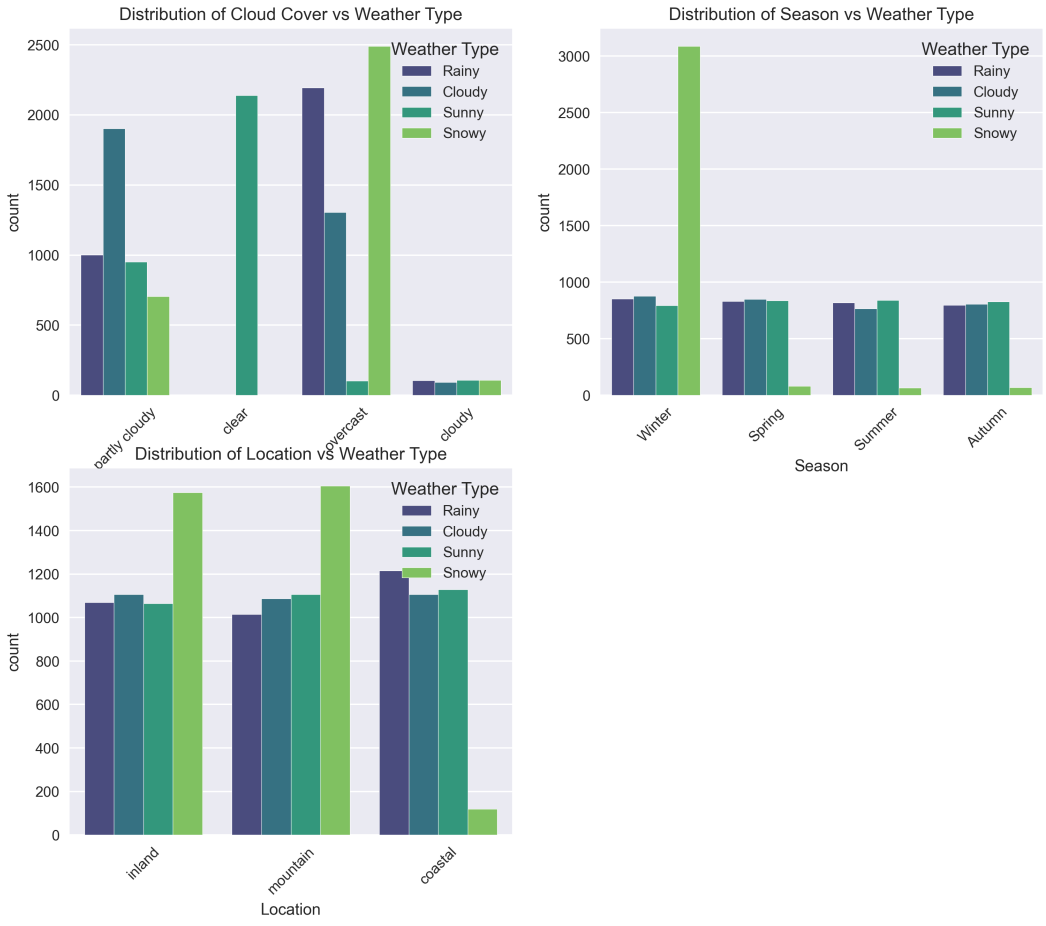

Figure 3: Phân bố loại thời tiết theo các thuộc tính phân loại (Cloud Cover, Season, Location).

2.2.4 Tương quan giữa các biến số

Ma trận tương quan Pearson (Hình 4) và biểu đồ Boxplot (Hình 5) cung cấp cái nhìn sâu hơn về mối quan hệ giữa các biến:

- **Đa cộng tuyến:** Không ghi nhận hiện tượng đa cộng tuyến nghiêm trọng giữa các biến độc lập (không có cặp biến nào có hệ số tương quan $|r| > 0.8$). Điều này cho phép giữ lại toàn bộ các đặc trưng cho quá trình huấn luyện.
- **Mối quan hệ vật lý:** *Humidity* (Độ ẩm) có tương quan dương với *Precipitation* ($r = 0.64$) và tương quan âm với *Visibility* ($r = -0.48$). Kết quả này phù hợp với quy luật vật lý: độ ẩm cao thường đi kèm mưa và làm giảm tầm nhìn.
- **Khả năng phân tách của Độ ẩm:** Phân tích Boxplot cho thấy *Humidity* là một đặc trưng quan trọng để phân biệt các lớp. Thời tiết *Sunny* có độ ẩm thấp nhất (trung vị $\approx 50\%$), phân biệt rõ ràng với *Rainy* và *Snowy* (độ ẩm thường trên 80%).

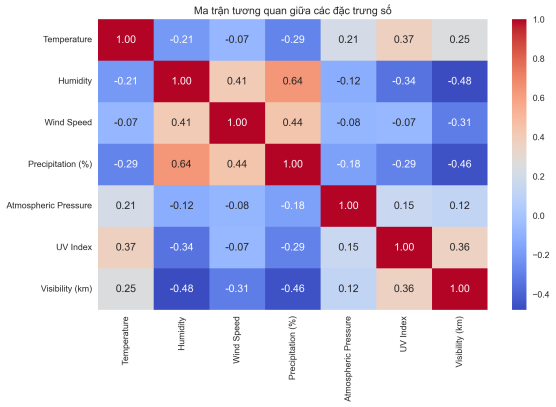

Figure 4: Ma trận tương quan Pearson giữa các biến định lượng.

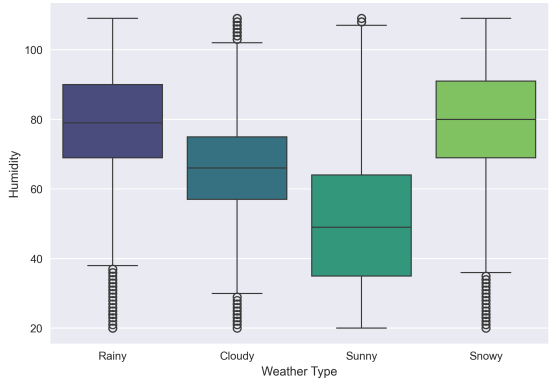

Figure 5: Biểu đồ hộp (Boxplot) thể hiện phân bố độ ẩm (Humidity) theo từng loại thời tiết.

Kiểm tra giá trị thiếu.

Thống kê trên toàn bộ 11 thuộc tính cho thấy không xuất hiện giá trị thiếu; mỗi hàng đều đầy đủ dữ liệu (0/13,200, tương đương 0.00%). Vì vậy không cần nội suy hay loại bỏ mẫu.

Feature	Missing (%)
Temperature	0.0%
Humidity	0.0%
Wind Speed	0.0%
Precipitation (%)	0.0%
Cloud Cover	0.0%
Atmospheric Pressure	0.0%
UV Index	0.0%
Season	0.0%
Visibility (km)	0.0%
Location	0.0%
Weather Type	0.0%

Table 1: Tỉ lệ missing của các thuộc tính

2.3 Quy trình tiền xử lý dữ liệu

Dựa trên các kết luận từ quá trình EDA, một quy trình tiền xử lý tự động (Pipeline) được thiết kế và hiện thực hóa thông qua `ColumnTransformer` của thư viện Scikit-learn.

2.3.1 Phân chia dữ liệu (Data Splitting)

Dữ liệu được chia thành 3 tập độc lập theo tỷ lệ **80/10/10**:

- **Training set (80%):** Sử dụng để huấn luyện mô hình.
- **Validation set (10%):** Sử dụng để tinh chỉnh siêu tham số và lựa chọn mô hình tối ưu.
- **Test set (10%):** Sử dụng để đánh giá hiệu năng cuối cùng.

Quá trình chia dữ liệu sử dụng phương pháp *Stratified Split* (phân tầng) dựa trên nhãn *Weather Type*. Điều này đảm bảo tỷ lệ các lớp nhãn được duy trì đồng nhất giữa các tập dữ liệu, tránh hiện tượng lệch phân phối (distribution shift).

2.3.2 Kỹ thuật xử lý đặc trưng (Feature Engineering)

Quy trình xử lý được áp dụng riêng biệt cho từng nhóm thuộc tính như sau:

1. **Xử lý biến số có phân phối lệch (Skewed Numerical Features):** Áp dụng cho *Wind Speed* và *Visibility (km)*.
   - *Bước 1:* Biến đổi logarit $x' = ln(1 + x)$ nhằm giảm độ lệch và nén các giá trị ngoại lai.
   - *Bước 2:* Chuẩn hóa (Standardization) để đưa dữ liệu về phân phối chuẩn tắc ($\mu = 0, \sigma = 1$)
2. **Xử lý biến số thông thường:** Áp dụng cho các biến số còn lại (*Temperature, Humidity, Precipitation, ...*).
   - Thực hiện chuẩn hóa `StandardScaler` để đồng bộ thang đo giữa các biến, hỗ trợ các thuật toán dựa trên khoảng cách (SVM) và gradient (Logistic Regression) hội tụ nhanh hơn.

3. **Xử lý biến phân loại:** Áp dụng cho *Cloud Cover, Season, Location.*
    - Sử dụng kỹ thuật `OneHotEncoder` để chuyển đổi các biến định danh thành các vector nhị phân. Tham số `handle_unknown='ignore'` được thiết lập để đảm bảo tính ổn định của hệ thống khi gặp các giá trị lạ trong quá trình kiểm thử.
4. **Mã hóa nhãn (Label Encoding):** Biến mục tiêu được chuyển đổi từ dạng chuỗi ký tự sang dạng số nguyên $(0, 1, 2, 3)$ bằng `LabelEncoder` để phù hợp với yêu cầu đầu vào của các thuật toán phân loại.

2.4 Tích hợp và quản lý luồng dữ liệu

Hệ thống tiền xử lý được đóng gói trong đối tượng `ColumnTransformer` (khởi tạo qua hàm `make_preprocessor`) và được tích hợp vào quy trình huấn luyện thông qua cơ chế lưu trữ và tải lại (serialization) các thành phần dữ liệu (artifacts). Để đảm bảo tính nhất quán giữa giai đoạn Tiền xử lý (Notebook 1) và giai đoạn Huấn luyện (Notebook 2), dữ liệu sau khi phân chia và cấu trúc bộ tiền xử lý được lưu trữ dưới dạng các tệp nhị phân thông qua thư viện `joblib`:

- **Dữ liệu đặc trưng (Features):** Các tập dữ liệu được lưu trữ nguyên bản (chưa qua biến đổi số học) để đảm bảo không rò rỉ thông tin thống kê.
    - Tập huấn luyện: `X_train_proc.pkl`
    - Tập kiểm định: `X_val_proc.pkl`
    - Tập kiểm tra: `X_test_proc.pkl`
- **Dữ liệu nhãn (Labels):** Các nhãn mục tiêu tương ứng được lưu tại `y_train.pkl`, `y_val.pkl`, và `y_test.pkl`.
- **Bộ tiền xử lý:** Cấu trúc `ColumnTransformer` (chưa thực hiện `fit`) được lưu tại `preprocessor.pkl`.

Trong giai đoạn mô hình hóa, các tệp này được tải lại. Một `Pipeline` của `scikit-learn` sẽ được thiết lập bằng cách ghép nối đối tượng `preprocessor` (tải từ `preprocessor.pkl`) với thuật toán phân loại. Quá trình biến đổi dữ liệu (transform) sẽ diễn ra trực tiếp bên trong Pipeline khi gọi hàm `fit` trên dữ liệu thô từ `X_train_proc.pkl`, đảm bảo rằng mọi bước chuẩn hóa đều được thực hiện khép kín trong quá trình huấn luyện.

3 Các mô hình học máy được sử dụng

Dựa trên đặc điểm phân phối dữ liệu và yêu cầu bài toán phân loại đa lớp, nghiên cứu lựa chọn ba mô hình đại diện cho các phương pháp tiếp cận khác nhau: Support Vector Machine (SVM) (phương pháp hình học/khoảng cách), Logistic Regression (phương pháp xác suất/tuyến tính) và XGBoost (phương pháp học kết hợp/cây quyết định).
Tất cả các mô hình được triển khai trong cùng một *Pipeline* với quy trình tiền xử lý thống nhất Φ. Ký hiệu $x \in \mathbb{R}^{11}$ là vector thuộc tính gốc và $z = \Phi(x)$ là vector đặc trưng sau tiền xử lý, bao gồm:

(i) Biến đổi $\ln(1+x)$ cho *Wind Speed* và *Visibility*, sau đó chuẩn hóa Z-score.

(ii) Chuẩn hóa Z-score cho các thuộc tính số còn lại.

(iii) Mã hóa One-Hot cho *Cloud Cover*, *Season*, *Location*.

3.1 Logistic Regression (Đa lớp)

3.1.1 Cơ sở lý thuyết

Logistic Regression đa lớp mô hình hóa xác suất qua hàm Softmax:

$$P(y = k \mid \mathbf{z}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{z} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^\top \mathbf{z} + b_j)}. \tag{1}$$

Hàm mất mát (Cross-Entropy + Ridge):

$$J(\mathbf{W}) = -\sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{I}(y_i = k) \log P(y_i = k \mid \mathbf{z}_i) + \frac{\lambda}{2} \|\mathbf{W}\|_F^2. \tag{2}$$

Tối ưu bằng L-BFGS, phù hợp với dữ liệu trung bình và đảm bảo hội tụ nhanh.

3.1.2 Chiến lược áp dụng và Tương thích dữ liệu

- **Vai trò baseline:** Logistic Regression là mô hình tuyến tính, dùng để đánh giá độ phi tuyến của bài toán.
- **Xử lý biến phân loại:** One-Hot giúp mô hình học trọng số cho từng trạng thái rời rạc của *Season*, *Location*, *Cloud Cover*.

3.2 Support Vector Machine (SVM) với Kernel RBF

3.2.1 Cơ sở lý thuyết

SVM tìm kiếm một siêu phẳng quyết định nhằm cực đại hóa lề (margin) giữa các lớp dữ liệu. Đối với dữ liệu không thể phân tách tuyến tính, phương pháp *Soft Margin* được áp dụng.
Bài toán tối ưu hoá được biểu diễn như sau:

$$\min_{\mathbf{w},b,\xi} \left(\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \right) \tag{3}$$

với điều kiện:

$$y_i(\mathbf{w}^\top \phi(x_i) + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0.$$

Để xử lý tính phi tuyến, sử dụng Kernel RBF:

$$K(\mathbf{z}, \mathbf{z}') = \exp\left(-\gamma \|\mathbf{z} - \mathbf{z}'\|^2\right), \qquad \gamma > 0. \tag{4}$$

3.2.2 Chiến lược áp dụng và Tương thích dữ liệu

- **One-vs-Rest:** Bài toán gồm 4 lớp → xây dựng 4 bộ phân lớp nhị phân độc lập.
- **Tiền xử lý:** SVM dựa trên khoảng cách nên bắt buộc chuẩn hóa Z-score. Biến đổi log giúp giảm ảnh hưởng outlier.
- **Tham số:** C điều chỉnh mức phạt sai số; γ xác định độ cong của biên quyết định.

3.3 XGBoost (Extreme Gradient Boosting)

3.3.1 Cơ sở lý thuyết

XGBoost là mô hình boosting dựa trên cây quyết định. Ở bước thứ t, cây mới f_t tối thiểu hóa:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(\mathbf{z}_i) + \frac{1}{2} h_i f_t^2(\mathbf{z}_i) \right] + \Omega(f_t), \tag{5}$$

với:

$$\Omega(f_t) = \gamma T + \frac{\lambda}{2} \|w\|^2.$$

Tiêu chí chọn điểm chia:

$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma. \tag{6}$$

3.3.2 Chiến lược áp dụng và Tương thích dữ liệu

- **Học tương tác phi tuyến:** XGBoost tự động mô hình hóa các tổ hợp phức tạp (temperature × precipitation → snowy).
- **Xử lý tốt dữ liệu hỗn hợp:** Không cần chuẩn hóa, nhưng log-transform giúp ổn định phân phối của *Wind Speed*.

4 Thiết lập thực nghiệm

Quy trình thực nghiệm được thiết kế để đảm bảo tính khách quan và khả năng tái lập, bao gồm các giai đoạn chuẩn bị dữ liệu, thiết lập không gian tìm kiếm tham số và đánh giá hiệu năng trên tập kiểm định.

4.1 Quy trình và Thiết lập chung

4.1.1 Dữ liệu và phân chia mẫu

Bộ dữ liệu *Weather Type Classification* (13.200 bản ghi) được phân chia theo tỷ lệ 80/10/10. Cụ thể:

- **Train (80%):** 10.560 mẫu dùng để huấn luyện mô hình.
- **Validation (10%):** 1.320 mẫu dùng để tinh chỉnh siêu tham số.
- **Test (10%):** 1.320 mẫu dùng để đánh giá hiệu năng cuối cùng.

Phương pháp phân chia *stratified split* với `random_state=42` được áp dụng nhằm duy trì tỷ lệ phân bố nhãn đồng nhất giữa các tập dữ liệu, tránh hiện tượng lệch pha phân phối (distribution shift).

4.1.2 Cấu hình tiền xử lý

Các bước xử lý đặc trưng được áp dụng nhất quán thông qua Pipeline:

- **Log-transform:** Áp dụng biến đổi ($\ln(1+x)$) cho hai thuộc tính có độ lệch cao là *Wind Speed* và *Visibility (km)*, sau đó chuẩn hoá *z-score*.
- **Standardization:** Chuẩn hoá *z-score* trực tiếp cho các thuộc tính số còn lại.
- **Encoding:** Mã hoá One-Hot cho các thuộc tính phân loại (*Cloud Cover*, *Season*, *Location*) với thiết lập `handle_unknown='ignore'` để xử lý các giá trị lạ.

4.1.3 Chiến lược đánh giá

Ba thuật toán được khảo sát bao gồm: **Logistic Regression** (đa lớp, L2), **SVM** (RBF) và **XGBoost**.

- **Tiêu chí chọn mô hình:** Chỉ số **Macro-F1** trên tập *Validation* là thước đo chính. Trong trường hợp điểm số tương đương, ưu tiên cấu hình có *Accuracy* cao hơn.
- **Chiến lược Refit:** Sau khi chọn được tham số tối ưu trên *Validation*, mô hình được huấn luyện lại trên tập hợp nhất (*Train + Validation*) trước khi đánh giá trên *Test*.

4.2 Không gian tìm kiếm siêu tham số

Các lưới tham số (Grid) được thiết lập để khảo sát khả năng của từng mô hình:

- **Logistic Regression:** Khảo sát tham số nghịch đảo điều chuẩn $C \in \{0.3, 1.0, 3.0, 10.0, 30.0\}$ với bộ giải `solver='lbfgs'`.
- **SVM (RBF):** Khảo sát tham số điều chuẩn $C \in \{0.3, 1.0, 3.0, 10.0\}$ và hệ số kernel ($\gamma \in$ `'scale'`, `'auto'`).

- **XGBoost:** Khảo sát 5 cấu hình đại diện (A, B, C, D, E) với sự thay đổi của các tham số cấu trúc cây: `n_estimators` (200–500), `max_depth` (3–6), `learning_rate` (0.03–0.1), `subsample` và `colsample_bytree`.

4.3 Kết quả tối ưu hóa trên tập Validation

4.3.1 Kết quả định lượng

Kết quả thực nghiệm trên tập Validation cho từng nhóm mô hình được trình bày chi tiết tại các bảng dưới đây.

Cấu hình	Tham số	Macro-F1 (val)	Accuracy (val)
C1	`C=10.0`	0.870600	0.870455
C3	`C=30.0`	0.870600	0.870455
C10	`C=1.0`	0.869818	0.869697
C30	`C=3.0`	0.869818	0.869697
C0.3	`C=0.3`	0.868267	0.868182

Table 2: Năm cấu hình Logistic Regression (L2, multinomial) và kết quả trên `validation`.

Cấu hình	Tham số	Macro-F1 (val)	Accuracy (val)
C10_gscale	`C=10.0, gamma=scale, probability=True, random_state=42`	0.907752	0.907576
C0.3_gscale	`C=3.0, gamma=scale, probability=True, random_state=42`	0.905384	0.904545
C3_gauto	`C=3.0, gamma=auto, probability=True, random_state=42`	0.904836	0.904545
C1_gscale	`C=1.0, gamma=scale, probability=True, random_state=42`	0.903493	0.903030
C3_gscale	`C=0.3, gamma=scale, probability=True, random_state=42`	0.903258	0.0.903030

Table 3: Năm cấu hình SVM (RBF) và kết quả trên `validation`.

Cấu hình	Tham số	Macro-F1 (val)	Accuracy (val)
C	`n_estimators=500, max_depth=4, learning_rate=0.05, subsample=0.8, colsample_bytree=1.0`	0.916141	0.915909
B	`n_estimators=300, max_depth=5, learning_rate=0.1, subsample=1.0, colsample_bytree=1.0`	0.915993	0.915909
E	`n_estimators=400, max_depth=3, learning_rate=0.03, subsample=1.0, colsample_bytree=0.8`	0.913776	0.913636
A	`n_estimators=200, max_depth=3, learning_rate=0.1, subsample=1.0, colsample_bytree=1.0`	0.912331	0.912121
D	`n_estimators=200, max_depth=6, learning_rate=0.05, subsample=0.8, colsample_bytree=0.8`	0.910829	0.910606

Table 4: Năm cấu hình XGBoost (`multi:softprob`) và kết quả trên `validation`.

4.3.2 Phân tích và Lựa chọn

Số liệu thực nghiệm cho thấy sự phân hóa rõ rệt về hiệu năng:

- **Logistic Regression:** Đạt mức trần hiệu năng tại (Macro-F1 ≈ 0.8706). Việc nới lỏng điều chuẩn (tăng C) không cải thiện kết quả, phản ánh giới hạn của mô hình tuyến tính đối với dữ liệu này.
- **Mô hình phi tuyến:** Cả SVM và XGBoost đều vượt trội, đạt ngưỡng > 0.90. XGBoost (Cấu hình C) đạt kết quả cao nhất (0.916141), nhỉnh hơn khoảng 0.8% so với SVM (cấu hình C10_gscale).
- **Tính ổn định:** Các biến thể cấu hình của XGBoost cho kết quả rất đồng đều, chứng tỏ mô hình ít nhạy cảm với các thay đổi nhỏ trong siêu tham số.

Dựa trên kết quả này, các bộ tham số tối ưu (Bảng 5) được lựa chọn để huấn luyện lại và kiểm thử.

Mô hình	**Cấu hình**	**Chi tiết tham số**	**Macro-F1 (Val)**
Logistic Regression	C1	`C=1.0`	0.8706
SVM (RBF)	C10_gscale	`C=10.0, gamma='scale'`	0.907752
XGBoost	C	`n_est=500, depth=4, lr=0.05, sub=0.8, col=1.0`	0.916141

Table 5: Bộ siêu tham số tối ưu được lựa chọn cho giai đoạn kiểm thử.

5 Kết quả và phân tích đánh giá

Phần này trình bày kết quả thực nghiệm theo quy trình đã thiết lập tại Mục 4. Sau khi xác định bộ siêu tham số tối ưu trên tập `validation`, các mô hình được huấn luyện lại trên tập dữ liệu hợp nhất và đánh giá một lần duy nhất trên tập `test` để đảm bảo tính khách quan. Kết quả được phân tích theo ba khía cạnh: tổng quan hiệu năng, chi tiết từng mô hình và so sánh đối chiếu theo nhãn.

5.1 Tổng quan kết quả trên tập kiểm tra

Bảng 6 tóm tắt hiệu năng của ba mô hình. Kết quả cho thấy xu hướng tăng dần về độ chính xác từ mô hình tuyến tính đến mô hình phi tuyến và mô hình tổ hợp.

Mô hình	**Macro-F1**	**Accuracy**
Logistic Regression (L2, Multinomial)	0.874634	0.874242
SVM (Kernel RBF)	0.913052	0.912879
XGBoost (Ensemble)	**0.9199**	**0.919697**

Table 6: Tổng hợp hiệu năng trên tập `test`.

Bảng 7 trình bày điểm F1-score của từng nhãn thời tiết, cho thấy sự cải thiện đáng kể ở các nhãn khó như *Cloudy* và *Rainy* khi chuyển sang các mô hình phi tuyến.

Nhãn	**Logistic Regression**	**SVM (RBF)**	**XGBoost**
Cloudy	0.8360	0.8932	0.8964
Rainy	0.8610	0.8997	0.9163
Snowy	0.9077	0.9309	0.9364
Sunny	0.8938	0.9285	0.9305

Table 7: Chi tiết F1-score theo nhãn trên tập `test`.

5.2 Phân tích chi tiết từng mô hình

5.2.1 Logistic Regression

Kết quả định lượng và trực quan hóa (Hình 6) cho thấy giới hạn của mô hình tuyến tính:

- **Ma trận nhầm lẫn:** Mô hình gặp khó khăn lớn khi phân tách *Cloudy* và *Rainy*. Có 29 mẫu *Cloudy* bị nhầm thành *Rainy*, và 23 mẫu *Rainy* bị nhầm thành *Cloudy*. Số mẫu đúng của *Cloudy* chỉ đạt 283.
- **Đường cong ROC:** AUC trung bình trên 0.95; tuy nhiên, lớp *Cloudy* (AUC = 0.9553) và *Rainy* (AUC = 0.9541) thấp hơn đáng kể so với *Sunny* và *Snowy*, phản ánh sự chồng lấp dữ liệu.

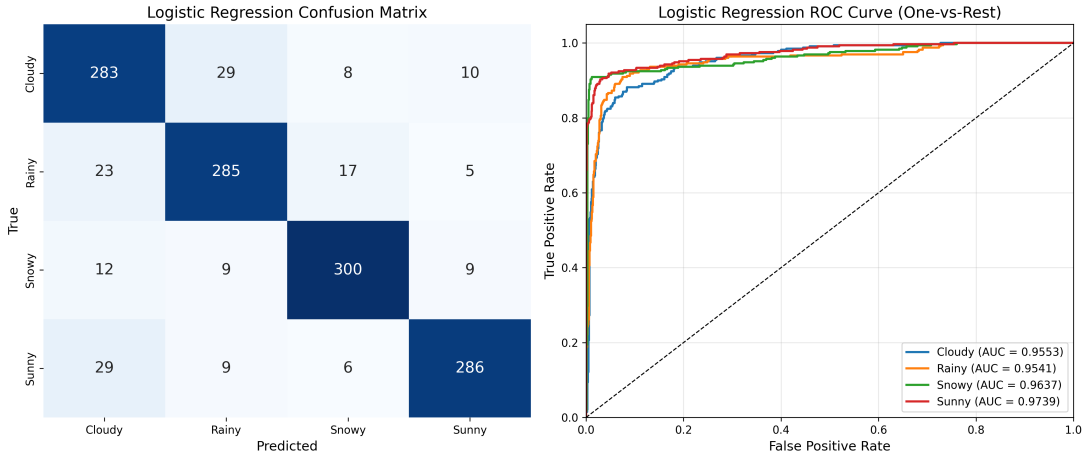

Figure 6: Logistic Regression trên `test`: (trái) Ma trận nhầm lẫn; (phải) Đường cong ROC.

5.2.2 Support Vector Machine (SVM)

Kernel RBF giúp SVM ánh xạ dữ liệu sang không gian cao chiều và cải thiện đáng kể độ phân tách (Hình 7):

- **Cải thiện phân lớp:** *Cloudy* tăng từ 283 (LogReg) lên 301 mẫu đúng; *Rainy* từ 285 lên 296. Tổng lỗi giữa hai nhãn này giảm mạnh.
- **ROC:** Tất cả AUC đều vượt 0.98; các đường cong tiệm cận góc trên trái, cho thấy khả năng mô hình hóa mối quan hệ phi tuyến tốt.

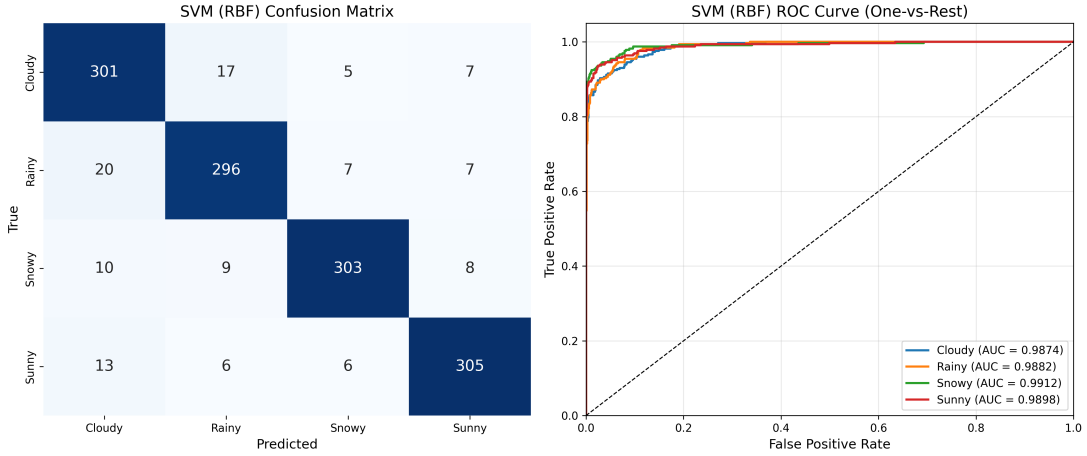

Figure 7: SVM (RBF) trên `test`: (trái) Ma trận nhầm lẫn; (phải) Đường cong ROC.

5.2.3 XGBoost

XGBoost thể hiện hiệu năng vượt trội nhất (Hình 8):

- **Độ chính xác cao:** *Sunny* đạt 308 mẫu đúng; *Cloudy* đạt 303 mẫu đúng — cao nhất trong ba mô hình.
- **Giảm thiểu sai số:** Các ô ngoài đường chéo hầu như tiệm cận 0; ví dụ, chỉ 3 mẫu *Snowy* bị nhầm sang *Rainy*.
- **ROC:** AUC gần 0.996 cho *Snowy* và *Sunny*, thể hiện ranh giới quyết định cực kì sắc nét.

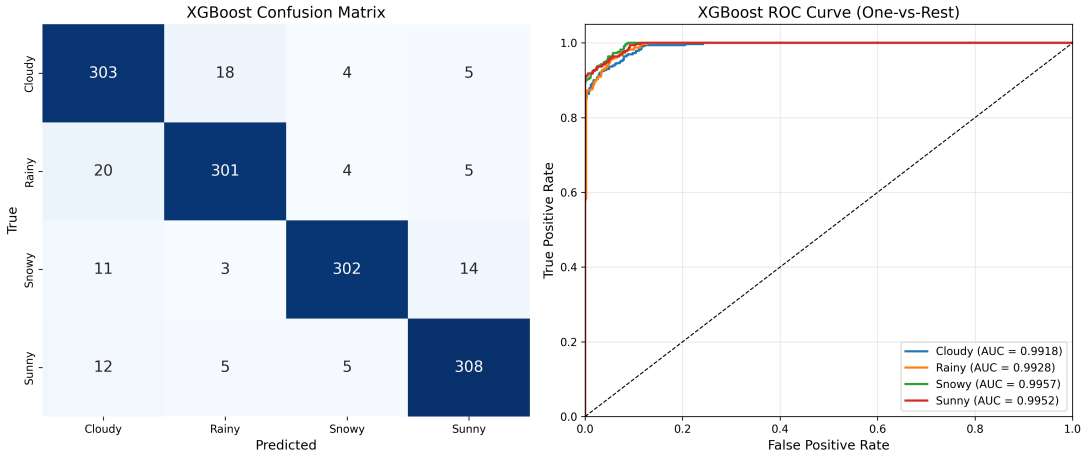

Figure 8: XGBoost trên `test`: (trái) Ma trận nhầm lẫn; (phải) Đường cong ROC.

5.3 So sánh đối chiếu

Bảng 8 trình bày mức chênh lệch F1-score (ΔF1) của XGBoost so với hai mô hình khác.

Nhãn	**ΔF1 (vs SVM)**	**ΔF1 (vs LogReg)**
Cloudy	+0.0033	+0.0604
Rainy	+0.0166	+0.0553
Snowy	+0.0056	+0.0287
Sunny	+0.0021	+0.0368

Table 8: Mức cải thiện F1-score (Δ) của XGBoost so với SVM và Logistic Regression.

- **Cloudy/Rainy:** Đây là khu vực giao thoa dữ liệu mạnh nhất. XGBoost hơn Logistic Regression khoảng +0.06 và hơn SVM khoảng +0.003, thể hiện ưu thế của mô hình cây trong việc học các ngưỡng đặc trưng.
- **Snowy/Sunny:** Hai nhãn này có biên tách tự nhiên; SVM và XGBoost có hiệu năng gần tương đương ($\Delta \approx 0.002-0.005$).

Tổng kết. Thứ hạng hiệu năng nhất quán trên tập kiểm tra:

$$\textbf{XGBoost} > \textbf{SVM (RBF)} > \textbf{Logistic Regression}.$$

Macro-F1 trên `test` của XGBoost đạt 0.9199, gần với kết quả trên `validation` (0.916141), chứng tỏ mô hình tổng quát tốt và không bị quá khớp.

6 Kết luận

6.1 Tổng kết quy trình nghiên cứu

Báo cáo đã xây dựng và thực nghiệm một quy trình nhất quán cho bài toán *phân loại thời tiết* trên tập dữ liệu tổng hợp. Quy trình tiền xử lý được thiết kế dựa trên đặc tính phân phối của dữ liệu, bao gồm:

(i) Áp dụng biến đổi ($ln(1+x)$) cho hai thuộc tính có độ lệch cao (*Wind Speed*, *Visibility*) và chuẩn hoá *z-score* cho toàn bộ biến định lượng.

(ii) Mã hoá *One-Hot* cho các thuộc tính định danh (*Cloud Cover*, *Season*, *Location*).

(iii) Phân chia dữ liệu theo tỷ lệ **8/1/1** (Train/Validation/Test) với chiến lược phân tầng (*stratified*) để đảm bảo tính đại diện của mẫu.

Ba mô hình học máy đại diện gồm **Logistic Regression**, **SVM (Kernel RBF)** và **XGBoost** đã được cài đặt, tối ưu hóa siêu tham số trên tập *validation* và đánh giá độc lập trên tập *test*.

6.2 Đánh giá hiệu năng mô hình

Kết quả thực nghiệm cho thấy sự vượt trội của các phương pháp phi tuyến đối với dữ liệu khí tượng hỗn hợp:

- **XGBoost** đạt hiệu năng cao nhất với **Macro-F1 = 0.9199** và Accuracy = 0.9197.
- **SVM (RBF)** bám sát với **Macro-F1 = 0.9131**.
- **Logistic Regression** đạt kết quả thấp nhất (**Macro-F1 = 0.8746**), đóng vai trò là mức chuẩn (baseline) tuyến tính.

Phân tích sâu theo từng nhãn cho thấy sự chênh lệch hiệu năng chủ yếu nằm ở cặp nhãn *Cloudy* và *Rainy*. Đây là vùng giao thoa dữ liệu phức tạp, nơi các luật quyết định phân cấp của XGBoost và ranh giới mềm của SVM xử lý hiệu quả hơn hẳn so với đường ranh giới cứng của mô hình tuyến tính. Ngược lại, các nhãn *Snowy* và *Sunny* được phân loại rất tốt bởi cả ba mô hình nhờ các đặc trưng phân biệt rõ rệt.

6.3 Hạn chế của nghiên cứu

Mặc dù đạt kết quả khả quan, nghiên cứu vẫn tồn tại một số hạn chế:

1. **Dữ liệu:** Bộ dữ liệu *synthetic* có thể chưa phản ánh đầy đủ tính nhiễu và sự bất định của dữ liệu thời tiết thực tế.
2. **Không gian tìm kiếm:** Quá trình tối ưu siêu tham số mới chỉ dừng lại ở phương pháp Grid Search trên không gian nhỏ (5 cấu hình đại diện cho XGBoost), chưa khai thác hết tiềm năng tối đa của mô hình.
3. **Phương pháp đánh giá:** Việc sử dụng chiến lược tách mẫu cố định (hold-out) 8/1/1, tuy tiết kiệm tài nguyên, vẫn tiềm ẩn rủi ro phụ thuộc vào cách chia dữ liệu ngẫu nhiên so với phương pháp kiểm định chéo (Cross-Validation).

6.4 Hướng phát triển

Để cải thiện và mở rộng nghiên cứu, các hướng đi sau được đề xuất:

- **Tối ưu hóa nâng cao:** Áp dụng *Bayesian Optimization* để dò tìm siêu tham số hiệu quả hơn, đặc biệt là các tham số điều chuẩn (α, λ) cho XGBoost.
- **Mở rộng mô hình:** Thử nghiệm các thuật toán boosting hiện đại khác như *LightGBM* hoặc *CatBoost*, vốn được tối ưu hóa tốt hơn cho dữ liệu có biến phân loại.
- **Kiểm định chéo (K-Fold):** Thay thế việc chia tập cố định bằng kiểm định chéo 5 hoặc 10 lần (folds) để đánh giá độ ổn định và phương sai của mô hình chính xác hơn.
- **Phân tích lỗi (Error Analysis):** Tập trung phân tích các mẫu bị phân loại sai (misclassified cases) để đề xuất các kỹ thuật trích chọn đặc trưng (feature engineering) mới, giúp tách biệt tốt hơn hai lớp *Cloudy* và *Rainy*.

6.5 Kiến nghị thực tiễn

Dựa trên kết quả phân tích, **XGBoost** là sự lựa chọn được kiến nghị cho bài toán này nhờ khả năng cân bằng tốt nhất giữa độ chính xác tổng thể và khả năng xử lý các lớp dữ liệu khó. SVM với Kernel RBF là phương án dự phòng tin cậy nếu ưu tiên tính ổn định dựa trên khoảng cách hình học.

Tài liệu tham khảo

Nguồn tham khảo

[1] S. Raschka. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv:1811.12808*, 2018. Available at: https://arxiv.org/abs/1811.12808.

[2] C. Cortes, V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018.

[3] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

[4] T. Chen, C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.

[5] D. D. Nguyen. *Machine Learning – Support Vector Machine (Chapter 6)*. Lecture notes, Faculty of Computer Science and Engineering, HCMUT.

[6] Nikhil7280. *Weather Type Classification Dataset*. Kaggle. Truy cập ngày 22 tháng 11 năm 2025.
URL: https://www.kaggle.com/datasets/nikhil7280/weather-type-classification