ĐẠI HỌC QUỐC GIA HÀ NỘI TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Phạm Tuấn Đạt
Mã sinh viên : 22028218

Phân loại lớp phủ (mái nhà)

Môn học: *Trí tuệ nhân tạo (2425II_INT3401E_2)*

HÀ NỘI - 2025

I. Tóm tắt bài toán

1.1 Giới thiệu bài toán

Trong thời đại số hóa và đô thị hóa nhanh chóng, việc quản lý hiệu quả không gian xây dựng trở nên cấp thiết hơn bao giờ hết. Một trong những giải pháp công nghệ mang tính đột phá là ứng dụng Machine learning để phát hiện và phân loại mái nhà từ ảnh vệ tinh hoặc ảnh chụp trên cao. Mô hình này không chỉ giúp xác định chính xác vị trí, hình dạng và đặc điểm mái nhà, mà còn mở ra nhiều cơ hội ứng dụng trong các lĩnh vực trọng yếu. Giúp tự động hóa quy trình thay thế các phương pháp thủ công tốn kém, đặc biệt trong công tác quy hoạch, cấp phép xây dựng và giám sát vi phạm. Cải thiện hiệu quả phát triển đô thị thông minh, quản lý hạ tầng đô thị một cách thông minh hơn. Tối ưu tiềm năng năng lượng tái tạo, nhận diện đặc điểm mái nhà giúp đánh giá khả năng lắp đặt tấm pin mặt trời, từ đó giúp phát triển năng lượng xanh.

1.2 Tổng quan giải pháp

Để giải quyết bài toán, tôi xác định có các bước chính sau:

- Thu thập và xử lí dữ liệu: Sử dụng ảnh vệ tinh làm dữ liệu đầu vào, thực hiện tiền xử lí như cắt, chuẩn hoá dữ liệu để đảm bảo chất lượng huấn luyện mô hình.
- Lựa chọn và xây dựng mô hình học máy: Chọn thuật toán phù hợp phụ thuộc vào đặc điểm dữ liệu và mục tiêu nhận diện. Mô hình sẽ học cách phân biệt mái nhà dựa trên các đặc trưng hình ảnh
- Huấn luyện và tối ưu mô hình : Sử dụng tập dữ liệu đã gán nhãn để huấn luyện và tối ưu mô hình
- Đánh giá và triển khai: Đánh giá mô hình bằng các chỉ số như độ chính xác, recall, F1-score, sau đó triển khai mô hình để nhận biết mái nhà trên tập test.

II. Dữ liệu sử dụng

2.1 Mô tả dữ liệu

- Dữ liệu đầu vào bao gồm ảnh vệ tinh Sentinel 1A/B, Sentinel 2A/B : Dữ liệu được thu thập từ vệ tinh Sentinel 1 và 2 của Uỷ ban vũ trụ Châu Âu triển khai và vận hành. Sentinel 1 cung cấp dữ liệu về cường độ phản xạ của mặt đất với radar, giúp nhận diện các cấu trúc hình học (mái ngói, mái bằng, v.v). Sentinel 2 cung cấp dữ liệu về màu tự nhiên (nhận diện màu sắc của đa phần các mái nhà)
- Dữ liệu ảnh vệ tinh Google Earth Pro cung cấp ảnh
- Shapefile của tỉnh Ninh Bình : Dữ liệu khu vực Ninh Bình được cắt từ file bản đồ Việt Nam GADM (Database of Global Administrative Areas) với toạ độ có trước.

2.2 Xử lí dữ liệu

a) Chuẩn bị dữ liệu
    - Truy cập lấy dữ liệu của 2 vệ tinh Sentinel 1, Sentinel 2 từ Google Earth Engine. Sau khi truy cập được dữ liệu, shapefile tỉnh Ninh Bình được gán vào và ảnh từ 2 vệ tinh sẽ được cắt theo shapefile này.

```
var table: Table projects/ee-22028218/assets/Ninh_Binh

function clip(img) {
  return img.clip(table)
}

var collection = ee.ImageCollection('COPERNICUS/S1_GRD')
                  .filterDate('2023-01-01', '2023-12-31')
                  .filterBounds(table)
                  .map(clip)
                  .filter(ee.Filter.eq('orbitProperties_pass', 'ASCENDING'));
print(collection)
```

Ảnh minh hoạ cắt shapefile tại dữ liệu của Sentinel-1

- Dữ liệu sau khi cắt ảnh sẽ được chạy hàm median() để lấy các trung vị của mỗi pixel trong tập hợp ảnh vệ tinh. Việc lấy trung vị giúp giảm nhiễu do mây, bóng mây hoặc các yếu tố ngoại cảnh tạm thời của dữ liệu vệ tinh. Sau đó xuất dữ liệu lấy được dưới dạng file .tif

```
var composite = collection.median().toFloat()
```

Hàm median() tính trung vị cho mỗi pixel

- Dùng ứng dụng Google Earth Pro, lấy mẫu các mái nhà và không phải mái nhà thường gây nhầm lẫn (như cây cối, vùng nước) của một vùng thành phố Hoa Lư của Ninh Bình, nơi có địa hình điển hình của tỉnh Ninh Bình.

- Sử dụng Arcmap, tạo ra các điểm ngẫu nhiên trên các vùng được lấy mẫu trước đó bằng tool Create Random Points, và lưu lại các toạ độ của những điểm này dưới dạng bảng.

label	lat	lon
2	20.25132	105.9641
2	20.25131	105.9642
2	20.25135	105.9642
2	20.25132	105.9641
2	20.25135	105.9642
2	20.25129	105.9642
2	20.25136	105.9641
2	20.25132	105.9641
2	20.2513	105.9642

Ví dụ các toạ độ và label của các điểm ngẫu nhiên được xuất ra file excel.

- Ghép các trường dữ liệu có được từ file .tif trước đó và bảng toạ độ các điểm đã lấy mẫu bao gồm các trường sau, đối với dữ liệu vệ tinh Sentinel 1 sẽ cung cấp 2 trường : VV, VH (giá trị phản xạ dọc - dọc, giá trị phản xạ dọc - ngang để nhận biết bề mặt cứng, phẳng hay phân tán), đối với dữ liệu vệ tinh Sentinel 2 sẽ cung cấp 3 trường : B02, B03, B04 (nhận biết màu mái nhà so với vật thể khác) và 2 trường toạ độ : kinh độ, vĩ độ của các pixel lấy ngẫu nhiên trước đó.

label	lat	lon	vv	vh	b2	b3	b4
2	20.25132	105.9641	-4.67673	-12.2651	0.02385	0.01845	0.04935
2	20.25131	105.9642	-6.21829	-13.6618	0.02385	0.01765	0.04855
2	20.25135	105.9642	-6.21829	-13.6618	0.02385	0.01765	0.04855
2	20.25132	105.9641	-6.21829	-13.6618	0.02385	0.01765	0.04855
2	20.25135	105.9642	-7.28927	-13.6089	0.02385	0.01475	0.0489
2	20.25129	105.9642	-6.21829	-13.6618	0.02385	0.01765	0.04855
2	20.25136	105.9641	-5.44883	-12.0515	0.02385	0.01315	0.04735
2	20.25132	105.9641	-6.21829	-13.6618	0.02385	0.01765	0.04855
2	20.2513	105.9642	-6.21829	-13.6618	0.02385	0.01765	0.04855
2	20.25131	105.9641	-6.21829	-13.6618	0.02385	0.01765	0.04855
2	20.25137	105.9641	-7.28927	-13.6089	0.02385	0.01475	0.0489

Các trường dữ liệu sau khi được ghép với tọa độ tương ứng.

b) Phân tích và tiền xử lí dữ liệu

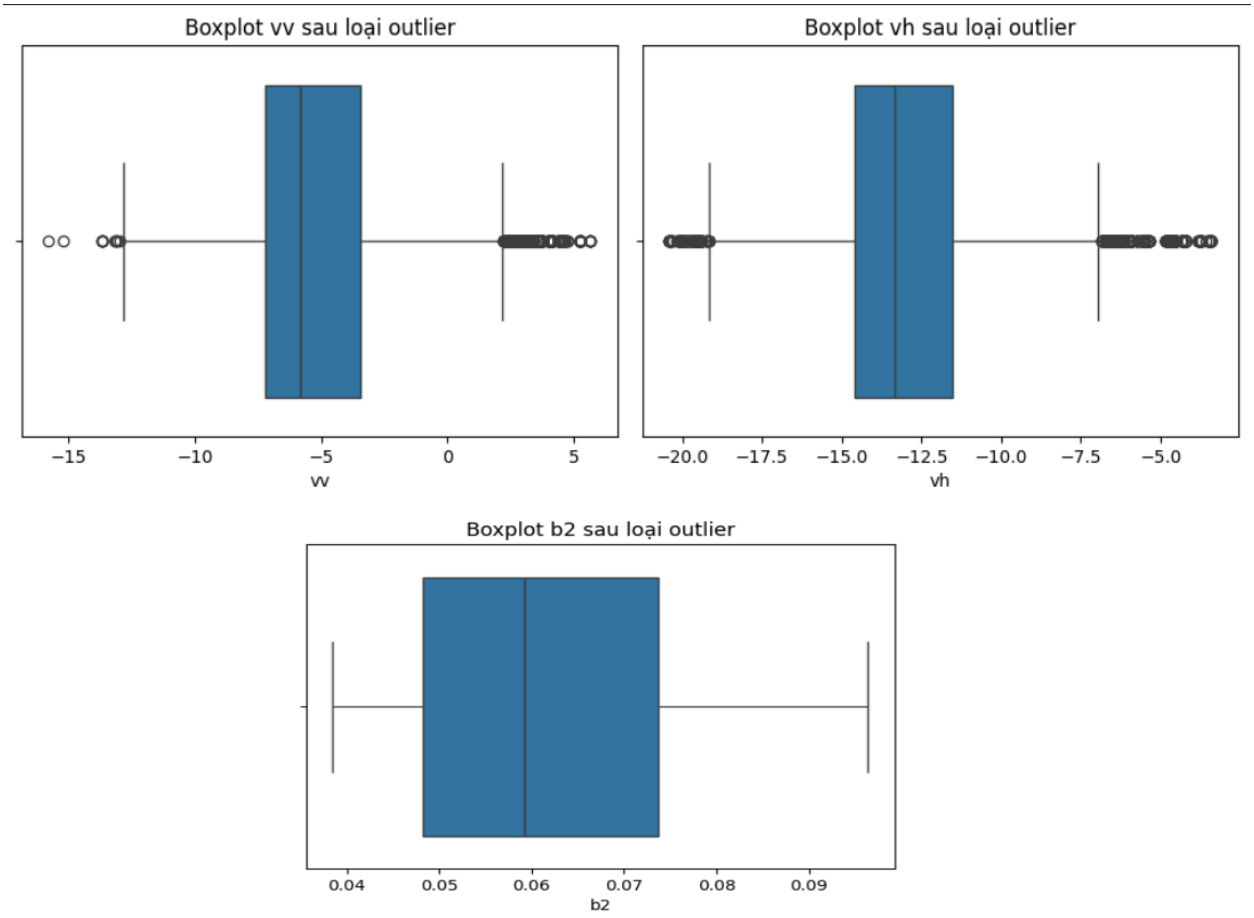
Sau bước tiền xử lý, dữ liệu thu được có đặc điểm sau:

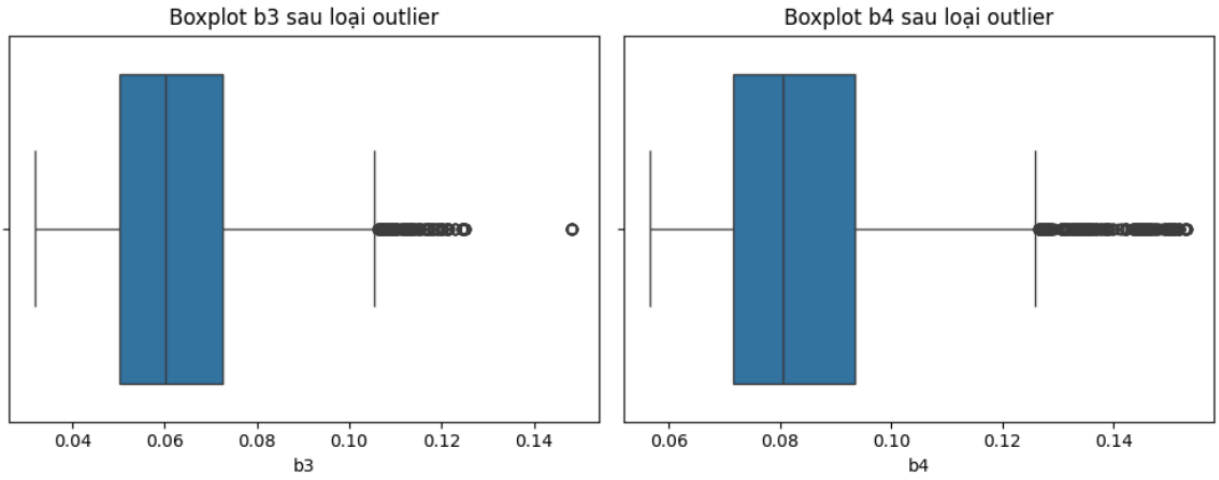
- Các trường không có giá trị NULL
- Ở 2 trường VV, VH, giá trị của VV và VH cần được log-transform do giá trị radar phân bố log và do radar thường có nhiễu, sai số và ảnh hưởng do các vật thể bất thường như mây nên có những giá trị của VV hoặc VH nằm xa với giá trị trung bình, hoặc giữa 2 giá trị có độ lệch quá lớn.
- Ở các trường B02, B03, B04 có những giá trị phản xạ có giá trị cực cao hoặc cực thấp do bóng mây

Vậy cần xử lý các giá trị ngoại lai của các trường này để tối ưu dữ liệu phục vụ cho việc học máy. Sử dụng phương pháp percentile để loại bỏ các outliers này khỏi tệp dữ liệu.

```python
feature_cols = ['vv', 'vh', 'b2', 'b3', 'b4']

def remove_outliers_percentile(df, cols, lower=1, upper=99):
    df_clean = df.copy()

    # Chuyển các cột sang kiểu số, lỗi chuyển đổi thành NaN
    for col in cols:
        df_clean[col] = pd.to_numeric(df_clean[col], errors='coerce')

    # Loại bỏ dòng có giá trị NaN sau khi chuyển đổi
    df_clean = df_clean.dropna(subset=cols)

    for col in cols:
        low = np.percentile(df_clean[col], lower)
        up = np.percentile(df_clean[col], upper)
        df_clean = df_clean[(df_clean[col] >= low) & (df_clean[col] <= up)]

    return df_clean

df_clean = remove_outliers_percentile(df, feature_cols)
```

Và đây là kết quả trực quan hoá dữ liệu sau khi loại bỏ outliers.





III. Mô hình sử dụng và phương pháp đánh giá

3.1. Chia tập train/test

- Dữ liệu là tập hợp các điểm ngẫu nhiên tượng trưng cho các pixel trong ảnh vệ tinh sẽ được chia tập train và test theo tỉ lệ tương ứng 80:20.
- Tỉ lệ giữa các label vẫn được giữ nguyên trong cả tập train và test phù hợp cho bài toán phân loại không cân bằng.

3.2. Lựa chọn mô hình phù hợp

a) Mô hình Random Forest
   - Ưu điểm :
     + Random Forest có khả năng học từ dữ liệu đa chiều, không tuyến tính như các trường dữ liệu VV/ VH và B02/B03/B04. Khác với SVM và Logistic Regression yêu cầu cao về dữ liệu tuyến tính.
     + Random Forest ít nhạy cảm với các outliers còn sót. Nếu vẫn chưa xử lý hết các giá trị ngoại lai, Random Forest vẫn hoạt động ổn định.
     + Khả năng tổng quát tốt khi mô hình tổ hợp từ nhiều cây quyết định, tránh overfitting. Đặc biệt có dữ liệu không đồng đều (mái nhà nhiều loại, nhiều màu sắc) và dữ liệu có nhiễu nhẹ do vệ tinh.
   - Nhược điểm :
     + Cần tinh chỉnh tham số (n-estimator và max_depth) cẩn thận vì có thể gây overfit nhẹ.
     + Yêu cầu bộ nhớ cao hơn Logistic Regression và SVM.

b) Mô hình Linear Regression
   - Ưu điểm:
     + Nắm bắt được quan hệ phi tuyến tính giữa các bands.
     + Có margin tối ưu, hạn chế outlier.
   - Nhược điểm:
     + Thời gian huấn luyện tăng nhanh, đặc biệt với 24000 mẫu có thể bị chậm đáng kể.
     + Cần chuẩn hoá dữ liệu tốn công.

c) Mô hình SVM
   - Ưu điểm:

+ Đơn giản, nhanh chóng trong khâu huấn luyện lẫn dự đoán.
+ Ít tốn bộ nhớ, dễ triển khai.

- Nhược điểm:
  + Nhạy cảm giữa các bands liên quan tới nhau.
  + Cần chuẩn hoá, có thể kém chính xác nếu dữ liệu phân lớp phi tuyến mạnh.

3.3. Phương pháp đánh giá

a) Các phương pháp sẽ sử dụng

Ta sử dụng các phương pháp sau đây để đánh giá hiệu quả của mô hình:

1. Confusion Matrix

- Cung cấp cái nhìn chi tiết về hiệu suất dự đoán của mô hình qua các tham số TP (True positive thể hiện số lượng mái nhà được dự đoán đúng), TN (True Negative, số lượng không phải mái nhà được dự đoán đúng), FP (False Positive, số lượng không phải mái nhà bị dự đoán thành mái nhà) và FN (False Negative, số lượng mái nhà nhưng bị dự đoán là không phải mái nhà)

2. Độ chính xác là tỷ lệ dự đoán đúng trên tổng số mẫu, không đủ thể hiện hiệu suất nếu mẫu mái nhà ít hơn so với mẫu không phải mái nhà
   Accuracy = (TP + TN) / (TP + TN + FP + FN)
3. Độ nhạy (Recall) : đảm bảo mô hình không bỏ sót quá nhiều mái nhà
   Recall = TP / (TP + FN)

- Tỉ lệ recall này nói lên được mô hình dự đoán được bao nhiêu mẫu thực sự là mái nhà trên tổng số mẫu thực sự là mái nhà

- Nếu mô hình tìm được càng nhiều mái nhà thực sự (và có thể dự đoán sai mẫu đó là mái nhà mặc dù nó không phải mái nhà) thì Recall phải cao, lý tưởng thì Recall = 1

4. F1-Score : trung bình điều hoà giữa Precision và Recall

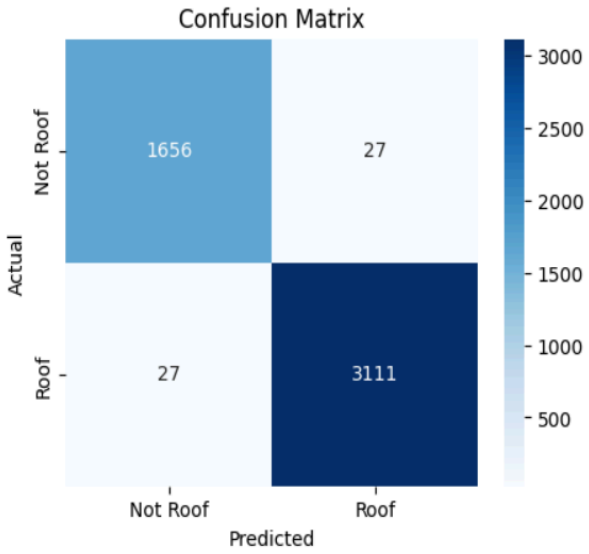
F1-Score = 2 x (Precision x Recall) / (Precision + Recall)
với Precision = TP / (TP + FP)

- Nếu mô hình dự đoán ra mẫu nào là mái nhà và mẫu đó luôn là mái nhà (và có thể bỏ sót mái nhà do dự đoán mẫu đó không phải mái nhà) thực sự thì Precision phải cao, lý tưởng thì Precision = 1. Vì vậy cần dùng F1-Score để cân bằng được độ chính xác của dự đoán mái nhà và khả năng phát hiện mái nhà.

b) Kết quả đánh giá của từng mô hình

- Mô hình Random Forest:
  + Lấy n-estimator 1 giá trị thường được sử dụng là 100 thu được kết quả như sau:

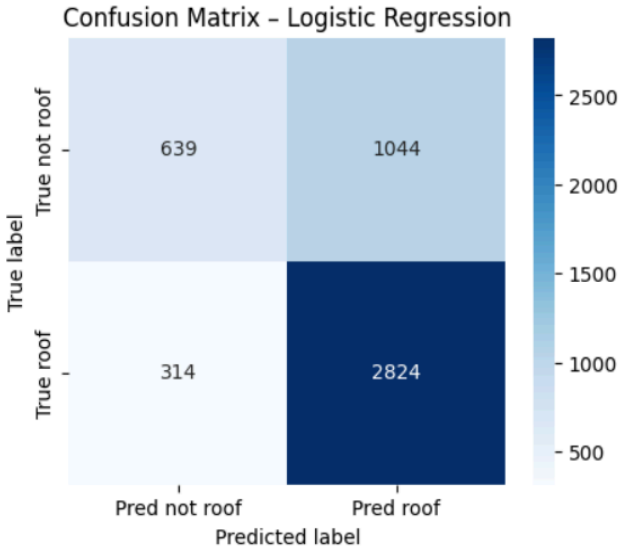
```
Confusion Matrix:
[[1656   27]
 [  27 3111]]
TP (True Positive): 3111
TN (True Negative): 1656
FP (False Positive): 27
FN (False Negative): 27
Accuracy: 0.9888
Recall: 0.9914
Precision: 0.9914
F1-Score: 0.9914
```



Kết quả nhận thấy độ chính xác cao, để tiết kiệm tài nguyên không cần tăng giá trị của n-estimator nữa mà vẫn đạt được độ hiệu quả.

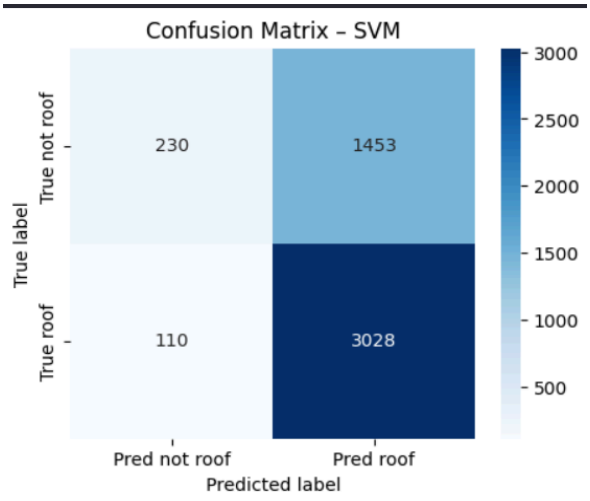
- Mô hình Logistic Regression:
  - + Đối với các tham số: max_iter = 1000 đảm bảo thuật toán sẽ hội tụ, solver = lbfgs được khuyến khích sử dụng cho dữ liệu có kích thước vừa, nhiều biến. Penalty = 12 mặc định là lựa chọn an toàn, tránh overfitting.
  - + Ta thu được kết quả như sau:

```
Confusion Matrix:
[[ 639 1044]
 [ 314 2824]]
TP (True Positive): 2824
TN (True Negative): 639
FP (False Positive): 1044
FN (False Negative): 314
Accuracy:  0.7183
Recall:    0.8999
Precision: 0.7301
F1-Score:  0.8062
```



- Mô hình SVM:
  - + Đối với các tham số : kernel = ‘RBF’ chuyển không gian dữ liệu sang dạng phi tuyến, do dữ liệu vệ tinh Sentinel thường phi tuyến không gian. C = 1.0 mặc định, cân bằng giữa overfitting và underfitting. gamma = ‘scale’ mặc định. probability = ‘False’ do chỉ cần phân lớp, giúp tăng tốc độ suy luận.
  - + Kết quả thu được là:

```
Confusion Matrix:
 [[ 230 1453]
 [ 110 3028]]
TP: 3028, TN: 230, FP: 1453, FN: 110
Accuracy : 0.6758
Precision: 0.6757
Recall   : 0.9649
F1-Score : 0.7949
```



=> Sau kết quả thu được giữa ba mô hình, nhận thấy Random Forest khởi tạo đơn giản và lại mang lại hiệu quả dự đoán cao nhất.