ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ - TIN HỌC**

🙡🕮🙣

****

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**ĐỀ TÀI: XÂY DỰNG HỆ THỐNG PHÂN TÍCH PHẢN HỒI CỦA KHÁCH HÀNG VỀ MẶT HÀNG ĐIỆN THOẠI DI ĐỘNG**

SVTH : Đoàn Ngọc Thế Vinh

Lớp : 41K14

GVHD: ThS. Nguyễn Văn Chức

*Đà Nẵng, tháng 5 năm 2019*

# LỜI CÁM ƠN

Trước hết em xin gửi lời cảm ơn sâu sắc đến quý thầy cô khoa Thống kê – Tin học trường Đại học Kinh tế - Đại học Đà Nẵng đã cung cấp những kiến thức quý báu và bổ ích trong suốt bốn năm học vừa qua, tạo điều kiện thuận lợi giúp đỡ em trong quá trình học tập và hoàn thành chuyên đề tốt nghiệp.

Trong quá trình thực hiện đề tài, em xin chân thành cám ơn thầy Nguyễn Văn Chức đã tận tình hướng dẫn, giúp đỡ, giải đáp những thắc mắc khó khăn của em khi hoàn thành đề tài này.

Tuy nhiên vì hạn chế về kiến thức và thời gian thực hiện đề tài có hạn nên còn có nhiều thiếu sót rất mong nhận được sự thông cảm, đánh giá và đóng góp ý kiến của quý thầy cô và các bạn.

Em xin chân thành cảm ơn !

# LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, dữ liệu và kết quả nêu trong đề tài là trung thực và chưa từng công bố trong các công trình khác. Nếu không đúng như nêu trên, tôi xin hoàn toàn chịu trách nhiệm.

*Đà Nẵng, Ngày Tháng Năm 2019*

*Người thực hiện*

*Đoàn Ngọc Thế Vinh*

# MỤC LỤC

[LỜI CÁM ƠN i](#_Toc7794269)

[LỜI CAM ĐOAN ii](#_Toc7794270)

[MỤC LỤC iii](#_Toc7794271)

[DANH MỤC HÌNH VÀ BẢNG v](#_Toc7794272)

[LỜI MỞ ĐẦU 1](#_Toc7794273)

[CHƯƠNG 1 GIỚI THIỆU VỀ MACHINE LEARNING VÀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN 3](#_Toc7794274)

[1.1. Tổng quan về Machine Learning: 3](#_Toc7794275)

[1.1.1. Giới thiệu về Machine Learning: 3](#_Toc7794276)

[1.1.2. Các kĩ thuật của Machine Learning: 3](#_Toc7794277)

[1.1.3. Ứng dụng của Machine Learning: 5](#_Toc7794278)

[1.2. Tổng quan về xử lý ngôn ngữ tự nhiên (Natural Language Processing): 7](#_Toc7794279)

[1.2.1. Giới thiệu về NLP (Natural Language Processing): 7](#_Toc7794280)

[1.2.2. Phân loại NLP: 7](#_Toc7794281)

[1.2.3. Ứng dụng cơ bản của NLP: 7](#_Toc7794282)

[CHƯƠNG 2 GIỚI THIỆU BÀI TOÁN PHÂN TÍCH PHẢN HỒI KHÁCH HÀNG 10](#_Toc7794283)

[2.1. Giới thiệu về bài toán phân tích phản hồi. 10](#_Toc7794284)

[2.2. Lựa chọn môi trường xây dựng và triển khai hệ thống. 11](#_Toc7794285)

[2.2.1. Giới thiệu về ngôn ngữ Python: 11](#_Toc7794286)

[2.2.2. Giới thiệu một số thư viện được sử dụng: 12](#_Toc7794287)

[2.2.3. Quy trình giải quyết bài toán: 14](#_Toc7794288)

[CHƯƠNG 3 QUY TRÌNH XÂY DỰNG MÔ HÌNH PHÂN TÍCH PHẢN HỒI CỦA KHÁCH HÀNG VỀ MẶT HÀNG ĐIỆN THOẠI DI ĐỘNG 15](#_Toc7794289)

[3.1. Thu thập dữ liệu: 16](#_Toc7794290)

[3.1.1. Giới thiệu công cụ Selenium Webdriver: 16](#_Toc7794291)

[3.1.2. Trực quan hoá dữ liệu đã thu thập được 17](#_Toc7794292)

[3.2. Tiền xử lý dữ liệu: 22](#_Toc7794293)

[3.2.1. Giới thiệu về tiền xử lý trong NLP: 22](#_Toc7794294)

[3.2.2. Các bước trong tiền xử lý dữ liệu: 22](#_Toc7794295)

[3.3. Vector hóa dữ liệu: 27](#_Toc7794296)

[3.3.1. Giới thiệu về mô hình không gian vector (Vector space model): 27](#_Toc7794297)

[3.3.2. Giới thiệu về biểu diễn vector bằng TF-IDF: 29](#_Toc7794298)

[3.4. Mô hình hóa dữ liệu: 32](#_Toc7794299)

[3.4.1. Giới thiệu một số thuật toán: 32](#_Toc7794300)

[3.4.2. Lựa chọn thuật toán cho bài toán: 35](#_Toc7794301)

[3.4.3. Giới thiệu về Logistic Regression: 36](#_Toc7794302)

[3.5. Đánh giá mô hình: 40](#_Toc7794303)

[3.5.1. Tại sao phải đánh giá mô hình: 40](#_Toc7794304)

[3.5.2. Đánh giá bằng Confusion Matrix: 40](#_Toc7794305)

[3.5.3. Đánh giá bằng đường cong ROC: 44](#_Toc7794306)

[CHƯƠNG 4 TRIỂN KHAI MÔ HÌNH PHÂN TÍCH PHẢN HỒI CHƯA ĐƯỢC GÁN NHÃN 47](#_Toc7794307)

[4.1. Kết quả triển khai 47](#_Toc7794308)

[KẾT LUẬN 52](#_Toc7794309)

[TÀI LIỆU THAM KHẢO 53](#_Toc7794310)

# DANH MỤC HÌNH VÀ BẢNG

**Danh mục hình:**

[Hình 2. 1: Quy trình xây dựng và phân tích phản hồi của khách hàng 14](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722907)

[*Hình 3. 1: Quy trình xây dựng mô hình phân tích phản hồi khách hàng…………… 15*](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722888)

[Hình 3. 2: Kiến trúc của Selenium Webdriver 17](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722889)

[Hình 3. 3: Dữ liệu được thu thập từ công cụ Selenium webdriver. 18](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722890)

[Hình 3. 4: Biểu đồ cột biểu diễn số sao của bình luận được crawl. 19](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722891)

[Hình 3. 5: Biểu đồ tròn thể hiện tỉ lệ phần trăm phân loại nhãn. 19](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722892)

[Hình 3. 6: Biểu đồ cột biểu diễn số hãng điện thoại có trong bộ dữ liệu. 20](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722893)

[Hình 3. 7: Biểu đồ thể hiện số lượng trong từng bình luận của khách hàng. 20](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722894)

[Hình 3. 8: WordCloud từ ngữ có trong bộ dữ liệu được thu thập. 21](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722895)

[Hình 3. 9: Biểu đồ thể hiện top 50 từ xuất hiện nhiều nhất trong mỗi cuốn sách tiếng anh. 25](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722896)

[Hình 3. 10: Vector được biểu diễn dưới dạng one hot. 27](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722897)

[Hình 3.11:Vector biểu diễn dưới dạng sự phân tán. 27](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722898)

[Hình 3. 12: Wordcloud của bộ từ được lấy ra từ Vocabulary của Tf-Idf. 30](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722899)

[Hình 3. 13: Mô hình K-NN 33](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722900)

[Hình 3. 14: Đồ thị tuyến tính Logistic Regression. 36](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722901)

[Hình 3. 15: Biểu đồ biểu diễn hệ số lớn nhất và nhỏ nhất của Logistic Regression đã được huấn luyện dựa trên tf-idft. 38](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722902)

[Hình 3. 16: Normoalization confusion matrix của bộ dữ liệu test sau khi được predict. 42](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722903)

[Hình 3. 17: Unnormoalization confusion matrix của bộ dữ liệu test sau khi được predict. 42](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722904)

[Hình 3. 18: Đồ thị Receiver Operating Characteristic curve 44](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722905)

[Hình 3. 19: Đồ thị đường cong ROC của mô hình 45](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722906)

[*Hình 4. 2: Biếu đồ tròn biểu diển tỉ lệ cảm xúc sau khi phân tích dự đoán …………..49*](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722915)

[Hình 4. 1: Biểu đồ cột biểu diễn kết quả sau khi phân tích dự đoán 49](file:///C:\Users\vntdoan\Downloads\DA2019%20word%20(1)%20(1).docx#_Toc7722916)

**Danh mục bảng:**

[Bảng 4. 1: Bảng dữ liệu bao gồm 10 bình luận chưa được gán nhãn 47](#_Toc7905921)

[Bảng 4. 2: Bảng dữ liệu bao gồm 10 bình luận sau khi tiền xữ lý. 48](#_Toc7905922)

[Bảng 4. 3: Bảng dữ liệu bao gồm 10 được dán nhãn sau khi được áp dụng mô hình đã xây dựng. 49](#_Toc7905923)

# LỜI MỞ ĐẦU

Ngày nay, với sự phát triển vượt bậc của khoa học và công nghệ, đặc biệt là sự bùng nổ của Internet với các phương tiện truyền thông xã hội, thương mại điện tử,… Đã cho phép mọi người không chỉ chia sẻ thông tin trên đó mà còn thể hiện thái độ, quan điểm của mình đối với các sản phẩm, dịch vụ và các vấn đề xã hội khác. Vì vậy mà Internet đã trở nên vô cùng quan trọng và là nguồn cung cấp lượng thông tin vô cùng lớn và hữu ích.

Hầu hết các doanh nghiệp đều luôn muốn quan tâm đến ý kiến, phản hồi của khách hàng về sản phẩm, dịch vụ của họ như thế nào. Các đánh giá của khách hàng một mặt giúp cho những người dùng khác định hướng trong việc chọn lựa sản phẩm, mặt khác giúp cho các doanh nghiệp định hướng cải tiến chất lượng. Số lượng đánh giá về một sản phẩm mà chúng ta nhận được ngày càng tăng và có thể đến từ nhiều nguồn khác nhau (web bán hàng, diễn đàn, blog, mạng xã hội ...). Vì vậy, để có thể tổng hợp ý kiến phản hồi của khách hàng về chất lượng, thì phải tự động hóa được công việc thu thập và phân tích đánh giá.

Công nghệ phân lớp dữ liệu đã, đang và sẽ phát triển mạnh mẽ trước những khao khát tri thức của con người. Trong những năm qua, phân lớp dữ liệu đã thu hút sự quan tâm các nhà nghiên cứu trong nhiều lĩnh vực khác nhau như học máy (machine learning), hệ chuyên gia (expert system), thống kê (statistics) ... Công nghệ này cũng ứng dụng trong nhiều lĩnh vực thực tế như: thương mại, nhà băng, maketing, nghiên cứu thị trường, bảo hiểm, y tế, giáo dục ... Phân lớp văn bản là bài toán cơ bản trong khai phá quan điểm. Các hệ thống phân lớp văn bản là các hệ thống phải có khả năng xác định, khai phá ra nội dung thông tin. Có thể coi phân lớp quan điểm là bài toán phân lớp văn bản theo hai lớp tích cực và tiêu cực.

Từ đó bài toán ở đây đặt ra là làm thế nào để xây dựng một mô hình máy học đã được huấn luyện, để rồi dùng mô hình đó phân tích những đánh giá phản hồi mới về một dòng sản phẩm di động mới nào đó cho ra đó là đánh giá tích cực hay là tiêu cực với độ chính xác cao.

Mục tiêu đề tài:

* Hiểu được khái niệm cơ bản của Machine Learning và xữ lý ngôn ngữ tự nhiên
* Quy trình xây dựng một mô hình phân tích 2 lớp
* Cách đánh giá một mô hình.
* Triển khai mô hình phân tích bộ dữ liệu chưa được gán nhãn.

Cấu trúc đề tài: Ngoài các phần như mở đầu, kết luận, đề tài gồm 4 chương:

* Chương 1: Giới thiệu tổng quan về Machine Learning và xữ lý ngôn ngữ tự nhiên.

Nội dung: Trình bày khái niệm, phân nhóm, ứng dụng của Machine Learning và xữ lý ngôn ngữ tự nhiên

* Chương 2: Giới thiệu bài toán phân tích phản hồi khách hàng.

Nội dung: Hiểu được bài toán phân lớp là gì, quy trình để giải bài toán phân lớp quan điểm.

* Chương 3: Quy trình xây dựng mô hình phân tích phản hồi của khách hàng về mặt hàng điên thoại di động.

Nội dung: Các bước xây dựng mô hình phân tích phản hồi của khách hàng về mặt hàng điện thoại di động, cách đánh giá một mô hình phân lớp.

* Chương 4: Triển khai mô hình trên bộ dữ liệu chưa gán nhãn.

Nội dung: Kết quả sau khi cho mô hình đã xây dựng phân tích bộ dữ liệu chưa được gán nhãn.

Kết quả dự kiến: Xây dựng thành công một mô hình phân tích ý kiến của khách hàng về mặt hàng điện thoại di động từ đó sử dụng nó để phân tích những dữ liệu mới.

# CHƯƠNG 1 GIỚI THIỆU VỀ MACHINE LEARNING VÀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

## Tổng quan về Machine Learning:

### Giới thiệu về Machine Learning:

Những năm gần đây, AI – Artificial Intelligence ( Trí tuệ nhân tạo), và cụ thể hơn là Machine Learning (Học máy hoặc máy học) nổi lên như một bằng chứng của cuôc cách mạng lần thứ tư ( 1 – Động cơ hơi nước, 2 – Năng lượng điện, 3 – Công nghệ thông tin ). Trí tuệ nhân tạo đang len lỏi vào mọi lĩnh vực trong cuộc sống mà có thể chúng ta không nhận ra. Ví dụ như: Xe tự hành của Google và Tesla, hệ thông tự nhận diện khuôn mặt (tag) của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google Deepmind,… Chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning.

Machine Learning là một tập con của AI. Theo định nghĩa của Wikipedia: “ Machine Learning is the subfield of computer science that gives computers the ability to learn without being explicitly programmed”. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa học máy tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, Machine Learning đã biến lượng tài nguyên đó thành tri thức rất có ích cho con người hoặc các hãng công nghệ lớn đó.

### Các kĩ thuật của Machine Learning:

Theo phương thức học, các thuật toán Machine Learning thường được chia làm 3 nhóm: Học có giám sát, học không giám sát, học bán giám sát.

#### Học có giám sát:

Học có giám sát là thuật toán dự đoán đầu ra (*out come*) của một dữ liệu mới (*new input*) dựa trên các cặp (*input, outcome*) đã biết từ trước. Cặp dữ liệu này còn được gọi là (*data, label – dữ liệu, nhãn*). Học có giám sát là nhóm phổ biến nhất trong các thuật toán Machine Learning.

Một cách toán học, học có giám sát là khi chúng ta có một tập hợp các biến đầu vào X={x1, x2,…, xN} và một tập hợp nhãn tương ứng Y={y1, y2,…, yN}, trong đó các xi, yi là các vector. Các cặp dữ liệu biết trước (xi, yi) ∈ X × Y được gọi là tập dữ liệu huấn luyện. Từ tập dữ liệu huấn luyện này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử tương ứng của tập Y

Mục đích là xấp xỉ hàm số thật tốt để khi có một dữ liệu mới, chúng ta có thể tính được nhãn tương ứng của nó .

Ví dụ 1: Khá giống với cách học của con người khi còn nhỏ, ta đưa bảng chữ cái cho một đứa trẻ và chỉ cho chúng đây là chữ A, đây là chữ B. Sau một vài lần được dạy thì trẻ có thể nhận biết được đâu là chữ A, đâu là chữ B trong một cuốn sách mà chúng chưa nhìn thấy bao giờ.

Ví dụ 2: Thuật toán dò các khuôn mặt trong một bức ảnh đã được phát triển từ rất lâu. Thời gian đầu, facebook sử dụng thuật toán này để chỉ ra các khuôn mặt trong một bức ảnh và yêu cầu người dùng tag friends - tức gán nhãn cho mỗi khuôn mặt. Số lượng cặp dữ liệu (khuôn mặt, tên người) càng lớn, độ chính xác ở những lần tự động tag tiếp theo sẽ càng lớn.

Học có giám sát có thể chia ra thành hai loại thuật toán chính đó là: Classification (Phân lớp) và Regression (Hồi quy).

**\* Chuyên đề “*Xây dựng mô hình phân tích phản hồi của khách hàng về mặt hàng điện thoại di động*” sẽ sử dụng học có giám sát để xây dựng mô hình phân tích và lựa chọn thuật toán hồi quy.**

#### Học không giám sát:

Trong thuật toán này, chúng ta không biết được *outcome* hay *nhãn* mà chỉ có dữ liệu đầu vào. Học không giám sát sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Một cách toán học, học không giám sát là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng.

Học có giám sát có thể chia ra thành hai loại thuật toán chính đó là: Clustering (Phân cụm) và Association (Luật kết hợp)

#### Học bán giám sát:

Học bán giám sát là một lớp của kỹ thuật học máy, sử dụng cả dữ liệu đã gán nhãn và chưa gán nhãn để huấn luyện - điển hình là một lượng nhỏ dữ liệu có gán nhãn cùng với lượng lớn dữ liệu chưa gán nhãn. Học bán giám sát đứng giữa học không giám sát (không có bất kì dữ liệu đã được nhãn nào) và có giám sát (toàn bộ dữ liệu đều được gán nhãn).

### Ứng dụng của Machine Learning:

Sau đây là các lĩnh vực phổ biến mà machine learing góp mặt:

#### Xử lý ảnh

Bài toán xử lý ảnh(Image Processing) giải quyết các vấn đề phân tích thông tin từ hình ảnh hay thực hiện một số phép biến đổi. Một số ví dụ là:

* Gắn thẻ hình ảnh(Image Tagging), giống như Facebook, một thuật toán tự động phát hiện khuôn mặt của bạn và bạn bè trên những bức ảnh. Về cơ bản, thuật toán này học từ những bức ảnh mà bạn tự gắn thẻ cho mình trước đó.
* Nhận dạng ký tự(Optical Character Recognition), là một thuật toán chuyển dữ liệu trên giấy tờ, văn bản thành dữ liệu số hóa. Thuật toán phải học cách nhận biết ảnh chụp của một ký tự là ký tự nào.
* Ô tô tự lái(Self-driving cars), một phần cơ chế sử dụng ở đây là xử lý ảnh. Một thuật toán machine learning giúp phát hiện các mép đường, biển báo hay các chướng ngại vật bằng cách xem xét từng khung hình video từ camera.

#### Phân tích văn bản

Phân tích văn bản(Text analysis) là công việc trích xuất hoặc phân lọi thông tin từ văn bản. Các văn bản ở đây có thể là các facebook posts, emails, các đoạn chats, tài liệu,… Một số ví dụ phổ biến là:

* Lọc spam(Spam filtering), là một trong những ứng dụng phân loại văn bản được biết và sử dụng nhiều nhất. Ở đây, phân loại văn bản là xác định chủ đề cho một văn bản. Bộ lọc spam sẽ học cách phân loại một email có phải spam không dựa trên nội dung và tiêu đề của email.
* Phân tích ngữ nghĩa(Sentiment Analysis), học cách phân loại một ý kiến là tích cực, trung tính hay tiêu cực dựa trên nội dung văn bản của người viết.
* Khai thác thông tin(Information Extraction), từ một văn bản, học cách để trích xuất các thông tin hữu ích. Chẳng hạn như trích xuất địa chỉ, tên người, từ khóa,…

#### Khai phá dữ liệu

Khai phá dữ liệu(Data mining) là quá trình khám phá ra các thông tin có giá trị hoặc đưa ra các dự đoán từ dữ liệu. Định nghĩa này có vẻ bao quát, nhưng bạn hãy nghĩ về việc tìm kiếm thông tin hữu ích từ một bảng dữ liệu rất lớn. Mỗi bản ghi sẽ là một đối tượng cần phải học, và mỗi cột là một đặc trưng. Chúng ta có thể dự đoán giá trị của một cột của bản ghi mới dựa trên các bản ghi đã học. Hoặc là phân nhóm các bản ghi của bản. Sau đây là những ứng dụng của khai phá dữ liệu:

* Phát hiện bất thường(Anomaly detection), phát hiện các ngoại lệ, ví dụ như phát hiện gian lận thẻ tín dụng. Bạn có thể phát hiện một giao dịch là khả nghi dựa trên các giao dịch thông thường của người dùng đó.
* Phát hiện các quy luật(Association rules), ví dụ, trong một siêu thị hay một trang thương mại điện tử. Bạn có thể khám phá ra khách hàng thường mua các món hàng nào cùng nhau. Dễ hiểu hơn, khách hàng của bạn khi mua món hàng A thường mua kèm món hàng nào? Các thông tin này rất hữu ích cho việc tiếp thị sản phẩm.
* Gom nhóm(Grouping), ví dụ, trong các nền tảng SaaS, người dùng được phân nhóm theo hành vi hoặc thông tin hồ sơ của họ.
* Dự đoán(Predictions), các cột giá trị(của một bản ghi mới trong database). Ví dụ: bạn có thể dự đoán giá của căn hộ dựa trên các dữ liệu về giá các căn hộ bạn đã có.

#### Trò chơi điện tử & Robot

Trò chơi điện tử(Video games) và robot(Robotics) là lĩnh vực lớn có sự góp mặt của machine learning. Nếu ta có một nhân vật cần di chuyển và tránh các chướng ngại vật trong game. Machine learning có thể học và giải quyết công việc này thay bạn. Một kỹ thuật phổ biến được áp dụng trong trường hợp này là Học tăng cường(Reinforcement learning). Ở đó, máy sẽ học tăng cường với mục tiêu là giải quyết nhiệm vụ trên. Học tăng cường là tiêu cực nếu nó va phải chướng ngại vật, là tích cực nếu nó chạm tới đích.

Một thành tựu gần đây nhất là cỗ máy Alpha Go của Google DeepMind đã đánh bại kỳ thủ cờ vậy số 1 thế giới. Trong khi cờ vây là một trò chơi có không gian trạng thái cực kỳ lớn.

## Tổng quan về xử lý ngôn ngữ tự nhiên (Natural Language Processing):

### Giới thiệu về NLP (Natural Language Processing):

NLP(Natural Language Processing) là khái niệm để chỉ các kĩ thuật, phương pháp thao tác trên ngôn ngữ tự nhiên bằng máy tính. Bạn cần phân biệt ngôn ngữ tự nhiên (ví dụ như tiếng Việt, tiếng Anh, tiếng Nhật… là những ngôn ngữ trong giao tiếp thường ngày) và ngôn ngữ nhân tạo ( như ngôn ngữ lập trình, ngôn ngữ máy, …).

Xử lý ngôn ngữ chính là xử lý thông tin khi đầu vào là “Dữ liệu ngôn ngữ” (dữ liệu cần biến đổi) tức là dữ liệu “văn bản” hay “tiếng nói”. Các dữ liệu liên quan đến ngôn ngữ viết và nói đang dần trở nên kiểu dữ liệu chính con người có và lưu trữ dưới dạng điện tử. Đặc điểm chính của các kiểu dữ liệu nàu là không có cấu trúc hoặc nửa cấu trúc và không thể lưu trữ trong các khuôn dạng cố định như các bảng biểu.

### Phân loại NLP:

Trong NLP có 2 quan điểm cơ bản :

*1. Xử lý các từ ngữ bằng máy tính.*

*2. Làm cho máy tính hiểu được các từ ngữ.*

Hiện tại, cả 2 hướng này đều đang được tích cực nghiên cứu và phát triển, nhờ đó rất nhiều các hệ thống hiệu quả đã và đang được tạo ra.

### Ứng dụng cơ bản của NLP:

Xử Lý Ngôn Ngữ Tự Nhiên có vai trò hết sức quan trọng trong ngành Khoa Học Máy Tính. Nó có vô vàn ứng dụng hữu ích trong cuộc sống cũng như nghiên cứu. Chúng ta có thể điểm qua một vài ứng dụng của xử lý ngôn ngữ tự nhiên như:

* Nhận dạng chữ viết: Có hai kiểu nhận dạng, thứ nhất là nhận dạng chữ in, ví dụ nhận dạng chữ trên sách giáo khoa rồi chuyển nó thành dạng văn bản điện tử như dưới định dạng doc của Microsoft Word chẳng hạn. Phức tạp hơn là nhận dạng chữ viết tay, có khó khăn bởi vì chữ viết tay không có khuôn dạng rõ ràng và thay đổi từ người này sang người khác. Với chương trình nhận dạng chữ viết in có thể chuyển hàng ngàn đầu sách trong thư viện thành văn bản điện tử trong thời gian ngắn. Nhận dạng chữ viết của con người có ứng dụng trong khoa học hình sự và bảo mật thông tin (nhận dạng chữ ký điện tử).
* Nhận dạng tiếng nói: Nhận dạng tiếng nói rồi chuyển chúng thành văn bản tương ứng. Giúp thao tác của con người trên các thiết bị nhanh hơn và đơn giản hơn, chẳng hạn thay vì gõ một tài liệu nào đó bạn đọc nó lên và trình soạn thảo sẽ tự ghi nó ra. Đây cũng là bước đầu tiên cần phải thực hiện trong ước mơ thực hiện giao tiếp giữa con người với robot. Nhận dạng tiếng nói có khả năng trợ giúp người khiếm thị rất nhiều.
* Tổng hợp tiếng nói: Từ một văn bản tự động tổng hợp thành tiếng nói. Thay vì phải tự đọc một cuốn sách hay nội dung một trang web, nó tự động đọc cho chúng ta. Giống như nhận dạng tiếng nói, tổng hợp tiếng nói là sự trợ giúp tốt cho người khiếm thị, nhưng ngược lại nó là bước cuối cùng trong giao tiếp giữa robot với người.
* Dịch tự động (Machine translate): Như tên gọi đây là chương trình dịch tự động từ ngôn ngữ này sang ngôn ngữ khác. Một phần mềm điển hình về tiếng Việt của chương trình này là Evtrans của Softex, dịch tự động từ tiếng Anh sang tiếng Việt và ngược lại, phần mềm từng được trang web vdict.com mua bản quyền, đây cũng là trang đầu tiên đưa ứng dụng này lên mạng. Tháng 10 năm 2008 có hai công ty tham gia vào lĩnh vực này cho ngôn ngữ tiếng Việt là công ty Lạc Việt (công ty phát hành từ điển Lạc Việt) và Google, một thời gian sau đó Xalo.vn cũng đưa ra dịch vụ tương tự.
* Tìm kiếm thông tin (Information retrieval): Đặt câu hỏi và chương trình tự tìm ra nội dung phù hợp nhất. Thông tin ngày càng đầy lên theo cấp số nhân, đặc biệt với sự trợ giúp của Internet việc tiếp cận thông tin trở lên dễ dàng hơn bao giờ hết. Việc khó khăn lúc này là tìm đúng nhất thông tin mình cần giữa bề bộn tri thức và đặc biệt thông tin đó phải đáng tin cậy. Các máy tìm kiếm dựa trên giao diện web như Google hay Yahoo hiện nay chỉ phân tích nội dung rất đơn giản dựa trên tần suất của từ khoá và thứ hạng của trang và một số tiêu chí đánh giá khác để đưa ra kết luận, kết quả là rất nhiều tìm kiếm không nhận được câu trả lời phù hợp, thậm chí bị dẫn tới một liên kết không liên quan gì do thủ thuật đánh lừa của các trang web nhằm giới thiệu sản phẩm (có tên tiếng Anh là SEO viết tắt của từ Search Engine Optimization). Thực tế cho đến bây giờ chưa có máy tìm kiếm nào hiểu được ngôn ngữ tự nhiên của con người trừ trang www.ask.com được đánh giá là "hiểu" được những câu hỏi có cấu trúc ở dạng đơn giản nhất. Mới đây cộng đồng mạng đang xôn xao về trang Wolfram Alpha, được hứa hẹn là có khả năng hiểu ngôn ngữ tự nhiên của con người và đưa ra câu trả lời chính xác. Lĩnh vực này hứa hẹn tạo ra bước nhảy trong cách thức tiếp nhận tri thức của cả cộng đồng.
* Tóm tắt văn bản: Từ một văn bản dài tóm tắt thành một văn bản ngắn hơn theo mong muốn nhưng vẫn chứa những nội dung thiết yếu nhất.
* Khai phá dữ liệu (Data mining) và phát hiện tri thức: Từ rất nhiều tài liệu khác nhau phát hiện ra tri thức mới. Thực tế để làm được điều này rất khó, nó gần như là mô phỏng quá trình học tập, khám phá khoa học của con người, đây là lĩnh vực đang trong giai đoạn đầu phát triển. Ở mức độ đơn giản khi kết hợp với máy tìm kiếm nó cho phép đặt câu hỏi để từ đó công cụ tự tìm ra câu trả lời dựa trên các thông tin trên web mặc cho việc trước đó có câu trả lời lưu trên web hay không (giống như trang Yahoo! hỏi và đáp, nơi chuyên đặt các câu hỏi để người khác trả lời), nói một cách nôm na là nó đã biết xử lý dữ liệu để trả lời câu hỏi của người sử dụng, thay vì máy móc đáp trả những gì chỉ có sẵn trong bộ nhớ.

**\*Chuyên đề “*Xây dựng mô hình phân tích phản hồi của khách hàng về mặt hàng điện thoại di động*” là bài toán ứng dụng của NLP. Nói rõ hơn đây là bài toán *Phân loại văn bản* thành hai ý kiến *Positive (Tích cực) và Negative (Tiêu cực)***

# CHƯƠNG 2 GIỚI THIỆU BÀI TOÁN PHÂN TÍCH PHẢN HỒI KHÁCH HÀNG

## Giới thiệu về bài toán phân tích phản hồi.

Bài toán phân tích phản hồi được quy về bài toán phân lớp dữ liệu là quá trình phân lớp một đối tượng dữ liệu vào một hay nhiều lớp cho trước nhờ một mô hình phân lớp mà mô hình này được xây dựng dựa trên một tập hợp các đối tượng dữ liệu đã được gán nhãn từ trước gọi là tập dữ liệu học (tập huấn luyện). Quá trình phân lớp còn được gọi là quá trình gán nhãn cho các đối tượng dữ liệu.

Như vậy, nhiệm vụ của bài toán phân lớp dữ liệu là cần xây dựng mô hình (bộ) phân lớp để khi có một dữ liệu mới vào thì mô hình phân lớp sẽ cho biết dữ liệu đó thuộc lớp nào.

Có nhiều bài toán phân lớp dữ liệu, như phân lớp nhị phân, phân lớp đa lớp, phân lớp đa trị,….

Phân lớp nhị phân là quá trình tiến hành việc phân lớp dữ liệu vào một trong hai lớp khác nhau dựa vào việc dữ liệu đó có hay không một số đặc tính theo quy định của bộ phân lớp.

Phân lớp đa lớp là quá trình phân lớp với số lượng lớp lớn hơn hai. Như vậy, tập hợp dữ liệu trong miền xem xét được phân chia thành nhiều lớp chứ không đơn thuần chỉ là hai lớp như trong bài toán phân lớp nhị phân. Về bản chất, bài toán phân lớp nhị phân là trường hợp riêng của bài toán phân lớp đa lớp.

Trong phân lớp đa trị, mỗi đối tượng dữ liệu trong tập huấn luyện cũng như các đối tượng mới sau khi được phân lớp có thể thuộc vào từ hai lớp trở lên. Ví dụ như trang web về việc bùng phát bệnh cúm gia cầm, thủy cầm tại một số tính phía Bắc vừa thuộc về lĩnh vực y tế liên quan đến lây bệnh sang người nhưng cũng thuộc về lĩnh vực kinh tế liên quan đến ngành chăn nuôi… 4 Trong những trường hợp như vậy, việc sắp xếp một tài liệu vào nhiều hơn một lớp là phù hợp với yêu cầu thực tế.

Đề tài “Xây dựng hệ thông phân tích quan điểm của khách hàng về mặt hàng điện thoại di động” của chúng ta là bài toán phân loại bình luận thành 2 quan điểm là **Positive (tích cực)** và **Negative (tiêu cực)** cho nên ta có thể quy về bài toán phân lớp nhị phân (binary). Và nh đã nói từ đầu thì ta sẽ sử dụng thuật toán học có giám sát để xây dựng mô hình cho bài toán này.

## Lựa chọn môi trường xây dựng và triển khai hệ thống.

### Giới thiệu về ngôn ngữ Python:

Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level) ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.

Python đã được hình thành vào cuối những năm 1980, và việc thực hiện nó vào tháng 12 năm 1989 bởi Guido van Rossum tại Centrum Wiskunde & Informatica (CWI) ở Hà Lan như là một kế thừa cho ngôn ngữ ABC (tự lấy cảm hứng từ SETL) có khả năng xử lý ngoại lệ và giao tiếp với Hệ điều hành Amoeba. Van Rossum là tác giả chính của Python, và vai trò trung tâm của ông trong việc quyết định hướng phát triển của Python.

Trải qua hơn hàng chục năm phát triển, Python là một trong những ngôn ngữ được sử dụng nhiều nhất trong dậy lập trình và nghiên cứu khoa học. Rất nhiều trường đại học sử dụng Python để dậy về lập trình cho các sinh viên ngành Khoa Học Máy Tính. Rất nhiều công ty lớn sử dụng Python để xây dựng hệ thống như Google, Youtube, Instagram, Dropbox, Atlassian... Python là một ngữ sử dụng được cho nhiều mô hình lập trình, đơn giản khi học và sử dụng. Tôi sử dụng Python chưa lâu nhưng khi so sánh việc Code sử dụng Pythong thì nó ngắn hơn rất nhiều so với khi viết bằng PHP hoặc Java. Bạn có thể bay bổng tự do với Python hoặc cũng có thể bắt nó trở lên vững chắc và mạnh mẽ như Java. Theo những thông tin mà tôi được biết thì Python cũng là một ngôn ngữ rất phát triển trong lĩnh vực Data Science và Machine Learning. Python cũng cung cấp những hàm và thư viện xử lý ngôn ngữ tuyệt vời. Scikit-learn và Tensor-flow là 2 thư viện Machine Learning nổi tiếng được viêt bằng Python. Các đặc điểm của Python:

* Ngữ pháp đơn giản, dễ đọc.
* Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented)
* Hỗ trợ module và hỗ trợ gói (package)
* Xử lý lỗi bằng ngoại lệ (Exception)
* Kiểu dữ liệu động ở mức cao.
* Có các bộ thư viện chuẩn và các module ngoài, đáp ứng tất cả các nhu cầu lập trình.
* Có khả năng tương tác với các module khác viết trên C/C++ (Hoặc Java cho Jython, hoặc .Net cho IronPython).
* Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

Python đã trở thành ngôn ngữ chung cho nhiều ứng dụng khoa học dữ liệu. Nó có nhiều thư viện dùng để load dữ liệu, trực quan hoá, thống kê, xử lý ngôn ngữ tự nhiên, xử lý hình ảnh…

### Giới thiệu một số thư viện được sử dụng:

#### Thư viện Sklearn:

Scikit-learn (viết tắt là sklearn) là một thư viện mã nguồn mở dành cho học máy - một ngành trong trí tuệ nhân tạo, rất mạnh mẽ và thông dụng với cộng đồng Python, được thiết kế trên nền NumPy và SciPy. Scikit-learn chứa hầu hết các thuật toán machine learning hiện đại nhất, đi kèm với documentations, luôn được cập nhật. Đây là một thư viện hổ trợ các thuật toán Machine Learning như Decison Tree, KNN, Navie Bayes, SVM(Support Vector Machine), ANN (Artificial Neural Network), Linear Regression, K-Mean, Logistic Regression… Scikit-learn là một thư viện viết bằng python và rất dễ sử dụng.

*Tại sao nên dùng scikit-learn?*

- Hỗ trợ hầu hết các thuật toán của machine learning một cách đơn giản, hiệu quả mà chúng ta không cần phải mất công ngồi cài đặt lại.

- Có tài liệu hướng dẫn sử dụng

- Độ tin cậy cao do scikit-learn được xây dựng bởi các chuyên gia hàng đầu

- Có nguồn dữ liệu phong phú: iris, digit, …

*Khi nào thì sử dụng scikit-learn?*

Đơn giản là khi cần giải quyết bài toán có thể áp dụng được học máy. Tất nhiên là bạn vẫn có thể giải quyết mà không cần dùng tới sự hỗ trợ của scikit-learn và bạn biết chắc chắn là không hề đơn giản rồi.

*Sử dụng scikit-learn như thế nào?*

Trước tiên máy bạn phải được cài đặt Python rồi. Sau đó bạn có thể vào trực tiếp trang chủ của scikit-learn để xem cách cài đặt nó. Đơn giản nhất là sử dụng Anaconda

Như đã nói, sklearn được xây dựng trên NumPy và SciPy nên để sử dụng sklearn, chúng ta bắt buộc phải có 2 packages này. Tất nhiên ta nên cài đặt matplotlib, một package không thể thiếu trong scientific plotting. Cuối cùng, chúng ta cần cài thêm package pandas, phục vụ cho data wrangling và analysis.

#### Thư viện Pandas:

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi.

*Tại sao lại dùng thư viện pandas?*

* + DataFrame đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục;
  + Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv, text, excel, sql database, hdf5;
  + Liên kết dữ liệu thông minh, xử lý được trường hợp dữ liệu bị thiếu. Tự động đưa dữ liệu lộn xộn về dạng có cấu trúc;
  + Dễ dàng thay đổi bố cục của dữ liệu;
  + Tích hợp cơ chế trượt, lập chỉ mục, lấy ra tập con từ tập dữ liệu lớn.
  + Có thể thêm, xóa các cột dữ liệu;
  + Tập hợp hoặc thay đổi dữ liệu với group by cho phép bạn thực hiện các toán tử trên tập dữ liệu;
  + Hiệu quả cao trong trộn và kết hợp các tập dữ liệu;
  + Lập chỉ mục theo các chiều của dữ liệu giúp thao tác giữa dữ liệu cao chiều và dữ liệu thấp chiều;
  + Tối ưu về hiệu năng;
  + Pandas được sử dụng rộng rãi trong cả học thuật và thương mại. Bao gồm thống kê, thương mại, phân tích, quảng cáo,…

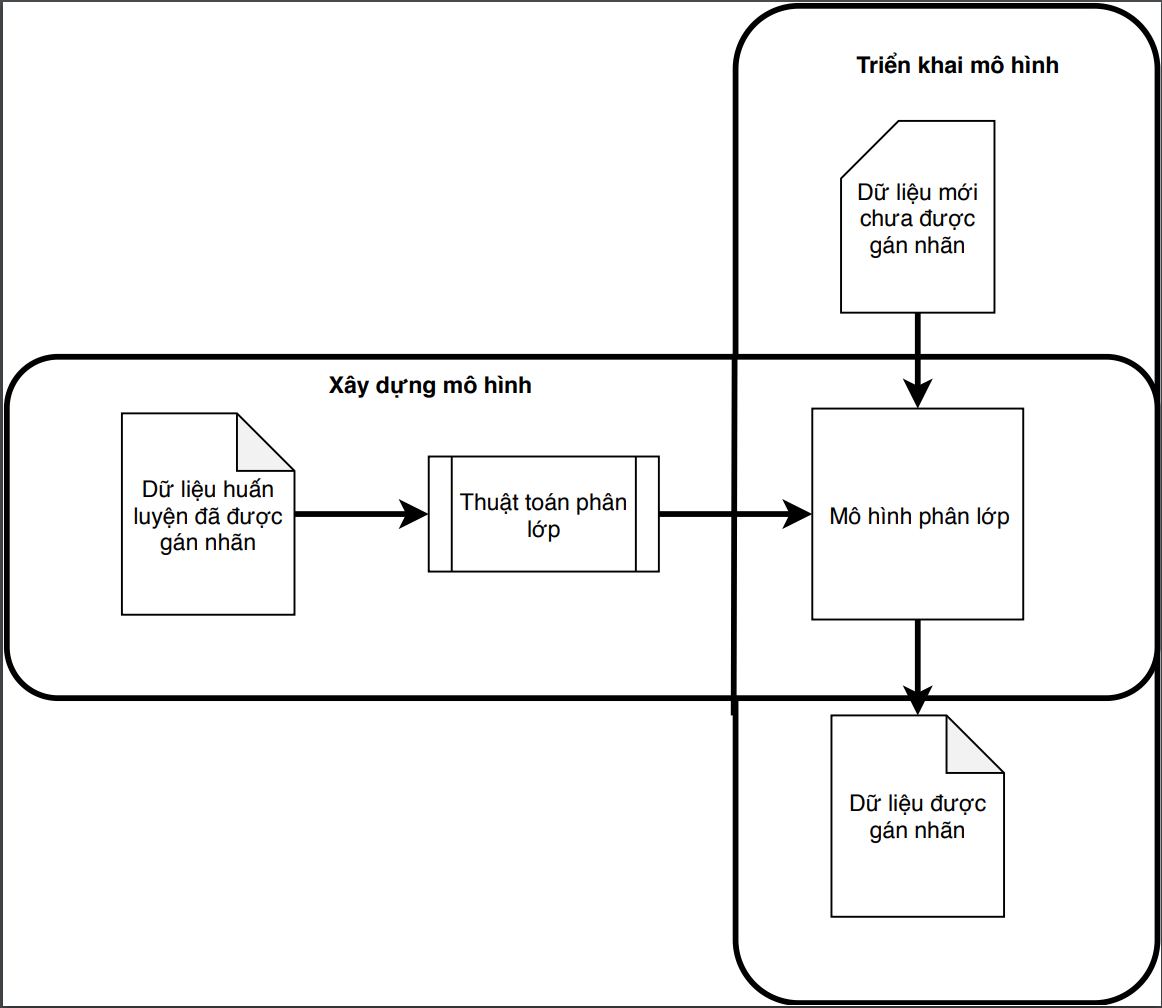
#### Thư viện Mathplotlb:

mathplotlib.pyplot là một tập hợp các hàm mà matplotlib có thể làm việc như MATLAB. Mỗi hàm pyplot tạo lên một sự thay đổi trong đồ thị, ví dụ như tạo một khung đồ thị, vẽ dồ thị trên khung đã tạo, hoặc vẽ nhiều đường biểu diễn trên cùng 1 đồ thị, dãn nhãn cho các trục,... Các trạng thái khác nhau trong matplotlib.pyplot lưu giữ các hàm đã gọi, qua đó lưu mọi thứ của đồ thị hiện tại và vùng đồ thị được vẽ, và hàm của đồ thị được vẽ trực tiếp trên trục tọa độ (axes ở đây mang nghĩa về đồ thị của hình được vẽ chứ không phải dịnh nghĩa nhiều trục trong toán học).

### Quy trình giải quyết bài toán:

Quá trình phân lớp dữ liệu thường gồm hai bước: Xây dựng mô hình (tạo bộ phân lớp) và triển khai mô hình đó để phân lớp dữ liệu chưa được gán.

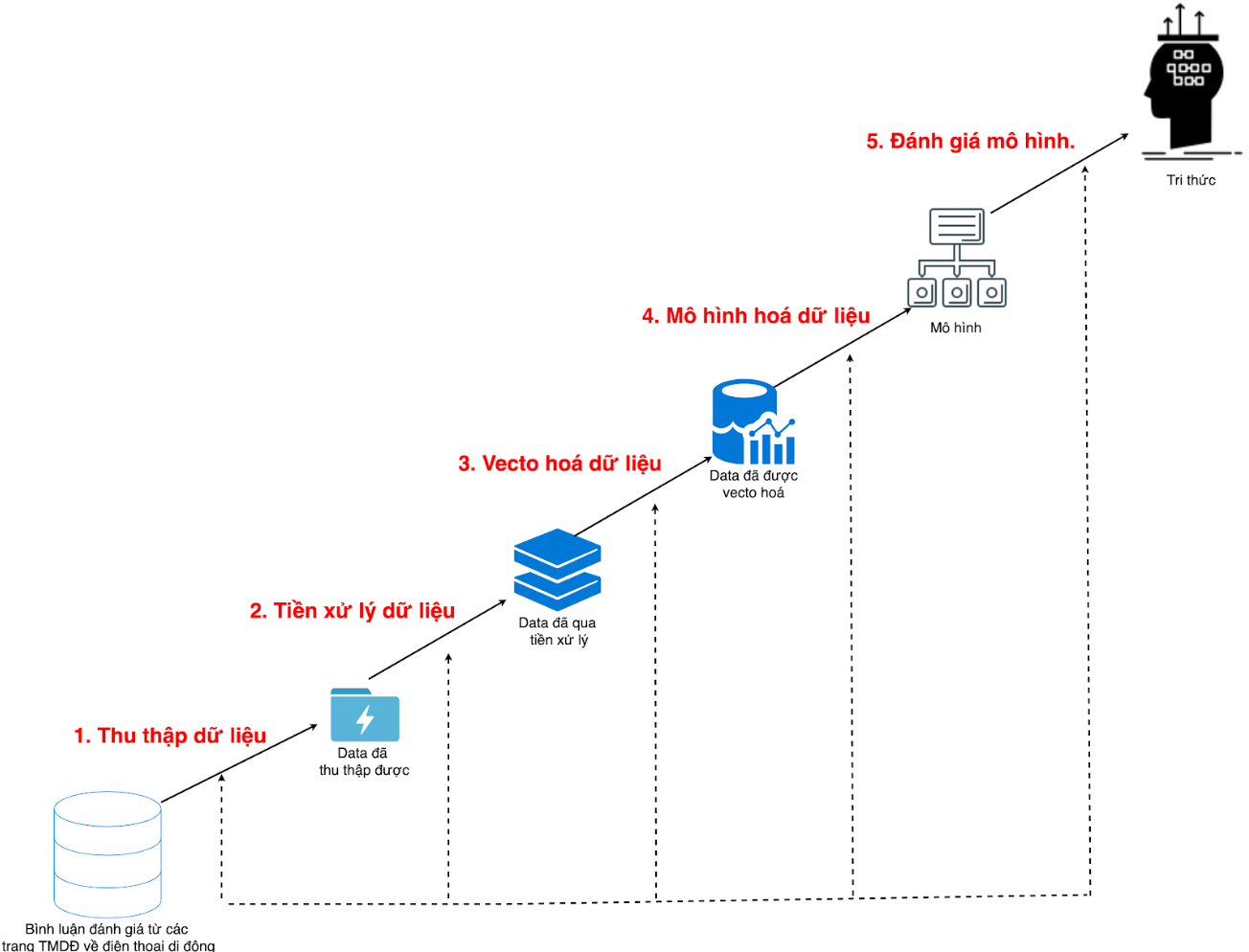
* Xây dựng mô hình: Đầu vào ở đây là dữ liệu đã được gán nhãn sẵn, đầu ra là một mô hình phân lớp đã được huấn luyện với độ chính xác cao. Ở đây bài toán của chúng ta sẽ xây dựng mô hình phân tích phản hồi của khách hàng về mặt hàng điện thoại di động.



Hình 2. : Quy trình xây dựng và phân tích phản hồi của khách hàng

* Triển khai mô hình: Đầu vào ở đây là dữ liệu mới chưa được gán nhãn, đầu ra là dữ liệu đã được dự đoán gán nhãn với độ chính xác cao. Đây là mục đích của bài toán chúng ta là dự đoán phản hồi của khách hàng là Positive hay Negative.

# CHƯƠNG 3 QUY TRÌNH XÂY DỰNG MÔ HÌNH PHÂN TÍCH PHẢN HỒI CỦA KHÁCH HÀNG VỀ MẶT HÀNG ĐIỆN THOẠI DI ĐỘNG



Hình 3. : Quy trình xây dựng mô hình phân tích phản hồi khách hàng.

Bước 1: Thu thập dữ liệu.

Bước 2: Tiền xử lý dữ liệu.

Bước 3: Vector hoá dữ liệu.

Bước 4: Mô hình hoá dữ liệu.

Bước 5: Đánh giá mô hình.

## Thu thập dữ liệu:

Để thực hiện xây dựng một mô hình phân lớp thì trước hết đầu tiên ta cần một bộ dữ liệu lớn, đối với bài toán này ta cần phải lấy được dữ liệu là các comment, bình luận của khách hàng từ các trang đánh giá điện thoại, các trang thương mai điện tử về mặt hàng điện thoại như thegioididong.com, fptshop.com.vn, tiki.vn …

### Giới thiệu công cụ Selenium Webdriver:

Selenium là bộ kiểm thử tự động miễn phí (Mã nguồn mở) tự động dành cho các ứng dụng web trên các trình duyệt và các nền tảng khác nhau.

Chủ yếu nó dùng để thực hiện automation test (Kiểm thử tự động), nhưng không giới hạn ở đó nó cho phép bạn mở một trình duyệt bất kì mà bạn chọn thực hiện các tác vụ như một con người như: click chuột, nhập thông tin đã có sẳn, tìm kiếm thông tin cụ thể có trên trang web.

Người dùng có thể dùng miễn phí

Kiến trúc đơn giản:

+ Điều khiển trình duyệt từ hệ điều hành.

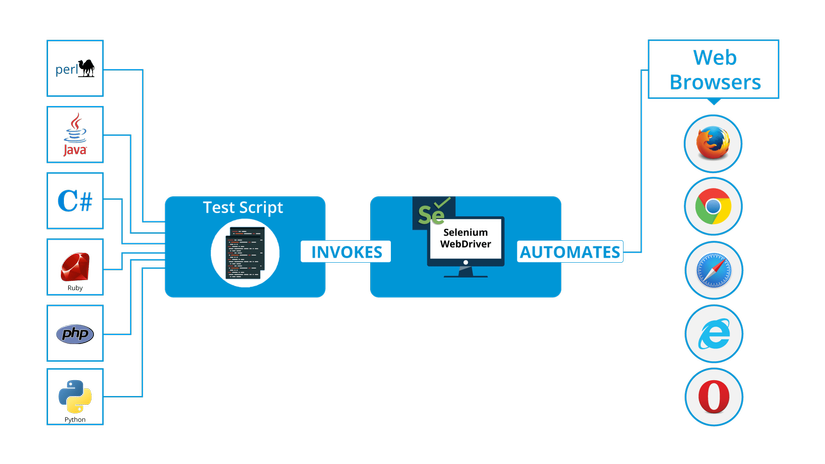
+ Bạn chỉ cần IDE và trình duyệt là xong.

Ngôn ngữ lập trình hỗ trợ: Web Driver hỗ trợ bạn viết kịch bản bằng các ngôn ngữ khác nhau như Java, .NET, PHP, Python, Perl, Ruby và bạn có thể sử dụng các điều kiện if else, các vòng lặp để tăng tính chính xác cho kịch bản.

Tốc độ: Khi so sánh với các công cụ khác của bộ Selenium, WebDriver là công cụ nhanh nhất trong số tất cả do tương tác trực tiếp từ hệ điều hành tới trình duyệt.



Selenium Webdriver được hỗ trợ trên các trình duyệt: Firefox, Google Chrome, Internet Explorer, Opera browser, Sarafi...



Hình 3. : Kiến trúc của Selenium Webdriver

Chính vì những tiện ích đó mà Selenium Webdriver mang lại, những người làm trong big-data, đào dữ liệu, cào website…có thể thu thập được dữ liệu một cách dễ dàng.

### Trực quan hoá dữ liệu đã thu thập được

Bằng việc sử dụng công cụ Selenium Webdriver để thực hiện việc cào dữ liệu thông qua hai trang web là thegioididong.com, fptshop.com.vn đã thu thập được gần 5000 bộ dữ liệu.

Trong một dòng dữ liệu bao gồm:

* Bình luận
* Tên hãng
* Điểm đánh giá
* Độ dài của bình luận
* Cảm xúc

Trong đó bình luận, tên hảng, điểm đánh giá là được cào trực tiếp từ website. Độ dài của bình luận được tính toán thông qua độ dài của chuỗi bình luận.

Đối với cột cảm xúc sẽ được tạo trong quá trình tiền xử lý dữ liệu. Cảm xúc thì được chia ra 2 loại là Positive (tích cực) và Negative (tiêu cực), đây là cột được gán nhãn cho dữ liệu. Cảm xúc được xác định dựa vào Điểm đánh giá.

Điểm đánh giá từ 1 đến 5 sao:

* **Từ 1 > 3 sao: Negative (tiêu cực)**

Ví dụ một số bình luận tiêu cực:

* Rate 1: “Pin nhanh hết kinh khủng. Sạc đầy 6h tối đế 9h tối còn 72% mặc dù rất ít xài, chỉ lướt face, zalo. Mình ko chơi game, máy ko tải 1 game nào thế mà mới Chỉ lướt face ko đụng game mà đôi khi còn bị đơ, zalo cũng bị đơ có khi ko gọi đc. Chế độ siêu tiết kiệm pin cũng như vậy, ko tiết kiệm đc gì nhiều cả chỉ thấy bất tiện hơn. Với tầm giá gần 9tr quả thật ko đáng, mới mua mà thiệt là chán nản.”
* Rate 2: “Thật sự thất vọng về sản phẩm Chiếc đầu tiên mình mua máy hoạt động thì bình thường nhưng nắp lưng lại ấn ọp ẹp chỗ sát đèn flash và góc máy. Mang sp den đổi cái mới nhưng về dùng mình vào Facebook xem live trym 1.2 phút lại bị loát hình chậm.đã so sánh cùng sản phẩm và cũng đã đổi nhìu mang wf mà vẫn bị. Đem máy đến kiểm tra thì cửa hàng bó tay. Thật sự thất vọng”
* Rate 3: “Sản phẩm đẹp.. Pin ngon.. Nhưng camera quá tệ.. Tệ nhất về phần xóa phông.. Thua các dòng máy rẻ tiền nữa”
* **Từ 4 > 5 sao: Positive (tích cực)**

Ví dụ một số bình luận tiêu cực:

* Rate 4: “Nói chung với cái giá hơn 20 triệu cho một chiếc với màn hình LCD thì chắc chắn có nhiều người e ngại có nên quyết định mua hay không, nhưng mà theo tôi đang sử dụng thì dòng XR này nó mặc dù chưa đạt đến chữ GIỎI nhưng nó lại hơn nhiều so với chữ KHÁ. Nếu ai đang phân vân thì theo tôi XR cũng đáng để dùng và nó cũng đáng với hai chữ trải nghiêm.”
* Rate 5: “Mua được 1 tháng, mình có đôi chút cảm nhận: Màn hình cực nét, tấm nền Samoled làm màn hình cực nịnh mắt Máy rất mượt, rất ít hiện tượng lag giật, đa nhiệm tốt Camera thì khỏi chê, ảo lung linh, vẫn hy vọng bản cập nhật tới samsung sẽ add thêm tính năng Bokeh như S9+ và A6+ Mình vẫn hy vọng pin máy sẽ cao hơn, ít ra là nhỉnh hơn 300 mAh như J7 prime, như bản khác là C8 cũng như J7+ pin cũng chỉ có 2800 mAh, nhưng cũng đủ dùng cho mình được 1 ngày rưỡi! Xài 4G onscreen 6 tiếng, wifi tầm 8 - 9 tiếng. 15% đầu tụt hơi nhanh nhưng phần trăm pin cuối tụt rất lâu. Để qua đêm tụt rất ít và có hôm không tụt tí nào. Thiết kế máy khá mỏng, kính cường lực chống trầy tốt, dễ lau sạch, loa hơi nhỏ một chút nhưng không vấn đề, sóng 4G, wifi mạnh. Như máy này có phiên bản 64GB và có máy màu hồng, sẽ tuyệt hơn nếu fpt có bán phiên bản này!!”

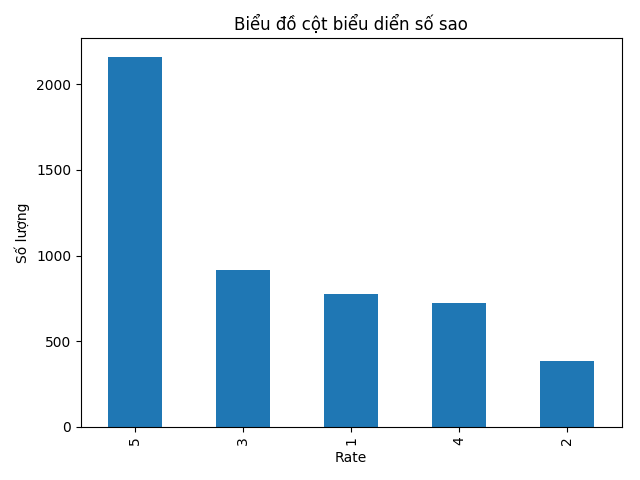


Hình 3. : Dữ liệu được thu thập từ công cụ Selenium webdriver.

Dữ liệu sau khi được cào về được lưu vào file có định dạng bằng excel:

Đối với Machine Learning, sau khi đã thu thập dữ liệu thì công việc tiếp theo là phải hiểu được dữ liệu. Vì vậy việc Visualization (trực quan hoá) dữ liệu là điều cần thiết. Trực quan hóa là một cách nhanh chóng và dễ dàng để truyền tải các khái niệm một cách phổ quát, đặc biệt là cho những người không quen thuộc với dữ liệu của bạn. Bất cứ khi nào làm việc với dữ liệu, trực quan thường là một phần cần thiết của phân tích.

Sau đây là một số biểu đồ trực quan hoá gần 5000 bộ dữ liệu đã được thu thập được sử dụng những bộ thư việc có sẵn của python (Matplot). Rút ra được một số đặc điểm của dữ liệu từ đó có thể hiểu được rõ dữ liệu đang có.



Hình 3. : Biểu đồ cột biểu diễn số sao của bình luận được crawl.

Dựa vào biểu đồ ta có thể thấy được:

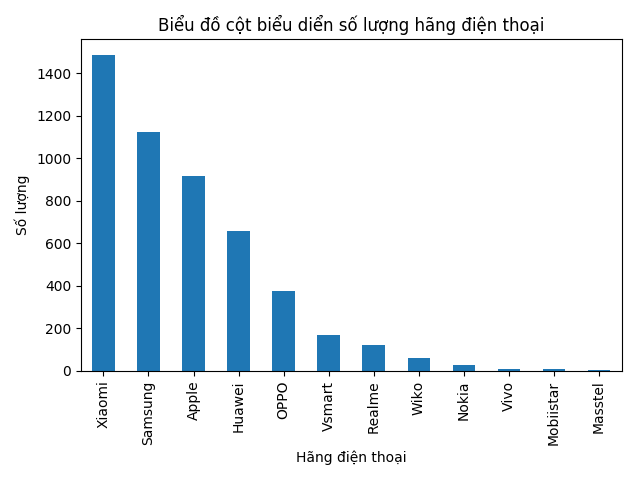
* Trong bộ dữ liệu của chúng ta thì 5 sao (>2000) có số lượng được đánh giá nhiều nhất tiếp đến là 3 và thấp nhất là 2 sao.(<500)



Hình 3. : Biểu đồ tròn thể hiện tỉ lệ phần trăm phân loại nhãn.

Dựa vào biểu đồ ta có thể thấy được:

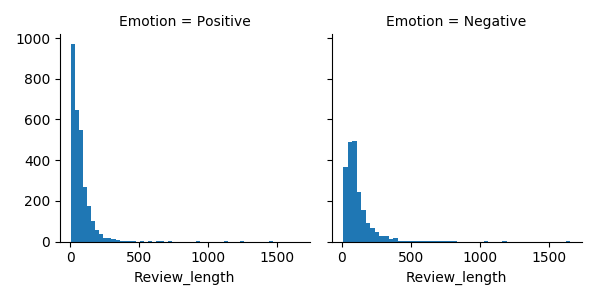
* Positve chiếm hơn 58% trong bộ dữ liệu.
* Negative chiếm gần 42% trong bộ dữ liệu.
* Điều này có nghĩa trong bộ dữ liệu, nhãn được gán cho dữ liệu gần bằng nhau tăng độ chính xác cho mô hình phân lớp.



Hình 3. : Biểu đồ cột biểu diễn số hãng điện thoại có trong bộ dữ liệu.

Dựa vào biểu đồ ta có thể thấy được:

* Hãng điện thoại Xiaomi có số lượng bình luận cao nhất trong bộ dữ liệu (>1400)
* Tiếp theo là Samsung (>1700) và Apple (>800)
* Hãng điện thoại có số lượng bình luận thấp nhất trong bộ dữ liệu là Masstel (<50)

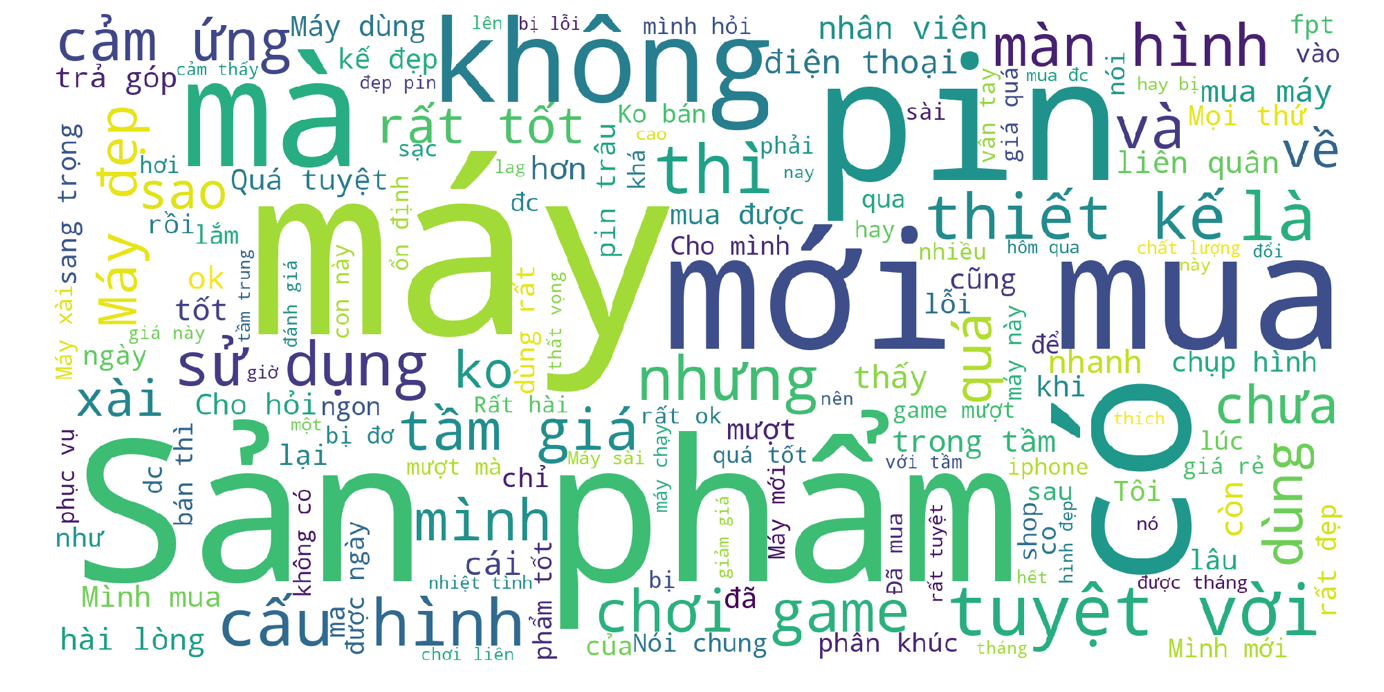


Hình 3. : Biểu đồ thể hiện số lượng trong từng bình luận của khách hàng.

Dựa vào biểu đồ trên:

* Nhìn tổng thể thì độ dài bình luận thiên về Negative dài hơn so với Positive. Điều này có thể gây ra một số vấn đề ảnh hưởng đến kết quả phân tích.

Bên cạnh một số thư viện giúp ta có thể vẽ được những biểu đồ trực quan hoá dữ liệu thì python còn có một thư viện rất tuyệt vời tên là Wordcloud:



Hình 3. : WordCloud từ ngữ có trong bộ dữ liệu được thu thập.

Wordcloud là một kỹ thuật trực quan hoá dữ liệu được sử dụng để thể hiện dữ liệu văn bản, trong đó kích thước của mỗi từ tương ứng với tần suất xuất hiện của chúng trong văn bản hoặc trong bộ dữ liệu. Wordcloud được sử dụng rộng rãi trong Machine learning đặc biệt là NLP.

Nhìn vào Wordcloud được xuất ra từ bộ dữ liệu cuả chúng ta có thể nhận thây rõ những từ phổ biến có trong bộ dữ liệu như: Sản, phẩm, máy, mới, mua, pin… Những từ ít phổ biến hơn như: như, hơi, ok…

**\*Như vậy thông qua việc sử dụng các thư viện có sẳn, chúng ta đã trực quan hoá bộ dữ liệu của mình giúp ta hiểu rõ hơn bộ dữ liệu đang có và có những bước điều chỉnh phù hợp nhầm tăng độ chính xác cho mô hình phân tích sau đó.**

## Tiền xử lý dữ liệu:

### Giới thiệu về tiền xử lý trong NLP:

Pre-processing – tiền xử lý dữ liệu, xử lý sơ bộ văn bản: xóa bỏ những kí tự, những mã điều khiển, những vùng không cần thiết cho hệ thống vì văn bản vốn dĩ được liệt kê mà không có cấu trúc, nên để nguyên vậy để xử lý là rất khó khăn.

### Các bước trong tiền xử lý dữ liệu:

Đối với bộ dữ liệu này chúng ta cần thực hiện:

* Làm sạch text
* Tách từ
* Loại bỏ Stopwords

#### Làm sạch text:

Mục đích bước này là loại bỏ noise trong data của bạn. Đa phần noise là các thẻ HTML, JavaScript, và đương nhiên nếu cứ để noise để tiến hành xử lý sẽ dẫn đến kết quả xử lý không tốt.

Thông thường chúng ta hay loại bỏ noise là các thẻ HTML và JS như trên tuy nhiên thực tế noise có thể không chỉ là HTML, JS, cũng có thể là những cụm từ không cần thiết, hay ký tự không có ý nghĩa ($%&##").

Tiếp đến là chuyển text về chữ viết thường, loại bỏ chữ số.

Kết quả sau qua trình làm sạch text:

*Ví dụ:*

*“<h1>Điện thoại quá tệ, không nên mua loại này<h2>”*

*“điện thoại quá tệ không nên mua loại này”*

#### Tách từ:

Thuật ngữ "tách từ" trong Tiếng Anh là "word segmentation", dịch ra Tiếng Việt là "tách từ" hoặc "phân đoạn từ".

Tách từ, về mặt biểu hiện, là gom nhóm các từ đơn liền kề thành một cụm từ có ý nghĩa. Ví dụ: "Cách tách từ cho Tiếng Việt." sau khi tách từ thì thành "Cách tách từ cho Tiếng\_Việt ." Về hình thức, các từ đơn được gom nhóm với nhau bằng cách nối với nhau bằng ký tự gạch dưới "\_", trong trường hợp này là từ Tiếng\_Việt. Sau khi thực hiện tách từ thì mỗi từ (token) trong câu được cách nhau bởi một khoảng trắng, trong trường hợp này như "Tiếng\_Việt ." thì từ "Tiếng\_Việt" cách đấu "." bởi 1 khoảng trắng. Đây là quy ước chung cho tất cả các ngôn ngữ của bài toán tách từ trong xử lý ngôn ngữ tự nhiên. Việc quy ước như vậy là để tạo thành chuẩn chung và để dễ xử lý hơn trong lập trình.

Tại sao phải tách từ cho Tiếng Việt?

Về mặt ngữ nghĩa, việc tách từ văn bản đầu vào trước khi đưa vào huấn luyện mô hình máy học là để giải quyết các bài toán liên quan đến ngữ nghĩa của văn bản, tức là kết quả đầu ra mang tính suy luận dựa trên việc hiểu ý nghĩa của văn bản đầu vào. Ví dụ như các dạng bài toán: phát hiện đạo văn, tóm tắt văn bản, hỏi đáp tự động, hỗ trợ khách hàng tự động, phân tích cảm xúc văn bản, dịch máy, trợ lý ảo...

Mục tiêu của việc tách từ văn bản đầu vào là để khử tính nhập nhằng về ngữ nghĩa của văn bản. Tùy vào từng loại ngôn ngữ có những đặc điểm khác nhau mà việc tách từ văn bản cũng có độ khó khăn khác nhau. Đựa theo đặc điểm của ngôn ngữ tự nhiên mà ngôn ngữ được phân thành các loại:

Ngôn ngữ hòa kết (flexional), ví dụ: Đức, Latin, Hi Lạp, Anh, Nga...

* Ngôn ngữ chắp dính (agglutinate), ví dụ: Thổ Nhĩ Kỳ, Mông Cổ, Nhật Bản, Triều Tiên,...
* Ngôn ngữ đơn lập (isolate), là ngôn ngữ phi hình thái, không biến hình, đơn âm tiết, ví dụ: Việt Nam, Hán,...
* Với ngôn ngữ hòa kết như Tiếng Anh, thì việc tách từ khá đơn giản vì ranh giới từ được nhận diện bằng khoảng trắng và dấu câu.

Với ngôn ngữ Tiếng Việt, thuộc loại hình đơn lập, mang đặc điểm là từ Tiếng Việt không biến đổi hình thái, ranh giới từ không được xác định mặc nhiên bằng khoản trắng. Tiếng Việt có đặc điểm là ý nghĩa ngữ pháp nằm ở ngoài từ, phương thức ngữ pháp chủ yếu là trật tự từ và từ hư. Cho nên có trường hợp một câu có thể có nhiều ngữ nghĩa khác nhau tuỳ vào cách ta tách từ như thế nào, gây nhập nhằng về ngữ nghĩa của câu. Ví dụ:

* Với câu "Xoài phun thuốc sâu không ăn." có thể được tách từ như sau, với ý nghĩa hoàn toàn khác nhau:

*Xoài / phun thuốc / sâu / không / ăn.*

*Xoài / phun / thuốc sâu / không / ăn.*

* Với câu "Ăn cơm không được uống rượu.", có thể được tách từ như sau:

*Ăn / cơm / không / được / uống / rượu.*

*Ăn / cơm không / được / uống / rượu.*

* Với câu "Mẹ vào ca ba con ngủ với dì.", có thể được tách từ như sau:

*Mẹ / vào / ca ba / con / ngủ / với / dì.*

*Mẹ / vào ca / ba con / ngủ / với / dì.*

Điều đó cho thấy, công việc tách từ trong Tiếng Việt không phải là chuyện dễ dàng, vì nó tạo ra các câu có ngữ nghĩa hoàn toàn khác nhau, gây ảnh hưởng đến chất lượng huấn luyện mô hình học. Chính vì vậy, công việc tách từ là rất quan trọng đối với xử lý ngôn ngữ Tiếng Việt, nhất là khi giải quyết các bài toán liên quan đến ngữ nghĩa của văn bản.

Để thực hiện tách từ cho bộ dữ liệu cuả chúng ta, chúng ta có thể dùng công cụ vnTokenizer của tác giả Lê Hồng Phương. Bộ công cụ này hỗ trợ cho nhiều ngôn ngữ lập trình trong đó có Python.

Kết quả sau khi tách từ:

*“sản phẩm dùng thật tuyệt vời, sang trọng, đẳng cấp và rất mượt khi chơi game”*

*“sản\_phẩm dùng thật tuyệt\_vời sang\_trọng đẳng\_cấp và rất mượt khi chơi game”*

#### Loại bỏ Stopwords:

StopWords là những từ xuất hiện nhiều trong ngôn ngữ tự nhiên, tuy nhiên lại không mang nhiều ý nghĩa. Ở tiếng việt StopWords là những từ như: để, này, kia... Tiếng anh là những từ như: is, that, this...

Có rất nhiều cách để loại bỏ StopWords nhưng có 2 cách chính là:

• Dùng từ điển

• Dựa theo tần suất xuất hiện của từ

**Dùng từ điển:**

Cách này đơn giản nhất, chúng ta tiến hành filter văn bản, loại bỏ những từ xuất hiện trong từ điển StopWords:

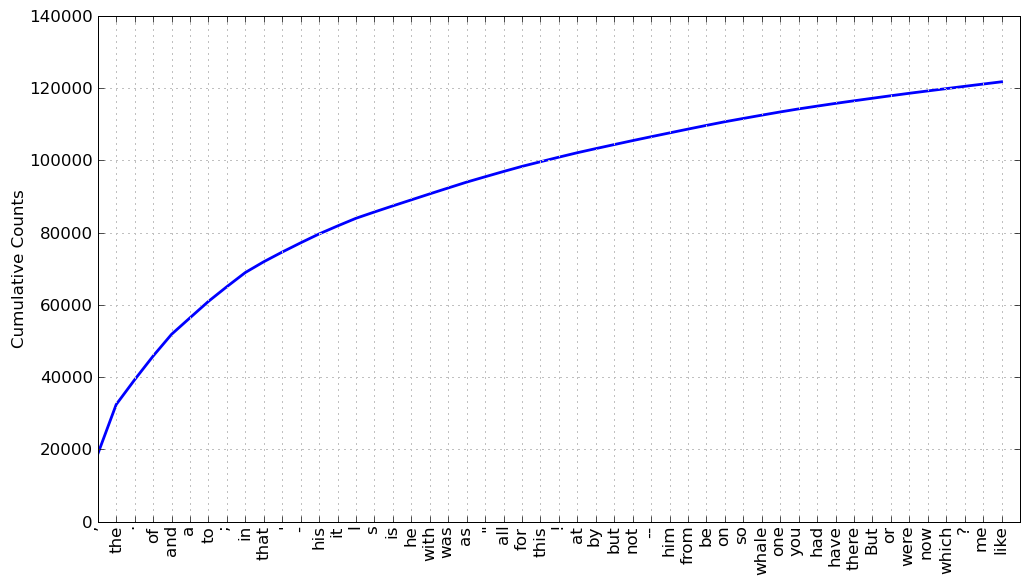
['bị', 'bởi', 'cả', 'các', 'cái', 'cần', 'càng', 'chỉ', 'chiếc', 'cho', 'chứ', 'chưa', 'chuyện', 'có', 'có\_thể', 'cứ', 'của', 'cùng', 'cũng', 'đã', 'đang', 'đây', 'để', 'đến\_nỗi', 'đều', 'điều', 'do', 'đó', 'được', 'dưới', 'gì', 'khi', 'không', 'là', 'lại', 'lên', 'lúc', 'mà', 'mỗi', 'một\_cách', 'này', 'nên', 'nếu', 'ngay', 'nhiều', 'như', 'nhưng', 'những', 'nơi', 'nữa', 'phải', 'qua', 'ra', 'rằng', 'rằng', 'rất', 'rất', 'rồi', 'sau', 'sẽ', 'so', 'sự', 'tại', 'theo', 'thì', 'trên', 'trước', 'từ', 'từng', 'và', 'vẫn', 'vào', 'vậy', 'vì', 'việc', 'với', 'vừa']

**Dựa theo tần suất xuất hiện của từ:**

Với cách này, chúng ta tiến hành đếm số lần xuất hiện của từng từ trong data sau đó sẽ loại bỏ những từ xuất hiện nhiều lần (cũng có thể là ít lần). Khoa học đã chứng minh những từ xuất hiện nhiều nhất thường là những từ không mang nhiều ý nghĩa

.

Trên là top 50 từ xuất hiện nhiều nhất trong mỗi cuốn sách, dễ dàng nhận thấy chúng không mang nhiều ý nghĩa. Chính vì thế chúng ta sẽ loại bỏ những từ như thế này.



Hình 3. : Biểu đồ thể hiện top 50 từ xuất hiện nhiều nhất trong mỗi cuốn sách tiếng anh.

**Để thực việc loại bỏ Stopwords cho bộ dữ liệu của chúng ta thì mình đã chọn cách đơn giản là dùng bộ từ điển Stopwords có sẵn.**

Kết quả sau khi đã tách từ:

*“sản\_phẩm dùng thật tuyệt\_vời sang\_trọng đẳng\_cấp* ***và rất*** *mượt* ***khi*** *chơi game”*

*“sản\_phẩm dùng thật tuyệt\_vời sang\_trọng đẳng\_cấp mượt chơi game”*

**\*Như vậy sau khi thực hiện đủ các quá trình tiền xử lý dữ liệu thì dữ liệu đã có thể sẵn sàng vector hoá.**

**Kết quả sau khi tiền xử lý dữ liệu:**

*“Sản phẩm dùng thật tuyệt vời, sang trọng, đẳng cấp và rất mượt khi chơi game”*

*“sản\_phẩm dùng thật tuyệt\_vời sang\_trọng đẳng\_cấp mượt chơi game”*

## Vector hóa dữ liệu:

### Giới thiệu về mô hình không gian vector (Vector space model):

Nói một cách ngắn gọn, Vector space model (Mô hình không gian vector) là một mô hình đại số (algebraic model) thể hiện thông tin văn bản như một vector, các phần tử của vector này thể hiện mức độ quan trọng của một từ và cả sự xuất hiện hay không xuất hiện (Bag of words) của nó trong một tài liệu.

Mô hình này biểu diễn văn bản như những điểm trong không gian Euclid n-chiều, mỗi chiều tương ứng với một từ trong tập hợp các từ. Phần tử thứ i, là di của vector văn bản cho biết số lần mà từ thứ i xuất hiện trong văn bản. Sự tương đồng của hai văn bản được định nghĩa là khoảng cách giữa các điểm, hoặc là góc giữa những vector trong không gian.

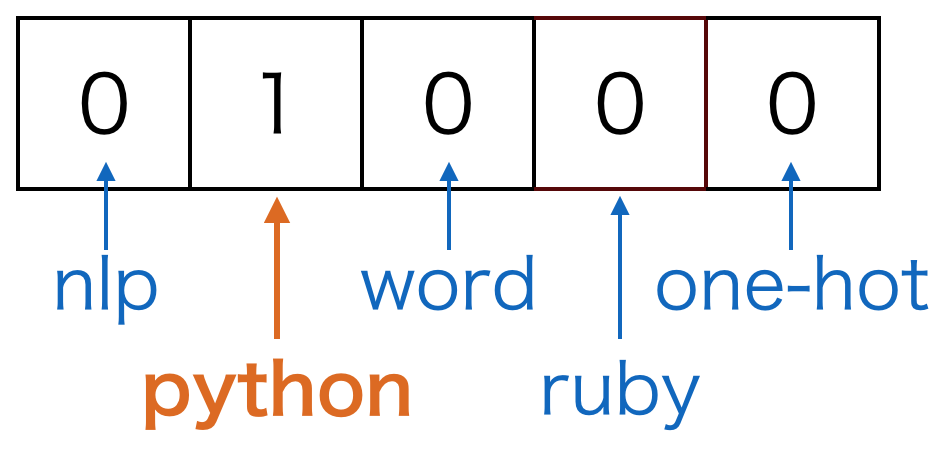
Mặc dù đơn giản, nhưng mô hình không gian vector và những biến thể của nó hiện nay vẫn là cách phổ biến để biểu diễn văn bản trong Data mining và Information retrieval. Tuy nhiên, một trong những điểm yếu của vector space model số chiều lớn (high-dimensonal), có khoảng cỡ chục triệu chiều trong không gian vector nếu như chúng ta áp dụng nó vào web search engine.

Phương pháp vector hoá thường có 2 dạng:

* Sử dụng one-hot
* Biểu thị sự phân tán

**Sử dụng one-hot:**

Giả sử ta có danh sách các từ sau: (nlp, python, word, ruby, one-hot) khi vector hoá từ python, ta sẽ được: [0;1;0;0;0]



Hình 3. : Vector được biểu diễn dưới dạng one hot.

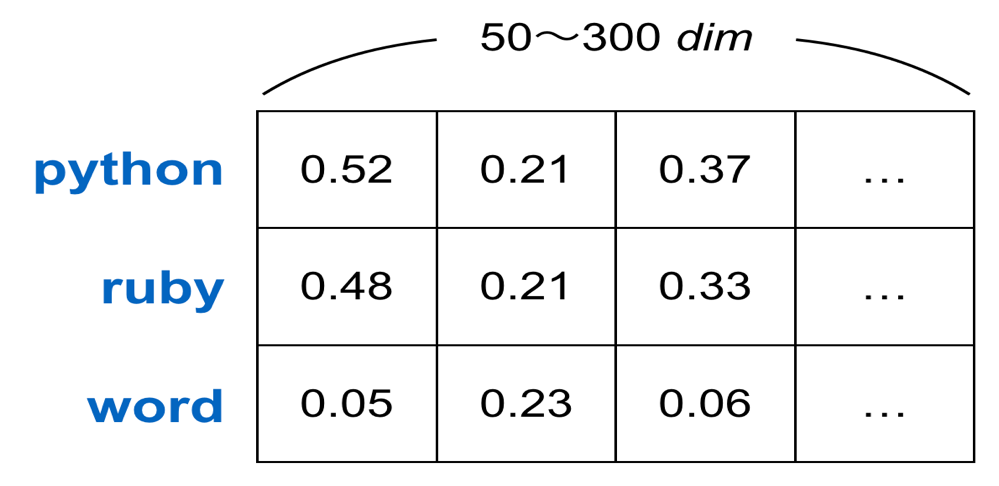
Tương tự với nlp thì vị trí nlp sẽ là 1, còn lại là 0, cứ như vậy ta sẽ biểu diễn được tất cả các từ. Lấy luôn hình trên làm ví dụ ta sẽ được:

* nlp: [1;0;0;0;0]
* python: [0;1;0;0;0]
* word: [0;0;1;0;0]
* ruby: [0;0;0;1;0]
* one-hot: [0;0;0;0;1]

**Biểu thị sự phân tán:**

Phân tán ở đây nghĩa là tần suất phân bố, xuất hiện của từ đó trên mỗi chủ đề(topic), hoặc là mỗi đoạn văn bản khác nhau. Thực tế thông thường số chiều khoảng 50-500.

Việc phân loại từ như trên rất quan trọng trong NLP, ví dụ như bài toán Text Classification, bạn cần phân biệt, tìm ra các từ cùng chủ đề. Hay bài toán Intent detection, bạn cũng cần tìm ra mối quan hệ giữa input và training data.



Hình 3. :Vector biểu diễn dưới dạng sự phân tán.

Như ví dụ sau:

Như trên ví dụ số chủ để khoảng 50-300 chủ đề, khi biểu diễn theo kiểu này, dễ dàng nhận thấy vấn đề của one-hot đã được giải quyết, khi tính tích vô hướng sẽ nhận thấy ngay python và ruby có mối liên quan rất lớn, ngược lại so với word không có nhiều mối tương đồng.

### Giới thiệu về biểu diễn vector bằng TF-IDF:

Viết tắt của thuật ngữ tiếng Anh term frequency – inverse document frequency,tf-idf là trọng số của một từ trong văn bản thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản.

Mỗi từ trong không gian vector sẽ có một trọng số, có nhiều phương pháp xếp hạng khác nhau, nhưng tf-idf (term frequency–inverse document frequency) là một phương pháp phổ biến để đánh giá và xếp hạng một từ trong một tài liệu. MySQL fulltext search cũng sử dụng phương pháp này. Về cơ bản thì tf-idf là một kỹ thuật (cụ thể là ranking function) giúp chuyển đổi thông tin dưới dạng văn bản thành một Vector space model thông qua các trọng số. Vector space model và tf-idf được phát triển bởi Gerard Salton vào đầu thập niên 1960s.

Thuật toán này thường được sử dụng vì: trong ngôn ngữ luôn có những từ xảy ra thường xuyên với các từ khác. Có nghĩa là luôn có một tập các từ mà tần số xuất hiện, sử dụng nhiều hơn các từ khác , điều này đúng trong bất kì ngôn ngữ nào.

**Tf - Term frequency :** là tần suất xuất hiện của một từ trong một đoạn văn bản. Với những đoạn văn bản có độ dài khác nhau, sẽ có những từ xuất hiện nhiều ở những đoạn văn bản dài thay vì những đoạn văn bản ngắn. Vì thế, tần suất này thường được chia cho độ dài của đoạn văn bản như một phương thức chuẩn hóa (normalization). TF được tính bởi công thức:

/T

Trong đó : là một từ trong đoạn văn bản.

: là tần suất xuất hiện của trong đoạn văn bản .

T : là tổng số từ trong đoạn văn bản đó.

**IDF - Inverse Document Frequency:** dùng để ước lượng mức độ quan trọng của từ đó như thế nào . Khi tính tần số xuất hiện tf thì các từ đều được coi là quan trọng như nhau. Tuy nhiên có một số từ thường được được sử dụng nhiều nhưng không quan trọng để thể hiện ý nghĩa của đoạn văn ,

Ví dụ:

* Từ nối: và, nhưng, tuy nhiên, vì thế, vì vậy, …
* Giới từ: ở, trong, trên, …
* Từ chỉ định: ấy, đó, nhỉ, …

IDF được tính bởi công thức:

Trong đó: N là tổng số đoạn văn bản

Tập mẫu là số văn bản chứa

Và cuối cùng **TF-IDF** được tính bởi:

Ví dụ : Một bình luận có 100 từ chứa từ “Tốt” 3 lần thì . Giả sử có 10000 bình luận mà từ “Tốt” xuất hiện trong 1000 bình luận thì

Trọng số tf.idf tính theo cách trên sẽ cho biết độ quan trọng của một từ trong văn bản. Ví dụ trong một văn bản thể thao, hai từ “Pin trâu” và “máy” xuất hiện nhiều thì sẽ có điểm tf cao. Tuy nhiên nếu xét trên tập tất cả các bình luận, từ “máy” hầu như xuất hiện ở tất cả các tài liệu và vì vậy sẽ có điểm idf rất thấp (gần như bằng 0). Ngược lại từ “Pin trâu” hầu như chỉ xuất hiện trong các bình luận về tích cực nên sẽ có điểm idf khá cao. Vì vậy trọng số tf.idf của từ “Pin tốt” sẽ cao, biểu hiện rằng đây là từ quan trọng trong văn bản, còn trọng số tương ứng của từ “máy” rất thấp, cho thấy đây là một từ kém quan trọng.

Triển khai **TF-IDF** cho bộ dữ liệu của chúng ta: Trong thư viện Sklearn trong đó có funtion TfidfVectorizer sẽ giúp ta thực hiện TF-IDF một cách dễ dàng:

**from** sklearn.feature\_extraction.text **import** TfidfVectorizer  
tf=TfidfVectorizer(min\_df=5,max\_df=0.8,max\_features=3000,sublinear\_tf=**True**)

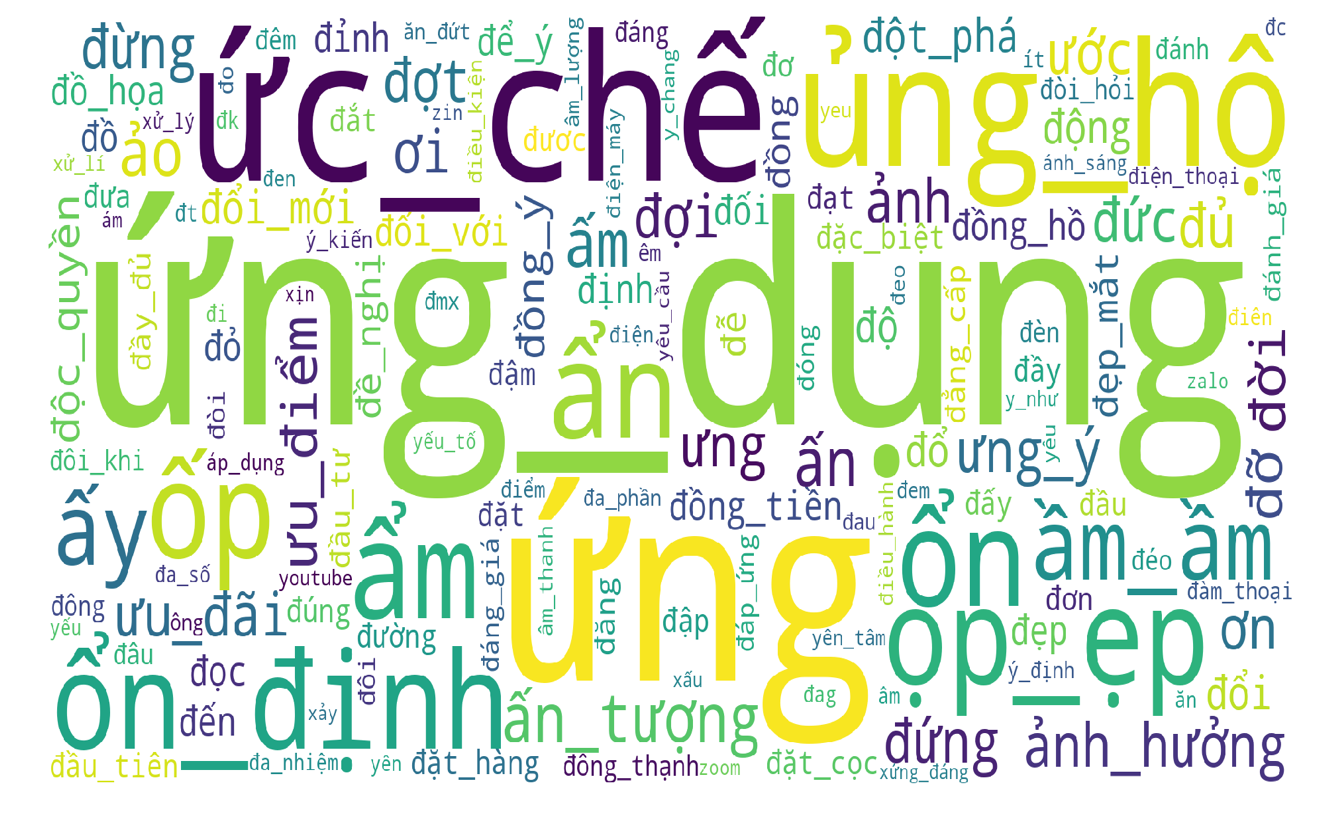
tf.fit(data[**'Review'**].values.astype(**'U'**))

X = tf.transform(data[**'Review'**].values.astype(**'U'**))  
y = np.array(data[**'Emotion'**]).flatten()

Hàm TfidfVectorizer có các tham số chúng ta cần chú ý là:

* min\_df : loại bỏ những từ nào từ có tần suất suất hiện nhỏ hơn min\_df ( tính theo số đếm)
* max\_df : loại bỏ những từ nào từ có tần suất xuất hiện lớn hơn max\_df ( tính theo %)
* sublinear\_tf: Scale term frequency bằng logarithmic scale
* stop\_words loại bỏ stop word, chúng ta đã làm trước đó nên không cần tham số này
* max\_features lựa chọn số character vào vocabulary
* vocabulary nếu chúng ta đã xây dựng vocabulury trước đó thì không cần max\_features

Sau khi vector ta có bộ từ điển gồm 1517 từ có trong bộ dữ liệu, ta có word cloud của bộ dữ liệu sau quá trình tiền dữ liệu được lấy ra từ Vocabulary của Tf-Idf : Một số từ xuất hiện khá nhiều như ứng dụng, ức chế, ủng hộ, ọp ẹp, ổn định, ẩn…



Hình 3. : Wordcloud của bộ từ được lấy ra từ Vocabulary của Tf-Idf.

## Mô hình hóa dữ liệu:

### Giới thiệu một số thuật toán:

Đối với bài toán phân lớp học có giám sát thì có nhiều thuật toán có thể áp dụng để modeling dữ liệu như Logistic Regression, Naive Bayes, Multi-layer Perceptron, Decision Tree, K-Nearest Neighbor…

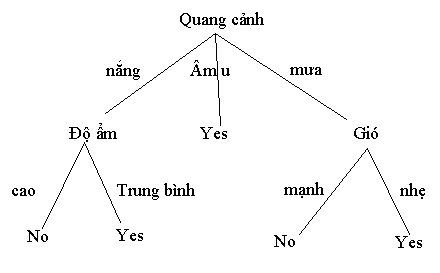
Tuỳ vào từng bài toán, từng bộ dữ liệu thì mỗi mô hình thuật toán trên có các đặc điểm hiệu suất khác nhau. Không có một thuật toán nhất định nào là được cho là thích hợp với tất cả cả bài toán các bộ dữ liệu. Vì sao lại như vậy, vì mỗi bộ dữ liệu có các đặc điểm khác nhau như: độ lớn, giá trị số, là vector…

Vì vậy để chọn một thuật toán phù hợp, cho ra một mô hình phân lớp có độ chính xác cao thì ta sẽ cho bộ dữ liệu chạy qua một số thuật toán rồi lựa chọn thuật toán thích hợp nhất để xây dựng mô hình.

#### Cây quyết định (Decision Trees)

Cây quyết định là công cụ hỗ trợ quyết định sử dụng biểu đồ dạng cây hoặc mô hình của các quyết định và kết quả có thể xảy ra của chúng, bao gồm kết quả sự kiện ngẫu nhiên, chi phí tài nguyên và lợi ích.

Dưới đây là một ví dụ điển hình của cây quyết định:



Cây quyết định này cho ta gợi ý về việc có đi đá bóng hay không. Ví dụ, quang cảnh có nắng, độ ẩm trung bình thì tôi sẽ đi đá bóng. Ngược lại, nếu trời mưa, gió mạnh thì tôi sẽ không đi đá bóng nữa.

Cây quyết định tuy là mô hình khá cũ, khá đơn giản những vẫn còn được ứng dụng khá nhiều và hiệu quả. Đứng dưới góc nhìn thực tế, cây quyết định là một danh sách tối thiểu các câu hỏi dạng yes/no mà người ta phải hỏi, để đánh giá xác suất đưa ra quyết định đúng đắn.

#### Phân loại Bayes (Naïve Bayes Classification)

Phân loại Bayes là một nhóm các phân loại xác suất đơn giản dựa trên việc áp dụng định lý Bayes với các giả định độc lập (naïve) giữa các đặc tính.

Trong đó: P(A|B) là xác suất có điều kiện A khi biết B, P(A) là xác suất giả thuyết A (tri thức có được về giải thuyết A trước khi có dữ liệu B), P(B|A) là xác suất có điều kiện B khi biết giả thuyết A,P(B) là xác suất của dữ liệu quan sát B không quan tâm đến bất kỳ giả thuyết A nào.

Thuật toán này được áp dụng trong một số bài toán như:

* Đánh dấu một email là spam hay không.
* Phân loại bài viết tin tức thuộc lĩnh vực công nghệ, chính trị hay thể thao.
* Kiểm tra một đoạn văn bản mang cảm xúc tích cực hay tiêu cực.
* Sử dụng cho các phần mềm nhận diện khuôn mặt. …

#### Hồi quy logistic (Logistic Regression)

Hồi quy logistic là một cách thống kê mạnh mẽ để mô hình hóa một kết quả nhị thức với một hoặc nhiều biến giải thích. Nó đo lường mối quan hệ giữa biến phụ thuộc phân loại và một hoặc nhiều biến độc lập bằng cách ước tính xác suất sử dụng một hàm logistic, là sự phân bố tích lũy logistic.

Thuật toán này được sử dụng trong một số trường hợp:

* Điểm tín dụng ( quyết định có cho khách hàng vay vốn hay không)
* Đo mức độ thành công của chiến dịch marketing
* Dự đoán doanh thu của một sản phẩm nhất định
* Dự đoán động đất ….

#### Neural network:

Mạng nơ-ron nhân tạo (Neural Network - NN) là một mô hình lập trình rất đẹp lấy cảm hứng từ mạng nơ-ron thần kinh. Kết hợp với các kĩ thuật học sâu (Deep Learning - DL), NN đang trở thành một công cụ rất mạnh mẽ mang lại hiệu quả tốt nhất cho nhiều bài toán khó như nhận dạng ảnh, giọng nói hay xử lý ngôn ngữ tự nhiên.

Lấy cảm hứng từ mạng nơ-ron sinh học, mạng NN được hình thành từ các tầng nơ-ron nhân tạo. Mạng NN gồm 3 kiểu tầng chính là tầng vào (input layer) biểu diễn cho đầu vào, tầng ra (output layer) biểu diễn cho kết quả đầu ra và tầng ẩn (hidden layer) thể hiện cho các bước suy luận trung gian. Mỗi nơ-ron sẽ nhận tất cả đầu vào từ các nơ-ron ở tầng trước đó và sử dụng một hàm kích hoạt dạng (activation function) phi tuyến như sigmoid, ReLU, tanh để tính toán đầu ra.

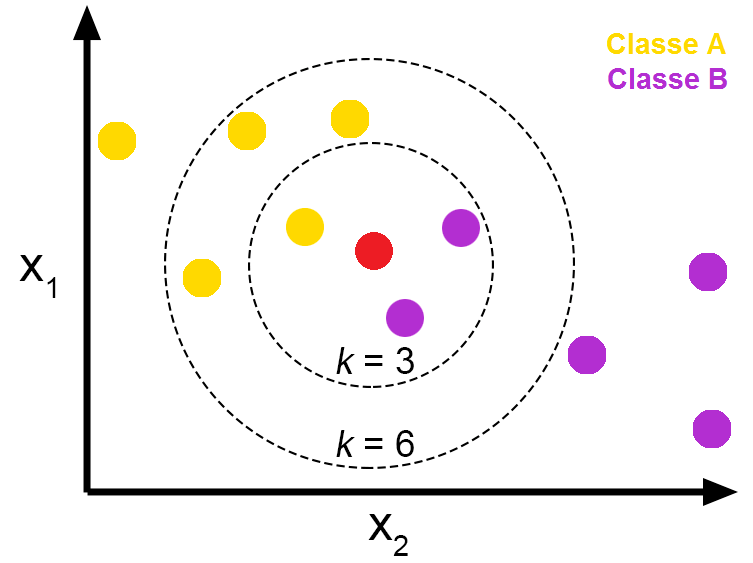
#### K-Nearest Neighbors

K-Nearest Neighbors algorithm (K-NN) được sử dụng rất phổ biến trong lĩnh vực Data Mining. K-NN là phương pháp để phân lớp các đối tượng dựa vào khoảng cách gần nhất giữa đối tượng cần xếp lớp (Query point) và tất cả các đối tượng trong Training Data.

Một đối tượng được phân lớp dựa vào K láng giềng của nó. K là số nguyên dương được xác định trước khi thực hiện thuật toán. Người ta thường dùng khoảng cách Euclidean để tính khoảng cách giữa các đối tượng.

Thuật toán K-NN được mô tả như sau:

* + Xác định giá trị tham số K (số láng giềng gần nhất)
  + Tính khoảng cách giữa đối tượng cần phân lớp (Query Point) với tất cả các đối tượng trong training data (thường sử dụng khoảng các Euclidean)
  + Sắp xếp khoảng cách theo thứ tự tăng dần và xác định K láng giềng gần nhất với Query Point
  + Lấy tất cả các lớp của K láng giềng gần nhất đã xác định
  + Dựa vào phần lớn lớp của láng giềng gần nhất để xác định lớp cho Query Point



Hình 3. : Mô hình K-NN

### Lựa chọn thuật toán cho bài toán:

Để đánh giá một hệ thống phân lớp, cách đơn giản nhất và hay được sử dụng nhất là **Accuracy (độ chính xác)**. Cách đánh giá này đơn giản là tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử.

Hiệu năng của một mô hình thường được đánh giá dựa trên tập dữ liệu kiểm thử (test data). Cụ thể, giả sử đầu ra của mô hình khi đầu vào là tập kiểm thử được mô tả bởi vector y\_pred - là vector dự đoán đầu ra với mỗi phần tử là class được dự đoán của một điểm dữ liệu trong tập kiểm thử. Ta cần so sánh giữa vector dự đoán y\_pred này với vector class thật của dữ liệu, được mô tả bởi vector y\_test.

Chia bộ dữ liệu thành hai phần là training (huấn luyện) và testing (kiểm thử), ta sẽ dùng hàm train\_test\_split của sklearn để chia dữ liệu:

**From** sklearn.model\_selection **import** train\_test\_split  
X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=7,shuffle=**True**)

Trong đó: X\_train là dữ liệu huấn luyện

y\_train là nhãn huấn luyện

X\_test là dữ liệu kiểm thử

y\_test là nhãn kiểm thử

Test\_size = 0.2 tức là bộ dữ liệu kiểm thử (X\_test , y\_test) sử dụng 20% bộ dữ liệu (gần bằng 1000 dòng) => bộ dữ liệu huấn luyện (X\_train, y\_train) sử dụng 80% (gần bằng 4000 dòng)

Tiến hành đưa dữ liệu vào các một số model thuật toán để tính Accuracy (độ chính xác) để so sánh.

**from** sklearn.linear\_model **import** LogisticRegression  
**from** sklearn.tree **import** DecisionTreeClassifier  
**from** sklearn.neighbors **import** KNeighborsClassifier  
**from** sklearn.naive\_bayes **import** MultinomialNB  
**from** sklearn.neural\_network **import** MLPClassifier

names=[]  
models=[]

models.append((**'Logistic Regression'**, LogisticRegression()))  
models.append((**'Multinomial Naive Bayes '**, MultinomialNB()))  
models.append((**'Neural network'**, MLPClassifier()))  
models.append((**'Decision Tree'**, DecisionTreeClassifier()))  
models.append((**'K-Nearest Neighbor'**, KNeighborsClassifier()))  
  
**for** name, model **in** models:  
 names.append(name)  
 model.fit(X\_train,y\_train)  
 y\_pred=model.predict(X\_test)  
 acc=accuracy\_score(y\_test,y\_pred)  
 print(**"Độ chính xác {} là {}%"**.format(name, round(acc\*100,2)))

Chạy chương trình cho ra kết quả:

* Độ chính xác Logistic Regression là 84.76%
* Độ chính xác Multinomial Naive Bayes là 83.75%
* Độ chính xác Neural network là 79.82%
* Độ chính xác Decision Tree là 78.71%
* Độ chính xác K-Nearest Neighbor là 62.87%

Theo như kết quả ta thấy có sự chênh lệch về Accuracy của từng thuật toán trong đó Logistic Regression và Naive Bayes có độ chính xác cao nhất và không chênh nhau bao nhiêu. Ở đây chúng ta sẽ lựa chọn **Logistic Regression** làm thuật toán để xây dựng mô hình của chúng ta.

### Giới thiệu về Logistic Regression:

#### Tính chất của Logistic Regression:

**Mặc dù có tên là Regression, tức một mô hình cho fitting, Logistic Regression lại được sử dụng nhiều trong các bài toán Classification.**

Lấy ví dụ bài toán của chúng ta: phân tích phản hồi của khách hàng thành Positive và Negative.

Gọi biến phụ thuộc là y (y=1: Positive, y=0: Negative)

Dự báo P(y=1)

Vì vậy P(y=0) = 1 - P(y=1)

Hàm Logisitic:

P(y=1)=

Trong đó:

β1, β2,…,βk: Hệ số hồi quy của các yếu tố đầu vào (hay còn gọi là biến độc lập).

x1, x2,…,xk: Hệ số hồi quy cho biết độ mạnh cũng như chiều của sự ảnh hưởng của các yếu tố dẫn đến xác suất xảy ra sự kiện nghiên cứu.

Trong bài toán trên ta cần dự báo ra giá trị y=1 hoặc y =0 (Positive hay Negative)

Đầu ra được thể hiện dưới dạng xác suất (probability), cho nên ta sử dụng một giá trị ngưỡng t = 0.5:

* Nếu P(y=1) 0.5: dự báo y = 1
* Nếu P(y=0) 0.5: dự báo y = 0

**Boundary tạo bởi Logistic Regression có dạng tuyến tính. Vậy nên mô hình này rất phù hợp với loại dữ liệu mà hai class**

Thật vậy, theo lập luận ở phần trên thì chúng ta cần kiểm tra:

Đặt z =

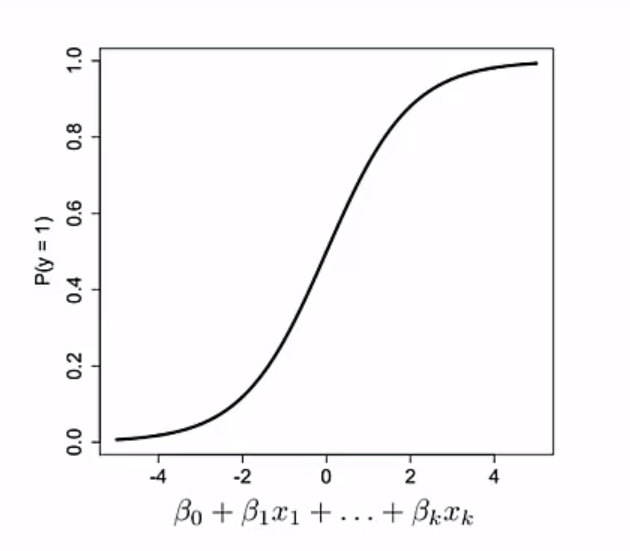
Dự báo y = 1 nếu P(y=1) 0.5:

P(y=1)= 0.5

⇔ z 0

⇔

Tương tự dự báo y = 0 nếu P(y=1) < 0.5:



Hình 3. : Đồ thị tuyến tính Logistic Regression.

Nói cách khác, boundary giữa hai class là đường có phương trình f(x)=. Đây chính là phương trình của một siêu mặt phẳng => Logistic Regression tạo ra boundary có dạng tuyến tính.

#### Đưa dữ liệu huấn luyện vào Logistic Regression của thư viện sklearn:

**from** sklearn.linear\_model **import** LogisticRegression

**from** sklearn.model\_selection **import** train\_test\_split X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=7,shuffle=**True**)

Model = LogisticRegression()

model.fit(X\_train,y\_train)

Trong đó:

* X\_train là dữ liệu huấn luyện
* y\_train là nhãn huấn luyện
* X\_test là dữ liệu kiểm thử
* y\_test là nhãn kiểm thử

X\_train, y\_train là bộ dữ liệu huấn luyện được lấy ra 80%(= 3964 bình luận) dòng từ bộ dữ liệu sau đó được đưa vào model để huấn luyện.

X\_test, y\_test là bộ dữ liệu kiểm thử được lấy ra 20% dòng tự bộ dữ liệu sẽ dùng để đánh giá mô hình sau này.

Lấy ra 25 từ có ý nghĩa “Positve” và “Negative” có trong bộ từ điển:



Hình 3. : Biểu đồ biểu diễn hệ số lớn nhất và nhỏ nhất của Logistic Regression đã được huấn luyện dựa trên tf-idft.

Các từ có hệ số âm ở bên trái thuộc về các từ mà theo mô hình là biểu thị cho các bình luận Negative, trong khi đó các hệ số dương ở bên phải thuộc về các từ mà theo mô hình là biểu thị cho các bình luận Positive. Hầu hết các từ khá trực quan cho từng quan điểm, như là: thất\_vọng, kém, đơ, chậm, tệ, chán, lag, thua, yếu, giật, mờ, tạm, tụt, đắt là các từ biểu thị chính xác cho ý kiến Negative về điện thoại, còn ở Positive có: đỉnh, ok, ưng, thích, nhiệt\_tình, ngon, đáng, sang\_trọng, mượt\_mà, rẻ, tuyệt, đẹp, tuyệt\_vời. Bên cạnh đó có một số từ chưa phản anh rõ về 1 trong cả 2 quan điểm như samsung, thông\_báo, phần\_mềm, cảm\_ứng, thiết\_kế, trả\_góp…

Thông qua đây ta biết được người dùng khi đánh giá các mặt hàng điện thoại di động thường để ý đến: Tốc độ xử lý, thiết kế, pin, phần mềm, phương thức thanh toán…

Tạo function model dùng để dự đoán dữ liệu mới sau này:

**def** model(text):

**for** i **in** text:

test=tf.transform(i)

**return** model.predict(test)

## Đánh giá mô hình:

### Tại sao phải đánh giá mô hình:

Khi xây dựng một mô hình Machine Learning, chúng ta cần một phép đánh giá để xem mô hình sử dụng có hiệu quả không và để so sánh khả năng của các mô hình.

Có rất nhiều cách đánh giá một mô hình phân lớp. Tuỳ vào những bài toán khác nhau mà chúng ta sử dụng các phương pháp khác nhau. Các phương pháp thường được sử dụng là: Accuracy score, confusion matrix, ROC curve, Area Under the Curve, Precision and Recall, F1 score, Top R error…

Cách tính sử dụng Accuracy như ở trên chỉ cho chúng ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác. Bây giờ ta sẽ áp dụng một số phương pháp để đánh giá rõ hơn mô hình vừa xây dựng của chúng ta.

### Đánh giá bằng Confusion Matrix:

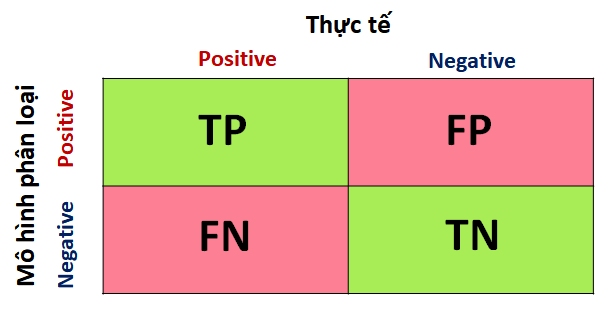
Trước khi ta bắt đầu thì hãy chú ý những định nghĩa và viết tắt sau:

* True Positive ( Tích cực đúng) = **TP**
* False Positive (Tích cực sai) = **FP**
* True Negative (Tiêu cực đúng) = **TN**
* False Positive (Tiêu cực sai) = **FP**

True (Đúng) chỉ sự tương hợp, khi kết quả phân loại phù hợp với giá trị thực tế; False (Sai) chỉ sự bất xứng hay nhầm lẫn, khi mô hình phân loại nhầm so với thực tế.

Confusion matrix hay error matrix. Confusion matrix là thuật ngữ của giới Machine learning, tương đương với khái niệm contingency table hay cross-table bên giới thống kê. Nó có bản chất là một bảng phân phối tần số 2 chiều (bảng chéo) cho phép trình bày tỉ lệ tương hợp và bất xứng giữa Thực tế và kết quả phân loại của quy luật cần kiểm tra (mô hình).Trong phạm vi của chúng ta, ta chỉ xét trường hợp đơn giản nhất của confusion matrix áp dụng cho bài toán nhị phân (Binary classification). Khi đó, Confusion matrix trình bày tần suất của 4 tổ hợp: TP (True Positive), TN (True Positive), FP (False positive) và FN (False negetive). True (Đúng) chỉ sự tương hợp, khi kết quả phân loại phù hợp với giá trị thực tế; False (Sai) chỉ sự bất xứng hay nhầm lẫn, khi mô hình phân loại nhầm so với thực tế.Về cơ bản, confusion matrix thể hiện có bao nhiêu điểm dữ liệu thực sự thuộc vào một class, và được dự đoán là rơi vào một class.

Để hiểu rõ hơn, hãy xem bảng dưới đây:



Ma trận thu được được gọi là confusion matrix. Nó là một ma trận vuông với kích thước mỗi chiều bằng số lượng lớp dữ liệu. Giá trị tại hàng thứ i, cột thứ j là số lượng điểm lẽ ra thuộc vào class i nhưng lại được dự đoán là thuộc vào class j.

Cách biểu diễn trên đây của confusion matrix còn được gọi là unnormalized confusion matrix, tức ma confusion matrix chưa chuẩn hoá. Để có cái nhìn rõ hơn, ta có thể dùng normalized confuion matrix, tức confusion matrix được chuẩn hoá. Để có normalized confusion matrix, ta lấy mỗi hàng của unnormalized confusion matrix sẽ được chia cho tổng các phần tử trên hàng đó. Như vậy, ta có nhận xét rằng tổng các phần tử trên một hàng của normalized confusion matrix luôn bằng 1. Điều này thường không đúng trên mỗi cột.

Sử dụng function confusion\_matrix của thư viện sklearn để xây dựng unnormalized confusion matrix và normalized confusion matrix cho bộ dữ liệu test của chúng ta:

**from** sklearn.metrics **import** confusion\_matrix

y\_pre = model.predict(X\_test)

cnf\_matrix = confusion\_matrix(y\_test, y\_pre)

normalized\_confusion\_matrix=cnf\_matrix/cnf\_matrix.sum( asix=**1**, keepdims=**True**)

print(cnf\_matrix)

print(normalized\_confusion\_matrix

Output:

* Unnormalization confusion matrix:

[[314; 110]

[ 41; 526]]

* Normal confusion matrix:

[[0.74056604; 0.25943396]

[0.07231041; 0.92768959]]

Bộ dữ liệu test của chúng ta lấy 20% dữ liệu từ bộ dữ liệu đã cào được từ đầu. Tức là tổng cộng có 991 điểm trong bộ dữ liệu test. Trong đó có 424 dữ liệu có nhãn là Negative (tiêu cực) và 567 dữ liệu là Positive (tích cực)

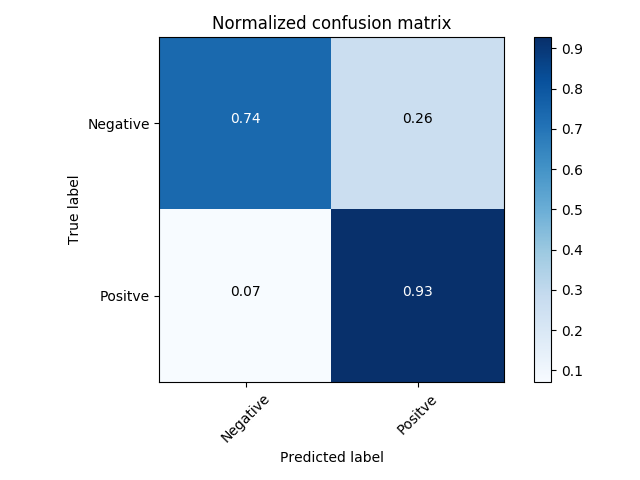
Dựa vào Unnormalization confusion matrix ta có:

* + TP = 526
  + FP = 41
  + TN = 314
  + FN = 110

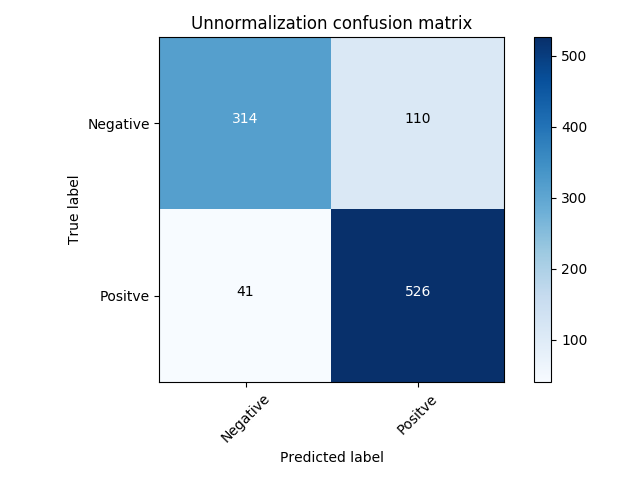
Dựa vào Normalization confusion matrix vào ta có:

* + TP = 93%
  + FP = 7%
  + TN = 74%
  + FN = 26%
* Tức là sau khi bộ dữ liệu test chạy qua mô hình thì có:
* Trong 567 điểm positive ta đự đoán đúng được 526 điểm Positive (93%), đã dự đoán sai 41 điểm Positive (7%).
* Trong 424 điểm Nagative ta dự đoán đúng được 314 điểm Negative (74%), đã dự đoán sai 110 điểm Negative (24%).

Confusion matrix thường được minh hoạ bằng màu sắc để có cái nhìn rõ ràng hơn. Sử dụng thư viện matplotlb để vẽ biểu đồ dựa theo hai matrix trên.



Hình 3. : Normoalization confusion matrix của bộ dữ liệu test sau khi được predict.



Hình 3. : Unnormoalization confusion matrix của bộ dữ liệu test sau khi được predict.

Cách biểu diễn bằng màu này rất hữu ích. Các ô màu đậm thể hiện các giá trị cao. Một mô hình tốt sẽ cho một confusion matrix có các phần tử trên đường chéo chính có giá trị lớn, các phần tử còn lại có giá trị nhỏ. Nói cách khác, khi biểu diễn bằng màu sắc, đường chéo có màu càng đậm so với phần còn lại sẽ càng tốt. Từ hai hình trên ta thấy rằng confusion matrix đã chuẩn hoá mang nhiều thông tin hơn.

* Sự khác nhau được thấy ở ô trên cùng bên trái. Lớp dữ liệu Negative được phân loại không thực sự tốt nhưng trong unnormalized confusion matrix, nó vẫn có màu đậm như hai ô còn lại trên đường chéo chính.

### Đánh giá bằng đường cong ROC:

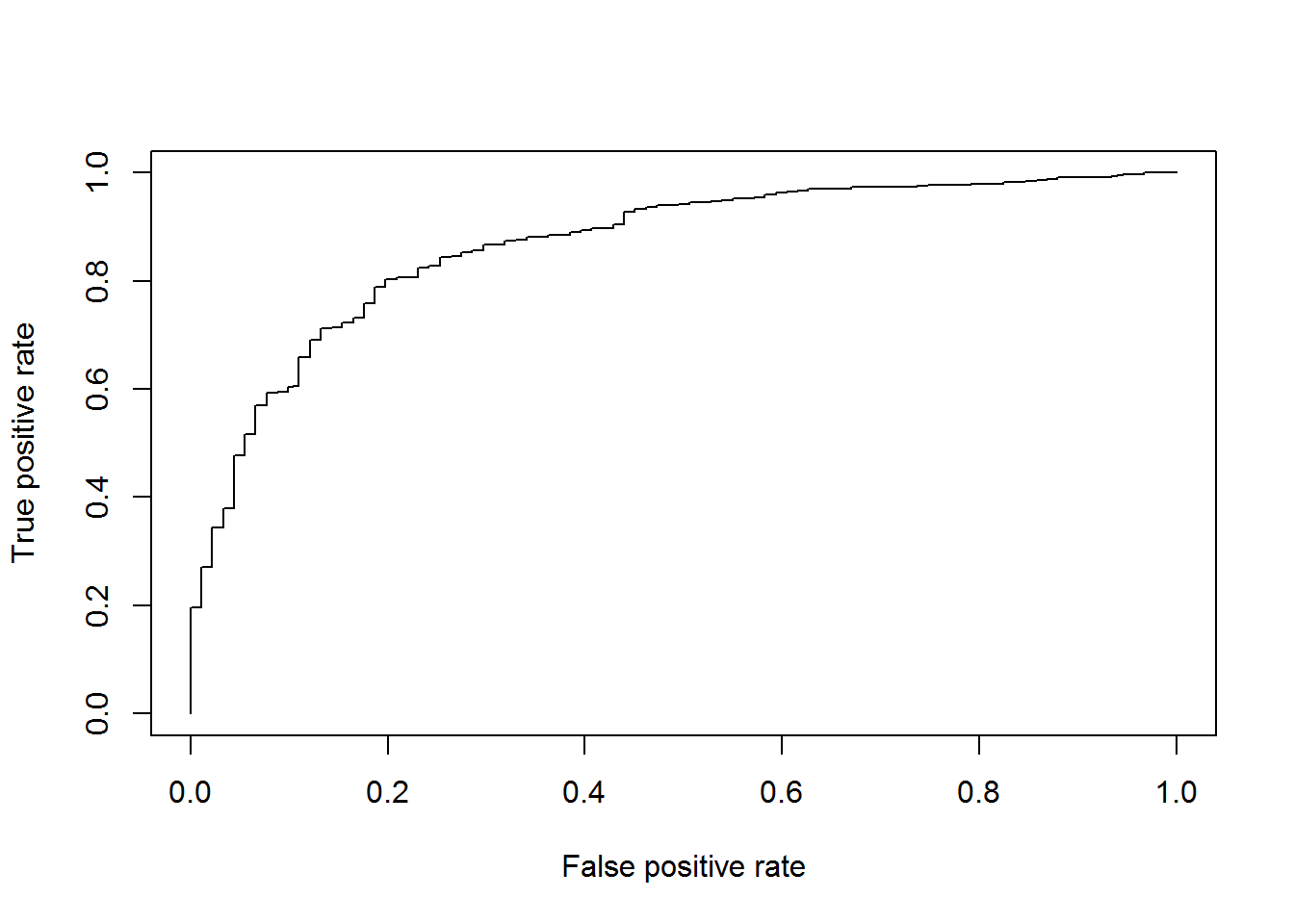
ROC (Receiver operating characteristic) là một đồ thị được sử dụng khá phổ biến trong validation các model phân loại nhị phân. Đường cong này được tạo ra bằng cách biểu diễn tỷ lệ dự báo true positive rate (TPR) dựa trên tỷ lệ dự báo failse positive rate (FPR) tại các ngưỡng Threshold khác nhau. Trong machine learning Chúng ta gọi true positive rate là độ nhạy sensitivity tức là xác xuất dự báo đúng một sự kiện là positive. Tỷ lệ false positive rate là probability of false alarm (tỷ lệ cảnh báo sai, một sự kiện là negative nhưng coi nó là positive). Như vậy ROC curve sẽ thể hiện mối quan hệ, sự đánh đổi và ý nghĩa lựa chọn một model phù hợp của độ nhạy và tỷ lệ cảnh báo sai.

Trong một số bài toán, việc tăng hay giảm FNR, FPR có thể được thực hiện bằng việc thay đổi một ngưỡng (threshold) nào đó. Khi ta sử dụng thuật toán Logistic Regression, đầu ra của mô hình có thể là các lớp cứng 0 hay 1, hoặc cũng có thể là các giá trị thể hiện xác suất để dữ liệu đầu vào thuộc vào lớp 1. Khi sử dụng thư viện sklearn Logistic Regression, ta có thể lấy được các giá trị xác xuất này bằng phương thức predict\_proba(). Mặc định, ngưỡng được sử dụng là 0.5, tức là một điểm dữ liệu x sẽ được dự đoán rơi vào lớp 1 nếu giá trị predict\_proba(x) lớn hơn 0.5 và ngược lại.

Nếu bây giờ ta coi lớp 1 là lớp Positive, lớp 0 là lớp Negative, câu hỏi đặt ra là làm thế nào để tăng FPR để giảm FNR? Chú ý rằng tăng FNR đồng nghĩa với việc giảm TPR vì tổng của chúng luôn bằng 1.

Một kỹ thuật đơn giản là ta thay giá trị threshold từ 0.5 xuống một số nhỏ hơn. Chẳng hạn nếu chọn threshold = 0.3, thì mọi điểm được dự đoán có xác suất đầu ra lớn hơn 0.3 sẽ được dự đoán là thuộc lớp Positive. Nói cách khác, tỉ lệ các điểm được phân loại là Positive sẽ tăng lên, kéo theo cả False Positive Rate và True Positive Rate cùng tăng lên (cột thứ nhất trong ma trận tăng lên). Từ đây suy ra cả FNR và TNR đều giảm.

Nếu ngược lại, tất nhiên là ở mức độ nào đó, như bài toán xác định email rác chẳng hạn, ta cần tăng threshold lên một số lớn hơn 0.5. Khi đó, hầu hết các điểm dữ liệu sẽ được dự đoán thuộc lớp 0, tức Negative, và cả TNF và FNR đều tăng lên, tức TPR và FPR giảm xuống.



Hình 3. : Đồ thị Receiver Operating Characteristic curve

Như vậy, ứng với mỗi giá trị của threshold, ta sẽ thu được một cặp (FPR, TPR). Biểu diễn các điểm (FPR, TPR) trên đồ thị khi thay đổi threshold từ 0 tới 1 ta sẽ thu được một đường được gọi là Receiver Operating Characteristic curve hay ROC curve. (Chú ý rằng khoảng giá trị của threshold không nhất thiết từ 0 tới 1 trong các bài toán tổng quát. Khoảng giá trị này cần được đảm bảo có trường hợp TPR/FPR nhận giá trị lớn nhất hay nhỏ nhất mà nó có thể đạt được).

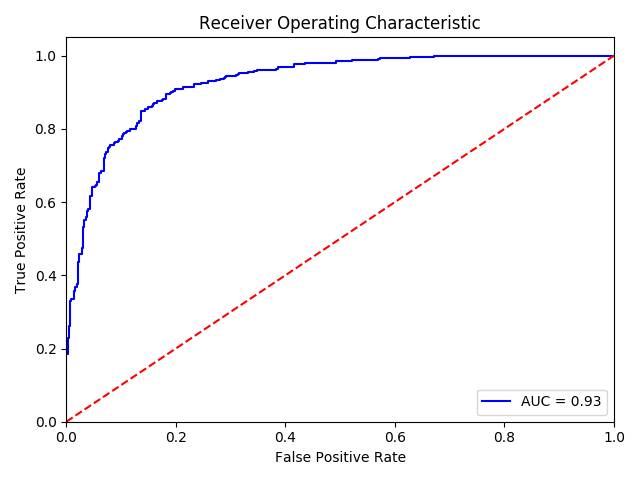
Dựa trên ROC curve, ta có thể chỉ ra rằng một mô hình có hiệu quả hay không. Một mô hình hiệu quả khi có FPR thấp và TPR cao, tức tồn tại một điểm trên ROC curve gần với điểm có toạ độ (0, 1) trên đồ thị (góc trên bên trái). Curve càng gần thì mô hình càng hiệu quả.

Có một thông số nữa dùng để đánh giá mà tôi đã sử dụng ở trên được gọi là Area Under the Curve hay AUC. Đại lượng này chính là diện tích nằm dưới ROC curve màu xanh. Giá trị này là một số dương nhỏ hơn hoặc bằng 1. Giá trị này càng lớn thì mô hình càng tốt.

Sử dụng thư viện Sklearn và mabplot để vẽ đường cong ROC và tính AUC của mô hình đã huấn luyện.

*# Tính fpr và tpr cho tất cả thresholds***import** sklearn.metrics **as** metrics  
probs = model.predict\_proba(X\_test)  
preds = probs[:,1]  
fpr, tpr, threshold = metrics.roc\_curve(y\_test, preds)  
roc\_auc = metrics.auc(fpr, tpr)  
*# Vẽ đường cong ROC***import** matplotlib.pyplot **as** plt  
plt.title(**'Receiver Operating Characteristic'**)  
plt.plot(fpr, tpr, **'b'**, label = **'AUC = %0.2f'** % roc\_auc)  
plt.legend(loc = **'lower right'**)  
plt.plot([0, 1], [0, 1],**'r--'**)  
plt.xlim([0, 1])  
plt.ylim([0, 1.05])  
plt.ylabel(**'True Positive Rate'**)  
plt.xlabel(**'False Positive Rate'**)  
plt.show()

* Output



Hình 3. : Đồ thị đường cong ROC của mô hình

**\* Như vậy sau khi đánh giá mô hình bằng Confusion Matrix và đường cong ROC ta có thể nhận định được bộ mô hình đánh giá khá tốt độ chính xác khoảng 85%**. **AUC = 0.93 khá tốt. Nhưng mô hình phân lớp dữ liệu thuộc lớp Negative vẫn chưa được tốt ( chỉ dự đoán đúng 74% trên tổng số Negative trong bộ dữ liệu test)**

# CHƯƠNG 4 TRIỂN KHAI MÔ HÌNH PHÂN TÍCH PHẢN HỒI CHƯA ĐƯỢC GÁN NHÃN

## Kết quả triển khai

Khi một mặt hàng hoặc một loại điện thoại mới ra mắt, thì sẽ có nhiều bình luận, bài viết nhận xét về nó. Những bình luận đó có thể ở các bình luận trong các trang thương mại điện tử, facebook, youtube, các bài báo đánh giá… Các hãng sản xuất điện thoại hoặc các cửa hàng kinh doanh mặt hàng điện thoại di động sẽ muốn lấy được các dữ liệu bình luận đó và phân tích ý kiến của khách hàng từ đó nắm được ưu khuyết điểm của sản phẩm để đưa ra kế hoạch hoặc hoặc hướng đi tiếp theo.

Những dữ liệu đó khi được thu thập từ bình luận face book, youtube hoặc các bài báo…về sẽ không có nhãn Positive (tích cực) và Negative (tiêu cực). Những dữ liệu này được cho chạy qua mô hình đã xây dựng ở trên sẽ cho ra dữ liệu đã gán nhán tức là kết quả phân tích từng ý kiến của khách hàng.

Quy trình triển khai phân tích dữ liệu mới có một số bước giống như quy trình xây dựng mô hình như là tiền xử lý dữ liệu, vector hoá dữ liệu, chạy dữ liệu qua mô hình. Ở đây việc chạy dữ liệu qua mô hình thì dữ liệu mới này giống như dữ liệu test ở bước đánh giá mô hình.

Bộ dữ liệu cần phân tích dự đoán có thể được lấy trong bộ cơ sở của doanh nghiệp, từ cào về từ các bình luận trên các trang thương mại điện tử, facebook, youtube hoặc từ quá trình thu thập thông qua catalog, bảng hỏi…

Dưới đây là 10 bình luận của điện thoại Samsung galaxy S10 được lấy từ http://thegioididong.com

Bảng 4. : Bảng dữ liệu bao gồm 10 bình luận chưa được gán nhãn

|  |  |
| --- | --- |
|  | **Review** |
| 1 | E mới mua con s10 được gần 2 tuần mà sao xài nhanh nóng quá ạ Hay do cài đặt e chỉnh sai gì ạ |
| 2 | Máy hết pin nhanh quá, có cách nào khắc phục không, mọi thứ đều ổn chỉ có pin là tệ, 4sao |
| 3 | Dùng chưa đầy một tháng nhưng quá nhiều thất vọng thỉnh thoảng đọc báo treo không vuốt được, cảm ứng có cuộc gọi ko vuốt nghe được kém hơn cả huawei nếu ai có ý định rước em này nên suy nghĩ pin xuống nhanh nữa thất vọng |
| 4 | Màn hình cứ tự bật sáng khi bỏ túi,y như máy đểu ngày xưa, pin yếu,vân tay lì.. Quá thất vọng |
| 5 | Máy đẹp, mượt, tất cả đều hài lòng nhưng điểm yếu pin tụt quá nhanh dù tắt hết ứng dụng ngầm nên cho 4 sao |
| 6 | Máy đẹp, dùng mượt, pin hơi yếu tí. Dán màn nên vân tay hơi chậm. Nói chung đến thời điểm hiện tại khá hài lòng |
| 7 | Máy đẹp, nhanh , ngon, chỉ tiếc là Pin tụt hơi nhanh. Còn lại mọi cái đều tuyệt vời. |
| 8 | Đã mua và sử dụng từ đời s8.rất đẹp và ok.màu xanh khá đẹp.s10 nhỏ vừa tay hơn s10+.mình đã chọn s10. |
| 9 | Dùng màn hình đẹp, pin hơi yếu so với dòng A, wifile bắt khoảng cách xa yếu hơn dòng lumia |
| 10 | Nhận máy s10 từ hôm qua tại cửa hàng ở số 3 Quang Trung, Nha Trang. Điện thoại dùng quá ngon, màu xanh của máy đẹp, màn hình sáng, pin khá ổn, chụp hình ấn tượng. Chỉ có cái cảm ứng siêu âm khi ko có dán màn hình thì quá ngon, hy vọng sớm ra dán màn hình tích hợp. Nhân viên TGDĐ nhiệt tình, cảm ơn đã giao máy đúng hẹn. |

Từ bộ dữ liệu trên thực hiện tương tự như quá trình xây dựng mô hình, ta phải tiền xử lý dữ liệu trước khi đưa vào mô hình bao gồm: Làm sạch, tách từ, loại bỏ stop word, vecto hóa.

Bảng 4. 2: Bảng dữ liệu bao gồm 10 bình luận sau khi tiền xữ lý.

|  |  |
| --- | --- |
|  | **Review** |
| 1 | e mới mua con gần tuần sao xài nhanh nóng quá ạ hay cài\_đặt e chỉnh sai ạ |
| 2 | máy hết pin nhanh quá cách nào khắc\_phục mọi thứ ổn pin tệ |
| 3 | dùng đầy một tháng quá thất\_vọng thỉnh\_thoảng đọc báo treo vuốt cảm\_ứng cuộc\_gọi ko vuốt nghe kém hơn huawei ai ý\_định rước em suy\_nghĩ pin xuống nhanh thất\_vọng |
| 4 | màn\_hình tự bật sáng bỏ\_túi y\_như máy đểu ngày\_xưa pin yếu vân tay lì quá thất\_vọng |
| 5 | máy đẹp mượt tất\_cả hài\_lòng điểm yếu pin tụt quá nhanh dù tắt hết ứng\_dụng ngầm sao |
| 6 | máy đẹp dùng mượt pin hơi yếu tí dán màn vân tay hơi chậm nói\_chung đến thời\_điểm hiện\_tại khá hài\_lòng |
| 7 | máy đẹp nhanh ngon tiếc pin tụt hơi nhanh còn mọi tuyệt\_vời |
| 8 | mua sử\_dụng đời đẹp ok màu xanh khá đẹp nhỏ tay hơn mình chọn |
| 9 | dùng màn\_hình đẹp pin hơi yếu dòng a wifile bắt khoảng\_cách xa yếu hơn dòng lumia |
| 10 | nhận máy hôm\_qua cửa\_hàng ở số quang\_trung nha trang điện\_thoại dùng quá ngon màu xanh máy đẹp màn\_hình sáng pin khá ổn chụp hình ấn\_tượng cảm\_ứng siêu\_âm ko dán màn\_hình quá ngon hy\_vọng sớm dán màn\_hình tích\_hợp nhân\_viên tgdđ nhiệt\_tình cảm\_ơn giao máy đúng hẹn |

Sau đó bộ dữ liệu được vectơ hóa và qua model để dự phân tích ta được kết quả là từ những bình luận chưa được gán nhãn sẽ cho ra những bình luận đó được gán nhãn dự đoán:

Bảng 4. : Bảng dữ liệu bao gồm 10 được dán nhãn sau khi được áp dụng mô hình đã xây dựng.

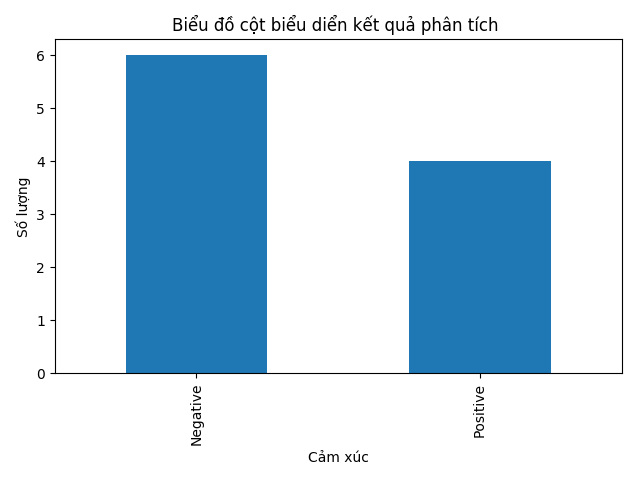
|  |  |  |
| --- | --- | --- |
|  | **Review** | **Result** |
| 1 | E mới mua con s10 được gần 2 tuần mà sao xài nhanh nóng quá ạ Hay do cài đặt e chỉnh sai gì ạ | **Negative** |
| 2 | Máy hết pin nhanh quá, có cách nào khắc phục không, mọi thứ đều ổn chỉ có pin là tệ, 4sao | **Negative** |
| 3 | Dùng chưa đầy một tháng nhưng quá nhiều thất vọng thỉnh thoảng đọc báo treo không vuốt được, cảm ứng có cuộc gọi ko vuốt nghe được kém hơn cả huawei nếu ai có ý định rước em này nên suy nghĩ pin xuống nhanh nữa thất vọng | **Negative** |
| 4 | Màn hình cứ tự bật sáng khi bỏ túi,y như máy đểu ngày xưa, pin yếu,vân tay lì.. Quá thất vọng | **Negative** |
| 5 | Máy đẹp, mượt, tất cả đều hài lòng nhưng điểm yếu pin tụt quá nhanh dù tắt hết ứng dụng ngầm nên cho 4 sao | **Negative** |
| 6 | Máy đẹp, dùng mượt, pin hơi yếu tí. Dán màn nên vân tay hơi chậm. Nói chung đến thời điểm hiện tại khá hài lòng | **Positive** |
| 7 | Máy đẹp, nhanh , ngon, chỉ tiếc là Pin tụt hơi nhanh. Còn lại mọi cái đều tuyệt vời. | **Positive** |
| 8 | Đã mua và sử dụng từ đời s8.rất đẹp và ok.màu xanh khá đẹp.s10 nhỏ vừa tay hơn s10+.mình đã chọn s10. | **Positive** |
| 9 | Dùng màn hình đẹp, pin hơi yếu so với dòng A, wifile bắt khoảng cách xa yếu hơn dòng lumia | **Negative** |
| 10 | Nhận máy s10 từ hôm qua tại cửa hàng ở số 3 Quang Trung, Nha Trang. Điện thoại dùng quá ngon, màu xanh của máy đẹp, màn hình sáng, pin khá ổn, chụp hình ấn tượng. Chỉ có cái cảm ứng siêu âm khi ko có dán màn hình thì quá ngon, hy vọng sớm ra dán màn hình tích hợp. Nhân viên TGDĐ nhiệt tình, cảm ơn đã giao máy đúng hẹn. | **Positive** |

Từ kết quả phân tích dự đoán, ta có thể trực quan hoá chúng bằng một số biểu đồ.

**\*Nhìn vào kết quả, thì dữ liệu sau khi đã được phân tích và được gán nhãn thì độ chính xác khá tốt (khoảng 85%). Từ đó, ta có thể dự đoán khá chính xác quan điểm của từng khách hàng về sản phẩm là tiêu cực hay tích cực. Từ đó đưa ra những quyết định chiến lược kinh doanh tiếp theo dựa theo đó.**



Hình 4. : Biếu đồ tròn biểu diển tỉ lệ cảm xúc sau khi phân tích dự đoán



Hình 4. : Biểu đồ cột biểu diễn kết quả sau khi phân tích dự đoán

# KẾT LUẬN

1. Kết quả đã đạt được:

Xây dựng thành công hệ thống phân tích ý kiến của khách hàng về mặt hàng điện thoại di động bao gồm các bước: Tiền xử lý dữ liệu, vector hoá dữ liệu, mô hình hoá, đánh giá mô hình.

Tìm hiểu và sử dụng tốt các thư viện có sẳn trong ngôn ngữ Python để thực hiện đề tài như: Sklean, Selenium Webdriver, Pyvi, Pandas, Matplotlb, Seaborn…

Triển khai thành công hệ thống để dự đoán dữ liệu thực chưa gán nhãn với độ chính xác khá tốt khoảng 85%. Từ đó có thể áp dụng thực tiễn vào thực tế giúp các doanh nghiệp hoặc các bên có liên quan về mặt hàng điện thoại di đông có thể đưa ra những chiến lược, kế hoạch tốt.

1. Khó khăn:

Tài liệu tham khảo về kiến thức thuộc phạm trù của đề tài còn ít, đa số là tiếng anh.

Việc thu thập dữ liệu gây tốn nhiều thời gian và sức lực cho nên bộ dữ liệu thu thập được chưa được nhiều (khoảng 5000 bình luận).

Bộ dữ liệu training chưa được tốt: nhãn Positive và Negative của bộ dữ liệu huấn luyện chưa được gán chuẩn xác. Vì vậy tỉ lệ dự đoán đúng Positve của mô hình cao hơn tỉ lệ dự đoán đúng của Negative. Hệ thống xây dựng từ bước đầu đến cuối vẫn còn rời rạc chưa thống nhất.

1. Hướng phát triển trong tương lai:

Thu thập thêm nhiều bộ dữ liệu huấn luyện và kiểm thử (>10000 bình luận)

Xây dựng mô hình phân tích nhiều lớp: thêm một lớp Neutral (Trung tính). Áp dụng Deep Learning để có thể phân tích kỹ hơn.

Tiền xử lý dữ liệu tốt hơn, bổ sung thêm: Kiểm tra chính tả tiếng Việt, thêm dấu tiếng Việt, Loại bỏ thêm stopword phù hợp với bộ dữ liệu. Gán nhãn tốt hơn cho bộ dữ liệu huấn luyện. Từ đó tăng độ chính xác của mô hình hơn

Áp dụng hướng đối tượng trong lúc xây dựng hệ thống.

Kết hợp với doanh nghiệp để áp dụng thực tế mô hình đã xây dựng.

|  |  |
| --- | --- |
| Dạng tài liệu | Chi tiết |
| Sách | Müller, A. (2016). *Introduction to machine learning with python.* |
| Trang web | (n.d.). Retrieved from Giới thiệu tiền xử lý trong xử lý ngôn ngữ tự nhiên: https://codetudau.com/gioi-thieu-tien-xu-ly-trong-xu-ly-ngon-ngu-tu-nhien/index.html  (n.d.). Retrieved from *Giới thiệu tiền xử lý trong xử lý ngôn ngữ tự nhiên:* https://codetudau.com/gioi-thieu-tien-xu-ly-trong-xu-ly-ngon-ngu-tu-nhien/index.html  Chức, N. V. (n.d.). *Mô hình hồi qui logistic (Logistic Regression Model)*. Retrieved from http://bis.net.vn/forums/t/484.aspx  Tiệp, V. H. (n.d.). Retrieved from *Machine Learning cơ bản:* https://machinelearningcoban.com |

# TÀI LIỆU THAM KHẢO