BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG

PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO MÔN HỌC MÁY

Tên đề tài:

PHÂN TÍCH, XỬ LÝ DỮ LIỆU THẢM HỌA TOÀN CẦU VÀ TIẾN HÀNH DỰ ĐOÁN

Giảng viên hướng dẫn:	**Vũ Thị Hạnh**
Sinh viên thực hiện:	***Nguyễn Thị Tuyết Nhung*** ***Trần Nguyễn Diễm Hạnh*** ***Lê Văn Tính***
Mssv:	***2351067105*** ***2351067093*** ***2351067117***
Lớp:	***S26-65CNTT***

2025

Lời Cám Ơn

Để hoàn thành được bài tiểu luận này, em xin chân thành cảm ơn Ban Giám hiệu, các khoa, phòng và quý thầy, cô của trường Phân hiệu Đại học Thuỷ Lợi, những người đã tận tình giúp đỡ và tạo điều kiện cho em trong quá trình học tập. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến cô Vũ Thị Hạnh - người đã trực tiếp giảng dạy và hướng dẫn em thực hiện bài tiểu luận này bằng tất cả lòng nhiệt tình và sự quan tâm sâu sắc.

Trong quá trình thực hiện bài tiểu luận này, do hiểu biết còn nhiều hạn chế nên bài làm khó tránh khỏi những thiếu sót. Em rất mong nhận được những lời góp ý của quý thầy cô để bài tiểu luận ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

Mục Lục

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TPHCM, ngày 24 tháng 12 năm 2025

Ký và ghi rõ họ tên

Vũ Thị Hạnh

Chương I: Giới Thiệu Bài Toán

1. Đặt vấn đề và giới thiệu đề tài:

- Trong những năm gần đây, thế giới chứng kiến sự gia tăng đáng kể về tần suất và mức độ nghiêm trọng của các thảm họa toàn cầu, từ thiên tai tự nhiên như bão, lũ lụt, hạn hán, động đất đến các thảm họa nhân tạo như dịch bệnh, xung đột hay tai nạn công nghiệp. Những thảm họa này không chỉ gây thiệt hại lớn về sinh mạng và tài sản mà còn ảnh hưởng sâu rộng đến kinh tế, xã hội và môi trường của các quốc gia, đặc biệt là những nước đang phát triển với hệ thống phòng chống thiên tai còn hạn chế.

- Một trong những thách thức lớn hiện nay là dự báo sớm và ứng phó kịp thời. Việc không có thông tin dự báo chính xác hoặc phối hợp cứu trợ chậm trễ có thể làm gia tăng thiệt hại và kéo dài quá trình phục hồi của cộng đồng. Trong bối cảnh dữ liệu về thảm họa ngày càng phong phú và sẵn có từ các nguồn quốc tế, việc khai thác dữ liệu này để xây dựng các mô hình dự báo thảm họa trở nên rất cần thiết.

- Trong đồ án này, nhóm thực hiện bài toán hồi quy dự đoán mức độ nghiêm trọng của thiên tai (severity_index) dựa trên các đặc trưng về hậu quả, vị trí và phản ứng cứu trợ . Đây là một bài toán hồi quy (regression) , được tiếp cận bằng các phương pháp học máy khác nhau nhằm đánh giá khả năng dự đoán và so sánh hiệu quả từng mô hình.

2. Mục tiêu đề tài:

2.1. Mục tiêu tổng quát:

Xây dựng và đánh giá các mô hình học máy nhằm dự đoán mức độ nghiêm trọng của thảm họa (Severity Index) và hiệu quả ứng phó (Response Efficiency Score) dựa trên bộ dữ liệu thực tế Global Disaster Response Dataset từ năm 2018–2024, thu nhập từ Kaggle bao gồm khoảng 50.000 bản ghi.

2.2. Mục tiêu cụ thể:

- Khám phá và xử lý dữ liệu thảm họa toàn cầu:

    + Phân tích các đặc điểm của dataset từ Kaggle, bao gồm các loại thảm họa, địa điểm, thời gian xảy ra và mức độ thiệt hại.

- Mô hình học máy áp dụng để dự báo thảm họa:

+ **Random Forest Regression**: để dự báo Severity Index, phản ánh mức độ nghiêm trọng của thảm họa.

+ **Linear Regression** (mô hình nền) : để dự báo **Response Efficiency Score**, phản ánh mức độ hiệu quả ứng phó với thảm họa.

+ **XGBoost**: để tối ưu hóa độ chính xác dự báo và so sánh hiệu suất

- Thực hiện tuning siêu tham số và trực quan hóa mô hình (vẽ cây quyết định)

- Xây dựng chức năng dự đoán với dữ liệu người dùng nhập vào

3. Phương pháp nghiên cứu:

- Thu thập và chuẩn bị dữ liệu; Nguồn dữ liệu: Kaggle dataset về các thiên tai toàn cầu từ 2018–2024.

- Chuẩn bị mỗi trường và cấu trúc thư mục.

- Tiền xử lý dữ liệu (Preprocessing).

- Phân tích khám phá dữ liệu (EDA).

- Xây dựng và huấn luyện mô hình.

- Tối ưu siêu tham số.

- Phân tích kết quả: So sánh hiệu suất các mô hình dựa trên các chỉ số hồi quy: RMSE, MAE và R^2.

- Phân thích tầm quan trọng của các đặc trưng.

- Dự đoán với dữ liệu nhập từ người dùng.

Toàn bộ pipeline được tổ chức theo hướng module hóa, giúp mã nguồn rõ ràng , dễ mở rộng và tái sử dụng.

Chương II: Mô tả dữ liệu, xử lý và xây dựng hệ thống :

1. Mô tả dữ liệu

- Dữ liệu được sử dụng là "Global disaster" từ Kaggle :

https://www.kaggle.com/datasets/ethicalstar/global-disaster-2018-to-2024

- Nguồn dữ liệu: Kaggle, tổng hợp các thảm họa toàn cầu từ năm 2018 đến 2024.

- Mục đích: Hỗ trợ nghiên cứu, phân tích và dự báo thảm họa toàn cầu dựa trên dữ liệu lịch sử.

1.1. Các biến chính trong dataset:

Data gồm các cột sau:

Cột	Kiểu dữ liệu	Mô tả

Date	Date	Ngày xảy ra thảm họa
Country	Categorical	Quốc gia xảy ra thảm họa
Disaster_type	Categorical	Loại thảm họa
Severity_index	Float	Chỉ số đánh giá mức độ nghiêm trọng của thảm họa
Causualities	Integer	Số người chết và bị thương
Economic_loss_usd	Float	Thiệt hại kinh tế tính bằng USD
Response_time_hours	Float	Thời gian phản ứng cứu trợ (giờ)
Aid_amount_usd	Float	Số tiền viện trợ được cung cấp (USD)
Response_efficiency_score	Float	Chỉ số đánh giá hiệu quả phản ứng cứu trợ
Recovery_days	Integer	Số ngày phục hồi sau thảm họa
Latitude	Float	Vĩ độ nơi xảy ra thảm họa
Longitude	Float	Kinh độ nơi xảy ra thảm họa

1.2 Đặc điểm dữ liệu:

- Bao gồm cả biến phân loại (country, disaster_type) và biến số (severity_index, casualties, economic_loss_usd, ...).

- Các cột liên quan đến cứu trợ (response_time_hours, aid_amount_usd, response_efficiency_score) giúp đánh giá hiệu quả ứng phó thảm họa.

- Các cột vị trí (latitude, longitude) có thể được sử dụng cho trực quan hóa hoặc phân tích địa lý.

- Severity_index (biến mục tiêu).

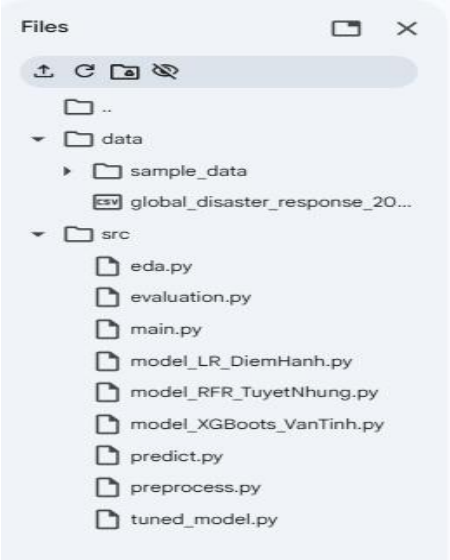
- Response_efficiency_score (biến mục tiêu phụ).

2. Chuẩn bị môi trường và cấu trúc thư mục

Hệ thống được tổ chức theo cấu trúc:

- src/ : chứa toàn bộ các module xử lý và đánh giá.

- data/ : chứa dữ liệu đầu vào.



Việc tách riêng thư mục giúp:

- Dễ quản lý mã nguồn.

- Thuận tiện khi triển khai và bảo trì.

3. Tiền xử lí :

3.1. Mục tiêu tiền xử lý:

Tiền xử lý dữ liệu là bước quan trọng nhằm:

- Làm sạch dữ liệu.

- Ngăn chặn rò rỉ dữ liệu (data leakage).

- Chuẩn hóa dữ liệu để phù hợp với các thuật toán học máy.

- Đảm bảo mô hình học được các đặc trưng có ý nghĩa.

3.2. Các bước tiền xử lý chi tiết

- Bước 1: Tải dữ liệu

Dữ liệu được đọc từ file CSV bằng thư viện pandas. Đây là định dạng phổ biến trong các bài toán phân tích dữ liệu.

```
# ===== 1. LOAD DATA =====
df = pd.read_csv(path)
```

- Bước 2: Xử lý dữ liệu ngày tháng

Cột date được chuyển sang kiểu datetime, sau đó tách thành:

+ year : năm xảy ra thảm họa.

+ month : tháng xảy ra thảm họa.

Cột date gốc được loại bỏ vì không cần thiết.

```
# ===== 2. DATE FEATURES =====
df["date"] = pd.to_datetime(df["date"], errors="coerce")
df["year"] = df["date"].dt.year
df["month"] = df["date"].dt.month
df.drop(columns=["date"], inplace=True)
```

-> Việc tách năm và tháng giúp mô hình học được yếu tố thời gian mà không phải xử lý trực tiếp dữ liệu dạng chuỗi.

- Bước 3: Tách biến mục tiêu (Target variables)

- Hệ thống xử dụng 2 biến mục tiêu:

+ severity_index: mức độ nghiêm trọng của thảm họa.

+ response_time_hours: thời gian phản hồi cứu trợ.

- Hai biến này được:

+ Ép kiểu số (numeric).

```
# ===== 3. TARGETS =====
y_raw = pd.to_numeric(df["severity_index"], errors="coerce")
y_eff_raw = pd.to_numeric(df["response_efficiency_score"], errors="coerce")

X_raw = df.drop(
    columns=["severity_index", "response_efficiency_score"]
)
```

+ Loại bỏ các giá trị không hợp lệ (NaN).

```
# ===== 4. DROP NaN  ===== #
temp_df = pd.concat(
    [
        X_raw,
        y_raw_df, # Use DataFrame here
        y_eff_raw_df, # Use DataFrame here
    ],
    axis=1,
)

temp_df.dropna(
    subset=["severity_index_target", "response_efficiency_target"],
    inplace=True,
)

temp_df = temp_df.reset_index(drop=True)
```

+ Tách hoàn toàn khỏi tập đặc trưng đầu vào để tránh rò rỉ dữ liệu.

```
# TÁCH LẠI
y = temp_df["severity_index_target"]
y_efficiency = temp_df["response_efficiency_target"]
X = temp_df.drop(
    columns=["severity_index_target", "response_efficiency_target"]
)
```

- Bước 4: Chuẩn hóa biến mục tiêu.

Cả hai biến mục tiêu được chuẩn hóa về khoảng [0, 1] bằng MinMaxScaler.

- Giúp mô hình học ổn định hơn.

- Cho phép so sánh và phân loại mức độ (LOW - MEDIUM - HIGH).

- Thuận tiện cho việc diễn giải kết quả.

```
# ===== 5. SCALE TARGET (0-1) =====
y_scaler = MinMaxScaler()
y_eff_scaler = MinMaxScaler()

y = pd.Series(
    y_scaler.fit_transform(y.to_frame()).flatten(),
    index=y.index,
)

y_efficiency = pd.Series(
    y_eff_scaler.fit_transform(y_efficiency.to_frame()).flatten(),
    index=y_efficiency.index,
)
```

Scaler của từng biến mục tiêu được lưu lại để đảo ngược chuẩn hóa khi dự đoán.

- Bước 5: Loại bỏ rò rỉ dữ liệu (Data Leakage)

Cột recovery_days bị loại bỏ khỏi tập đặc trưng vì:

- Biến này phản ánh hậu quả sau thảm họa.

- Không thể sử dụng khi dự đoán ở thời điểm ban đầu.

- Nếu giữ lại sẽ làm mô hình "gian lận" kết quả.

```
# ===== 6. REMOVE TARGET LEAKAGE =====
if "recovery_days" in X.columns:
    X = X.drop(columns=["recovery_days"])
```

- Bước 6: Xử lý biến phân loại và biến số

- Biến phân loại (object):

+ Điền giá trị thiếu bằng “Unknown”.

+ Mã hóa bằng One-Hot Encoding.

+ Loại bỏ cột đầu tiên để tránh đa cộng tuyến.

```
# ===== 8. CATEGORICAL =====
X_cat = pd.get_dummies(
    X[cat_cols].fillna("Unknown"),
    drop_first=True,
)
```

- Biến số:

+ Ép kiểu số.

+ Điền giá trị thiếu bằng trung vị (median).

+ Chuẩn hóa bằng StandardScaler.

```
# ===== 9. NUMERICAL =====
for col in num_cols:
    X[col] = pd.to_numeric(X[col], errors="coerce")

X[num_cols] = X[num_cols].fillna(X[num_cols].median())

X_scaler = StandardScaler()
X_num = X_scaler.fit_transform(X[num_cols])

X_num = pd.DataFrame(
    X_num,
    columns=num_cols,
    index=X.index,
)
```

- Bước 7: Hợp nhất dữ liệu

Cuối cùng:

- Các biến số đã chuẩn hóa.

- Các biến phân loại đã mã hóa.

Được hợp nhất thành tập dữ liệu đầu vào hoàn chỉnh X_processed.

3.3. Kết quả đầu ra của tiền xử lý

Hàm tiền xử lý trả về:

- X_processed: tập đặc trưng đã xử lý.

- y_targets: dictionary chứa hai biến mục tiêu.

- X_scaler: scaler của dữ liệu đầu vào.

- y_scalers: scaler của các biến mục tiêu.

- feature_names: danh sách tên đặc trưng cuối cùng.

- Toàn bộ dữ liệu ở dạng số, sẵn sàng cho huấn luyện mô hình.

4. Phân tích dữ liệu khám phá (EDA)

4.1. Phân bố biến mục tiêu

Biểu đồ histogram được sử dụng để.

4.2. Chuẩn hóa dữ liệu (Feature Scaling)

Biểu đồ histogram được sử dụng để:

- Quan sát phân bố severity_index

-> Phân bố không đồng đều, phần lớn tập trung ở mức thấp đến trung bình, ít mẫu có mức độ nghiêm trọng rất cao, dữ liệu có xu hướng lệch phải.

- Quan sát phân bố response_time_hours

-> Dữ liệu phân bố trải rộng, có sự chênh lệch rõ rệt giữa các thảm họa nhỏ, lớn. Một số điểm ngoại lai (outliers) với thời gian phải hồi rất cao.

- Phát hiện skewness và outliner

4.3. Phân tích mối tương quan giữa các biến

Heatmap tương quan được xây dựng cho các biến số nhằm:

- Xác định mối quan hệ tuyến tính giữa các đặc trưng.

- Phát hiện các biến có ảnh hưởng mạnh đến biến mục tiêu.

- casualties, economic_loss_usd, aid_amount_usd -> có tương quan dương với severity_index.

- Các biến thời gian (year, month) có tương quan thấp.

-> Mức độ nghiêm trọng của thảm họa chịu ảnh hưởng chủ yếu bởi thiệt hại và thương vong, không phụ thuộc nhiều vào yếu tố thời gian.

4.4. Phân tích các đặc trưng quan trọng

Các biến có độ tương quan cao nhất với biến mục tiêu được trực quan bằng biểu đồ scatter để:

- Quan sát xu hướng.

- Phát hiện quan hệ tuyến tính hoặc phi tuyến.

Chương III : Phương pháp/mô hình Học máy áp dụng :

1. Random Forest Regression (severity_index)

1.1 Lý do lựa chọn mô hình Random Forest

Biến mục tiêu severity_index phản ảnh mức độ nghiêm trọng của thảm họa và chịu ảnh hưởng đồng thời bởi nhiều yếu tố như:

- Số thương vòng.

- Thiệt hại kinh tế.

- Mức viện trợ.

- Yếu tố thời gian.

Qua phân tích EDA, có thể nhận thấy:

- Mối quan hệ giữa các biến không thuần tuyến tính.

- Dữ liệu tồn tại nhiễu và outlier,

- Các biến đầu vào có mức độ ảnh hưởng không đồng đều.

Do đó, mô hình Random Forest Regression được lựa chọn vì các ưu điểm sau:

- Có khả năng mô hình hóa quan hệ phi tuyến.

- Hoạt động tốt với dữ liệu nhiều chiều.

- Ít bị overfitting hơn Decision Tree đơn lẻ.

- Cung cấp thông tin tầm quan trọng của đặc trưng, giúp giải thích mô hình.

1.2 Cấu trúc mô hình Random Forest Regression

Random Forest là một mô hình ensemble learning, bao gồm nhiều cây quyết định (Decision Tree).

Mỗi cây được huấn luyện trên:

- Một tập con ngẫu nhiên của dữ liệu (bootstrap sampling).

- Một tập con ngẫu nhiên của các đặc trưng.

Kết quả dự đoán cuối cùng là trung bình dự đoán của tất cả các cây.

-> Điều này giúp:

- Giảm phương sai (variance).

- Tăng khả năng tổng quát hóa.

- Hạn chế hiện tượng học quá khớp.

1.3 Thiết lập và tối ưu siêu tham số (Hyperparameter Tuning)

1.3.1. Lý do cần tuning

Nếu sử dụng Random Forest với tham số mặc định:

- Mô hình dễ quá phức tạp.

- Thời gian huấn luyện dài.

- Nguy cơ overfitting.

Vì vậy, đồ án sử dụng GridSearchCV để tìm bộ siêu tham số tối ưu.

1.3.2. Không gian siêu tham số được tìm kiếm

Tham số	**Ý nghĩa**	**Giá trị thử nghiệm**
n_estimators	Số lượng cây trong rừng	50, 100
max_depth	Độ sâu tối đa của mỗi cây	5, 8
min_samples_split	Số mẫu tối thiểu để tách nút	5, 10
min_samples_leaf	Số mẫu tiểu thiểu tại lá	1, 2
bootstrap	Lấy mẫu có hoàn lại	True, False

-> Phạm vi tham số được giới hạn có chủ đích để:

- Giảm thời gian huấn luyện.

- Tránh mô hình phức tạp.

- Phù hợp với quy mô dữ liệu.

1.3.3. Phương pháp đánh giá trong tuning

- Cross-validation: 3-fold

- Tiêu chí đánh giá: Negative Mean Squared Error (-MSE)

- Số tổ hợp tham số thử nghiệm: 10

-> Cách tiếp cận này đảm bảo:

- Mô hình được đánh giá trên nhiều tập dữ liệu khác nhau.

- Giảm rủi ro chọn tham số theo may mắn.

1.4. Kết quả huấn luyện mô hình RandomForest

Sau quá trình RandomizedSearchCV, hệ thống lựa chọn được mô hình Random Forest tối ưu cho bài toán dự đoán severity_index.

1.4.1. Kết quả đánh giá trên tập test

Các chỉ số thu được:

```
----- Các chỉ số hồi quy -----
RMSE: 0.10450398837690295
MAE : 0.07939897991501649
R2  : 0.7655429548228418
```

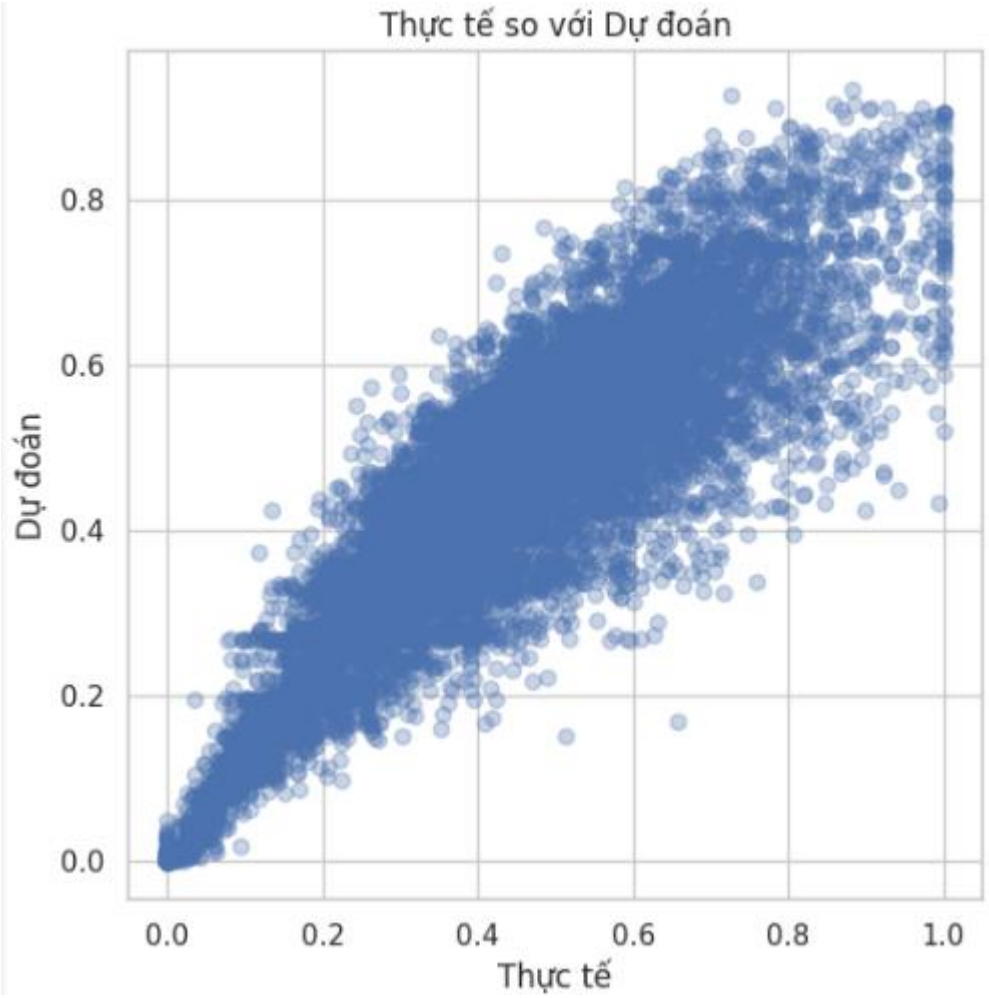
- RMSE: thấp -> Sai số dự đoán trung bình nhỏ.

- MAE: ổn định -> Dự đoán không bị lệch quá nhiều so với giá trị thực.

- R^2 Score: cao -> Mô hình giải thích được phần lớn phương sai của dữ liệu

=> Điều này cho thấy mô hình phù hợp tốt với dữ liệu thực tế.

1.4.2. Phân tích biểu đồ Actual vs Predicted



Biểu đồ so sánh giữa giá trị thực và giá trị dự đoán cho thấy:

- Các điểm dữ liệu tập trung gần đường chéo.

- Sai lệch nhỏ ở vùng giá trị thấp và trung bình.

- Sai lệch tăng nhẹ ở các giá trị rất cao (do số lượng mẫu ít).

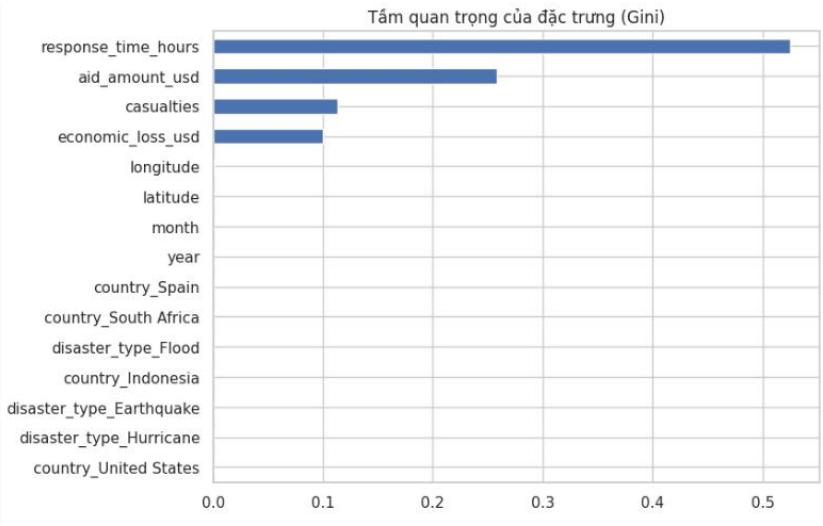
=> Nhận xét:

- Mô hình dự đoán tốt phần lớn trường hợp.

- Hiệu quả cao hơn rõ rệt so với Liner Regression.

1.5. Phân tích tầm quan trọng của các đặc trưng (Feature Importance)

1.5.1. Feature Importance theo Gini



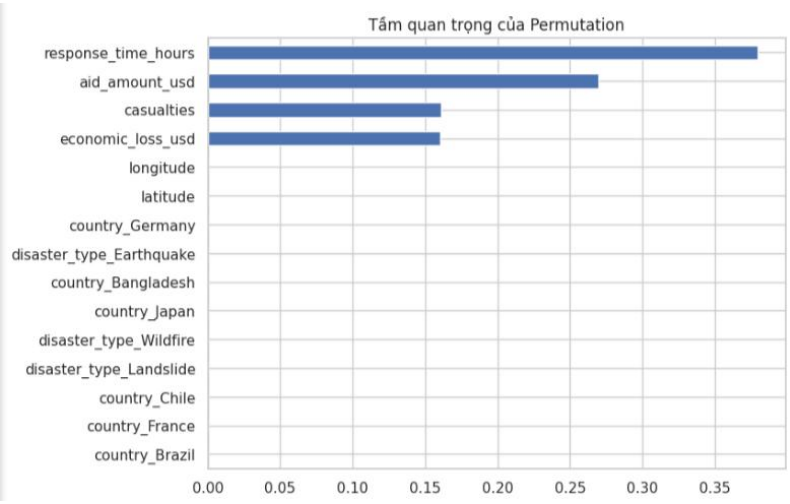
Kết quả cho thấy các đặc trưng quan trọng nhất bao gồm: econimic_loss_usd, casualties, aid_amount_usd, reponse_time_hours, year.

=> Giải thích:

- Thiệt hại kinh tế và thương vong là yếu tố quyết định trực tiếp mức độ nghiêm trọng.

- Thời gian phản hồi và mức viện trợ phản ánh quy mô và mức độ khẩn cấp của thảm họa.

1.5.2. Permutation Importance



Khi hoán vị các đặc trưng quan trọng:

- Sai số mô hình tăng đáng kể.

- Thứ tự quan trọng gần giống với Gini Importance.

=> Kết luận:

```
Top 10 đặc trưng quan trọng:
response_time_hours     0.525330
aid_amount_usd          0.258163
casualties              0.112956
economic_loss_usd       0.100004
longitude               0.000895
latitude                0.000751
month                   0.000372
year                    0.000329
country_Spain           0.000128
country_South Africa    0.000066
dtype: float64
```

- Các đặc trưng quan trọng là thực sự có ảnh hưởng.

- Không phải do mô hình học ngẫu nhiên

1.6. Trực quan hóa cây quyết định trong Random Forest

Một cây quyết định đại diện được trực quan hóa với độ sâu tối đa là 3 tầng.

Quan sát cho thấy:

- Các nút gốc thường chia theo economic_loss_usd hoặc casualties.

- Ngưỡng chia hợp lý và phù hợp với thực tế.

=> Điều này giúp :

- Tăng khả năng giải thích mô hình.

- Chứng minh mô hình không phải "hộp đen hoàn toàn".

2. Linear Models (Linear Regression & Ridge Regression)

2.1. Linear Regression

2.1.1. Nguyên lí hoạt động của mô hình Linear Regression

- Linear Regression (Hồi quy tuyến tính) là mô hình học máy thuộc nhóm supervised learning, dùng để dự đoán biến liên tục.

- Mô hình giả định mối quan hệ tuyến tính giữa biến đầu vào và biến mục tiêu theo công thức: $y= w0+w1.x1+w2.x2+...+wn.xn$

- Nguyên lý chính:

  - Tìm bộ trọng số w sao cho sai số giữa giá trị thực tế và giá trị dự đoán là nhỏ nhất.

  - Mô hình được tối ưu bằng cách tối thiểu hóa hàm mất mát Mean Squared Error (MSE).

-Trong bài toán này:

  - Linear Regression được sử dụng để dự đoán Response Efficiency Score (RES).

  -Dữ liệu đầu vào được chuẩn hóa (scaling) trước khi huấn luyện nhằm đảm bảo các đặc trưng có cùng thang đo.

2.1.2. Phân tích dữ liệu (EDA) cho Linear Regression.

-Phân bố của biến mục tiêu Response Efficiency Score được trực quan hóa bằng histogram:

  - Giúp đánh giá dữ liệu tập trung hay phân tán.

  - Kiểm tra khả năng phù hợp với mô hình tuyến tính.

- Ma trận tương quan (Correlation Heatmap) được sử dụng để:

  - Xác định mức độ ảnh hưởng của các đặc trưng số đến Response Efficiency Score.

- Phát hiện mối quan hệ tuyến tính giữa các biến đầu vào và biến mục tiêu.

- Kết quả EDA cho thấy:

- Một số đặc trưng có tương quan rõ ràng với RES.

- Dữ liệu phù hợp để áp dụng mô hình hồi quy tuyến tính.

2.1.3. Huấn luyện mô hình Linear Regression

```
lin_reg_eff = LinearRegression()
lin_reg_eff.fit(X_train_scaled, y_eff_train)
```

- Mô hình được huấn luyện trên training set đã chuẩn hóa.

- Dự đoán được thực hiện trên test set.

- Các chỉ số đánh giá:

- RMSE (Root Mean Squared Error.

- MAE (Mean Absolute Error.

- R^2 (Coefficient of Determination.

-Kết quả đánh giá cho thấy:

- Sai số dự đoán ở mức chấp nhận được.

- Giá trị R^2 tương đối cao, cho thấy mô hình giải thích tốt biến thiên của dữ liệu.

2.2. Ridge Regression (Linear Regression với Regularization)

2.2.1. Nguyên lí hoạt động của Ridge Regression

- Ridge Regression là phiên bản mở rộng của Linear Regression, bổ sung L2 regularization vào hàm mất mát: Loss=MSE+α∑w^2

- Mục đích:

    - Giảm độ lớn của các hệ số.

    - Hạn chế hiện tượng overfitting

    - Ổn định mô hình khi có khả năng xảy ra đa cộng tuyến

2.2.2 Hyperparameter tuning cho Ridge Regression

- GridSearchCV được sử dụng để tìm giá trị **alpha** tối ưu

- Các tham số được thử nghiệm:

    alpha = [0.01, 0.1, 1, 10, 100]

- Phương pháp đánh giá:

    - 5-fold Cross Validation

    - Thước đo: R^2 score

```
grid_search = GridSearchCV(
    estimator=ridge,
    param_grid=param_grid,
    cv=5,
    scoring='r2',
    n_jobs=-1
)
```

- Sau tuning, mô hình Ridge tốt nhất được chọn để dự đoán trên tập test

2.2.3. So sánh hiệu năng Linear Regression và Ridge Regression

Kết quả cho thấy Linear Regression và Ridge Regression cho hiệu năng gần như tương đương. Điều này cho thấy dữ liệu không gặp hiện tượng overfitting nghiêm trọng và các đặc trưng không có đa cộng tuyến mạnh. Do đó, việc thêm regularization trong Ridge Regression không mang lại cải thiện rõ rệt so với Linear Regression.

2.3 Trực quan hóa – Actual vs Predicted

2.3.1. Linear Regression

- Biểu đồ scatter plot so sánh giá trị RES thực tế và dự đoán:

    - Đường y = x biểu diễn dự đoán hoàn hảo.

    - Các điểm dữ liệu phân bố gần đường y = x cho thấy mô hình dự đoán tốt.

2.3.2. Ridge Regression

- Biểu đồ Actual vs Predicted sau tuning có phân bố gần tương tự Linear Regression.

- Điều này xác nhận lại rằng:

    - Ridge Regression không cải thiện đáng kể so với mô hình gốc.

3. XGBoost

3.1. Nguyên lí hoạt động của mô hình XGBoost

- XGBoost (Extreme Gradient Boosting) là một mô hình học máy thuộc nhóm ensemble learning, sử dụng nhiều decision tree được huấn luyện tuần tự theo nguyên lý gradient boosting.

- Nguyên lý chính:

+ Mỗi cây quyết định mới được xây dựng để sửa lỗi (residual) của các cây trước đó

+ Mô hình tối ưu một hàm mất mát (loss function) bằng gradient descent

+ Kết quả cuối cùng là tổng có trọng số của tất cả các cây

- Trong bài toán này, XGBoost được sử dụng ở dạng regression với hàm mất mát:

+ Mean Squared Error (reg:squarederror).

3.2. Lí do lựa chọn mô hình XGBoost

XGBoost được lựa chọn vì các ưu điểm sau:

- Hoạt động rất tốt với dữ liệu phi tuyến

- Xử lý hiệu quả mối quan hệ phức tạp giữa các biến

- Có khả năng giảm overfitting nhờ:

+ Learning rate

+ Subsampling

+ Regularization

3.3. Train mô hình XGBoost

```
xgb_sev = XGBRegressor(
    objective='reg:squarederror',
    random_state=42,
    tree_method='hist',
    n_jobs=-1
)
```

- Mô hình được huấn luyện trên tập training set và đánh giá trên test set.

3.4. Hyperparameter tuning

- Để nâng cao hiệu quả mô hình, GridSearchCV được sử dụng để tìm bộ tham số tối ưu.

- Các tham số được tối ưu:

+ n_estimators

+ max-depth

+ learning_rate

+ subsample

+ colsample_bytree

```
gs = GridSearchCV(
    xgb_sev,
    param_grid,
    scoring='neg_root_mean_squared_error',
    cv=KFold(n_splits=5),
    n_jobs=-1
)
```

- Mô hình được đánh giá bằng RMSE trung bình trên 5-fold cross-validation.

3.5. Trực quan hóa – Actual vs Predicted

- Biểu đồ Actual vs Predicted được sử dụng để trực quan hóa chất lượng dự đoán:

+ Các điểm nằm gần đường y = x → mô hình dự đoán tốt

+ Phân tán xa → sai số lớn

- Kết quả cho thấy mô hình XGBoost dự đoán Severity Index khá chính xác.

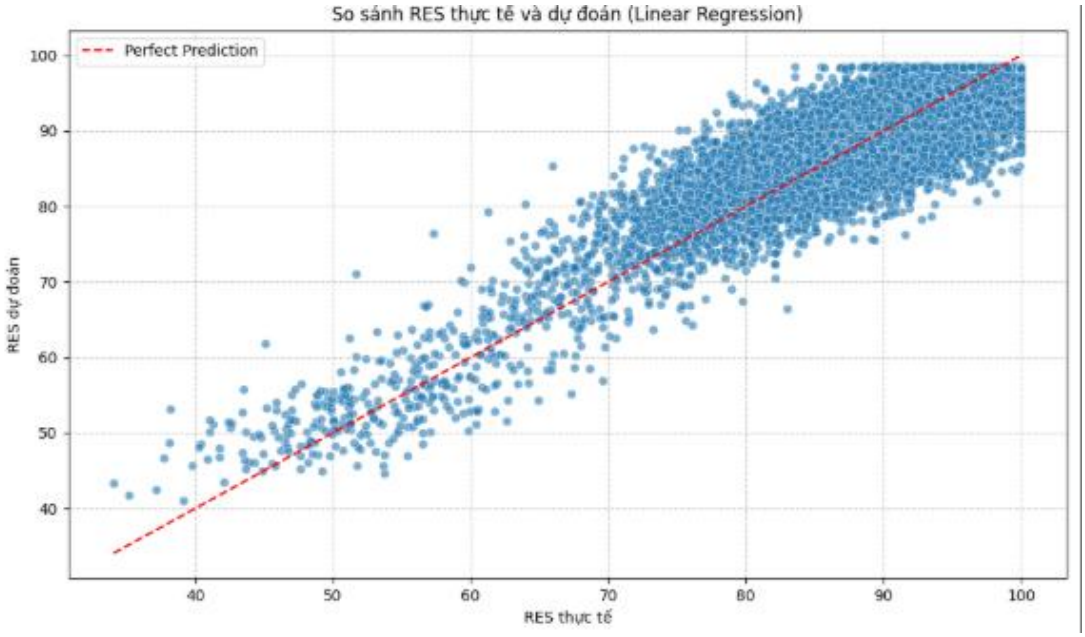
Chương IV : Kết quả bước đầu và nhận xét

1. Kết quả bước đầu:

- Linear Regression:

+ Mô hình Linear Regression được huấn luyện để dự đoán Response Efficiency Score (RES) đạt các kết quả sau:
RMSE = 4.6684
MAE = 3.7426
R^2 = 0.7829

-Đánh giá mô hình:



+ Biểu đồ Actual vs Predicted cho thấy:

- Các điểm dữ liệu phân bố tương đối gần đường y = x.

- Sai số dự đoán ở mức chấp nhận được và khá ổn định.

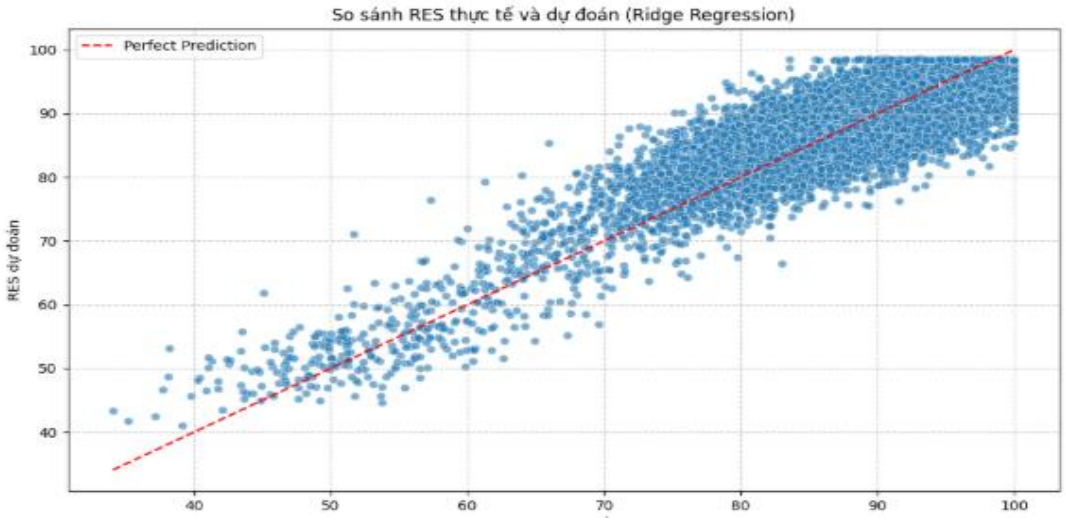
- Mô hình học được xu hướng tuyến tính giữa các biến đầu vào và Response Efficiency Score.

-Ridge Regression:

+ Mô hình Ridge Regression (Linear Regression có regularization) sau khi tuning tham số alpha đạt các kết quả:
RMSE = 4.6684

MAE = 3.7426
R^2 = 0.7829

- Đánh giá mô hình:



+ Biểu đồ Actual vs Predicted cho thấy:

- Phân bố điểm dữ liệu tương tự Linear Regression.

- Không có sự cải thiện rõ rệt về độ chính xác so với mô hình Linear ban đầu.

- Điều này cho thấy dữ liệu không gặp hiện tượng overfitting nghiêm trọng.

- XGBoots:

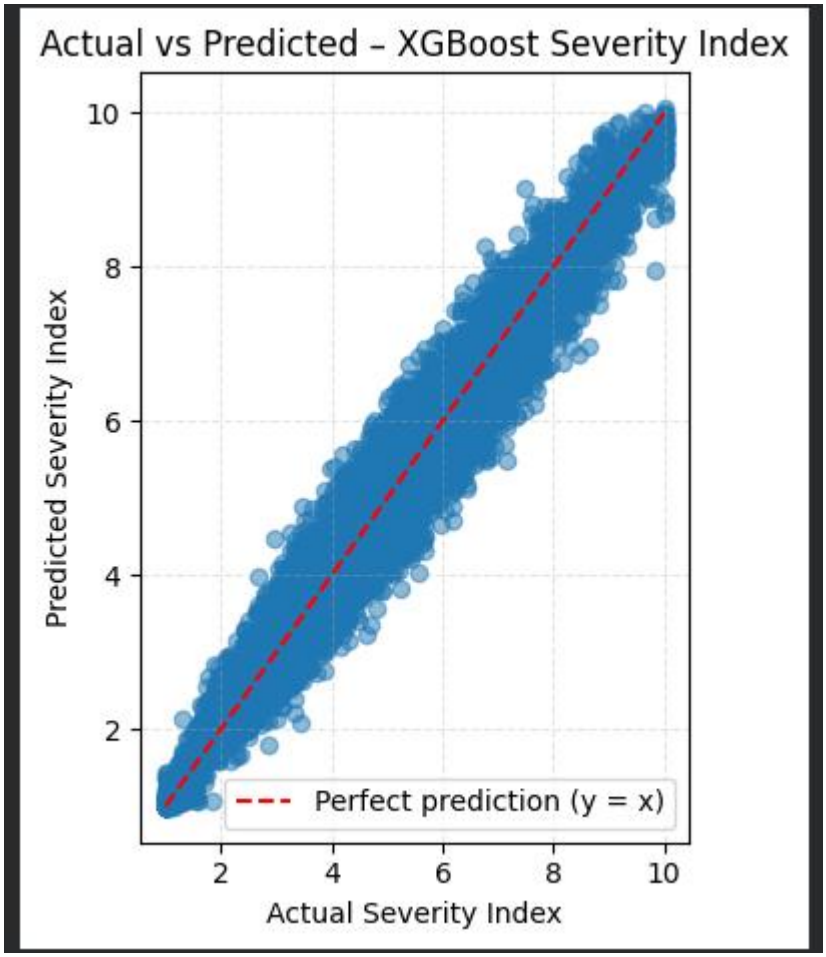
+ Mô hình XGBoost Regression sau khi được huấn luyện và tối ưu siêu tham số cho bài toán dự đoán Severity Index đạt được các kết quả sau:

RMSE = 0.4175

MAE = 0.3273

R^2 = 0.9538

- Đánh giá mô hình:



- Biểu đồ Actual vs Predicted cho thấy:

+Các điểm dữ liệu tập trung gần đường $y = x$.

+ Sai số nhỏ và ổn định.

+ Mô hình học tốt mối quan hệ giữa các biến đầu vào.

2. Nhận xét:

- Ưu và Nhược Điểm:

Mô hình	Ưu điểm	Nhược Điểm
Random Forest	-Xử lý tốt các mối quan hệ phi tuyến giữa các biến. - Giảm nguy cơ overfitting	- Mô hình phức tạp, khó giải thích chi tiết so với Logistic Regression.

	nhờ ensemble các cây quyết định. - Hoạt động tốt với dữ liệu lớn và đa chiều.	- Thời gian huấn luyện và dự đoán lâu hơn đối với tập dữ liệu rất lớn.
Linear Regression, Ridge Regression	- Mô hình đơn giản, dễ triển khai và dễ diễn giải. - Phù hợp làm mô hình nền (baseline) cho bài toán hồi quy. - Ridge Regression giúp giảm nguy cơ overfitting khi dữ liệu có đa cộng tuyến.	- Khả năng mô hình hóa mối quan hệ phi tuyến còn hạn chế. - Hiệu suất không cao bằng các mô hình boosting khi dữ liệu phức tạp.
XGBoots	- Cải thiện hiệu suất bằng boosting, học từ lỗi của các cây trước đó. - Hiệu quả cao trên dữ liệu phi tuyến, đa lớp và nhiều biến số. - Khả năng kiểm soát overfitting tốt - Hỗ trợ đánh giá feature importance và trực quan hóa.	- Thời gian huấn luyện dài, đặc biệt với dataset lớn. - Mô hình khá phức tạp

Chương V: Phân công nhiệm vụ từng thành viên:

STT	Họ và Tên	Công việc	Tiến độ thực hiện
1	Nguyễn Thị Tuyết Nhung	- Tìm hiểu và phân tích dữ liệu. - Sắp xếp logic và tiền xử lý dữ liệu 60%. - Xây dựng và viết báo báo cáo cho baseline model Random Forest. - Chỉnh sửa Word, viết chương II, III, IV, VI.	100%
2	Lê Văn Tính	- Tìm hiểu và phân tích dữ liệu. - Import thư viện, tải và khám phá dữ liệu. - Tiền xử lý dữ liệu 40%. - Tìm hiểu, xây dựng và viết báo cáo cho mô hình XGBoots. - Viết Word phần mở đầu, chương I với mô tả dữ liệu, chương III, IV, VI. - Thử nghiệm các kỹ thuật xử lý mất cân	100%

		bằng khác (định hướng lần tới)	
3	Trần Nguyễn Diễm Hạnh	- Tìm hiểu và phân tích dữ liệu. - Tìm hiểu, xây dựng và viết báo cáo cho mô hình Linear Regression. - Viết Word chương III. - Sử dụng thêm các chỉ số để tối ưu đánh giá mô hình.	100%

Chương VI: Các tài liệu tham khảo

- Dataset Global disaster Kaggle: https://www.kaggle.com/datasets/ethicalstar/global-disaster-2018-to-2024

- Mô hình XGBoots:

https://www.geeksforgeeks.org/machine-learning/xgboost/

- Hướng đi của đồ án:

https://www.studocu.vn/vn/document/truong-dai-hoc-da-lat/tai-lieu-ve-dong-y/ung-dung-hoc-may-trong-du-bao-nguy-co-thien-tai-final-exam/133012233