TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN KỸ THUẬT DỮ LIỆU

LẠI HỮU TRÁC : 19133059

NGUYỄN DUY PHƯỚC : 19133003

Đề Tài:

TÌM HIỂU VỀ EXPLAINABLE AI VÀ ỨNG DỤNG

TIỂU LUẬN CHUYÊN NGÀNH

GIÁO VIÊN HƯỚNG DẪN

ThS. QUÁCH ĐÌNH HOÀNG

Hồ Chí Minh, 2022

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN KỸ THUẬT DỮ LIỆU

LẠI HỮU TRÁC : 19133059

NGUYỄN DUY PHƯỚC : 19133003

Đề Tài:

TÌM HIỂU VỀ EXPLAINABLE AI VÀ ỨNG DỤNG

TIỂU LUẬN CHUYÊN NGÀNH

GIÁO VIÊN HƯỚNG DẪN

ThS. QUÁCH ĐÌNH HOÀNG

Hồ Chí Minh, 2022

ĐH SƯ PHẠM KỸ THUẬT TPHCM
KHOA CNTT
****

XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
****

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên 1: Lại Hữu Trác MSSV: 19133059

Họ và tên sinh viên 2: Nguyễn Duy Phước MSSV: 19133003

Ngành: Kỹ thuật dữ liệu

Tên đề tài: Tìm hiểu về Explainable AI và ứng dụng

Họ và tên giáo viên hướng dẫn: ThS.Quách Đình Hoàng

NHẬN XÉT:

1. Về nội dung và đề tài khối lượng thực hiện:

.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....

3. Khuyết điểm:

.....
.....
.....
.....

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

Tp.Hồ Chí Minh, ngày...tháng...năm 2022

Giáo viên hướng dẫn

Ký & ghi rõ họ tên

ĐH SƯ PHẠM KỸ THUẬT TPHCM
KHOA CNTT
****

XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
****

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên sinh viên 1: Lại Hữu Trác MSSV: 19133059

Họ và tên sinh viên 2: Nguyễn Duy Phước MSSV: 19133003

Ngành: Kỹ thuật dữ liệu

Tên đề tài: Tìm hiểu về Explainable AI và ứng dụng

Họ và tên giáo viên phản biện:

NHẬN XÉT:

1. Về nội dung và đề tài khối lượng thực hiện:

.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....

3. Khuyết điểm:

.....
.....
.....
.....

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

Tp.Hồ Chí Minh, ngày...tháng...năm 2022

Giáo viên phản biện

Ký & ghi rõ họ tên

LỜI CẢM ƠN

Trong quá trình nghiên cứu đề tài, các giảng viên đã luôn hỗ trợ, hướng dẫn sinh viên, với tất cả sự kính trọng, chúng tôi xin được bày tỏ lòng biết ơn đến quý Thầy Cô đã luôn theo dõi và hướng dẫn trong suốt thời gian thực hiện đề tài.

Đầu tiên, chúng tôi xin gửi lời cảm ơn sâu sắc nhất đến Ban giám hiệu trường Đại học Sư phạm Kỹ Thuật Thành phố Hồ Chí Minh đã tạo điều kiện, môi trường học tập chất lượng, hiệu quả để chúng tôi có thể phát huy một cách tốt nhất việc nghiên cứu đề tài.

Đồng thời, chúng tôi xin gửi lời cảm ơn đến Ban chủ nhiệm khoa Công nghệ Thông tin và các Thầy Cô khoa Công nghệ Thông tin - Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã tạo môi trường học tập và làm việc chuyên nghiệp, nhiệt tình giảng dạy để chúng tôi thực hiện tốt đề tài nói riêng và sinh viên trong khoa Công nghệ Thông tin nói chung trong quá trình học tập và làm việc tại trường.

Đặc biệt, chúng tôi xin gửi lời cảm ơn chân thành nhất đến Thầy **Quách Đình Hoàng** – Giáo viên hướng dẫn tiểu luận chuyên ngành – Khoa Công nghệ Thông tin – Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh, đã hướng dẫn, quan tâm, góp ý và luôn đồng hành cùng chúng tôi trong những giai đoạn khó khăn nhất của đề tài.

Tuy nhiên vì thời gian hoàn thành đề tài ngắn, nên đề tài khó lòng tránh khỏi những sai sót và hạn chế nhất định. Kính mong nhận được những phản hồi, đóng góp ý kiến và chỉ bảo thêm từ Quý Thầy Cô, để chúng tôi có thể đạt được những kiến thức hữu ích, nâng cao trình độ để phục vụ cho sự nghiệp sau này.

Xin chân thành cảm ơn

KẾ HOẠCH THỰC HIỆN

Tuần	Thời gian	Nội dung công việc	Ghi chú
Tuần 3	05/9 - 11/9	Lựa chọn và xác định đề tài tiểu luận chuyên ngành	
Tuần 4	12/9 - 18/9	Tìm hiểu sơ lược về đề tài	
Tuần 5	19/9 – 25/9	Tìm hiểu sơ lược về XAI	
Tuần 6	26/9 - 2/10	Tìm hiểu về LIME	
Tuần 7	3/10 - 9/10	Tìm hiểu về SHAP	
Tuần 8, 9	10/10 - 23/10	Tìm hiểu về Anchors	

Tuần 10, 11	24/10 - 6/11	Tìm hiểu về phương pháp ICE	
Tuần 12	7/11 - 13/11	Tìm hiểu phương pháp Feature Interaction, Partial Dependence Plot (PDP)	
Tuần 13	14/11 - 20/11	Tìm hiểu về các công trình nổi trội cho XAI	
Tuần 14	21/11 - 27/11	Thực hiện tìm hiểu, xây dựng mô hình demo, hoàn thiện báo cáo	
Tuần 15	28/11 - 4/12	Thực hiện xây dựng mô hình demo, hoàn thiện báo cáo	
Tuần 16	5/12 - 11/12	Thực hiện xây dựng mô hình demo, hoàn thiện báo cáo	
Tuần 17	12/12 - 18/12	Hoàn thiện báo cáo	

MỤC LỤC

DANH SÁCH HÌNH VẼ

CHƯƠNG 1: MỞ ĐẦU

1.1 TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Trí tuệ nhân tạo (AI) trong nhiều năm qua chủ yếu là lĩnh vực được tập trung nhiều trên lý thuyết, không có nhiều ứng dụng tác động trong thế giới thực. Điều này đã thay đổi hoàn toàn trong thập kỷ như sự kết hợp của các máy móc mạnh mẽ hơn, các thuật toán học tập được cải thiện cũng như khả năng truy cập dễ dàng hơn vào lượng dữ liệu khổng lồ đã kích hoạt các tiến bộ trong học máy (ML) và dẫn đến việc ứng dụng rộng rãi trong công nghiệp [1]. Khoảng năm 2012, phương pháp học sâu (DL) [2] bắt đầu thống trị các điểm chuẩn về độ chính xác, đạt được thành tích vượt bậc và ngày càng hoàn thiện trong những năm tiếp theo. Kết quả là ngày nay, rất nhiều vấn đề trong thế giới thực trong các lĩnh vực khác nhau, trải dài từ bán lẻ và ngân hàng đến y học được giải quyết trong khi sử dụng các mô hình học máy. Những công cụ AI này có thể tạo ra kết quả chính xác cao, nhưng một số cũng rất phức tạp. Sự phức tạp này đã khiến các nhà nghiên cứu và các nhà hoạch định chính sách đặt câu hỏi: làm sao để biết tại sao AI lại đưa ra quyết định này mà không phải quyết định khác, khi nào thì mô hình dự đoán thành công, khi nào thất bại, khi nào có thể tin vào dự đoán của AI? Những câu hỏi trên cho ta thấy nhu cầu hiểu được lí do đằng sau một dự đoán từ mô hình AI ngày càng cần thiết. Song song với đó là việc giữ cân bằng giữa độ chính xác và khả năng giải thích của 1 mô hình AI - điều mà rất nhiều mô hình ML và DL chưa giải quyết được. Do đó, XAI [3] ra đời để giúp các thuật toán trở nên minh bạch hơn và thay đổi các mô hình AI theo hướng lấy người dùng làm trung tâm (user-centricity). Điều này giúp mô hình AI dễ ứng dụng trong nhiều lĩnh vực khác nhau. Bên cạnh đó, XAI không chỉ giúp việc đánh giá độ tin cậy của 1 mô hình AI trở nên dễ dàng hơn mà còn giúp người dùng có thể tương tác và học hỏi qua lại với AI..

Nhận thấy việc nghiên cứu và giải thích được các mô hình AI vô cùng cấp bách và có ý nghĩa thực tiễn. Nhóm chúng tôi lựa chọn nghiên cứu và khai thác đề tài **"Tìm hiểu về Explainable AI và ứng dụng".**

1.2 MỤC TIÊU VÀ NHIỆM VỤ NGHIÊN CỨU

Mục tiêu của đề tài là tập trung nghiên cứu cơ sở lý thuyết của XAI, các cấp độ của bài toán, khai thác chiều sâu bài toán, các cách giải quyết bài toán phổ biến hiện nay. Trong đề tài này, chúng tôi muốn xây dựng một mô hình nhận diện hình ảnh và sử dụng phương

pháp LIME để giải thích cho kết quả nhận diện được. Để đạt được điều đó chúng tôi tập trung vào tìm hiểu một số vấn đề sau:

- Tìm hiểu cơ sở lý thuyết của XAI
- Tìm hiểu phương pháp sử dụng để diễn giải các mô hình học máy.
- Ứng dụng bài toán vào tập dữ liệu cụ thể để trực quan hóa bài toán.
- Đánh giá và giải thích kết quả

1.3 CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

XAI thường sử dụng một trong hai cách tiếp cận: phân tích hộp đen hoặc các mô hình có thể diễn giải. Phân tích hộp đen là phương pháp truyền thống vì nó chỉ đơn giản là mở hộp có sẵn của thuật toán và kiểm tra dữ liệu bên trong. Các mô hình có thể diễn giải giống như XAI có thể giải mã được bằng thiết kế.Những mô hình này nhằm mục đích để phân tích, giống như một chiếc máy tính trong tủ kính. Có thể rất phức tạp để tạo ra chúng, nhưng nhu cầu về XAI thân thiện với người dùng đang khuyến khích các nhà phát triển tiếp tục nghiên cứu và đổi mới công nghệ có thể diễn giải được.

1.4 KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC

Nhóm chúng tôi mong muốn sau khi thực hiện quá trình nghiên cứu nhiều công trình cũng như các ứng dụng từ các tác giả đi trước, nhóm có thể học hỏi và đúc kết thành một bài báo cáo khai thác sâu về nội dung lý thuyết của Explainable AI.

Về phần ứng dụng, để trực quan hóa bài toán nhóm sẽ xây dựng một mô hình đơn giản để trực quan hóa kết quả sau khi phân tích từ tập dữ liệu nhằm có cái nhìn cụ thể hơn cũng như thấy được sự hữu ích khi áp dụng vào thực tế.

1.5 BỐ CỤC LUẬN VĂN

Bố cục của tiểu luận chuyên ngành được tổ chưc như sau:

1. Mở đầu

1.1 Tính cấp thiết của đề tài

1.2 Mục tiêu và nhiệm vụ nghiên cứu

1.3 Cách tiếp cận và phương pháp nghiên cứu

1.4 Kết quả dự đoán được

1.5 Bố cục luận văn

2. Nội dung

2.1 Tổng quan về XAI

2.2 Tổng quan về các phương pháp trong XAI

2.2.1 Local Explanations

2.2.1.1 LIME

2.2.1. 2 SHAP

2.2.1.3 Anchor

2.2.1.4 ICE

2.2.2 Global Explanations

2.2.2.1 Partial Dependence Plot (PDP)

2.2.2.2 Feature Inter-action

3. Thực nghiệm

3.1 Bai toán

3.2 Dữ liệu

3.3 Phương pháp và kết quả

4. Kết luận

4.1 Kết quả đạt được

4.2 Hạn chế

4.3 Hướng phát triển

CHƯƠNG 2: NỘI DUNG

2.1 TỔNG QUAN VỀ EXPLAINABLE AI

2.1.1 Định nghĩa về Explainable AI

Trước hết, để hiểu Explainable AI (XAI) [4] là gì ta cần hiểu AI là như thế nào? Trí tuệ nhân tạo (AI) là một thuật ngữ rộng. Nó mô tả một loạt các công cụ và phương pháp cho phép các hệ thống máy tính thực hiện các nhiệm vụ phức tạp hoặc hành động trong môi trường đầy thách thức. Những năm gần đây đã chứng kiến những tiến bộ đáng kể trong công nghệ AI và nhiều người hiện đang tương tác với các hệ thống được hỗ trợ bởi AI hàng ngày. Những công cụ AI này có thể tạo ra kết quả chính xác cao, nhưng một số cũng rất phức tạp. Sự phức tạp này đã khiến các nhà nghiên cứu và các nhà hoạch định chính sách đặt câu hỏi - liệu có thể hiểu được cách thức hoạt động của AI hay AI là một back box (hộp đen) [5]

XAI là một tập hợp các quy trình và phương pháp cho phép người dùng hiểu và tin tưởng vào kết quả đầu ra được tạo ra bởi các thuật toán học máy. XAI được sử dụng để mô tả một mô hình AI, tác động dự kiến và những thành phần ảnh hưởng đến tính khách quan của bài toán. Nó giúp mô tả tính chính xác, công bằng, minh bạch và kết quả của mô hình trong việc ra quyết định do AI hỗ trợ. AI có thể giải thích được rất quan trọng đối với một tổ chức trong việc xây dựng niềm tin và sự tự tin khi đưa các mô hình AI vào sản xuất. Khả năng giải thích của AI cũng giúp một tổ chức áp dụng cách tiếp cận có trách nhiệm để phát triển AI

2.1.2 Lợi ích của Explainable AI

- Giảm chi phí cho những sai lầm

  Các lĩnh vực nhạy cảm với quyết định như Y học, Tài chính, Pháp lý, v.v., bị ảnh hưởng nhiều trong trường hợp dự đoán sai. Giám sát kết quả làm giảm tác động của kết quả sai sót và xác định nguyên nhân gốc rễ để cải thiện mô hình cơ bản. Kết quả là những dự đoán của AI viết ra trở nên thực tế hơn để sử dụng và tin tưởng theo thời gian.

- Giảm tác động của việc sai lệch mô hình

  Các mô hình AI đã cho thấy bằng chứng đáng kể về sự thiên vị. Các ví dụ bao gồm thiên vị giới tính đối với Apple Cards, thiên vị chủng tộc bằng Autonomous Vehicles

và thành kiến chủng tộc của Amazon Rekognition. Một hệ thống có thể giải thích có thể làm giảm tác động của những dự đoán thiên vị như vậy gây ra bằng cách giải thích các tiêu chí ra quyết định.

- Lỗi có thể được giảm thiểu

  Các mô hình AI luôn có một số lỗi với dự đoán của chúng và việc cho phép một người có thể chịu trách nhiệm về những lỗi đó có thể làm cho hệ thống tổng thể hiệu quả hơn

- Độ tin cậy và tuân thủ quy tắc

  Mọi suy luận, cùng với lời giải thích của nó, có xu hướng làm tăng sự tự tin của hệ thống. Một số hệ thống quan trọng của người dùng, chẳng hạn như xe tự hành, chẩn đoán y tế, lĩnh vực tài chính, v.v., đòi hỏi sự tự tin cao từ người dùng để sử dụng tối ưu hơn.

  Với áp lực ngày càng tăng từ các cơ quan quản lý, các công ty phải thích ứng và triển khai XAI để nhanh chóng tuân thủ quy tắc của các cơ quan chức năng

- Hiệu suất mô hình

  Một trong những chìa khóa để tối đa hóa hiệu suất là hiểu được những điểm yếu tiềm ẩn. Sự hiểu biết tốt hơn về những gì các mô hình đang làm và tại sao chúng đôi khi thất bại, thì càng dễ dàng cải thiện chúng. Khả năng giải thích là một công cụ mạnh mẽ để phát hiện các sai sót trong mô hình và sự thiên vị trong dữ liệu giúp xây dựng lòng tin cho tất cả người dùng. Nó có thể giúp xác minh các dự đoán, để cải thiện các mô hình và để có được những hiểu biết mới về vấn đề hiện tại. Phát hiện các sai lệch trong mô hình hoặc tập dữ liệu sẽ dễ dàng hơn khi chúng ta hiểu mô hình đang làm gì và tại sao nó lại đi đến dự đoán của nó

- Ra quyết định sáng suốt

  Việc sử dụng chính của các ứng dụng học máy trong kinh doanh là ra quyết định tự động. Tuy nhiên, thường thì chúng ta muốn sử dụng các mô hình chủ yếu cho thông tin chi tiết phân tích. Ví dụ: chúng ta có thể đào tạo một mô hình để dự đoán doanh số bán hàng tại cửa hàng trên một chuỗi bán lẻ lớn bằng cách sử dụng dữ liệu về vị trí, giờ mở cửa, thời tiết, thời gian trong năm, sản phẩm được vận chuyển, quy mô cửa hàng, v.v. Mô hình này sẽ cho phép chúng ta dự đoán doanh số bán hàng trên các cửa hàng của chúng ta vào bất kỳ ngày nào trong năm trong nhiều điều kiện thời tiết

khác nhau. Tuy nhiên, bằng cách xây dựng một mô hình có thể giải thích, chúng ta có thể thấy động lực chính của doanh số bán hàng là gì và sử dụng thông tin này để tăng doanh thu.

2.1.3 Explainable AI được sử dụng ở đâu?

- Y tế

  Máy học và công nghệ AI đã được sử dụng và triển khai trong môi trường chăm sóc sức khỏe. Tuy nhiên, các bác sĩ không thể giải thích lý do tại sao một số quyết định hoặc dự đoán nhất định được đưa ra. Điều này đặt ra những hạn chế về cách thức và vị trí có thể áp dụng công nghệ AI.

  Với XAI, các bác sĩ có thể biết tại sao một bệnh nhân nào đó có nguy cơ nhập viện cao và phương pháp điều trị nào sẽ phù hợp nhất. Điều này cho phép các bác sĩ hành động dựa trên thông tin tốt hơn.

- Marketing:

  AI và học máy tiếp tục là một phần quan trọng trong nỗ lực tiếp thị của các công ty bao gồm các cơ hội ấn tượng để tối đa hóa lợi nhuận đầu tư tiếp thị thông qua thông tin chi tiết về kinh doanh do họ cung cấp.

  Các thông tin mạnh mẽ như vậy giúp định hướng các chiến lược tiếp thị, các nhà tiếp thị phải tự hỏi mình "Làm thế nào tôi có thể tin tưởng vào lý do đằng sau các đề xuất của AI cho các hành động tiếp thị của mình?"

  Với XAI, các nhà tiếp thị có thể phát hiện bất kỳ điểm yếu nào trong các mô hình AI của họ và giảm thiểu chúng, do đó nhận được kết quả và thông tin chi tiết chính xác hơn mà họ có thể tin tưởng. Điều này có thể thực hiện được vì XAI cung cấp cho họ sự hiểu biết tốt hơn về kết quả tiếp thị dự kiến, lý do đằng sau các hành động tiếp thị được đề xuất và chìa khóa để cải thiện hiệu quả với các quyết định tiếp thị nhanh hơn và chính xác hơn và tăng ROI tiếp thị của họ trong khi giảm chi phí tiềm năng

- Bảo hiểm:

  Với ngành bảo hiểm, XAI có tác động đáng kể. Các công ty bảo hiểm phải tin tưởng, hiểu và kiểm tra hệ thống AI của họ để tối ưu hóa toàn bộ tiềm năng của chúng.

  XAI đã chứng tỏ là một công cụ thay đổi cuộc chơi đối với nhiều công ty bảo hiểm. Với nó, các công ty bảo hiểm đang thấy việc thu hút khách hàng và chuyển đổi báo giá được cải thiện, tăng năng suất và hiệu quả, đồng thời giảm tỷ lệ yêu cầu bồi

thường và yêu cầu gian lận.

- Dịch vụ tài chính

  Các tổ chức tài chính đang tích cực tận dụng công nghệ AI. Họ tìm cách cung cấp cho khách hàng của mình sự ổn định tài chính, nhận thức về tài chính và quản lý tài chính.

  Với XAI, các dịch vụ tài chính cung cấp kết quả công bằng, không thiên vị và có thể giải thích được cho khách hàng và nhà cung cấp dịch vụ của họ. Nó cho phép các tổ chức tài chính đảm bảo tuân thủ các yêu cầu quy định khác nhau trong khi tuân theo các tiêu chuẩn đạo đức và công bằng.

  Một vài cách để XAI có thể mang lại lợi ích cho ngành tài chính gồm cải thiện dự báo thị trường, đảm bảo tính công bằng trong việc chấm điểm tín dụng, khám phá các yếu tố liên quan đến trộm cắp để giảm dương tính giả và giảm chi phí tiềm ẩn do thiên vị hoặc sai sót của AI.

2.2 TỔNG QUAN VỀ CÁC PHƯƠNG PHÁP TRONG EXPLAINABLE AI

2.2.1 Local Explanations

Local Explanations [6] giúp hiểu hành vi của mô hình trong vùng lân cận cục bộ, tức là nó đưa ra lời giải thích về từng đặc trưng trong dữ liệu và cách mỗi đặc trưng đóng góp riêng lẻ vào dự đoán của mô hình. Khả năng giải thích cục bộ giúp tìm ra nguyên nhân gốc rễ của một vấn đề cụ thể trong quá trình dự đoán. Nó cũng có thể được sử dụng để giúp chúng ta khám phá những đặc trưng nào có ảnh hưởng nhất trong việc đưa ra các quyết định về mô hình. Điều này rất quan trọng, đặc biệt là trong các ngành như tài chính và y tế, nơi các đặc trưng riêng lẻ cũng quan trọng như tất cả các đặc trưng kết hợp lại. Ví dụ, hãy tưởng tượng mô hình rủi ro tín dụng của chúng ta đã từ chối một người xin vay. Với khả năng giải thích tại cục bộ, chúng ta có thể biết lý do tại sao quyết định này được đưa ra và cách tư vấn tốt hơn cho người nộp đơn. Nó cũng giúp hiểu được sự phù hợp của mô hình để triển khai

Đối với Local Explanations ta có các phương pháp tiếp cận điển hình sau

- Feature Importances
- Rule Based
- Saliency Maps

- Prototype-Based Explanations
- Counterfactual Explanations

2.2.1.1 LIME

2.2.1.2.1 Giới thiệu

Giải thích cục bộ kiểu mẫu (Local interpretable model-agnostic explanations (LIME)) [7] là một bài báo trong đó tác giả đề xuất một cách triển khai các mô hình đại diện cục bộ. Các mô hình đại diện (surrogate models) được huấn luyện để xấp xỉ dự đoán của một mô hình hộp đen. Thay vì huấn luyện một mô hình đại diện

toàn cục (global surrogate model), LIME tập trung vào việc huấn luyện các

mô hình đại diện cục bộ để giải thích các dự đoán đơn lẻ

2.2.1.2.2 Ý tưởng

Ý tưởng của phương pháp rất rõ ràng và dễ hiểu. Đầu tiên, hãy quên dữ liệu đã đào tạo và tưởng tượng chúng ta chỉ có một mô hình hộp đen, nơi mà chúng ta cung cấp đầu vào và nó trả về dự đoán. Chúng ta có thể kiểm chứng mô hình ở bất cứ nơi nào và tại thời điểm nào ta muốn. Mục tiêu của chúng ta là hiểu tại sao một mô hình học máy sinh ra một dự đoán như vậy. LIME kiểm tra liệu dự đoán có thay đổi nếu ta biến đổi dữ liệu đầu vào. LIME sinh ra một tập dữ liệu mới bao gồm các mẫu đã được biến đổi và các dự đoán tương ứng của một mô hình hộp đen. Trên tập dữ liệu mới này, LIME sau đó huấn luyện một mô hình khả diễn giải bằng cách tính toán sai khác giữa các điểm dữ liệu được lấy mẫu với điểm dữ liệu được quan tâm. Mô hình khả diễn giải này có thể là bất cứ loại mô hình nào trong chương các mô hình khả diễn giải, ví dụ như Lasso hoặc một cây quyết định. Mô hình sau khi học sẽ là một phép xấp xỉ của các dự đoán từ mô hình học máy xét trên phương diện cục bộ, nhưng không nhất thiết đúng trên phương diện toàn cục. Khi này độ chính xác còn được gọi là sự nhất quán cục bộ (local fidelity)

- Công thức tính toán

$$\text{explanation}(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Hình 1 : Công thức tính LIME

Mô hình giải thích cho mẫu dữ liệu x là mô hình g (ví dụ mô hình hồi quy tuyến tính), và g sẽ tối thiểu hóa mất mát L (ví dụ sai số bình phương tối thiểu MSE), được tính bởi khoảng cách giữa giải thích tới dự đoán của mô hình gốc (ví dụ một mô hình tăng tốc gradient), với ràng buộc là độ phức tạp của mô hình Ω(g) được giữ ở một giá trị nhỏ (để ta có ít đặc trưng). G là một họ các giải thích khả dĩ, ví dụ tất cả các mô hình hồi quy tuyến tính khả dĩ. Giá trị π_x định nghĩa độ lớn của các hàng xóm lân cận một điểm dữ liệu x ta đang xem xét cho việc giải thích. Trong thực tế, LIME chỉ tối ưu phần mất mát. Người dùng phải tự định nghĩa độ phức tạp (ví dụ chọn số lượng đặc trưng lớn nhất có thể cho mô hình hồi quy tuyến tính)

- Các bước huấn luyện các mô hình đại diện cục bộ

  - Lựa chọn mẫu mà ta quan tâm.

  - Biến đổi tập dữ liệu và lưu lại các dự đoán của các mô hình hộp đen

  trên tập dữ liệu đã biến đổi này.

  - Đánh trọng số các mẫu dữ liệu mới tương ứng với khoảng cách tới điểm

  dữ liệu ta đang quan tâm ở bước đầu.

  - Huấn luyện một mô hình mang trọng số và khả diễn giải trên tập dữ

  liệu đã được biến đổi.

  - Giải thích dự đoán bằng cách diễn giải mô hình cục bộ

2.2.1.2 SHAP

SHAP (Shapley Additive Explanations) [8] [9] một phương pháp để giải thích các dự đoán riêng lẻ, dựa trên lý thuyết trò chơi để tính các giá trị Shapley tối ưu. Giá trị Shapley là một cách tiếp cận được sử dụng rộng rãi từ lý thuyết trò chơi hợp tác đi kèm với các đặc tính mong muốn. Các giá trị đặc trưng của một phiên bản dữ liệu đóng vai trò là người chơi

trong một liên minh. Giá trị Shapley là đóng góp cận biên trung bình của một giá trị đặc trưng trên tất cả các liên minh có thể có

2.2.1.2.1 Giới thiệu

Mục tiêu của SHAP là giải thích dự đoán của một mẫu dữ liệu x thông qua tính mức đóng góp của mỗi đặc trưng cho phép dự đoán. Phương pháp giải thích SHAP là tính các giá trị Shapley dựa trên lý thuyết trò chơi liên minh (coalition). Các giá trị đặc trưng của một mẫu dữ liệu làm các người chơi trong một liên minh. Giá trị Shapley cho biết các để phân phối công bằng các "payout" (= phép dự đoán) giữa các đặc trưng. Một người chơi có thể là một giá trị đặc trưng riêng hoặc có thể là một nhóm các giá trị đặc trưng. Ví dụ, để giải thích một hình ảnh, pixel có thể được nhóm thành superpixels và phép dự đoán được phân phối giữa chúng. Một cải tiến mà SHAP mang đến là giải thích giá trị Shapley được biểu diễn dưới dạng phương pháp cộng tính thuộc tính đặc trưng (additive feature attribution method), một mô hình tuyến tính. Quan điểm này kết nối các giá trị LIME và Shapley. SHAP chỉ định giải thích là:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

Hình 2: Công thức tính SHAP

Với g là mô hình giải thích, z' $\in \{0,1\}^M$ là vector liên minh (coalition vector), M là kích cỡ liên minh tối đa và $\phi_j \in R$ là phân bổ đặc trưng cho đặc trưng j, các giá trị Shapley. Cái mà ta gọi là "vector liên minh" được gọi là "các đặc trưng đơn giản hóa" (simplified feature) trong bài báo SHAP. Trong vector liên minh, đầu vào 1 có nghĩa là giá trị đặc trưng tương ứng là "có mặt" và 0 là "vắng mặt". Để tính toán các giá trị Shapley, chúng ta mô phỏng rằng chỉ một số giá trị đặc trưng đang tham gia ("có mặt") và một số thì không ("vắng mặt"). Việc biểu diễn dưới dạng mô hình tuyến tính của các liên minh là mẹo để tính toán các ϕ. Đối với x, mẫu dữ liệu đang quan tâm, vector liên minh x' là vector tất cả các giá trị 1, tức là tất cả các giá trị đặc trưng đều "có mặt". Từ đó, ta có được công thức đơn giản thành:

$$g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j$$

Hình 3: Công thức đơn giản của SHAP

2.2.1.2.2 KernelSHAP

KernelSHAP [10] một phương pháp sử dụng khung LIME để tính toán giá trị Shapley của mỗi đặc trưng lên dự đoán. KernelSHAP bao gồm 5 bước:

(a) Mẫu liên minh $z'_k \in \{0,1\}^M$, $k \in \{1, \cdots K\}$(Giá trị 1 tương ứng với đặc trưng có mặt trong liên minh, giá trị 0 tương ứng với đặc trưng vắng mặt)

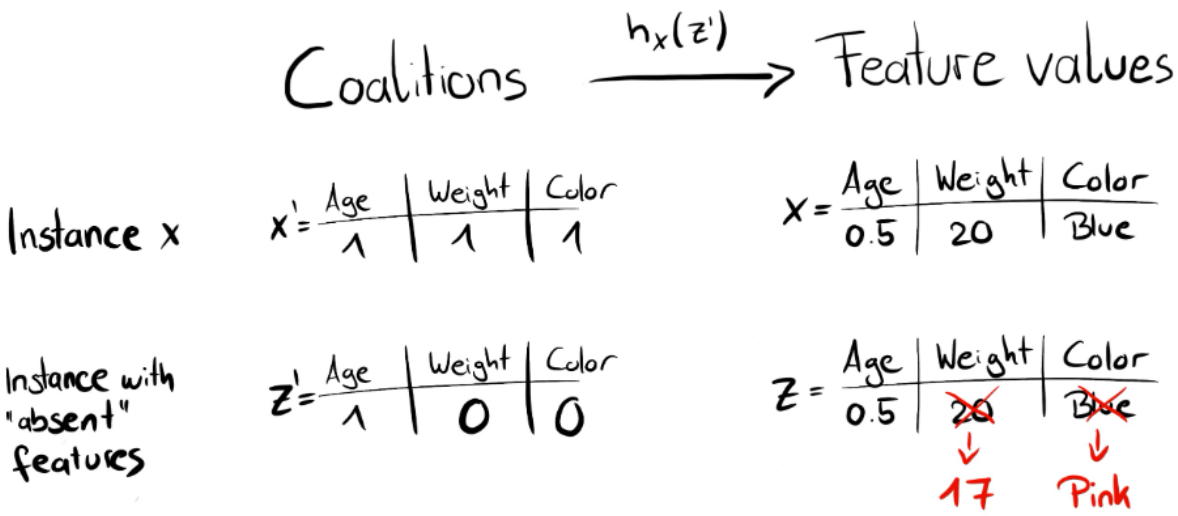
(b) Cho dự đoán với mỗi z'_k bằng mô hình phức tạp trước khi dự đoán z'_k bằng mô hình tuyến tính, ở đây chúng ta áp dụng mô hình f: $f(h_x(z'_k)$ với $h_x: \{0,1\} \overset{M}{\rightarrow} \mathbb{R}^p$

(c) Tính trọng số (weight) cho mỗi điểm dữ liệu z'_k tới đặc trưng đang được xét trong SHAP.

(d) Khớp mô hình tuyến tính đã tính trọng số.

(e) Trả lại các giá trị Shapley ϕ_k, các hệ số từ mô hình tuyến tính

Ta có thể tạo ra một liên minh ngẫu nhiên qua các phép lật đồng xu liên tục cho tới khi thu được các chuỗi 0 và 1. Ví dụ: vector (0, 1, 0, 1) mô tả rằng ta có một liên minh của đặc trưng thứ nhất và thứ ba. Số lượng K liên minh đã lấy mẫu (sampling) làm thành tập dữ liệu cho mô hình hồi quy. Đối tượng cho mô hình hồi quy là phép dự đoán cho một liên minh Để đi từ các liên minh của các giá trị đặc trưng tới các mẫu dữ liệu hợp lệ, ta cần hàm $h_x(z') = z$ với $h_x: \{0,1\} \overset{M}{\rightarrow} \mathbb{R}^p$. Hàm h_x ánh xạ từ 1 tới giá trị tương ứng từ mẫu dữ liệu x ta đang muốn giải thích. Với dữ liệu dạng bảng, nó ánh xạ các số 0 tới các giá trị của một mẫu dữ liệu khác ta lấy mẫu từ dữ liệu. Điều này nghĩa là ta đánh đồng "giá trị đặc trưng là vắng mặt" với "giá trị đặc trưng bị thay đổi bởi giá trị đặc trưng ngẫu nhiên từ dữ liệu." Từ dữ liệu dạng bảng, hình sau mô phỏng ánh xã từ các liên minh tới các giá trị đặc trưng:



Hình 4: Ví dụ minh hoạ về Kernel SHAP

Từ hình 4 ta thấy hàm h_x ánh xạ một liên minh tới một mẫu dữ liệu hợp lệ. Cho đặc trưng có mặt (1), h_x ánh xạ từ các giá trị đặc trưng cũa x. Cho các đặc trưng vắng mặt (0), ánh xạ từ các giá trị của một mẫu dữ liệu lấy mẫu ngẫu nhiên

2.2.1.2.3 TreeSHAP

Lundberg et. al (2018) [11] đề xuất TreeSHAP, một biến thể của SHAP cho các mô hình học máy dựa trên cây như cây quyết định, rừng ngẫu nhiên và cây tăng cường gradient (gradient boosted trees). TreeSHAP được giới thiệu là một giải pháp thay thế nhanh, theo mô hình cụ thể cho KernelSHAP, nhưng hóa ra nó có thể tạo ra các thuộc tích (attributions) về đặc trưng không trực quan. TreeSHAP xác định hàm giá trị bằng cách sử dụng kỳ vọng có điều kiện (conditional expectation) $E_{x_S|x_c}(f(x)|x_S)$ thay vì kỳ vọng cận biên (marginal expectation). Vấn đề với kỳ vọng có điều kiện là các đặc trưng không có ảnh hưởng lên hàm dự đoán f có thể nhận ước tính TreeSHAP khác 0. Ước tính khác 0 xảy ra khi đặc trưng tương quan với một đặc trưng khác mà gây ảnh hưởng thực sự lên dự đoán. TreeSHAP nhanh hơn bao nhiêu? So với KernelSHAP, nó làm giảm độ phức tạp tính toán từ $O(TL2^M)$ thành $O(TLD^2)$, với T là số lượng cây, L là số lượng lớn nhất các lá trong một cây và D là độ sâu tối đa của bất kì cây nào. TreeSHAP dùng kỳ vọng có điều kiện $E_{x_S|x_c}(f(x)|x_S)$ để ước lượng các ảnh hưởng. Ta sẽ cho chúng ta một số trực giác về các chúng ta tính kì vọng dự đoán (prediction expectation) cho một cây, một mẫu dữ liệu x và một tập con đặc trưng S. Nếu ta điều kiện hoá tất cả các đặc trưng – nếu S là tập tất cả các đặc trưng – phép đặc

trưng từ một nút (node) trong đó mẫu dữ liệu x nằm trong là kỳ vọng dự đoán. Nếu chúng ta không điều kiện hoá lên bất cứ đặc trưng nào – tập S rỗng – ta sẽ dùng trung bình trọng số hoá (weighted average) của các dự đoán trong tất cả các nút đầu cuối (terminal nodes). Nếu S chứa một số, nhưng không phải tất cả các đặc trưng, ta bỏ qua các dự đoán của các nút không truy cập được (unreachable nodes). Không thể truy cập có nghĩa là đường quyết định (decision path) mà dẫn đến nút này mẫu thuẫn với các giá trị xS. Từ các nút đầu cuối còn lại, ta trung bình các dự đoán trọng số hoá bằng kích thước các nút (tức số lượng mẫu huấn luyện trong nút đó). Trung bình của các nút đầu cuối còn lại, trọng số hoá bằng số lượng mẫu dữ liệu trên một nút, là kỳ vọng dự đoán cho x với S cho trước. Vấn đề ở đây là chúng ta phải áp dụng quy trình này cho mỗi tập con các gái trị đặc trưng S. TreeSHAP tính toán trong thời gian đa thức thay vì theo cấp số nhân (exponential). Ý tưởng cơ bản là đẩy tất cả các tập con có thể S xuống cây cùng một lúc. Cho mỗi nút quyết định (decision node) ta phải theo dỏi chung số lượng các tập con. Điều này phụ thuộc vào các tập con trong nút cha-mẹ (parent node). Ví dụ khi phần tách (split) đầu tiên trong một cây là đặc trưng x3, mọi tập con chứa đặc trưng x3 đều về một nút (nút mà x về). Các tập con không chứa x3 sẽ chuyển đến cả hai nút với trọng số giảm. Không may các tập con với kích thước khác nhau có trọng số khác nhau. Thuật toán phải theo dõi trọng số tổng thể (overall) của các tập con trong một nút. Điều này tăng phức tạp cho thuật toán. Ta để cho bài báo gốc cho chi tiết về TreeSHAP. Phép tính toán có thể mở rộng cho nhiều cây hơn: Nhờ vào Cộng tính của các giá trị Shapley, các giá trị Shapley của một nhóm cây (tree ensemble) là (trọng số) trung bình của các giá trị Shapley của các cây riêng lẽ

2.2.1.2.4 Mức quan trọng của đặc trưng SHAP (SHAP feature importance)

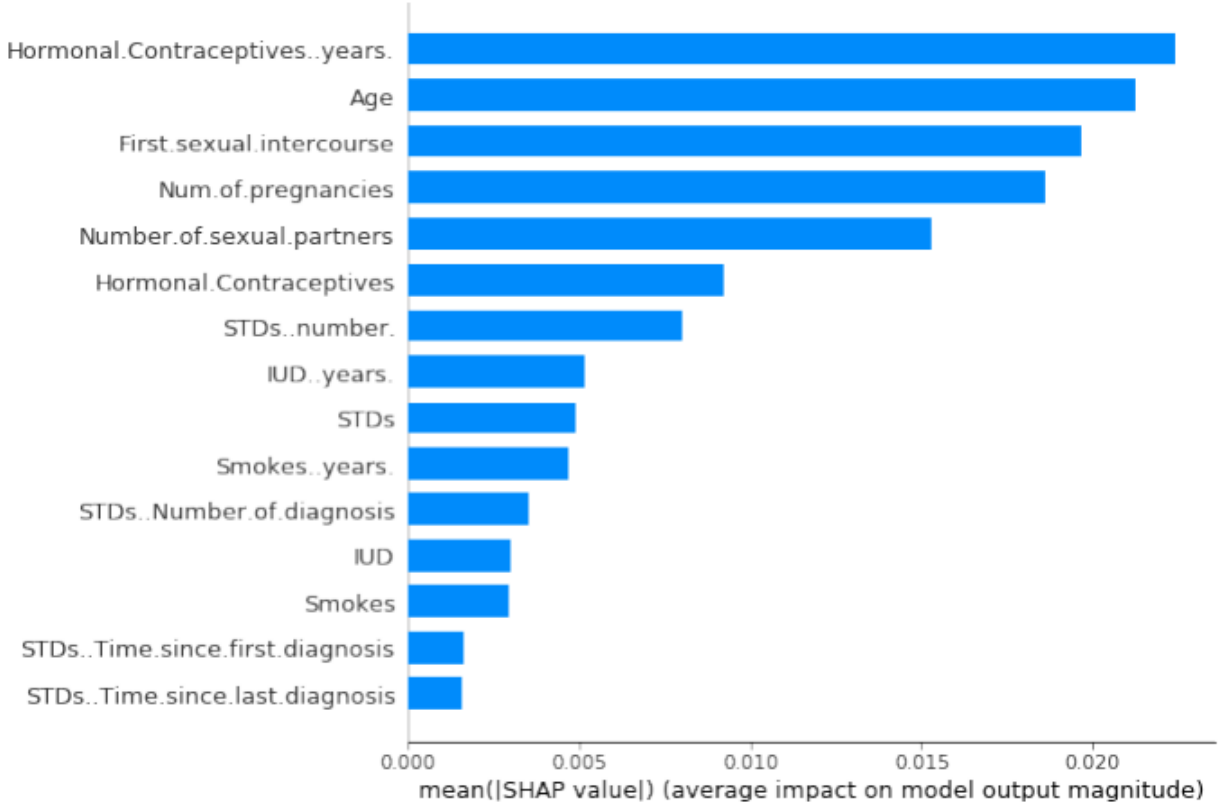
Ý tưởng đằng sau SHAP feature importance khá đơn giản: Các đặc trưng với giá trị tuyệt đối Shapley cao rất quan trọng. Vì chúng ta muốn mức quan trọng toàn cục (global importance), ta trung bình các giá trị tuyệt đối Shapley với mỗi đặc trưng xuyên suốt các dữ liệu:

$$I_j = \sum_{i=1}^{n} |\phi_j^{(i)}|$$

Hình 5: Công thức tính giá trị tuyệt đối Shaley

Tiếp theo, ta sắp xếp các đặc trưng giảm dần theo độ quan trọng và phác hoạ chúng. Hình

dưới đây cho thấy mức quan trọng giá trị SHAP cho huấn luyện rừng ngẫu nhiên trước đó để dự đoán ung thư cổ tử cung



Hình 6: Biểu đồ cho thấy mức độ đặc trưng SHAP của người ung thư cổ tử cung

Hình 6 cho thấy mức quan trọng của đặc trưng SHAP được đo bằng trung bình giá trị tuyệt đối Shapley. Số năm sử dụng các biện pháp tránh thai nội tiết tố là đặc trưng quan trọng nhất, thay đổi xác suất ung thư tuyệt đối dự đoán trên trung bình 2.4 điểm phần trăm (0.024 trên trục x)

Mức quan trọng của đặc trưng SHAP là một thay thế cho tầm quan trọng của đặc trưng hoán vị. Có một sự khác biệt lớn giữa cả hai mức đo độ quan trọng (important measures): Mức quan trọng của đặc trưng hoán vị dựa trên sự giảm hiệu suất của mô hình. SHAP dựa trên mức của các thuộc tính đặc trưng (feature attributions). Đồ thị mức quan trọng đặc trưng hữu ích, nhưng không chứa thông tin khác ngoài các mức quan trọng

2.2.1.3 Anchor

2.2.1.3.1 Giới thiệu

Anchors [12] giải thích các dự đoán riêng lẻ (individual predictions) của một mô hình phân loại hộp đen tuỳ ý bằng cách tìm ra quy tắc quyết định (decision rule) mà

"anchors" dự đoán một cách đầy đủ. Một quy tắc "anchors" dự đoán nếu các thay đổi trong các giá trị đặc trưng khác không ảnh hưởng đến dự đoán. Các anchors sử dụng kỹ thuật học tăng cường (reinforcement learning) kết hợp với thuật toán tìm kiếm trên đồ thị để giảm số lượng gọi lên mô hình (và từ đó thời gian chạy cần thiết) xuống mức tối thiểu trong khi vẫn có thể phục hồi từ tối ưu cục bộ (local optima). Ribeiro, Singh và Guestrin đã đề xuất thuật toán vào năm 2018 – đồng các nhà nghiên cứu đã giới thiệu thuật toán LIME.[4] Cách tiếp cận anchors (anchors approach) triển khai chiến lược dựa trên xáo trộn (perturbation-based strategy) để tạo ra các giải thích cục bộ cho các dự đoán về các mô hình học máy hộp đen. Tuy nhiên, thay vì các mô hình đại diện (surrogate models) được sử dụng bởi LIME, kết quả giải thích được thể hiện dưới dạng các quy tắc IF-THEN dễ hiểu, được gọi là neo. Các quy tắc này có thể được dùng lại vì chúng được giới hạn (scoped): các anchors bao gồm khái niệm về độ phủ (coverage), nêu chính xác các mẫu dữ liệu, có thể không nhìn thấy, mà chúng áp dụng vào. Tìm kiếm anchors liên quan đến vấn đề khám phá (exploration) hoặc máy đánh bạc đa cần (multi-armed bandit), bắt nguồn từ ngành học tăng cường. Cuối cùng, các điểm lân cận, hoặc là các điểm xáo trộn, được tạo và đánh giá cho mọi mẫu dữ liệu đang được giải thích. Làm như thế cho phép cách tiếp cận của ta bỏ qua cấu trúc hộp đen và các tham số bên trong để chúng có thể vẫn không quan sát và không thay đổi. Do đó, thuật toán là kiểu mẫu (model-agnostic), nghĩa là nó có thể được áp dụng cho bất kỳ phân loại mô hình nào.

Anchors A được định nghĩa như sau:

$$\mathbb{E}_{\mathcal{D}_x(z|A)}[1_{f(x)=f(z)}] \geqslant \tau, A(x) = 1$$

Hình 7: Công thức tính anchors

Trong đó:

• x là mẫu dữ liệu đang giải thích (ví dụ, một hàng trong bộ dữ liệu bảng)

• A là tập các predicates, tức quy tắc kết quả hoặc anchors, sao cho A(x) = 1 khi các đặc trưng predicates định nghĩa bởi A tương ứng với những giá trị đặc trưng của x.

• f kí hiệu mô hình phân loại được giải thích (ví dụ, mô hình mạng nơ-ron nhân tạo). Nó có thể được sử dụng để dự đoán một dán nhãn cho x và các xáo trộn của nó.

• $D_x(.|A)$ thể hiện phân phối các điểm lân cận trong x phù hợp với A.

• $0 \leq \tau \leq 1$ chỉ định độ chính xác giới hạn. Chỉ có các quy tắc thu được tính nhất quán lân cận (local fidelity) ít nhất là τ được coi là kết quả đúng.

2.2.1.3.2 Tìm anchors

Mặc dù mô tả toán học của các "anchors" khá rõ và đơn giản, xây dựng các quy tắc cụ thể là không thể. Điều đó đòi hỏi đánh 1 $f(x) = f(z)$ cho mọi $z \in D(.|A)$và không khả thi trong những không gian đầu vào tiếp diễn (continuous) hoặc lớn. Do đó các tác giả đề xuất thêm một tham số $0 \leq \delta \leq 1$ để tạo ra định nghĩa theo xác suất. Với cách này, các mẫu (samples) được xuất ra cho tới khi có độ thống kê tin cậy (statistical confidence) về tính chính xác của chúng. Định nghĩa theo xác suất là như sau:

$$P(prec(A) \geqslant \tau) \geqslant 1 - \delta \quad \text{with} \quad prec(A) = \mathbb{E}_{\mathcal{D}_x(z|A)}[1_{f(x)=f(z)}]$$

Hình 8: Công thức tính độ thống kê tin cậy

Hai định nghĩa trước đó được kết hợp và mở rộng qua khái niệm độ phủ. Cơ sở lý luận bao gồm tìm các quy tắc áp dụng được cho một phần lớn của không gian đầu vào của mô hình. Độ phủ được định nghĩa là xác suất của các "anchor" áp dụng lên các vùng lân cận của điểm đang xét, tức là không gian xáo trộn của nó:

$$cov(A) = \mathbb{E}_{\mathcal{D}_{(z)}}[A(z)]$$

Hình 9: Công thức tính độ phủ

Yếu tố này dẫn đến định nghĩa của các "anchor", có tính đến phép tối đa độ phủ:

$$\max_{A \text{ s.t. } P(prec(A) \geqslant \tau) \geqslant 1-\delta} cov(A)$$

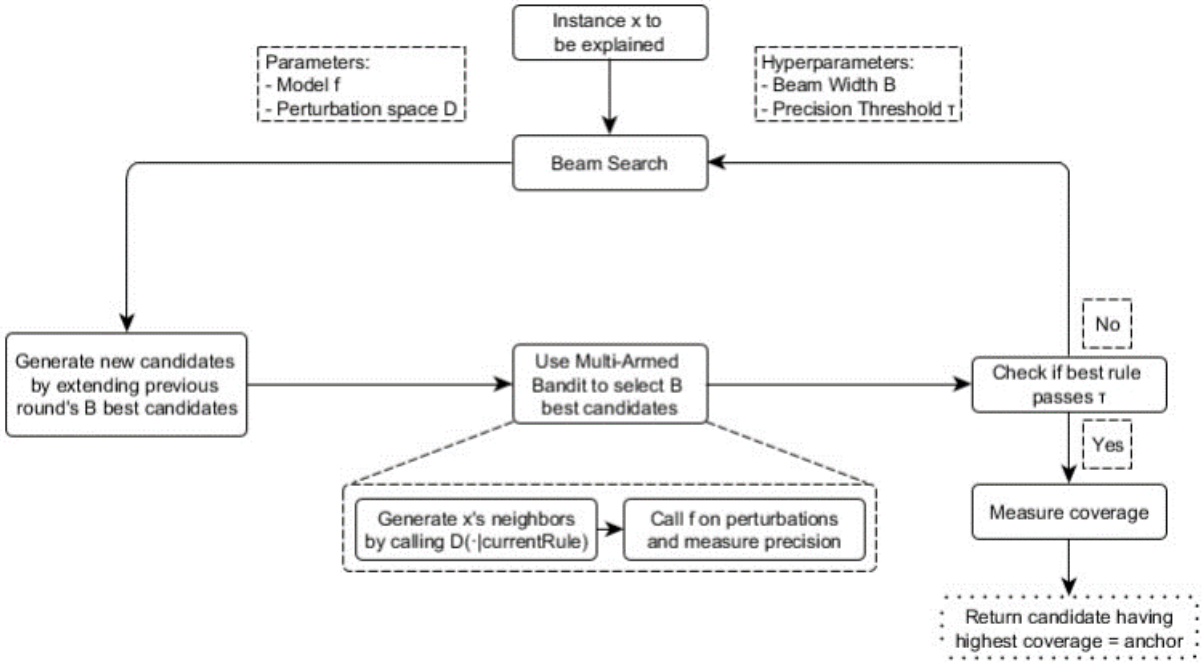
Hình 10: Công thức tính tối đa độ phủ

Do đó quá trình tiếp theo sẽ tìm quy tắc mà có độ phủ cao nhất trong các quy tắc thoả mãn (tất cả mà thoả mãn ngưỡng precision cho trước định nghĩa theo xác suất). Tất cả quy tắc này được cho là quan trọng hơn, do chúng diễn tả phần lớn hơn của mô hình. Chú ý các quy tắc với nhiều predicates có thiên hướng chính xác cao hơn các quy tắc ít predicates hơn.

Cụ thể, một quy tắc đã cố định mọi đặc trưng của x làm giảm số lượng đánh giá vùng lân cận tới các mẫu dữ liệu giống nhau. Do đó mô hình sẽ phân loại các vùng lân cận như nhau và độ chính xác của quy tắc là 1. Đồng thời một quy tắc đã cố định nhiều đặc trưng là cụ thể quá mức và chỉ áp dụng tới một số mẫu dữ liệu. Do đó có sự đánh đổi giữa precision và độ phủ. Cách tiếp cận mỏ neo dùng bốn thành phần chính để tìm những giải thích (explanations), được miêu tả trong hình sau:

- Thế hệ ứng viên : Tạo các ứng viên giải thích mới. Trong vòng đầu tiên, một ứng cử viên cho mỗi đặc trưng của x được tạo và sửa đổi giá trị tương ứng trong các xáo trộn. Ở các vòng khác, các ứng cử viên tốt nhất của vòng trước được mở rộng bởi một predicate đặc trưng chưa có trong đó.
- Xác định ứng viên tốt nhất : Các quy tắc từ ứng viên sẽ được so sánh liên quan đến quy tắc nào giải thích x tốt nhất. Về cuối, xáo trộn mà phù hợp với quy tắc hiện tạo ra và đang được quan sát sẽ được đánh giá bằng cách gọi mô hình. Tuy nhiên, những lần gọi này cần được giảm thiểu để hạn chế vượt giới hạn tính toán. Đây là lý do tại sao, tại cốt lõi thành phần này, có một máy đánh bạc đa cần (Multi-Armed-Bandit – MAB) thuần khám phá (pure-exploration). Các MAB được sử dụng để khám phá và khai thác hiệu quả các chiến lược khác nhau (được gọi là cần (arm) tương tự máy đánh bạc) bằng cách sử dụng lựa chọn tuần tự (sequential selection). Trong cài đặt đã cho, mỗi quy tắc từ ứng cử viên được xem là một cánh tay có thể được kéo. Mỗi lần kéo, những lân cận tương ứng được đánh giá và do đó chúng ta có được thêm thông tin về payoff của quy tắc từ ứng viên (độ chính xác trong trường hợp các "neo"). Do đó, độ chính xác cho biết quy tắc mô tả mẫu dữ liệu được giải thích tốt như thế nào.
- Kiểm định chính xác của ứng viên : Lấy thêm mẫu trong trường hợp không có độ tin cậy thống kê nào mặc dù thấy ứng viên vượt quá ngưỡng τ .
- Tìm kiếm chùm đã sửa đổi (Modified Beam Search) : Tất cả các thành phần trên được tập hợp trong một tìm kiếm chùm (beam search), là thuật toán tìm kiếm đồ thị và một biến thể của tìm kiếm theo chiều rộng (breadth-first algorithm). Nó mang B ứng cử viên tốt nhất của mỗi vòng qua vòng tiếp theo (trong đó B được gọi là Độ rộng chùm (Beam Width)). Số lượng B quy tắc tốt nhất này được sử dụng để tạo những quy tắc mới. Việc tìm kiếm chùm tia tiến hành qua nhiều nhất featureCount(x) vòng, vì mỗi đặc trưng chỉ được đưa vào một quy tắc nhiều nhất một lần. Do đó, ở

mỗi vòng i, nó tạo ra các ứng cử viên có chính xác i predicates và chọn số B tốt nhất. Do đó, bằng cách cho B lớn, thuật toán có nhiều khả năng tránh tối ưu cục bộ. Đổi lại, điều này đòi hỏi một lượng lớn các lần gọi mô hình và do đó làm tăng lượng tải tính toán.



Hình 11: Các thành phần của thuật toán anchors và mối quan hệ tương quan bên trong (đã được đơn giản hoá)

Cách tiếp cận này dường như là một công thức hoàn hảo để thu được thông tin hợp lý về thống kê (statistically sound information) về lý do tại sao bất kỳ hệ thống nào phân loại một mẫu dữ liệu theo cách nó đã làm. Nó thử nghiệm một cách có hệ thống với đầu vào mô hình và kết luận bằng cách quan sát các đầu ra tương ứng. Nó dựa vào các phương pháp học máy được thiết lập và nghiên cứu kỹ lưỡng (như MAB) để giảm số lượng cuộc gọi được thực hiện cho mô hình. Điều này làm giảm đáng kể thời gian chạy thuật toán.

2.2.1.3.3 Độ phức tạp và thời gian chạy (Runtime)

Biết hành vi thời gian chạy tiệm cận của các cách tiếp cận anchors giúp đánh giá mức độ hiệu quả của nó đối với các vấn đề cụ thể. Cho B là độ rộng chùm tia và p là số lượng tất cả các đặc trưng. Sau đó, thuật toán anchors phải tuân theo độ phức tạp:

$$\mathcal{O}(B \cdot p^2 + p^2 \cdot \mathcal{O}_{\mathrm{MAB}[B \cdot p, B]})$$

Hình 12: Công thức tính độ phức tạp của Anchors

Vùng biên cận này trừu tượng hóa từ các siêu tham số độc lập theo vấn đề (problem-independent hyperparameters), chẳng hạn như độ tin cậy thống kê δ. Bỏ qua các siêu tham số giúp giảm độ phức tạp của vùng biên cận (boundary). Vì MAB trích ra các ứng cử viên B tốt nhất trong số các ứng cử viên B.p mỗi vòng, hầu hết các MAB và thời gian chạy của chúng nhân hệ số p nhiều hơn bất kỳ tham số nào khác. Do đó, điều sau trở nên rõ ràng: hiệu quả của thuật toán giảm với những vấn đề có nhiều đặc trưng (feature abundant problems)

2.2.1.4 Individual Conditional Expectation (ICE)

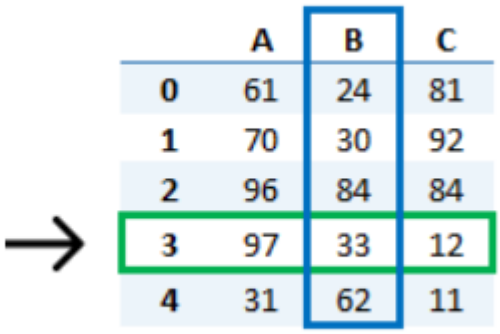
2.2.1.4.1 Giới thiệu

Phác hoạ ICE [13] gồm một đường cho mỗi điểm dữ liệu, thể hiện dự đoán cho điểm dữ liệu đó thay đổi thế nào khi một đặc trưng bị thay đổi. Phác hoạ phụ thuộc riêng(PDP) cho ảnh hưởng trung bình của một đặc trưng là một phương pháp toàn cục do không tập trung vào một điểm dữ liệu cụ thể mà tính trung bình trên tất cả. Phương pháp tương tự cho từng điểm dữ liệu riêng lẻ là ICE (Goldstein et al. 2017) [5] . Phác hoạ ICE minh họa sự phụ thuộc của dự đoán với một đặc trưng cho từng điểm dữ liệu riêng biệt, trả về một đường cho mỗi điểm dữ liệu, khác với PDP chỉ có một đường cho tất cả, tương đương với việc lấy trung bình tất cả các đường trong ICE. Mỗi đường này được tính bằng cách chỉ thay đổi giá trị của đặc trưng duy nhất đang xét bằng một giá trị khác, dự đoán điểm dữ liệu được thay đổi này bằng mô hình hộp đen. Kết quả nhận được là một tập các điểm cho các dự đoán khi thay đặc trưng bằng các giá trị cho phép

2.2.1.4.2 Tính toán các giá trị cho đường cong ICE

Các biểu đồ ICE theo truyền thống được sử dụng để hiểu các tương tác và sự khác biệt trong các tập hợp con dữ liệu như một phần của phân tích Phụ thuộc một phần (PD). Tuy nhiên, như đã đề cập trước đó, vì một ICE mô tả các quan sát riêng lẻ, nên có khả năng sử dụng nó để tập trung vào một trường hợp cụ thể mà chúng ta quan tâm

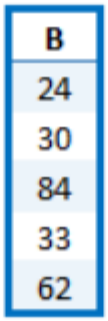
- Tìm trường hợp và đặc trưng chúng ta quan tâm



	A	B	C
0	61	24	81
1	70	30	92
2	96	84	84
3	97	33	12
4	31	62	11

Hình 13: Ví dụ tìm đặc trưng trong ICE

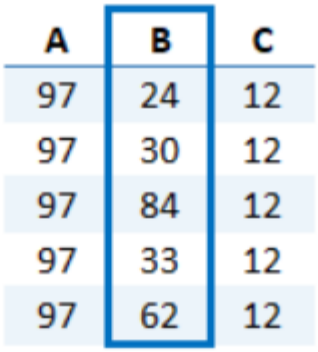
- Tìm các giá trị duy nhất của đặc trưng



B
24
30
84
33
62

Hình 14: Ví dụ tìm giá trị duy nhất trong ICE

- Đối với mỗi giá trị này, hãy tạo một phiên bản với các giá trị đặc trưng khác. Nói cách khác, sửa các giá trị đặc trưng khác và hoán vị giá trị của đặc trưng quan tâm



A	B	C
97	24	12
97	30	12
97	84	12
97	33	12
97	62	12

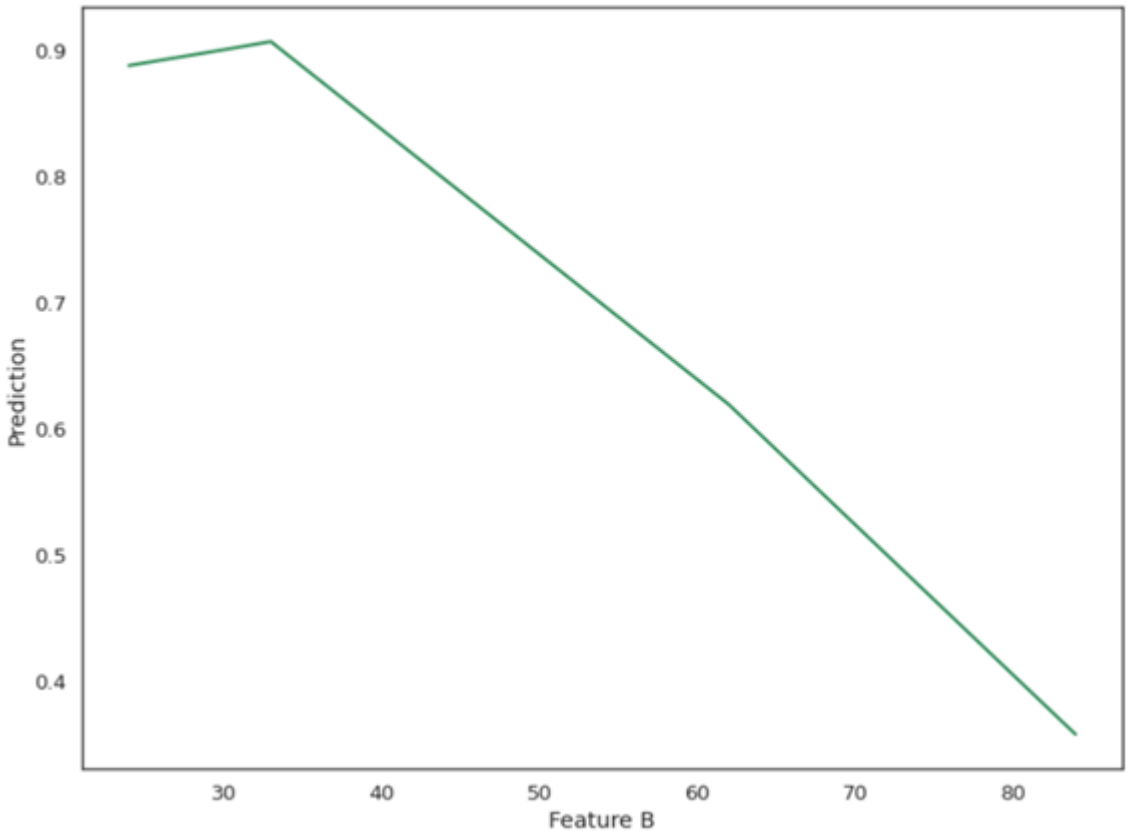
Hình 15: Ví dụ về sửa các giá trị đặc trưng khác

- Đưa ra dự đoán cho từng kết hợp

A	B	C	$\hat{y}$
97	24	12	0.89
97	30	12	0.90
97	33	12	0.91
97	62	12	0.62
97	84	12	0.36

Hình 16: Tính toán dự đoán

- Lấy các giá trị dự đoán cho từng trường hợp và vẽ đường cong cho các dự đoán



Hình 17: Đường cong minh hoạ trong ICE

2.2.1.4.3 Phác hoạ ICE đạo hàm (Derivative ICE Plot)

Phác hoạ ICE đạo hàm (d-ICE cho ta biết có sự thay đổi không và thay đổi theo hướng nào. Với phác hoạ này, ta dễ dàng nhận ra tập giá trị của đặc trưng mà dự đoán của mô hình hộp đen thay đổi (ít nhất cho vài điểm dữ liệu). Nếu không có tương tác giữa đặc trưng đang xét x_S và các đặc trưng khác x_C, hàm dự đoán có thể biểu diễn như sau:

$$\hat{f}(x) = \hat{f}(x_S, x_C) = g(x_S) + h(x_C), \quad \text{with} \quad \frac{\delta \hat{f}(x)}{\delta x_S} = g'(x_S)$$

Hình 18:Hàm dự đoán ICE đạo hàm

Không có tương tác, mỗi đạo hàm riêng sẽ giống nhau với mọi điểm dữ liệu. Nếu chúng khác nhau, điều này là có sự tương tác và có thể thấy được từ phác hoạ d-ICE. Ngoài việc dựng từng đường cong cho đạo hàm của hàm dự đoán, đưa ra độ lệch chuẩn của đạo hàm làm nổi bật vùng giá trị của đặc trưng có sự không đồng nhất. Phác hoạ d-ICE thường có chi phí tính toán cao và do đó không thực tế.

2.2.2 Global Explanations

2.2.2.1 Partial Dependence Plot (PDP)

2.2.1.1.1 Giới thiệu

Phác họa phụ thuộc riêng (partial dependence plot) (viết tắt là PDP hoặc phác họa PD) [14] thể hiện các ảnh hưởng biên (marginal effect) của một hoặc hai đặc trưng có trong dự đoán đầu ra của một mô hình học máy (J. H. Friedman 2001) [7]. Một phác họa phụ thuộc riêng thể hiện được mối quan hệ giữa một mục tiêu (target) và đặc trưng là tuyến tính, đơn điệu hoặc phức tạp hơn. Ví dụ, khi áp dụng trong một mô hình hồi quy tuyến tính (linear regression model), những đồ thị phụ thuộc riêng luôn luôn thể hiện mối quan hệ tuyến tính.

Đối với bài toán hồi quy, hàm phụ thuộc riêng (PD function) được định nghĩa như sau:

$$\hat{f}_{xS}(x_S) = E_{x_C}\left[\hat{f}(x_S, x_C)\right] = \int \hat{f}(x_S, x_C) d\mathbb{P}(x_C)$$

Hình 19 : Công thức tính hàm phụ thuộc riêng

Biến x_S là những đặc trưng mà phác họa PD cần phải biểu diễn và x_C là những đặc trưng khác được dùng trong mô hình học máy $\hat{f}$. Thông thường tập S chỉ chứa một hoặc hai đặc trưng. S là những đặc trưng mà ta muốn tìm hiểu ảnh hưởng của chúng lên phép dự đoán. Các véc tơ đặc trưng x_S và x_C được kết hợp để tạo thành toàn bộ không gian đặc trưng x. Sự phụ thuộc riêng (PD) hoạt động qua phép biên hóa (marginalization) ở đầu ra của mô hình học máy trên phân phối (distribution) của đặc trưng trong tập C, để cho hàm thể hiện quan hệ giữa các đặc trưng ta quan tâm trong tập S và dự đoán đầu ra. Bằng cách loại bỏ các đặc trưng khác, ta thu được một hàm phụ thuộc chỉ vào các đặc trưng trong tập S mà đã bao gồm sự tương tác (interaction) giữa các đặc trưng khác.

Hàm đặc trưng riêng (partial function) $\hat{f}_{x_s}$ được xấp xỉ bằng cách tính trung bình trong dữ liệu huấn luyện, hay còn gọi là phương pháp Monte Carlo:

$$\hat{f}_{x_S}(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)})$$

Hình 20: Phương pháp tính Monte Carlo

Hàm đặc trưng riêng cho ta biết giá trị trung bình của ảnh hưởng biên lên phép dự đoán ra sao với một (nhiều) giá trị cho trước của những đặc trưng trong S. Trong công thức này, $x_c^{(i)}$ là những giá trị đặc trưng (feature values) từ tập dữ liệu cho các đặc trưng mà chúng ta không quan tâm tới, và n là số lượng các mẫu dữ liệu (instances) trong bộ dữ liệu. Một giả định của PDP là những dặc trưng cửa tập C không có tương quan (correlated) với những đặc trưng của S. Nếu phép gỉa định này bị vi phạm, các giá trị trung bình tính toán cho phác họa đặc trưng riêng sẽ có những điểm dữ liệu (gần như) không khả thi.

Với bài toán phân lớp trong các mô hình học máy có kết quả xác suất (probabilities) ở đầu ra, phác họa đặc trưng riêng thể hiện xác suất của một lớp nhất định, cho trước các giá trị khác nhau của đặc trưng trong S. Cách đơn giản để thể hiện nhiều lớp là thể hiện 1

đường phác họa hoặc phác họa riêng cho mỗi lớp. Phác họa đặc trưng riêng là phương pháp toàn cục (global): Phương pháp này xem xét các mẫu dữ liệu và cho ra một tuyên bố về mối quan hệ toàn cục của một đặc trưng với dự đoán đầu ra.

2.2.2.1.2 Tầm quan trọng của đặc trưng dựa trên PDP

Greenwell và cộng sự. (2018) [8] đã đề xuất một thước đo tầm quan trọng của đặc trưng dựa trên sự phụ thuộc một phần đơn giản. Động lực cơ bản là một PDP phẳng chỉ ra rằng đặc trưng này không quan trọng và PDP càng thay đổi thì đặc trưng đó càng quan trọng. Đối với các đặc trưng số, tầm quan trọng được định nghĩa là độ lệch của từng giá trị đặc trưng duy nhất so với đường cong trung bình:

$$I(x_S) = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(\hat{f}_S(x_S^{(k)}) - \frac{1}{K}\sum_{k=1}^{K}\hat{f}_S(x_S^{(k)}))^2}$$

Hình 21: Cônh thức tính độ lệch của từng giá trị

Lưu ý rằng ở đây $x_S^{(k)}$ là K giá trị duy nhất của đặc trưng X_S. Đối với các đặc trưng phân loại, chúng tôi có:

$$I(x_S) = (max_k(\hat{f}_S(x_S^{(k)})) - min_k(\hat{f}_S(x_S^{(k)})))/4$$

Hình 22: Công thức tính độ lệch cho nhiều giá trị

Đây là phạm vi của các giá trị PDP cho các danh mục duy nhất chia cho bốn. Cách tính độ lệch kỳ lạ này được gọi là quy tắc phạm vi. Nó giúp ước tính sơ bộ độ lệch khi chúng ta chỉ biết phạm vi. Và mẫu số bốn đến từ phân phối chuẩn chuẩn: Trong phân phối chuẩn, 95% dữ liệu là trừ hai và cộng hai độ lệch chuẩn xung quanh giá trị trung bình. Vì vậy, phạm vi chia cho bốn đưa ra một ước tính sơ bộ có thể đánh giá thấp phương sai thực tế.

Tầm quan trọng của đặc trưng dựa trên PDP này nên được giải thích cẩn thận. Nó chỉ nắm bắt tác dụng chính của đặc trưng và bỏ qua các tương tác đặc trưng có thể xảy ra. Một đặc trưng có thể rất quan trọng dựa trên các phương pháp khác, chẳng hạn như tầm quan trọng của đặc trưng hoán vị, nhưng PDP có thể không thay đổi vì đặc trưng này ảnh hưởng đến dự đoán chủ yếu thông qua đặc trưng này với các đặc trưng khác. Một nhược điểm khác

của biện pháp này là nó được xác định trên các giá trị duy nhất. Một giá trị đặc trưng duy nhất chỉ với một trường hợp được đưa ra cùng trọng số trong tính toán tầm quan trọng như một giá trị với nhiều trường hợp.

2.2.2.2 Feature Inter-action

Khi các đặc trưng tương tác với nhau trong mô hình, dự đoán không thể biểu diễn thành tổng của ảnh hưởng của các đặc trưng, bởi vì ảnh hưởng của đặc trưng này phụ thuộc vào giá trị của đặc trưng khác. Khẳng định của Aristotle "Toàn bộ lớn hơn tổng của từng phần" được áp dụng khi xuất hiện sự tương tác [15]

2.2.2.2.1 Giới thiệu

Nếu một mô hình dự đoán dựa trên hai đặc trưng, ta có thể phân tích dự đoán thành 4 phần: một giá trị hằng số, một cho đặc trưng thứ nhất, một cho đặc trưng thứ hai, một cho tương tác giữa hai đặc trưng. Tương tác giữa hai đặc trưng là sự thay đổi ở dự đoán xảy ra khi ta thay đổi đặc trưng sau khi xem xét ảnh hưởng của từng đặc trưng. Lấy ví dụ một mô hình dự đoán giá của một ngôi nhà, dựa trên hai đặc trưng là kích cỡ (lớn hay nhỏ) và vị trí (tốt hay xấu), có thể đưa ra một trong bốn kết quả sau đây:

Location	Size	Prediction
tốt	lớn	300,000
tốt	nhỏ	200,000
xấu	lớn	250,000
xấu	nhỏ	150,000

Hình 23: Ví dụ về Feature Inter-action

Bây giờ ta hãy thử xem xét một ví dụ có sự tương tác:

Location	Size	Prediction
tốt	lớn	400,000
tốt	nhỏ	200,000
xấu	lớn	250,000
xấu	nhỏ	150,000

Hình 24: Ví dụ về Feature Inter-action

Tương tự, ta có thể phân tích dự đoán thành các phần sau: Giá trị hằng số (150,000), ảnh hưởng của kích cỡ (+100,000 nếu lớn, +0 nếu nhỏ), ảnh hưởng của vị trí (+50,000 nếu lớn, +0 nếu nhỏ). Tuy nhiên ở đây ta cần thêm một thành phần nữa cho sự tương tác: +100,000 nếu ngôi nhà có kích cỡ lớn và ở vị trí tốt. Có sự tương tác giữa kích cỡ và vị trí, do chênh lệch trong dự đoán của ngôi nhà có kích cỡ lớn và nhỏ phụ thuộc vào vị trí. Một cách để ước lượng mức độ tương tác là đo độ biến thiên của dự đoán phụ thuộc vào tương tác giữa các đặc trưng. Phép đo này gọi là thống kê-H, đề xuất bởi Friedman và Popescu (2008).

2.2.2.2.2 Lý thuyết: thống kê-H của Friedman

Chúng ta sẽ giải quyết hai trường hợp: một phép đo tương tác hai chiều cho ta biết liệu hai đặc trưng có tương tác với nhau không và tương tác trong trường hợp nào, một phép đo tương tác toàn phần cho ta biết liệu một đặc trưng có tương tác với tất cả các đặc trưng còn lại không và tương tác trong trường hợp nào. Về mặt lý thuyết, ta có thể đo sự tương tác giữa một số tùy ý các đặc trưng, nhưng hai trường hợp trên là đủ. Nếu hai đặc trưng không tương tác với nhau, ta có thể phân tích hàm phụ thuộc riêng như sau (giả sử hàm được căn tại không):

$$PD_{jk}(x_j, x_k) = PD_j(x_j) + PD_k(x_k)$$

Hình 25: Công thức tính hàm phụ thuộc riêng theo Friedman

với $PD_{jk}(x_j, x_k)$là hàm phụ thuộc riêng hai chiều của cả hai đặc trưng, $PD_j(x_j)$và $PD_k(x_k)$ là hàm đặc trưng riêng của từng đặc trưng. Tương tự, nếu đặc trưng đang xét không tương

tác với bất kì đặc trưng nào khác, ta có thể phân tích dự đoán thành tổng của hai hàm phụ thuộc riêng, một chỉ phụ thuộc vào đặc trưng j đang xét, một cho tất cả đặc trưng ngoại trừ j:

$$\hat{f}(x) = PD_j(x_j) + PD_{-j}(x_{-j})$$

Hình 26: Tổng hàm phụ thuộc riêng của nhiều giá trị

với $PD_{-j}(x_{-j})$ là hàm phụ thuộc riêng cho tất cả các đặc trưng ngoại trừ j.

Việc phân tích này giúp biể diễn hàm phụ thuộc riêng (hoặc toàn bộ dự đoán) mà không có tương tác (giữa đặc trưng j và k, hoặc giữa đặc trưng j và tất cả đặc trưng còn lại). Sau đó, ta sẽ đo sự chênh lệch giữa hàm phụ thuộc riêng quan sát được với hàm đã được phân tích bên trên. Ta tính phương sai của kết quả cho hàm phụ thuộc riêng (để đo tương tác giữa hai đặc trưng) hoặc toàn bộ hàm dự đoán (để đo tương tác giữa một đặc trưng và tất cả đặc trưng còn lại). Phương sai tính qua tương tác (chênh lệch giữa hàm phụ thuộc riêng quan sát và không có tương tác) được dùng làm thống kê có tương tác. Thống kê bằng 0 nếu không có tương tác, bằng 1 nếu tất cả phương sai của P Djk hoặc ˆf giải thích qua tổng của các hàm phụ thuộc riêng. Giá trị thống kê có tương tác giữa hai đặc trưng bằng 1 nghĩa là từng

hàm phụ thuộc riêng là hằng số và dự đoán chỉ bị ảnh hưởng bởi tương tác. Thống kê-H có thể lớn hơn 1, lúc này sẽ khó giải thích hơn. Điều này xảy ra khi phương sai của tương tác hai phía lớn hơn phương sai của đồ thị phụ thuộc riêng hai chiều.

Về mặt công thức, thống kê-H đề xuất bởi Friedman và Popescu cho tương tác giữa hai đặc trưng j và k được tính như sau:

$$H_{jk}^2 = \sum_{i=1}^{n} \left[PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)}) \right]^2 / \sum_{i=1}^{n} PD_{jk}^2(x_j^{(i)}, x_k^{(i)})$$

Hình 27: Công thức tính tương tác giữa hai đặc trưng j và k

Tương tự ta có thể tính cho tương tác giữa đặc trưng j và tất cả đặc trưng còn lại:

$$H_j^2 = \sum_{i=1}^{n} \left[\hat{f}(x^{(i)}) - PD_j(x_j^{(i)}) - PD_{-j}(x_{-j}^{(i)}) \right]^2 / \sum_{i=1}^{n} \hat{f}^2(x^{(i)})$$

Hình 28: Công thức tính tương tác giữa j và các đặc trưng còn lại

Thống kê-H có chi phí tính toán khá cao vì ta cần phải xét từng điểm dữ liệu, và với mỗi điểm dữ liệu ta cần tính hàm phụ thuộc riêng qua n điểm dữ liệu. Trường hợp tệ nhất, ta cần gọi hàm dự đoán của mô hình $2n^2$lần để tính thống kê-H hai phía (j với k) và $3n^2$lần cho thống kê-H toàn phần (j với tất cả). Để tăng tốc tính toán, ta có thể lấy mẫu từ n điểm dữ liệu. Tuy nhiên, điều này khiến phương sai khi ước lượng phụ thuộc riêng tăng lên, dẫn đến thống kê-H không ổn định. Do đó, nếu ta lấy mẫu để giảm chi phí tính toán, cần phải lấy đủ số mẫu. Friedman và Popescu cũng đề xuất một phương pháp kiểm định giả thiết thống kê để đánh giá xem thống kê-H có khác 0 đủ nhiều không. Giả thiết không là không có sự tương tác. Để sinh ra thống kê tương tác dưới giả thiết không, ta phải thay đổi mô hình sau cho không có tương tác giữa đặc trưng j với k hoặc với các đặc trưng còn lại. Không phải mô hình nào cũng làm được điều này. Cách kiểm định này do vậy phụ thuộc vào mô hình, và sẽ không được trình bày ở đây. Thống kê có tương tác cũng có thể áp dụng cho bài toán phân lớp nếu dự đoán là giá trị xác suất.

CHƯƠNG 3: THỰC NGHIỆM

3.1 BÀI TOÁN

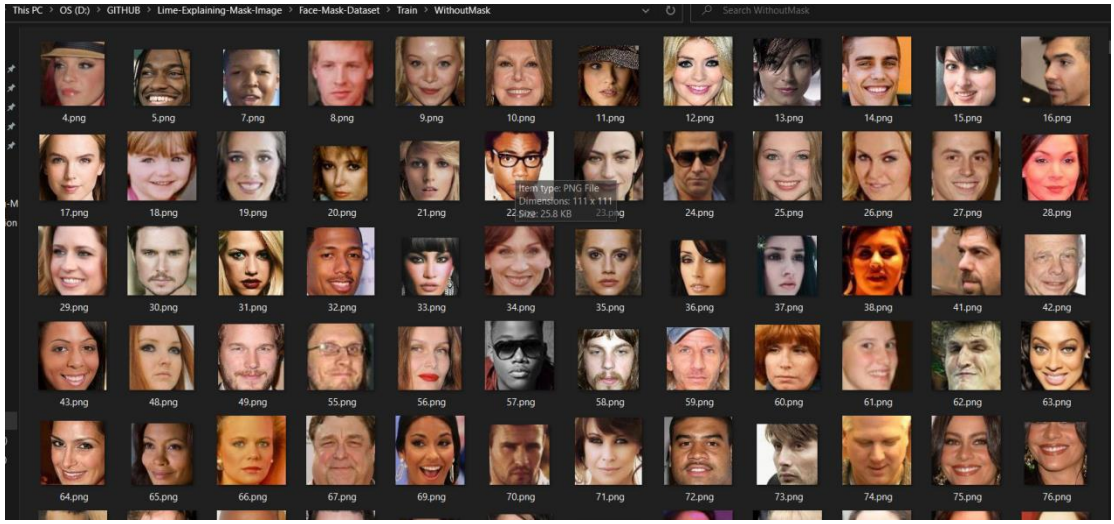
Trong một khoảng thời gian dài, tình hình dịch bệnh COVID diễn ra trên nhiều nơi trên thế giới. Một trong những phương pháp phòng chống lây lan bệnh là đeo khẩu trang y tế ở những nơi công cộng để tránh lẫy nhiễm bệnh. Sức mạnh của máy tính hiện nay có thể giúp chúng ta trong việc phát hiện người không đeo khẩu trang nơi làm việc, khu công cộng. Sức mạnh của các mô hình "deep learning" đã giúp chúng ta tạo ra các thiết bị nhận diện người đeo khẩu trang thông qua camera. Nhóm muốn tạo ra một mô hình nhận diện đeo khẩu trang thông qua hình ảnh và thông qua mô hình đó nhóm dùng mô hình thuật toán "XAI" để giải thích mô hình nhận diện đó. Thông qua thuật toán đó, nhóm có thể giải thích cách mà mô hình ra một quyết định.

3.2 DỮ LIỆU

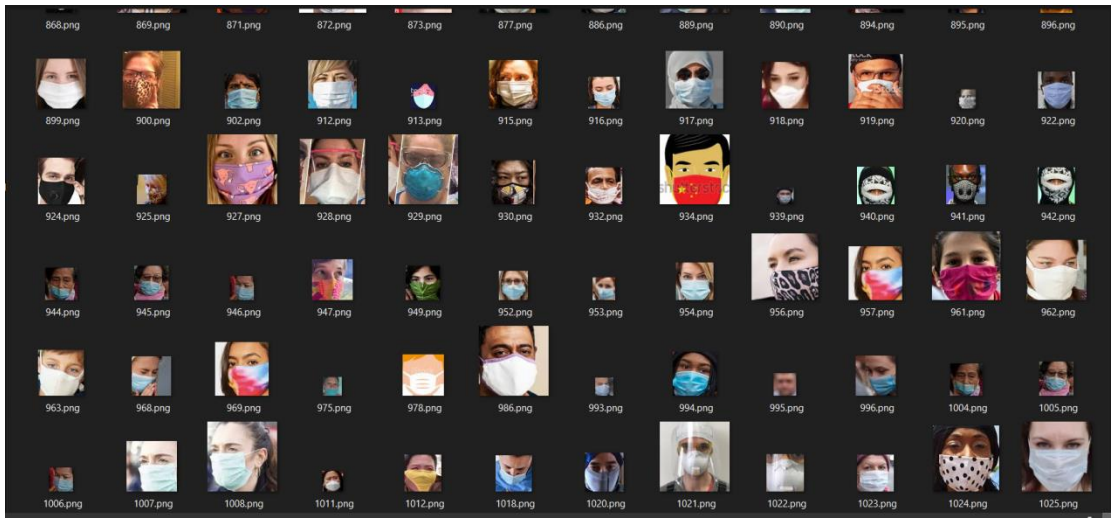
Tập dữ liệu được nhóm lấy để xây dựng mô hình từ nguồn: https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset.Tập dữ liệu này gồm 12 nghìn ảnh với dạng files.pnp có nhiều kích cỡ ảnh khác nhau.

Tập dữ liệu chia làm 3 folder. Mỗi folder sẽ chia là 2 folder có tên là "with mask" và "without mask". Đây cũng chinh 2 nhãn mà mô hình cần dự đoán.

Folder thứ nhất là folder tập dữ liệu để train có 10000 ảnh chia đều cho 2 nhãn cần dự đoán.



Hình 29: Ảnh trong folder tập dữ liệu train với nhãn "without mask"



Hình 30: Ảnh trong folder tập dữ liệu train với nhãn "with mask"

Tương tự dữ liệu trong các folder dữ liệu test, validation có 2 folder với tên 2 nhãn cần dự đoán nhưng số lượng ảnh khoảng 2 000 ảnh .

3.3 PHƯƠNG PHÁP VÀ KẾT QUẢ

Trong bài toán này nhóm chọn mô hình thử nghiệm như sau :

- Deep learning: CNN (Convolutional Neural Networks) [2]
- Explainable AI: LIME (Local Interpretable Model-agnostic Explanations)

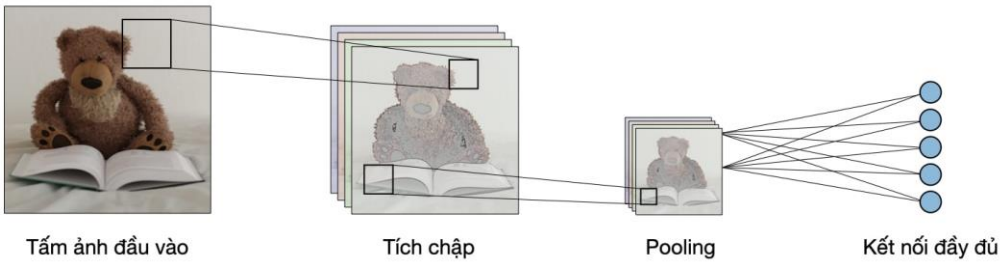
3.3.1 Mô hình CNN

3.3.1.1 Giới thiệu

CNN (Convolutional neural networks): là một trong những mô hình học sâu được dùng phổ biến hiện nay. Mô hình này chứa nhiều lớp phức tạp có thế kết nối hoàn toàn hoặc được gộp chung. Các lớp phức hợp này thực hiện trích xuất các đặc trưng nổi trội của dữ liệu đầu vào. Chính từ ưu điểm đó, chúng thường được sử dụng để xử lý ảnh và cho thấy sự chính xác rất cao. Mô hình được sử dụng trong nhiều ứng dụng tiến tiến của trí tuệ nhân tạo, bao gồm nhận dạng khuôn mặt, xử lí ngôn ngữ tự nhiên và còn nhiều lĩnh vực khác.

Một mạng nơron mà mỗi nút ở tầng n được kết nối với tất cả các nốt ở tầng n-1 và tầng n+1, được gọi là fully connected neural networks. Khi dùng một fully connected neural networks để giải quyết một bài toán xử lí ảnh sẽ sinh ra vấn đề là số lượng tham số (parameter hay còn được gọi là param) rất lớn. Lấy ví dụ: Ta có một ảnh RGB có kích thước 200x200, thì ta có 120000 pixel (200 x 200 x 3). Tức là input layer ta có 120000 nút. Giả sử số lượng nút hidden layer 1 là 1000. Số lượng trọng số *W* giữa lớp input layer và lớp hidden layer 1 sẽ là 120000 *1000=120000000. Số lượng param trong một mô hình hình còn nhiều hơn số lượng đó nhiều lần vì chúng ta chỉ mới tính ở giữa lớp input layer và hidden layer. **CNN** sử dụng phép tích chập để giảm khối lượng phép tính cũng như số lượng param trong một layer, giúp trích xuất và giữ lại các đặc trựng của ảnh. Từ đó, việc phân loại hay nhận dạng diễn ra tốt hơn. Với các bộ lọc khác nhau, chúng ta thu được các đặc trưng của của một ảnh như đặc điểm của vật thể, gam màu, độ sáng, … Mạng **CNN** thường dùng cho các bài toán xử lí ảnh, chính ví thế chúng ta cùng tìm hiểu về kiến trúc của nó và cách nó hoạt động.

Kiến trúc của mô hình **CNN** truyền thống là một loại neural được cấu hình từ các lớp sau như hình 3.1



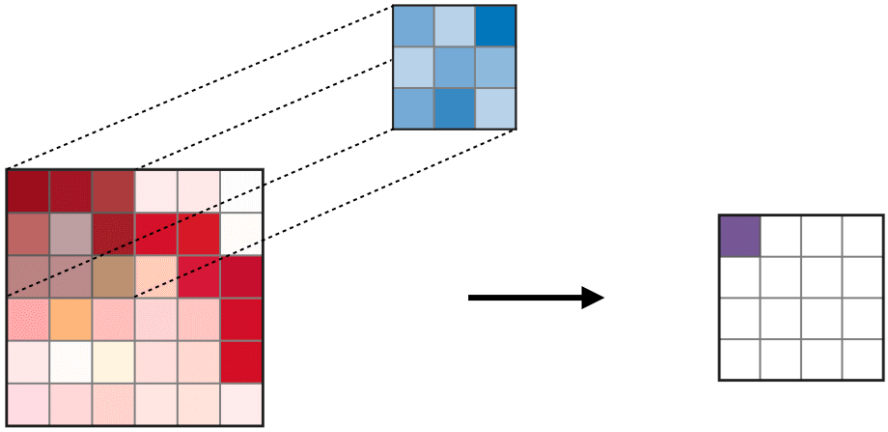
Hình 31: Kiến trúc mạng CNN truyền thống [3]

Lớp tính chập (thường được gọi tắt là lớp CONV) và lớp pooling có thể điều chỉnh theo các siêu tham số (hyperparammeter) được nói rõ ở phần tiếp theo.

3.3.1.2 Các lớp trong mô hình CNN

- Lớp tích chập (CONV):

  Lớp tích chập là lớp đầu tiên trích xuất các đặc trưng của ảnh. Nó duy trì các mối quan hệ giữa các pixel bằng cách tính các đặc trưng của ảnh bằng sử dụng các ô vuông nhỏ của dữ liệu đầu vào. Lớp tích chập sử dụng các bộ *K* (kích thước ô vuông NxN) để thực hiện phép tích chập khi đưa *K* đi qua đầu vào *I* (là kích thức của ảnh MxM theo chiều từ trái sang phải, từ trên xuống dưới. Các hyperparameter của các bộ lọc này bao gồm kích thướtc bộ lọc *K* và độ trượt(strike). Kết quả thu được gọi là feature map hay là activation map. Hình 32 sẽ cho thấy quá trình thực hiện phép tích chập giữa ảnh dầu vào *I* và bộ lọc *K*:

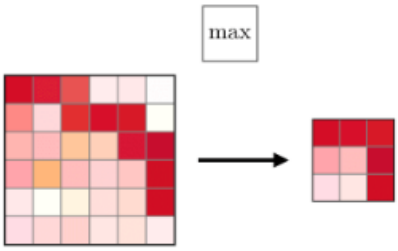
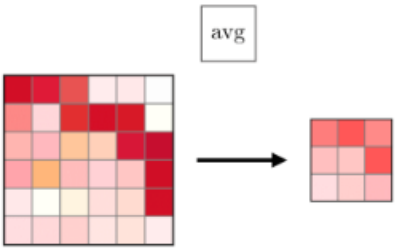


Hình 32: Mô tả phép tích chập ở lớp CONV [3]

- Lớp Pooling (POOL):

  Lớp Pooling là lớp giảm bớt số lượng param. Là phép giảm kích thước của một mẫu (downsampling). Thường dùng sau lớp tích chập, giúp tang tính bất biến của không

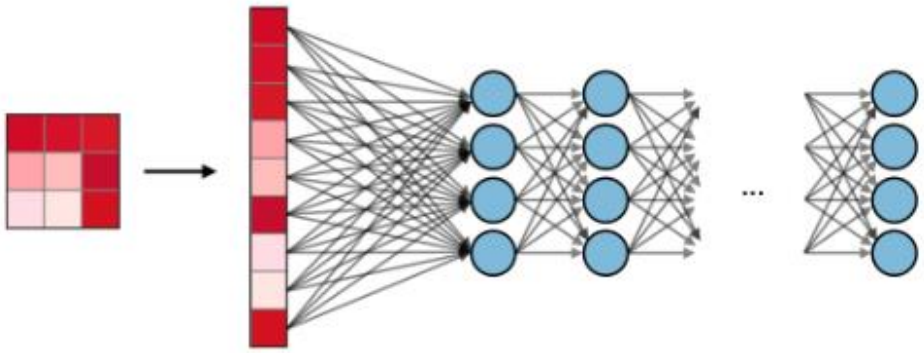
gian. Cụ thể, max pooling, average pooling và sum pooling là những dạng pooling thường gặp, tương ứng với giá trị lớn nhất, trung bình, tổn giá trị được lấy. Trong **CNN,** chúng ta thường dùng max với average pooling. Ở mức độ end user chúng ta có thể hiểu ý nghĩa cảu lớp này là giúp chúng ta lấy các đặc trưng nổi trội trong các feature map hoặc tính trung bình các đặc trưng đó.

Kiểu	Max pooling	Average pooling
Chức năng	Từng phép pooling chọn giá trị lớn nhất trong khu vực mà nó đang được áp dụng	Từng phép pooling tính trung bình các giá trị trong khu vực mà nó đang được áp dụng
Minh họa		
Nhận xét	• Bảo toàn các đặc trưng đã phát hiện • Được sử dụng thường xuyên	• Giảm kích thước feature map • Được sử dụng trong mạng LeNet

Hình 33: Mô tả phép tích chập ở lớp CONV [2]

- Lớp Fully Connected (FC):

  Lớp Fully Connected là lớp nhận dữ liệu đã được làm phẳng, mỗi đầu vào được kết nối với tất cả các neural. Trong mô hình mạng **CNN**, tầng kết nối đầy đủ thường nằm ở cuối mạng và tối ưu hóa mục tiêu của mạng.

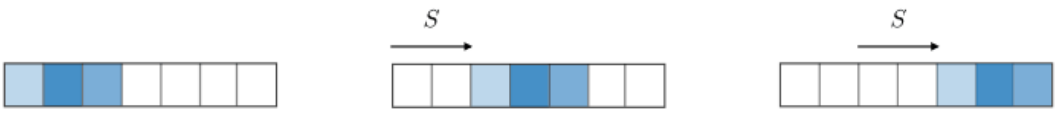


Hình 34: Mô tả lớp Fully Connected [2]

3.3.1.3 Các siêu tham sô của bộ lọc

Mỗi tầng CONV có chứa bộ lọc *K*. Bộ lọc này chứa các siêu tham số như strike, padding.

- Strike: là số pixel thay đổi trên ma trận đầu vào. Khi ta strike = 2 thì ta di chuyển các kernel hai pixel sau mỗi lần thực hiện phép tính. Ở hình 35 sẽ mô tả cách hoạt động của Strike.

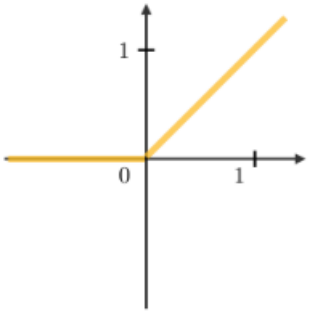
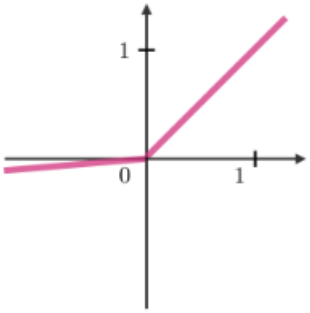
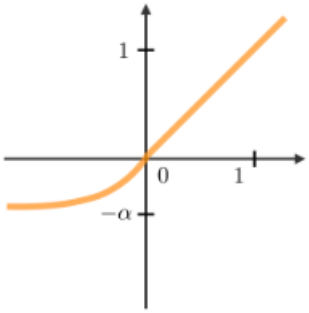


Hình 35: Mô tả hoạt động của Strike = 2 [3]

- Zera-padding: là quá trình thêm N số không vào các biên của ảnh đầu vào. Sau khi thực hiện phép tích chập, các feature map sẽ có kich thức nhỏ hơn kích thước ban đầu. Để tiếp tục ở lớp tích chập tiếp theo, chúng ta cần thêm N số không vào viền các feature map vừa tính được đề nó cùng kích thước với I, cụ thể đó là N = 2. Zero-padding còn giúp giải quyết một vấn đề khác là khi xét các giá trị ở biên của đầu vào I chúng ta sẽ không thể tìm ra một ma trận với tâm là điểm đó và có kích thước bằng bộ lọc K, chính vì thế chúng ta sẽ sử dụng padding để thêm viền 0 vào I để có thể sử dụng cả các pixel ở biên của đầu vào cho phép tích chập.

3.3.1.4 Các hàm kích hoạt thường gặp trong CNN

- Rectified Linear Unit (ReLU): là một hàm kích hoạt thường dùng trong các mô hình **CNN.** Mục đích của hàm là nó tăng tính phi tuyến cho mạng. Các biến thể khác của ReLU được mô tả ở hình 36.

ReLU	Leaky ReLU	ELU
$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ với $\epsilon \ll 1$	$g(z) = \max(\alpha(e^z - 1), z)$ với $\alpha \ll 1$
		
• Độ phức tạp phi tuyến tính có thể thông dịch được về mặt sinh học	• Gán vấn đề ReLU chết cho những giá trị âm	• Khả vi tại mọi nơi

Hình 36: Các loại hàm ReLu và chức năng từng hàm [2]

- Softmax: hàm softmax là một hàm khái quát hóa của hàm logistic, lấy đầu vào là một vector chứa các giá trị x ∈ R^n và đưa ra một vector gồm các xác suất p ∈ R^n thông qua một hàm softmax ở tầng cuối. Nó được định nghĩa như sau:

$$y = \begin{bmatrix} p_1 \\ p_2 \\ . \\ . \\ p_n \end{bmatrix} \text{với } p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

3.3.2 Xây dựng mô hình cho bài toán

3.3.2.1 Kiến trúc mô hình

Đầu tiên chúng tôi xây dựng mô hinh CNN với kiến trúc như sau:

+ 5 lớp tích chập (CONV).

+ 5 lớp pooling.

+ 1 lớp flatten.

+ 2 lớp dense.

Kiến trúc mô hình được minh họa ở hình 37.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 198, 198, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 99, 99, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 97, 97, 32)        4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 48, 48, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 46, 46, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 23, 23, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 21, 21, 64)        36928
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 10, 10, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 8, 8, 64)          36928
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 128)               131200
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 258
=================================================================
Total params: 228,898
Trainable params: 228,898
Non-trainable params: 0
_________________________________________________________________
```

Hình 37: Kiến trúc mô hình CNN xây dựng

Sau khi thực hiện thử nghiệm nhiều siêu tham số ở lớp CONV, kích thước của lớp pooling và tăng số epoch thì chúng tôi nhận kết quả rất tốt với các thông số như hình dưới và với epoch là 30 trên tập validation.

```
model = tf.keras.models.Sequential([
    # Note the input shape is the desired size of the image 200x 200 with 3 bytes color
    # The first convolution
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(200, 200, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    # The second convolution
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The third convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fourth convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # The fifth convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    # Flatten the results to feed into a dense layer
    tf.keras.layers.Flatten(),
    # 128 neuron in the fully-connected layer
    tf.keras.layers.Dense(128, activation='relu'),
    # 2 output neurons for 2 classes with the softmax activation
    tf.keras.layers.Dense(2, activation='softmax')
])
```

Hình 38: Mô tả các siêu tham số của mô hình

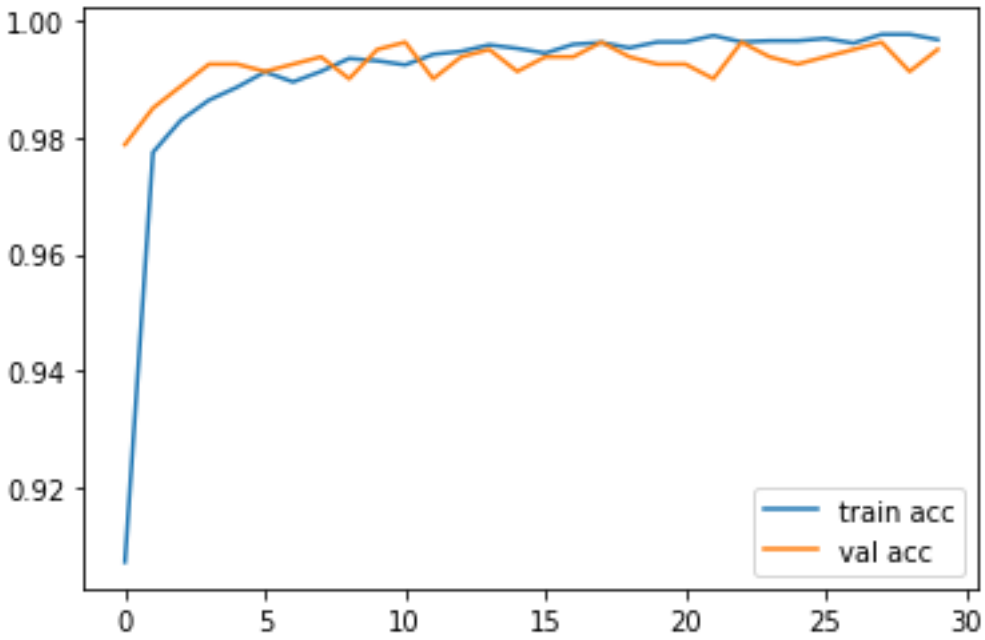
```
ImageFile.LOAD_TRUNCATED_IMAGES = True
history = model.fit(
        train_generator,
        steps_per_epoch=int(total_sample/batch_size),
        epochs=n_epochs,
        validation_data=val_set,
        verbose=1)
```

```
Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/30
312/312 [==============================] - 333s 1s/step - loss: 0.3843 - acc: 0.8176 - val_loss: 0.0533 - val_acc: 0.9787
Epoch 2/30
312/312 [==============================] - 181s 581ms/step - loss: 0.0709 - acc: 0.9747 - val_loss: 0.0515 - val_acc: 0.9850
Epoch 3/30
312/312 [==============================] - 183s 586ms/step - loss: 0.0519 - acc: 0.9826 - val_loss: 0.0288 - val_acc: 0.9887
Epoch 4/30
312/312 [==============================] - 184s 588ms/step - loss: 0.0432 - acc: 0.9869 - val_loss: 0.0193 - val_acc: 0.9925
Epoch 5/30
312/312 [==============================] - 181s 581ms/step - loss: 0.0364 - acc: 0.9887 - val_loss: 0.0306 - val_acc: 0.9925
Epoch 6/30
312/312 [==============================] - 181s 580ms/step - loss: 0.0361 - acc: 0.9909 - val_loss: 0.0300 - val_acc: 0.9912
Epoch 7/30
```

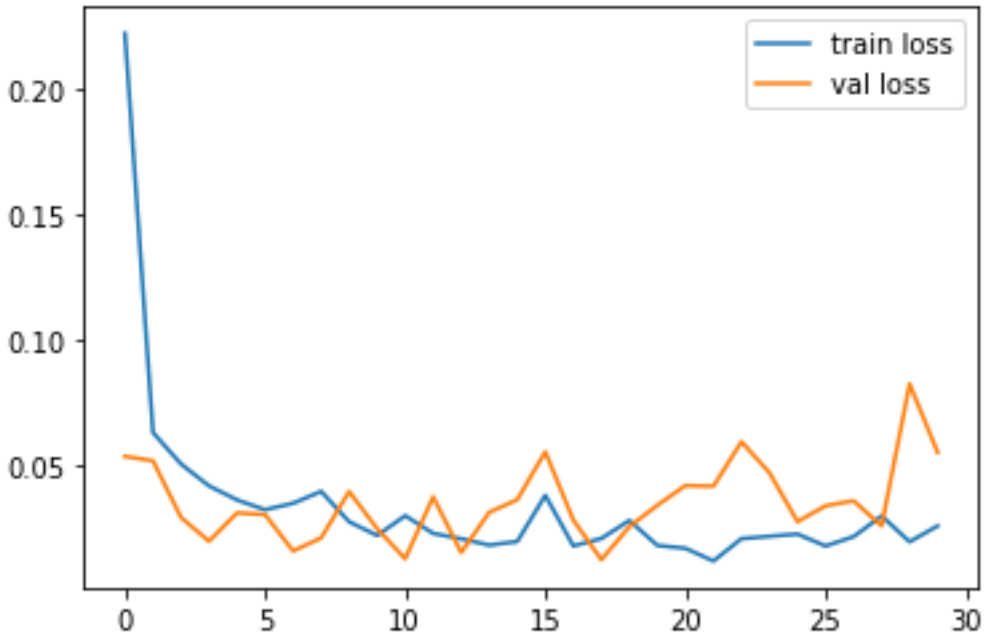
Hình 39: Kết quả huấn luyện mô hình với epoch = 30

3.3.2.2 Kết quả

Sau khi huấn luyện mô hình ta thu được kết quả trên tập validation có độ chính xác khoảng 99%. Ta có thể thấy qua 2 biểu đồ sau:



Hình 40: Biểu đồ thể độ chính xác trên tập train và tập validation.



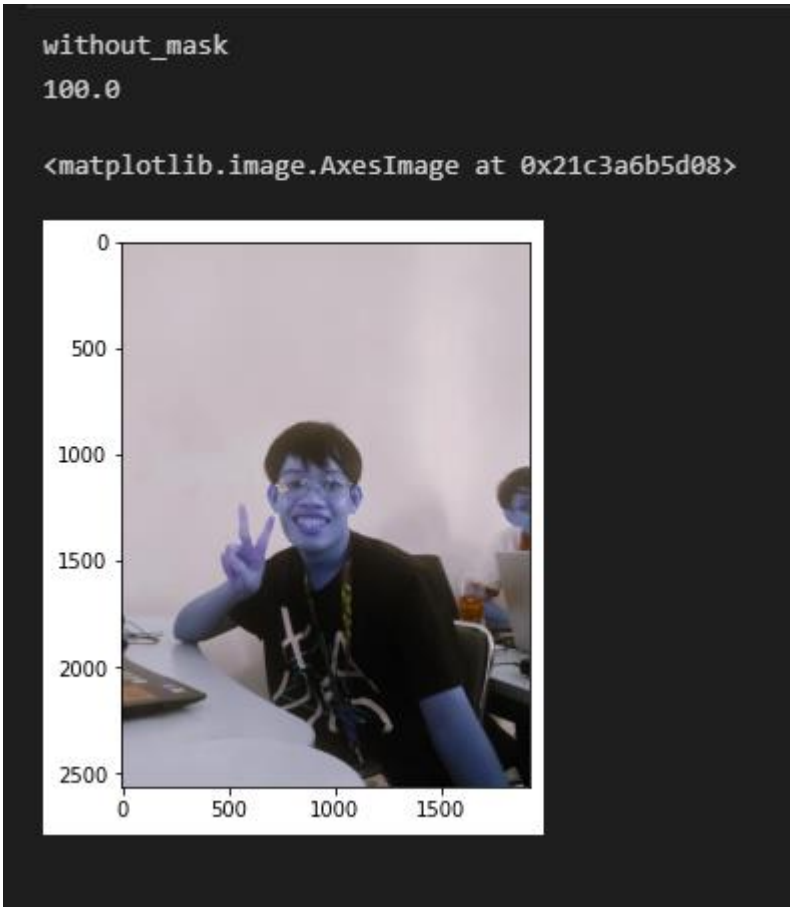
Hình 41: Biểu đò thể hiện độ mất mát ở tập train và tập validation

Kết quả dự đoán trên tập test là khoảng 98.7%

```
loss, accuracy = model.evaluate(test_set)
print('Test accuracy:' , accuracy * 100,'%')
```

```
[9]  ✓ 18.9s

31/31 [==============================] - 18s 604ms/step - loss: 0.1243 - acc: 0.9869
Test accuracy: 98.68951439857483 %
```

Hình 42: Kết quả trên tập test



Hình 43: Kết quả trên một ảnh

Chúng tôi quyết định dùng mô hình **CNN** vừa xây dựng bên trên dùng để làm mô hình phức tạp để cho thuật toán **LIME** giải thích cách thức mô hình đưa ra quyết định là người trong ảnh không đeo khẩu trang hoặc đeo khẩu trang.

3.3.3 Thuật toán LIME và kết quả.

3.3.3.1 Cách thuật toán LIME làm việc.

1. Hoán vị dữ liệu.

2. Tính toán chênh lệch giữa dữ liệu hoán vị và quan sát dữ liệu ban đầu.

3. Đưa ra dự đoán về dữ liệu mới bằng mô hình phức tạp.

4. Chọn m các đặc trưng mô tả tốt nhất kết quả mô hình phức tạp từ dữ liệu được hoán vị.

5. Khớp với một mô hình đơn giản với dữ liệu hoán vị bới m đặc trưng và điểm tương đồng dưới dạng trọng số.

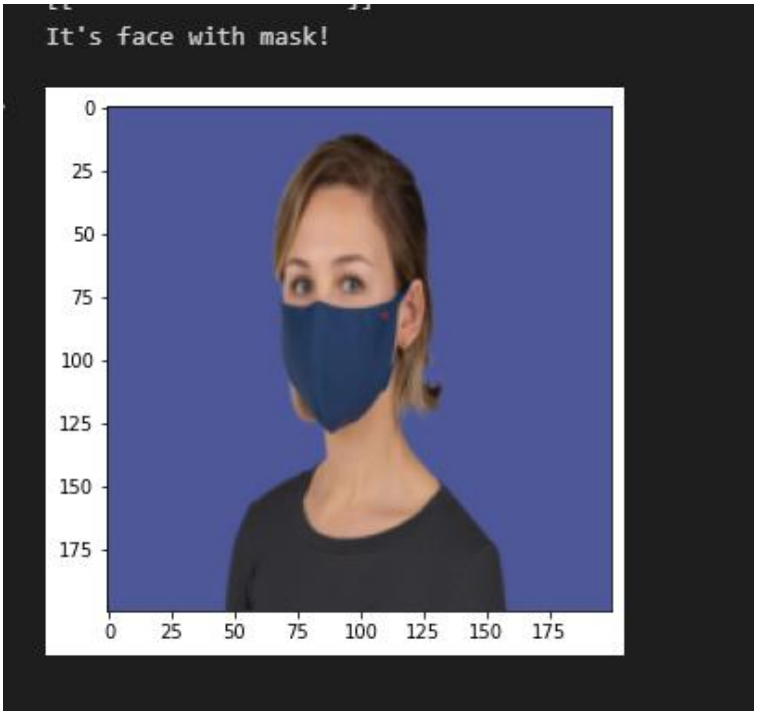
6. Đặc trưng trộng số từ mô hình đơn giản sẽ giải thích cho hành vi cục bộ của mô hình phức tap.

*3.3.3.2 Áp dụng cho mô hình nhận diện khẩu trang***.**

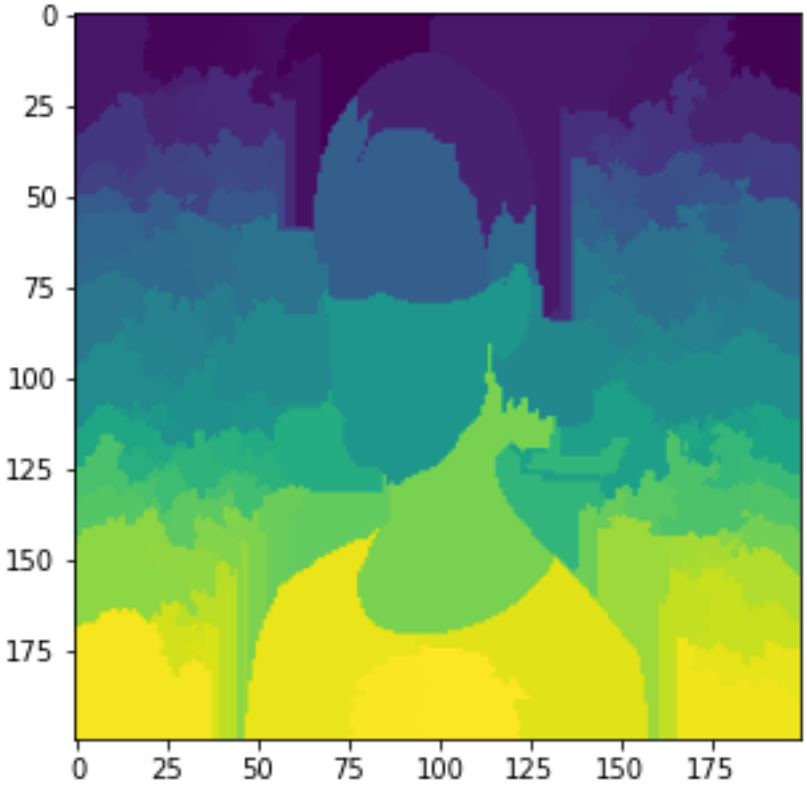
*3.3.3.2.1. Hoán vị dữ liệu***.**

Trước tiên, chúng ta phải biết các đặc trưng của một ảnh chúng ta dùng. Giá trị pixel có thể xem như là một đặc trưng của ảnh. Nhưng trong thuật toán **LIME** chúng tôi dùng superpixel (phân vùng ảnh) [4] là đặc trưng. Superpixel nhóm các pixel có điểm chung lại. Vì chúng cho ta nhiều thông tin hơn các pixel. Có rất nhiều cách để phân vùng ảnh pixel của một ảnh như [5], Quickshift và còn nhiều thuật toán khác có thể phân vùng ảnh.

Ở đây chúng tôi dùng thuật toán Quickshift và chia hình thành 130 superpixels sau là hình ảnh kết quả thu được:



Hình 44: Ảnh ban đầu



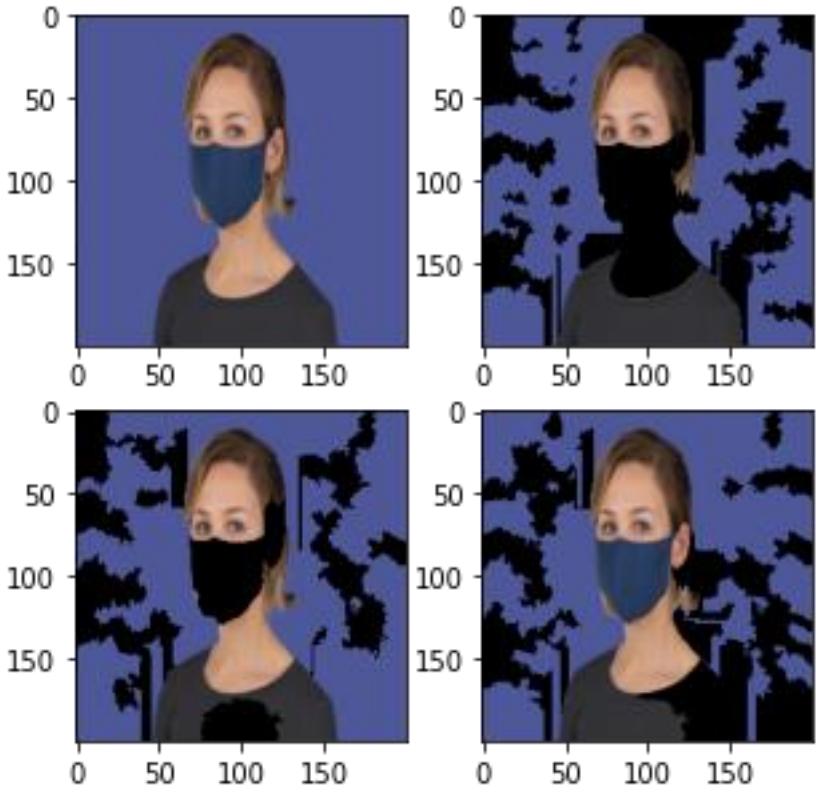
Hình 45: Kết quả sau khi phân vùng ảnh

Sau đó, chúng tôi bắt đầu tạo ảnh hoán vị từ ảnh ban đầu. Ảnh hoán vị được tạo ra từ việc tắt ngẫu nhiên một số superpixel trong ảnh gốc ban đầu. Ở đây chúng tôi tạo ra khoảng 1000 ảnh hoán vị và nhãn cho mô hình **CNN** ở trên dự đoán. Sau dâu là hình một ảnh hoán vị.

```
array([1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1,
       1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1,
       0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1,
       1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1])
```

Hình 46: Kết quả của một ảnh hoán vị

Ở dây 1 có ý nghĩa là superpixel tại vị trí đó bật và 0 có nghĩa là superpixel tại vị trí đó tắt trong ảnh hoán vị. Hình 47 mô tả ảnh gốc và ảnh hoán vị.



Hình 47: Kết quả của một ảnh hoán vị

3.3.3.2.2. Tính khoảng cách giữa ảnh gốc và ảnh hoán vị

Chúng tôi dùng "cosine distances" [6] để tính khoàng cách ảnh gốc và ảnh hoán vị.

3.3.3.2.3. Tính trọng số cho ảnh hoán vị

Sau khi chúng tôi có được tập ảnh hoán vị. Những ảnh có gần ảnh gốc thì chúng tôi đánh trọng số cao và ngược lại với ảnh xa với ảnh gốc. “Exponential kernel” [7] dùng với độ rộng kernel là 25 cho trọng số. Dựa vào độ rộng kernel giúp chúng ta xác định “locality” (vùng cục bộ) xung quanh ảnh gốc.

```
# Exponential kernel
def kernel(d, kernel_width):
    return np.sqrt(np.exp(-(d ** 2) / kernel_width ** 2))


# exponential kernel with kernel width 25
kernel_fn = partial(kernel, kernel_width=25)


# Samples are given weights using exponential kernel
weights=kernel_fn(distances)
```

Hình 48: Ảnh code tính trọng số các ảnh hoán vị

.

3.3.3.2.4. Chọn các đặc trưng quan trọng

Việc chọn chọn các đặc trưng quan trọng bằng việc dùng các mô hình cục bộ. Trọng số trong mô hình tuyến tính dùng trong dữ liệu ở trên (dữ liệu hoán vị được dự đoán bằng mô hình **CNN** và trọng số). Trên thực tế, chúng ta có rất nhiều cách chọn ra top các đặc trưng ví dụ như lấy chọn chuyển tiếp, loại bỏ ngược, thậm chí chúng ta cũng có thể dùng “regularized linear” như Lasso và Ridge regression. Ở đây, chúng tôi dùng Ridge regression cho bộ dữ liệu và thu 10 đặc trưng hoặc các superpixel như minh họa ở hình 48.

```
array([ 49,  76,  98,  57,  75,  73,  26,  39,  60, 103], dtype=int64)
```

Hình 49: Top 10 đặc trưng (hoặc superpixel)

3.3.3.2.5 Dự đoán trên mô hình tuyến tính cục bộ

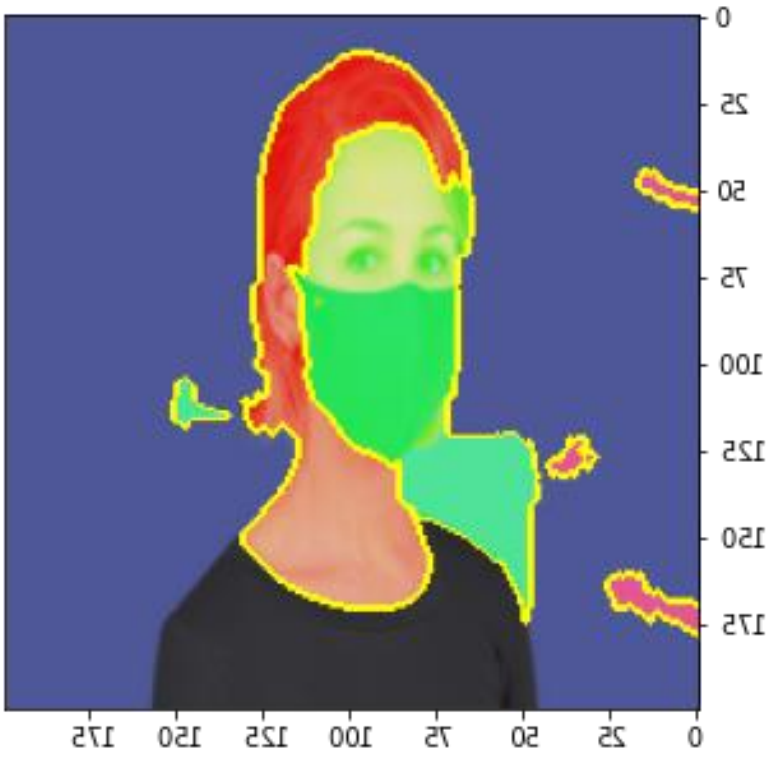
Sau khi có 10 đặc trưng hàng đầu thì một mô hình tuyến tính cục bộ có trọng số, chúng tôi tiến hành giải thích dự đoán của mô hình phức tạp trong khu vực cục bộ của ảnh gốc. Các thông tin về mô hình tuyến tính được minh họa ở hình 50.

```
# After getting the features Ridge regression is used to fit the local model

model_regressor = Ridge(alpha=0.01, fit_intercept=True)

easy_model = model_regressor
easy_model.fit(data[:, top_features],
               labels_column, sample_weight=weights)
prediction_score = easy_model.score(
    data[:, top_features],
    labels_column, sample_weight=weights)

local_pred = easy_model.predict(data[0, top_features].reshape(1, -1))
```

Hình 50: Thông tin về mô hình tuyến tính

3.3.3.2.6 Kết quả

Hình 51 mô tả kết quả của cuối cùng chúng tôi. Chúng tôi muốn thấy 10 superpixel quan trọng trong ảnh. Các superpixel có màu xanh lá là các đặc điểm có ảnh hưởng tích cực đến dự đoán của mô hình **CNN** tại hình này là kết quả "đeo khẩu trang" và ngược lại các superpixel có màu đỏ là đặc điểm có ảnh hưởng tiêu cực đến dự đoán kết quả của mô hình



Hình 51: Kết quả giải thích của LIME

Từ giải thích trên, chúng tôi nhận thấy các superpixel gần mũi và miệng có ảnh hưởng tích cực đến dự đoán của mô hình CNN của chúng tôi. Đặt biệt, trong trường hợp này là với nhãn “đeo khẩu trang”. Đồng thời, chúng tôi nhận thấy các superpixel gần tai có đóng góp tiêu cực đến kết quả dự đoán. Từ đó, chúng ta thấy được lợi ích của **LIME** cho việc tối ưu hóa các hành vi của mô hình phức tạp.

CHƯƠNG 4: KẾT LUẬN

4.1 KẾT QUẢ ĐẠT ĐƯỢC

4.1.1 Ý nghĩa khoa học

Báo cáo đã trình bày các cơ sở lí thuyết về "Explanable Artifical Intelligence" và lí thuyết cơ bản về CNN cho phần ứng dụng. Nội dung chính của đề tài là trình bày cơ sở lý thuyết các phương pháp thường dùng để giải thích một mô hình học máy. Ngoài ra còn cho thấy ứng dụng của bài toán vào giải thích được kết quả nhận diện hình ảnh thông qua phương pháp LIME. Thông qua đề tài, chúng tôi nắm bắt được các phương pháp cơ bản trong XAI dùng để xử giải thích cho các mô hình học máy có thể giải thích được. Và thông qua ứng dụng demo, chúng tôi cũng nâng thêm hiểu biết và kỹ năng sử dụng python cùng các thư viện hỗ trợ để tạo các mô hình deep learning và phân tích dữ liệu. Bên cạnh đó, chúng tôi còn nâng cao thêm được khả năng đọc hiểu tài liệu, khả năng làm việc nhóm và khả năng trình bày báo cáo khoa học.

4.1.2 Ý nghĩa thực tiễn

Chúng tôi biết được nhiều phương pháp, mô hình để diễn giải được các thuật toán học máy và ứng dụng được vào bài toán cụ thể. Thông qua việc thực hiện đề tài, chúng tôi biết được việc diễn giải học máy đang trở thành một trong những hướng nghiên cứu rất phát triển đặc biệt là tại các doanh nghiệp. Khả năng diễn giải của mô hình có nhiều ý nghĩa trong các lĩnh vực như y tế, bảo hiểm, tài chính, ... Trong các lĩnh vưc này, việc giải thích tại sao mô hình đưa ra những quyết định có thể người quản lý đưa ra các giải pháp kinh doanh hiệu quả, hoặc các bác sỹ đưa ra các phương pháp điều trị kịp thời cho người bệnh.

Bên cạnh đó, sau khi thực hiện xây dựng phương pháp LIME cho nhận diện ảnh giúp chúng tôi biết nhiều hơn về thư viện Tensorflow, biết rõ hơn về cách thức hoạt động của mô hình CNN.

4.2 HẠN CHẾ

Do sự hạn chế về nguồn lực và thời gian, chúng tôi chọn tập trung vào nghiên cứu lý thuyết của "Explanable Artifical Intelligence". Về phần thực nghiệm trên tập dữ liệu về hình ảnh đeo khẩu trang do giới hạn về cơ sở vật chất, không có một cấu hình máy đủ tốt nên nhóm chúng tôi chỉ đủ để thử nghiệm trên một vài siêu tham số. Giới hạn kiến thức toán học cũng là một yếu tố cản trở việc nghiên cứu của chúng tôi.

4.3 HƯỚNG PHÁT TRIỂN

Lý thuyết được trình bày trong bào cáo có thể được áp dụng để giải quyết các bài toán phân tích trong doanh nghiệp như: giải thích được những dự đoán đưa ra là đúng, qua đó giúp doanh nghiệp đưa ra các giải pháp phù hợp

Báo cáo cũng mở ra nhiều hướng nghiên cứu trong tương lai như:

- Nghiên cứu sâu về nhiều phương pháp cho một mô hình học máy.
- Nghiên cứu về các thuật toán deep learning khác và có thể diễn giải được các thuật toán đó.

Về phần thực nghiệm, ta hoàn toàn có thể mở rộng để áp dụng tập dữ liệu khác nhau vào xây dựng mô hình đoán có tính thực tiễn, thử nghiệm nhiều thuật toán khác hơn, thay đổi nhiều bộ siêu tham số hơn để cải thiện độ chính xác của mô hình.

TÀI LIỆU THAM KHẢO

[1] M. I. Jordan, T. M. Mitchell , "Machine learning: Trends, perspectives, and prospects," 2015.

[2] Yann LeCun, Yoshua Bengio, Geoffrey Hinton , "Deep learning," 2015.

[3] Gunning D., Aha D., "DARPA's Explainable Artificial Intelligence (XAI) Program," 2019.

[4] Fatima Hussain, Rasheed Hussain, SMIEEE, and Ekram Hossain, FIEEE, "Explainable Artificial Intelligence (XAI): An Engineering Perspective," 2021.

[5] Francesco Ventura, Tania Cerquitelli, "What's in the box? Explaining the black-box model through an evaluation of its interpretable features," 2019.

[6] Miao Liu, Amit Dhurandhar,Ronny Luss, "Local Explanations for Reinforcement Learning," 2022.

[7] Carlos Guestrin,Sameer Singh,Marco Tulio Ribeiro, "“Why Should I Trust You?”Explaining the Predictions of Any Classifier," 2016.

[8] G. Gibbons, "What is the Shape of a Black Hole?," 2012.

[9] Mayukha Pa, P. Sai Ram Aditya, "Local Interpretable Model Agnostic Shap Explanations for machine learning," 2022.

[10] S.-I. L. Ian Covert, "Improving KernelSHAP: Practical Shapley Value Estimation via Linear Regression," 2021.

[11] Gabriel Laberge, Yann Pequignot, "Understanding Interventional TreeSHAP : How and Why it Works," 2022.

[12] Gianluigi Lopardo, Damien Garreau, Frederic Precioso, "A Sea of Words: An In-Depth Analysis of Anchors for Text Data," 2022.

[13] Alex Goldstein, Adam Kapelner, Justin Bleich, Emil Pitkin, "Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation," 2013.

[14] Christoph Molnar, Timo Freiesleben, Gunnar König, Giuseppe Casalicchio, Marvin N. Wright, Bernd Bischl, "Relating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process," 2021.

[15] Seyedehzahra Khoshmanesh, Tuba Yavuz, Robyn R. Lutz, "Learning Feature Interactions With and Without Specifications," 2021.

[16] WikiPedia, Convolutional neural network, 2022.

[17] Amidi, Afshine Amidi và Shervine, Convolutional Neural Networks cheatsheet.

[18] WikiPedia, Phân vùng ảnh, 2022.

[19] J. Rambhia, SLIC based Superpixel Segmentation, 2013.

[20] WikiPedia, Cosine similarity, 2022.

[21] D. Duvenaud, The Kernel Cookbook: Advice on Covariance functions.

[22] M. I. Jordan, T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, 2015.

[23] C. Molnar, Interpretable Machine Learning, 2022.

[24] Yann LeCun , Yoshua Bengio, Geoffrey Hinton , "Deep learning," p. 521, 2015.

[25] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "“Why Should I Trust You?” Explaining the Predictions of Any Classifier," 2016 .

[26] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use," 2019.

[27] Scott M. Lundberg, Su-In Lee, "A Unified Approach to Interpreting Model," 2017.

[28] P. Sai Ram Aditya, Mayukha Pal, "Local Interpretable Model Agnostic Shap Explanations for machine learning," 2022.