

# Phân loại thời tiết trên dữ liệu bảng với Logistic Regression & SVM & XGBoost

Báo cáo Bài tập lớn Học máy (CO3117)

---

Hồ Anh Dũng (2310543) Đào Quang Dương (2310579)

Hà Bảo Nhi (2312496) Mai Doãn Chiến (2110060)

Tháng 11, 2025

Khoa Khoa học và Kỹ thuật Máy tính  
Trường Đại học Bách Khoa, ĐHQG-HCM

## 1. Giới thiệu chung

---

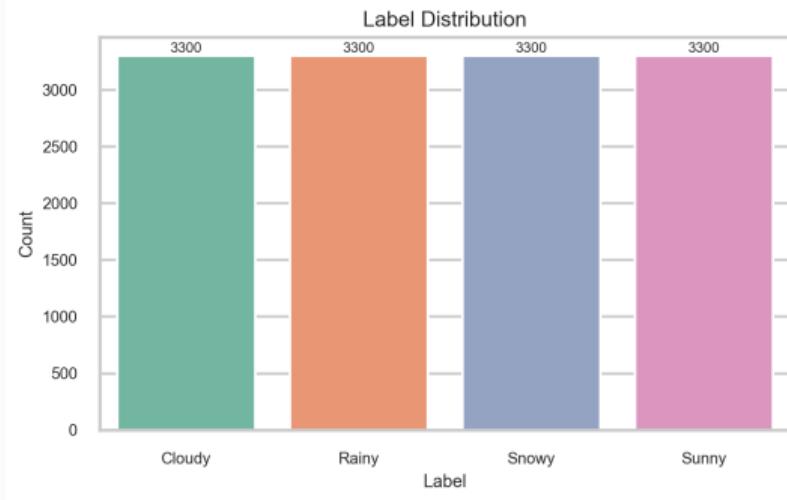
# Bối cảnh & Mục tiêu

## Bối cảnh:

- Bài toán dự đoán nhãn thời tiết (*Cloudy, Rainy, Snowy, Sunny*) từ các thuộc tính số và phân loại.
- Sử dụng bộ dữ liệu *Weather Type Classification* (Kaggle) - dữ liệu tổng hợp có chứa nhiễu và ngoại lai (outliers).

## Mục tiêu:

- Xây dựng quy trình tiền xử lý dữ liệu chuẩn (xử lý lệch, chuẩn hóa, mã hóa).
- So sánh hiệu năng của ba mô hình đại diện: **Logistic Regression**, **SVM (RBF)**, và **XGBoost**.
- Đánh giá dựa trên Accuracy, Macro-F1 và



**Hình 1:** Phân bố 4 nhãn thời tiết cân bằng.

## **2. Dữ liệu và Tiền xử lý dữ liệu**

---

# Phân tích dữ liệu (EDA): Thuộc tính số

## Phân tích độ lệch (Skewness):

### ▪ Lệch phải mạnh (Skew > 1):

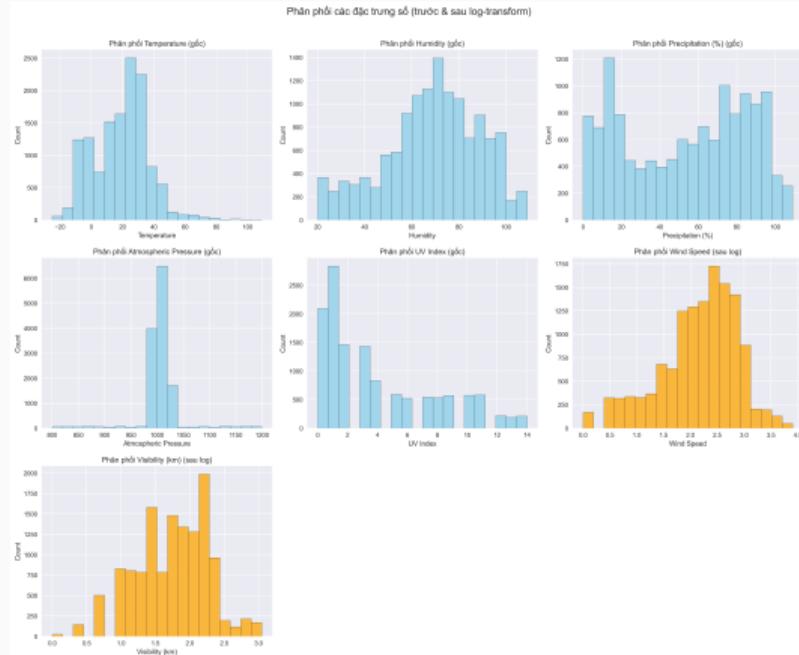
- *Wind Speed* ( $\text{Skew} \approx 1,36$ ).
- *Visibility* ( $\text{Skew} \approx 1,23$ ).
- **Đặc điểm:** Mật độ tập trung ở giá trị thấp, đuôi dài về phía giá trị cao chứa ngoại lai.

### ▪ Các biến còn lại:

- *Precipitation*: Phân phối đa đỉnh (Multimodal).
- *Atmospheric Pressure*: Xấp xỉ chuẩn (Bell-curve).

## Quyết định xử lý:

- Chỉ áp dụng biến đổi **Logarit**  $\ln(1 + x)$  cho *Wind Speed* và *Visibility*.
- Các biến khác giữ nguyên phân phối gốc trước khi chuẩn hóa Z-score.

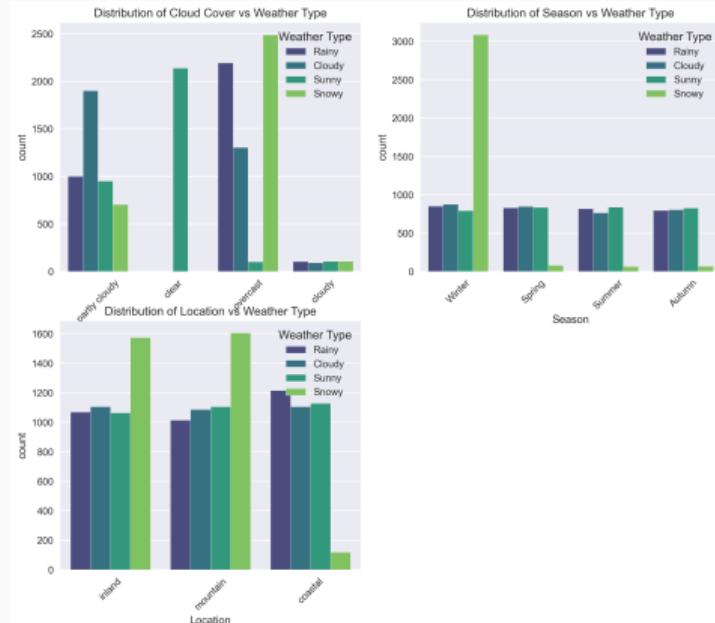


Hình 2: Phân phối trước (trên) và sau (dưới) biến đổi Logarit.

# Phân tích dữ liệu (EDA): Thuộc tính phân loại và quan hệ với nhãn

## Quan hệ giữa thuộc tính và nhãn:

- **Cloud Cover (Mây):** Phân loại rất mạnh.
  - *Clear* → Tương ứng tuyệt đối với **Sunny**.
  - *Overcast* → Chủ yếu là **Rainy** hoặc **Cloudy**.
- **Season (Mùa):** Tính chất mùa vụ rõ rệt.
  - *Winter* là điều kiện tiên quyết cho **Snowy**.
- **Location (Vị trí):**
  - *Mountain* → Thiên về **Snowy**.
  - *Coastal* → Thiên về **Rainy**.

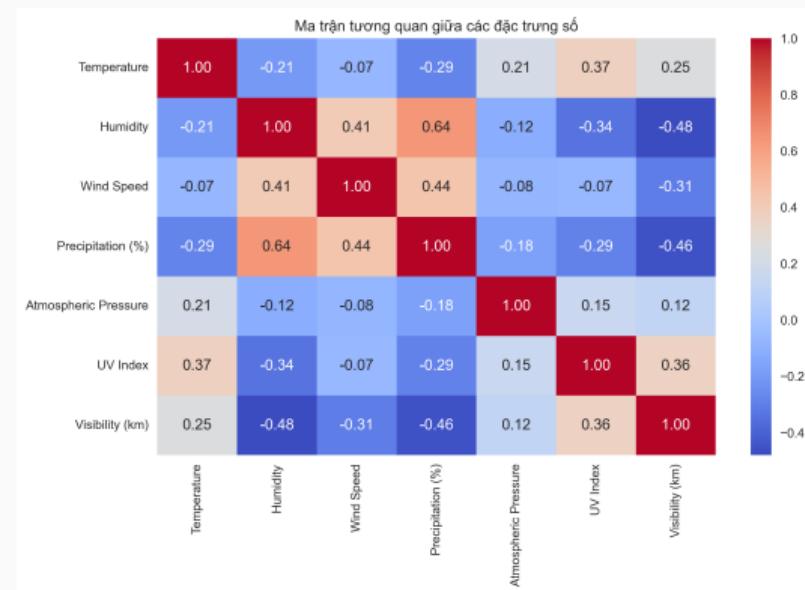


**Hình 3:** Phân bố nhãn theo từng thuộc tính phân loại.

# Phân tích dữ liệu (EDA): Tương quan biến số

## Ma trận tương quan Pearson:

- **Đa cộng tuyến:** Không phát hiện cặp biến nào có hệ số tương quan quá cao ( $|r| > 0.8$ ), cho phép giữ lại toàn bộ đặc trưng.
- **Mối quan hệ vật lý hợp lý:**
  - *Humidity* tỉ lệ thuận với *Precipitation* ( $r = 0.64$ ).
  - *Humidity* tỉ lệ nghịch với *Visibility* ( $r = -0.48$ ).
- **Kết luận:** Các biến số độc lập tương đối tốt, hỗ trợ mô hình học được nhiều khía cạnh dữ liệu.



Hình 4: Heatmap tương quan giữa các biến số.

# Quy trình Tiên xử lý (Pipeline)

Quy trình được tự động hóa bằng ColumnTransformer và Pipeline:

- **Bước 1: Xử lý biến lệch (Skewed Features)**
  - Áp dụng  $\ln(1 + x)$  cho *Wind Speed* và *Visibility*.
  - Sau đó chuẩn hóa Z-score.
- **Bước 2: Xử lý biến số thường**
  - Chuẩn hóa Z-score ( $\mu = 0, \sigma = 1$ ) cho các biến còn lại (*Temp*, *Humidity*...).
  - *Lý do:* Hỗ trợ SVM và Logistic Regression hội tụ tốt hơn.
- **Bước 3: Xử lý biến phân loại**
  - Mã hóa **One-Hot** cho *Cloud Cover*, *Season*, *Location*.
  - Thiết lập `handle_unknown='ignore'` để xử lý giá trị lạ khi kiểm thử.

### **3. Các mô hình học máy được sử dụng**

---

## Mô hình 1: Logistic Regression (Baseline)

- **Loại mô hình:** Tuyến tính, đa lớp (Multinomial).
- **Hàm kích hoạt:** Softmax để tính xác suất thuộc về từng lớp  $k$ :

$$p(y = k | \mathbf{z}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{z} + b_k)}{\sum \exp(\mathbf{w}_j^\top \mathbf{z} + b_j)}$$

- **Hàm mất mát:** Cross-Entropy với điều chỉnh  $L_2$  (Ridge) để tránh quá khớp.
- **Vai trò:** Làm mức chuẩn (baseline) để đánh giá độ phức tạp phi tuyến của dữ liệu.

## Mô hình 2: Support Vector Machine (SVM)

### Cơ sở lý thuyết:

- Tìm siêu phẳng tối ưu cực đại hóa lề (margin) giữa các lớp.
- Sử dụng **Kernel RBF** (Radial Basis Function) để ánh xạ dữ liệu sang không gian cao chiều:

$$K(\mathbf{z}, \mathbf{z}') = \exp(-\gamma \|\mathbf{z} - \mathbf{z}'\|^2)$$

### Ưu điểm:

- Hiệu quả với dữ liệu có ranh giới phi tuyến phức tạp.
- Ổn định nhờ cơ chế tối ưu hóa lồi toàn cục.

SVM đặc biệt mạnh mẽ khi không gian đặc trưng được chuẩn hóa tốt (Z-score).

## Mô hình 3: XGBoost (Ensemble)

### Cơ chế hoạt động:

- Kết hợp tuần tự các cây quyết định (Gradient Boosting).
- Mỗi cây mới học sửa lỗi (residual) của các cây trước đó.
- Hàm mục tiêu tối ưu hóa cả sai số dự đoán và độ phức tạp của cây (Regularization).

XGBoost thường là SOTA (State-of-the-Art) cho dữ liệu dạng bảng.

### Ưu điểm với dữ liệu bảng:

- Tự động học các tương tác phi tuyến giữa các đặc trưng.
- Xử lý tốt các ngưỡng quyết định (ví dụ:  $Humidity > 80\%$  thì mưa).

## **4. Thiết lập thực nghiệm**

---

# Thiết lập thực nghiệm

- **Phân chia dữ liệu:**
  - Tỷ lệ **80% / 10% / 10%** cho Train / Validation / Test.
  - Phương pháp **Stratified Split**: Đảm bảo tỷ lệ 4 nhãn thời tiết đồng đều trong mọi tập con.
- **Chiến lược đánh giá:**
  - Tinh chỉnh siêu tham số (Hyperparameter Tuning) trên tập **Validation**.
  - Tiêu chí chọn mô hình: **Macro-F1** cao nhất.
  - Huấn luyện lại (Refit) trên tập hợp nhất (Train + Val) trước khi đánh giá trên Test.
- **Không gian tham số:**
  - **LogReg**:  $C \in \{0.3, 1, 3, 10, 30\}$ .
  - **SVM**:  $C \in \{0.3, 1, 3, 10\}$ ,  $\gamma \in \{\text{scale, auto}\}$ .
  - **XGBoost**: Grid Search trên 5 cấu hình đại diện (depth, learning rate, n\_estimators).

## **5. Kết quả và Phân tích đánh giá**

---

# Kết quả tổng quan trên tập Test

## Nhận xét chung:

- Hiệu năng tăng dần từ mô hình tuyến tính đến phi tuyến.
- XGBoost** đạt kết quả cao nhất ở mọi chỉ số.
- SVM bám sát XGBoost, khẳng định tính hiệu quả của Kernel RBF.

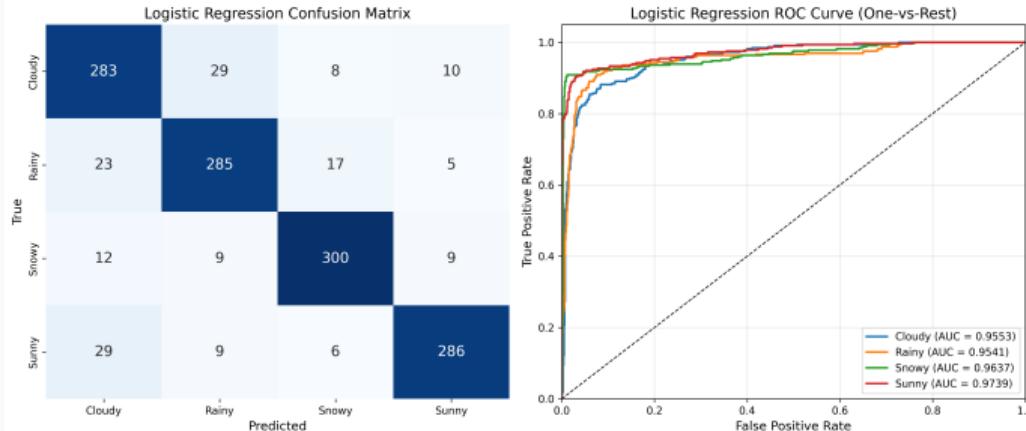
Mô hình	Macro-F1	Accuracy
Logistic Regression	0.874634	0.874242
SVM	0.913052	0.912879
<b>XGBoost</b>	<b>0.9199</b>	<b>0.919697</b>

Nhãn	$\Delta F1$ (vs SVM)	$\Delta F1$ (vs LogReg)
Cloudy	+0.0033	+0.0604
Rainy	+0.0166	+0.0553
Snowy	+0.0056	+0.0287
Sunny	+0.0021	+0.0368

## Chi tiết theo nhãn (F1-score):

- Cloudy/Rainy:** XGBoost cải thiện đáng kể so với LogReg ( $\Delta \approx +0.06$ ).
- Snowy/Sunny:** SVM và XGBoost có hiệu năng gần tương đương ( $\Delta \approx 0.002$ – $0.005$ ).

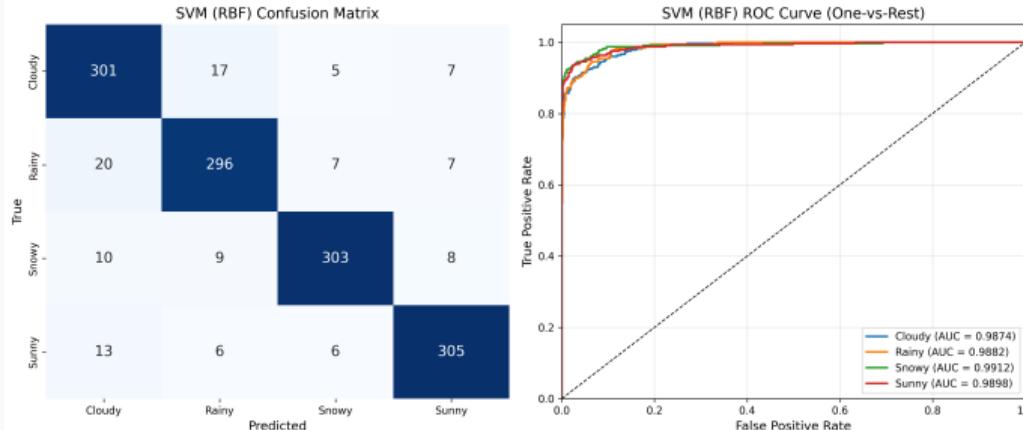
# Phân tích chi tiết: Logistic Regression



## Hạn chế:

- **Nhầm lẫn lớn:** Giữa *Cloudy* và *Rainy* (tổng cộng 52 ca nhầm lẫn).
- **Nguyên nhân:** Ranh giới tuyến tính không thể phân tách vùng giao thoa dữ liệu của hai lớp này (độ ẩm cao, ít nắng).
- **ROC:** Đường cong của *Cloudy/Rainy* thấp hơn hẳn so với *Sunny/Snowy*.

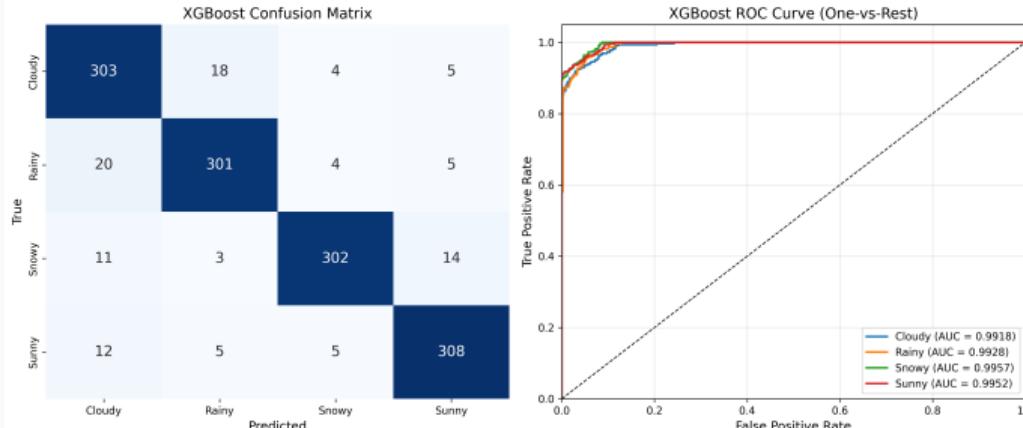
# Phân tích chi tiết: Support Vector Machine (SVM)



## Hiệu quả của Kernel RBF:

- **Cải thiện rõ rệt:** Macro-F1 đạt **0.9086**, vượt xa mô hình tuyến tính (0.8831).
- **Xử lý vùng giao thoa:** Số lượng mẫu nhầm lẫn giữa *Cloudy* và *Rainy* giảm mạnh so với Logistic Regression.
- **ROC Curve:** AUC trung bình đạt  $> 0.98$ , đường cong tiệm cận góc trái trên, chứng tỏ khả năng phân tách tốt trong không gian cao chiều.

# Phân tích chi tiết: XGBoost (Best Model)



## Ưu điểm vượt trội:

- Độ chính xác cao:** Nhãn *Sunny* đạt 308 mẫu đúng, *Cloudy* đạt 303 mẫu (cao nhất).
- Giảm sai số:** Các ô nhầm lẫn ngoài đường chéo có giá trị rất nhỏ.
- ROC:** AUC đạt xấp xỉ 0.996 cho các lớp dễ (*Sunny/Snowy*), đường cong gần như vuông góc.

## 6. Kết luận & Hướng phát triển

---

# Kết luận & Kiến nghị

- **Về dữ liệu:**
  - Quy trình tiền xử lý ( $\text{Log1p} + \text{Z-score}$ ) là bước đệm quan trọng giúp ổn định mô hình.
  - Dữ liệu có tính phi tuyến mạnh, đặc biệt ở cặp nhãn *Cloudy - Rainy*.
- **Về mô hình:**
  - **XGBoost** là lựa chọn tối ưu nhất, cân bằng giữa độ chính xác và khả năng tổng quát hóa.
  - SVM là phương án dự phòng tin cậy. Logistic Regression chỉ nên dùng làm baseline.

**Hạn chế:** Dữ liệu synthetic có thể chưa phản ánh hết độ nhiễu thực tế; không gian tìm kiếm tham số còn nhỏ.

# Hướng phát triển

## Kỹ thuật:

- Áp dụng **Bayesian Optimization** để tối ưu siêu tham số hiệu quả hơn Grid Search.
- Thủ nghiệm các thuật toán Boosting khác như **LightGBM**, **CatBoost** (tối ưu cho biến phân loại).

## Đánh giá:

- Sử dụng **K-Fold Cross-Validation** thay cho hold-out để đánh giá độ ổn định tốt hơn.
- Phân tích sâu các miscalculation (Error Analysis) để đề xuất thêm đặc trưng mới (feature engineering).

Cảm ơn Quý Thầy và các bạn đã  
lắng nghe!