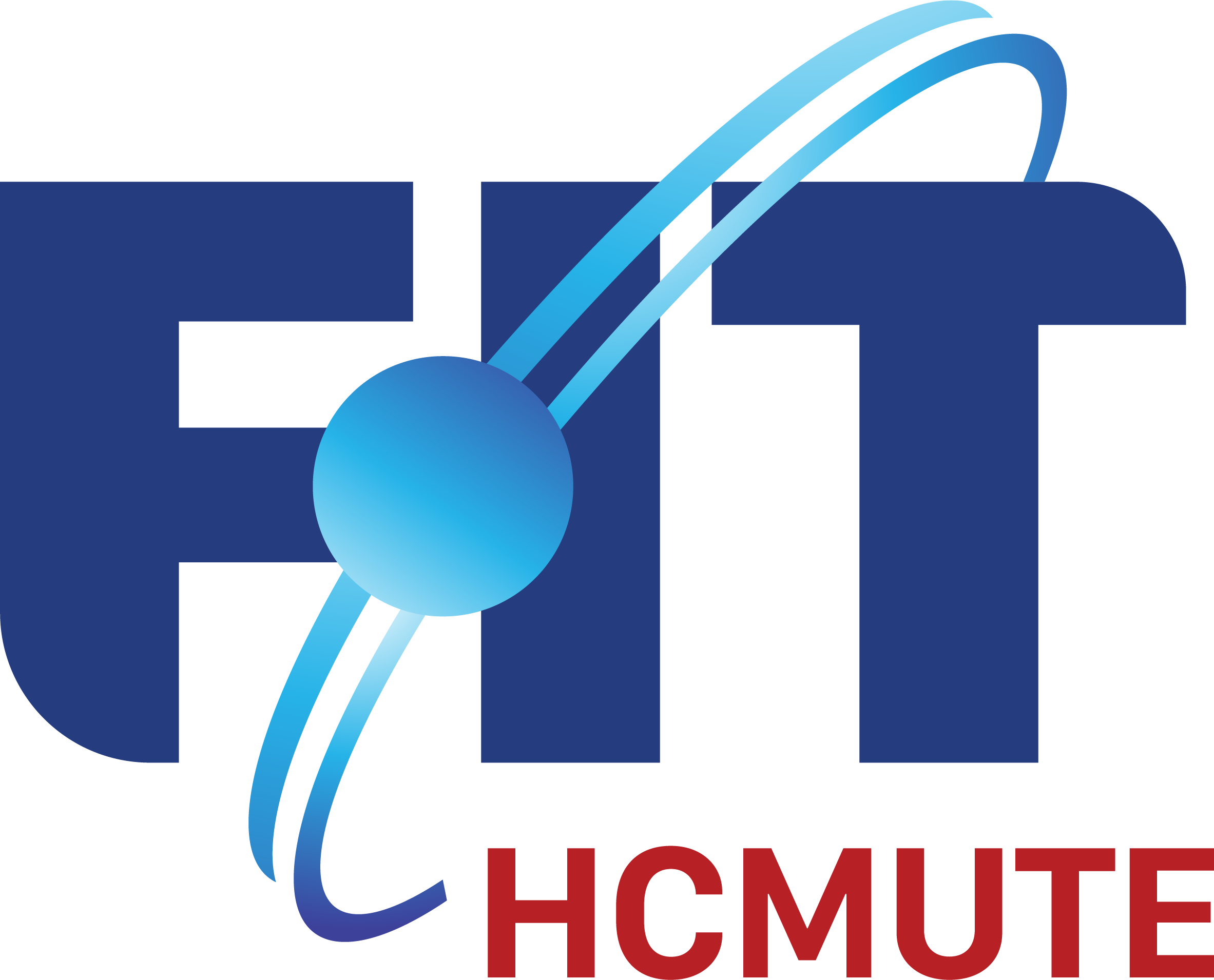
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KỸ THUẬT DỮ LIỆU**



Đề Tài:

**LẠI HỮU TRÁC : 19133059 NGUYỄN DUY PHƯỚC : 19133003**

**TÌM HIỂU VỀ EXPLAINABLE AI VÀ ỨNG DỤNG**

KHÓA LUẬN TỐT NGHIỆP KỸ SƯ KTDL

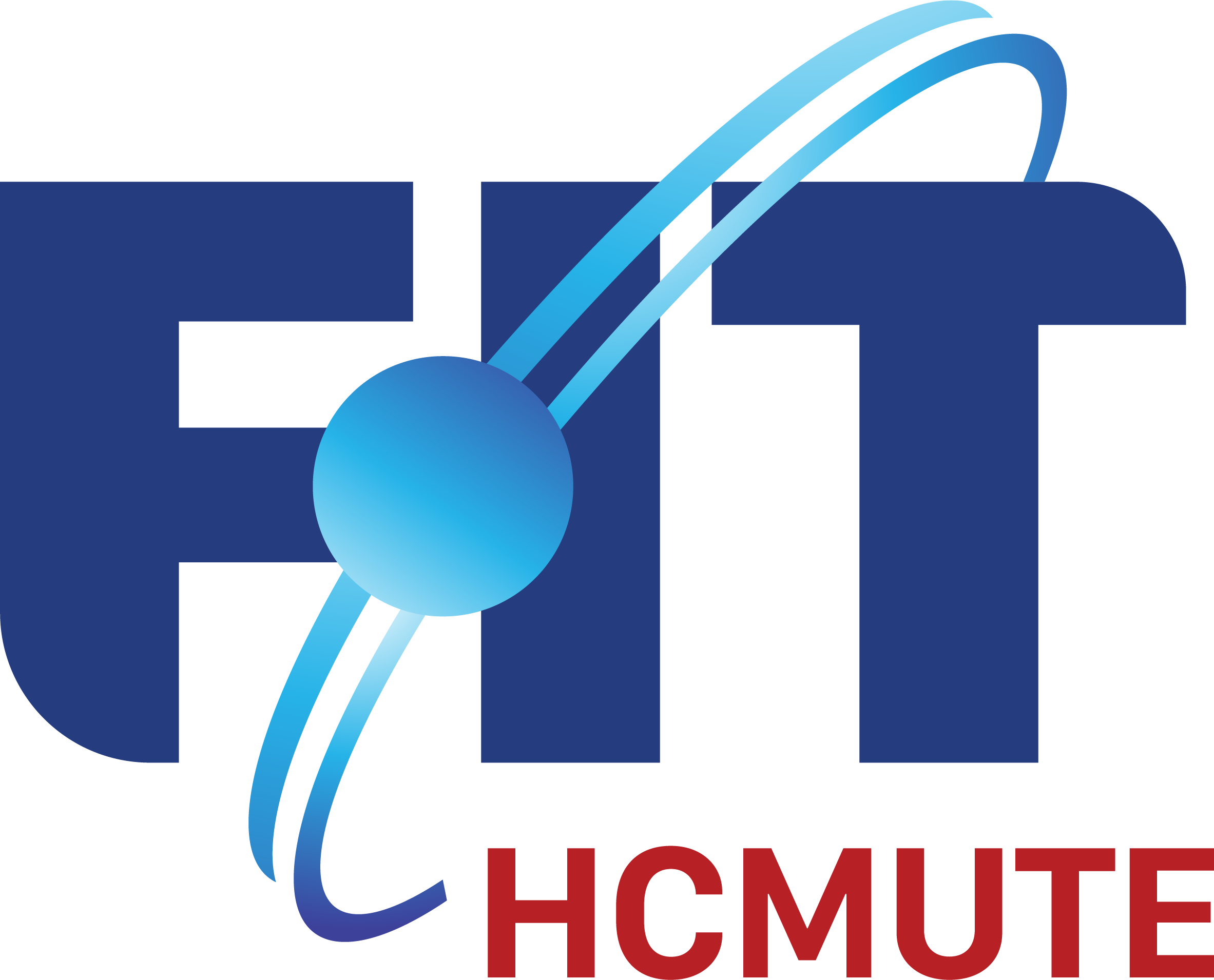
**GIÁO VIÊN HƯỚNG DẪN**

**ThS. QUÁCH ĐÌNH HOÀNG**

Hồ Chí Minh, 2023

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KỸ THUẬT DỮ LIỆU**



Đề Tài:

**LẠI HỮU TRÁC : 19133059**

**NGUYỄN DUY PHƯỚC : 19133003**

**TÌM HIỂU VỀ EXPLAINABLE AI VÀ ỨNG DỤNG**

KHÓA LUẬN TỐT NGHIỆP KỸ SƯ KTDL

**GIÁO VIÊN HƯỚNG DẪN**

**ThS. QUÁCH ĐÌNH HOÀNG**

Hồ Chí Minh, 2023

**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 khoá luận tốt nghiệp – 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*

|  |  |
| --- | --- |
| Trường ĐH Sư Phạm Kỹ Thuật TP.HCM  Khoa : CNTT |  |

**ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP**

Họ và Tên SV thực hiện 1 : Mã Số SV :

Họ và Tên SV thực hiện 2 : Mã Số SV :

Thời gian làm luận văn : từ : Đến :

Chuyên ngành :

Tên luận văn :

GV hướng dẫn :

**Nhiệm Vụ Của Luận Văn :**

1. Tìm hiểu về cơ sở lý thuyết Explainable AI
2. Tìm hiểu bài toán dựa trên cơ sở lý thuyết đã tiềm hiểu
3. Thực hiện ứng dụng trên cơ sở lý thuyết đã tìm hiểu để trực quan hóa dữ liệu.

# KẾ HOẠCH THỰC HIỆN

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thời gian | Nội dung công việc | Ghi chú |
| 1 | 14/2 - 21/2 | Lựa chọn và xác định đề tài tiểu luận chuyên ngành |  |
| 2 | 21/2 – 28/2 | Tìm hiểu sơ lược về đề tài |  |
| 3 | 1/3-8/3 | Tìm hiểu sơ lược về XAI |  |
| 4 | 9/3-16/3 | Tìm hiểu về LIME |  |
| 5 | 17/3-23/3 | Tìm hiểu về SHAP |  |
| 6 | 24/3-1/4 | Tìm hiểu về Anchors |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 7 | 2/4-9/4 | Tìm hiểu về phương pháp ICE |  |
| 8 | 10/4-17/4 | Tìm hiểu phương pháp Feature Interaction, Partial Dependence Plot (PDP) |  |
| 9 | 18/4-25/4 | Tìm hiểu về các công trình nổi trội cho XAI |  |
| 10 | 26/4-15/5 | Thực hiện tìm hiểu, xây dựng mô hình demo, hoàn thiện  báo cáo |  |
| 11 | 16/5-30/5 | Thực hiện xây dựng  mô hình demo, hoàn thiện báo cáo |  |
| 12 | 1/6 – 7/6 | Thực hiện xây dựng  mô hình demo, hoàn thiện báo cáo |  |
| 13 | 7/6 – 30/6 | Tiếp tục làm phần demo của bài toán và viết báo cáo phần ứng dụng |  |
| 14 | 1/7-9/7 | Hoàn thiện, rà soát báo cáo và hoàn chỉnh phần ứng dụng. |  |

# MỤC LỤC

[KẾ HOẠCH THỰC HIỆN 7](#_Toc139895229)

[MỤC LỤC 9](#_Toc139895230)

[DANH SÁCH HÌNH VẼ 13](#_Toc139895231)

[DANH MỤC TỪ VIẾT TẮT 16](#_Toc139895232)

[CHƯƠNG 1: MỞ ĐẦU 17](#_Toc139895233)

[**1.1 TÍNH CẤP THIẾT CỦA ĐỀ TÀI** 17](#_Toc139895234)

[**1.2 MỤC TIÊU VÀ NHIỆM VỤ NGHIÊN CỨU** 17](#_Toc139895235)

[**1.3 CÁCH TIẾP CẬN VÀ PHƯƠNG PHÁP NGHIÊN CỨU** 18](#_Toc139895236)

[**1.4 KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC** 18](#_Toc139895237)

[**1.5 BỐ CỤC LUẬN VĂN** 18](#_Toc139895238)

[CHƯƠNG 2: TỔNG QUAN VỀ XAI VÀ CÁC PHƯƠNG PHÁP TRONG XAI 20](#_Toc139895239)

[**2.1 TỔNG QUAN VỀ EXPLAINABLE AI** 20](#_Toc139895240)

[2.1.1 Định nghĩa về Explainable AI 20](#_Toc139895241)

[2.1.2 Lợi ích của Explainable AI 21](#_Toc139895242)

[2.1.3 Explainable AI được sử dụng ở đâu? 23](#_Toc139895243)

[2.1.4 Đánh giá về lời giải thích 24](#_Toc139895244)

[**2.2 TỔNG QUAN VỀ CÁC PHƯƠNG PHÁP TRONG EXPLAINABLE AI** 25](#_Toc139895245)

[2.2.1 Local Explanations 25](#_Toc139895246)

[*2.2.1.1 LIME* 25](#_Toc139895247)

[*2.2.1.2.1 Giới thiệu* 26](#_Toc139895248)

[*2.2.1.2.2* *Ý tưởng* 26](#_Toc139895249)

[*2.2.1.2 SHAP* 27](#_Toc139895250)

[*2.2.1.2.1 Giới thiệu* 28](#_Toc139895251)

[**2.2.1.2.2 KernelSHAP** 29](#_Toc139895252)

[*2.2.1.2.3 TreeSHAP* 30](#_Toc139895253)

[*2.2.1.2.4 Mức quan trọng của đặc trưng SHAP (SHAP feature importance)* 31](#_Toc139895254)

[*2.2.1.3 Anchor* 33](#_Toc139895255)

[*2.2.1.3.1 Giới thiệu* 33](#_Toc139895256)

[*2.2.1.3.2* *Tìm anchors* 34](#_Toc139895257)

[*2.2.1.3.3* *Độ phức tạp và thời gian chạy (Runtime)* 37](#_Toc139895258)

[*2.2.1.4 Individual Conditional Expectation (ICE)* 38](#_Toc139895259)

[*2.2.1.4.1 Giới thiệu* 38](#_Toc139895260)

[*2.2.1.4.2* *Tính toán các giá trị cho đường cong ICE* 38](#_Toc139895261)

[*2.2.1.4.3* *Phác hoạ ICE đạo hàm (Derivative ICE Plot)* 41](#_Toc139895262)

[2.2.2 Global Explanations 42](#_Toc139895263)

[*2.2.2.1 Partial Dependence Plot (PDP)* 42](#_Toc139895264)

[*2.2.1.1* *.1 Giới thiệu* 42](#_Toc139895265)

[*2.2.2.1.2* *Tầm quan trọng của đặc trưng dựa trên PDP* 43](#_Toc139895266)

[*2.2.2.2* *Feature Inter-action* 44](#_Toc139895267)

[*2.2.2.2.1 Giới thiệu* 45](#_Toc139895268)

[*2.2.2.2.2* *Lý thuyết: thống kê-H của Friedman* 46](#_Toc139895269)

[CHƯƠNG 3: TỔNG QUAN VỀ NLP VÀ BÀI TOÁN SENTIMENT ANALYSIS 49](#_Toc139895270)

[**3.1 Tổng quan về xử lý ngôn ngữ tự nhiên (NLP)** 49](#_Toc139895271)

[3.1.1 Định nghĩa về NLP 49](#_Toc139895272)

[3.1.2 Những lợi ích của NLP 49](#_Toc139895273)

[**3.2 Tổng quan về Sentiment Analysis** 50](#_Toc139895274)

[3.2.1 Tổng quan về Sentiment Analysis 50](#_Toc139895275)

[3.2.2 Tại sao Sentiment Analysis lại quan trọng 52](#_Toc139895276)

[**3.3 Các kĩ thuật world embedding** 53](#_Toc139895277)

[**3.3.1 Định nghĩa** 53](#_Toc139895278)

[**3.3.2 Các loại word embeding** 53](#_Toc139895279)

[*3.3.2.1 Bag of words* 53](#_Toc139895280)

[*3.3.2.2 TF-IDF (term frequency – inverse document frequency)* 54](#_Toc139895281)

[*3.3.2.3 Glove* 55](#_Toc139895282)

[CHƯƠNG 4*.*MỘT SỐ THUẬT TOÁN MACHINE LEARNING VÀ DEEP LEARNING 57](#_Toc139895283)

[4.1 Logistic Regression 57](#_Toc139895284)

[4.1.1 Giới thiệu 57](#_Toc139895285)

[4.1.2 Mô hình Logistic Regression 57](#_Toc139895286)

[4.2 SVM 58](#_Toc139895287)

[**4.2.1 Giới thiệu** 58](#_Toc139895288)

[**4.2.2 Ý tưởng phương pháp** 59](#_Toc139895289)

[**4.2.3 Nội dung phương pháp** 59](#_Toc139895290)

[**4.2.4 Bài toán phân 2 lớp với SVM** 60](#_Toc139895291)

[**4.2.5 Bài toán nhiều lớp với SVM** 61](#_Toc139895292)

[4.3 Tổng quan về Deep Learning 61](#_Toc139895293)

[4.3.1 Giới thiệu 61](#_Toc139895294)

[4.3.2 Cách thức hoạt động 62](#_Toc139895295)

[4.4 Một số thuật toán trong Deep Learning 62](#_Toc139895296)

[4.4.1 RNN 62](#_Toc139895297)

[*4.4.1.1 Giới thiệu* 62](#_Toc139895298)

[*4.4.1.2 Kiến trúc* 63](#_Toc139895299)

[*4.4.1.3 Các hàm kích hoạt thường dùng trong RNN* 64](#_Toc139895300)

[4.4.2 LSTM 64](#_Toc139895301)

[*4.4.2.1 Giới thiệu* 64](#_Toc139895302)

[*4.4.2.2 Kiến trúc và cách hoạt động của LSTM* 65](#_Toc139895303)

[*4.4.2.3 Tổng quan về Bi - LSTM* 66](#_Toc139895304)

[4.4.3 CNN 67](#_Toc139895305)

[*4.4.3.1 Giới thiệu* 67](#_Toc139895306)

[*4.4.3.2 Kiến trúc tổng quan của mạng CNN* 69](#_Toc139895307)

[*4.4.3.3 Mô hình kết hợp của CNN-LSTM* 73](#_Toc139895308)

[CHƯƠNG 5: THỰC NGHIỆM 74](#_Toc139895309)

[**5.1 BÀI TOÁN** 74](#_Toc139895310)

[**5.2 DỮ LIỆU** 75](#_Toc139895311)

[**5.3 PHƯƠNG PHÁP VÀ KẾT QUẢ** 75](#_Toc139895312)

[CHƯƠNG 6: KẾT LUẬN 55](#_Toc139895313)

[**6.1 KẾT QUẢ ĐẠT ĐƯỢC** 55](#_Toc139895314)

[6.1.1 Ý nghĩa khoa học 55](#_Toc139895315)

[6.1.2 Ý nghĩa thực tiễn 55](#_Toc139895316)

[**6.2 HẠN CHẾ** 56](#_Toc139895317)

[**6.3 HƯỚNG PHÁT TRIỂN** 56](#_Toc139895318)

[TÀI LIỆU THAM KHẢO 57](#_Toc139895319)

# DANH SÁCH HÌNH VẼ

[Hình 1: Công thức tính LIME 26](#_Toc135930986)

[Hình 2: Công thức tính SHAP 28](#_Toc135930987)

[Hình 3: Công thức đơn giản của SHAP 28](#_Toc135930988)

[Hình 4: Ví dụ minh hoạ về Kernel SHAP 30](#_Toc135930989)

[Hình 5: Công thức tính giá trị tuyệt đối Shaley 32](#_Toc135930990)

[Hình 6: Biểu đồ cho thấy mức độ đặc trưng SHAP của người ưng thư cổ tử cung 32](#_Toc135930991)

[Hình 7: Công thức tính anchors 34](#_Toc135930992)

[Hình 8: Công thức tính độ thống kê tin cậy 34](#_Toc135930993)

[Hình 9: Công thức tính độ phủ 35](#_Toc135930994)

[Hình 10: Công thức tính tối đa độ phủ 35](#_Toc135930995)

[Hình 11: Các thành phần của thuật toán anchors và mối quan hệ tương quan 37](#_Toc135930996)

[Hình 12: Công thức tính độ phức tạp của Anchors 37](#_Toc135930997)

[Hình 13: Ví dụ tìm đặc trưng trong ICE 39](#_Toc135930998)

[Hình 14: Ví dụ tìm giá trị duy nhất trong ICE 39](#_Toc135930999)

[Hình 15: Ví dụ về sửa các giá trị đặc trưng khác 39](#_Toc135931000)

[Hình 16: Tính toán dự đoán 40](#_Toc135931001)

[Hình 17: Đường cong minh hoạ trong ICE 40](#_Toc135931002)

[Hình 18:Hàm dự đoán ICE đạo hàm 41](#_Toc135931003)

[Hình 19 : Công thức tính hàm phụ thuộc riêng 42](#_Toc135931004)

[Hình 20: Phương pháp tính Monte Carlo 42](#_Toc135931005)

[Hình 21: Cônh thức tính độ lệch của từng giá trị 43](#_Toc135931006)

[Hình 22: Công thức tính độ lệch cho nhiều giá trị 43](#_Toc135931007)

[Hình 23: Ví dụ về Feature Inter-action 44](#_Toc135931008)

[Hình 24: Ví dụ về Feature Inter-action 45](#_Toc135931009)

[Hình 25: Công thức tính hàm phụ thuộc riêng theo Friedman 45](#_Toc135931010)

[Hình 26: Tổng hàm phụ thuộc riêng của nhiều giá trị 46](#_Toc135931011)

[Hình 27: Công thức tính tương tác giữa hai đặc trưng j và k 46](#_Toc135931012)

[Hình 28: Công thức tính tương tác giữa j và các đặc trưng còn lại 47](#_Toc135931013)

[Hình 29 : Minh họa các cấp độ phân tích tình cảm 50](#_Toc135931014)

[Hình 30: Kiến trúc của một mạng RNN [17] 54](#_Toc135931015)

[Hình 31: Các hàm kích hoạt thường dùng trong RNN [18] 55](#_Toc135931016)

[Hình 32: Kiến trúc của một node trong mạng LSTM [18] 57](#_Toc135931017)

[Hình 33: Kiến trúc của mạng Bi -LSTM [19] 59](#_Toc135931018)

[Hình 34: Cấu trúc của mạng CNN truyền thống [18] 60](#_Toc135931019)

[Hình 35: Cấu trúc mô hình CNN gồm 3 lớp cho bài toán dự đoán 11 nhãn [20] 60](#_Toc135931020)

[Hình 36: Kiến trúc mạng tích chập cho bài toán phân loại ảnh [21] 61](#_Toc135931021)

[Hình 37: Mô tả tích chập ảnh với bộ lọc (kenel ) [22] 62](#_Toc135931022)

[Hình 38: Các loại hàm ReLU và chức năng [18] 63](#_Toc135931023)

[Hình 39: Max pooling va Average pooling [18] 64](#_Toc135931024)

[Hình 40: Mô phỏng về lớp fully connected [18] 65](#_Toc135931025)

[Hình 41: Kiến trúc mô hình CNN-LSTM 66](#_Toc135931026)

[Hình 42: Đồ thị hàm sigmod 67](#_Toc135931027)

[Hình 43: Siêu phẳng chia dữ liệu thành 2 lớp + và - với khảng cách biên lớn nhất. Các điểm gần nhất (điểm được khoanh tròn) là các Support Vecotr. 69](#_Toc135931028)

[Hình 44: Minh họa bài toán 2 phân lớp bằng phương pháp SVM 70](#_Toc135931029)

[Hình 45: Word embedding từ "engine" 71](#_Toc135931030)

[Hình 46: Biểu diễn từ dưới dạng vector 73](#_Toc135931031)

[Hình 47: Cấu trúc mô hình Logistic Regression 77](#_Toc135931032)

[Hình 48: Kết quả của mô hình Logistic Regression 77](#_Toc135931033)

[Hình 49: Cấu trúc mô hình SVM 78](#_Toc135931034)

[Hình 50: Cấu trúc mô hình CNN-LSTM 79](#_Toc135931035)

[Hình 51: Quá trình huấn luyện mô hình CNN-LSTM với 5 epoch 79](#_Toc135931036)

[Hình 52: Độ chính xác của mô hình CNN-LSTM 80](#_Toc135931037)

[Hình 53: Kiến trúc mô hình BiLSTM 80](#_Toc135931038)

[Hình 54: Các tham số huấn luyện mô hình BiLSTM 81](#_Toc135931039)

[Hình 55: Kết quả của mô hình trên tập dữ liệu test 81](#_Toc135931040)

[Hình 56: Giao diện giải thích trên Streamlit 82](#_Toc135931041)

[Hình 57: Các thuật toán trong select box 82](#_Toc135931042)

[Hình 58: Lựa chọn phương pháp giải thích 83](#_Toc135931043)

[Hình 59: Kết quả giải thích với phương pháp LIME 83](#_Toc135931044)

[Hình 60: Kết quả giải thích phương pháp SHAP 84](#_Toc135931045)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Ký hiệu từ viết tắt** | **Từ đầy đủ** |
| 1 | AI | Artificial intelligence |
| 1 | XAI | Explainable artificial intelligence |
| 2 | ML | Machine learning |
| 3 | DL | Deep learning |
| 4 | LIME | Local Interpretable Model-Agnostic Explanations |
| 5 | SHAP | SHapley Additive exPlanations |
| 6 | RNN | Recurrent Neural Network |
| 7 | LSTM | Long short term memory |
| 8 | SVM | Support vector machine |
| 9 | NLP | Natural language processing |
| 10 | Bi - LSTM | Bi-directional long short term memory |
| 11 | CNTT | Công nghệ thông tin |
| 12 | BoW | Bag of Words |

# 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. Khoảng năm 2012, phương pháp học sâu (DL) 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 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 một 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 xử lý và phân tích khía cạnh của quan điểm, ý kiến, tình cảm của đối tượng nghiên cứu cụ thể và sử dụng phương pháp LIME, SHAP để giải thích cho kết quả dự đoán 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 chức như sau:

1. Mở đầu
   1. Tính cấp thiết của đề tài
   2. Mục tiêu và nhiệm vụ nghiên cứu
   3. Cách tiếp cận và phương pháp nghiên cứu
   4. Kết quả dự đoán được
   5. Bố cục luận văn
2. Nội dung
   1. Tổng quan về XAI
   2. Tổng quan về các phương pháp trong XAI

2.3 Tổng quan về xử lý ngôn ngữ tự nhiên (NLP)

2.4 Tổng quan về Sentiment Analysis

2.5 Tổng quan về Deep Learning và một số thuật toán trong Deep Learning và Machine Learning

3. Thực nghiệm

3.1 Bài 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: TỔNG QUAN VỀ XAI VÀ CÁC PHƯƠNG PHÁP TRONG XAI

## **2.1 TỔNG QUAN VỀ EXPLAINABLE AI**

### 2.1.1 Định nghĩa về Explainable AI

Trước hết, để hiểu Explainable AI (XAI) 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)

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.

Để thúc đẩy phát triển XAI, NIST (Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ) đã công bố bốn nguyên tắc:

* Explanation: Các hệ thống cung cấp bằng chứng hoặc (các) lý do đi kèm cho tất cả các kết quả đầu ra
* Meaningful: Các hệ thống cung cấp các giải thích dễ hiểu đối với người dùng cá nhân.
* Explanation accuracy: Lời giải thích phản ánh chính xác quy trình tạo đầu ra của hệ thống
* Knowledge limits: Hệ thống chỉ hoạt động trong các điều kiện mà nó được thiết kế hoặc khi đầu ra của nó đạt đủ mức độ tin cậy

2.1.2 Tầm quan trọng của XAI

XAI là một thành phần quan trọng để phát triển, chiến thắng và duy trì niềm tin vào các hệ thống tự động. Nếu không có sự tin tưởng, AI và cụ thể là AI cho hoạt động CNTT (AIOps) sẽ không được chấp nhận hoàn toàn, khiến quy mô và độ phức tạp của các hệ thống hiện đại không thể vượt xa những gì có thể đạt được so với các hoạt động thủ công và tự động hóa truyền thống.

Khi niềm tin được thiết lập, thông lệ “AI washing”: ngụ ý rằng một sản phẩm hoặc dịch vụ do AI điều khiển khi vai trò của AI ít hoặc vắng mặt trở nên rõ ràng, giúp cả người thực hiện và khách hàng thực hiện thẩm định AI của họ. Khi giao nhiệm vụ cho bất kỳ hệ thống nào tìm câu trả lời hoặc đưa ra quyết định, đặc biệt là những hệ thống có tác động trong thế giới thực, chúng ta bắt buộc phải giải thích cách một hệ thống đưa ra quyết định, cách nó ảnh hưởng đến kết quả hoặc tại sao các hành động được coi là cần thiết.

### 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, … 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.

### 2.1.4 Đánh giá về lời giải thích

Hiện nay vẫn chưa có sự đồng thuận thực sự về đánh giá khả năng diễn giải trong học máy. Cũng không rõ làm thế nào để đo lường nó. Nhưng có một số nghiên cứu ban đầu về vấn đề này và nỗ lực xây dựng một số phương pháp tiếp cận để đánh giá. Năm 2017, Doshi-Velez and Kim đã đề xuất ba cấp độ chính để đánh giá khả năng diễn giải:

* Đánh giá mức ứng dụng (tác vụ thực tế) : Đưa phần giải thích vào sản phẩm và để người dùng cuối kiểm tra. Hãy tưởng tượng phần mềm phát hiện gãy xương với một thành phần học máy định vị và đánh dấu vết nứt trong tia X. Ở cấp độ ứng dụng, các bác sĩ X quang sẽ trực tiếp kiểm tra phần mềm phát hiện vết nứt để đánh giá mô hình. Điều này đòi hỏi thiết lập thử nghiệm tốt và hiểu biết về cách đánh giá chất lượng.
* Đánh giá mức con người (nhiệm vụ đơn giản): Đây là một đánh giá cấp ứng dụng đơn giản hóa. Sự khác biệt là những thí nghiệm này không được thực hiện với các chuyên gia có kiến ​​thức hoặc kỹ năng đặc biệt trong một lĩnh vực cụ thể, mà với những người không chuyên. Điều này làm cho các thử nghiệm rẻ hơn (đặc biệt nếu các chuyên gia trong lĩnh vực này là bác sĩ X quang) và dễ dàng tìm được nhiều người thử nghiệm hơn. Một ví dụ sẽ là hiển thị cho người dùng các giải thích khác nhau và người dùng sẽ chọn giải thích tốt nhất.
* Đánh giá mức chức năng (tác vụ ủy quyền): Đánh giá mức chức năng không cần đến con người. Điều này hoạt động tốt nhất khi lớp mô hình được sử dụng đã được đánh giá bởi người khác trong đánh giá cấp độ con người. Ví dụ, có thể biết rằng người dùng cuối hiểu cây quyết định. Trong trường hợp này, một đại diện cho chất lượng giải thích có thể là độ sâu của cây. Những cây ngắn hơn sẽ có điểm giải thích tốt hơn. Sẽ hợp lý hơn nếu thêm ràng buộc rằng hiệu suất dự đoán của cây vẫn tốt và không giảm quá nhiều so với cây có độ sâu lớn hơn.

## **2.2 TỔNG QUAN VỀ CÁC PHƯƠNG PHÁP TRONG EXPLAINABLE AI**

### 2.2.1 Local Explanations

Local Explanations [1] 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*

LIME (Local interpretable model-agnostic explanations [2] là một phương pháp 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.

* + - * 1. *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

Description automatically generated

Hình : 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 hàm 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 πx 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.

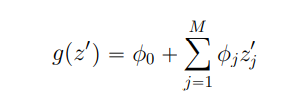
- Giải thích dự đoán bằng cách diễn giải mô hình cục bộ.

#### *2.2.1.2 SHAP*

SHAP (Shapley Additive Explanations) [1] [3] [4] 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 nguời 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:



Hình : Công thức tính SHAP

Với g là mô hình giải thích, z’ là vector liên minh (coalition vector), M là kích cỡ liên minh tối đa và 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:

A close-up of a logo

Description automatically generated with low confidence

Hình : Công thức đơn giản của SHAP

##### **2.2.1.2.2** **KernelSHAP**

KernelSHAP [5] 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  (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 bằng mô hình phức tạp trước khi dự đoán bằng mô hình tuyến tính, ở đây chúng ta áp dụng mô hình f: với

(c) Tính trọng số (weight) cho mỗi điểm dữ liệu 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 , 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  với  . Hàm 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 xạ từ các vector liên minh tới các giá trị đặc trưng:

Text, letter

Description automatically generated

Hình : Ví dụ minh hoạ về Kernel SHAP

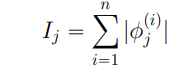
Từ hình 4 ta thấy hàm ánh xạ một vector liên minh tới một mẫu dữ liệu hợp lệ. Cho đặc trưng có mặt (1), ánh xạ từ các giá trị đặc trưng của x. Cho các đặc trưng vắng mặt (0), ánh xạ từ các giá trị của một mẫu dữ liệu lấy mẫu ngẫu nhiên.

##### *2.2.1.2.3* *TreeSHAP*

Lundberg et. al (2018) đề xuất TreeSHAP [6], một biến thể của SHAP cho các mô hình học máy dựa trên cây như cây quyết định, Random Forest 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)  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. So với KernelSHAP, nó làm giảm độ phức tạp tính toán từ thành , 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 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 đặt điều kiện tất cả các đặc trưng – nếu S là tập tất cả các đặc trưng thì dự đoán từ một nút (node) có mẫu dữ liệu x nằm trong sẽ 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 cộng trọng số (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 lấy trung bình các dự đoán có trọng số bằng kích thước các nút (tức số lượng mẫu huấn luyện trong nút đó). Trung bình của các nút đầu cuối còn lại, được tính trọng số 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 S có thể xuống cây cùng một lúc. Cho mỗi nút quyết định (decision node) ta phải theo dõi chung số lượng các tập con. Điều này phụ thuộc vào các tập con trong nút cha (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:



Hình : 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 Random Forest trước đó để dự đoán ung thư cổ tử cung.

Chart

Description automatically generated

Hình : Biểu đồ cho thấy mức độ đặc trưng SHAP của người ưng 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 [1] [7] 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. 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 anchors. 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:

Text

Description automatically generated

Hình : 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.

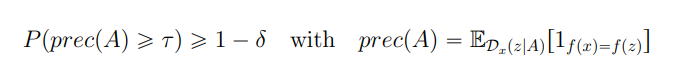
• f kí hiệu mô hình phân loại được giải thích (ví dụ, mô hình mạng nơ-ron nhân tạo). Nó có thể được sử dụng để dự đoán một dán nhãn cho x và các xáo trộn của nó.

• thể hiện phân phối các điểm lân cận trong x phù hợp với A.

• 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.

* + - * 1. *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 giá 1 f(x) = f(z) cho mọi 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 δ 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:



Hình : 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:

Text

Description automatically generated

Hình : Công thức tính độ phủ

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

Text

Description automatically generated

Hình : 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 anchors 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 “anchors”). 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.

Diagram

Description automatically generated

Hình : 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.

* + - * 1. *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:

Text

Description automatically generated with medium confidence

Hình : 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 [8] 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) [9] . 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.

* + - * 1. *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

Table

Description automatically generated

Hình : Ví dụ tìm đặc trưng trong ICE

* Tìm các giá trị duy nhất của đặc trưng

A picture containing graphical user interface

Description automatically generated

Hình : 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.

Table

Description automatically generated

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

* Đưa ra dự đoán cho từng kết hợp.

Table

Description automatically generated

Hình : Tính toán dự đoán

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

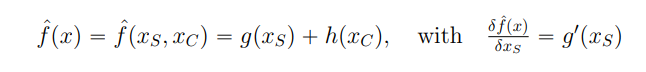
Chart, line chart

Description automatically generated

Hình : Đường cong minh hoạ trong ICE

* + - * 1. *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 xS và các đặc trưng khác xC, hàm dự đoán có thể biểu diễn như sau:



Hình :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)*

* + - 1. *.1 Giới thiệu*

Phác họa phụ thuộc riêng (partial dependence plot) (viết tắt là PDP hoặc phác họa PD) [1] [10] 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) [11]. 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:

A picture containing text, watch

Description automatically generated

Hình : Công thức tính hàm phụ thuộc riêng

Biến xS là những đặc trưng mà phác họa PD cần phải biểu diễn và xC là những đặc trưng khác được dùng trong mô hình học máy . 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 xS và xC 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) được xấp xỉ bằng cách tính trung bình trong dữ liệu huấn luyện, hay còn gọi là phương pháp Monte Carlo:

Text, schematic

Description automatically generated

Hình : 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, 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.

* + - * 1. *Tầm quan trọng của đặc trưng dựa trên PDP*

Greenwell và cộng sự. (2018) [12] 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:

A picture containing text, clock, screenshot

Description automatically generated

Hình : Cônh thức tính độ lệch của từng giá trị

Lưu ý rằng ở đây là K giá trị duy nhất của đặc trưng XS. Đối với các đặc trưng phân loại, chúng tôi có:

Text

Description automatically generated with medium confidence

Hình : 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.

* + - 1. *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. [13]

##### *2.2.2.2.1 Giới thiệu*

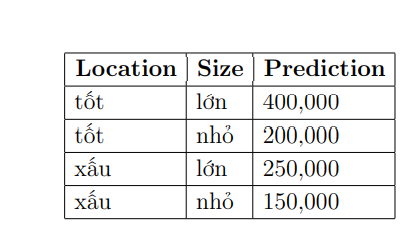
Nếu một mô hình dự đoán dựa trên hai đặc trưng, ta có thể phân tích dự đoán thành 4 phần: một giá trị hằng số, một cho đặc trưng thứ nhất, một cho đặc trưng thứ hai, một cho tương tác giữa hai đặc trưng. Tương tác giữa hai đặc trưng là sự thay đổi ở dự đoán xảy ra khi ta thay đổi đặc trưng sau khi xem 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:

Table

Description automatically generated

Hình : Ví dụ về Feature Inter-action

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



Hình : 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).

* + - * 1. *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):

Text

Description automatically generated

Hình : Công thức tính hàm phụ thuộc riêng theo Friedman

với là hàm phụ thuộc riêng hai chiều của cả hai đặc trưng, và 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:

Text

Description automatically generated

Hình : Tổng hàm phụ thuộc riêng của nhiều giá trị

với 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:

A picture containing Word

Description automatically generated

Hình : 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:

Text

Description automatically generated

Hình : 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 lần để tính thống kê-H hai phía (j với k) và 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: TỔNG QUAN VỀ NLP VÀ BÀI TOÁN SENTIMENT ANALYSIS

## **3.1 Tổng quan về xử lý ngôn ngữ tự nhiên (NLP)**

### 3.1.1 Định nghĩa về NLP

Xử lý ngôn ngữ tự nhiên là một nhánh nghiên cứu của trí tuệ nhân tạo, cho phép máy tính hiểu ngôn ngữ của con người, cho dù ngôn ngữ đó được viết, nói hay thậm chí là viết nguệch ngoạc. NLP được sử dụng trong nhiều loại sản phẩm và dịch vụ hàng ngày. Một số cách phổ biến nhất mà NLP được sử dụng là thông qua các trợ lý kỹ thuật số được kích hoạt bằng giọng nói trên điện thoại thông minh, các chương trình quét email được sử dụng để xác định thư rác và các ứng dụng dịch thuật giải mã ngôn ngữ nước ngoài. [14]

### 3.1.2 Những lợi ích của NLP

* Thực hiện phân tích quy mô lớn

Công nghệ NLP cho phép phân tích văn bản ở quy mô lớn trên tất cả các loại tài liệu, hệ thống nội bộ, email, dữ liệu mạng xã hội, đánh giá trực tuyến, v.v. Xử lý lượng dữ liệu khổng lồ chỉ trong vài giây hoặc vài phút, điều này sẽ mất vài ngày hoặc vài tuần để phân tích thủ công

* Phân tích khách quan và chính xác hơn

Khi thực hiện các nhiệm vụ lặp đi lặp lại (và thực sự là nhàm chán), như đọc và phân tích các câu trả lời khảo sát mở và các dữ liệu văn bản khác, con người dễ mắc sai lầm hoặc có thể có những thành kiến ​​cố hữu có thể làm sai lệch kết quả. Các công cụ do NLP cung cấp có thể được đào tạo theo ngôn ngữ và tiêu chí của doanh nghiệp bạn, thường chỉ trong một vài bước. Vì vậy, một khi bạn thiết lập và chạy chúng, chúng sẽ hoạt động chính xác hơn nhiều so với con người. Và bạn có thể điều chỉnh và tiếp tục đào tạo các mô hình của mình khi thị trường hoặc ngôn ngữ kinh doanh của bạn phát triển.

* Giảm chi phí vận hành

Các công cụ NLP hoạt động ở bất kỳ quy mô nào bạn cần, 24/7, trong thời gian thực. Bạn cần ít nhất một vài nhân viên làm việc toàn thời gian để thực hiện phân tích dữ liệu thủ công nhưng với các công cụ NLP SaaS, bạn có thể giữ nhân viên ở mức tối thiểu. Khi bạn kết nối các công cụ NLP với dữ liệu của mình, bạn sẽ có thể phân tích phản hồi của khách hàng khi đang di chuyển, vì vậy bạn sẽ biết ngay khi nào khách hàng gặp sự cố với sản phẩm hoặc dịch vụ của bạn.

* Cải thiện sự hài lòng của khách hàng

Các công cụ NLP cho phép bạn tự động phân tích và sắp xếp các yêu cầu dịch vụ khách hàng theo chủ đề, ý định, mức độ khẩn cấp, tình cảm, v.v. và chuyển chúng trực tiếp đến bộ phận hoặc nhân viên thích hợp, vì vậy bạn không bao giờ bỏ mặc khách hàng. Và việc thực hiện phân tích NLP trên các cuộc khảo sát về mức độ hài lòng của khách hàng có thể giúp bạn nhanh chóng khám phá mức độ hài lòng của khách hàng ở mọi giai đoạn trong hành trình của họ.

* Hiểu rõ hơn về thị trường

NLP đang có tác động rất lớn đến tiếp thị. Khi bạn sử dụng NLP để hiểu ngôn ngữ của cơ sở khách hàng của mình, bạn sẽ hiểu rõ hơn về phân khúc thị trường, được trang bị tốt hơn để nhắm mục tiêu trực tiếp đến khách hàng của mình và giảm tỷ lệ khách hàng rời bỏ.

* Xử lí khối lượng lớn dữ liệu dạng text (văn bản)

Máy móc ngày nay có thể phân tích nhiều dữ liệu dựa trên ngôn ngữ hơn con người mà không mệt mỏi và theo cách nhất quán, không thiên vị. Xem xét lượng dữ liệu phi cấu trúc đáng kinh ngạc được tạo ra hàng ngày, từ hồ sơ y tế đến phương tiện truyền thông xã hội, tự động hóa sẽ rất quan trọng để phân tích đầy đủ dữ liệu văn bản và giọng nói một cách hiệu quả. [15]

## **3.2 Tổng quan về Sentiment Analysis**

### 3.2.1 Tổng quan về Sentiment Analysis

Phân tích cảm xúc (sentiment analysis), còn được gọi là khai phá quan điểm hoặc AI cảm xúc, là việc sử dụng NLP, phân tích văn bản, ngôn ngữ học máy tính và sinh trắc học để xác định, trích xuất, định lượng và nghiên cứu một cách có hệ thống các trạng thái cảm xúc và thông tin chủ quan. Nó được áp dụng rộng rãi cho tiếng nói của các tài liệu khách hàng như đánh giá và phản hồi khảo sát, phương tiện truyền thông xã hội và trực tuyến cũng như tài liệu chăm sóc sức khỏe cho các ứng dụng từ tiếp thị đến dịch vụ khách hàng đến y học lâm sàng.

Nghiên cứu Sentiment Analysis được chia làm 3 mức độ :

* **Document Level** : Document level classification hoạt động tốt nhất khi tài liệu được viết bởi một người duy nhất và thể hiện một ý kiến ​​​​/cảm xúc về một thực thể duy nhất
* **Sentence Level** : nhiệm vụ ở cấp độ này tương tự như cấp độ document nhưng đi sâu vào từng câu và xác định xem mỗi câu thể hiện ý kiến, quan điểm tích cực, tiêu cực hay trung lập.
* **Aspect level**: Khác với cấp độ tài liệu và cấp độ câu, cấp độ khía cạnh phân loại cảm xúc xem xét cả cảm xúc và thông tin mục tiêu, vì cảm xúc luôn có mục tiêu. Như đã đề cập trước đó, mục tiêu thường là một thực thể hoặc một khía cạnh của thực thể. Đưa ra một câu và một khía cạnh mục tiêu, mức độ phân loại nhằm mục đích suy ra hướng cảm xúc của câu về khía cạnh mục tiêu. Ví dụ, trong câu "màn hình rất rõ nét nhưng thời lượng pin là quá ngắn." cảm xúc là tích cực nếu khía cạnh mục tiêu là "màn hình" nhưng tiêu cực nếu khía cạnh mục tiêu là "thời lượng pin".

A picture containing text, font, diagram, line

Description automatically generated

Hình : Minh họa các cấp độ phân tích tình cảm

Một số dạng phân tích tình cảm [16]:

* Fine-grained Sentiment Analysis : Mô hình phân tích cảm xúc này này giúp xác định được độ chính xác của các tính chất. Các tính chất chính được phân chia thành: rất tích cực, tích cực, trung tính, tiêu cực hoặc rất tiêu cực. Việc phân chia chi tiết như vậy rất phù hợp để đánh giá các cuộc trò chuyện. Đối với thang điểm đánh giá từ 1 đến 5, bạn có thể coi 1 là rất tiêu cực và 5 là rất tích cực. Đối với thang điểm từ 1 đến 10, bạn có thể coi 1-2 là rất tiêu cực và 9-10 là rất tích cực.
* Aspect-Based : Trong khi phân tích chi tiết xác định tính chất tổng thể cảm xúc của khách hàng trong cuộc trò chuyện, thì phân tích dựa trên khía cạnh sẽ đi sâu hơn, nhận dạng cụ thể từng khía cạnh trong lời nói của họ. Chẳng hạn khi khách hàng nói rằng “máy ảnh gặp khó khăn trong điều kiện ánh sáng nhân tạo”. Với phân tích dựa trên khía cạnh, chúng ta không chỉ đánh giá được đây là cảm xúc tiêu cực mà còn có thể xác định rằng người dùng đó đã nhận xét tiêu cực về đối tượng “máy ảnh”.
* Emotion Detection : cảm xúc ở đây có thể bao gồm các sắc thái tức giận, buồn bã, hạnh phúc, thất vọng, sợ hãi, lo lắng, hoảng sợ… Hệ thống phát hiện cảm xúc thường sử dụng từ vựng – một tập hợp các từ truyền tải những cảm xúc nhất định. Một số bộ phân loại nâng cao cũng sử dụng các thuật toán học máy mạnh mẽ.
* Intent Analysis: Việc xác định chính xác mục đích của người tiêu dùng có thể giúp công ty tiết kiệm thời gian, tiền bạc và công sức, bởi các doanh nghiệp có thể không tiếp tục chăm sóc những khách hàng không tiềm năng và chưa có kế hoạch mua hàng. Phân tích mục đích chính xác có thể giải quyết nhiều vấn đề cho doanh nghiệp như vậy. Phân tích mục đích giúp doanh nghiệp xác định mục đích của người tiêu dùng – cho dù khách hàng có ý định mua hàng hay chỉ đang lướt qua. Nếu khách hàng sẵn sàng mua hàng, bạn có thể theo dõi họ và nhắm mục tiêu họ bằng các quảng cáo. Nếu người tiêu dùng chưa sẵn sàng mua, chúng ta có thể tiết kiệm thời gian và nguồn lực bằng cách không quảng cáo cho họ.

### 3.2.2 Tại sao Sentiment Analysis lại quan trọng

Kể từ khi con người cởi mở hơn trong việc bày tỏ suy nghĩ và cảm xúc, phân tích cảm xúc nhanh chóng trở thành một công cụ thiết yếu để theo dõi và hiểu cảm xúc trong tất cả các loại dữ liệu. Tự động phân tích phản hồi của khách hàng, chẳng hạn như ý kiến ​​trong các câu trả lời khảo sát và các cuộc trò chuyện trên mạng xã hội, cho phép các thương hiệu tìm hiểu điều gì khiến khách hàng hài lòng hoặc thất vọng, để họ có thể điều chỉnh sản phẩm và dịch vụ đáp ứng nhu cầu của khách hàng. Ví dụ: sử dụng phân tích cảm xúc để tự động phân tích hơn 4.000 câu trả lời trong các cuộc khảo sát về mức độ hài lòng của khách hàng có thể giúp bạn khám phá lý do tại sao khách hàng hài lòng hoặc không hài lòng ở mỗi giai đoạn sử dụng sản phẩm. Có thể bạn muốn theo dõi tình cảm thương hiệu để có thể phát hiện những khách hàng không hài lòng ngay lập tức và phản hồi càng sớm càng tốt. Có thể bạn muốn so sánh tình cảm từ quý này sang quý khác để xem liệu bạn có cần phải thay đổi hay không. Sau đó, bạn có thể tìm hiểu sâu hơn về dữ liệu định tính của mình để xem tại sao tâm lý lại giảm hoặc tăng.

## **3.3 Các kĩ thuật world embedding**

**3.3.1 Định nghĩa**

NLP là một hệ thống phức tập mà chúng ta dùng để diễn đạt ngữ nghĩa. Word embedding (nhúng từ) là một kỹ thuật biểu diễn từ cho phép các từ có nghĩa tương tự có thể được hiểu bởi các thuật toán học máy.

Về mặt kỹ thuật, nó là phép ánh xạ các từ thành vector cảu số thực bằng cách sử dụng mạng neural, mô hình xác suất hoặc giảm số chiều trên ma trận từ.

Hình 44 bên dưới cho ta thấy một từ sau khi chúng ta world embedding.

A picture containing text, font, logo, graphics

Description automatically generated

Hình : Word embedding từ "engine"

**3.3.2 Các loại word embeding**

##### *3.3.2.1 Bag of words*

* Giới thiệu

Bag of words (BoW) là một trong những kĩ thuật vector hóa phổ biến, chúng ta có thể nói đây

là một phương pháp trích xuất vị trí của từ trong câu. Bag of words chỉ xem số lần xuất

hiện của từ trong văn bản mà không quan tâm đến ngữ nghĩa hay thứ tự các từ. Chính

từ lí do đó nó được gọi là "túi" các từ.

* Cách hoạt động:

Văn bản là một đối tượng cực kì lộn xộn và không có cấu trúc. Bag of words giúp ta có

thể chuyển đổi văn bản thành các vector có độ dài cố định. Để hiểu cách hoạt động của BoW, chúng ta sẽ bắt đầu với một ví dụ. Có 2 câu như sau:“John likes to watch movies. Mary likes movies too.” , “John also likes to watch football games.” . Các câu này sẽ được biểu diễn ở dạng tập hợp các từ và trong mỗi câu, ta sẽ thực hiện loại bỏ số lần xuất hiện lặp lại của từ và sử dụng số lượng từ để biểu diễn: {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1}, {"John":1,"also":1,"likes":1,"to":1,"watch":1,"football":1, "games":1}. Giả sử những câu này là một phần của dữ liệu văn bản, dưới đây là tần suất từ được kết hợp cho toàn bộ dữ liệu văn bản.

{"John":2,"likes":3,"to":2,"watch":2,"movies":2,"Mary":1,"too":1,"also":1,"football":1,"games":1}. Cấu trúc từ vựng ở trên của tất cả các từ, với số lượng từ tương ứng của chúng, sẽ được sử dụng để tạo các vectơ cho mỗi câu. Độ dài của vectơ sẽ luôn bằng kích thước của tập hợp từ vựng. Trong trường hợp này, độ dài vectơ là 11. Để biểu diễn các câu gốc trong một vectơ, mỗi vectơ sẽ được khởi tạo bằng tất cả các giá trị số không - [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. Tiếp theo là lặp lại và so sánh với từng từ trong tập hợp từ vựng của chúng ta và tăng giá trị vectơ nếu câu có chứa từ đó.

John likes to watch movies. Mary likes movies too.[1, 2, 1, 1, 2, 1, 1, 0, 0, 0]

John also likes to watch football games.[1, 1, 1, 1, 0, 0, 0, 1, 1, 1]

Ví dụ, trong câu 1, từ likes xuất hiện ở vị trí thứ hai và xuất hiện hai lần. Vì vậy, phần tử thứ hai của vectơ cho câu 1 sẽ là 2: [1, 2, 1, 1, 2, 1, 1, 0, 0, 0]. Vectơ luôn tỷ lệ thuận với kích thước từ vựng của chúng ta.

*3.3.2.2 TF-IDF (term frequency – inverse document frequency)*

Trong NLP, thuật ngữ TF-IDF (term frequency – inverse document frequency) là phương pháp thống kê các từ để xác định mức độ quan trọng của các từ trong đoạn văn bản trong tập nhiều đoạn văn bản khác nhau. TF-IDF giúp chuyển đổi dạng biểu diễn văn bản thành dạng VSM (Vector Space Model) hoặc thành những vector thưa thớt. [23]

* TF(Term Frequency): là tần suất xuất hiện của từ trong văn bản. Tần suất này được chia cho độ dài của đoạn văn bản như một phương thức chuẩn hóa (normalization). TF được thực hiện bởi công thức: (trong đó *t* là một từ trong văn bản; là tần đó xuất hiện của từ *t* trong văn bản *d* ; *T* là tổng số từ có trong văn bản). [23]
* IDF (Inverse Document Frequency): độ quan trọng của từ. Đối với TF thì tính quan trọng của các từ đều bằng nhau như thế biểu hiện đúng tính quan trọng của các từ. Vì vậy , IDF giúp chúng ta bù trừ những từ xuất hiện nhiều lần và tăng độ quan trọng những từ ít xuất hiện nhưng có ý nghĩa đặc biệt. Công thức tính (trong đó N là tổng đoạn văn bản, tập là số văn bản chứ từ *t*).
* TD-IDF được thực hiện bới phép tính :

Với phương pháp TF-IDF, Chúng tôi dùng cho mô hình “logistic regression” và mô hình SVM.

*3.3.2.3 Glove*

Glove là từ tắt của từ globla vector – một dự án open source của Stanford nhắm tạo ra các vector biểu diễn trọng số của các từ. Các vector này được huấn luyện bởi Glove, các mô hình sử dụng thông tin về mối quan hệ ngữ nghĩa giữa các từ với nhau, từ đó cho chung ta kết quả tốt hơn trong các bài toán NLP. [24]

A picture containing text, diagram, line, font

Description automatically generated

Hình : Biểu diễn từ dưới dạng vector

Trang chủ của Glove cung cấp cho chúng ta 4 files chứa các vector được huấn luyện từ dữ liệu của nhiều web khác nhau (đều rất lớn). Chúng tôi sử dụng “Wikipedia 2014 + Gigaword 5”, đây là thư mục nhỏ nhất (“glove.6B.zip”) có dung lượng khoảng 822 MB. Nó được huấn luyện trong một kho ngữ liệu 6 tỷ từ vựng trong tập từ vựng chứ 400 nghìn từ khác nhau.

Khi tải về glove về, chung tôi có 4 files txt: glove.6B.50d.txt, glove.6B.100d.txt, glove.6B.200d.txt, glove.6B.300d.txt. Ở đây 50d, 100d, 200d, 300d biểu diễn số chiều (dimension) của vector từ. Chúng tôi dùng vector 200 chiều để huấn luyện mô hình BiLstm cho bài toán Sentiment analysics.

# CHƯƠNG 4*.*MỘT SỐ THUẬT TOÁN MACHINE LEARNING VÀ DEEP LEARNING

### 4.1 Logistic Regression

#### 4.1.1 Giới thiệu

Logistic Regression là 1 thuật toán phân loại được dùng để gán các đối tượng cho 1 tập hợp giá trị rời rạc (như 0, 1, 2, ...). Một ví dụ điển hình là phân loại Email, gồm có email công việc, email gia đình, email spam, ... Giao dịch trực tuyến có là an toàn hay không an toàn, khối u lành tính hay ác tình. Thuật toán trên dùng hàm sigmoid logistic để đưa ra đánh giá theo xác suất. Ví dụ: Khối u này 80% là lành tính, giao dịch này 90% là gian lận, ...

#### 4.1.2 Mô hình Logistic Regression

Đầu ra dự đoán của logistic regression thường được viết chung dưới dạng

Trong đó θ được gọi là logistic function.

Mục tiêu của hồi quy logistic là đào tạo một bộ phân loại về lớp của một quan sát đầu vào mới. Ở đây chúng ta dùng hàm sigmoid cho bộ phân loại sẽ giúp chúng tôi đưa ra quyết định này

Hàm sigmoid:

Để tạo xác suất, chúng ta sẽ chuyển z qua hàm sigmoid, σ(z). Các

hàm sigmoid (được đặt tên vì nó trông giống chữ s) cũng được gọi là hàm logistic và đặt tên cho hồi quy logistic

A picture containing line, plot, text, diagram

Description automatically generated

Hình : Đồ thị hàm sigmod

Hàm mất mát - Loss function

Đối với các bài toán phân loại (classification), hàm mất mát (loss function) thường được sử dụng đó là cross entropy loss

cross entropy loss =

Trong đó n là số cá thể (instance) của tập dữ liệu

Để hiểu cách hoạt động của cross entropy loss, chúng ta hãy cùng xét trường hợp sau. Chúng ta có cá thể (instance) thuộc nhóm (class) 1, tương ứng với  bằng 1, tuy nhiên Logistic regression model của chúng ta lại dự đoán  bằng 0, khi đó  tương ứng với log(0) và có giá trị âm vô cùng, loss của chúng ta sẽ rất lớn, chứng tỏ model đang dự đoán sai. Điều tương tự sẽ xảy ra trong trường hợp cá thể thuộc nhóm (class) 0, nhưng lại được dự đoán với xác suất là 1.

### 4.2 SVM

**4.2.1 Giới thiệu**

Là phương pháp dựa trên nền tảng lý thuyết thống kê cảu một nền tảng toán học chặc chẽ để đảm bảo rằng kết quả tìm được là chính xác.

Là thuật toán học giám sát (supervied learning) được sử dụng cho bài toán classification.

Là phương pháp thử nghiệm, đưa ra 1 trong những phương pháp mạnh và chính xác nhất trong số các thuật toán nổi tiếng về bài toán classification.

SVM là một phương pháp có tính tổng quát cao nên có thể áp dụng cho nhiều loại toán nhận dạng và phân loại.

**4.2.2 Ý tưởng phương pháp**

Cho trước một tập huấn luyện, được biễu diễn trong không gian vector, trong đó mỗi dữ liệu là một điểm, phương pháp này tìm ra một siêu phẳng quyết định tốt nhất có thể chia các điểm trên không gian này thành 2 lớp riêng biệt (đối với bài toán 2 class) tương ứng với lớp + với lớp -. Chất lượng của siêu phảng này được quyết định bởi khoảng cách (được gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khi đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt, đồng thời việc phân loại càng chính xác.

Mục đích của phương pháp SVM là tìm được khoảng cách biên lớn nhất, điều này được minh họa ở hình 43.

A picture containing sketch, line, white, black and white

Description automatically generated

Hình : Siêu phẳng chia dữ liệu thành 2 lớp + và - với khảng cách biên lớn nhất. Các điểm gần nhất (điểm được khoanh tròn) là các Support Vecotr.

**4.2.3 Nội dung phương pháp**

**Cơ sở lí thuyết**

SVM thực chất là một bài toán tối ưu, mục tiêu của thuật toán này là tìm được một không gian F và siêu phẳng quyết định f trên F sao cho sai số phân loại là thấp nhất.

Bài toán SVM có thể giải bằng kỹ thuật sử dụng toán tử Lagrange để biến đổi về thành dạng đẳng thức. Một đặc điểm thú vị của SVM là mặt phẳng quyết định chỉ phụ thuộc các Support Vector và nó có khoảng cách đến mặt phẳng quyết.

Cho dù các điểm khác bị xóa đi thì thuật toán vẫn cho kết quả giống như ban đầu. Đây chính là điểm nổi bật của phương pháp SVM so với các phương pháp khác vì tất cả các dữ liệu trong tập huấn luyện đều được đùng để tối ưu hóa kết quả.

**Tóm lại**: trong trường hợp nhị phân phân tách tuyến tính, việc phân lớp được thực hiện qua hàm quyết định , hàm này thu được bằng việc thay đổi vectơ chuẩn w, đây là vectơ để cực đại hóa viền chức năng.

Việc mở rộng SVM để phân đa lớp hiện nay vẫn đang được đầu tƣ nghiên cứu. Có một phương pháp tiếp cận để giải quyết vấn để này là xây dựng và kết hợp nhiều bộ phân lớp nhị phân SVM (Chẳng hạn: trong quá trình luyện với SVM, bài toán phân m lớp có thể được biến đổi thành bài toán phân 2\*m lớp, khi đó trong mỗi hai lớp, hàm quyết định sẽ được xác định cho khả năng tổng quát hóa tối đa). Trong phương pháp này có thể đề cập tới hai cách là một-đổi-một, một-đối-tất cả.

**4.2.4 Bài toán phân 2 lớp với SVM**

Bài toán đặt ra là: Xác định hàm phân lớp để phân lớp các mẫu trong tƣơng lai, nghĩa là với một mẫu dữ liệu mới xi thì cần phải xác định xi được phân vào lớp +1 hay lớp -1.

Để xác định hàm phân lớp dựa trên phương pháp SVM, ta sẽ tiến hành tìm hai siêu phẳng song song sao cho khoảng cách y giữa chúng là lớn nhất có thể để phân tách hai lớp này ra làm hai phía. Hàm phân tách tƣơng ứng với phương trình siêu phẳng nằm giữa hai siêu phẳng tìm được.

A picture containing text, receipt, font, line

Description automatically generated

Hình : Minh họa bài toán 2 phân lớp bằng phương pháp SVM

Các điểm mà nằm trên hai siêu phẳng phân tách được gọi là các Support Vector. Các điểm này sẽ quyết định đến hàm phân tách dữ liệu.

**4.2.5 Bài toán nhiều lớp với SVM**

Để phân nhiều lớp thì SVM nguyên bản sẽ chia không gian vector thành 2 phần và quá trình này lặp lại nhiều lần. Khi đó hàm quyết định phân dữ liệu vào lớp thứ i của tập n, đối với bài toán 2 lớp sẽ như sau:

Những phân tử x là support vector sẽ thỏa điều kiện

Như vậy, bài toán phân nhiều lớp sử dụng phương pháp SVM hoàn toàn có thể thực hiện giống như bài toán 2 class. Bằng cách sử dụng chiến lược “one – against – one).

Giả sử bài toán cần phân loại có n class (n>2), chiến lược “one – against – one) sẽ thực hiện lần phân lớp nhị phân sử dụng phương pháp SVM. Mỗi lớp sẽ thự hiện phân tách với n-1 lớp còn lại để xác định n-1 hàm phân tách dựa vào bài toán phân 2 class bằng phương pháp SVM.

### 4.3 Tổng quan về Deep Learning

#### 4.3.1 Giới thiệu

Deep Learning (học sâu) có thể được xem là một lĩnh vực con của Machine Learning (học máy) – ở đó các máy tính sẽ học và cải thiện chính nó thông qua các thuật toán. Deep Learning được xây dựng dựa trên các khái niệm phức tạp hơn rất nhiều, chủ yếu hoạt động với các mạng nơ-ron nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người. Thật ra các khái niệm liên quan đến mạng nơ-ron nhân tạo và Deep Learning đã xuất hiện từ khoảng những năm 1960, tuy nhiên nó lại bị giới hạn bởi khả năng tính toán và số lượng dữ liệu lúc bấy giờ. Trong những năm gần đây, những tiến bộ trong phân tích dữ liệu lớn (Big Data) đã cho phép ta tận dụng được tối đa khả năng của mạng nơ-ron nhân tạo.

Mạng nơ-ron nhân tạo chính là động lực chính để phát triển Deep Learning. Các mạng nơ-ron sâu (DNN) bao gồm nhiều lớp nơ-ron khác nhau, có khả năng thực hiện các tính toán có độ phức tạp rất cao. Deep Learning hiện đang phát triển rất nhanh và được xem là một trong những bước đột phá lớn nhất trong Machine Learning.

#### 4.3.2 Cách thức hoạt động

Mạng nơ-ron nhân tạo trong Deep Learning được xây dựng để mô phỏng khả năng tư duy của bộ não con người. Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, số lượng layer càng nhiều thì mạng sẽ càng “sâu”. Trong mỗi layer là các nút mạng (node) và được liên kết với những lớp liền kề khác. Mỗi kết nối giữa các node sẽ có một trọng số tương ứng, trọng số càng cao thì ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.

Mỗi nơ-ron sẽ có một hàm kích hoạt, về cơ bản thì có nhiệm vụ “chuẩn hoá” đầu ra từ nơ-ron này. Dữ liệu được người dùng đưa vào mạng nơ-ron sẽ đi qua tất cả layer và trả về kết quả ở layer cuối cùng, gọi là output layer. Trong quá trình huấn luyện mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm ra bộ giá trị của trọng số sao cho phán đoán là tốt nhất. Các hệ thống Deep Learning yêu cầu phần cứng phải rất mạnh để có thể xử lý được lượng dữ liệu lớn và thực hiện các phép tính phức tạp. Nhiều mô hình Deep Learning có thể mất nhiều tuần, thậm chí nhiều tháng để triển khai trên những phần cứng tiên tiến nhất hiện nay.

### 4.4 Một số thuật toán trong Deep Learning

#### 4.4.1 RNN

##### *4.4.1.1 Giới thiệu*

Mạng neural hồi tiếp (RNN) là một loại mạng neual nhân tạo sử dụng dữ liệu

tuần tự hoặc dữ liệu chuỗi thời gian. Thuật toán này thường được sử dụng cho các vấn đề

thứ tự hoặc thời gian, ví dụ như dịch ngôn ngữ, xử lý ngôn ngữ tự nhiên, nhận dạng giọng

nói và chú thích hình ảnh. Hiện nay thuật toán đã được tích hợp vào nhiều ứng dụng phổ

biến như Siri, tìm kiếm bằng giọng nói và Google dịch. Giống như mạng neural tích chập (CNN), mạng neural thực hiện lặp lại việc đào tạo để học. Trong khi mạng neural truyền thống giả định rằng đầu vào và đầu ra là độc lập với nhau, đầu ra của mạng neural hồi tiếp phụ thuộc vào các phần tử trước đó trong chuỗi.

Mặc dù các sự kiện trong tương lai cũng sẽ hữu ích trong việc xác định đầu ra của một trình tự nhất định nhưng mạng neural hồi tiếp thực hiện lặp lại một chiều nên không thể tính tới các sự kiện này trong các dự đoán của chúng. Các biến thể của RNN có thể khắc phục yếu điểm này và mang lại kết quả thuyết phục hơn chính vì thể ngày nay người ta đa số sử dụng các biến thể của RNN để giải quyết các bài toán thực tiễn.

RNN hình tượng hóa mỗi trạng thái ẩn tương ứng với mỗi nút trong mạng thành

một mốc thời gian, để tính trạng thái ẩn hoặc giá trị kích hoạt (activation) ở thời điểm t+1

ta sẽ phải dùng trạng thái ẩn ở thời điểm t làm đầu vào, ở thời điểm t=0, giá trị này thường

được khai báo bằng 0. Để dễ hiểu ta có thể hình dung một mạng RNN như một bộ phim, ý tưởng chính của mô hình RNN là chúng ta sẽ dùng các dữ liệu từ phút đầu tiên đến phút thứ 29 của bộ phim để dự đoán diễn biến xảy ra ở phút thứ 30, lúc này để tính toán diễn biến ở mốc thời gian t=30 chúng ta sẽ phải dùng dữ liệu được lưu ở mốc t=29. Vì có đầy đủ các dữ kiện đã xảy ra trước đó nên việc dự đoán diễn biến tiếp theo sẽ chính xác hơn rất nhiều. Đây cũng chính là lý do mà RNN đặc biệt hiệu quả hơn CNN trong các bài toán sinh (generate), dự đoán khi mà nó có thể lan truyền được các dữ liệu mang tính lịch sử.

##### *4.4.1.2 Kiến trúc*

A diagram of a graph

Description automatically generated with low confidence

Hình : Kiến trúc của một mạng RNN [17]

Trong RNN, các input xt sẽ được kết hợp với hidden layer *ht-1*​ bằng hàm fw​ để tính toán ra hidden layer *ht* hiện tại và output yt sẽ được tính ra từ *ht* ​, W là tập các trọng số và nó được ở tất cả các cụm, các L1, L2,, …, Ltlà các hàm mất mát sẽ được giải thích sau. Như vậy kết quả từ các quá trình tính toán trước đã được "nhớ" bằng cách kết hợp thêm *ht-1*​ ​ tính ra *ht* để tăng độ chính xác cho những dự đoán ở hiện tại. Cụ thể quá trình tính toán được viết dưới dạng toán như sau:

*ht  =* *fw (ht-1, xt )*

Hàm *fw* ​ chúng ta sẽ xử dụng hàm **tanh**, công thức trên sẽ trở thành:

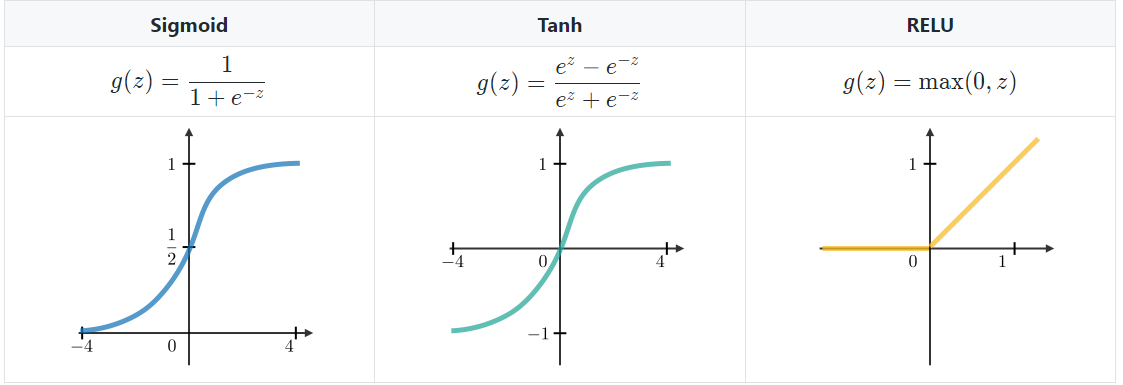
*ht*​=*tanh*(*Whh*​*ht*−1​+*Wxh*​ *xt*)

yt  = Why ht

RNN nó sử dụng 3 ma trận trọng số cho 2 quá trình tính toán: *Whh* kết hợp với "bộ nhớ trước" h t-1​ và W xh kết hợp với xt để tính ra "bộ nhớ của bước hiện tại" *ht* từ đó kết hợp với W hy để tính ra yt.

*4.4.1.3* *Các hàm kích hoạt thường dùng trong RNN*

Các hàm kích hoạt thường dùng trong các mạng RNN được miêu tả như hình:



Hình : Các hàm kích hoạt thường dùng trong RNN [18]

Hàm mất mát – Loss function

Trong trường hợp của mạng neural hồi quy, hàm mất mát L của tất cả các bước thời gian được định nghĩa dựa theo mất mát ở mọi thời điểm như sau:

### 4.4.2 LSTM

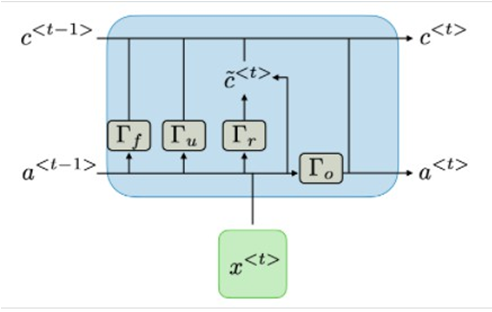
##### *4.4.2.1 Giới thiệu*

Hai điểm yêu cố hữu của một mạng RNN là không cập nhật được dữ liệu từ quá xa cũng như không xem xét các dữ liệu trong tương lai ở thời điểm hiện tại. LSTM được ra đời như một cách tối ưu để khắc phục vấn đề này, về mặt thuật ngữ thì hai điểm yếu này được gọi là vanishing và exploding gradient. Lí do chúng thường xảy ra đó là rất khó để có được sự phụ thuộc dài hạn vì multiplicative gradient có thể tăng/giảm theo hàm mũ tương ứng với số lượng các tầng. Tức là sự phụ thuộc dài hạn sẽ tăng theo hàm mũ qua từng layer của mạng RNN, điều này làm cho việc truy xuất thông tin từ một node cách rất xa node hiện tại là rất khó khăn.

Những thuật ngữ khoa học trên có lẽ sẽ rất khó hiểu với những người không hoạt động trong lĩnh vực học máy, chính vì thể hãy sử dụng ví dụ về bộ phim ở phần RNN để dễ dàng hình dung hơn. Ở phút thứ 30, việc ta tra cứu diễn biến của phim ở phút thứ nhất có thể rất dễ dàng, tuy nhiên nếu bộ phim kéo dài 4-5 tiếng đồng hồ, ở phút thứ 250 của bộ phim có một diễn biến cần chúng ta biết được diễn biến ở phút đầu tiên thì mới có thể dự đoán được, điều này sẽ rất khó khăn vì phút đầu tiên cách rất xa so với phút thứ 250. Chưa kể nếu chúng ta có thể xem xét các diễn biến ở các phút sau phút thứ 250 cũng là một điều tuyệt vời để có thể dự đoán diễn biến ở phút đó. LSTM làm được điều này tuy nhiên chi phí để hiện thực hóa nó là rất đắt đỏ và chi tiết bên trong mô hình rất phức tạp, gần như chúng ta sẽ dùng nó như một hộp đen mà rất khó diễn giải.

##### *4.4.2.2 Kiến trúc và cách hoạt động của LSTM*

Thay vì chỉ dùng một hàm kích hoạt đơn giản bên trong mỗi node như RNN truyền thống, LSTM dùng mô hình sau để thực hiện xét duyệt trạng thái ẩn qua 4 cổng như sau:



Hình : Kiến trúc của một node trong mạng LSTM [18]

Đầu tiên chúng ta sẽ nói về 4 loại cổng (gate) như trên hình đã biểu diễn đó là

Γu,Γr,Γf ,Γo:

• Γu: Cổng này sẽ xem xét dữ liệu cũ và quyết định xem dữ liệu đó nên có tầm quan trọng như thế nào ở hiện tại.

• Γr: Cổng này sẽ xem xét các dữ liệu ở tương lai nếu có và quyết định xem có nên bỏ qua dữ liệu đó hay không.

• Γf : Cổng này sẽ quyết định có xóa ô hay không, tức là chúng ta có nên quên dữ liệu ở thời điểm hiện tại hay không.

• Γo: Cổng này sẽ quyết định biểu thị đầu ra như thế nào và ở mức độ bao nhiêu.

Tiếp theo, chúng ta cần hiểu các đặc tính a,c, c˜ là gì, các đặc tính này sẽ được tính bằng công thức như sau:

• c˜<t> = tanh(Wc[Γr ∗ a<t−1>,x<t>] + bc) với các giá trị a,x,b giống như một mạng

RNN truyền thống đã mô tả ở phần RNN

• c<t> = Γu∗c˜<t> + Γf ∗ c<t−1>

• a<t> = Γo ∗ c<t>

Xét về mặt kiến trúc tổng quát, mạng LSTM cũng là một mạng RNN nên chúng ta sẽ không trình bày lại về cách nó hoạt động, chúng ta chỉ giải thích những điểm đặc biệt của LSTM. Nhờ những thay đổi bên trong 1 node như đã đề cập ở trên, LSTM trở thành một trong những mô hình được áp dụng nhiều nhất với các bài toán dịch tự động, sinh nhạc,... Nó thể hiện hiệu suất vượt trội tuy nhiên việc triển khai còn gặp hạn chế lớn về vấn đề chi phí

##### *4.4.2.3 Tổng quan về Bi - LSTM*

Bộ nhớ dài hạn hai chiều (Bi-LSTM) là quá trình tạo ra bất kỳ mạng thần kinh nào có thông tin trình tự theo cả hai hướng ngược (tương lai đến quá khứ) hoặc tiến (quá khứ đến tương lai).

Theo hai chiều, dữ liệu đầu vào của chúng tôi chảy theo hai hướng, làm cho một bi-lstm khác với LSTM thông thường. Với LSTM thông thường, chúng ta có thể tạo luồng đầu vào theo một hướng, ngược hoặc xuôi. Tuy nhiên, theo hai hướng, chúng ta có thể tạo luồng đầu vào theo cả hai hướng để bảo toàn thông tin trong tương lai và quá khứ. Để giải thích rõ hơn, hãy lấy một ví dụ.

Trong câu “cậu bé đi đến …..” chúng ta không thể điền vào chỗ trống. Tuy nhiên, khi chúng ta có một câu trong tương lai “ cậu bé ra khỏi trường”, chúng ta có thể dễ dàng dự đoán khoảng trống trong quá khứ giống như điều chúng ta muốn thực hiện bằng mô hình của mình và LSTM hai chiều cho phép mạng thần kinh thực hiện điều này.

A picture containing diagram, text, screenshot, line

Description automatically generated

Hình : Kiến trúc của mạng Bi -LSTM [19]

Trong sơ đồ, chúng ta có thể thấy kiến trúc của Bi-LSTM thường bao gồm một lớp đầu vào, hai lớp ẩn (một cho LSTM tiến và một cho LSTM lùi) và một lớp đầu ra. LSTM chuyển tiếp xử lý chuỗi đầu vào theo hướng thuận trong khi LSTM lùi xử lý chuỗi đầu vào theo hướng ngược. Các đầu ra từ cả hai LSTM sau đó được nối và đưa vào lớp đầu ra. BI-LSTM thường được sử dụng khi cần có trình tự sắp xếp các tác vụ. Loại mạng này có thể được sử dụng trong các mô hình phân loại văn bản, nhận dạng giọng nói và dự báo.

### 4.4.3 CNN

*4.4.3.1 Giới thiệu*

Mạng nơ-ron tích chập (Convolution Nerual Network) là một dạng đặt biệt của mạng nơ-ron được dùng trong việc xử lý các dữ liệu trong không gian 2 chiều. Mạng nơ-ron tích chập được áp dụng nhiều ứng dụng thực tế trong đó có bài toán phân loại văn bản, đặt biệt trong lĩnh vực thị giác máy thì mạng nơ-ron tích chấp rất mạnh. Tên của mạng được bát nguồ từ toán tử tích chập (convolution) trong toán học. Tích chập là một dạng đặt biệt của phép toán tuyến tính. Mạng tích chập có thể được định nghĩa là một mạng nở-ron trong đó sử dụng phép tính toán tích chập thay cho phép nhân thông thường ở ít nhất một lớp mạng. Như vậy, kiến trúc mạng tích chập cũng bao gồm nhiều lớp mạng như nhiều lớp nhưng pahir có 1 lớp dùng phép tính tích chập. Hình 34 là một kiến trúc truyền thống của mạng tích chập gồm 3 lớp chức năng cơ bản mạng.

A picture containing sketch, mammal, screenshot, art

Description automatically generated

Hình : Cấu trúc của mạng CNN truyền thống [18]

Trong hình 35 là cấu trúc bài toán CNN cho xử lí ngôn ngữ tự nhiên. CNN cho bài toán NLP cũng tương tự bài toán thị giác máy, điểm khác biệt ở đây là thay các pixel chúng ta có ma trận cảu vector từ.

A picture containing diagram, plan, line, text

Description automatically generated

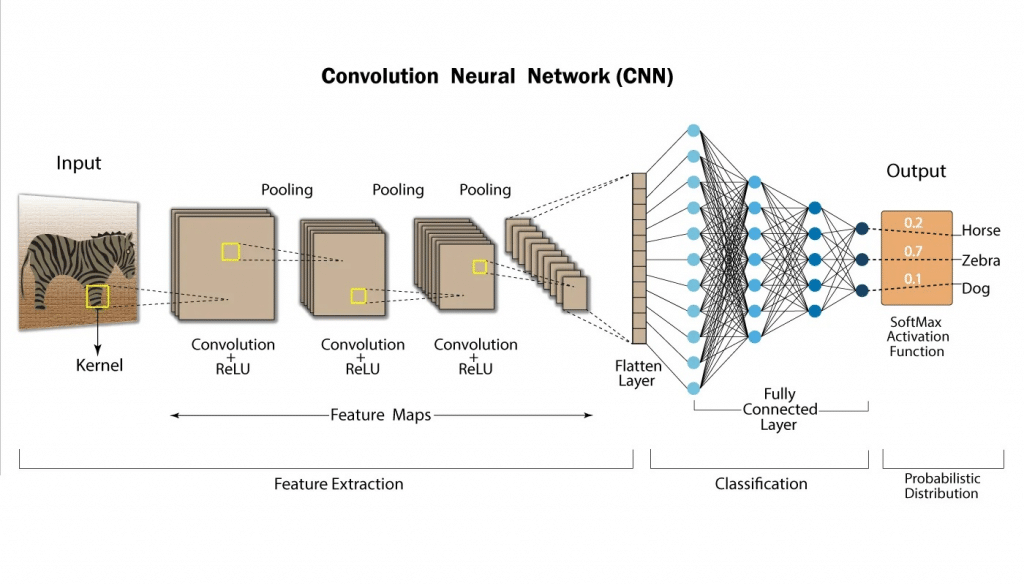
Hình : Cấu trúc mô hình CNN gồm 3 lớp cho bài toán dự đoán 11 nhãn [20]

*4.4.3.2 Kiến trúc tổng quan của mạng CNN*

Mạng nơ-ron tích chập là sự kết hợp của mạng nơ-ron với các lớp tích chập và lớp các lớp kết nối kết nối đầy đủ. Một mô hình mạng CNN có kiến trúc như hình 34.

* Lớp tích chập (CONV):

Lớp tích chập thực hiện tích chập ảnh đầu vào với các bộ lọc hay kenel. Số lượng các bộ lọc có thể lên đến vài chục hoăc hơn tùy thuộc vào ứng dụng. Giá trị các phân tử trong các bộ lọc thu được trong quá trình huấn luyện. Lớp tích chập thực hiện gán các giá trị 0 xung quanh biên ảnh nhằm tạo ra các feature map có kích thước với ảnh ban đầu. Hình 36 mô tả kiến trúc một mạng nơ-ron tích chập với 2 lớp convolution, 2 lớp pooling và 2 lớp fully connected. Các ngõ ra tích cực cho phép nhận dạng các ảnh đầu vào. Có 3 ngõ ra tương đương với 3 nhãn là horse, zebra, dog.



Hình : Kiến trúc mạng tích chập cho bài toán phân loại ảnh [21]

Mạng nơ-ron tích chập trong hình 36 có kiến trúc gần giống với kiến trúc cơ bản nhưng số lượng các lớp tăng gấp đôi. Ảnh ngõ vào là ma trận 2 chiều với các giá trị nhị phân trong hình 37. Bộ lọc được sử dụng có kích thước 2x2. Giả sử bộ lọc có các giá trị như trong hình 37. Nếu chúng ta thực hiện kết gán 0 vào biên ảnh ban dầu thì kết quả ma trận ngõ ra lớp hơn ma trân bạn ban đầu. Các phân tử của ma trận ngõ ra (feature map) thu được từ phép tính tổng cảu phép nhân 2 của phân tử bộ lọc và ảnh, khi trượt bộ lọc hết ảnh.

A picture containing screenshot, text, diagram, rectangle

Description automatically generated

Hình : Mô tả tích chập ảnh với bộ lọc (kenel ) [22]

Trong hình 37, ma trận thu được từ việc nhân ma trận bộ lọc khi trượt hết ma trận ngõ ra. Mỗi bước trượt có giá trị 1 diểm ảnh. Sử dụng phép nhân từng phân tử cảu ma trận rồi cộng hết lại ta tính được giá trị ngõ ra cho ma trận feature map. Như trình bày ở đầu, các bộ lọc thu được trong quá trình huấn luyện mô hình thya vì sử dụng các bộ lọc xử lí phổ thông. Với các thông số được thiết lập trước khi xâu dựng các phép toán tích chập.

* + - Deepth: độ sâu, tương ứng với số lượng bộ lọc hay số lượng kernel. Số lượng bộ lọc quyết định số lương feature map ngõ ra. Trong hình 36, chúng ta sử dụng 3 bộ lọc cho các ảnh ngõ vảo để tạ ra 3 ma trận feature khác nhâu
    - Fillter size: kích thước bộ lọc. Được định nghĩa là chiều cao hay chiều rộng của bộ lọc. Các bộ này sẽ trượt hết ma trân ảnh ngõ vòa khi thực hiện tích chập. Khi tích chập thực hiện trong đó bộ không bị lật được gọi là phép tương quan chéo (cross-correlation) hơn là tích chập. Tuy nhiên, nó không có gây ảnh hưởng đến quá trình tích chập khi chúng ta giả định rằng các bộ lọc được huấn luyện như các bộ lọc xử lí ảnh đã bị lật.
    - Strike: Bước trượt. Bước trượt quyết định số điểm ảnh thay dổi sau mỗi lần trượt ma trận kernel trên ma trận ảnh. Bình thường nếu chúng ta không bỏ qua điểm ảnh nào thì strike=1. Nếu strike=2 thì ta thu được ma trân feature sau khi tích chập bằng ¼ kích thước ma trân đầu vào.
    - Zero-padding: ảnh đầu vào thường được gán các giá trị 0 cho các đường biên để tạo ra các feature map có kích thước không đổi với ảnh đầu vào trong trường hợp strike=1.Ma trận thu được từ lớp convolution có chiều dài L’ được xác đinh theo công thức sau:

Trong đó, L là chiều dài ảnh đầu vào, K là kích thước bộ lọc, P là số lượng kích thước đường biên được gán thêm 0 và S là Strike.

* Lớp hàm kích hoạt thượng gặp
  + Rectified Linear Unit: Tầng rectified linear unit (ReLU) là một hàm kích hoạt g được sử dụng trên tất cả các thành phần. Mục đích của nó là tăng tính phi tuyến tính cho mạng. Những biến thể khác của ReLU được tổng hợp ở ảnh sau:

A picture containing text, screenshot, line, diagram

Description automatically generated

Hình : Các loại hàm ReLU và chức năng [18]

* + Softmax: Hàm được sử dụng nhiều trong lĩnh vực học sâu. Hàm thường dùng làm hàm kích hoạt ở ngõ ra hơn là các nơ-ron lớp ẩn (hidden). Ngõ ra cảu hàm softmax tương úng với phân bố xác suất phân loại của các ngõ ra. Công thức hàm soooftmax như sau: Trong đó z là vector ngõ ra của nơ-ron. Ngõ ra hàm softmax cho biết xác suất của ngõ ra trong tổng các ngõ ra .
* Lớp pooling

Các mạng CNN cơ bản có 3 lớp. Lớp convolution thực hiện phép tính tích chập tạo ra tập giá trình tích cực tuyến tính trong các feature map. Các giá trị này được cho qua hàm kích hoạt phi tuyến như hàm ReLU. Lớp pooling còn được gọi là tang detector. Lớp thứ 3 trong mô hình là lớp pooling được dùng để hiểu chỉnh ngõ ra. Hàm pooling có liên quan đến các ngõ ra lân cận. Các hàm pooling thường được áp dụng trong các mô hình CNN là hàm Max Pooling, Average Pooling hoặc Sum pooling. Chức năng của pooling giúp chúng ta giữ lại những được trưng nổi trội nhất trong feature map hoặc tính trung bình các đặc trưng đó:

A screenshot of a graph

Description automatically generated with low confidence

Hình : Max pooling va Average pooling [18]

* Lớp kết nối đầy đủ (Fully connected)

Lớp Fully connected được sử dụng mạng nhiều lớp với hàm kích hoạt thường là softmax cho lớp ở ngõ ra. Ngõ vào của lớp fully connected là một vector một chiều. Vì vậy, ma trận lớp pooling cuối cùng thường chuyển thành một vector và đưa đến lớp fully connected.

A picture containing sketch, screenshot, diagram, line

Description automatically generated

*Hình 40: Mô phỏng về lớp fully connected [18]*

*4.4.3.3 Mô hình kết hợp của CNN-LSTM*

Với lí thuyết layer LSTM nói ở phần trên kết hợp với kết trúc của mô hình mạng nơ-ron tích chập, chúng ta tạo ra một mô hình cho bài toán classification NLP. Cấu trúc mô hình ban đầu gồm một lớp tích chập ban đầu sẽ nhận các nhúng từ làm đầu vào. Đầu ra của lớp tiếp theo sẽ được gộp thành một kích thước nhỏ hơn, sau đó được vào lớp LSTM. Cấu trúc cơ bản của mô hình là lớp tích chập trích xuất những tính năng cục bộ từ dữ liệu và lớp LSTM sau đó sẽ dụng thứ tự của các tính năng đã nói trích xuất từ lớp tích chập trên để tìm hiểu về các thứ tự văn bản của dữ liệu đầu vào.

A diagram of a dropout

Description automatically generated with medium confidence

Hình : Kiến trúc mô hình CNN-LSTM

**CHƯƠNG 5: THỰC NGHIỆM**

**5.1 BÀI TOÁN**

AI có thể giải thích (XAI) là một lĩnh vực đang phát triển nhanh chóng nhằm cung cấp tính minh bạch và khả năng diễn giải cho các mô hình máy học phức tạp. Trong bối cảnh phân loại cảm xúc, XAI có thể giúp cung cấp thông tin chi tiết về cách mô hình đưa ra dự đoán và những tính năng mà mô hình đang sử dụng để xác định cảm xúc được thể hiện trong một văn bản nhất định. Phân loại cảm xúc là một nhiệm vụ đầy thách thức do tính chất chủ quan của cảm xúc và sự phức tạp của ngôn ngữ con người. Các mô hình học máy truyền thống, chẳng hạn như mạng lưới thần kinh, có thể đạt được độ chính xác cao trong các nhiệm vụ phân loại cảm xúc nhưng thường thiếu tính minh bạch và khả năng diễn giải. Điều này có thể gây khó khăn cho các doanh nghiệp và tổ chức trong việc hiểu lý do đằng sau những dự đoán của mô hình và đưa ra quyết định sáng suốt dựa trên những dự đoán đó. Các kỹ thuật XAI có thể giúp giải quyết vấn đề này bằng cách cung cấp thông tin chi tiết về hoạt động bên trong của mô hình. Ví dụ: các kỹ thuật về tầm quan trọng của tính năng có thể được sử dụng để xác định các tính năng mà mô hình đang sử dụng để đưa ra dự đoán của nó. Điều này có thể giúp các doanh nghiệp và tổ chức hiểu được khía cạnh nào của văn bản là quan trọng nhất để xác định cảm xúc.

## **5.2 DỮ LIỆU**

Tập dữ liệu được nhóm lấy để xây dựng mô hình từ nguồn: <https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp>. Đây là tập dữ liệu về phân loại cảm xúc dưới dạng text (văn bản) và được gán vào các nhãn: sadness (buồn rầu), anger (tức giận), love (yêu), surprise (ngạc nhiên), fear (sợ hãi), joy (vui vẻ)

Tập dữ liệu chia làm 3 tập train, test và validation dươi dạng file txt.

Tập train train có 16000 dòng cho 6 nhãn cần dự đoán. Tương tự, tập test và tập validation có 2000 dòng cho 6 nhãn dự đoán.

**5.3 PHƯƠNG PHÁP VÀ KẾT QUẢ**

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

* Machine learning: Logistic regression, SVM
* Deep learning: CNN - LSTM, Bi-LSTM
* Explainable AI: LIME , SHAP

Với cách World Embedding, chúng tôi sẽ dùng mô hình TF-IDF và Glove, độ đo chính nhóm dùng là accuracy của mô hình.

* Mô hình Logistic Regression

Logistic Regression – là một trong những thuật toán học có giám sát được sử dụng nhiều nhất cho bài toán phân loại trong đó có bài toán phân loại văn bản. Logistic Regression là một mô hình tuyến tính trong bài toán phân nhãn – thuật ngữ tuyến tính có ý nghĩa rất quan trọng vì nó là sự kết hợp tuyến tính của đầu vào và các tham số để đưa ra kết quả dự đoán. Logistic Regression được xem là một mô hình mạng nơ-ron dơn giản giống mạng nơ-ron “perceptron”.

Với cách chuyển đổi từ thành các tính năng: nhóm dùng “CountVectorizer” của thư viện scikit-learn để chuyển câu văn thành ma trận từ và dùng thư viện NLTK để xóa các kí tự, từ không quan trọng trong câu. Tiếp theo, nhóm dùng TF-IDF để đánh trọng cho các từ trong ma trận từ tạo ở trên.

Các tham số nhóm dùng cho mô hình Logistic Regression cho bài toán như sau: “solver: newton-cg” một trong những tham số dùng để tối ưu hóa hàm mất mát cho mô hình và “L2 regularization” – là một kĩ thuật giúp mạng nơ-ron tránh bị overfitting.

A screenshot of a computer program

Description automatically generated with medium confidence

Hình : Cấu trúc mô hình Logistic Regression

Kết quả sau khi huấn luyện mô hình logistic Regression với các tham số trên, nhóm thu được độ chính xác của mô hình trên tập dữ liệu test là 86,1%.

A picture containing text, screenshot, font, line

Description automatically generated

Hình : Kết quả của mô hình Logistic Regression

* Mô hình Support Vector Machine (SVM)

SVM là một mô hình có nét tương đồng với mô hình Logistic Regression với cách mà hàm tối ưu mất mát sẽ quyết định các ranh giới giữa các điểm dữ liệu. Sự khác biệt chính, SVM dùng “kernel function” – hàm biến đổi một không gian phức tạp, phi tuyến tính thành một không gian có số chiều cao hơn để có thể tìm thấy một siêu phẳng thích hợp phân tách các điểm dữ liệu. SVM sẽ tìm cách tối đa khoảng cách cảu từng điểm dữ liệu từ siêu phẳng bằng cách sử dụng “support vector”.

Với các bước chuẩn bị dữ liệu cho huấn luyện mô hình SVM giống với Logistic Regression ở những bước đầu. Còn về tham sô cho mô hình “linear SVM” như sau: “loss: hinge” là hàm mất mát với tối ưu SGD (stochastic gradient descent) và L2 -regularization và một số thông số khác.

A screenshot of a computer program

Description automatically generated with medium confidence

Hình : Cấu trúc mô hình SVM

Kết quả thu được với mô hình SVM cho bài toán là 85% khi nhóm thực mô hình trên tập dữ liệu test.

* Mô hình CNN-LSTM

Với cấu trúc mô hình bao gồm:

+ 1 lớp embedding input dữ liệu

+ 1 lớp dropout

+ 1 lớp convolution

+ 1 lớp pooling

+1 lớp lstm

+ 1 lớp dense

A picture containing text, receipt, diagram, font

Description automatically generated

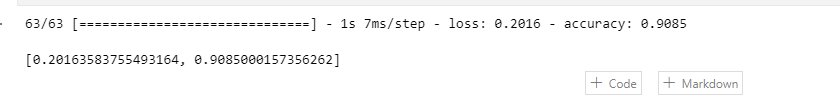
Hình : Cấu trúc mô hình CNN-LSTM

Với mô hình này, nhóm huấn luyện với các tham số bacth\_size, epochs và một số tham số khác. Nhóm thường thay đổi các tham số bacth\_size, epochs, chỉ số ở lớp dense, lớp dropout để cho mô hình đạt kết quả tốt nhất có thể. Thì với mô hình CNN-LSTM huấn luyện với tham số bacth\_size=256, epochs = 5 thì cho kết quả tốt nhât với độ chính xác là 90% trên tập dữ liệu test.

A picture containing text, receipt, font, white

Description automatically generated

Hình : Quá trình huấn luyện mô hình CNN-LSTM với 5 epoch



Hình : Độ chính xác của mô hình CNN-LSTM

* Mô hình BiLSTM

Với mô hình BiLSTM, nhóm thực hiện mô hình với 5 lớp:

+ 1 lớp embedding

+3 lớp Bidirectional

+1 lớp dense

Với mô hình nay, nhóm huấn luận trên máy vật lí không có GPU hỗ trợ thì train mất khoảng 32 tiếng. Sau nhóm dùng google colab có hỗ trợ GPU, nhóm huấn luyện mô hình tốn ít thơi gian hơn. Cũng như mô hình CNN-LSTM thì nhóm cũng thay đổi các tham số epochs, bacth\_size, … và dùng thêm Glove để đánh thêm trọng số cho ma trận vector từ để tăng độ chính xác mô hình.

A picture containing text, font, screenshot, diagram

Description automatically generated

Hình : Kiến trúc mô hình BiLSTM

A screenshot of a computer

Description automatically generated with medium confidence

Hình : Các tham số huấn luyện mô hình BiLSTM

Với các tham số huấn luyện trong hình 53, thì mô hình thu được kết quả với độ chính xác mô hình 93% trên tập dữ liệu test.

A screenshot of a computer code

Description automatically generated with low confidence

Hình : Kết quả của mô hình trên tập dữ liệu test

Sau khi train bốn mô hình, nhóm sử dụng LIME và SHAP để giải thích được dự đoán và triển khai cách giải thích được lên Streamlit.

A screenshot of a computer

Description automatically generated with low confidence

Hình : Giao diện giải thích trên Streamlit

Hình 56 là giao diện khi triển khai lên Streamlit với ô textbox để nhập câu thể hiện cảm xúc. “Choose classifier” sẽ cho là các lựa chọn thuật toán mà nhóm đã huấn luyện dùng để giải thích.

A screenshot of a computer

Description automatically generated with low confidence

Hình : Các thuật toán trong select box

“Choose option Explainer” sẽ cho lựa chọn phương pháp giải thích là LIME và SHAP.

A picture containing text, screenshot, line, font

Description automatically generatedHình : Lựa chọn phương pháp giải thích

Khi nhập một câu thể hiện cảm xúc vào và chọn thuật toán phân loại ta sẽ được lời giải thích về thuật toán đó. Ví dụ ta nhập câu “im updating my blog because i feel shitty”. Kết quả giải thích với LIME sẽ như hình sau

A screenshot of a computer

Description automatically generated with medium confidence

Hình : Kết quả giải thích với phương pháp LIME

Nhìn vào hình 59 ta thấy được với câu ví dụ ở trên từ “shitty” đóng góp vào kết quả dự đoán. LIME chỉ hiển thị những từ ảnh hưởng đến kết quả dự đoán.

A picture containing text, screenshot, line, plot

Description automatically generated

Hình : Kết quả giải thích phương pháp SHAP

Hình 60 với phương pháp SHAP cho ta thấy được các từ trong câu ảnh hưởng tích cực hay tiêu cực. Trong đó những từ màu đỏ ảnh hưởng tích cực vào dự đoán, những từ màu xanh ảnh hưởng tiêu cực vào kết quả dự đoán.

# CHƯƠNG 6: KẾT LUẬN

## **6.1 KẾT QUẢ ĐẠT ĐƯỢC**

### 6.1.1 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 dự đoán cảm xúc thông qua phương pháp LIME, SHAP. 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.

### 6.1.2 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, SHAP cho dự đoán cảm xúc chúng tôi biết nhiều hơn về thư viện Tensorflow, biết rõ hơn về cách thức hoạt động của mô hình RNN, LSTM.

## **6.2 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”.

## **6.3 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 như áp dụng thêm tập dữ liệu bằng tiếng Việt, 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] | C. Molnar, Interpretable Machine Learning, 2023-03-02, 2023. |
| [2] | Carlos Guestrin,Sameer Singh,Marco Tulio Ribeiro, "“Why Should I Trust You?”Explaining the Predictions of Any Classifier," 2016. |
| [3] | Scott Lundberg and Su-In Lee, "A Unified Approach to Interpreting Model Predictions," no. 2, 2017. |
| [4] | Mukund Sundararajan and Amir Najmi, "The many Shapley values for model explanation," 2019. |
| [5] | Ian Covert and Su-In Lee, "Improving KernelSHAP: Practical Shapley Value Estimation via Linear Regression," pp. 2 - 9, 2020. |
| [6] | Gabriel Laberge and Yann Pequignot, ""Understanding Interventional TreeSHAP : How and Why it Works"," 2022. |
| [7] | Gianluigi Lopardo, Damien Garreau and Frederic Precioso, "A Sea of Words: An In-Depth Analysis of Anchors for Text Data," 2022. |
| [8] | Alex Goldstein, Adam Kapelner, Justin Bleich and Emil Pitkin, ""Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation"," *Computational and Graphical Statistics,* pp. 44-65, 2015. |
| [9] | Goldstein, Alex, Adam Kapelner, Justin Bleich, and Emil Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," *Computational and Graphical Statistics,* pp. 44-65, 2015. |
| [10] | Christoph Molnar, Timo Freiesleben, Gunnar König, Giuseppe Casalicchio, Marvin N. Wright and Bernd Bischl, "elating the Partial Dependence Plot and Permutation Feature Importance to the Data Generating Process," pp. 4-10, 2021. |
| [11] | J. H. Friedman, "Greedy function approximation: A gradient boosting machine," pp. 1189-1232, 2001. |
| [12] | Greenwell, Brandon M., Bradley C. Boehmke and Andrew J. McCarthy, "A simple and effective model-based variable importance measure," 2018. |
| [13] | Seyedehzahra Khoshmanesh, Tuba Yavuz and Robyn R. Lutz, "Learning Feature Interactions With and Without Specifications," 2021. |
| [14] | Coursera, "What is Natural Language Processing? Definition and Examples". |
| [15] | R. Wolff, "7 Benefits of Natural Language Processing (NLP)," Monkey Learn, 2020. |
| [16] | VinBigData, "Kỹ thuật phân tích quan điểm (Sentiment Analysis)," 2022. |
| [17] | D. Duong, "Recurrent Neural Network(Phần 1): Tổng quan và ứng dụng," 2018. |
| [18] | Amidi, Afshine Amidi và Shervine, Convolutional Neural Networks and Recurrent Neural Networks cheatsheet. |
| [19] | Y. Verma, Complete Guide To Bidirectional LSTM (With Python Codes). |
| [20] | Muhammad Zain Amin and Noman Nadeem, "Convolutional Neural Network: Text Classification Model for Open Domain Question Answering System," 2018. |
| [21] | N. Hưng, "Neural Network là gì? Đặc điểm và ứng dụng của Neural Network". |
| [22] | Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, "Dive into Deep Learning," 2023. |
| [23] | N. V. Hung, "Trích chọn thuộc tính trong đoạn văn bản với TF-IDF," 20 10 2017. [Online]. Available: https://viblo.asia/p/trich-chon-thuoc-tinh-trong-doan-van-ban-voi-tf-idf-Az45bAOqlxY. |
| [24] | T. t. n. tạo, "Hướng dẫn sử dụng GloVe," 26 5 2019. [Online]. Available: https://trituenhantao.io/kien-thuc/huong-dan-su-dung-glove/. |