

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



# BÁO CÁO LUẬN VĂN TỐT NGHIỆP

---

Xây dựng mô hình gán nhãn vật thể từ ảnh vệ tinh

Giảng viên hướng dẫn : ThS. Trương Quang Hải

Giảng viên phản biện : TS. Nguyễn Thanh Bình

Nhóm sinh viên thực hiện : Bùi Tấn Kính - 51301974

Phan Văn Hưng - 51301653

Phạm Ngọc Lam - 51301989



## Lời cam đoan

Luận văn của chúng tôi có tham khảo các tài liệu, bài báo, trang web như được trình bày ở mục tài liệu tham khảo và ở mỗi tham khảo chúng tôi đều trích dẫn nguồn gốc. Chúng tôi xin cam đoan rằng ngoài những trích dẫn từ các tham khảo trên, toàn bộ nội dung trong báo cáo là do chính chúng tôi thực hiện và được sự hướng dẫn của ThS. Trương Quang Hải. Nếu phát hiện có bất kỳ sự gian lận nào, chúng tôi xin chịu hoàn toàn trách nhiệm.

TP. Hồ Chí Minh, 12/2017  
Phan Văn Hưng  
Bùi Tân Kính  
Phạm Ngọc Lam



## Lời cảm ơn

Dưới sự chỉ bảo tận tình của quý Thầy Cô của trường Đại học Bách Khoa TP. Hồ Chí Minh, đặc biệt là quý Thầy Cô khoa Khoa học và Kỹ thuật Máy Tính, chúng tôi đã được tiếp thu những kiến thức quý báu trong lĩnh vực công nghệ thông tin cũng như các kỹ năng liên quan. Những kiến thức này là nền tảng cực kỳ quan trọng cho quá trình thực hiện luận văn cũng như là hành trang cần thiết cho quá trình nghiên cứu và làm việc trong tương lai.

Với những kiến thức này cũng như sự nỗ lực và hợp tác của các thành viên trong nhóm, chúng tôi đã hoàn thành luận văn của mình và đạt được những kết quả nhất định. Với những kết quả đạt được này, chúng tôi xin được gửi lời cảm ơn chân thành đến quý Thầy Cô trường Đại học Bách Khoa TP. Hồ Chí Minh, quý Thầy Cô khoa Khoa học và Kỹ thuật Máy tính đã truyền đạt những kiến thức và tạo điều kiện để chúng tôi được học tập và nghiên cứu trong suốt thời gian qua.

Chúng tôi xin tỏ lòng biết ơn sâu sắc đến ThS. Trương Quang Hải đã tận hình hướng dẫn chúng tôi trong suốt thời gian học tập, nghiên cứu và hiện thực đề tài. Thầy đã luôn theo sát và hỗ trợ chúng tôi trong việc định hướng, tìm hiểu và phát triển đề tài, cũng như có những hướng dẫn và nhận xét quý giá trong suốt thời gian hoàn thành bài báo cáo này.

Do kiến thức và kỹ năng còn hạn hẹp, không tránh khỏi những thiếu sót trong cách hiểu và trình bày, nhóm chúng tôi rất mong nhận được sự góp ý của quý Thầy Cô để có được những kết quả tốt hơn.

Cuối cùng, chúng tôi xin gửi lời chúc sức khỏe và những điều tốt đẹp nhất đến quý Thầy Cô, các bậc phụ huynh và các anh chị.

TP. Hồ Chí Minh, 12/2017  
Phan Văn Hưng  
Bùi Tân Kính  
Phạm Ngọc Lam



## Tóm tắt luận văn

Luận văn của chúng tôi diễn giải hướng giải quyết và cách hiện thực các mô hình nhận dạng vật thể trong hình ảnh vệ tinh, bao gồm 10 loại vật thể cần nhận dạng:

1. Tòa nhà bao gồm: tòa nhà lớn, khu dân cư ...
2. Công trình nhân tạo
3. Đường lớn
4. Đường nhỏ bao gồm: lối đi bộ, đường mòn ...
5. Cây bao gồm: rừng cây, nhóm cây, các loại cây riêng lẻ ...
6. Thảm thực vật bao gồm: đất trống trọt, vụ mùa ...
7. Đường thủy
8. Nước đứng bao gồm: ao, hồ ...
9. Xe lớn bao gồm xe tải, xe buýt ...
10. Xe nhỏ bao gồm: xe ô tô, xe máy ...

Bố cục của luận văn gồm có các phần như sau:

Chương 1 là chương giới thiệu tổng quan đề tài. Ở đó sẽ trình bày tầm quan trọng của ảnh vệ tinh, nhu cầu xã hội trong việc gán nhãn vật thể trong ảnh vệ tinh. Lý do vì sao chúng tôi chọn đề tài này?

Chương 2 sẽ trình bày các kiến thức mà chúng tôi tìm hiểu được liên quan đến đề tài. Đó là những kiến thức cơ bản về quá trình phân lớp dữ liệu cũng như các thuật toán liên quan như: Support Vector Machine, Logistic Regression. Ngoài ra chúng tôi còn trình bày khái quát thư viện sử dụng trong quá trình hiện thực và bài toán phân lớp cho dữ liệu dạng hình ảnh vệ tinh.

Chương 3 chúng tôi sẽ giới thiệu khái quát tập dữ liệu sử dụng trong luận văn, xử lý dữ liệu trên tập dữ liệu đã cung cấp và việc xây dựng mô hình nhận dạng cũng như kết quả đạt được trong từng mô hình.

Chương 4 là chương tổng kết, đánh giá những việc chúng tôi đã làm được và đề xuất hướng phát triển.

Chương 5 là chương cuối cùng trình bày những tài liệu chúng tôi đã tham khảo.

# Mục lục

<b>Mục lục</b>	<b>4</b>	
<b>Danh sách hình vẽ</b>	<b>6</b>	
<b>1</b>	<b>Giới thiệu</b>	<b>8</b>
1.1	Giới thiệu đề tài . . . . .	8
1.2	Lý do chọn đề tài . . . . .	8
1.3	Mục tiêu . . . . .	8
<b>2</b>	<b>Cơ sở lý thuyết</b>	<b>9</b>
2.1	Bài toán phân lớp . . . . .	9
2.1.1	Khái niệm[1] . . . . .	9
2.1.1.1	Cây quyết định[3] . . . . .	10
2.1.1.2	Mạng Bayes[5] . . . . .	10
2.1.1.3	Support Vector Machine (SVM)[6]	11
2.1.2	Phương pháp huấn luyện[8] . . . . .	11
2.2	Quá trình phân lớp dữ liệu . . . . .	12
2.2.1	Thu thập dữ liệu . . . . .	12
2.2.2	Tiền xử lí dữ liệu[11] . . . . .	13
2.2.2.1	Làm sạch dữ liệu[13] . . . . .	14
2.2.2.2	Tích hợp dữ liệu [20] . . . . .	20
2.2.2.3	Thu giảm dữ liệu[21] . . . . .	21
2.2.2.4	Chuyển đổi dữ liệu[23] . . . . .	23
2.2.3	Xây dựng mô hình phân lớp[24] . . . . .	24
2.2.4	Vấn đề overfitting[27] . . . . .	25
2.2.4.1	Dịnh nghĩa . . . . .	25
2.2.4.2	Nguyên nhân . . . . .	26
2.2.4.3	Phương pháp tránh overfitting . . . . .	26
2.2.5	Dánh giá mô hình[28] . . . . .	26
2.2.6	Phương pháp cải tiến hiệu suất phân loại[30] . . . . .	28
2.2.6.1	Khái niệm kết hợp các bộ phân loại . . . . .	28
2.2.6.2	Các cách tiếp cận phương pháp kết hợp các bộ phân loại . . . . .	28
2.3	Các thuật toán phân lớp dữ liệu . . . . .	31
2.3.1	Support Vector Machine (SVM)[32] . . . . .	31
2.3.1.1	Giới thiệu . . . . .	31
2.3.1.2	Ý tưởng của giải thuật . . . . .	31
2.3.1.3	Xây dựng bài toán tối ưu cho Support Vector Machine	32
2.3.2	Logistic Regression[36] . . . . .	34
2.3.2.1	Giới thiệu . . . . .	34



2.3.2.2	Mô hình Logistic Regression . . . . .	34
2.3.2.3	Xây dựng hàm matsu và phương pháp tối ưu . . . . .	35
2.3.2.4	Xây dựng hàm chi phí . . . . .	35
2.3.2.5	Tối ưu hàm chi phí . . . . .	36
2.3.3	Áp dụng phân loại nhị phân cho bài toán phân lớp[23] . . . . .	37
2.3.3.1	One-vs-one . . . . .	37
2.3.3.2	Hierarchical . . . . .	37
2.3.3.3	One-vs-rest . . . . .	38
2.4	Phân lớp cho dữ liệu dạng hình ảnh vệ tinh . . . . .	39
2.4.1	Giới thiệu . . . . .	39
2.4.2	Nhu cầu trong việc phân loại ảnh vệ tinh . . . . .	39
2.4.3	Các kỹ thuật trong xử lý ảnh vệ tinh . . . . .	39
2.4.4	Các phương pháp phân loại ảnh vệ tinh . . . . .	40
<b>3</b>	<b>Phương pháp đề xuất và đánh giá</b>	<b>43</b>
3.1	Khảo sát dữ liệu . . . . .	43
3.1.1	Giới thiệu tổng quát về tập dữ liệu . . . . .	43
3.1.2	Mô tả tập ảnh . . . . .	46
3.1.3	Tập dữ liệu huấn luyện và đánh giá . . . . .	50
3.1.3.1	Tập dữ liệu huấn luyện . . . . .	50
3.1.3.2	Dánh giá tập dữ liệu huấn luyện . . . . .	51
3.1.4	Chuyển đổi tập dữ liệu huấn luyện . . . . .	53
3.2	Tiền xử lý dữ liệu . . . . .	54
3.3	Xây dựng mô hình phân loại dựa trên phương pháp Stochastic Gradient Descent . . . . .	56
3.3.1	Xây dựng mô hình . . . . .	56
3.3.2	Kết quả đạt được . . . . .	65
<b>4</b>	<b>Kết luận</b>	<b>70</b>
4.1	Kết quả đạt được . . . . .	70
4.2	Hạn chế . . . . .	70
4.3	Hướng phát triển . . . . .	70
<b>5</b>	<b>Tài liệu tham khảo</b>	<b>71</b>
<b>Tài liệu</b>		<b>71</b>

## Danh sách hình vẽ

1	Ví dụ bài toán phân loại e-mail[2]	9
2	Phân lớp bằng cây quyết định[4]	10
3	Cực đại khoảng cách lề trong SVM[7]	11
4	Tập dữ liệu dùng để huấn luyện trong Nhận dạng chữ viết tay[9]	12
5	Bộ cơ sở dữ liệu chữ viết tay MNIST[10]	13
6	Tổng quan quá trình tiền xử lí dữ liệu[12]	14
7	Làm mịn dữ liệu bằng phương pháp phân khoáng	16
8	Ảnh hưởng của phần tử ngoại lai đến mô hình dự đoán[15]	17
9	Giải thuật DBSCAN[16]	18
10	Biểu đồ Box Plot cho ví dụ trên[18]	19
11	Biểu đồ phân bố dữ liệu trong ví dụ[19]	19
12	Quá trình xây dựng một bộ phân loại[25]	25
13	Phân loại đối với dữ liệu mới[26]	25
14	Các chỉ số đánh giá độ chính xác hệ thống phân loại[29]	26
15	Minh họa độ đo Jaccard	27
16	Mô hình phân loại hỗn hợp[31]	28
17	Các mặt phân cách hai lớp dữ liệu [33]	31
18	Phân tích bài toán SVM[34]	32
19	Các điểm gần mặt phân cách nhất của hai lớp được khoanh tròn[35]	33
20	Các activation function khác nhau[37]	35
21	Ví dụ về sự phân bố các lớp trong bài toán phân loại nhiều lớp	38
22	Dự đoán tòa nhà.	44
23	Dự đoán thảm thực vật.	44
24	Dự đoán đường lớn.	45
25	Dự đoán đường nhỏ.	45
26	Dự đoán xe nhỏ.	45
27	Dự đoán đường thủy.	46
28	Ảnh 3-band minh họa với id 6110_3_1	47
29	Ảnh A minh họa với id 6110_3_1	48
30	Ảnh M minh họa với id 6110_3_1	49
31	Ảnh P minh họa với id 6110_3_1	50
32	Cấu trúc tập huấn luyện	51
33	Lượt đồ tần suất xuất hiện của các nhãn trong tập huấn luyện	52
34	Các phương tiện giao thông bị gán sai nhãn	53
35	Ảnh RGB nhãn loại 1 thường xuất hiện	54
36	Ảnh RGB nhãn loại 1 bất thường	55
37	Ảnh M nhãn loại 1 thường xuất hiện	55
38	Ảnh M nhãn loại 1 bất thường	55
39	Lượt đồ tóm tắt quá trình xây dựng mô hình gán nhãn	56



---

41	Hình ảnh minh họa trước và sau khi loại bỏ phần tử biên.	58
43	Nhóm 1.	60
45	Nhóm 2.	61
47	Nhóm 3.	62
49	Nhóm 4.	63
51	Nhóm 5.	64
53	Hình ảnh RGB tập kiểm tra và nhãn của chúng.	66
55	Hình ảnh RGB tập kiểm tra và nhãn của chúng.	67
56	Màu đại diện cho các nhãn	67
58	Hình ảnh dự đoán được khi áp dụng Log (ở giữa) và SVM (bên phải).	68
60	Hình ảnh dự đoán được khi áp dụng Log (ở giữa) và SVM (bên phải).	69



## 1 Giới thiệu

### 1.1 Giới thiệu đề tài

Trong những thập kỷ gần đây, hình ảnh vệ tinh ngày càng trở nên đa dạng và phong phú. Nó đóng một vai trò quan trọng trong việc cung cấp thông tin địa lý, cho phép quan sát trái đất chính xác và các phép đo địa hình. Ảnh vệ tinh giúp cho việc theo dõi gián tiếp sự thay đổi của một khu vực mà không tốn quá nhiều thời gian đo đạc thực tế. Nó cho phép giám sát sự tác động của biến đổi khí hậu, sự thay đổi địa hình, cơ sở hạ tầng của một khu vực.

Số lượng dữ liệu nhận được tại các trung tâm dữ liệu vệ tinh là rất lớn và nó đang phát triển theo cấp số nhân. Vì công nghệ đang phát triển với tốc độ nhanh chóng, nên cần hình thành cơ chế hiệu quả và đáng tin cậy để trích xuất và giải thích các thông tin có giá trị từ các hình ảnh vệ tinh. Chính vì sự gia tăng đó dẫn tới nhu cầu gán nhãn cho ảnh vệ tinh ngày càng tăng cao, việc gán nhãn trở thành thách thức của cộng đồng khai phá dữ liệu. Việc nhận dạng hàng loạt ảnh bằng mắt thường rất là khó khăn và tốn thời gian, nên dẫn đến việc sử dụng các giải thuật học máy để gán nhãn vật thể trong ảnh vệ tinh là một nhu cầu tất yếu.

Trong luận văn này sẽ tập trung vào việc xây dựng mô hình gán nhãn ảnh vệ tinh và một số giải thuật học máy để áp dụng vào quá trình gán nhãn ảnh vệ tinh.

### 1.2 Lý do chọn đề tài

Bởi vì sự cần thiết trong việc phân loại, gán nhãn các thành phần trong ảnh vệ tinh như đã trình bày ở mục 1.1. Đã có rất nhiều nhà phát triển và nhóm nghiên cứu về vấn đề này và đạt được kết quả nhất định. Nhận thấy được thách thức trong việc thực hiện thành công đề tài tác động rất lớn đến việc lựa chọn đề tài này. Một số câu hỏi đã được đặt ra:

- Làm thế nào để có thể nhận dạng được vật thể trong hình ảnh vệ tinh?
- Đặc trưng nào cần được lấy ra để nhận dạng?
- Giải thuật nào tốt cho việc nhận dạng?
- Sản phẩm gì sẽ được tạo ra sau khi kết thúc?

### 1.3 Mục tiêu

- Hiểu rõ được bản chất của đề tài cũng như hướng phát triển sau này.
- Hiểu được cơ chế của các giải thuật học máy đã học.
- Hiện thực gán được ít nhất 5/10 nhãn.

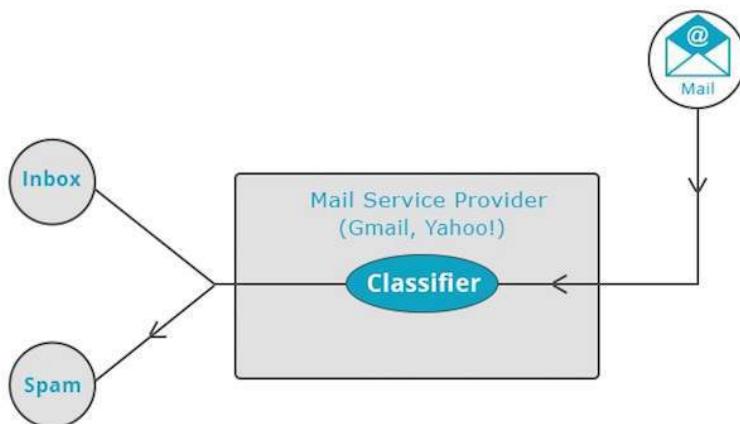
## 2 Cơ sở lý thuyết

### 2.1 Bài toán phân lớp

#### 2.1.1 Khái niệm[1]

Bài toán phân lớp là quá trình gán nhãn một đối tượng dữ liệu vào một hay nhiều lớp đã cho trước nhờ một mô hình phân loại. Mô hình này được cấu tạo từ các thuật toán. Dữ liệu đầu vào của mô hình là các đặc trưng của mẫu dữ liệu và dữ liệu đầu ra là các *nhãn* tương ứng với các điểm dữ liệu đó. Quá trình phân lớp là quá trình gán nhãn cho đối tượng dữ liệu mới.

Như vậy, nhiệm vụ của bài toán phân lớp là cần tìm một mô hình phân loại để khi có dữ liệu mới thì có thể xác định được dữ liệu đó thuộc vào phân lớp nào một cách chính xác nhất có thể.



Hình 1: Ví dụ bài toán phân loại e-mail[2]

Bài toán phân lớp được ứng dụng rộng rãi trong thực tế như nhận dạng khuôn mặt, nhận dạng giọng nói, phát hiện thư rác, chẩn đoán y khoa hay nhận dạng hình ảnh...

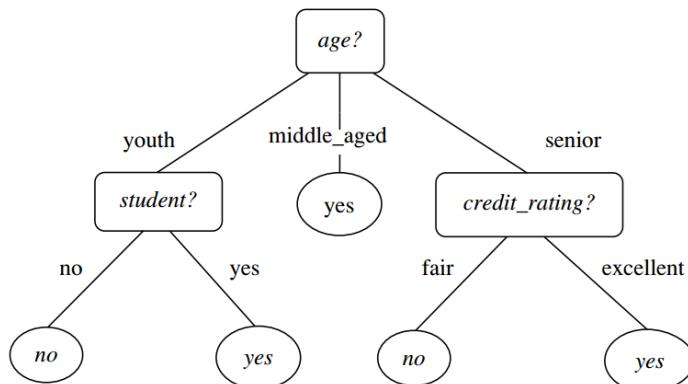
**Ví dụ:** Một nhân viên ngân hàng cần phân tích dữ liệu của khách hàng để biết được những người nào thì có thể cho vay, những người nào thì không. Một nhà quản lý tiếp thị cần phân tích dữ liệu khách hàng để biết những khách hàng có hồ sơ thu nhập như thế nào thì họ sẽ có nhu cầu mua máy tính. Trong y học, bệnh ung thư vú có 3 phương pháp điều trị tùy vào triệu chứng cũng như cơ địa của mỗi người. Các nhà nghiên cứu sẽ phân tích dữ liệu điều trị của bệnh nhân trong quá khứ để có thể dự đoán được với hồ sơ như thế này thì sẽ điều trị theo phương pháp nào. Trong các ví dụ trên, nhiệm vụ phân tích dữ liệu là để phân loại, trong đó một mô hình được xây dựng để dự đoán các *nhãn*. Ví dụ, trường hợp thứ nhất có 2 *nhãn* là "*cho vay*" hoặc "*không cho vay*", thứ 2 là "*có mua máy tính*" hoặc "*không mua máy tính*", thứ 3 là "*phương pháp A*", "*phương pháp B*" hoặc "*phương pháp C*". Các *nhãn* này được biểu diễn bằng các giá trị rời rạc, trong đó thứ tự các giá trị này là không có nghĩa. Ví dụ, trường hợp thứ nhất, ta có thể gán các *nhãn* "*cho vay*" và

"không cho vay" tương ứng với hai giá trị là 0 và 1. Tương tự cho các ví dụ còn lại.

Một số mô hình hay sử dụng trong bài toán phân loại:

### 2.1.1.1 Cây quyết định[3]

Cây quyết định là một cấu trúc cây, trong đó mỗi node trong biểu thị cho một phép phân nhánh tương ứng cho một thuộc tính, mỗi nhánh biểu thị cho một kết quả của một phép thử. Các node lá biểu thị cho lớp hoặc các phân bô lớp. Node trên cùng trong một cây được gọi là gốc. Minh họa cho cây quyết định, *Hình 2* lấy ví dụ phân lớp nhu cầu mua máy tính theo các thuộc tính. Trong đó, các node trong biểu diễn bằng các hình chữ nhật. Các quyết định (node lá) được biểu diễn bằng hình eclipse. Để phân lớp một mẫu chưa biết, những giá trị thuộc tính của mẫu đó được thử ngược lại trên cây quyết định. Một đường dẫn từ gốc đến một node lá là cơ sở cho việc dự đoán lớp của một mẫu. Cây quyết định có thể dễ dàng chuyển đổi sang một tập các luật phân lớp. Cơ sở toán học của cây quyết định là thuật toán tham lam, thuật toán này đã xây dựng cây quyết định đệ quy từ trên xuống dưới, theo phương pháp chia để trị.



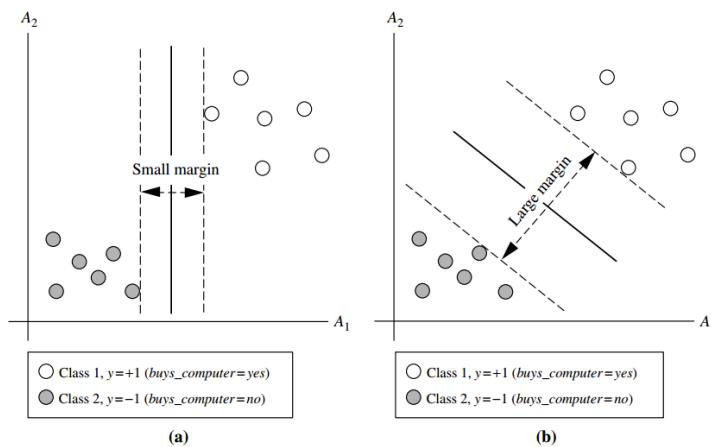
**Hình 2:** Phân lớp bằng cây quyết định[4]

### 2.1.1.2 Mạng Bayes[5]

Bayesian là phương pháp phân lớp dựa vào thống kê. Ta có thể dự đoán xác suất của các lớp trong tập dữ liệu, dựa vào xác suất này có thể xếp các mẫu vào các lớp riêng biệt. Thuật toán phân lớp Bayesian giả thiết rằng giá trị các thuộc tính của một lớp độc lập với giá trị của các thuộc tính khác. Giả thiết này còn được gọi là lớp độc lập có điều kiện, nó làm đơn giản các tính toán sau này. Mạng Bayesian là một đồ thị, trên đồ thị cho phép biểu diễn mối quan hệ giữa các thuộc tính.

### 2.1.1.3 Support Vector Machine (SVM)[6]

SVM là một phương pháp mới để phân lớp dữ liệu. Nó dễ sử dụng hơn mạng neural, tuy nhiên nếu không sử dụng nó chính xác thì dễ bị bỏ qua một số bước đơn giản nhưng cần thiết, dẫn đến kết quả không được thỏa mãn. Mục đích của phương pháp SVM là tạo ra ra một mô hình từ tập dữ liệu mẫu. Mô hình này có khả năng dự đoán lớp cho các mẫu dữ liệu. SVM tìm ra một hàm quyết định phi tuyến trong tập mẫu bằng cách ánh xạ hoàn toàn các mẫu học vào một không gian đặc trưng kích thước lớn có thể phân lớp tuyến tính và phân lớp dữ liệu trong không gian này bằng cách cực đại khoảng cách lề (geometric margin) và cực tiểu lỗi học cùng một lúc.



**Hình 3:** Cực đại khoảng cách lề trong SVM[7]

### 2.1.2 Phương pháp huấn luyện[8]

Trong lĩnh vực học máy, có 4 phương pháp để huấn luyện mô hình, đó là: học có giám sát, học không giám sát, học bán giám sát và học tăng cường. Trong đó, phương pháp *học có giám sát* thường được sử dụng trong các bài toán phân loại.

**Học có giám sát:** là phương pháp dự đoán *nhãn* của một dữ liệu mới dựa trên các cặp (*mẫu dữ liệu, nhãn*) đã biết từ trước.

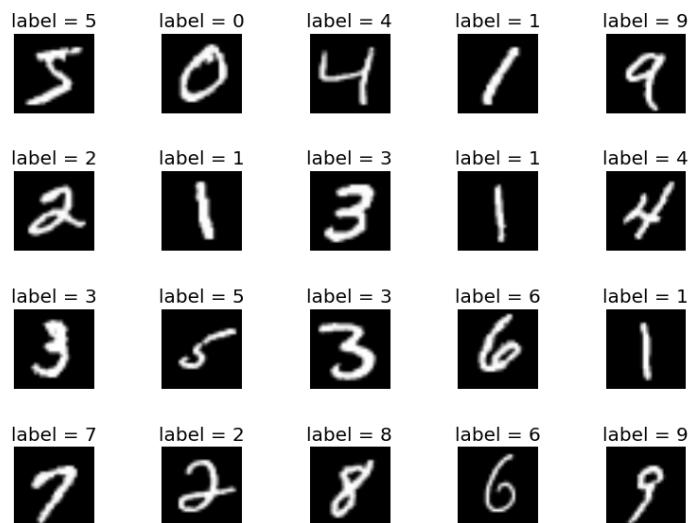
Nói theo cách khác, chúng ta có một tập dữ liệu đầu vào  $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  và một tập nhãn tương ứng  $\mathbf{Y} = y_1, y_2, \dots, y_N$ . Trong đó,  $\mathbf{x}_i, y_i$  là các vector. Các cặp dữ liệu biết trước  $(\mathbf{x}_i, y_i) \in \mathbf{X}, \mathbf{Y}$  được gọi là tập dữ liệu huấn luyện. Từ tập dữ liệu huấn luyện này, chúng ta cần tạo ra một mô hình<sup>1</sup> ánh xạ mỗi phần tử thuộc tập X sang một phần tử (xấp xỉ) tương ứng thuộc tập Y:

$$y_i \approx f(\mathbf{x}_i), \forall i = 1, 2, \dots, N.$$

Mục đích của việc huấn luyện là xấp xỉ hàm số  $f$  thật tốt để khi có một dữ liệu mới  $\mathbf{x}$ , chúng ta có thể dự đoán được nhãn tương ứng của chúng thông qua hàm số  $y = f(\mathbf{x})$ .

<sup>1</sup>Trong luận văn này, mô hình phân loại được biểu diễn dưới dạng hàm số.

Ví dụ:



**Hình 4:** Tập dữ liệu dùng để huấn luyện trong Nhận dạng chữ viết tay[9]

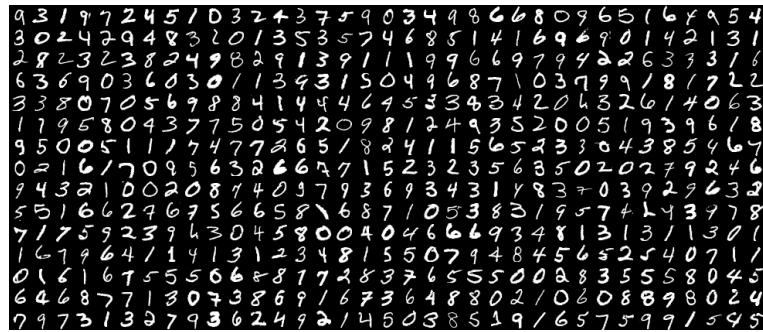
Trong nhận dạng chữ viết tay, ta có ảnh của hàng nghìn ví dụ của mỗi chữ số được viết bởi nhiều người khác nhau. Chúng ta đưa các bức ảnh này vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào. Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số. Khi nhận được một bức ảnh mới, mô hình sẽ dự đoán bức ảnh đó chứa chữ số nào.

## 2.2 Quá trình phân lớp dữ liệu

### 2.2.1 Thu thập dữ liệu

Đây là một bước rất quan trọng trong việc xây dựng mô hình phân loại. Thông qua dữ liệu đầu vào, các thuật toán điều chỉnh các trọng số để biểu diễn quy luật của dữ liệu, từ đó có thể đưa ra dự đoán đối với dữ liệu mới. Dữ liệu đầu vào càng phong phú, chính xác thì mô hình dự đoán sẽ cho kết quả càng tốt. Chính vì vậy, chúng ta cần thu thập dữ liệu từ những nguồn có uy tín.

**Ví dụ:** Với bài toán nhận dạng chữ viết tay, tập dữ liệu đầu vào sẽ là tập ảnh chữ viết tay. Tập dữ liệu nổi tiếng về nhận dạng chữ viết tay là MNIST.



Hình 5: Bộ cơ sở dữ liệu chữ viết tay MNIST[10]

## 2.2.2 Tiết xử lí dữ liệu[11]

Dữ liệu trong thế giới thực rất dễ bị nhiễu, bị thiếu do kích thước khổng lồ của chúng. Do được thu thập từ nhiều nguồn không đồng nhất nên dữ liệu không nhất quán. Chất lượng dữ liệu thấp sẽ làm giảm hiệu suất khai thác dữ liệu. Chính vì vậy, cần có một quá trình tiền xử lí để nâng cao chất lượng dữ liệu. Quá trình này gồm một số kỹ thuật như:

- **Làm sạch dữ liệu<sup>2</sup>:** Giúp loại bỏ nhiễu và chỉnh sửa sự không nhất quán trong dữ liệu bằng việc điền thêm các giá trị bị thiếu, làm mịn các giá trị bị nhiễu, xác định và loại các giá trị bất thường, giải quyết sự thiếu nhất quán trong dữ liệu.
- **Tích hợp dữ liệu<sup>3</sup>:** Kết hợp dữ liệu từ nhiều nguồn thành một kho dữ liệu thống nhất.
- **Thu giảm dữ liệu<sup>4</sup>:** Giảm kích thước dữ liệu nhưng vẫn giữ được những đặc trưng vốn có.
- **Chuyển đổi dữ liệu<sup>5</sup>:** Chuyển đổi dữ liệu về cùng một độ đo.

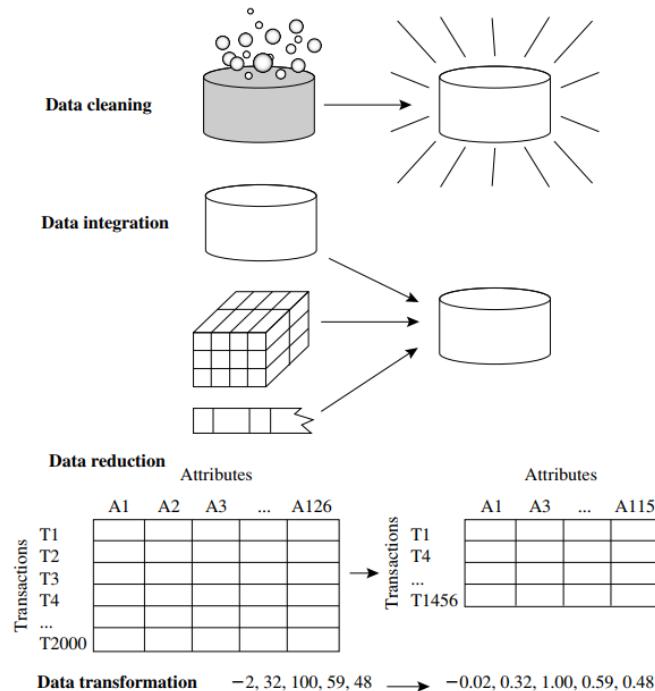
Các kỹ thuật này không loại trừ nhau mà có thể hỗ trợ cho nhau.

<sup>2</sup>Thuật ngữ tiếng Anh: Data cleaning

<sup>3</sup>Thuật ngữ tiếng Anh: Data integration

<sup>4</sup>Thuật ngữ tiếng Anh: Data reduction

<sup>5</sup>Thuật ngữ tiếng Anh: Data Transformation



**Hình 6:** Tổng quan quá trình tiền xử lý dữ liệu[12]

### 2.2.2.1 Làm sạch dữ liệu[13]

- **Dữ liệu bị thiếu**

Trong quá trình thu thập, chắc chắn sẽ có lúc một số phần tử trong dữ liệu bị thiếu mất vài trường. Sau đây là một số giải pháp xử lý vấn đề này.

- *Điền thêm các trường bị thiếu:* Phương pháp này tốn khá nhiều thời gian và đôi khi không khả thi nếu tập dữ liệu có kích thước lớn.

- *Thay thế các trường bị thiếu bằng giá trị trung bình hoặc trung vị:* Đối với phân phối dữ liệu bình thường (đồi xứng), giá trị trung bình sẽ được sử dụng để thay thế cho các trường bị thiếu, trong khi với dữ liệu phân phối lệch chuẩn thì các phần tử trung vị sẽ dùng để thay thế. Ví dụ: Với phân phối dữ liệu liên quan đến thu nhập của khách hàng là đồi xứng và thu nhập bình quân là \$ 56.000. Giá trị \$ 56.000 sẽ được dùng để thay thế cho những dữ liệu khách hàng còn thiếu trường *thu nhập*.

- *Sử dụng giá trị trung bình hoặc trung vị thay cho các trường bị thiếu trong cùng một lớp:* Ví dụ: Nếu phân loại khách hàng theo khả năng rủi ro tín dụng, chúng ta có thể thay thế các giá trị thu nhập còn thiếu bằng giá trị thu nhập trung bình cho tất cả các khách hàng có cùng mức độ rủi ro tín dụng. Nếu phân phối dữ liệu lệch chuẩn thì dùng giá trị trung vị để thay thế.



- Sử dụng giá trị có thể xảy ra nhất để thay thế các giá trị bị thiếu: Ví dụ: có thể sử dụng tập thuộc tính khách hàng có sẵn để xây dựng mô hình dự đoán cho những giá trị bị thiếu.

**Nhận xét:** Trong các phương pháp trên, phương pháp cuối cùng hay được sử dụng nhất. So với các phương pháp khác, phương pháp này sử dụng hầu hết các thông tin từ dữ liệu hiện tại để dự đoán cho các giá trị bị thiếu, như vậy các giá trị được thêm vào sẽ hợp lý và có cơ sở.

#### • Dữ liệu bị nhiễu

Dữ liệu nhiễu là những dữ liệu bị lỗi một cách ngẫu nhiên trong quá trình thu thập.

Ví dụ: Trong giao dịch thẻ tín dụng, hành vi mua hàng của khách hàng được mô tả như một biến ngẫu nhiên. Khách hàng có thể tạo ra một số dữ liệu nhiễu như mua một bữa trưa lớn hơn mọi ngày, hoặc uống một lần vài ba cốc cà phê.

Phương pháp xử lí khi dữ liệu bị nhiễu<sup>6</sup>:

- *Phân khoảng*<sup>7</sup>: Là phương pháp làm mịn dữ liệu bằng cách tham khảo các giá trị xung quanh nó. Các dữ liệu được sắp xếp, phân chia vào các khoảng có tần số xuất hiện như nhau. Sau đó, các khoảng dữ liệu sẽ được biểu diễn bởi các giá trị trung bình, trung vị hoặc các giá trị biên... của các giá trị trong khoảng đó.

**Ví dụ:** Ban đầu chúng ta có 3 *khoảng*, *khoảng 1* có 3 giá trị là 4, 8, 15, *khoảng 2* có các giá trị là 21, 21, 24, *khoảng 3* có các giá trị là 25, 28, 34. Đối với phương pháp làm mịn bằng giá trị trung bình thì các giá trị trong khoảng sẽ được thay thế bằng các giá trị trung bình của các phần tử trong khoảng. Đối với *khoảng 1* là 9, *khoảng 2* là 22, ... Còn với phương pháp làm mịn bằng giá trị biên, thì các phần tử trong *khoảng* sẽ được thay thế bởi các phần tử biên gần nó nhất như trong hình minh họa.

<sup>6</sup>Hay còn gọi là *làm mịn dữ liệu*

<sup>7</sup>Thuật ngữ tiếng Anh: Binning



### Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15

Bin 2: 21, 21, 24

Bin 3: 25, 28, 34

### Smoothing by bin means:

Bin 1: 9, 9, 9

Bin 2: 22, 22, 22

Bin 3: 29, 29, 29

### Smoothing by bin boundaries:

Bin 1: 4, 4, 15

Bin 2: 21, 21, 24

Bin 3: 25, 25, 34

Hình 7: Làm mịn dữ liệu bằng phương pháp phân khoảng

- Phân tích dữ liệu bất thường[14]<sup>8</sup>

**Định nghĩa:** Dữ liệu bất thường là những điểm số liệu nằm cách xa so với phần lớn dữ liệu khác trong tập dữ liệu. Các phần tử bất thường có ảnh hưởng lớn đến độ chính xác của mô hình dự đoán. Chính vì vậy, việc phát hiện và xử lý các phần tử bất thường trước khi áp dụng mô hình dự đoán là điều rất cần thiết.

Dữ liệu bất thường xuất hiện do nhiều nguyên nhân như nhiễu từ môi trường, độ chính xác của thiết bị thu thập dữ liệu hoặc do sự bất cẩn của con người trong quá trình thu thập và tổng hợp dữ liệu...

Có nhiều loại dữ liệu bất thường như dữ liệu bất thường toàn cục<sup>9</sup>, dữ liệu bất thường theo ngoại cảnh<sup>10</sup>, tập dị biệt<sup>11</sup>. Trong đó, dữ liệu bất thường toàn cục có đặc điểm đơn giản nhất và hầu hết các phương pháp phát hiện dữ liệu bất thường đều nhằm phát hiện loại này.

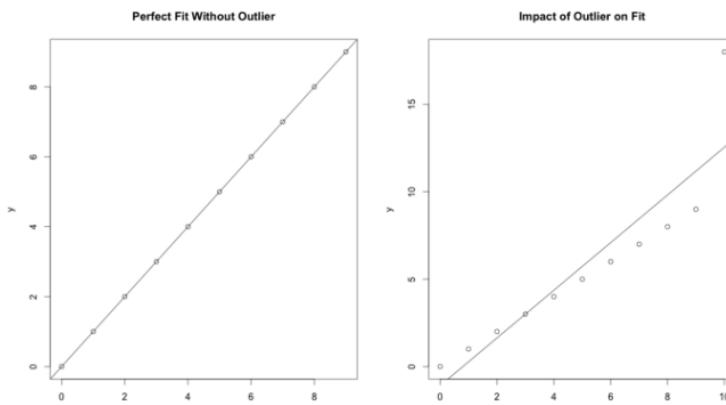
*Dữ liệu bất thường toàn cục:* Một phần tử được xem là điểm dữ liệu bất thường toàn cục nếu đặc điểm của nó khác biệt đáng kể so với phần còn lại của tập dữ liệu. Các

<sup>8</sup>Thuật ngữ tiếng Anh: Outlier analysis

<sup>9</sup>Thuật ngữ tiếng Anh: Global outlier

<sup>10</sup>Thuật ngữ tiếng Anh: Contextual outlier

<sup>11</sup>Thuật ngữ tiếng Anh: Collective outlier



Hình 8: Ảnh hưởng của phần tử ngoại lai đến mô hình dự đoán[15]

phần tử thuộc loại này thường được gọi là điểm dị biệt.

**Ví dụ:** Trong công lĩnh vực an ninh mạng, nếu hành vi giao tiếp của một máy tính khác so với các máy tính còn lại trong hệ thống (ví dụ, một lượng lớn gói tin được gửi đi trong một thời gian ngắn), hành vi này có thể xem như một dữ liệu bất thường toàn cầu và máy tính đó có khả năng cao là bị xâm nhập trái phép.

- Một số phương pháp phát hiện phần tử bất thường

- DBSCAN

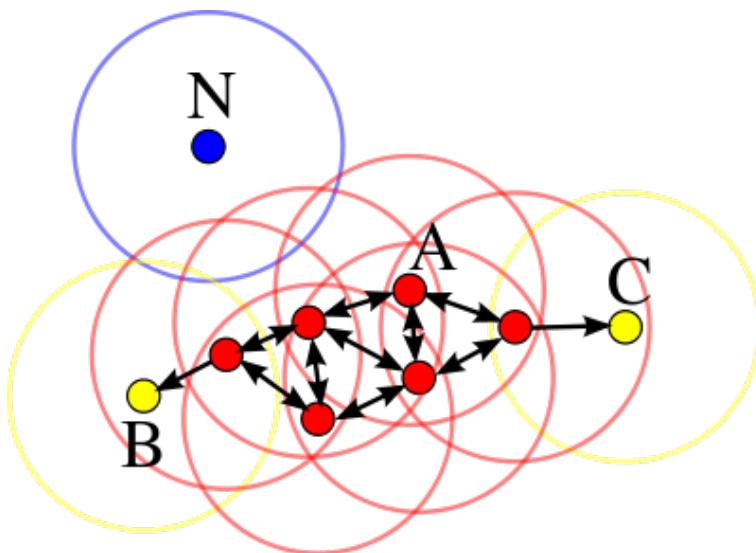
- \* **Ý tưởng**

Đối với mỗi đối tượng của một cụm, láng giềng trong một bán kính cho trước ( $Eps$ ) phải chứa ít nhất một số lượng tối thiểu các đối tượng ( $MinPts$ ). Thuật toán DBSCAN gom cụm dữ liệu thông qua 2 bước: Chọn đối tượng bất kỳ thỏa mãn điều kiện đối tượng lõi làm đối tượng hạt giống. Tìm các đối tượng tối được theo mật độ từ đối tượng hạt giống.

- \* **Thuật toán**

Thuật toán phân cụm dữ liệu dựa DBSCAN kiểm soát thông số  $Eps$  của mỗi điểm dữ liệu. Nếu như số  $Eps$  của một điểm  $p$  chứa nhiều hơn  $MinPts$  thì một cụm mới với điểm  $p$  nòng cốt được thiết lập. Sau đó lặp lại việc tập hợp các đối tượng trực tiếp từ đối tượng nòng cốt này. Thuật toán dừng khi không còn điểm mới nào được thêm vào trong bất kỳ cụm nào.

Sau khi chạy giải thuật trên tập dữ liệu, những cụm nào có số lượng phần tử ít dưới một ngưỡng nào đó thì các phần tử trong cụm đó được xem là phần tử bất thường.



**Hình 9:** Giải thuật DBSCAN[16]

– Dùng biểu đồ Box Plot[17]

\* **Ý tưởng**

Sơ đồ Box Plot là một phương pháp đồ họa điển hình giúp xác định giới hạn trên và dưới mà nếu dữ liệu nằm ngoài giới hạn này thì được xem là dữ liệu bất thường.

\* **Giải thuật**

Xét tập dữ liệu gồm N phần tử.

$$Q_2 \text{ là phần tử trung vị}^{12} \text{ của tập dữ liệu: } Q_2 = \frac{1}{4}(N + 1)$$

$$Q_1 \text{ là tứ phân vị thứ nhất}^{13}: Q_1 = \frac{1}{4}(N + 1)$$

$$Q_3 \text{ là tứ phân vị thứ ba}^{14}: Q_3 = \frac{3}{4}(N + 1)$$

Khoảng giữa hai tứ phân vị<sup>15</sup>, kí hiệu là IQR:  $IQR = Q_3 - Q_1$ ,

$$\text{Khi đó, giới hạn dưới: } LowerLimit = Q_1 - 1.5 * IQR$$

$$\text{Giới hạn trên: } UpperLimit = Up3Up3 + 1.5 * IQR$$

Bất kì giá trị nào nằm ngoài hai giới hạn trên thì bị xem là phần tử bất thường.

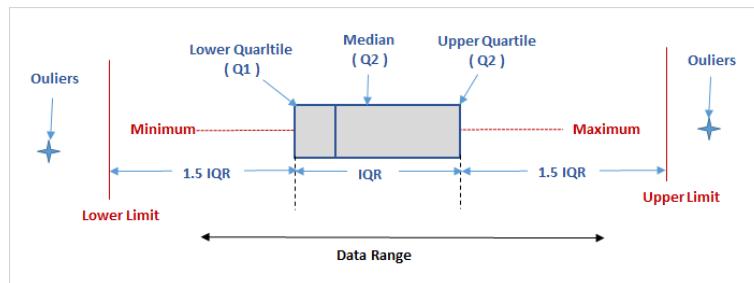
**Ví dụ:** Cho tập dữ liệu gồm N = 90 phần tử: 30, 171, 184, 201, 212, 250, 265, 270, 272, 289, 305, 306, 322, 322, 336, 346, 351, 370, 390, 404, 409, 411, 436, 437, 439, 441, 444, 448, 451, 453, 470, 480, 482, 487, 494, 495, 499, 503, 514, 521, 522, 527, 548, 550, 559, 560, 570, 572, 574, 578, 585, 592, 592, 607, 616,

<sup>12</sup>Thuật ngữ tiếng Anh: Median

<sup>13</sup>Thuật ngữ tiếng Anh: Lower quartile

<sup>14</sup>Thuật ngữ tiếng Anh: Upper quartile

<sup>15</sup>Thuật ngữ tiếng Anh: Interquartile Range



Hình 10: Biểu đồ Box Plot cho ví dụ trên[18]

618, 621, 629, 637, 638, 640, 656, 668, 707, 709, 719, 737, 739, 752, 758, 766, 792, 792, 794, 802, 818, 830, 832, 843, 858, 860, 869, 918, 925, 953, 991, 1000, 1005, 1068, 1441.

Phần tử trung vị  $Q_2$  là phần tử thứ  $\frac{1}{2}(90 + 1) = 45.5$ . Vậy  $Q_2$  chính là trung bình cộng của phần tử thứ 45 và 46,  $Q_2 = \frac{559 + 560}{2} = 559.5$ .

Tương tự, phần tử Q1 là phần tử thứ  $\frac{1}{4}(90 + 1) = 22.75$ . Vậy  $Q_1 = 411 + \frac{436 - 411}{4} = 429.75$ .

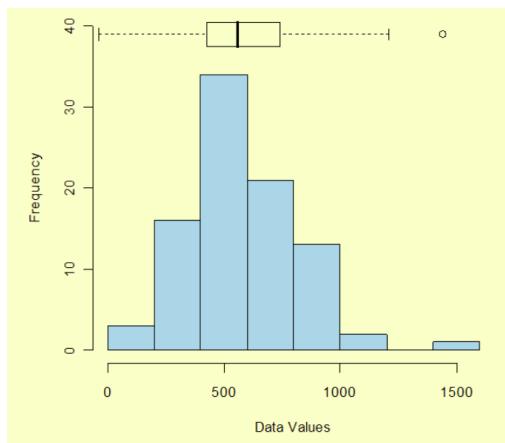
Phần tử Q3 là phần tử thứ  $\frac{3}{4}(90 + 1) = 68.25$ . Vậy  $Q_3 = 739 + \frac{752 - 739}{4} = 742.25$ .

$$IQR = Q_3 - Q_1 = 742.25 - 429.75 = 312.5$$

$$LowerLimit = Q_1 - 1.5 * IQR = 429.75 - 1.5 * 312.5 = -39.0$$

$$UpperLimit = Q_3 + 1.5 * IQR = 742.25 + 1.5 * 312.5 = 1211.0$$

Vậy, phần tử bất thường là: 1441.



Hình 11: Biểu đồ phân bố dữ liệu trong ví dụ[19]



- **Cách xử lí phần tử bất thường:**

- **Delete rows containing outlier:**

Chúng ta sẽ loại bỏ các phần tử bất thường khỏi tập dữ liệu.

- **Change value to mean:**

Các giá trị outlier sẽ được thay bằng giá trị trung bình.

- **Change value to null:**

Xóa giá trị bất thường và đặt lại là null (empty).

- **Change value to specific value:**

Đổi giá trị phần tử bất thường thành một giá trị cụ thể (do người phân tích, chuyên gia đề xuất).

### 2.2.2.2 Tích hợp dữ liệu [20]

Trong công đoạn tiền xử lí dữ liệu thường đòi hỏi phải tích hợp dữ liệu (kết hợp dữ liệu từ nhiều nguồn thành một kho dữ liệu). Việc tích hợp dữ liệu chính xác có thể giúp giảm thiểu, tránh dư thừa dữ liệu và sự không nhất quán trong tập dữ liệu kết quả.

- **Phân tích tương quan<sup>16</sup>**

Sự dư thừa dữ liệu là một vấn đề quan trọng trong tích hợp dữ liệu. Một thuộc tính có thể bị dư thừa nếu nó có thể được suy ra từ những thuộc tính khác. Ví dụ, thuộc tính doanh thu hàng năm có thể được suy ra từ thuộc tính doanh thu hàng tháng.

- **$\chi^2$  Kiểm tra tương quan đối với dữ liệu định danh<sup>17</sup>**

Đối với dữ liệu định danh, một mối quan hệ tương quan giữa hai thuộc tính, A và B, có thể được phát hiện bằng một bài kiểm tra  $\chi^2$  chi-square. Giả sử A có c giá trị khác biệt, cụ thể là  $a_1, a_2, \dots, a_c$ . B có r giá trị khác biệt, cụ thể là  $b_1, b_2, \dots, b_r$ . Cho  $(A_i, B_j)$  biểu thị sự kiện chung mà thuộc tính A lấy giá trị  $a_i$  và thuộc tính B lấy giá trị  $b_j$ , đó là  $(A = a_i, B = b_j)$ . Khí đó, giá trị  $\chi^2$  được tính theo công thức:

$$\chi^2 = \sum_{i=1}^c \sum_{j=1}^r \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

trong đó:

$o_{ij}$  là tần số thực tế của  $(A_i, B_j)$

$e_{ij}$  là tần số mong đợi của  $(A_i, B_j)$

$$e_{ij} = \frac{\text{count}(A = a_i) * \text{count}(B = b_j)}{n}$$

<sup>16</sup>Thuật ngữ tiếng Anh: Correlation analysis

<sup>17</sup>Thuật ngữ tiếng Anh: Correlation Test for Nominal Data



với:

$$n = c * r$$

$count(A = a_i)$  là số lần  $A = a_i$  xuất hiện,

$count(B = b_j)$  là số lần  $B = b_j$  xuất hiện.

Chỉ số *chi-square* cho biết 2 thuộc tính  $A$  và  $B$  có độc lập với nhau hay không.

Chỉ số  $\chi^2$  với mức độ tự do  $(c - 1) * (r - 1)$  được thể hiện trong bảng phân phối  $\chi^2$  trong các sách xác suất thống kê. Nếu chỉ số này lớn hơn chỉ số chúng ta tính được thì kết luận 2 thuộc tính  $A$  và  $B$  độc lập.

### • Vấn đề xung đột dữ liệu

Tích hợp dữ liệu cũng bao gồm việc phát hiện và giải quyết sự xung đột dữ liệu.

Sự xung đột dữ liệu có thể bắt nguồn từ việc thu thập dữ liệu từ nhiều nguồn mà dữ liệu tại mỗi nguồn lại được biểu diễn theo một cách riêng, dẫn đến không nhất quán. Ngoài ra, còn một số lý do khác như dữ liệu được thiết kế kém, có nhiều trường tùy chọn, lỗi nhập liệu, lỗi cố ý (ví dụ: người dùng không muốn tiết lộ về thông tin cá nhân của họ) và sự phân rã dữ liệu.

Vậy, làm thế nào để phát hiện dữ liệu bị xung đột? Vấn đề này phụ thuộc nhiều vào người thu thập và xử lý dữ liệu. Khi thu thập và xử lý dữ liệu, cần để ý một số vấn đề như: loại dữ liệu và tên miền của mỗi thuộc tính là gì? Miền giá trị của mỗi thuộc tính. Chúng ta cũng có thể phát hiện xu hướng của dữ liệu thông qua các chỉ số như giá trị trung bình, trung vị, ... Độ lệch tiêu chuẩn của mỗi thuộc tính là gì? Ngoài ra, vì dữ liệu được thu thập từ nhiều nguồn khác nhau nên chúng có thể được biểu diễn theo các cách khác nhau. Ví dụ: ngày giờ ở Việt Nam được biểu diễn theo dạng  $dd/mm/yyyy$  nhưng ở Mỹ, ngày giờ lại được lưu trữ theo định dạng  $mm/dd/yyyy$ . Chính vì vậy, khi xử lý dữ liệu, chúng ta cũng cần để ý dữ liệu về cùng một định dạng để tránh xung đột dữ liệu.

#### 2.2.2.3 Thu giảm dữ liệu[21]

Việc phân tích và khai thác dữ liệu phức tạp với lượng dữ liệu lớn sẽ mất rất nhiều thời gian, làm cho việc phân tích trở nên khó khăn và đòi hỏi khả thi. Kĩ thuật thu giảm dữ liệu có thể được áp dụng để có được một tập dữ liệu mới với kích thước nhỏ hơn nhiều nhưng vẫn giữ được những đặc trưng của tập dữ liệu ban đầu. Nghĩa là việc khai thác trên tập dữ liệu mới sẽ cho kết quả tương tự như khi phân tích trên tập dữ liệu ban đầu với thời gian nhanh hơn và cách tiến hành dễ dàng hơn.

Các chiến lược thu giảm dữ liệu bao gồm: thu giảm kích thước, thu giảm số lượng và nén dữ liệu. Trong đó, phương pháp phân tích thành phần chính sẽ đại diện cho chiến lược giảm số lượng dữ liệu.



- Phân tích thành phần chính (PCA)<sup>[22]</sup><sup>18</sup>

- **Ý tưởng**

Trong thực tế, số lượng thuộc tính của dữ liệu có thể rất lớn, tới vài nghìn. Ngoài ra, số lượng điểm dữ liệu cũng rất nhiều. Vì vậy, việc giảm số chiều dữ liệu là một bước quan trọng trong nhiều bài toán phân tích dữ liệu.

Mục tiêu của thu giảm số chiều dữ liệu, là tìm một hàm số lấy đầu vào là một điểm dữ liệu ban đầu  $x \in R^D$  với  $D$  rất lớn, và tạo ra một điểm dữ liệu mới  $z \in R^K$  với  $K < D$

Cách đơn giản nhất để thu giảm dữ liệu từ  $D$  chiều sang  $K$  chiều với  $K < D$  là chỉ giữ lại  $K$  thuộc tính quan trọng nhất. Tuy nhiên, ban đầu, chúng ta không biết được đâu là thuộc tính quan trọng, hoặc có thể các thuộc tính đều quan trọng như nhau thì việc bỏ đi một lượng các thuộc tính sẽ làm mất đặc trưng của dữ liệu.

Chính vì vậy, phương pháp PCA sẽ tìm ra một hệ cơ sở mới sao cho thông tin của dữ liệu chủ yếu tập trung ở một vài chiều.

- **Giải thuật**

Tính vector kì vọng của toàn bộ dữ liệu:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n$$

trừ mỗi điểm dữ liệu cho vector kì vọng của toàn bộ dữ liệu

$$\hat{x} = x_n - \bar{x}$$

tính ma trận hiệp phương sai

$$S = \frac{1}{N} \hat{X} \hat{X}^T$$

Tính các trị riêng và vector riêng có *norm* bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.

Chọn  $K$  vector riêng ứng với  $K$  trị riêng lớn nhất để xây dựng ma trận  $U_K$  có các cột tạo thành một hệ trực giao.  $K$  vectors này, còn được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hóa.

Chiều dữ liệu ban đầu đã chuẩn hóa  $\hat{X}$  xuông không gian con tìm được.

Dữ liệu mới chính là toạ độ của các điểm dữ liệu trên không gian mới.

$$Z = U_K^T \hat{X}$$

Dữ liệu ban đầu có thể tính được xấp xỉ theo dữ liệu mới như sau:

$$x \approx U_K Z + \bar{x}$$

<sup>18</sup>Thuật ngữ tiếng Anh: Principal Component Analysis



#### 2.2.2.4 Chuyển đổi dữ liệu[23]

Các kĩ thuật của việc *Chuyển đổi dữ liệu* bao gồm:

- *Làm mịn*: Loại bỏ nhiễu trong dữ liệu. Bao gồm các kĩ thuật hồi quy, phân khoảng (binning), gom cụm.
- *Xây dựng thuộc tính*: Xây dựng thuộc tính đại diện cho đặc trưng của tập dữ liệu, giúp rút trích tối đa thông tin.
- *Tổng hợp dữ liệu*: Từ những dữ liệu cụ thể, dữ liệu được khái quát lên mức trừu tượng để tiện cho việc khai thác. Ví dụ, từ dữ liệu doanh thu hàng tháng, có thể tổng hợp thành doanh thu hàng năm để đánh giá mức độ phát triển của doanh nghiệp.
- *Chuẩn hóa dữ liệu*: giá trị của dữ liệu được thu nhỏ về một thang đo nhất định. Ví dụ, thang điểm của các trường là khác nhau, có trường dùng thang điểm 10, có trường dùng thang điểm 100. Chính vì vậy, cần đưa chúng về cùng một thang đo để tiện xử lí.

#### Chuẩn hóa dữ liệu

Các đơn vị được dùng có thể ảnh hưởng đến kết quả phân tích dữ liệu. Ví dụ, cùng một đơn vị khối lượng, nhưng nếu ta thay đổi các đơn vị đo từ tấn, tạ, kilogram, ... thì sẽ cho ra các kết quả phân tích khác nhau. Để tránh việc phụ thuộc vào các đơn vị đo khác nhau cho cùng một thuộc tính, dữ liệu trước khi phân tích phải được chuẩn hóa. Phổ biến nhất là chuyển đổi dữ liệu về các thang đo như [0.0, 1.0] hoặc [-1.0, 1.0]. Việc chuẩn hóa dữ liệu đặc biệt hữu ích cho các thuật toán liên quan đến mạng nơ-ron hoặc gom cụm, ...

Có nhiều phương pháp chuẩn hóa như *chuẩn hóa dạng min-max*, *chuẩn hóa z-score*, *chuẩn hóa tỉ lệ thập phân*...

- *Chuẩn hóa min-max*

Thực hiện việc chuẩn hóa tuyến tính dữ liệu gốc. Giả sử  $\min A$  và  $\max A$  là giá trị nhỏ nhất và giá trị lớn nhất của thuộc tính A. *Chuẩn hóa min-max* sẽ chuyển đổi giá trị  $v$  của A sang giá trị  $v'$  nằm trong khoảng  $[\text{new\_min}A, \text{new\_max}A]$  theo công thức:

$$v' = \frac{v - \min A}{\max A - \min A} (\text{new\_max}A - \text{new\_min}A) + \text{new\_min}A$$

Chuẩn hóa min-max bảo đảm mối quan hệ giữa các giá trị của dữ liệu gốc. Nó sẽ phát hiện ra được các lỗi “vượt quá giới hạn” nếu các dữ liệu input trong tương lai rơi ra ngoài khoảng giá trị ban đầu của A.

- *Chuẩn hóa z-score*

Trong *chuẩn hóa z-score*, các giá trị của thuộc tính A sẽ được chuẩn hóa dựa trên giá trị trung bình và độ lệch chuẩn của A. Một giá trị  $v$  của A sẽ được chuẩn hóa thành giá trị  $v'$  bằng công thức sau:



$$v' = \frac{v - \bar{A}}{\sigma_A}$$

trong đó,

$\bar{A}$  là giá trị trung bình của A.

$\sigma_A$  là độ lệch chuẩn của A.

**Nhận xét:** Phương pháp chuẩn hóa này hữu ích trong trường hợp ta không biết được giá trị nhỏ nhất và lớn nhất của A, hoặc khi có những giá trị bất thường trong dữ liệu.

- *Chuẩn hóa bằng tỉ lệ thập phân*

Theo phương pháp này, giá trị  $v$  sẽ được chuẩn hóa bằng cách dời dấu phẩy thập phân của nó, dựa vào giá trị tuyệt đối lớn nhất của A. Một giá trị  $v$  sẽ được chuẩn hóa thành  $v'$  bởi công thức sau:

$$v' = \frac{v}{10^j}$$

trong đó,

$j$  là giá trị nhỏ nhất sao cho  $\text{Max}(|v'|) < 1$

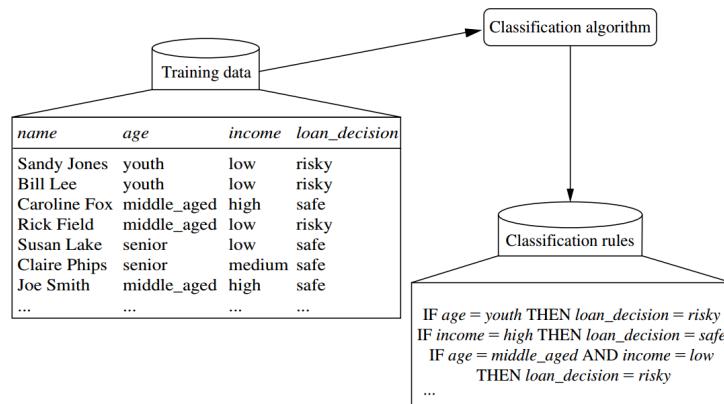
**Ví dụ:** Giả sử giá trị của thuộc tính A nằm trong khoảng từ -986 đến 917. Giá trị tuyệt đối lớn nhất của A là 986. Vậy  $j=3$  (vì  $102 < 986 < 103$ ). Giá trị 986 sẽ được chuẩn hóa thành -0.986 và 917 được chuẩn hóa thành 0.917.

**Lưu ý:** Việc chuẩn hóa có thể làm thay đổi chút ít giá trị ban đầu, đặc biệt là ở 2 phương pháp sau. Chúng ta cũng cần lưu lại các tham số chuẩn hóa (ví dụ như giá trị trung bình và độ lệch chuẩn nếu dùng chuẩn hóa *z-score*) để các dữ liệu được bổ sung thêm sau này cũng được chuẩn hóa theo cùng công thức.

### 2.2.3 Xây dựng mô hình phân lớp[24]

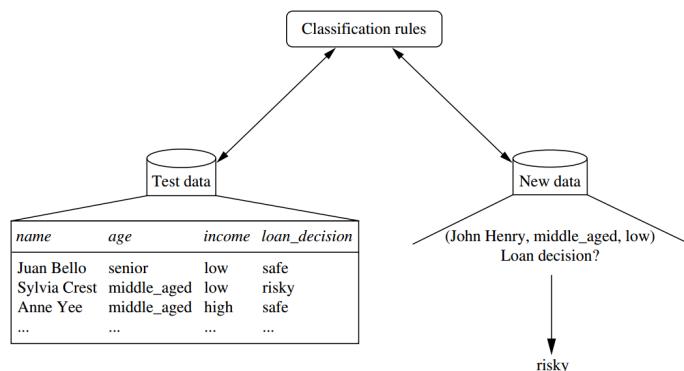
Phân loại dữ liệu là một quá trình gồm 2 bước, bao gồm bước xây dựng mô hình phân loại và phân loại (sử dụng mô hình để dự đoán *nhãn* cho tập dữ liệu).

Trong bước đầu tiên, bộ phân loại được xây dựng để mô tả tập dữ liệu đã được gán *nhãn* trước. Trong đó, một thuật toán phân lớp xây dựng bộ phân loại bằng cách phân tích tập dữ liệu huấn luyện gồm các điểm dữ liệu và các *nhãn* tương ứng. Mỗi điểm dữ liệu được đại diện bởi một vector có  $n$  thuộc tính. Tương ứng với các điểm dữ liệu này là các *nhãn* đã được gán trước. Thuật toán phân lớp sẽ cố gắng ánh xạ mỗi điểm dữ liệu sang *nhãn* tương ứng của chúng một cách chính xác nhất có thể.



**Hình 12:** Quá trình xây dựng một bộ phân loại[25]

Trong bước thứ 2, mô hình sau khi được huấn luyện sẽ dùng để phân loại. Một vài chỉ số sẽ được sử dụng để đánh giá mức độ phân loại chính xác của mô hình. Để khách quan, tập dữ liệu dùng để đánh giá mô hình phải độc lập với tập dữ liệu huấn luyện. Chúng được gọi là tập đánh giá. Thông thường, từ tập dữ liệu ban đầu sẽ chia thành 2 tập là tập huấn luyện và tập đánh giá. Trong đó, tập dữ liệu huấn luyện chiếm hai phần ba tập dữ liệu ban đầu, còn lại là tập đánh giá.



**Hình 13:** Phân loại đối với dữ liệu mới[26]

## 2.2.4 Vấn đề overfitting[27]

### 2.2.4.1 Định nghĩa

*Overfitting* là hiện tượng khi mô hình đạt hiệu suất cao trên tập dữ liệu huấn luyện nhưng lại kém chính xác khi áp dụng trên tập dữ liệu đánh giá.



#### 2.2.4.2 Nguyên nhân

*Overfitting* xảy ra có thể do một số nguyên nhân phổ biến như: tập dữ liệu huấn luyện bị nhiễu, dữ liệu huấn luyện không bao quát tất cả các trường hợp của vấn đề hoặc có thể do mô hình phân loại quá phức tạp với quá nhiều thông số so với lượng dữ liệu quan sát được.

#### 2.2.4.3 Phương pháp tránh overfitting

- **Xác nhận chéo<sup>19</sup>**

Fương pháp này được sử dụng khi tập dữ liệu quá nhỏ. Lúc đây, tập xác nhận sẽ rất nhỏ dẫn đến *overfitting* trên cả tập huấn luyện và tập xác nhận.

Chia tập huấn luyện ra thành  $k$  tập con không có phần tử chung, kích thước gần bằng nhau. Tại mỗi lần tối ưu mô hình, được gọi là *run*, một trong  $k$  tập con sẽ được sử dụng làm tập xác nhận, còn lại là tập huấn luyện. Mô hình cuối cùng được đánh giá dựa trên trung bình các lần đánh giá ứng với mỗi *run*.

- **Early stopping**

Trong các bài toán tối ưu hàm chi phí, sau mỗi lần lặp thì giá trị hàm chi phí sẽ giảm dần. Phương pháp này giúp dừng vòng lặp trước khi hàm chi phí đạt giá trị quá nhỏ, giúp tránh *overfitting*.

Sau mỗi lần lặp, các chỉ số lỗi nhằm đánh giá hiệu suất mô hình sẽ được tính. Vòng lặp sẽ được thực hiện đến khi nào các chỉ số lỗi này đạt cực tiểu và bắt đầu có xu hướng tăng dần.

#### 2.2.5 Đánh giá mô hình[28]

Có nhiều chỉ số để đánh giá độ chính xác của một mô hình phân loại. Sau đây, chúng tôi sẽ trình bày một vài chỉ số đánh giá phổ biến.

Measure	Formula
accuracy, recognition rate	$\frac{TP+TN}{P+N}$
error rate, misclassification rate	$\frac{FP+FN}{P+N}$
sensitivity, true positive rate, recall	$\frac{TP}{P}$
specificity, true negative rate	$\frac{TN}{N}$
precision	$\frac{TP}{TP+FP}$
$F, F_1, F$ -score, harmonic mean of precision and recall	$\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$
$F_\beta$ , where $\beta$ is a non-negative real number	$\frac{(1+\beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}$

**Hình 14:** Các chỉ số đánh giá độ chính xác hệ thống phân loại[29]

<sup>19</sup>Thuật ngữ tiếng Anh: Cross-validation

Trong đó,

*TP*: Số lượng các ví dụ thuộc lớp *y* được phân loại chính xác vào lớp *y*

*FP*: Số lượng các ví dụ không thuộc lớp *y* bị phân loại nhầm vào lớp *y*

*TN*: Số lượng các ví dụ không thuộc lớp *y* được phân loại chính xác

*FN*: Số lượng các ví dụ thuộc lớp *y* bị phân loại nhầm

$P = TP + FN$ : số lượng ví dụ thuộc lớp *y* được phân loại bởi hệ thống

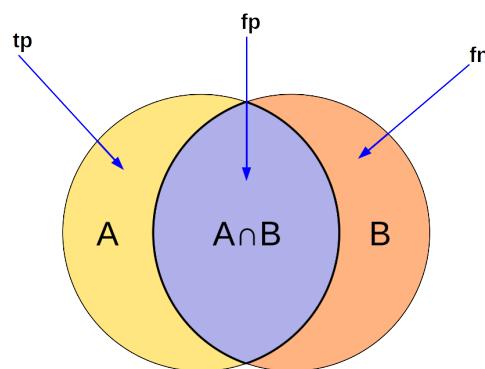
$N = FP + TN$ : số lượng ví dụ không thuộc lớp *y* được phân loại bởi hệ thống

### Dộ đo Jaccard

Trong luận văn này, chỉ số Jaccard được sử dụng để đánh giá hiệu suất của mô hình dự đoán.

Dộ đo Jaccard đo lường sự tương đồng giữa các bộ mẫu hữu hạn.

Dộ đo Jaccard của hai tập A và B được tính như sau:



**Hình 15:** Minh họa độ đo Jaccard

$$Jaccard_{A,B} = \frac{TP}{TP + FP + FN} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (1)$$

với:

*A*: là tập kết quả dự đoán của hệ thống.

*B*: tập kết quả chính xác.

$|A \cap B|$ : Số lượng kết quả mà hệ thống phân loại chính xác.

Nếu A và B điều rỗng,  $J(A, B) = 1$ .

$0 \leq J(A, B) \leq 1$

Dộ đo Jaccard càng tiến tới 1 thì kết quả dự đoán được càng tốt, càng gần kết quả thực tế.

**Nhận xét:** Trên đây là một vài chỉ số thông dụng để đánh giá hiệu suất của các mô hình phân loại. Việc đánh giá hiệu suất hệ thống thông qua các chỉ số trên chỉ thật sự có hiệu quả khi tập dữ liệu huấn luyện và tập dữ liệu đánh giá phải độc lập với nhau. Hơn nữa, việc phân bổ các phần tử thuộc từng lớp phải thật đồng đều thì quá trình huấn luyện mới thật sự cho kết quả tốt.

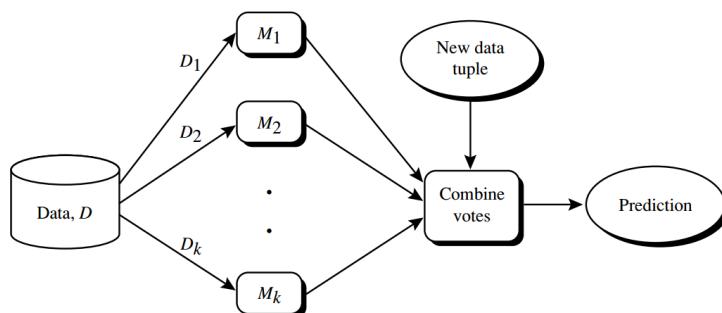
### 2.2.6 Phương pháp cải tiến hiệu suất phân loại[30]

Phương pháp kết hợp các bộ phân loại sẽ được đề cập trong phần này như là một giải pháp để nâng cao hiệu suất quá trình phân lớp.

#### 2.2.6.1 Khái niệm kết hợp các bộ phân loại

Bộ kết hợp các bộ phân loại (Ensemble) là tập hợp của các bộ phân loại cơ bản, trong đó mỗi bộ phân loại cơ bản có thể là một mô hình phân loại như: *cây quyết định, mạng bayes, mạng nơ-ron,...* Khi một phần tử mới được phân loại, nó được xử lý bởi các bộ phân loại cơ bản. Kết quả cuối cùng được kết hợp theo một cách nào đó từ kết quả của các bộ phân loại cơ bản. Để có một bộ phân loại kết hợp tốt, thì các bộ phân loại cơ bản phải có độ tương quan thấp.

Trong đó, *bagging*, *booting* là những giải thuật phổ biến cho kiểu phân loại này.



Hình 16: Mô hình phân loại hỗn hợp[31]

#### 2.2.6.2 Các cách tiếp cận phương pháp kết hợp các bộ phân loại

Có 2 cách để xây dựng một bộ phân loại kết hợp: thứ nhất là xây dựng mỗi bộ phân loại một cách độc lập với nhau, sau đó sử dụng phương pháp biểu quyết để chọn ra kết quả cuối cùng của bộ kết hợp. Tức là mỗi bộ phân loại cơ bản sẽ được xây dựng độc lập với các bộ phân loại khác bằng cách thay đổi tập dữ liệu huấn luyện đầu vào, thay đổi các đặc trưng trong tập huấn luyện. Thứ hai là xây dựng các bộ phân loại cơ bản và gán trọng số cho các kết quả của mỗi bộ phân loại. Việc lựa chọn một bộ phân loại cơ bản ảnh hưởng tới việc lựa chọn của các bộ phân loại cơ bản khác và trọng số được gán cho chúng. Trong đó, *bagging* đại diện cho phương pháp thứ nhất, *booting* đại diện cho phương pháp thứ 2.



## • Phương pháp Bagging

### – Mô hình hoạt động

*Bagging* tạo ra các bộ phân loại từ các tập mẫu con có lặp từ tập mẫu ban đầu và một thuật toán học máy, mỗi tập mẫu sẽ tạo ra một bộ phân loại cơ bản. Các bộ phân loại sẽ được kết hợp bằng phương pháp biểu quyết theo số đông. Tức là khi có một ví dụ cần được phân loại, mỗi bộ phân loại sẽ cho ra một kết quả. Và kết quả nào xuất hiện nhiều nhất sẽ được lấy làm kết quả của bộ kết hợp.

### – Giải thuật

*Bagging* tạo ra N tập huấn luyện được chọn có lặp từ tập dữ liệu huấn luyện ban đầu. Trong đó, các ví dụ huấn luyện có thể được chọn hơn một lần hoặc không được chọn lần nào. Từ mỗi tập huấn luyện mới, *Bagging* cho chạy với một thuật toán học máy  $L_b$  để sinh ra M bộ phân loại cơ bản  $h_m$ . Khi có một phần tử phân loại mới, kết quả của bộ kết hợp sẽ là kết quả nhận được nhiều nhất khi chạy M bộ phân loại cơ bản.

## • Phương pháp Booting

### – Mô hình hoạt động

Khác với phương pháp *Bagging*, xây dựng bộ phân loại kết hợp với các ví dụ huấn luyện có trọng số bằng nhau, phương pháp *Boosting* xây dựng bộ phân loại kết hợp với các ví dụ huấn luyện có trọng số khác nhau. Sau mỗi bước lặp, các ví dụ huấn luyện được dự đoán sai sẽ được đánh trọng số tăng lên, các ví dụ huấn luyện được dự đoán đúng sẽ được đánh trọng số nhỏ hơn. Điều này giúp cho *Boosting* tập trung vào cải thiện độ chính xác cho các ví dụ được dự đoán sai sau mỗi bước lặp.

### – Giải thuật

Một thuật toán *boosting* ban đầu được định nghĩa là một thuật toán dùng để chuyển một thuật toán học máy yếu thành một thuật toán học máy mạnh. Có nghĩa là nó chuyển một thuật toán học máy giải quyết một bài toán phân loại 2 lớp tốt hơn cách giải chọn ngẫu nhiên thành một thuật toán giải quyết rất tốt bài toán đó. Thuật toán *boosting* ban đầu của Schapire là một thuật toán đệ quy. Tại bước cuối của đệ quy, nó kết hợp các giả thuyết được tạo bởi thuật toán học máy yếu. Xác suất lỗi của bộ kết hợp này được chứng minh là nhỏ hơn xác suất lỗi của các giả thuyết yếu. *Adaboost* là một thuật toán kết hợp một tập các bộ phân loại được làm đa dạng bằng việc chạy thuật toán học máy với phân bố khác nhau trên tập huấn luyện.



– **Thuật toán Adaboost**

$Adaboost((x_1, y_1), \dots, (x_N, y_N), L_b, M)$

Khởi tạo  $D_1(n) = \frac{1}{N}$  cho N dữ liệu huấn luyện

Với  $m = 1, 2, \dots, M$

Tạo hàm giả thuyết  $h_m = L_b((x_1, y_1), \dots, (x_N, y_N), D_m)$ .

Tính toán lỗi của  $h_m : \epsilon_m = \sum_{n:h_m(x_n) \neq y_n} D_m(n)$

Nếu  $\epsilon_m \geq \frac{1}{2}$  thì gán  $M = m - 1$  và thoát khỏi vòng lặp

Cập nhật phân bố  $D_m$ :

$$\begin{cases} \frac{1}{2 * (1 - \epsilon_m)}, \text{ nếu } h_m(n) = y_n; \\ \frac{1}{2 * \epsilon_m}, \text{ trường hợp còn lại.} \end{cases}$$

Trả về: giả thuyết cuối cùng

$$h(x) = \operatorname{argmax}_{y \in Y} \sum_{m:h_m(x)=y} \log \frac{1 - \epsilon_m}{\epsilon_m}$$

**Nhận xét:** Phương pháp *bagging* xây dựng bộ phân loại kết hợp với việc đánh trọng số cho các mô hình cơ bản là nhau, trong khi phương pháp *boosting* đánh trọng số cho các mô hình cơ bản là khác nhau. Sau mỗi vòng lặp, các mô hình cơ bản phân loại sai sẽ được đánh trọng số cao lên. Việc này giúp giải thuật *boosting* tập trung cải thiện độ chính xác của các giai thuật phân loại sai.

## 2.3 Các thuật toán phân lớp dữ liệu

### 2.3.1 Support Vector Machine (SVM)[32]

#### 2.3.1.1 Giới thiệu

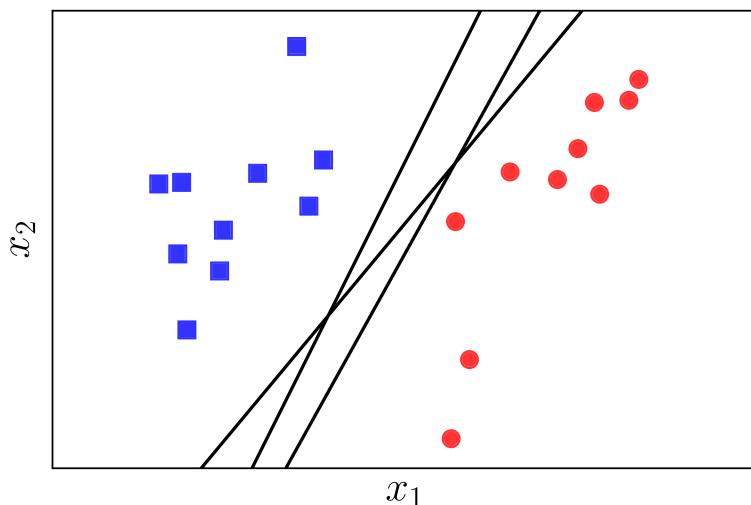
Bài toán phân lớp (*Classification*) và dự đoán (*Prediction*) là hai bài toán cơ bản và có rất nhiều ứng dụng trong tất cả các lĩnh vực như: học máy, nhận dạng, trí tuệ nhân tạo, v.v... Trong phần này, chúng tôi sẽ đi sâu trình bày phương pháp Support Vector Machines (SVM), một trong những phương pháp phân loại hiệu quả nhất hiện nay.

Phương pháp SVM được coi là công cụ mạnh cho các bài toán phân lớp phi tuyến được tác giả Vapnik và Chervonenkis phát triển mạnh mẽ năm 1995. Phương pháp này thực hiện việc phân lớp dựa trên nguyên lý cực tiểu hóa rủi ro có cấu trúc SRM (*Structural Risk Minimization*), được xem là một trong các phương pháp phân lớp giám sát không tham số tinh vi nhất cho đến nay. Các hàm công cụ đa dạng của SVM cho phép tạo không gian chuyển đổi để xây dựng mặt phẳng phân lớp.

#### 2.3.1.2 Ý tưởng của giải thuật

Cho trước một tập huấn luyện được biểu diễn trong không gian vector, ở đó, mỗi một phần tử là một điểm. Phương pháp SVM sẽ tìm ra một siêu mặt phẳng tốt nhất chia các điểm trong không gian thành 2 lớp riêng biệt tương ứng là "+" và "-". Độ tốt của siêu mặt phẳng được quyết định bởi khoảng cách của điểm dữ liệu gần nhất của mỗi lớp đến siêu mặt phẳng này, gọi là khoảng cách biên (margin). Hai khoảng cách này càng lớn thì siêu mặt phẳng quyết định càng tốt, đồng thời việc phân loại sẽ càng chính xác.

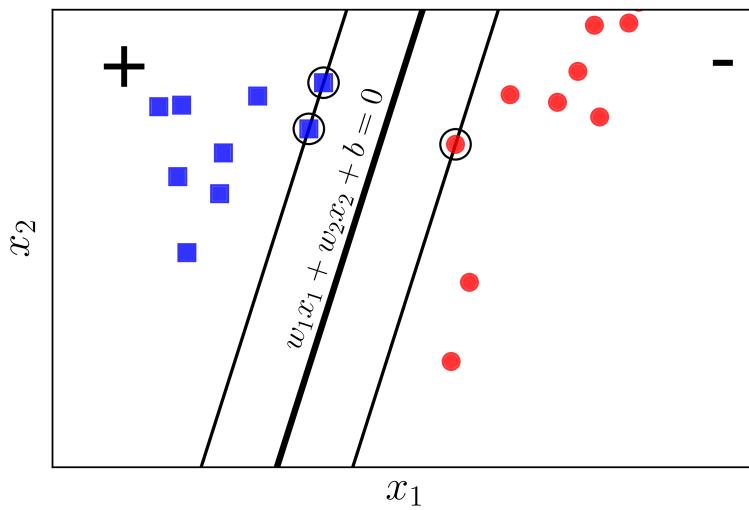
Mục đích của SVM là tìm khoảng cách biên cực đại của từng lớp.



Hình 17: Các mặt phân cách hai lớp dữ liệu [33]

### 2.3.1.3 Xây dựng bài toán tối ưu cho Support Vector Machine

Giả sử rằng các cặp dữ liệu của *training set* là  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$  với vector  $x_i \in R^d$  thể hiện đầu vào của một điểm dữ liệu và  $y_i$  là *nhãn* của điểm dữ liệu đó.  $d$  là số chiều của điểm dữ liệu và  $N$  là số điểm dữ liệu. Giả sử rằng *nhãn* của mỗi điểm dữ liệu được xác định bởi  $y_i = 1$  (class 1) hoặc  $y_i = -1$  (class 2).



**Hình 18:** Phân tích bài toán SVM[34]

Giả sử rằng các điểm vuông xanh thuộc *class 1*, các điểm tròn đỏ thuộc *class -1* và mặt  $w^T X + b = w_1 x_1 + w_2 x_2 + b = 0$  là mặt phân cách giữa hai lớp. Nhiệm vụ của chúng ta là tìm các hệ số  $w$  và  $b$  sao cho khoảng cách biên là lớn nhất.

Với cặp dữ liệu  $(x_n, y_n)$  bất kì, khoảng cách từ điểm đó tới mặt phân chia là:

$$\frac{y_n(w^T X_n + b)}{\|w\|_2}$$

Khi đó, *margin* được tính là khoảng cách gần nhất từ 1 điểm tới mặt đó (bất kể điểm nào trong hai lớp):

$$margin = \min_n \frac{y_n(w^T X_n + b)}{\|w\|_2}$$

Bài toán tối ưu trong SVM chính là bài toán tìm  $w$  và  $b$  sao cho *margin* này đạt giá trị lớn nhất:

$$(w, b) = \operatorname{argmax}_{w,b} \left\{ \min_n \frac{y_n(w^T X_n + b)}{\|w\|_2} \right\} \quad (2)$$

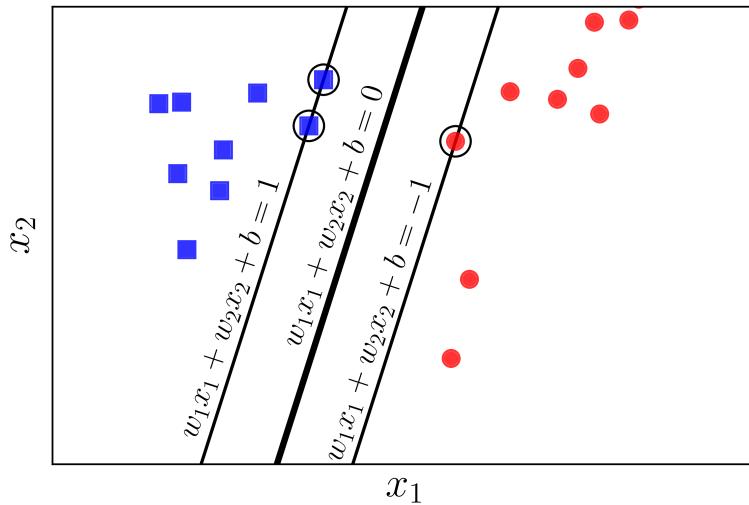
$$= \operatorname{argmax}_{w,b} \left\{ \frac{1}{\|w\|_2} \min_n y_n(w^T X_n + b) \right\} \quad (3)$$

Việc giải trực tiếp bài toán này sẽ rất phức tạp, nên bài toán này sẽ được đưa về dạng đơn giản hơn.

Nếu thay vector hệ số  $w$  bởi  $k * w$  và  $b$  bởi  $k * b$  trong đó  $k$  là hằng số dương thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức *margin* không đổi. Dựa trên tính chất này, có thể giả sử:

$$y_n(w^T X_n + b) = 1$$

với những điểm nằm gần mặt phân chia nhất.



**Hình 19:** Các điểm gần mặt phân cách nhất của hai lớp được khoanh tròn [35]

Như vậy, với mọi  $n$ , ta có:

$$y_n(w^T X_n + b) \geq 1$$

Vậy, bài toán tối ưu (3) có thể đưa về bài toán tối ưu có ràng buộc sau:

$$(w, b) = \operatorname{argmax}_{w,b} \frac{1}{\|w\|_2}$$

với điều kiện:

$$y_n(w^T X_n + b) \geq 1, \forall n = 1, 2, 3, \dots, N$$

Lấy nghịch đảo hàm mục tiêu, bình phương nó để được một hàm khả vi, và nhân với  $\frac{1}{2}$  để biểu thức đạo hàm trở nên hơn.

$$(w, b) = \operatorname{argmax}_{w,b} \frac{1}{2} \|w\|_2^2 \quad (4)$$



với điều kiện:

$$1 - y_n(w^T X_n + b) \leq 0, \forall n = 1, 2, 3, \dots, N \quad (5)$$

Trong bài toán (4), hàm mục tiêu là một *norm*, nên là một *hàm lỗi*. Các hàm bất đẳng thức ràng buộc là các hàm tuyến tính theo  $w$  và  $b$  nên chúng cũng là các *hàm lỗi*. Vậy bài toán tối ưu (4) có hàm mục tiêu là *lỗi*, và các hàm ràng buộc cũng là *lỗi*, nên nó là một bài toán *lỗi*. Hơn nữa, nó là một *Quadratic Programming*. Thậm chí, hàm mục tiêu là *strictly convex* vì  $\|w\|_2^2 = w^T I w$  và  $I$  là ma trận đơn vị - một ma trận xác định dương. Từ đây, có thể suy ra nghiệm của SVM là duy nhất.

Đến đây thì bài toán này có thể giải được bằng các công cụ hỗ trợ tìm nghiệm cho *Quadratic Programming*.

Tuy nhiên, việc giải bài toán này trở nên phức tạp khi số chiều  $d$  của không gian dữ liệu và số điểm dữ liệu  $N$  tăng lên cao.

Người ta thường giải bài toán đối ngẫu của bài toán này. Thứ nhất, bài toán đối ngẫu có những tính chất thú vị hơn khiến nó được giải hiệu quả hơn. Thứ hai, trong quá trình xây dựng bài toán đối ngẫu, người ta thấy rằng SVM có thể được áp dụng cho những bài toán mà dữ liệu không *tuyến tính*, tức các đường phân chia không phải là một mặt phẳng mà có thể là các mặt có hình thù phức tạp hơn.

**Xác định class cho một điểm dữ liệu mới:** sau khi tìm được mặt phân cách  $w^T X + b = 0$ , lớp của một điểm dữ liệu mới sẽ được xác định bằng cách:

$$\text{class}(X) = \text{sgn}(w^T X + b)$$

Trong đó hàm *sgn* là hàm xác định dấu, nhận giá trị 1 nếu đối số là không âm và -1 nếu ngược lại.

### 2.3.2 Logistic Regression[36]

#### 2.3.2.1 Giới thiệu

Trong phần này, chúng ta sẽ làm quen với một mô hình phân loại dữ liệu tuyến tính. Đây là phương pháp phân loại dữ liệu đơn giản hơn SVM nhưng hiệu quả không kém. Đó là Logistic Regression.

*Phân loại* cũng giống như vấn đề *hồi quy*, ngoại trừ các giá trị  $y$  muốn dự đoán chỉ chiếm một số nhỏ các giá trị rời rạc. Nếu bỏ qua thực tế là  $y$  có giá trị rời rạc, việc sử dụng thuật toán hồi quy tuyến tính để dự đoán là khả thi.

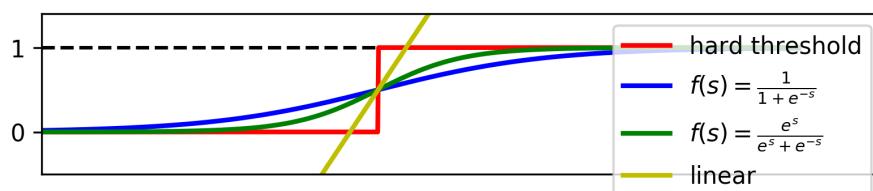
#### 2.3.2.2 Mô hình Logistic Regression

Dầu ra dự đoán của Logistic regression thường được viết chung dưới dạng:

$$f(x) = \theta(w^T x)$$

Trong đó  $\theta$  được gọi là hàm logistic. Cần chọn một *activity function* thỏa mãn các tính chất sau để áp dụng vào mô hình *Logistic Regression*:

- Là hàm số liên tục nhận giá trị thực, bị chặn trong khoảng (0, 1).
- Nếu coi điểm có tung độ là  $\frac{1}{2}$  làm điểm phân chia thì các điểm càng xa điểm này về phía bên trái có giá trị càng gần 0. Ngược lại, các điểm càng xa điểm này về phía phải có giá trị càng gần 1.
- Có đạo hàm tại mọi nơi.



**Hình 20:** Các activation function khác nhau[37]

Trong các hàm có tính chất trên, thì hàm *sigmoid*:

$$f(s) = \frac{1}{1 + e^{-s}} = \sigma(s)$$

được sử dụng nhiều nhất vì:

- Hàm số này bị chặn trong khoảng (0, 1).
- $\lim_{x \rightarrow -\infty} \sigma(s) = 0$        $\lim_{x \rightarrow +\infty} \sigma(s) = 1$
- $\sigma'(s) = \sigma(s)(1 - \sigma(s))$ .

### 2.3.2.3 Xây dựng hàm mất mát và phương pháp tối ưu

### 2.3.2.4 Xây dựng hàm chi phí

Với mô hình trên, giả sử xác suất điểm dữ liệu  $x$  rơi vào lớp 1 là  $f(w^T x)$  và rơi vào lớp 0 là  $1 - f(w^T x)$ . Vậy với các điểm dữ liệu huấn luyện ban đầu, có thể viết:

$$\begin{aligned} P(y_i = 1|x_i; w) &= f(w^T x_i) \\ P(y_i = 0|x_i; w) &= 1 - f(w^T x_i) \end{aligned}$$

Trong đó,  $P(y_i = 1|x_i; w)$  là xác suất xảy ra sự kiện đầu ra  $y_i = 1$  khi biết tham số mô hình là  $w$  và dữ liệu đầu vào  $x_i$ . Mục đích là tìm các hệ số  $w$  sao cho  $f(w^T x_i)$  càng gần với 1 càng tốt với các điểm dữ liệu thuộc lớp 1 và càng gần với 0 càng tốt với những điểm thuộc lớp 0.

Ký hiệu  $z_i = f(w^T x_i)$  và viết gộp lại hai biểu thức bên trên ta được:



$$P(y_i = 1|x_i; w) = z_i^{y_i}(1 - z_i)^{1-y_i}$$

Xét trên toàn bộ tập huấn luyện với  $\mathbf{X} = [x_1, x_2, \dots, x_N] \in R^{d*N}$  và  $\mathbf{y} = [y_1, y_2, \dots, y_N]$ , chúng ta cần tìm  $w$  để biểu  $P(\mathbf{y}|\mathbf{X}; w)$  đạt giá trị lớn nhất. Ở đây,  $\mathbf{X}, \mathbf{y}$  là các biến ngẫu nhiên.

Giả sử thêm rằng các điểm dữ liệu được sinh ra một cách ngẫu nhiên độc lập với nhau, khi đó:

$$P(\mathbf{y}|\mathbf{X}; w) = \prod_{i=1}^N P(y_i = 1|x_i; w) = \prod_{i=1}^N z_i^{y_i}(1 - z_i)^{1-y_i}$$

Việc tối ưu hàm số trên rất khó. Hơn nữa, khi  $N$  lớn, tích của  $N$  số nhỏ hơn 1 sẽ là một số rất bé, dễ dẫn tới sai số. Phương pháp đề xuất là lấy logarit tự nhiên 2 về để chuyển phép nhân thành phép cộng, sau đó nhân với -1 và gọi đó là hàm chi phí. Lúc này, bài toán tìm các tham số  $w$  để  $P(\mathbf{y}|\mathbf{X}; w)$  sẽ chuyển thành bài toán tìm giá trị nhỏ nhất của hàm chi phí:

$$J(w) = -\log P(\mathbf{y}|\mathbf{X}; w) = -\sum_{i=1}^N (y_i \log z_i + (1 - y_i) \log(1 - z_i))$$

### 2.3.2.5 Tối ưu hàm chi phí

Phương pháp *Stochastic Gradient Descent* (SGD) sẽ được dùng để tối ưu hàm chi phí. Theo SGD, công thức cập nhật trọng số mô hình cho một điểm dữ liệu sẽ có dạng:

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad (6)$$

Việc cập nhật này sẽ thực hiện lần lượt qua các điểm dữ liệu.

Quay lại bài toán Logistic Regression, hàm chi phí tại 1 điểm dữ liệu  $(x_i, y_i)$  là:

$$J(w; x_i, y_i) = -(y_i \log z_i + (1 - y_i) \log(1 - z_i))$$

Với đạo hàm:

$$\frac{\partial J(w; x_i, y_i)}{\partial w} = \left(\frac{y_i}{z_i} - \frac{1 - y_i}{1 - z_i}\right) \frac{\partial z_i}{\partial w} = \frac{z_i - y_i}{z_i(1 - z_i)} \frac{\partial z_i}{\partial w} \quad (7)$$

Với khảo sát ban đầu, chọn hàm Sigmoid là *activity function*, nên ta sẽ chọn  $z = \frac{1}{1 + e^{-w^T x}}$ , khi đó:

$$\frac{\partial z_i}{\partial w} = z_i(1 - z_i)$$



thay vào (7), ta được:

$$\frac{\partial J(w; x_i, y_i)}{\partial w} = z_i - y_i \quad (8)$$

Thay (8) vào (6), công thức cập nhật trọng số cho mô hình *Logistic Regression* theo phương pháp SGD sẽ là:

$$w = w - \alpha(z_i - y_i)x_i$$

Với  $z = \frac{1}{1 + e^{-w^T x}}$

Ban đầu, các trọng số được khởi tạo một cách ngẫu nhiên, sau đó việc cập nhật sẽ được thực hiện liên tục đến khi hàm chi phí đạt đến giá trị chấp nhận được thì dừng.

### 2.3.3 Áp dụng phân loại nhị phân cho bài toán phân lớp[23]

Các mô hình trên đều chỉ phân loại nhị phân. Trên thực tế, một bài toán phân loại thường sẽ có nhiều hơn 2 lớp. Vậy, làm sao để áp dụng mô hình phân loại nhị phân vào bài toán phân loại nhiều lớp?

Có 3 phương pháp phổ biến để áp dụng mô hình phân loại nhị phân cho bài toán phân loại nhiều lớp.

#### 2.3.3.1 One-vs-one

Xây dựng rất nhiều bộ phân loại nhị phân cho từng cặp lớp. Bộ thứ nhất phân biệt lớp 1 và lớp 2, bộ thứ hai phân biệt lớp 1 và lớp 3, ... Khi có một dữ liệu mới vào, đưa nó vào toàn bộ các bộ phân loại trên. Kết quả cuối cùng có thể được xác định bằng cách xem lớp nào mà điểm dữ liệu đó được phân vào nhiều nhất. Hoặc với *Logistic Regression* thì ta có thể tính tổng các xác suất tìm được sau mỗi bộ phân loại nhị phân.

Như vậy, nếu có N lớp thì sẽ cần đến  $\frac{N(N - 1)}{2}$  bộ phân loại. Đây là một con số lớn, không có lợi cho mặt tính toán. Hơn nữa, nếu dữ liệu ban đầu thuộc lớp 2 mà cho vô bộ phân loại lớp 5 và 6 thì kết quả sẽ là lớp 5 hoặc lớp 6. Như vậy là không hợp lý.

#### 2.3.3.2 Hierarchical

Ý tưởng của phương pháp này là ta sẽ gom các lớp tương đồng với nhau thành những lớp lớn, sau đó tiến hành xây dựng các bộ phân loại dùng để phân biệt các lớp này với nhau. Sau đó các bộ phân loại sẽ được xây dựng để nhận dạng các lớp nhỏ trong mỗi lớp lớn đó.

**Ví dụ:** chúng ta cần nhận dạng 4 số là 4, 5, 6, 7. Chúng ta thấy số 4 và 7 khá giống nhau nên ta gộp thành 1 lớp [4, 7], tương tự 5 và 6 thành 1 lớp [5, 6]. Ta xây dựng bộ phân loại [4, 7] với [5, 6]. Đồng thời, chúng ta xây dựng thêm 2 bộ phân loại số 4 với số 7,

số 5 với số 6. Sau khi biết dữ liệu đầu vào thuộc lớp [4, 7] hay [5, 6], ta sẽ cho chúng vào bộ nhận dạng tương ứng để cho ra kết quả cuối cùng.

Phương pháp này sẽ tiết kiệm số bộ phân loại hơn so với phương pháp *one-vs-one* nhưng có một nhược điểm là nếu dữ liệu bị phân loại sai ở bộ phân loại đầu tiên, thì kết quả cuối cùng chấn chấn sai.

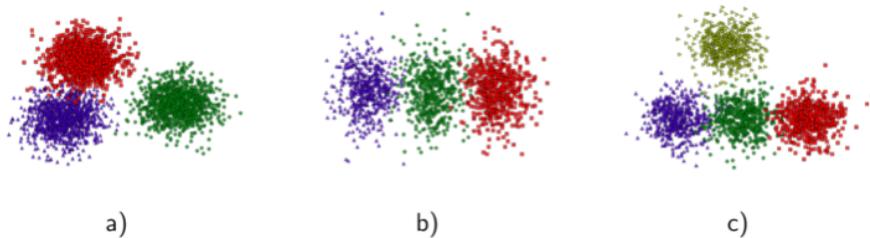
### 2.3.3.3 One-vs-rest

Đây là phương pháp được sử dụng phổ biến nhất.

Nếu có  $N$  lớp cần phân loại, sẽ có  $N$  bộ phân loại được xây dựng. Bộ phân loại thứ nhất dùng để phân loại lớp 1 hoặc không phải lớp 1, bộ phân loại thứ 2 dùng để nhận dạng lớp 2 hoặc không phải lớp 2, ..., bộ phân loại thứ  $N$  dùng để nhận dạng lớp  $N$  hay không phải lớp  $N$ . Sau đó, kết quả cuối cùng là lớp có xác suất dự đoán cao nhất.

**Ví dụ:** chúng ta cần nhận dạng chữ số viết tay từ 0 đến 9. Chúng ta sẽ xây dựng 10 bộ phân loại. Bộ phân loại thứ nhất biệt là số 0 hay không phải số 0. Bộ phân loại thứ 2 nhận sang số 1 hoặc không phải số 1... Bộ phân loại thứ 10 nhận dạng số 9 hoặc không phải số 9. Giả sử dữ liệu nhập vào là số 0. Bộ phân loại thứ nhất dự đoán mẫu là số 0 với xác suất 0.9, bộ phân loại thứ 2 dự đoán mẫu là số 1 với xác suất là 0.5, ..., bộ phân loại thứ 10 dự đoán mẫu là số 9 với xác suất 0.7. Giả sử dự đoán mẫu là số 0 với xác suất 0.9 là cao nhất thì kết luận mẫu là số 0.

**Lưu ý:** Chúng ta có thể kết hợp các phương pháp này với nhau để có thể đưa ra kết quả dự đoán tốt cũng như giảm số lượng bộ phân loại phải xây dựng.



**Hình 21:** Ví dụ về sự phân bố các lớp trong bài toán phân loại nhiều lớp

- Hình a, cả 3 phương pháp trên đều áp dụng được.
- Hình b, phương pháp *one-vs-rest* không áp dụng được vì lớp màu xanh và lớp không phải màu xanh (màu đỏ và lục) "không thể phân chia tuyến tính". Trong trường hợp này, phương pháp *one-vs-one* và *phân tầng* khả thi.
- Hình c, ban đầu phương pháp *one-vs-rest* không áp dụng được vì lí do giống trường hợp b. Nhưng nếu ta áp dụng phương pháp *phân tầng* để phân biệt lớp màu đỏ và không phải màu đỏ. Sau đó, nếu mẫu rơi vào lớp không phải màu đỏ thì ta dùng phương pháp *one-vs-rest* là hiệu quả nhất.



## 2.4 Phân lớp cho dữ liệu dạng hình ảnh vệ tinh

### 2.4.1 Giới thiệu

Hình ảnh vệ tinh rất đa dạng và đóng một vai trò quan trọng trong việc cung cấp thông tin địa lý. Các công nghệ viễn thám vệ tinh thu thập dữ liệu hoặc hình ảnh theo khoảng thời gian đều đặn. Số lượng dữ liệu nhận được tại các trung tâm dữ liệu là rất lớn và nó đang phát triển theo cấp số nhân. Cần có cơ chế hiệu quả để trích xuất và giải thích các thông tin có giá trị từ các hình ảnh vệ tinh lớn. Phân loại hình ảnh vệ tinh là một kỹ thuật mạnh mẽ để trích xuất thông tin từ rất nhiều hình ảnh vệ tinh.

Phân loại hình ảnh vệ tinh là một quá trình nhóm các điểm ảnh thành các lớp có ý nghĩa, nó cũng có thể được gọi là trích xuất thông tin từ hình ảnh vệ tinh. Phân loại hình ảnh vệ tinh không phức tạp, nhưng nhà phân tích phải mất nhiều quyết định và lựa chọn trong quá trình phân loại hình ảnh vệ tinh, nó liên quan đến việc giải thích hình ảnh viễn thám, khai thác dữ liệu không gian, nghiên cứu các loại thảm thực vật khác nhau như nông nghiệp và lâm nghiệp, nghiên cứu đô thị và xác định các mục đích sử dụng đất khác nhau trong một khu vực.

### 2.4.2 Nhu cầu trong việc phân loại ảnh vệ tinh

Phân loại hình ảnh vệ tinh đóng một vai trò chính trong việc trích xuất và giải thích thông tin có giá trị từ các hình ảnh vệ tinh lớn. Phân loại hình ảnh vệ tinh được yêu cầu cho:

- Khai phá dữ liệu không gian.
- Trích xuất thông tin cho một ứng dụng.
- Tạo bản đồ chuyên đề.
- Giải thích hình ảnh vệ tinh trực quan và kỹ thuật số.
- Khảo sát địa lý.
- Quản lý thiên tai.

### 2.4.3 Các kỹ thuật trong xử lý ảnh vệ tinh

Có một số phương pháp và kỹ thuật cho phân loại ảnh vệ tinh. Phương pháp phân loại ảnh vệ tinh có thể chia làm ba loại:

- Tự động: Các phương pháp phân loại hình ảnh vệ tinh tự động sử dụng các thuật toán được áp dụng có hệ thống toàn bộ ảnh vệ tinh vào các điểm ảnh nhóm vào các phân loại có ý nghĩa. Phần lớn các phương pháp phân loại thuộc nhóm này. Các phương pháp phân loại hình ảnh vệ tinh tự động được phân loại thành hai loại: giám sát và không giám sát.



- Giám sát: Các phương pháp phân loại theo giám sát yêu cầu đầu vào từ một nhà phân tích. Đầu vào từ nhà phân tích được gọi là tập huấn luyện. Mẫu huấn luyện là yếu tố quan trọng nhất trong các phương pháp phân loại hình ảnh vệ tinh được giám sát. Độ chính xác của phương pháp cao phụ thuộc vào mẫu lấy để huấn luyện. Các mẫu huấn luyện là hai loại, một loại được sử dụng để phân loại và một để giám sát độ chính xác phân loại. Các kỹ thuật phân loại khác nhau liên quan đến các loại phương pháp kết hợp khác nhau. Phân loại theo giám sát bao gồm các chức năng bổ sung như phân tích dữ liệu đầu vào, tạo mẫu huấn luyện, và xác định chất lượng của các mẫu huấn luyện. Mạng thần kinh nhân tạo: Các thuật toán nằm trong Mạng thần kinh nhân tạo (ANN) mô phỏng quá trình học tập của con người để kết hợp các nhãn có ý nghĩa chính xác với các điểm ảnh. Lợi thế của các thuật toán phân loại hình ảnh vệ tinh dựa trên ANN rất dễ dàng kết hợp các dữ liệu bổ sung vào quá trình phân loại và nâng cao độ chính xác phân loại.
  - Không giám sát: Kỹ thuật phân loại không giám sát sử dụng các cơ chế phân cụm để nhóm các điểm ảnh vệ tinh vào các lớp hoặc cụm không có nhãn. Sau đó các nhà phân tích chỉ định nhãn có ý nghĩa cho các cụm và tạo ra hình ảnh vệ tinh được phân loại tốt. Phân loại hình ảnh vệ tinh phổ biến nhất không được giám sát là ISODATA, Support Vector Machine và K-Means.
- Thủ công: Các phương pháp phân loại hình ảnh vệ tinh thủ công là những phương pháp mạnh mẽ, hiệu quả. Nhưng phương pháp thủ công tiêu tốn nhiều thời gian hơn. Trong các phương pháp thủ công, nhà phân tích phải làm quen với khu vực được bao phủ bởi hình ảnh vệ tinh. Tính hiệu quả và tính chính xác của phân loại, phụ thuộc vào kiến thức của nhà phân tích và sự quen thuộc trong lĩnh vực nghiên cứu.
  - Hỗn hợp: Các phương pháp phân loại hình ảnh vệ tinh kết hợp những ưu điểm của phương pháp tự động và thủ công. Phương pháp hỗn hợp (lai) sử dụng các phương pháp phân loại hình ảnh vệ tinh tự động để phân loại ban đầu, các phương pháp thủ công khác được sử dụng để tinh chỉnh phân loại và sửa lỗi.

#### 2.4.4 Các phương pháp phân loại ảnh vệ tinh

Phần này minh họa cho vài phương pháp phân loại hình ảnh vệ tinh gần đây.

- J. Shabnam và cộng sự đã giới thiệu phương pháp phân loại hình ảnh vệ tinh có giám sát để phân loại hình ảnh vệ tinh có độ phân giải rất cao thành các lớp cụ thể sử dụng logic mờ. Phương pháp này phân loại hình ảnh vệ tinh thành năm lớp chính: bóng tối, thảm thực vật, đường xá, đất xây dựng và đất trống. Phương pháp này sử dụng các kỹ thuật phân đoạn hình ảnh và mờ để phân loại hình ảnh vệ tinh. Nó áp dụng hai mức phân đoạn, phân đoạn cấp 1 xác định và phân loại bóng, thảm thực vật và đường. Phân đoạn cấp hai xác định các tòa nhà. Hơn nữa, nó sử dụng kiểm tra theo ngữ cảnh để phân loại phân đoạn và phân đoạn chưa được phân loại. Kỹ thuật mờ được sử dụng để nâng cao độ chính xác phân loại tại biên giới của vật thể.



- Trình bày phương pháp phân loại hình ảnh vệ tinh được giám sát để xác định nước, đất đô thị và đất xanh trên các hình ảnh vệ tinh. Phương pháp này tập huấn luyện cho mỗi lớp và tính giá trị ngưỡng bằng kỹ thuật k-means và LDA. Phương pháp trích xuất các tính năng cấp thấp từ hình ảnh vệ tinh và áp dụng thuật toán k-means để nhóm vào các cụm không có nhãn. Nhãn có ý nghĩa được gán cho các lớp không gắn nhãn bằng cách so sánh các giá trị ngưỡng với các tính năng được trích xuất.
- Mô tả phương pháp phân loại hình ảnh vệ tinh đại dương dựa trên quản lý theo bản thể học (ontology). Phương pháp này minh họa sức mạnh của bản thể học trong phân loại hình ảnh vệ tinh đại dương. Phương pháp trích xuất các tính năng cấp thấp từ hình ảnh vệ tinh đại dương và đại diện cho định dạng tệp cú pháp. Tệp cú pháp này được hợp nhất với các quy ước ghi chú và chú giải bản thể miền và các quy tắc ghi nhãn. Quy tắc ghi nhãn, quy tắc đào tạo, quy tắc cây quyết định nhị phân và các quy tắc chuyên gia được thể hiện bằng cách sử dụng ngôn ngữ SWRL. Phương pháp này tạo ra kết quả phân loại hình ảnh vệ tinh đại dương với sự hỗ trợ đào tạo, chuyên gia của con người, hỗ trợ ra quyết định và các quy tắc ghi nhãn. Cũng cung cấp một công cụ như là plug-in cho trình biên tập bản thể luận protégé. Công cụ này hỗ trợ các hình ảnh vệ tinh đại dương với sự hỗ trợ của các bản thể luận miền.
- S. Muhammad và cộng sự, đề xuất một phương pháp phân loại hình ảnh vệ tinh được giám sát bằng kỹ thuật cây quyết định. Phương pháp này trích xuất các tính năng từ hình ảnh vệ tinh dựa trên màu sắc pixel và cường độ. Các tính năng trích xuất giúp xác định các đối tượng cư trú trong các hình ảnh vệ tinh. Các phương pháp phân loại hình ảnh vệ tinh bằng cách sử dụng cây quyết định với sự hỗ trợ của các đối tượng được xác định.
- Selim đề xuất một phương pháp phân loại sử dụng kỹ thuật Bayesian. Phương pháp sử dụng thông tin không gian để phân loại hình ảnh vệ tinh có độ phân giải cao. Phương pháp thực hiện phân loại theo hai giai đoạn. Giai đoạn 1: các đặc trưng quang phổ và tính văn bản được trích ra cho mỗi pixel để huấn luyện các phân loại Bayes với các mô hình mật độ phi tham số rời rạc. Giai đoạn 2: thuật toán phân chia và ghép được sử dụng để chuyển đổi các bản đồ phân loại cấp độ pixel sang các vùng tiếp giáp.
- Kỹ thuật ISODATA là phương pháp phân loại vệ tinh không được giám sát phổ biến nhất. Nó tạo ra số lượng được xác định trước của các cụm hoặc lớp không được dán nhãn trong một hình ảnh vệ tinh. Sau đó các nhãn có ý nghĩa được gán cho các cụm. Các tham số ISODATA cần một số tham số kiểm soát số lượng các cụm và các lần lặp sẽ được chạy. Trong một vài trường hợp các cụm có thể chứa các điểm ảnh của các lớp khác nhau. Trong các tình huống như vậy, ISODATA sử dụng kỹ thuật tách cụm (cluster-busting) để ghi nhãn các lớp phức tạp.
- K-Means là một thống kê phổ biến và kỹ thuật khai thác dữ liệu. Phân chia n quan sát thành các nhóm k dựa trên giá trị trung bình của Euclidean. Ưu điểm với kỹ thuật



K-Means rất đơn giản để xử lý và thực hiện nhanh. Hạn chế với phương pháp này là nhà phân tích cần biết số lớp học tiên quyết.

- Support Vector Machine (SVM) là một phương pháp phân loại thống kê không được kiểm soát thông số phi tham số. Phương pháp này có thể được sử dụng để trích ra bản đồ sử dụng đất. SVM hoạt động trên giả định rằng không có thông tin về cách phân phối dữ liệu tổng thể. SVM làm giảm chi phí phân loại vệ tinh, tăng tốc độ và nâng cao độ chính xác.
- Phương pháp khoảng cách tối thiểu tính toán phô trung bình của mỗi lớp được xác định trước và chỉ định pixel cho một nhóm có khoảng cách thấp nhất. Để thực hiện và đơn giản để xử lý. Nhưng phương pháp khoảng cách tối thiểu chỉ xem xét giá trị trung bình. Phương pháp khoảng cách Mahalanobis rất giống với phương pháp khoảng cách tối thiểu. Nó sử dụng ma trận hiệp phương sai Thống kê để phân loại hình ảnh vệ tinh.
- Parallelepiped được thực hiện dựa trên các hộp hình chữ nhật cho mỗi lớp. Các ranh giới song song cho mỗi lớp được xác định trước. Ranh giới được xác định trước xác định điểm kiểm tra của hình ảnh kiểm tra và xác định lớp của pixel. Phương pháp song song là nhanh và dễ dàng để chạy, nhưng chồng chéo có thể tạo ra kết quả sai.
- Phương pháp tối đa khả năng (Maximum likelihood) là một cách tiếp cận giám sát thống kê để công nhận các mẫu. Nó phân bổ các điểm ảnh đến các lớp thích hợp dựa trên các giá trị xác suất của các điểm ảnh. Khả năng tối đa là một phương pháp hiệu quả để phân loại các điểm ảnh của hình ảnh vệ tinh. Nhưng đó là thời gian và không đủ dữ liệu thực địa cho ra kết quả kém.



### 3 Phương pháp đề xuất và đánh giá

#### 3.1 Khảo sát dữ liệu

##### 3.1.1 Giới thiệu tổng quát về tập dữ liệu

Tập dữ liệu đầu vào được lấy từ cuộc thi do kaggle tổ chức bao gồm các tập dữ liệu sau: train\_geojson\_v3, grid\_sizes.csv, sixteen\_band, three\_band, train\_wkt\_v4.csv. Trong đó:

- three\_band: thư mục chứa 450 ảnh RGB. Chúng có định dạng .tif, ví dụ : 6110\_0\_0.tif
- sixteen\_band: thư mục này chứa 450 ảnh loại A, 450 ảnh loại M và 450 ảnh loại P. Mỗi ảnh điều có định dạng tif. Ví dụ: 6110\_0\_0\_A.tif, 6110\_0\_0\_M.tif, 6110\_0\_0\_P.tif
- grid\_sizes.csv: file này chứa giá trị  $x_{max}$ ,  $y_{min}$  của từng id ảnh để áp dụng trong công thức chuyển đổi tập dữ liệu huấn luyện.
- train\_wkt\_v4.csv: file này bao gồm bao gồm 25 hình. Mỗi hình gồm 10 nhãn đã được phân loại với mỗi nhãn sẽ chứa tất cả các tọa độ địa lý đã được xử lý nằm trong khoảng [0, 1] theo trục x và [-1, 0] theo trục y.
- train\_geojson\_v3: tương tự như file train\_wkt\_v4.csv nhưng tọa độ của từng nhãn được lưu theo định dạng geojson.

Đầu vào trong tập dữ liệu là ảnh vệ tinh bao gồm:

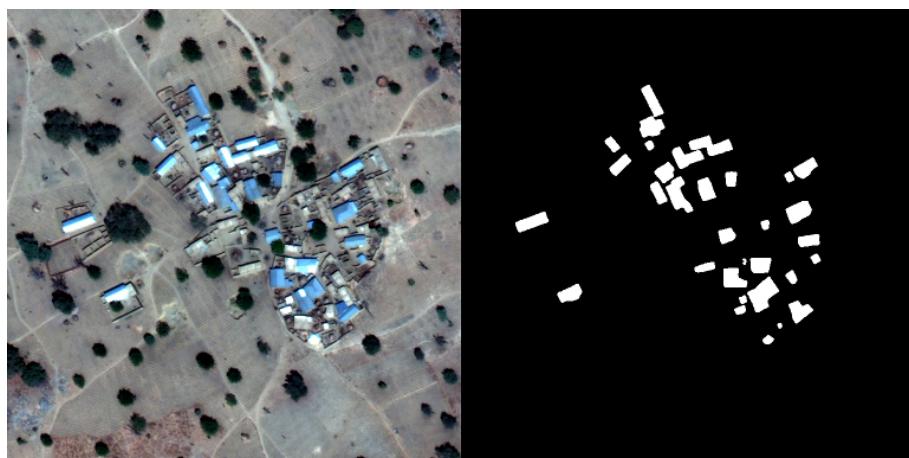
- Ảnh RGB + P có bước sóng nằm trong khoảng (450 - 690 nm), có độ phân giải 0.31m/pixel.
- Ảnh M có bước sóng nằm trong khoảng (400 - 1040 nm), có độ phân giải 1.24 m/pixel.
- Ảnh A có bước sóng nằm trong khoảng (1195-2365 nm), có độ phân giải 7.5 m/pixel.
- Tập huấn luyện: gồm 25 hình ảnh.
- Mỗi hình ảnh có kích thước 1km x 1km.

Đầu ra là quá trình phân loại tại mỗi pixel của hình ảnh sẽ thuộc loại nào trong các loại sau:

- 1. Tòa nhà bao gồm: tòa nhà lớn, khu dân cư ...
- 2. Công trình nhân tạo khác
- 3. Đường lớn
- 4. Đường nhỏ bao gồm: lối đi bộ, đường mòn ...

- 5. Cây bao gồm: rừng cây, nhóm cây, các loại cây riêng lẽ ...
- 6. Thảm thực vật bao gồm: đất trồng trọt, vụ mùa ...
- 7. Đường thủy
- 8. Nước đứng bao gồm: ao, hồ ...
- 9. Xe lớn bao gồm xe tải, xe buýt ...
- 10. Xe nhỏ bao gồm: xe ô tô, xe máy ...

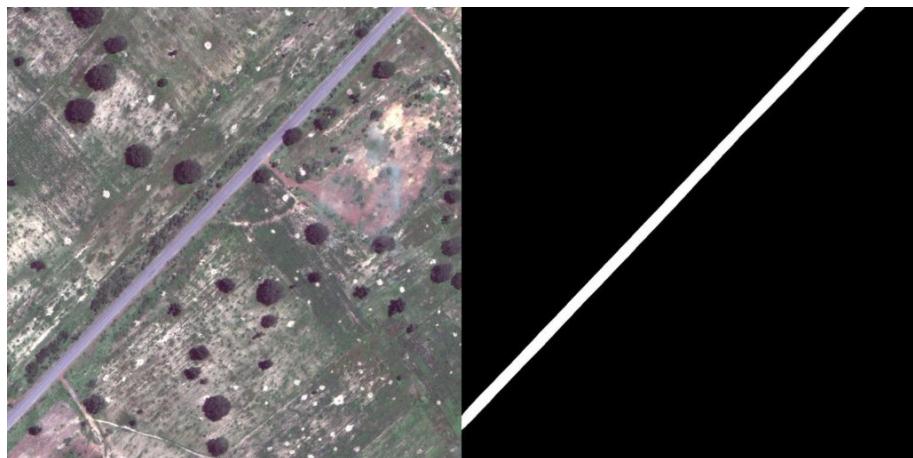
Bên dưới là một số mẫu kết quả cần hướng tới trong quá trình hiện thực:



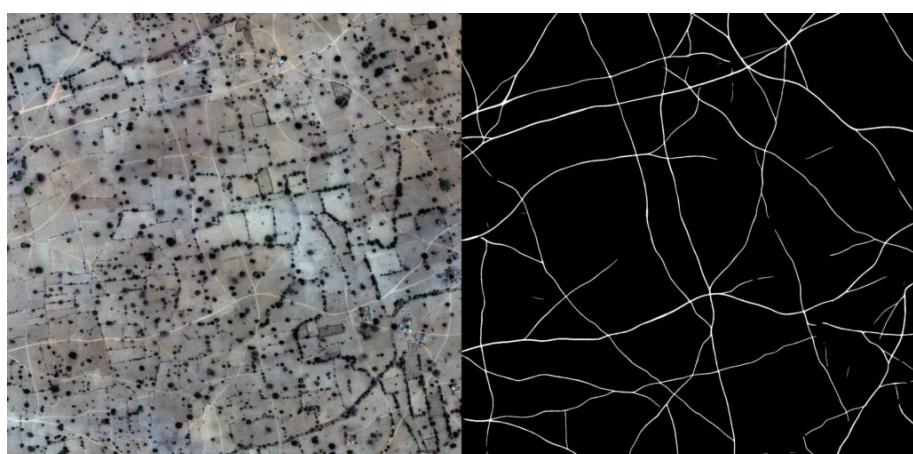
**Hình 22:** Dự đoán tòa nhà.



**Hình 23:** Dự đoán thảm thực vật.



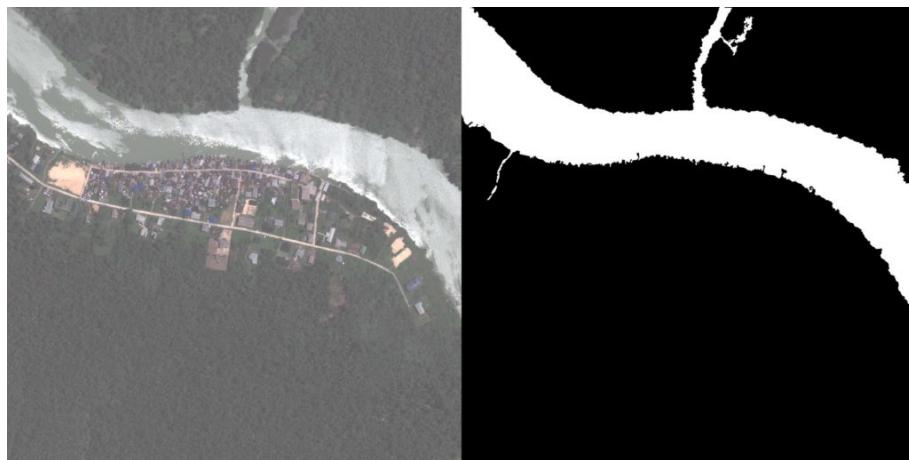
**Hình 24:** Dự đoán đường lớn.



**Hình 25:** Dự đoán đường nhỏ.



**Hình 26:** Dự đoán xe nhỏ.



**Hình 27:** Dự đoán đường thủy.

### 3.1.2 Mô tả tập ảnh

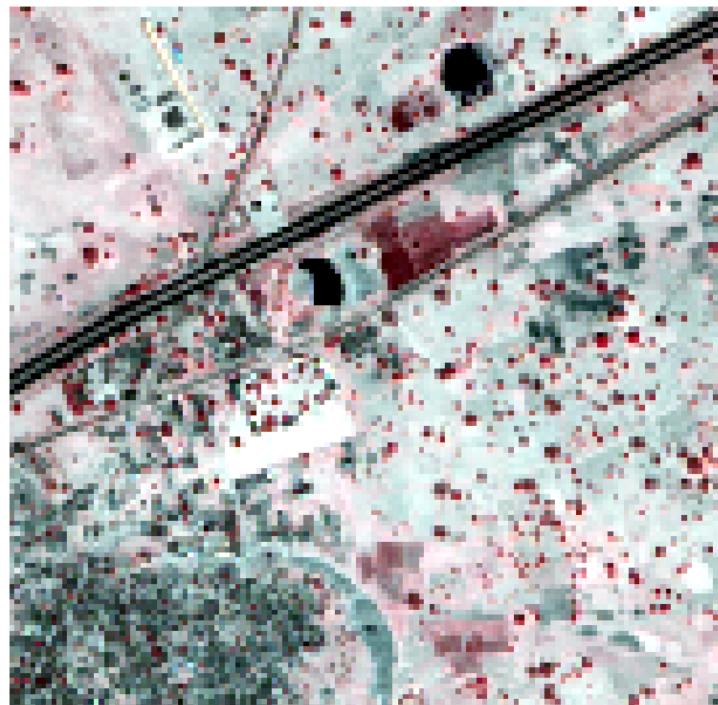
Mỗi hình ảnh trong tập ảnh là ảnh vệ tinh có kích thước 1km x 1km ở hai định dạng là 3-band và 16-band. Với mỗi một *id* ảnh cụ thể chúng sẽ được biểu diễn trên bốn loại ảnh khác nhau bao gồm ảnh RGB, ảnh A, ảnh M, ảnh P. Trong đó ảnh RGB là ảnh 3-band và ba ảnh còn lại thuộc vào tập 16-band. Hình ảnh trong tập ảnh có độ sâu màu là 11 bit và 14 bit thay cho 8 bit được sử dụng phổ biến, chính vì thế mỗi pixel mang nhiều thông tin hơn. Các hình ảnh multi-band được lấy từ dải multispectral (400 - 1040nm) và phạm vi sóng ngắn hồng ngoại (SWIR) (1195 - 2365nm). Tất cả các hình ảnh đều ở định dạng GeoTiff, phần mềm QGIS được sử dụng để xem ảnh.

- Ảnh 3-band: ảnh này là ảnh màu RGB tự nhiên, ảnh này có bước sóng nằm trong vùng ánh sáng nhìn thấy, nó là những hình ảnh mắt thường nhìn thấy được, đối với mỗi pixel thì có bộ ba số để biểu diễn cho màu tương ứng tại pixel đó. Nhìn tổng quan trong tập ảnh 3-band ta nhận thấy được kích thước chiều dài, chiều rộng của từng ảnh không giống nhau nhưng chúng có sự chênh lệch không đáng kể. Cụ thể chiều dài và chiều rộng của tập ảnh 3-band lần lượt giới hạn trong khoảng [3345 - 3350] và [3335 - 3403]. Ảnh 3-band thuộc kiểu 11 bit nên cường độ màu sắc nằm trong khoảng [0, 2047]. Mỗi pixel có độ phân giải 0.31m.



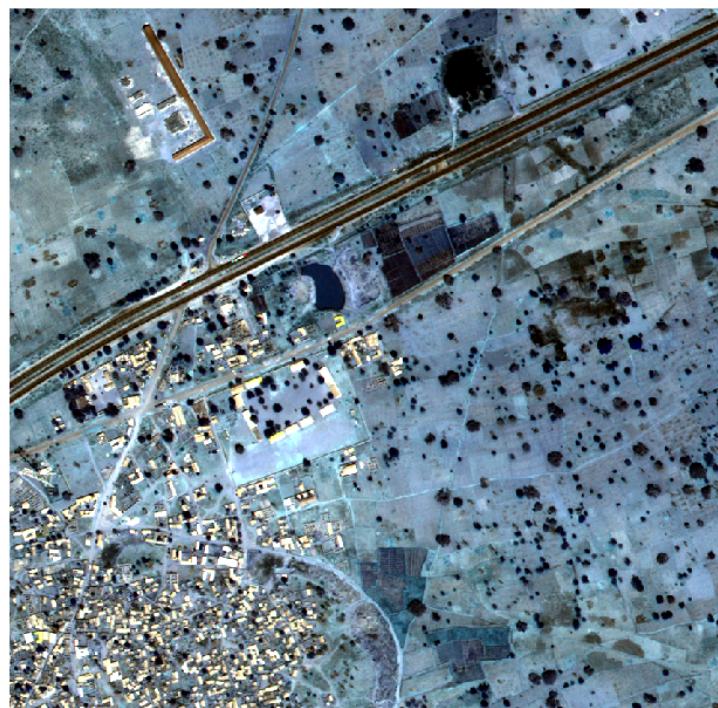
**Hình 28:** Ảnh 3-band minh họa với id 6110\_3\_1

- Ảnh 16-band: Các hình ảnh này có chứa thông tin các quang phổ bằng cách bắt các bước sóng rộng hơn. Bao gồm ba loại hình ảnh trong cùng một khu vực: ảnh P với độ phân giải cao chứa 1-band, một hình ảnh 8-band với độ phân giải thấp hơn là ảnh M và ảnh A có độ phân giải thấp nhất chứa 8-band.
  - Ảnh A: Với mỗi pixel sẽ có bộ tám số biểu diễn cho tám cường độ màu sắc tương ứng tại pixel đó, ảnh này có bước sóng nằm trong khoảng (1195 - 2365 nm) thuộc vùng ánh sáng không nhìn thấy. Ảnh A thuộc ảnh 14 bit nên cường độ màu giới hạn trong khoảng [0 - 16323]. Với chiều dài và chiều rộng của hình ảnh A nằm trong khoảng [133 - 134] và [132 - 137]. Mỗi pixel có độ phân giải 7.5m.



**Hình 29:** Ảnh A minh họa với id 6110\_3\_1

- Ảnh M: Với mỗi pixel sẽ có bộ tám số biểu diễn cho tám cường độ màu sắc tương ứng tại pixel đó, ảnh này có bước sóng nằm trong khoảng (400 - 1040 nm) nằm trong vùng ánh sáng nhìn thấy và không nhìn thấy. Ảnh M thuộc ảnh 11 bit nên cường độ màu giới hạn trong khoảng [0 - 2047]. Với chiều dài và chiều rộng của ảnh M giới hạn trong khoảng [834 - 838] và [831 - 851]. Mỗi pixel có độ phân giải 1.24m.



**Hình 30:** Ảnh M minh họa với id 6110\_3\_1

- Ảnh P: Với mỗi pixel sẽ có một số biểu diễn cho cường độ màu sắc tương ứng tại pixel đó. Ảnh P thuộc ảnh 11 bit nên cường độ màu giới hạn trong khoảng [0 - 2047] Với chiều dài và chiều rộng của ảnh P giới hạn trong khoảng [3348 - 3350] và [3336 - 3404]. Mỗi pixel có độ phân giải 0.31m.



**Hình 31:** Ảnh P minh họa với id 6110\_3\_1

### 3.1.3 Tập dữ liệu huấn luyện và đánh giá

#### 3.1.3.1 Tập dữ liệu huấn luyện

Tập huấn luyện là một file csv đã được xử lý từ file ảnh vệ tinh và chuyển đổi thành dữ liệu số giúp cho việc học thuận tiện hơn. Tập huấn luyện có chứa 25 ảnh.

Tập huấn luyện gồm có 3 trường: ImageId, ClassType, MultipolygonWKT

- ImageId: ID (định danh) của ảnh.
- ClassType (1-10): Nhãn của ảnh với ID là ImageId.
- MultipolygonWKT: bao gồm nhiều tập hợp, mỗi tập hợp là các tọa độ điểm, điểm đầu và cuối trùng nhau, khi nối tất cả các điểm trong cùng một tập hợp lại ta sẽ có được một đa giác, đa giác sẽ có nhãn là ClassType.

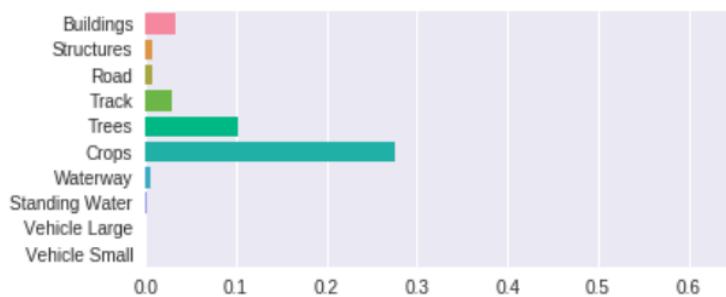
**Hình 32:** Cấu trúc tập huấn luyện

Dựa trên cấu trúc tập huấn luyện việc tìm hiểu khái niệm về polygon, multipolygon là gì rất cần thiết. Polygon là tập hợp tất cả các điểm sao cho các điểm này được nối với nhau thành một hình dạng bất kỳ khép kín, điểm đầu và điểm cuối sẽ được nối với nhau. Multipolygon sẽ chứa nhiều polygon. Như vậy polygon là tập hợp tất cả các điểm bao quanh bên ngoài một vật thể cần gán nhãn.

**Nhận xét:** Tọa độ các nhãn nằm trong tập huấn luyện nằm trong khoảng  $x = [0, 1]$  và  $y = [-1, 0]$  không đồng nhất với kích thước chiều dài và chiều rộng của ảnh nên rất khó để trích xuất đặc trưng từ ảnh dẫn đến việc học gấp khó khăn chính vì thế dẫn đến việc biến đổi tập dữ liệu này thành dạng máy tính có thể học được bằng cách dùng công thức biến đổi (được đề cập ở phần sau ở mục 3.1.4) để biến đổi dữ liệu số thực với  $x = [0, 1]$  và  $y = [-1, 0]$  sang một ma trận nhãn tương ứng với một ảnh. Ma trận này sẽ có chiều dài và chiều rộng bằng với chiều dài và chiều rộng của ảnh gốc tính theo đơn vị pixels. Từ tập dữ liệu hiện tại chuyển đổi tất cả các đa giác có cùng "ImageId" vào cùng một ma trận, mỗi phần tử trong ma trận là ánh xạ nhãn của đa giác tương ứng, các đa giác có cùng một nhãn sẽ được gán cùng giá trị. Sau khi dữ liệu được biến đổi sẽ có 25 ma trận tương ứng với ảnh gốc ban đầu, mỗi ma trận đó là ma trận ánh xạ của nhãn tương ứng của từng pixels trong ảnh gốc và 25 ma trận đó được dùng để huấn luyện.

### 3.1.3.2 Đánh giá tập dữ liệu huấn luyện

Lượt đồ khảo sát tần suất xuất hiện của các nhãn trong tập huấn luyện liệt kê bên dưới được lấy từ blog trên kaggle.



Hình 33: Lượt đồ tần suất xuất hiện của các nhãn trong tập huấn luyện

Từ lượt đồ kết hợp với việc khảo sát tập ảnh có trong tập huấn luyện có thể rút ra một số kết luận như sau:

- Tần suất xuất hiện của tất cả các nhãn không đồng đều nhau và có sự chênh lệch rất lớn, điều này dẫn đến một kết luận rằng việc huấn luyện một mô hình riêng cho từng nhãn sẽ làm việc tốt hơn so với việc sử dụng một mô hình để dự đoán cho tất cả các nhãn cùng một lúc. Tuy nhiên do thời gian không đủ cho việc tìm hiểu nên trong luận văn này chỉ có một mô hình để dự đoán cho tất cả các nhãn được sử dụng.
- Một trong những khó khăn gặp phải là thiếu dữ liệu huấn luyện. 25 bức ảnh được cung cấp bao gồm 25 khu vực khác nhau. Những hình ảnh này khá đa dạng: các tòa nhà, rừng, thảm thực vật, ao, hồ..., chúng có đặc trưng khác nhau mặc dù chúng thuộc cùng một nhãn. Điều này làm cho việc nhận dạng trở nên khó khăn hơn rất nhiều.
- Việc gán nhãn cho loại 9, 10 là phương tiện giao thông lớn (Vehicle Large) và phương tiện giao thông nhỏ (Vehicle Small) không đáng tin cậy. Hình ảnh dưới biểu diễn cho phương tiện giao thông đã được gán nhãn, chúng thường được gán nhầm với mảnh vỡ, pixel của đường...



**Hình 34:** Các phương tiện giao thông bị gán sai nhãn

Do phương tiện giao thông (nhãn 9 và 10) chiếm số lượng rất ít và có chứa nhiều đặc trưng khó có thể nhận dạng được nên nhóm quyết định không dự đoán hai nhãn 9 (phương tiện giao thông lớn) và nhãn 10 (phương tiện giao thông nhỏ).

#### 3.1.4 Chuyển đổi tập dữ liệu huấn luyện

Tập ảnh trong tập huấn luyện mà cuộc thi cung cấp tạo ra một tập hợp các tọa độ địa lý nằm trong khoảng  $x = [0, 1]$  và  $y = [-1, 0]$ . Những hình ảnh này được chụp từ cùng một vùng trên trái đất. Để sử dụng những hình ảnh này, tọa độ lưới của mỗi hình ảnh được cung cấp để biết được làm thế nào để quy mô chúng và sắp xếp chúng với các hình ảnh trong các điểm ảnh. Trong file grid\_sizes.csv sẽ nhận được giá trị  $Xmax$  và  $Ymin$  cho mỗi imageId. Đối với mỗi hình ảnh sẽ có thể có được chiều rộng ( $W$ ) và chiều cao ( $H$ ) từ raster hình ảnh. Đối với ảnh 3-band nó có kích thước  $3391 \times 3349 \times 3$  với  $W = 3349$  pixel và  $H = 3391$  pixel, sau đó có thể quy mô hóa dữ liệu như sau:

$$W' = W \cdot \frac{W}{W + 1} \quad (9)$$

$$x' = \frac{x}{x_{max}} \cdot W' \quad (10)$$

$$H' = H \cdot \frac{H}{H + 1} \quad (11)$$

$$y' = \frac{y}{y_{min}} \cdot H' \quad (12)$$

Với  $x'$ ,  $y'$  là tọa độ sau khi chuyển đổi,  $x'$  giới hạn trong khoảng  $[0, \text{chiều cao của ảnh}]$ ,  $y'$  giới hạn trong khoảng  $[0, \text{chiều rộng của ảnh}]$  như vậy sau khi chuyển từ tọa độ địa lý  $x = [0, 1]$ ,  $y = [-1, 0]$  sẽ được  $x' = [0, \text{chiều cao của ảnh}]$ ,  $y' = [0, \text{chiều rộng của ảnh}]$  mới nằm trong tọa độ ảnh.

### 3.2 Tiền xử lý dữ liệu

Trong quá trình hiện thực sẽ tiến hành việc chuyển đổi dữ liệu tập huấn luyện ban đầu là tọa độ polygon giới hạn trong khoảng  $[0, 1]$  và  $[-1, 0]$  để ánh xạ trực tiếp sang tọa độ pixel trong hình ảnh tương ứng bằng cách sử dụng công thức chuyển tọa độ ở mục 3.1.4. Mã màu tương ứng với mỗi tọa độ pixel được lấy để áp dụng vào quá trình huấn luyện.

Tiếp đến tiến hành thực hiện việc xử lý trên bốn loại hình ảnh RGB, ảnh A, ảnh M, ảnh P. Vì kích thước của bốn loại ảnh này là không bằng nhau khi xét trên cùng một Id ảnh chính vì thế cần phải thay đổi kích thước sang một kích thước chuẩn để dễ dàng trong việc gộp tất cả các đặc trưng lại với nhau. Cụ thể thực hiện việc chuyển kích thước ba ảnh RGB, ảnh A, ảnh P sang kích thước ảnh M bằng cách cắt bỏ phần chiều dài hoặc chiều rộng của ảnh RGB, ảnh A, ảnh P để cho kích thước chiều dài, chiều rộng trên từng ảnh bằng nhau. Sau đó chuyển kích thước của ba ảnh này sang kích thước của ảnh M. Sau đó gộp tất cả các mã màu của bốn ảnh trên từng pixel tạo thành ma trận  $xs$ , kết hợp với ma trận  $nhân$  được biến đổi thông qua việc chuyển tọa độ từ tập huấn luyện sang tọa độ pixel của hình ảnh tương ứng trong tập huấn luyện gọi là  $ys$ . Hai ma trận  $xs$ ,  $ys$  được sử dụng trong quá trình huấn luyện, sau đó có thể dự đoán được với tập dữ liệu mới dựa trên mô hình mà chúng tạo ra.

Đối với nhãn loại 1 là tòa nhà màu sắc thường xuất hiện trên tập ảnh RGB là màu xanh dương tuy nhiên chúng có một số ngoại lệ, một số tòa nhà lại có màu đỏ, đối với ảnh M chúng thường có màu trắng nhưng một số ngoại lệ có màu cam nhạt, đối với ảnh P chúng thường có màu xám nhưng một số ngoại lệ có màu trắng, đối với ảnh A chúng thường có màu xanh đen nhưng ngoại lệ sẽ có màu trắng. Chính vì thế việc loại bỏ những ngôi nhà ngoại lệ có trong tập huấn luyện có thể giúp cho tăng kết quả cho các nhãn còn lại.



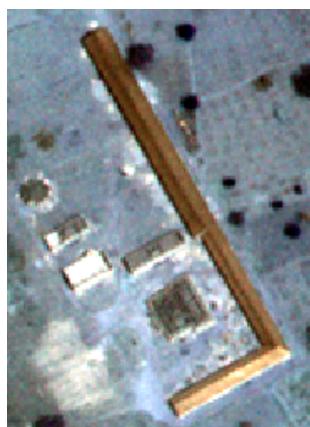
Hình 35: Ảnh RGB nhãn loại 1 thường xuất hiện



Hình 36: Ảnh RGB nhãn loại 1 bắt thường



Hình 37: Ảnh M nhãn loại 1 thường xuất hiện

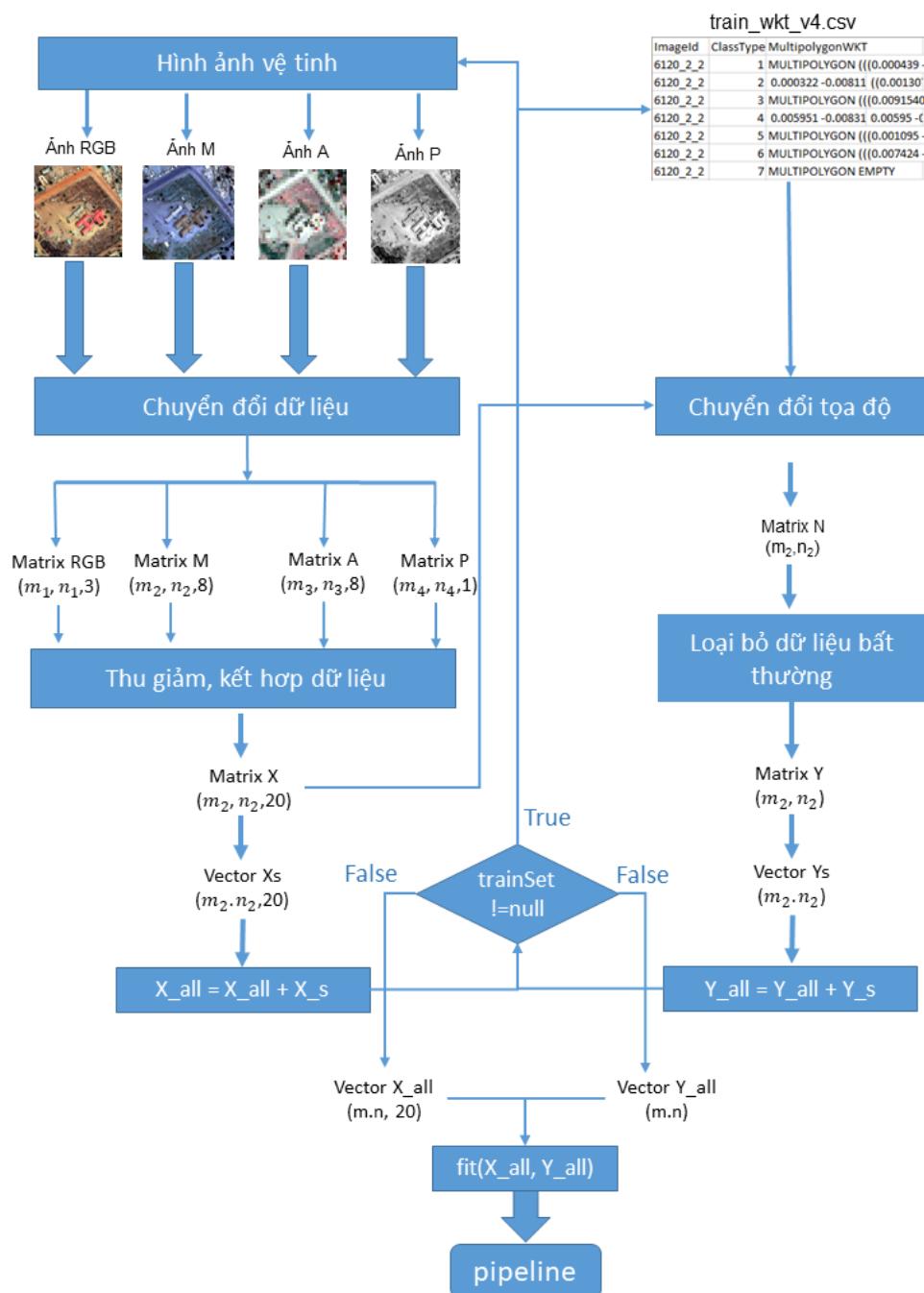


Hình 38: Ảnh M nhãn loại 1 bắt thường

### 3.3 Xây dựng mô hình phân loại dựa trên phương pháp Stochastic Gradient Descent

#### 3.3.1 Xây dựng mô hình

Quá trình xây dựng mô hình gán nhãn của luận văn được tóm tắt trong lược đồ sau:



**Hình 39:** Lược đồ tóm tắt quá trình xây dựng mô hình gán nhãn



Với hình ảnh được cung cấp từ cuộc thi, hàm `imread` trong thư viện `tiff[38]` sẽ được sử dụng trong quá trình chuyển đổi dữ liệu hình ảnh đầu vào, hàm `imread` được sử dụng như sau:

```
1     image = tiff.imread('{}.tif'.format(IM_ID+'_'+_type)).transpose([1, 2, 0])
2
```

Trong đó `IM_ID` là `ImageId` của hình ảnh, `_type` là các loại ảnh RGB, A, M, P. Đầu ra của quá trình chuyển đổi dữ liệu là bộ 4 ma trận RGB, A, M, P có kích thước khác nhau. Vì chiều dài và chiều rộng của các ma trận không chênh lệch nhau quá lớn (khảo sát ở chương 3 mục 3.1.2), nên các ma trận trên sẽ được cắt bỏ đi những cột hoặc dòng dôi ra để tạo thành ma trận vuông, điều này giúp cho quá trình thu giảm dữ liệu không làm thay đổi quá nhiều đặc trưng của ảnh.

Các ma trận RGB, A, M, P sau khi được cắt bỏ sẽ có chiều dài và chiều rộng bằng nhau, tiếp theo hàm `resize` của thư viện `cv2[39]` sẽ được sử dụng để thu giảm ma trận RGB, P và kéo dãn ma trận A về cùng kích cỡ với ma trận M, hàm `resize` được sử dụng như sau:

```
1     img_A, mask_A = imread_sixteenband(IM_ID, 'A', False)
2     size = min(get_size_tiff(img_A))
3     img_A = crop_img(img_A, size)
4     img_A_resize = cv2.resize(img_A, (xM, yM))
5
```

Đoạn code phía trên thể hiện quá trình chuyển đổi dữ liệu hình ảnh A thành ma trận A, sau khi chuyển đổi ma trận A được cắt bỏ đi những phần dôi ra và kéo dãn ma trận A về cùng kích cỡ với ma trận M.

Sau khi tất cả các ma trận RGB, A, M, P có cùng kích cỡ, quá trình kết hợp dữ liệu sẽ gộp chúng lại thành một ma trận X, ma trận này được gọi là ma trận đặc trưng. Sau đó ma trận X sẽ được chuyển đổi về dạng vector.

Từ hình ảnh M ta sẽ lấy được kích thước chiều rộng (W) và chiều cao (H), kết hợp kích thước này với thông tin trong tập tin `train_wkt_v4.csv` thông qua quá trình chuyển đổi dữ liệu huấn luyện đã nêu ở mục 3.1.4 sẽ tính được ma trận ánh xạ nhãn của hình ảnh có `ImageId` tương ứng. Ma trận này sẽ loại bỏ dữ liệu bất thường bằng cách loại đi những phần tử biên. Phần tử biên được nhận dạng dựa trên kết quả thực nghiệm sẽ nhận thấy được một số phần tử bất thường, các phần tử này sẽ được ghi lại vị trí và loại bỏ nhãn khi đưa vào lần huấn luyện tiếp theo. Phần tử biên là một số ngôi nhà (nhãn 1) có kích thước lớn và có đặc trưng khác biệt so với đa số các ngôi nhà khác (nhận xét phần 3.2). Khi áp dụng mô hình gán nhãn xây dựng cho những ngôi nhà có cùng đặc trưng như vậy trong tập kiểm tra thì đa phần mô hình gán nhầm sang những nhãn khác, vì thế những phần tử có đặc trưng như trên thay vì gán nhãn 1 chúng sẽ được gán thành những loại còn lại (nhãn 0), làm như vậy để giúp cho việc gán những nhãn còn lại chính xác hơn. Dựa vào

quá trình thực nghiệm một số nhãn  $1$  bị gán sai sẽ lưu lại, sau đó những nhãn đó sẽ được gán lại nhãn  $0$ . Hình sau là hình minh họa của ma trận nhãn trước và sau quá trình loại bỏ dữ liệu bất thường:



(a) 6140\_3\_1

(b) 6140\_3\_1\_Mask

(c) 6140\_3\_1\_No\_outliner

**Hình 41:** Hình ảnh minh họa trước và sau khi loại bỏ phần tử biên.

Kết thúc quá trình chuyển đổi, thu giảm, kết hợp dữ liệu cho ra được vector đặc trưng  $X_s$ . Kết thúc quá trình chuyển đổi tọa độ và loại bỏ dữ liệu bất thường cho ra được vector nhãn  $Y_s$ .

Vector đặc trưng  $X_s$  và vector nhãn  $Y_s$  vừa tính được sẽ được cộng dồn vào vector  $X_{all}$  và vector  $Y_{all}$ . Lặp lại quá trình trên cho đến khi tất cả các hình ảnh trong tập huấn luyện được chuyển đổi thành vector đặc trưng và vector nhãn, chúng sẽ được cộng dồn vào  $X_{all}$  và  $Y_{all}$ . Hai vector  $X_{all}$  và  $Y_{all}$  này sẽ được sử dụng để huấn luyện.

Luận văn này sử dụng hai giải thuật Logistic(Log) và Support Vector Machine(SVM) cho quá trình huấn luyện, phương pháp Stochastic Gradient Descent(SGD) được sử dụng để tối ưu hàm chi phí cho hai giải thuật trên.

Luận văn này sử dụng bộ thư viện Sklearn[40] với các thư viện: make\_pipeline[41], StandardScaler[42], SGDClassifier[43] cho quá trình huấn luyện. Cách sử dụng thư viện như sau:

```

1     pipeline = make_pipeline(StandardScaler(),SGDClassifier(loss='log',
2     warm_start=True,n_job = -1))
3     pipeline.fit(X_all,Y_all)
4

```

Trong đó hàm `make_pipeline` dùng để gói gọn các bước chuẩn hóa dữ liệu, xây dựng mô hình vào một đường ống. Dòng 1 liệt kê các giải thuật sẽ được sử dụng đối với dữ liệu huấn luyện, quá trình diễn ra lần lượt là: chuẩn hóa dữ liệu(`StandardScaler()`) sau đó áp dụng phương pháp SGD cho giải thuật Logistic(`log`) khi quá trình huấn luyện diễn ra. Có thể thay thế giải thuật Logistic thành giải thuật Support Vector Machine bằng cách đổi `loss='log'` thành `loss='hinge'`. Các quá trình trên sẽ được áp dụng cho tập dữ liệu huấn luyện  $X_{all}$  và  $Y_{all}$  thông qua câu lệnh `pipeline.fit(X_all, Y_all)`.



Vì tập dữ liệu nhỏ và hình ảnh trong tập dữ có nhiều đặc trưng khác nhau nên nhóm quyết định chia 25 hình ảnh của tập dữ liệu thành hai phần là tập huấn luyện<sup>20</sup> và tập kiểm tra<sup>21</sup>, không có tập đánh giá<sup>22</sup>. Trong đó số hình ảnh huấn luyện sẽ chiếm tỉ lệ cao hơn, chia theo tỉ lệ 'huấn luyện : kiểm tra' là '80 : 20' tương ứng với là 20 hình ảnh huấn luyện, 5 hình ảnh kiểm tra. Hình ảnh kiểm tra được lựa chọn thông qua quá trình khảo sát bên dưới.

Sau khi khảo sát chúng tôi nhận thấy được có thể chia tập hình ảnh ra làm 5 nhóm, những hình ảnh trong cùng một nhóm thì có chứa đặc trưng giống nhau. Và 5 hình ảnh đặc trưng trong từng nhóm sẽ được lựa chọn làm tập kiểm tra, 5 nhóm đó là:

Nhóm	Nhận đặc trưng	Nhận xét về mật độ
1	1, 3, 5, 6, 8	Phần lớn là nhãn 1 và nhãn 6, chúng tập trung thành cụm, có một số vùng của nhãn 1 có đặc trưng khác biệt lớn so với những vùng còn lại(phần tử biên). Nhãn 8 và nhãn 5 phân bố rải rác. Một số nhãn 4 xen lẫn trong nhãn 1.
2	4, 5, 6	Đa phần là loại 6. Loại 4, 5 xen lẫn với loại 6, các đặc trưng giữa các loại phân biệt rõ ràng.
3	1, 3, 4, 5, 6 , 7, 8	Loại 1 và 6 chiếm đa số. Đặc trưng của loại 5, 6, 8 gần giống nhau.
4	5	Loại 5 gần như chiếm hoàn toàn nhưng khó phân biệt được loại 5 và các loại còn lại.
5	4,5	Loại 4 và 5 chiếm đa số, tuy nhiên đặc trưng của loại 4 và 5 không rõ ràng, khó mà phân biệt được.

Bảng 1: Khảo sát tập dữ liệu hình ảnh bằng mắt thường.

Chú thích:

- |                         |                   |
|-------------------------|-------------------|
| (1) Nhà                 | (6) Thảm thực vật |
| (2) Công trình nhân tạo | (7) Nước chảy     |
| (3) Đường lớn           | (8) Nước đứng     |
| (4) Đường nhỏ           | (9) Xe lớn        |
| (5) Cây                 | (10) Xe nhỏ       |

<sup>20</sup> Thuật ngữ tiếng Anh: Training dataset

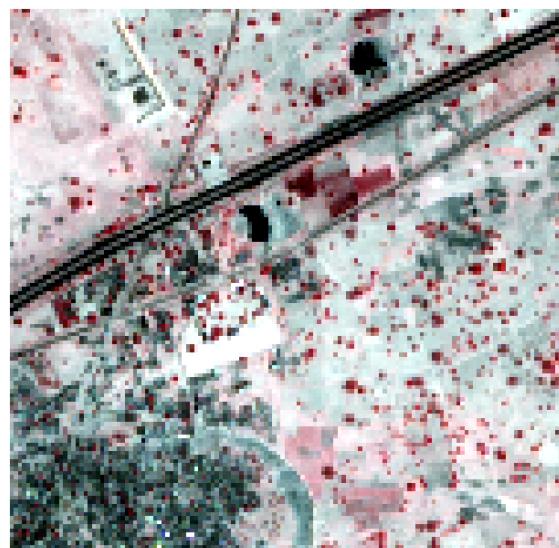
<sup>21</sup> Thuật ngữ tiếng Anh: Test dataset

<sup>22</sup> Thuật ngữ tiếng Anh: Validation dataset

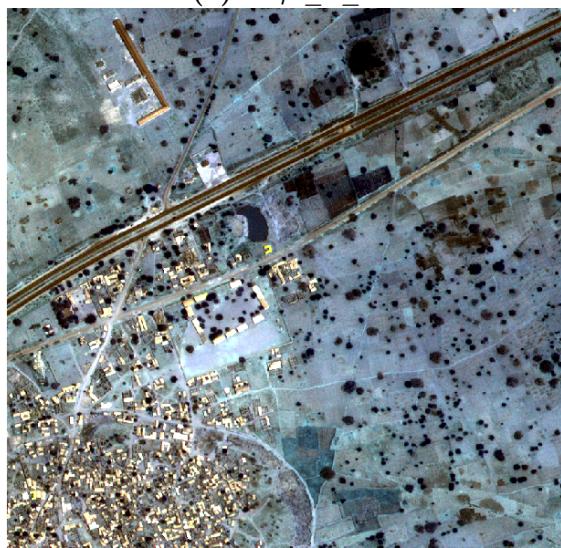
Hình ảnh đại diện cho nhóm 1:



(a) 6140\_3\_1



(b) 6140\_3\_1\_A



(c) 6140\_3\_1\_M

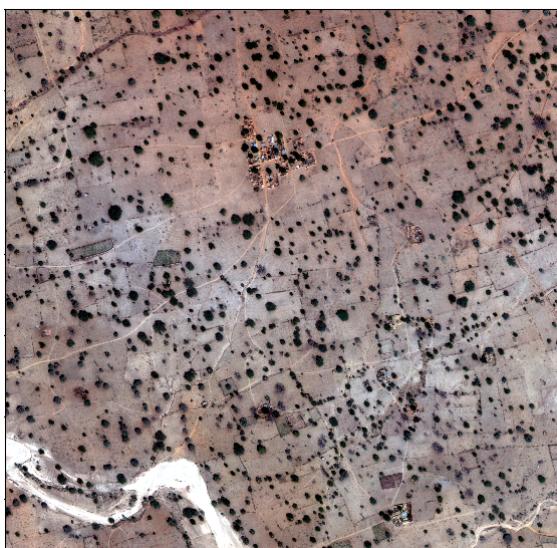


(d) 6140\_3\_1\_Mask

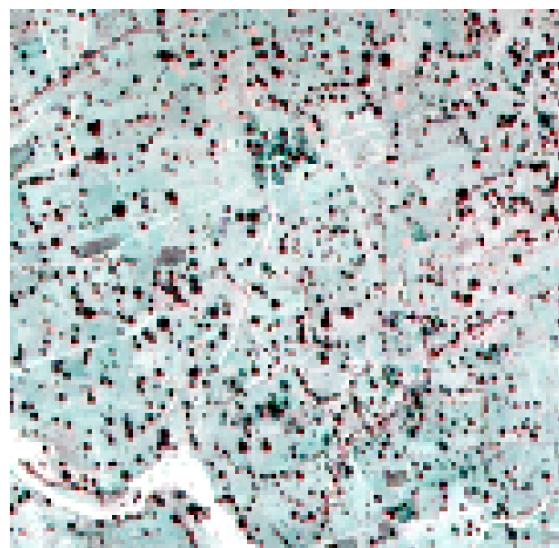
Hình 43: Nhóm 1.

**Dánh giá:** Nhìn chung các ảnh nhóm này có nhiều loại nhãn khác nhau và đa phần có thể nhận dạng được. Các nhãn có thể nhận dạng được chiếm phần lớn trong hình. Tuy nhiên có một số nhỏ nhãn như nước đứng (nhãn 8), đường lớn (nhãn 3), đường nhỏ (nhãn 4) chúng có đặc trưng tương tự cây (nhãn 5) và các loại khác (không thuộc 10 nhãn) làm cho việc nhận dạng khó hơn.

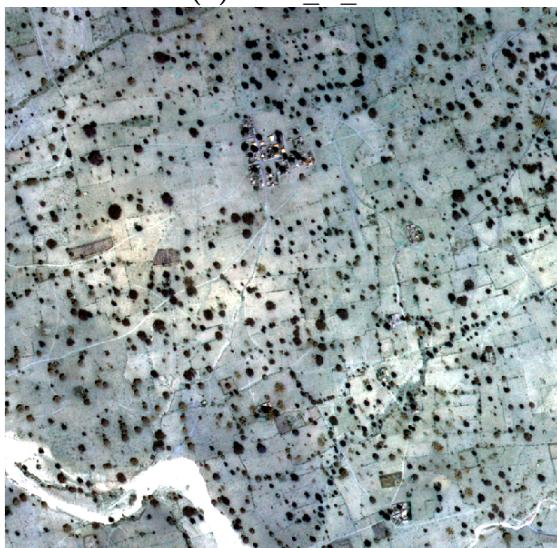
Hình ảnh đại diện cho nhóm 2:



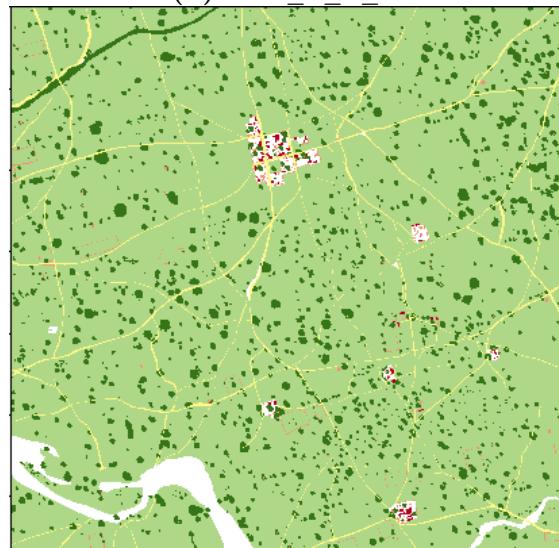
(a) 6060\_2\_3



(b) 6060\_2\_3\_A



(c) 6060\_2\_3\_M



(d) 6060\_2\_3\_Mask

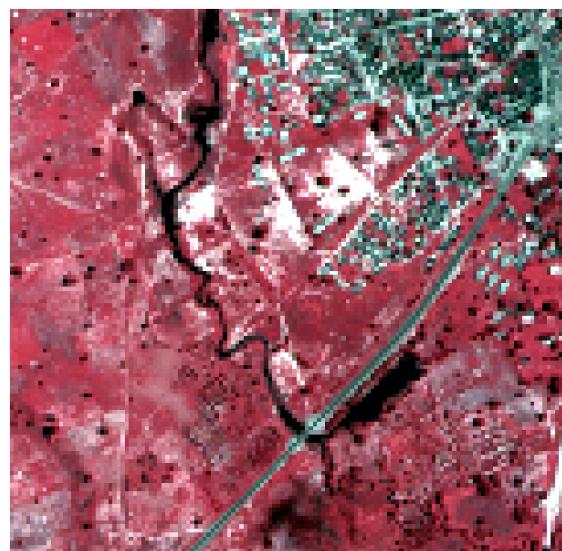
**Hình 45: Nhóm 2.**

**Đánh giá:** Các ảnh trong nhóm này chứa ít nhãn, chủ yếu là thảm thực vật (nhãn 6) và cây (nhãn 5), đa phần nhóm này có thể nhận dạng được với tỉ lệ cao, ít có nhầm lẫn với các loại nhãn khác.

Hình ảnh đại diện cho nhóm 3:



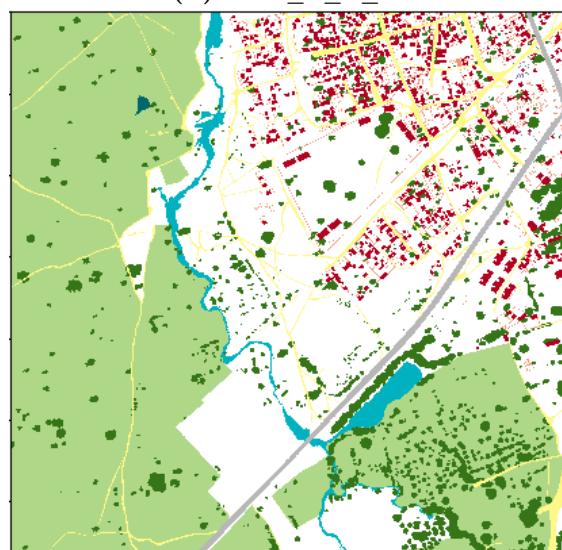
(a) 6100\_2\_2



(b) 6100\_2\_2\_A



(c) 6100\_2\_2\_M



(d) 6100\_2\_2\_Mask

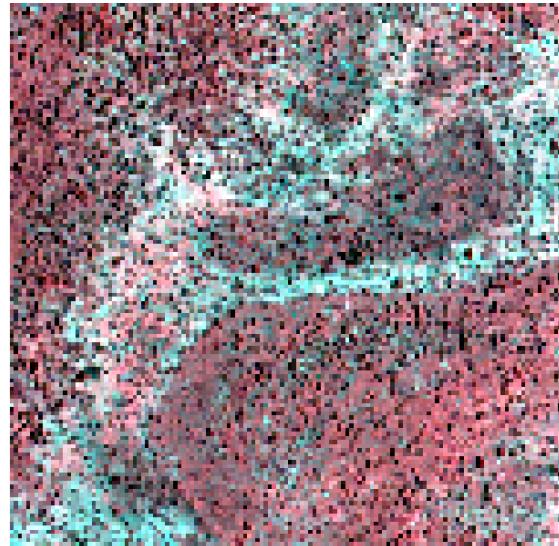
Hình 47: Nhóm 3.

**Đánh giá:** Thảm thực vật (nhãn 6) của nhóm này có đặc trưng gần giống cây (nhãn 5), hai nhãn này khó phân biệt được. Nước đứng (nhãn 8) có đặc trưng khác so với các nhãn còn lại nên có thể phân biệt được.

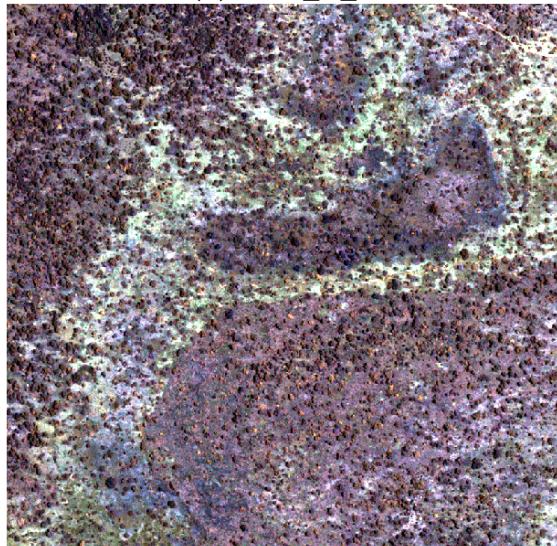
Hình ảnh đại diện cho nhóm 4:



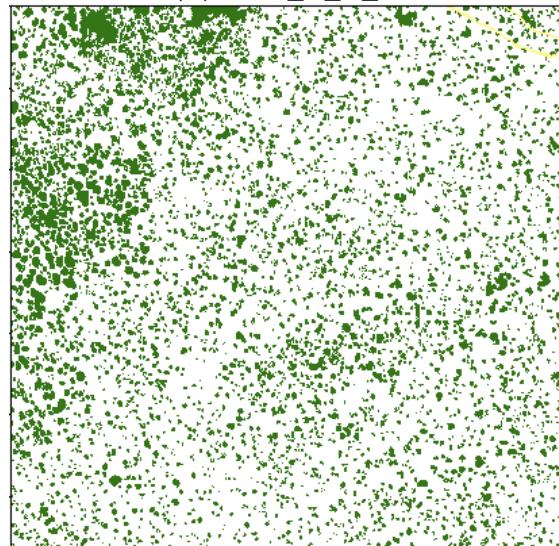
(a) 6170\_4\_1



(b) 6170\_4\_1\_A



(c) 6170\_4\_1\_M

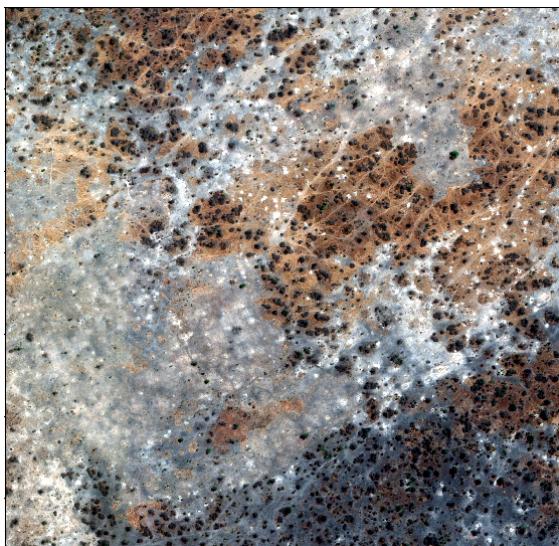


(d) 6170\_4\_1\_Mask

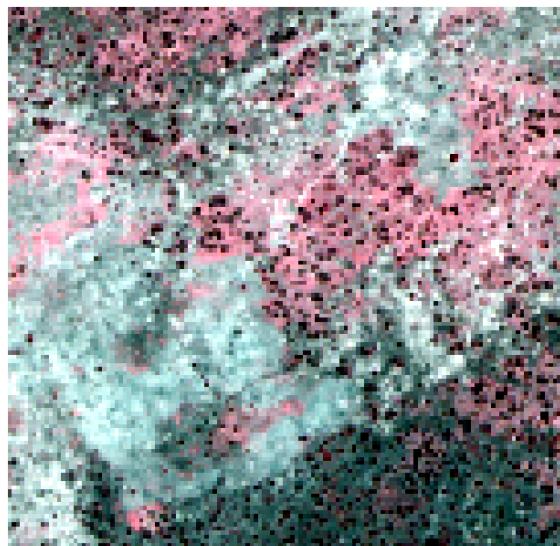
**Hình 49:** Nhóm 4.

**Dánh giá:** Nhóm 4 có những vùng mặc dù không phải là thảm thực vật (nhãn 6) hoặc cây (nhãn 5) nhưng lại có một vài đặc trưng gần giống với thảm thực vật và cây như màu xanh ở ảnh RGB, màu đỏ ở ảnh A. Điều đó làm cho việc gán nhãn nhóm này trở nên khó khăn hơn.

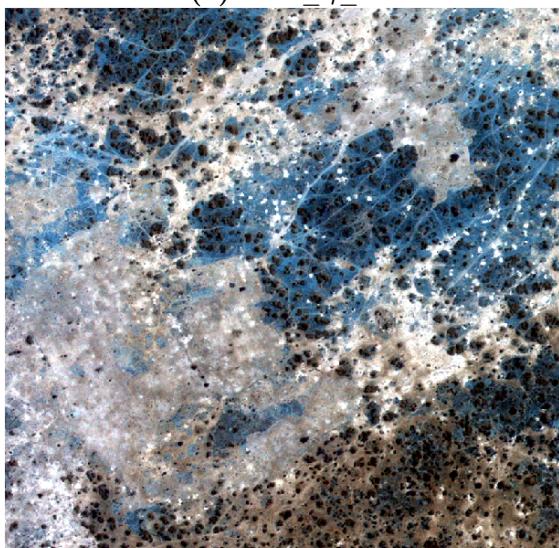
Hình ảnh đại diện cho nhóm 5:



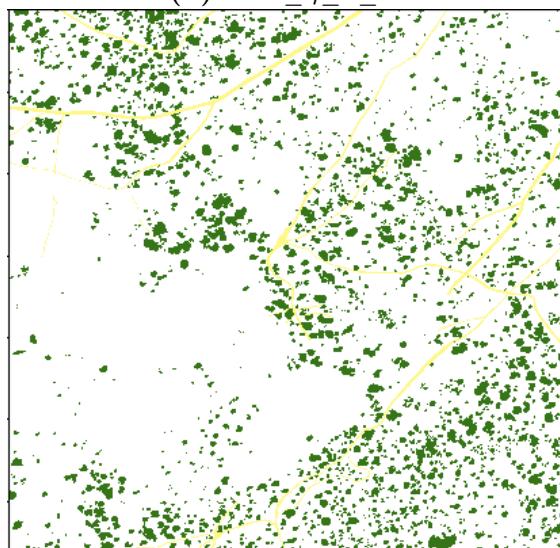
(a) 6010\_4\_2



(b) 6010\_4\_2\_A



(c) 6010\_4\_2\_M



(d) 6010\_4\_2\_Mask

**Hình 51: Nhóm 5.**

**Dánh giá:** Mặc dù nhóm 5 có cây (nhãn 5) nhưng đặc trưng của cây trong nhóm này lại khác so với cây những nhóm còn lại, điển hình như: trong hình ảnh RGB cây có màu đen xám trong khi đó ở các nhóm còn lại cây lại có màu xanh đậm, trong hình ảnh A đa phần cây có mày đen trong khi đó phần lớn cây có màu đỏ. Ở nhóm này đường nhỏ (nhãn 4) có đặc trưng riêng biệt hoàn toàn so với các nhóm còn lại. Điều này làm cho việc nhận dạng hình ảnh nhóm này cực kỳ khó khăn.

Sau khi áp dụng mô hình gán nhãn xây dựng được cho tập kiểm tra thì sẽ cho ra được 5 ma trận nhãn tương ứng với 5 hình trong tập kiểm tra, mỗi phần tử trong ma trận là



đại diện cho nhãn dự đoán được tại pixel đó. Độ đo Jaccard sẽ được sử dụng để đánh giá hiệu quả của mô hình vừa được xây dựng dựa vào 5 ma trận nhãn dự đoán được và 5 ma trận nhãn kết quả. Độ đo Jaccard trung bình được tính như sau: với mỗi loại nhãn  $x$  của mỗi hình ảnh thực hiện việc tính toán TP, FP và FN. Trong đó TP là tổng số pixel thuộc nhãn  $x$  được gán chính xác nhãn  $x$ , FP là số pixel không thuộc nhãn  $x$  bị gán nhầm là nhãn  $x$ , FN là tổng số pixel nhãn  $x$  bị gán nhầm. Cách tính TP, FP, FN được hiện thực như sau:

```
1     tp, fp, fn = (( predict & result).sum(),
2     ( predict & ~result).sum(),
3     (~predict & result).sum())
4
```

Trong đó  $predict$  là ma trận nhãn dự đoán được,  $result$  là ma trận nhãn kết quả, sau khi có được TP, FP, FN của mỗi nhãn cho từng hình sẽ tính tổng TP, tổng FP, và tổng FN của tất cả các hình ảnh, độ đo Jaccard của một nhãn được tính theo công thức (1) với TP, FP, FN là kết quả vừa tính được. Sau đó lấy trung bình tất cả các Jaccard của 10 nhãn sẽ có được độ đo Jaccard trung bình của bài toán.

### 3.3.2 Kết quả đạt được

Trong phần này, chúng tôi trình bày kết quả thực nghiệm của các giải thuật trên cùng một tập dữ liệu. Hai giải thuật nhóm đã thực hiện bao gồm Logistic Regression (Log) và Support Vector Machine (SVM). Nhóm đã sử dụng phương pháp Stochastic Gradient Descent (SGD) để tối ưu hóa hàm chi phí của hai giải thuật Logistic Regression và Support Vector Machine.

Quá trình huấn luyện và kiểm tra được thực hiện trên máy tính có cấu hình:

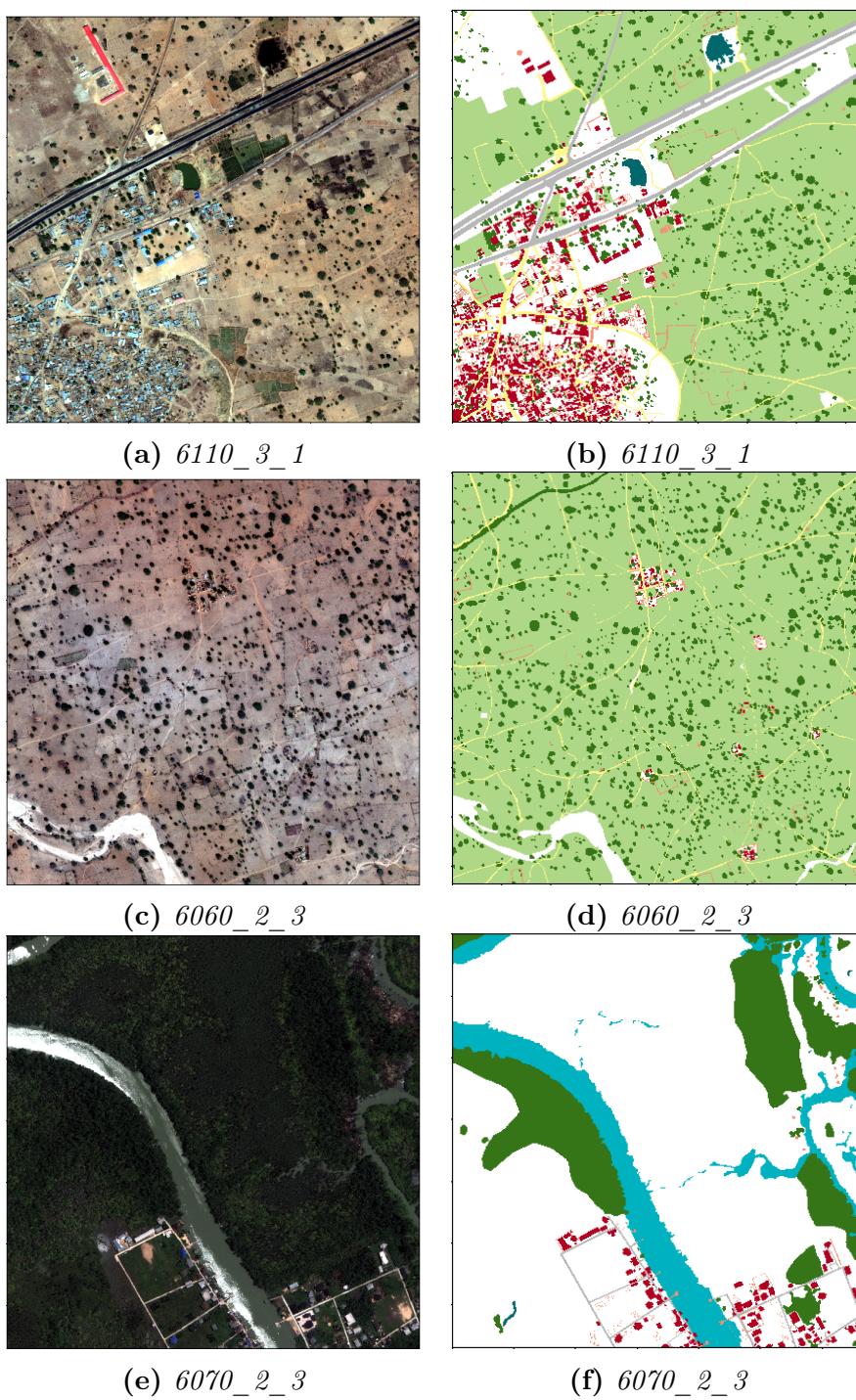
- Hệ điều hành: Ubuntu 16.04 64-bit
- CPU: Intel® Core™ i7-3537U CPU @ 2.00GHz × 4
- Ram: 8GB

Sau khi áp dụng phương pháp SGD với tập huấn luyện là vector đặc trưng  $X_{all}$  và vector nhãn  $Y_{all}$  cho hai mô hình Logistic Regression và Support Vector Machine chúng tôi thu được kết quả sau:

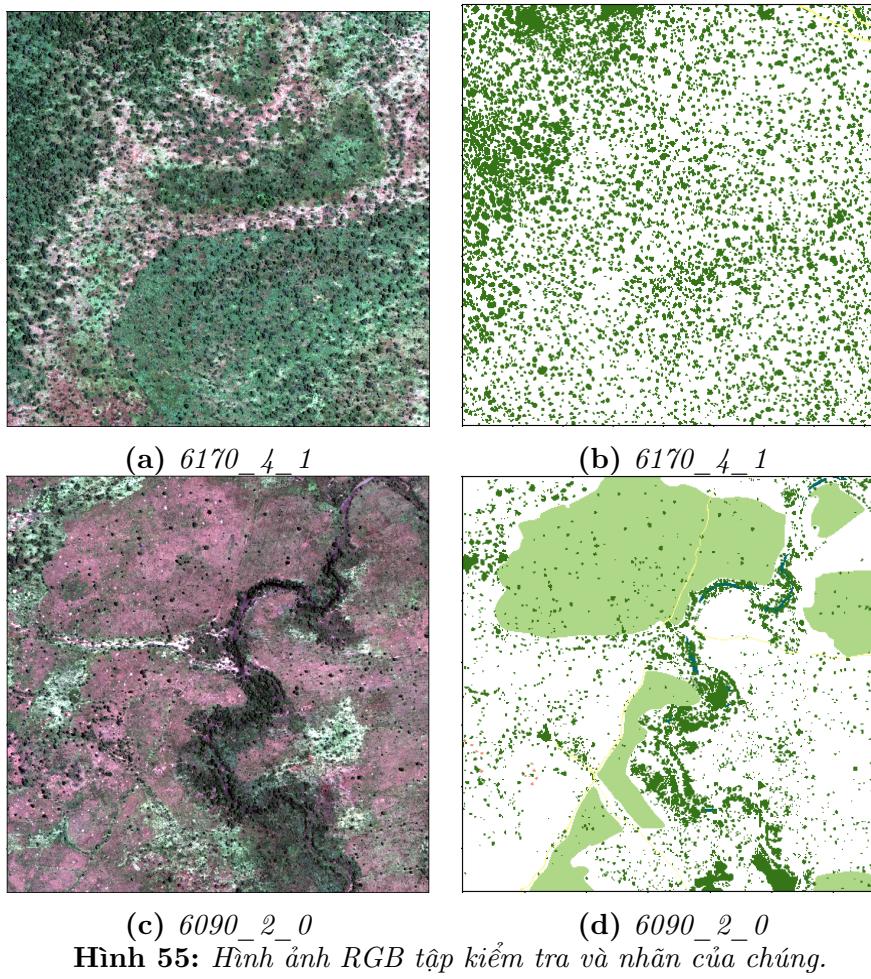
Hàm	Jaccard trung bình	Thời gian huấn luyện(s)	Thời gian kiểm tra(s)
Logistic	0.176557149139	594.6229	89.8820
SVM	0.135468646038	429.9821	91.4178

Bảng 2: So sánh độ chính xác của giải thuật Log và SVM

Bảng 2 là kết quả khi áp dụng hai mô hình Log và SVM cho tập kiểm tra bệnh dưới, trong đó mỗi hình ảnh là đại diện cho từng nhóm trong 5 nhóm nêu ở phần trên.  
Dưới đây là hình ảnh RGB của tập kiểm tra và nhãn thật của chúng.



Hình 53: Hình ảnh RGB tập kiểm tra và nhãn của chúng.



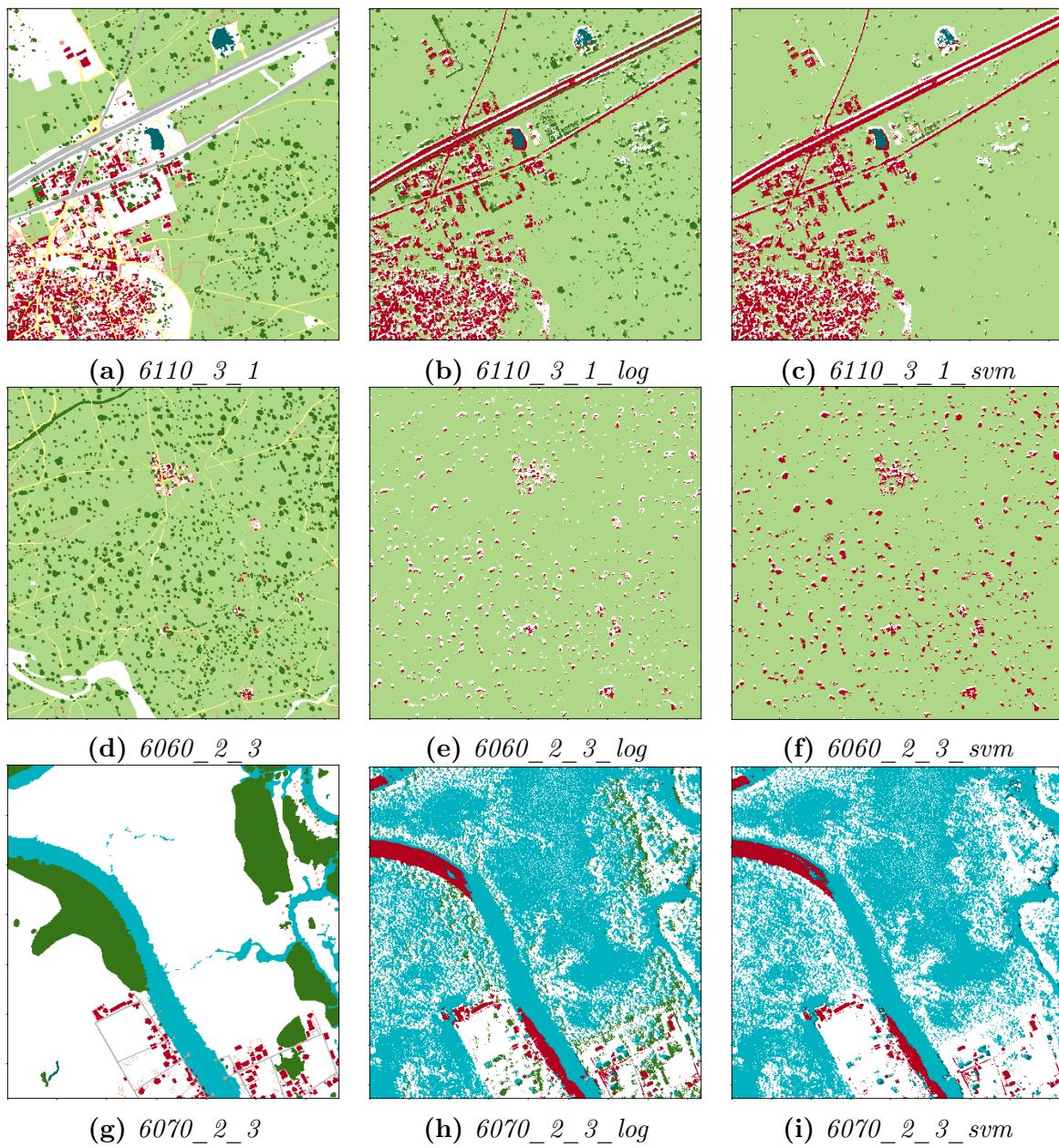
Hình 55: Hình ảnh RGB tập kiểm tra và nhãn của chúng.

Chú thích:

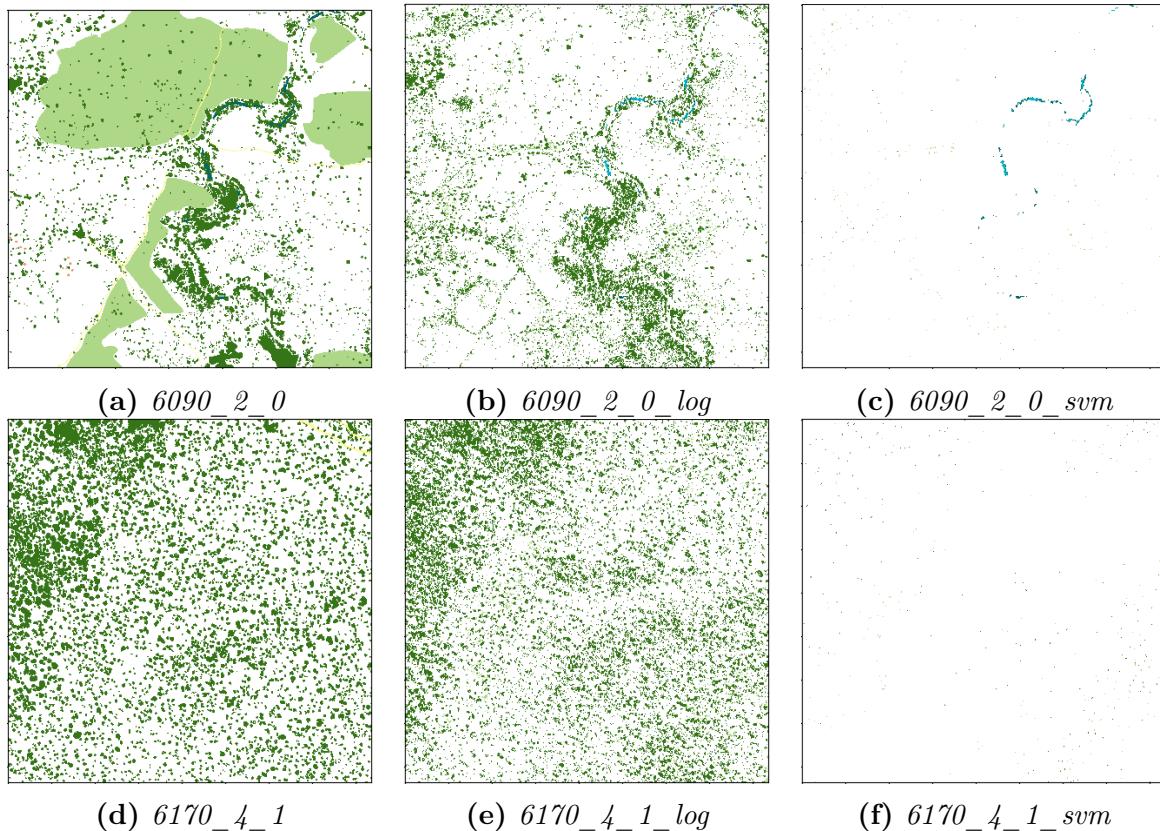
	1.Nhà		6.Thảm thực vật
	2.Công trình nhân tạo		7.Sông
	3.Đường lớn		8.Nước đứng
	4.Đường nhỏ		9.Xe lớn
	5.Cây		10.Xe nhỏ

Hình 56: Màu đại diện cho các nhãn

Hình ảnh bên dưới hiển thị cho nhãn gốc (bên trái) và nhãn dự đoán được khi áp dụng mô hình Logistic Regression (ở giữa) và Support Vector Machine (bên phải):



**Hình 58:** Hình ảnh dự đoán được khi áp dụng Log (ở giữa) và SVM (bên phải).



**Hình 60:** Hình ảnh dự đoán được khi áp dụng Log (ở giữa) và SVM (bên phải).

**Nhận xét:** Các bảng 2 và hình 58, hình 60 so sánh kết quả gán nhãn của các hình trên sử dụng cùng một điều kiện và thông số huấn luyện. Có thể thấy được mô hình phân loại Logistic Regression cho ra kết quả tốt hơn so với mô hình phân loại Support Vector Machine. Chúng tôi đã thực hiện hai giải thuật trên với nhiều bộ kiểm tra khác nhau, và kết quả đạt được là đa số Logistic Regression phân loại tốt hơn so với mô hình Support Vector Machine. Tuy nhiên đối với mô hình Support Vector Machine lại có thời gian chạy nhanh hơn rất nhiều so với mô hình Logistic Regression.

Nhìn chung cả hai mô hình đều dự đoán được một vài nhãn có đặc trưng rõ ràng như: nhà, cây, thảm thực vật, nước đứng, nước chảy. Những nhãn có dữ liệu huấn luyện ít và có đặc trưng tương đồng như với nhiều nhãn khác như: nước đứng, đường, xe cộ, công trình nhân tạo thì khó nhận dạng và bị gán nhầm sang nhãn khác.



## 4 Kết luận

### 4.1 Kết quả đạt được

Qua đề tài luận văn này, chúng tôi đã nghiên cứu và hiện thực thành công mô hình gán nhãn ảnh vệ tinh dựa trên hai giải thuật Logistic Regression và Support Vector Machine. Thông qua ảnh vệ tinh đã cho chúng tôi đã trích suất, biến đổi chúng thành dạng dữ liệu máy tính có thể học được, tạo tiền đề để áp dụng các giải thuật học máy khác.

Bên cạnh đó nhóm còn tìm hiểu một số giải thuật khai phá dữ liệu và hiện thực chúng.

### 4.2 Hạn chế

Mặc dù mô hình có thể dự đoán được một số nhãn có đặc trưng riêng biệt và dễ nhận dạng nhưng độ chính xác vẫn chưa cao, khó có thể áp dụng vào thực tế được. Một hạn chế nữa của Logistic Resgression là nó yêu cầu các điểm dữ liệu được tạo ra một cách độc lập với nhau. Trên thực tế, các điểm dữ liệu có thể bị ảnh hưởng bởi nhau. Ví dụ: cùng là nhãn cây (nhãn 5) nhưng có vùng thì có đặc trưng giống nhãn thảm thực vật (nhãn 6), vùng khác thì giống đặc trưng giống nhãn nước đứng (nhãn 8).

Bên cạnh đó còn những hạn chế khác như việc tập dữ liệu huấn luyện phân bố không đồng đều cho tất cả các nhãn cũng làm ảnh hưởng đến kết quả của việc gán nhãn, ví dụ: số lượng thảm thực vật (nhãn 6), cây (nhãn 5) chiếm tỉ lệ quá lớn so với những nhãn như: xe lớn (nhãn 9), xe nhỏ (nhãn 10), nước đứng (nhãn 8). Điều này dẫn đến kết luận rằng việc đào tạo một mô hình riêng lẻ cho từng nhãn sẽ làm việc tốt hơn nhiều so với một mô hình dự đoán tất cả các nhãn cùng một lúc.

### 4.3 Hướng phát triển

Với những hạn chế nêu trên trong tương lai nhóm sẽ cải thiện hệ thống như sau:

- Xây dựng riêng lẻ các bộ phân loại cho từng nhãn, kết hợp các bộ phân loại này với nhau để giúp quá trình gán nhãn cho hiệu suất cao nhất.
- Mở rộng các nhãn có thể nhận dạng được.
- Phát triển thành ứng dụng web.



## 5 Tài liệu tham khảo

### Tài liệu

- [1] Bài toán phân lớp trong Machine Learning (Classification in Machine Learning). [Trực tuyến]. Available:: <http://eitguide.net/bai-toan-phan-lop-trong-machine-learning-classification-machine-learning/>. [Đã truy cập 10 December 2017]
- [2] Mahout - Machine Learning. [Trực tuyến]. Available:: <https://www.tutorialspoint.com/mahout/mahout-machine-learning.htm>. [Đã truy cập 10 December 2017]
- [3] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.330-332.
- [4] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.331.
- [5] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.350-351.
- [6] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.408-410.
- [7] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.409.
- [8] Phân nhóm các thuật toán Machine Learning. [Trực tuyến]. Available:: <https://machinelearningcoban.com/2016/12/27/categories/>. [Đã truy cập 10 December 2017]
- [9] MNIST dataset introduction. [Trực tuyến]. Available:: <http://corochann.com/mnist-dataset-introduction-1138.html>. [Đã truy cập 10 December 2017]
- [10] Some samples of the MNIST database. [Trực tuyến]. Available:: <https://www.researchgate.net/figure/252028600-fig2-Figure-3-Some-samples-of-the-MNIST-database>. [Đã truy cập 10 December 2017]
- [11] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.85-87.
- [12] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.87.
- [13] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.88-89.



- [14] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.544-553.
- [15] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.544-553.
- [16] DBSCAN. [Trực tuyến]. Available:: <https://en.wikipedia.org/wiki/DBSCAN>. [Đã truy cập 12 December 2017]
- [17] What are outliers in the data? . [Trực tuyến]. Available:: <http://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm>. [Đã truy cập 12 December 2017]
- [18] What are outliers in the data? . [Trực tuyến]. Available:: <http://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm>. [Đã truy cập 12 December 2017]
- [19] What are outliers in the data? . [Trực tuyến]. Available:: <http://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm>. [Đã truy cập 12 December 2017]
- [20] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.93-99.
- [21] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.94-97.
- [22] Principal Component Analysis (phần 1/2). [Trực tuyến]. Available:: <https://machinelearningcoban.com/2017/06/15/pca/>. [Đã truy cập 10 December 2017]
- [23] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.112-115.
- [24] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.327-328.
- [25] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.328.
- [26] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.328.
- [27] Overfitting. [Trực tuyến]. Available: <https://machinelearningcoban.com/2017/03/04/overfitting/>. [Đã truy cập 14 December 2017]
- [28] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.364-369.



- [29] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.365.
- [30] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.377-382.
- [31] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011, p.378.
- [32] Support Vector Machine. [Trực tuyến]. Available:: <https://machinelearningcoban.com/2017/04/09/sm/>. [Đã truy cập 13 December 2017]
- [33] Support Vector Machine. [Trực tuyến]. Available:: <https://machinelearningcoban.com/2017/04/09/sm/>. [Đã truy cập 13 December 2017]
- [34] Support Vector Machine. [Trực tuyến]. Available:: <https://machinelearningcoban.com/2017/04/09/sm/>. [Đã truy cập 13 December 2017]
- [35] Support Vector Machine. [Trực tuyến]. Available:: <https://machinelearningcoban.com/2017/04/09/sm/>. [Đã truy cập 13 December 2017]
- [36] Logistic Regression . [Trực tuyến]. Available:: <https://machinelearningcoban.com/2017/01/27/logisticregression/>. [Đã truy cập 13 December 2017]
- [37] Logistic Regression . [Trực tuyến]. Available:: <https://machinelearningcoban.com/2017/01/27/logisticregression/>. [Đã truy cập 13 December 2017]
- [38] Read file Tiff. [Trực tuyến] . Available :  
<http://scikit-image.org/docs/dev/api/skimage.external.tifffile.html>. [Đã truy cập 15 December 2017]
- [39] Resizes an image. Available :  
[https://docs.opencv.org/2.4/modules/imgproc/doc/geometric\\_transformations.html](https://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html)[Đã truy cập 15 December 2017]
- [40] Introduction Sklearn. Available : [http://scikit-learn.org/stable/supervised\\_learning.html#supervised-learning](http://scikit-learn.org/stable/supervised_learning.html#supervised-learning)[Đã truy cập 15 December 2017]
- [41] Using pipeline in sklearn. Available :  
[http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.make\\_pipeline.html](http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.make_pipeline.html). [Đã truy cập 15 December 2017]



[42] Sklearn preprocessing. Available :

<http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> [Đã truy cập 15 December 2017]

[43] Linear classifiers (SVM, logistic regression, a.o.) with SGD training. Available:

[http://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.SGDClassifier.html](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html) [Đã truy cập 15 December 2017]