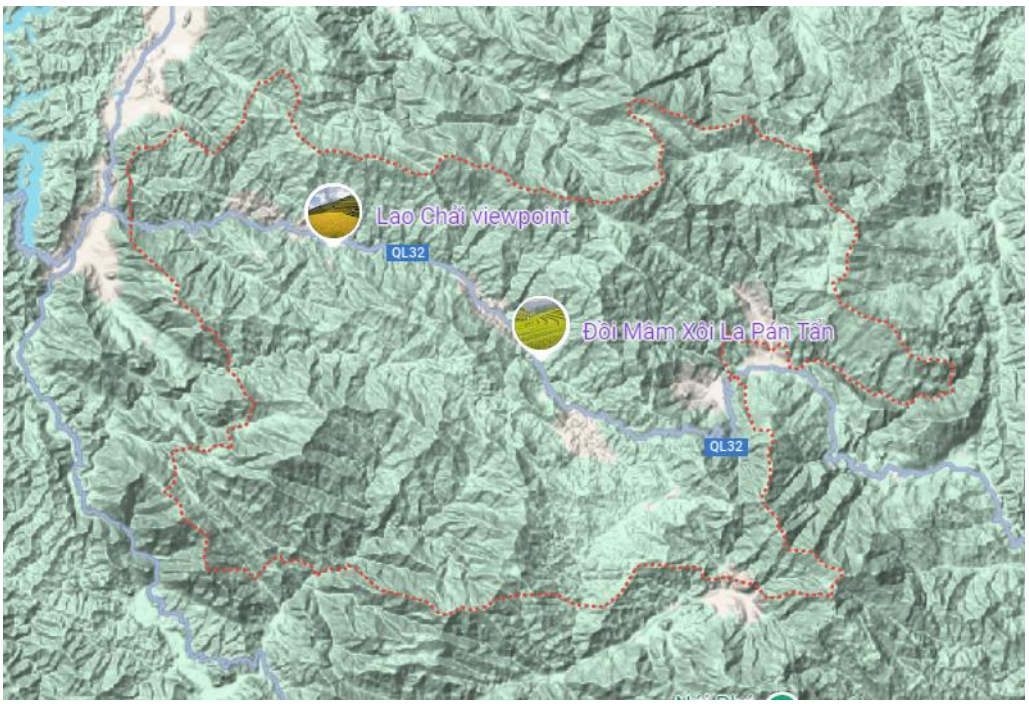
21002152
Nguyễn Hữu Trung Kiên

Bài tập số 2: Phân tích nguy cơ trượt lở (output=1) bằng các mô hình thống kê và học máy

Yên Bái là một tỉnh miền núi phía Bắc, với địa hình có độ dốc lớn, cao dần từ Đông Nam lên Tây Bắc. Địa hình khá phức tạp nhưng có thể chia thành 2 vùng lớn: vùng cao và vùng thấp. Vùng cao có độ cao trung bình 600m trở lên, chiếm 67,56% diện tích toàn tỉnh. Vùng thấp có độ cao dưới 600m, chủ yếu là địa hình đồi núi thấp, thung lũng bồn địa, chiếm 32,44% diện tích tự nhiên toàn tỉnh.

Đặc điểm địa chất phong hóa mạnh, các hoạt động nhân sinh, đặc biệt là lượng mưa trung bình các tháng mùa mưa lớn. Phân bố lượng mưa theo xu hướng tăng dần từ vùng thấp đến vùng cao đã tạo điều kiện cho trượt lở đất xảy ra. Khu vực có nguy cơ trượt lở cao là vùng đất cao, chủ yếu tập trung tại một số xã của các huyện: Mù Cang Chải, Trạm Tấu, Lục Yên, Văn Chấn. Do đó, để giảm thiểu thiệt hại do thiên tai nói chung, tai biến trượt lở đất nói riêng cần thiết phải có những giải pháp ứng phó đồng bộ, thống nhất trên cơ sở các nghiên cứu mang tính tổng hợp, kết hợp với các mô hình trực quan và học máy.



Trong bài tập lần này, em sẽ thực hiện phân tích sơ bộ về đặc điểm các yếu tố ảnh hưởng đến trượt lở tại Yên Bái, kết hợp với mô hình học máy **Random Forest.**

21002152
Nguyễn Hữu Trung Kiên

A. Phân tích dữ liệu

Cả 2 tập dữ liệu `CoTruot` và `KhongTruot`:

- Đều có 13 trường dữ liệu bao gồm: pointid, ma_cu, ma_moi, slope, elevation, thachhoc, dis_dutgay, dis_thachhoc, dis_river, dis_mainroad, dis_road, dis_dancu và output.
- Không có dữ liệu bị thiếu:

Tập `CoTruot`

```
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   pointid       2089 non-null   int64
 1   ma_cu         2089 non-null   int64
 2   ma_moi        2089 non-null   int64
 3   slope         2089 non-null   int64
 4   elevation     2089 non-null   int64
 5   thachhoc      2089 non-null   int64
 6   dis_dutgay    2089 non-null   int64
 7   dis_thachhoc  2089 non-null   int64
 8   dis_river     2089 non-null   int64
 9   dis_mainroad  2089 non-null   int64
 10  dis_road      2089 non-null   int64
 11  dis_dancu     2089 non-null   int64
 12  output        2089 non-null   int64
```

Tập `KhongTruot`

```
Data columns (total 13 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   pointid       240817 non-null  int64
 1   ma_cu         240817 non-null  int64
 2   ma_moi        240817 non-null  int64
 3   slope         240817 non-null  int64
 4   elevation     240817 non-null  int64
 5   thachhoc      240817 non-null  int64
 6   dis_dutgay    240817 non-null  int64
 7   dis_thachhoc  240817 non-null  int64
 8   dis_river     240817 non-null  int64
 9   dis_mainroad  240817 non-null  int64
 10  dis_road      240817 non-null  int64
 11  dis_dancu     240817 non-null  int64
 12  output        240817 non-null  int64
```

- Tuy nhiên, về số lượng bản ghi, tập `CoTruot` có 2089 bản ghi còn tập `KhongTruot` lại có tới 240817 bản ghi. Có thể nhận thấy sự chệnh lệch rất lớn giữa 2 tập dữ liệu, cần phải tiến hành tiền xử lý trước khi đưa vào mô hình học máy vì sự chênh lệch này có thể gây ra:
  - **Thiên lệch trong dự đoán**: Mô hình có thể thiên về dự đoán lớp 0 (lớp chiếm ưu thế) vì đó là cách dễ nhất để đạt được độ chính xác cao. Điều này làm giảm độ nhạy (recall) đối với lớp thiểu số (lớp 1), khiến mô hình khó nhận diện các trường hợp quan trọng của lớp này.
  - **Độ chính xác cao nhưng không hữu ích**: Khi lớp 0 chiếm đa số, độ chính xác có thể cao nhưng không phản ánh khả năng thực sự của mô hình trong việc dự đoán chính xác cả hai lớp. Ví dụ, nếu có 95% điểm dữ liệu là 0, mô hình có thể đạt độ chính xác 95% chỉ bằng cách luôn dự đoán lớp 0, nhưng khả năng phát hiện lớp 1 lại rất kém.
  - **Giảm khả năng tổng quát hóa**: Mô hình có thể khó học được các đặc điểm của lớp thiểu số (lớp 1), làm giảm khả năng dự đoán chính xác khi gặp các trường hợp mới trong lớp này.

Có một số cách tiếp cận để xử lý vấn đề mất cân bằng dữ liệu:

- **Tăng cường lớp thiểu số (Over-sampling)**: Nhân bản hoặc tạo thêm điểm dữ liệu cho lớp thiểu số (như phương pháp SMOTE - Synthetic Minority Over-sampling Technique) để cân bằng lại dữ liệu.
- **Giảm bớt lớp chiếm ưu thế (Under-sampling)**: Giảm số lượng điểm dữ liệu trong lớp chiếm ưu thế để đạt sự cân bằng với lớp thiểu số. Tuy nhiên, cách này có thể dẫn đến mất mát thông tin nếu dữ liệu ban đầu đã ít.
- **Điều chỉnh trọng số trong mô hình**: Trong một số thuật toán (ví dụ: Logistic Regression, SVM, Random Forest), có thể đặt trọng số cao hơn cho lớp thiểu số để mô hình ưu tiên phân loại đúng lớp này hơn.

Ở đây, em lựa chọn phương pháp chọn mẫu ngẫu nhiên để giảm bớt lớp chiếm ưu thế. Tuy nhiên, vẫn phải đảm bảo rằng, tập lấy mẫu giữ được những đặc trưng của tập ban đầu. Em áp dụng 2 kỹ thuật kiểm tra tính phân phối của dữ liệu:

- Kiểm tra thống kê: Phép kiểm định Kolmogorov-Smirnov (KS Test)
- Biểu đồ tương quan: Vẽ biểu đồ Heatmap

21002152
Nguyễn Hữu Trung Kiên

1. Phép kiểm định Kolmogorov-Smirnov (KS Test)

Phép kiểm định Kolmogorov-Smirnov (KS Test) là một kiểm định không tham số được sử dụng để so sánh hai tập dữ liệu nhằm xác định xem chúng có cùng phân phối hay không. KS Test có hai dạng chính:

- **One-sample KS Test**: Dùng để kiểm tra xem một mẫu có tuân theo một phân phối cụ thể không (ví dụ: phân phối chuẩn).
- **Two-sample KS Test**: Dùng để so sánh phân phối của hai mẫu độc lập xem chúng có đến từ cùng một phân phối không.

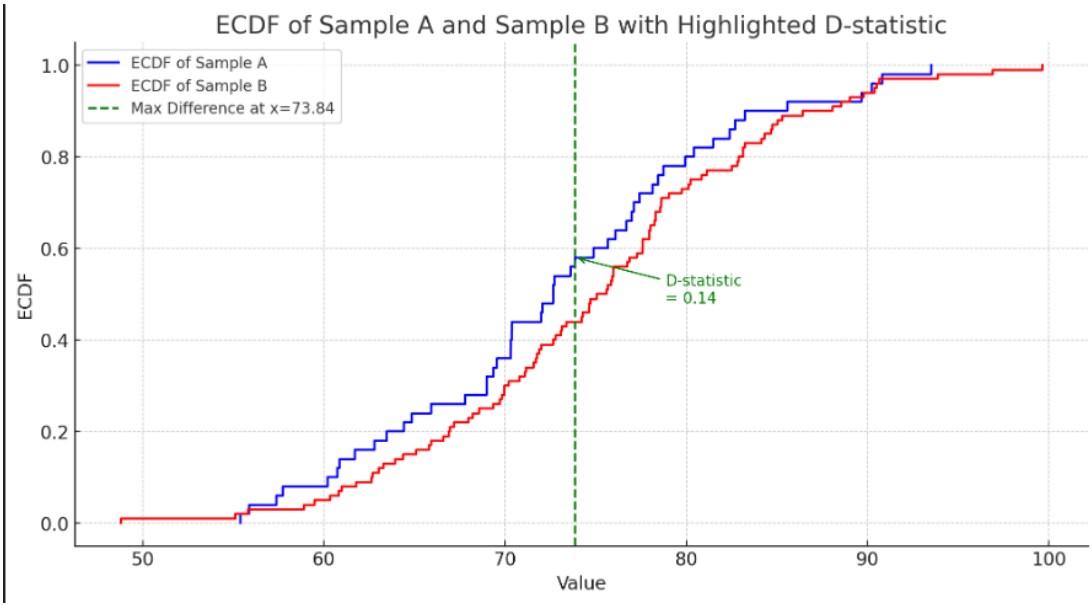
Cách thức hoạt động của Two-sample KS Test:

- Kiểm định sẽ tính toán khoảng cách lớn nhất (gọi là **D-statistic**) giữa các hàm phân phối tích lũy thực nghiệm (ECDF) của hai tập dữ liệu.
- Giá trị **D statistic** càng lớn cho thấy sự khác biệt càng lớn giữa hai phân phối.
- **p-value** được sử dụng để quyết định liệu sự khác biệt đó có ý nghĩa thống kê không. Nếu p-value nhỏ (thường nhỏ hơn 0.05), có thể kết luận rằng hai mẫu có phân phối khác nhau với mức ý nghĩa đã chọn.

D-statistic (KS statistic):

$$\mathrm{D} = \sup_x |F_A(x) - F_B(x)|$$

Trong đó $F_A(x)$ và $F_B(x)$ là các hàm phân phối tích lũy thực nghiệm của mẫu A và mẫu B.



21002152
Nguyễn Hữu Trung Kiên

p-value:

- **p-value** là một giá trị cho biết khả năng quan sát được dữ liệu nếu giả thuyết không (null hypothesis) là đúng.
  - **Giả thuyết không (H0)**: Hai mẫu dữ liệu có cùng phân phối.
  - **Giả thuyết đối (H1)**: Hai mẫu dữ liệu có phân phối khác nhau.
- Cách tính:

  Với mẫu A kích thước n_1 và mẫu B kích thước n_2:

$$N = \frac{n_1 * n_2}{n_1 + n_2}$$

$$p \approx Q(\lambda) = Q(\sqrt{N} * D)$$

$$Q(\lambda) = 2\sum_{k=1}^{\infty}(-1)^{k-1}\exp(-2k^2\lambda^2)$$

Kết quả của quá trình đánh giá:

Thuộc tính	D-statistic	p-value	p>0.05
Slope	0.008217917014119752	0.942131219417748	T
Elevation	0.004591242951111241	0.9999930254285724	T
Thachhoc	0.010257869118841856	0.7764220399307524	T
Dis_dutgay	0.006606111901504397	0.9934496214961532	T
Dis_thachhoc	0.0023224971356831103	1.0	T
Dis_river	0.017308333616579352	0.1686785687954392	T
Dis_mainroad	0.004276598797742448	0.9999990991104182	T
Dis_road	0.014454080964009242	0.3540902859178485	T
Dis_dancu	0.004192383327099591	0.9999995194934339	T

Kết luận:

Các thuộc tính của tập lấy mẫu vẫn giữ được những đặc trưng của tập ban đầu. Tuy nhiên, KS Test chỉ đánh giá trên các thuộc tính riêng lẻ, mà không kiểm tra phân phối kết hợp của nhiều thuộc tính cùng lúc. Điều này có nghĩa là ngay cả khi từng thuộc tính riêng biệt có phân phối giống nhau, phân phối tổng thể (gồm các tương quan và phụ thuộc giữa các thuộc tính) vẫn có thể khác.

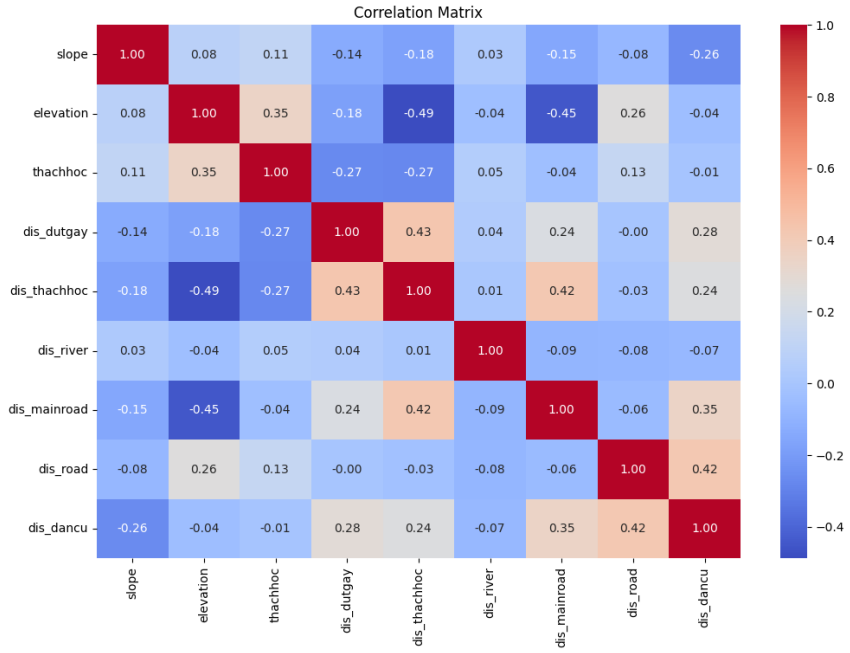
Để bù đắp vào hạn chế của KS Test, em sử dụng biểu đồ heatmap để so sánh phân phối kết hợp của 2 tập dữ liệu. Khi so sánh heatmap của hai tập, nếu các giá trị tương quan (các ô màu sắc) gần giống nhau, điều này cho thấy tập lấy mẫu đã giữ được đặc trưng tổng thể của tập ban đầu.

21002152
Nguyễn Hữu Trung Kiên

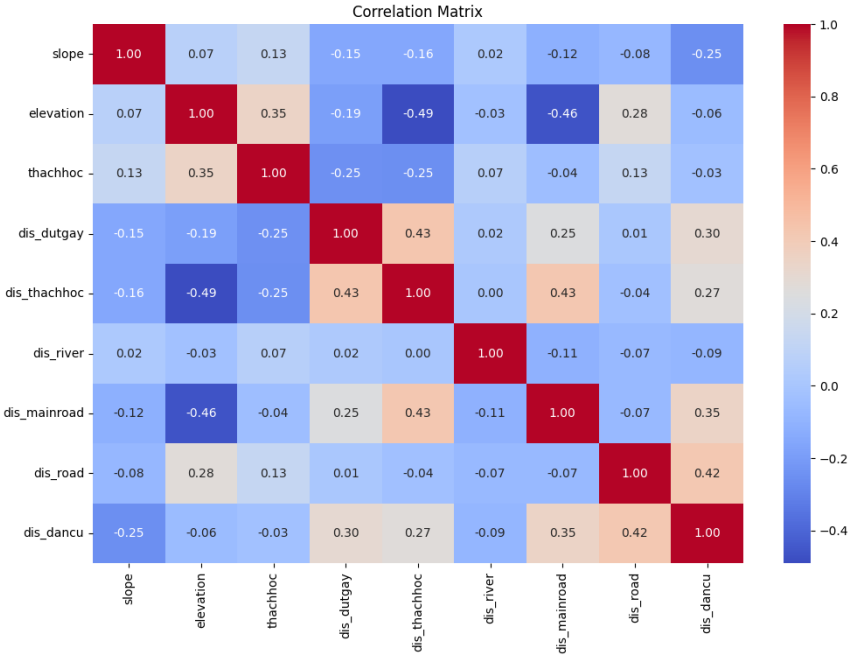
2. Biểu đồ tương quan heatmap

Tiến hành vẽ biểu đồ tương quan của 2 tập dữ liệu và so sánh:

- Tập dữ liệu ban đầu:



- Tập dữ liệu lấy mẫu:



Tương quan giữa các thuộc tính của 2 tập là tương tự nhau.

Tập lấy mẫu có thể được sử dụng thay thế cho tập ban đầu.

3. Giá trị ngoại lai

Mô hình **Random Forest** thường không bị ảnh hưởng quá nhiều bởi ngoại lai, vì một số lý do:

- **Khả năng tự điều chỉnh**: Random Forest là một mô hình **ensemble** (tập hợp) các cây quyết định (decision trees). Các cây quyết định trong Random Forest có thể xử lý khá tốt các giá trị ngoại lai vì mỗi cây chỉ phân chia dữ liệu theo các đặc trưng riêng lẻ, và các ngoại lai thường không đủ mạnh để làm thay đổi phân chia của các cây quyết định.
- **Chọn mẫu ngẫu nhiên**: Trong Random Forest, mỗi cây được huấn luyện trên một mẫu ngẫu nhiên (bootstrap sample) của dữ liệu. Điều này làm giảm khả năng ngoại lai ảnh hưởng đến mô hình vì mỗi cây sẽ chỉ học từ một phần nhỏ dữ liệu.
- **Ổn định qua nhiều cây**: Do Random Forest sử dụng nhiều cây quyết định (cụ thể sử dụng 200 cây, max_depth là 5), tác động của các ngoại lai được giảm thiểu khi kết quả của tất cả các cây được kết hợp lại. Các ngoại lai có thể ảnh hưởng đến một vài cây, nhưng chúng sẽ không đủ mạnh để làm sai lệch kết quả chung.

B. Mô hình học máy Random Forest

```
Accuracy: 0.89792663476874
Confusion Matrix:
 [[1200   67]
 [ 125  489]]
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.95      0.93      1267
           1       0.88      0.80      0.84       614

    accuracy                           0.90      1881
   macro avg       0.89      0.87      0.88      1881
weighted avg       0.90      0.90      0.90      1881
```

21002152
Nguyễn Hữu Trung Kiên

C. Phương pháp AHP

Gán giá trị số cho những so sánh chủ quan về mức độ quan trọng của từng yếu tố ảnh hưởng dựa theo bảng phân loại mức độ quan trọng của các chỉ tiêu để tiến hành xây dựng ma trận so sánh cặp và tính trọng số phù hợp. Vai trò của các yếu tố ảnh hưởng tới trượt lở được phản ánh như sau:

Yếu tố	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Trọng số
Slope(1)	1	3	3	4	4	4	7	7	9	0.2891
Elevation(2)	1/3	1	3	4	5	5	7	7	9	0.2340
Thachhoc(3)	1/3	1/3	1	3	4	5	7	7	7	0.1722
Dis_dutgay(4)	1/4	1/4	1/3	1	3	5	5	5	7	0.1212
Dis_thachhoc(5)	1/4	1/5	1/4	1/3	1	3	5	5	5	0.0836
Dis_river(6)	1/4	1/5	1/5	1/5	1/3	1	3	3	5	0.0551
Dis_mainroad(7)	1/7	1/7	1/7	1/5	1/5	1/3	1	3	3	0.0340
Dis_road(8)	1/7	1/7	1/7	1/5	1/5	1/3	1/3	1	3	0.0261
Dis_dancu(9)	1/9	1/9	1/7	1/7	1/5	1/5	1/3	1/3	1	0.0166
CI = 0.14235, RI = 1.45, CR = 0.0982 < 0.1 => Thỏa mãn										

Mỗi phần tử của ma trận thể hiện mức độ quan trọng của tiêu chí i so với tiêu chí j. Giá trị này có thể được chọn theo thang điểm 1-9 (theo quy tắc của Saaty), trong đó:

- 1: hai tiêu chí quan trọng như nhau.
- 3: tiêu chí i quan trọng hơn một chút so với j.
- 5: tiêu chí i quan trọng hơn đáng kể so với j.
- 7: tiêu chí i rất quan trọng so với j.
- 9: tiêu chí i cực kỳ quan trọng so với j.
- 2, 4, 6, 8: các giá trị trung gian.

Để tính trọng số:

- Chuẩn hóa ma trận: tổng các cột, chia mỗi phần tử trong cột cho tổng cột tương ứng.
- Trung bình cộng các giá trị trong mỗi hàng.

Tính CI (Consistency Index – Chỉ số nhất quán) được sử dụng để kiểm tra tính nhất quán của ma trận so sánh cặp trong phân tích thứ bậc (AHP):

- Nhân ma trận ban đầu A với vecto trọng số w
- Tính $\lambda_{max} = \frac{\sum \frac{Aw}{w}}{n}$ = 10.1388
- $\mathrm{CI} = \frac{\lambda_{max} - n}{\mathrm{n}-1}$, n là số yếu tố (n = 9) = 0.14235

RI (Random Consistency Index) là một giá trị được sử dụng để kiểm tra tính nhất quán của ma trận so sánh cặp trong phương pháp AHP:

21002152
Nguyễn Hữu Trung Kiên

Giá trị RI phụ thuộc vào số lượng yếu tố nnn trong ma trận. Dưới đây là bảng giá trị của RI theo số yếu tố:

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

CR (Consistency Ratio) là một chỉ số được sử dụng để đánh giá mức độ nhất quán của ma trận so sánh cặp trong phương pháp phân tích thứ bậc AHP.

$$CR = CI / RI$$

Do chỉ số CR nhỏ hơn 0.1 nên kết quả tính toán trọng số có thể chấp nhận được và có thể sử dụng cho các ứng dụng cao hơn.