**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**





**TIỂU LUẠN  
CHUYÊN ĐỀ TỐT NGHIỆP KHOA HỌC DỮ LIỆU 3**

**XÂY DỰNG MÔ HÌNH LỌC THƯ RÁC BẰNG THUẬT TOÁN NAIVE BAYES**

**GIẢNG VIÊN HƯỚNG DẪN :   ThS. PHẠM ĐÌNH TÀI**

**SINH VIÊN THỰC HIỆN :   VÕ THÁI HIỂN**

**MSSV :   2000005690**

**LỚP :   20DTH1D**

**CHUYÊN NGÀNH :   KHOA HỌC DỮ LIỆU**

**KHÓA :   2020**

**Tp.HCM, Tháng  11  Năm  2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**





**TIỂU LUẬN  
CHUYÊN ĐỀ TỐT NGHIỆP KHOA HỌC DỮ LIỆU 3**

**XÂY DỰNG MÔ HÌNH LỌC THƯ RÁC BẰNG THUẬT TOÁN NAIVE BAYES**

**GIẢNG VIÊN HƯỚNG DẪN :   ThS PHẠM ĐÌNH TÀI**

**SINH VIÊN THỰC HIỆN :   VÕ THÁI HIỂN**

**MSSV :   2000005690**

**LỚP :   20DTH1D**

**CHUYÊN NGÀNH :   KHOA HỌC DỮ LIỆU**

**KHÓA :   2020**

**Tp.HCM, Tháng  11  Năm  2023**

LỜI CẢM ƠN

Em xin gửi lời cảm ơn từ tận đáy lòng tới Thầy Sử Nhật Hạ và trường Đại học Nguyễn Tất Thành (NTTU) về sự hỗ trợ và cơ hội quý báu trong quá trình thực hiện chuyên đề tốt nghiệp khoa học dữ liệu 2. Đây là một hành trình quan trọng và đáng nhớ, và em không thể không bày tỏ lòng biết ơn của mình.

Thầy Sử Nhật Hạ, với kiến thức sâu rộng và tâm huyết với giảng dạy, đã là người hướng dẫn tận tâm và thông thái. Sự hướng dẫn của thầy không chỉ giúp em phát triển kỹ năng nghiên cứu mà còn giúp em hiểu sâu hơn về lĩnh vực em nghiên cứu. Em cảm ơn thầy về sự khuyến khích, hỗ trợ, và sự lãnh đạo trong suốt thời gian qua. Thầy là nguồn động viên quý báu cho sự thành công của em trong dự án này.

Tôi cũng muốn bày tỏ lòng biết ơn tới trường đại học NTT vì cơ hội và môi trường học tập tốt nghiệp chất lượng mà trường đã mang lại. Trường đại học NTT đã cung cấp cho em sự hỗ trợ về cơ sở vật chất, tài liệu tham khảo, và một cộng đồng học thuật đáng kính. Em rất tự hào vì đã được đào tạo tại Trường đại học NTT và mang danh hiệu cử nhân của trường.

Cuối cùng, em muốn gửi lời cảm ơn sâu sắc tới gia đình, bạn bè, và người thân của mình. Họ luôn ở bên em, hỗ trợ tinh thần và động viên khi tôi gặp khó khăn. Những người này là nguồn động viên quan trọng giúp em vượt qua mọi thử thách trong quá trình nghiên cứu và viết chuyên đề tốt nghiệp này.

Bài khoá luận này không thể hoàn thành mà không có sự đóng góp của tất cả những người trên. Em hi vọng rằng công trình này có thể mang lại giá trị cho cộng đồng và đóng góp vào lĩnh vực của Khoa học dữ liệu.

Cảm ơn tất cả mọi người một lần nữa vì sự hỗ trợ và đóng góp.

***Sinh viên thực hiện***

**Võ Thái Hiển**

LỜI MỞ ĐẦU

Chào các thầy cô hướng dẫn, các bạn học viên, và quý khách mời, em rất hân hạnh và tự hào được trình bày trước mặt quý vị chuyên đề tốt nghiệp khoa học dữ liệu lần thứ hai. Chuyên đề này đánh dấu một bước ngoặt quan trọng trong hành trình học tập và nghiên cứu của chúng tôi tại khoa khoa học dữ liệu.

Trong thời đại số hóa và thông tin hiện nay, dữ liệu đang trở thành nguồn tài nguyên quý báu và có tiềm năng biến đổi toàn bộ xã hội và kinh tế. Lĩnh vực khoa học dữ liệu đang trỗi dậy, và em đã dành nhiều tháng, thậm chí nhiều năm để tìm hiểu, nghiên cứu, và áp dụng các phương pháp và kỹ thuật trong lĩnh vực này. Em tin rằng chuyên đề này sẽ là cơ hội để em chia sẻ kiến thức và kết quả nghiên cứu của mình, đồng thời hi vọng rằng nó sẽ đem lại giá trị và kiến thức mới mẻ cho cộng đồng.

Chuyên đề tốt nghiệp khoa học dữ liệu lần này bao gồm những dự án thú vị và đa dạng, từ việc áp dụng machine learning và deep learning để giải quyết các vấn đề thực tế, đến việc phân tích và trực quan hóa dữ liệu để hiểu sâu hơn về xu hướng và mối quan hệ. Em cũng đã nghiên cứu các khía cạnh đạo đức và bảo mật trong việc sử dụng dữ liệu.

Chuyên đề này không thể thiếu sự hỗ trợ và động viên từ các thầy cô hướng dẫn, những người đã dành nhiều thời gian và kiến thức để hỗ trợ em trong suốt quá trình nghiên cứu và viết báo cáo. Em xin chân thành cảm ơn sự hướng dẫn và động viên của quý thầy cô.

Cuối cùng, em hy vọng rằng chuyên đề tốt nghiệp khoa học dữ liệu này sẽ đem lại cái nhìn sâu sắc hơn về cách chúng ta có thể sử dụng dữ liệu để giải quyết các vấn đề phức tạp và đóng góp vào sự phát triển của lĩnh vực khoa học dữ liệu. Em rất mong nhận được sự quan tâm và phản hồi tích cực từ quý vị.

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ …..… NĂM HỌC …….. - ….…** |

**PHIẾU CHẤM THI TIỂU LUẬN/BÁO CÁO**

BM-ChT-11

Môn thi: **Chuyên đề tốt nghiệp KHDL 2** Lớp học phần: **20DTH1D**

Nhóm sinh viên thực hiện : **2**

**1. Võ Thái Hiển**. Tham gia đóng góp: 50%

**2. Nguyễn Tiến Thành**. Tham gia đóng góp: 50%

Ngày thi: Phòng thi:

Đề tài tiểu luận/báo cáo của sinh viên : **Xây dựng mô hình lọc thư rác bằng Naive Bayes**

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo | ………………………………………………………… |  |  |
| Nội dung | …………………………………………………………  …………………………………………………………  ………………………………………………………… |  |  |
| * Các nội dung thành phần | ………………………………………………………… |  |  |
| * Lập luận | ………………………………………………………… |  |  |
| * Kết luận | …………………………………………………………  ………………………………………………………… |  |  |
| Trình bày | ………………………………………………………..  ………………………………………………………… |  |  |
| **TỔNG ĐIỂM** |  |  |  |

|  |  |
| --- | --- |
|  | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)* |

**ThS. Phạm Đình Tài**

MỤC LỤC

Nội dung

[LỜI CẢM ƠN i](#_Toc150514589)

[LỜI MỞ ĐẦU ii](#_Toc150514590)

[MỤC LỤC iv](#_Toc150514591)

[DANH MỤC HÌNH v](#_Toc150514592)

[CHƯƠNG 1./ GIỚI THIỆU 6](#_Toc150514593)

[1.1./ Giới thiệu tổng quan về đề tài 6](#_Toc150514594)

[1.1.1./ Thư rác là gì? 6](#_Toc150514595)

[1.1.2./ Tổng quan về thuật toán Naive Bayes 6](#_Toc150514596)

[1.1.3./ Công dụng của mô hình Naive Bayes trong lọc thư rác 7](#_Toc150514597)

[1.1.4./ Tổng quan đề tài 7](#_Toc150514598)

[1.1.5./ Ứng dụng của đề tài 8](#_Toc150514599)

[1.2./ Lý do chọn đề tài 8](#_Toc150514600)

[1.3./ Mục tiêu của đề tài 9](#_Toc150514601)

[1.4./ Đối tượng và phạm vi nghiên cứu 9](#_Toc150514602)

[1.5./ Tầm quan trọng của đề tài 10](#_Toc150514603)

[CHƯƠNG 2./ GIẢI QUYẾT BÀI TOÁN 12](#_Toc150514604)

[2.1./ Cơ sở lý thuyết 12](#_Toc150514605)

[2.1.1./ Nguyên tắc Bayes 12](#_Toc150514606)

[2.1.2./ Naive Bayes trong lọc thư rác 12](#_Toc150514607)

[2.2./ Thuật toán sử dụng 13](#_Toc150514608)

[2.3./ Thiết kế mô hình 15](#_Toc150514609)

[CHƯƠNG 3./ XÂY DỰNG BỘ LỌC THƯ RÁC BẰNG NAIVE BAYES 16](#_Toc150514610)

[3.1./ Bộ dữ liệu 16](#_Toc150514611)

[3.2./ Xây dựng bộ lọc thư rác bằng Naive Bayes 17](#_Toc150514612)

[3.3./ Kết luận 28](#_Toc150514613)

[3.4./ Ưu điểm và nhược điểm 28](#_Toc150514614)

[3.5./ Hướng khắc phục 30](#_Toc150514615)

[3.6./ Kết luận 31](#_Toc150514616)

[TÀI LIỆU THAM KHẢO 32](#_Toc150514617)

DANH MỤC HÌNH

[Hình 3. 1./ Ví dụ minh hoạ sử dụng thư viện Pandas để đọc dữ liệu csv 12](#_Toc150420206)

[Hình 3. 2./ Kết quả của sử dụng thư viện Pandas để đọc dữ liệu csv 12](#_Toc150420207)

[Hình 3. 3./ Ví dụ hiển thị tỉ lệ phần trăm 12](#_Toc150420208)

[Hình 3. 4./ Kết quả của hiển thị tỉ lệ phần trăm 12](#_Toc150420209)

[Hình 3. 5./ Ví dụ tạo một biến ngẫu nhiên 13](#_Toc150420210)

[Hình 3. 6./ Kết quả của tạo biến ngẫu nhiên 13](#_Toc150420211)

[Hình 3. 7./ Ví dụ kiểm tra trong tập huấn luyện 14](#_Toc150420212)

[Hình 3. 8./ Phương trình xác suất là tin nhắn rác 14](#_Toc150420213)

[Hình 3. 9./ Phương trình xác suất là tin nhắn thường 14](#_Toc150420214)

[Hình 3. 10./ Kết quả bộ dữ liệu lúc đầu 15](#_Toc150420215)

[Hình 3. 11./ Kết quả bộ dữ liệu đã làm sạch 16](#_Toc150420216)

[Hình 3. 12./ Ví dụ tạo một "vocabulary" 16](#_Toc150420217)

[Hình 3. 13./ Ví dụ tạo dữ liệu Dataframe 17](#_Toc150420218)

[Hình 3. 14./ Ví dụ code đếm số lần từ xuất hiện 17](#_Toc150420219)

[Hình 3. 15./ Chuyển đổi cuối cùng với tập dữ liệu 18](#_Toc150420220)

[Hình 3. 16./ Ví dụ thêm cột Label 18](#_Toc150420221)

[Hình 3. 17./ Ví dụ sử dụng thuật toán naïve bayes 19](#_Toc150420222)

[Hình 3. 18./ ví dụ code sử dụng công thức có Laplace smoothing 19](#_Toc150420223)

[Hình 3. 19./ Ví dụ code hàm classify() 20](#_Toc150420224)

[Hình 3. 20./ Kiểm tra bộ lọc thư rác 21](#_Toc150420225)

[Hình 3. 21./ Ví dụ hàm trả về nhãn phân loại 21](#_Toc150420226)

[Hình 3. 22./ Tạo một cột mới trong tập kiểm thử 22](#_Toc150420227)

[Hình 3. 23./ So sáng giá trị dự đoán với giá trị thực tế 22](#_Toc150420228)

CHƯƠNG 1./ GIỚI THIỆU

1.1./ Giới thiệu tổng quan về đề tài

1.1.1./ Thư rác là gì?

Thư rác là những thư điện tử, tin nhắn được gửi đến người nhận mà người nhận không mong muốn hoặc không có trách nhiệm phải tiếp nhận theo quy định. Thư rác thường mang nội dung là quảng cáo, giới thiệu sản phẩm hoặc thông tin không có giá trị.

Thư rác gây ra nhiều phiền toái cho người dùng bởi lượng thư rác ngày càng nhiều, chiếm dụng dung lượng lưu trữ và thời gian xử lý của người dùng. Do đó, việc lọc và chặn thư rác là rất quan trọng để bảo vệ người dùng khỏi những thư rác này.

Theo các kết quả tìm kiếm, thư rác thường có nội dung quảng cáo, giới thiệu sản phẩm, thông tin khiêu dâm hoặc lừa đảo để thu hút người nhận. Người dùng cần cảnh giác và áp dụng các biện pháp để ngăn chặn, lọc bỏ thư rác nhằm bảo vệ tài khoản email.

1.1.2./ Tổng quan về thuật toán Naive Bayes

Naive Bayes là một thuật toán phân loại dựa trên nguyên tắc của công thức Bayes. Nó được sử dụng rộng rãi trong các bài toán phân loại và lọc dữ liệu. Thuật toán này dựa trên giả định "naive" rằng các đặc trưng của dữ liệu là độc lập và có cùng ảnh hưởng đối với kết quả phân loại.

Thuật toán Naive Bayes được áp dụng trong nhiều lĩnh vực, bao gồm phân loại văn bản, lọc thư rác, phân loại ảnh, và nhiều bài toán khác. Nó được sử dụng để dự đoán xác suất của một mẫu thuộc vào một lớp cụ thể dựa trên các đặc trưng của mẫu đó.

Thuật toán Naive Bayes tính toán xác suất của một lớp dựa trên xác suất của các đặc trưng trong lớp đó. Nó sử dụng công thức Bayes để tính toán xác suất đồng thời của các đặc trưng và lớp, sau đó sử dụng nguyên tắc của Bayes để cập nhật xác suất dựa trên dữ liệu huấn luyện.

Một ưu điểm của thuật toán Naive Bayes là nhanh và hiệu quả trong việc xử lý các bài toán phân loại. Nó cũng có khả năng làm việc tốt với các tập dữ liệu lớn. Tuy nhiên, giả định "naive" của thuật toán có thể không phù hợp với một số bài toán có sự tương quan mạnh giữa các đặc trưng.

Trong quá trình xây dựng mô hình Naive Bayes, cần tiến hành tiền xử lý dữ liệu để loại bỏ nhiễu và chuẩn hóa dữ liệu. Sau đó, dữ liệu huấn luyện được sử dụng để tính toán các xác suất và xây dựng mô hình. Cuối cùng, mô hình được kiểm tra và đánh giá bằng cách sử dụng dữ liệu kiểm tra.

1.1.3./ Công dụng của mô hình Naive Bayes trong lọc thư rác

Mô hình Naive Bayes có công dụng quan trọng trong việc lọc thư rác. Dưới đây là một số công dụng của mô hình Naive Bayes trong lọc thư rác:

* Phân loại tự động: Mô hình Naive Bayes có khả năng phân loại tự động các email là thư rác hoặc không phải thư rác dựa trên các đặc trưng của email. Thuật toán Naive Bayes sử dụng xác suất để tính toán xác suất một email là thư rác hoặc không phải thư rác, và dựa vào đó để phân loại.
* Hiệu suất cao: Mô hình Naive Bayes có tốc độ xử lý nhanh và hiệu suất cao, đặc biệt là khi áp dụng cho các tập dữ liệu lớn. Điều này làm cho nó trở thành một lựa chọn phổ biến trong việc xây dựng mô hình lọc thư rác.
* Độ chính xác: Mô hình Naive Bayes có khả năng đạt được độ chính xác cao trong việc phân loại thư rác. Mặc dù có giả định "naive" về sự độc lập giữa các đặc trưng, Naive Bayes vẫn cho kết quả tốt trong nhiều trường hợp thực tế.
* Dễ triển khai: Mô hình Naive Bayes dễ triển khai và thực hiện. Thuật toán Naive Bayes không yêu cầu nhiều tham số và không đòi hỏi quá nhiều xử lý trước khi áp dụng.
* Áp dụng đa dạng: Mô hình Naive Bayes có thể được áp dụng trong nhiều lĩnh vực khác nhau ngoài lọc thư rác, bao gồm phân loại văn bản, phân loại tin tức, phân loại sản phẩm và nhiều ứng dụng khác.

1.1.4./ Tổng quan đề tài

Quá trình xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes gồm ba giai đoạn chính:

* Tiền xử lý dữ liệu: Giai đoạn này bao gồm việc tiền xử lý dữ liệu thư rác, như loại bỏ các từ không cần thiết, chuẩn hóa dữ liệu và chuyển đổi dữ liệu thành dạng phù hợp cho việc xây dựng mô hình.
* Xây dựng mô hình: Dữ liệu sau khi đã được tiền xử lý được sử dụng để xây dựng mô hình Naive Bayes. Mô hình này sẽ học từ dữ liệu huấn luyện và tạo ra các xác suất để phân loại các email là thư rác hoặc không phải thư rác.
* Kiểm tra và đánh giá mô hình: Mô hình được kiểm tra bằng cách sử dụng dữ liệu kiểm tra. Kết quả được đánh giá dựa trên các độ đo như độ chính xác, độ phủ và độ F1.

Naive Bayes có thể được áp dụng cho nhiều loại dữ liệu và bài toán phân loại khác nhau. Nó có thể hoạt động tốt trong các bài toán lớn và có thể được cải thiện bằng cách sử dụng các biến thể của thuật toán, như Multinomial Naive Bayes và Gaussian Naive Bayes.

Tuy nhiên, cần lưu ý rằng Naive Bayes có giả định "naive" về sự độc lập giữa các đặc trưng, do đó, nó có thể không hoạt động tốt trong các trường hợp mà giả định này không được thỏa mãn.

1.1.5./ Ứng dụng của đề tài

Ứng dụng của đề tài "Xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes" là giúp tự động phân loại và lọc các email để xác định xem chúng có phải là thư rác hay không. Thuật toán Naive Bayes được sử dụng trong quá trình xây dựng mô hình để học từ dữ liệu huấn luyện và tạo ra các xác suất để phân loại các email. Giúp tự động phân loại và lọc thư rác dựa trên mô hình đã được xây dựng.

Việc xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes có thể giúp cải thiện hiệu suất và hiệu quả trong việc lọc thư rác. Thuật toán này có khả năng xử lý các tập dữ liệu lớn và có thể được áp dụng trong nhiều lĩnh vực khác nhau.

Tuy nhiên, cần lưu ý rằng Naive Bayes có giả định "naive" về sự độc lập giữa các đặc trưng, do đó, nó có thể không hoạt động tốt trong các trường hợp mà giả định này không được thỏa mãn. Để đạt được kết quả tốt, việc tiền xử lý dữ liệu và chọn các đặc trưng phù hợp là rất quan trọng.

1.2./ Lý do chọn đề tài

Lý do chọn đề tài "Xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes" là vì thuật toán này đã được chứng minh là hiệu quả trong việc phân loại và lọc thư rác. Naive Bayes có thể xử lý các tập dữ liệu lớn và có khả năng làm việc tốt với các đặc trưng độc lập.

Mô hình lọc thư rác xây dựng bằng thuật toán Naive Bayes có thể giúp tự động phân loại và lọc các email, giúp người dùng tiết kiệm thời gian và công sức trong việc xử lý thư rác. Thuật toán này dựa trên xác suất và tính toán các xác suất để phân loại các email là thư rác hoặc không phải thư rác.

Một ưu điểm của thuật toán Naive Bayes là tốc độ xử lý nhanh và khả năng làm việc tốt với các tập dữ liệu lớn. Nó cũng có khả năng xử lý các đặc trưng độc lập, giúp đơn giản hóa quá trình xây dựng mô hình.

Tuy nhiên, cần lưu ý rằng Naive Bayes có giả định "naive" về sự độc lập giữa các đặc trưng, do đó, nó có thể không hoạt động tốt trong các trường hợp mà giả định này không được thỏa mãn. Để đạt được kết quả tốt, việc tiền xử lý dữ liệu và chọn các đặc trưng phù hợp là rất quan trọng.

1.3./ Mục tiêu của đề tài

Mục tiêu của đề tài "Xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes" là áp dụng thuật toán Naive Bayes để xây dựng một mô hình có khả năng phân loại và lọc các email, giúp người dùng nhận biết và loại bỏ thư rác một cách tự động.

Việc xây dựng mô hình lọc thư rác có nhiều ứng dụng trong thực tế. Nó giúp người dùng tiết kiệm thời gian và công sức trong việc xử lý thư rác, đồng thời cải thiện trải nghiệm sử dụng email. Mô hình này có khả năng phân loại các email thành hai nhóm: thư rác và không phải thư rác, dựa trên các đặc trưng của email như từ khóa, địa chỉ email gửi, nội dung và tiêu đề.

Thuật toán Naive Bayes được chọn vì tính đơn giản và hiệu quả của nó trong việc xử lý các bài toán phân loại. Nó có khả năng làm việc tốt với các tập dữ liệu lớn và có thể được áp dụng trong nhiều lĩnh vực khác nhau.

Tuy nhiên, cần lưu ý rằng Naive Bayes có giả định "naive" về sự độc lập giữa các đặc trưng, do đó, nó có thể không hoạt động tốt trong các trường hợp mà giả định này không được thỏa mãn. Để đạt được kết quả tốt, việc tiền xử lý dữ liệu và chọn các đặc trưng phù hợp là rất quan trọng.

1.4./ Đối tượng và phạm vi nghiên cứu

Đối tượng của đề tài "Xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes" là xây dựng một mô hình có khả năng phân loại và lọc các email để xác định xem chúng có phải là thư rác hay không. Mô hình này có thể được áp dụng cho các hộp thư email của người dùng, tổ chức hoặc doanh nghiệp.

Phạm vi nghiên cứu của đề tài bao gồm:

* Thu thập dữ liệu: Nghiên cứu sẽ tập trung vào việc thu thập dữ liệu email từ các nguồn khác nhau, bao gồm cả email thư rác và email không phải thư rác. Dữ liệu này sẽ được sử dụng để huấn luyện và đánh giá mô hình.
* Tiền xử lý dữ liệu: Giai đoạn này bao gồm việc tiền xử lý dữ liệu email, như loại bỏ các từ không cần thiết, chuẩn hóa dữ liệu và chuyển đổi dữ liệu thành dạng phù hợp cho việc xây dựng mô hình.
* Xây dựng mô hình: Dữ liệu sau khi đã được tiền xử lý được sử dụng để xây dựng mô hình Naive Bayes. Mô hình này sẽ học từ dữ liệu huấn luyện và tạo ra các xác suất để phân loại các email là thư rác hoặc không phải thư rác.
* Kiểm tra và đánh giá mô hình: Mô hình được kiểm tra bằng cách sử dụng dữ liệu kiểm tra. Kết quả được đánh giá dựa trên các độ đo như độ chính xác, độ phủ và độ F1.

Mục tiêu chính của đề tài là xây dựng một mô hình lọc thư rác hiệu quả bằng thuật toán Naive Bayes, giúp người dùng nhận biết và loại bỏ thư rác một cách tự động.

1.5./ Tầm quan trọng của đề tài

Tầm quan trọng của đề tài "Xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes" là giúp tự động phân loại và lọc các email để xác định xem chúng có phải là thư rác hay không. Đây là một vấn đề quan trọng trong việc quản lý email và bảo vệ người dùng khỏi thư rác.

Việc xây dựng mô hình lọc thư rác có nhiều ứng dụng và tầm quan trọng như sau:

* Tiết kiệm thời gian và công sức: Mô hình lọc thư rác giúp tự động phân loại và loại bỏ thư rác, giúp người dùng tiết kiệm thời gian và công sức trong việc xử lý email hàng ngày.
* Cải thiện trải nghiệm sử dụng email: Thư rác gây phiền toái và làm giảm hiệu quả sử dụng email. Mô hình lọc thư rác giúp người dùng nhận được các email quan trọng và hữu ích, cải thiện trải nghiệm sử dụng email.
* Bảo vệ an ninh và quyền riêng tư: Thư rác có thể chứa các liên kết độc hại, mã độc hoặc lừa đảo. Mô hình lọc thư rác giúp ngăn chặn các email độc hại và bảo vệ an ninh và quyền riêng tư của người dùng.
* Tăng cường hiệu suất làm việc: Với mô hình lọc thư rác hoạt động hiệu quả, người dùng có thể tập trung vào các email quan trọng và công việc chính, tăng cường hiệu suất làm việc.
* Áp dụng rộng rãi: Mô hình lọc thư rác có thể được áp dụng trong nhiều lĩnh vực và tổ chức khác nhau, từ cá nhân đến doanh nghiệp và tổ chức chính phủ.

Việc xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes có tầm quan trọng lớn trong việc giúp tự động phân loại và lọc thư rác, tiết kiệm thời gian, cải thiện trải nghiệm sử dụng email và bảo vệ an ninh và quyền riêng tư của người dùng.

CHƯƠNG 2./ GIẢI QUYẾT BÀI TOÁN

2.1./ Cơ sở lý thuyết

Xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes dựa trên lý thuyết của thuật toán Naive Bayes và lý thuyết xác suất để phân loại và lọc thư rác dựa trên xác suất và độc lập tuyến tính của các đặc trưng.

Naive Bayes là một thuật toán phân loại dựa trên lý thuyết xác suất và nguyên tắc Bayes. Thuật toán này giả định rằng các đặc trưng của dữ liệu là độc lập tuyến tính và sử dụng xác suất để tính toán xác suất một điểm dữ liệu thuộc vào một lớp cụ thể.

Cụ thể, trong việc lọc thư rác, mô hình Naive Bayes sẽ học từ dữ liệu huấn luyện để xác định xác suất một email là thư rác hoặc không phải thư rác dựa trên các đặc trưng của email như từ khóa, địa chỉ email gửi, nội dung và tiêu đề. Mô hình sử dụng nguyên tắc Bayes để tính toán xác suất này.

Mặc dù có giả định "naive" về sự độc lập giữa các đặc trưng, Naive Bayes vẫn cho kết quả tốt trong nhiều trường hợp thực tế. Điều này làm cho nó trở thành một lựa chọn phổ biến trong việc xây dựng mô hình lọc thư rác.

2.1.1./ Nguyên tắc Bayes

Nguyên tắc Bayes là một nguyên tắc trong lý thuyết xác suất, được đặt tên theo nhà toán học Thomas Bayes. Nguyên tắc này cho phép chúng ta cập nhật xác suất của một sự kiện dựa trên dữ liệu mới.

Công thức Bayes được sử dụng để tính xác suất có điều kiện của một sự kiện A khi đã biết sự kiện B đã xảy ra. Công thức này được biểu diễn như sau:

**P(A|B) = (P(B|A) \* P(A)) / P(B)**

Trong đó:

* P(A|B) là xác suất có điều kiện của sự kiện A đã biết sự kiện B đã xảy ra.
* P(B|A) là xác suất có điều kiện của sự kiện B đã biết sự kiện A đã xảy ra.
* P(A) và P(B) là xác suất không có điều kiện của sự kiện A và B.

2.1.2./ Naive Bayes trong lọc thư rác

Thuật toán Naive Bayes là một phương pháp phân loại dựa trên lý thuyết xác suất và nguyên tắc Bayes. Nó được gọi là "naive" vì giả định rằng các đặc trưng của dữ liệu là độc lập tuyến tính, tức là không có sự phụ thuộc giữa chúng.

Trong việc lọc thư rác, thuật toán Naive Bayes được áp dụng để phân loại một email là thư rác hay không. Để làm điều này, thuật toán tính toán xác suất một email thuộc vào mỗi lớp (thư rác hoặc không phải thư rác) dựa trên các đặc trưng của email.

Thuật toán Naive Bayes giả định rằng các đặc trưng là độc lập tuyến tính, điều này có nghĩa là xác suất của mỗi đặc trưng không phụ thuộc vào các đặc trưng khác. Mặc dù giả định này không thực tế trong nhiều trường hợp, Naive Bayes vẫn cho kết quả tốt trong nhiều bài toán thực tế, bao gồm lọc thư rác.

Mô hình Naive Bayes giả định rằng các đặc trưng của email là độc lập tuyến tính. Điều này có nghĩa là giả định rằng mỗi đặc trưng (ví dụ: từ khóa, địa chỉ email gửi, nội dung) không phụ thuộc vào nhau.

Các bước để xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes:

* Chuẩn bị dữ liệu huấn luyện: Thu thập một tập dữ liệu có chứa các email đã được gán nhãn là thư rác hoặc không phải thư rác.
* Tiền xử lý dữ liệu: Trước khi áp dụng thuật toán Naive Bayes, dữ liệu cần được tiền xử lý, bao gồm loại bỏ các ký tự đặc biệt, chuyển đổi văn bản thành các đặc trưng số học, và tiêu chuẩn hóa dữ liệu.
* Xây dựng mô hình Naive Bayes: Áp dụng thuật toán Naive Bayes để tính toán xác suất một email là thư rác hoặc không phải thư rác dựa trên các đặc trưng của email. Công thức Bayes được sử dụng để tính toán xác suất này.
* Đánh giá và tinh chỉnh mô hình: Sử dụng các phương pháp đánh giá mô hình như cross-validation để đánh giá hiệu suất của mô hình và tinh chỉnh các tham số để cải thiện mô hình.
* Áp dụng mô hình lọc thư rác: Sử dụng mô hình đã xây dựng để phân loại các email mới là thư rác hoặc không phải thư rác dựa trên xác suất tính toán.

Thông qua việc tính toán xác suất dựa trên nguyên tắc Bayes và giả định về độc lập tuyến tính của các đặc trưng, mô hình Naive Bayes có thể phân loại và lọc thư rác một cách hiệu quả.

2.2./ Thuật toán sử dụng

Thuật toán Naive Bayes là một thuật toán phân loại dựa trên nguyên tắc của định lý Bayes. Nó được sử dụng rộng rãi trong các ứng dụng lọc thư rác. Thuật toán này dựa trên giả định "naive" rằng các đặc trưng của dữ liệu độc lập với nhau.

Để xây dựng mô hình lọc thư rác bằng thuật toán Naive Bayes, ta cần có một tập dữ liệu huấn luyện và một tập dữ liệu kiểm tra. Tập dữ liệu huấn luyện sẽ được sử dụng để huấn luyện mô hình, trong khi tập dữ liệu kiểm tra sẽ được sử dụng để đánh giá hiệu suất của mô hình.

Thuật toán Naive Bayes dựa trên việc tính toán xác suất của một mẫu dữ liệu thuộc vào từng lớp. Cụ thể, thuật toán tính toán xác suất có điều kiện của mỗi đặc trưng dựa trên lớp của mẫu dữ liệu và sử dụng các xác suất này để dự đoán lớp của mẫu dữ liệu mới.

Ví dụ về sử dụng thuật toán Naive Bayes trong mô hình lọc thư rác:

Giả sử ta có một tập dữ liệu huấn luyện gồm các mẫu thư rác và các mẫu thư không rác. Mỗi mẫu dữ liệu được biểu diễn bằng các đặc trưng như từ khóa, độ dài, v.v. Ta sử dụng thuật toán Naive Bayes để huấn luyện mô hình và sau đó sử dụng mô hình này để phân loại các mẫu dữ liệu mới là thư rác hay không.

Để sử dụng thuật toán Naive Bayes trong Python, ta có thể sử dụng thư viện scikit-learn. Dưới đây là một ví dụ về mã code sử dụng thuật toán Naive Bayes trong scikit-learn:

|  |
| --- |
| from sklearn.naive\_bayes import MultinomialNB  from sklearn.feature\_extraction.text import CountVectorizer  # Tạo vector đặc trưng từ tập dữ liệu huấn luyện  vectorizer = CountVectorizer()  X\_train = vectorizer.fit\_transform(train\_data)  # Huấn luyện mô hình Naive Bayes  model = MultinomialNB()  model.fit(X\_train, train\_labels)  # Áp dụng mô hình vào tập dữ liệu kiểm tra  X\_test = vectorizer.transform(test\_data)  predicted\_labels = model.predict(X\_test) |

Trong ví dụ trên, ta sử dụng lớp `MultinomialNB` trong scikit-learn để tạo và huấn luyện mô hình Naive Bayes. Ta sử dụng lớp `CountVectorizer` để tạo vector đặc trưng từ tập dữ liệu văn bản. Sau đó, ta sử dụng mô hình đã huấn luyện để dự đoán nhãn của các mẫu dữ liệu trong tập kiểm tra.

2.3./ Thiết kế mô hình

**Thu thập dữ liệu:** Thu thập dữ liệu từ nguồn tương ứng với vấn đề cụ thể mà bạn muốn giải quyết. Dữ liệu này nên bao gồm các thông tin đầu vào và nhãn cho việc huấn luyện và kiểm thử mô hình.

**Tiền xử lý dữ liệu:** Làm sạch và chuẩn hóa dữ liệu. Điều này có thể bao gồm loại bỏ dữ liệu nhiễu, chia thành các tập huấn luyện và kiểm thử, biểu diễn dữ liệu theo định dạng thích hợp cho mô hình Naive Bayes.

**Tạo từ vựng (Vocabulary):** Xây dựng danh sách từ vựng từ dữ liệu huấn luyện. Điều này bao gồm việc tách từ (tokenization) và loại bỏ từ dừng (stop words) nếu cần.

**Tính toán xác suất:** Tính toán các xác suất cần thiết cho mô hình Naive Bayes, bao gồm xác suất trước (prior probabilities), xác suất điều kiện (conditional probabilities), và tham số Smoothing (nếu áp dụng Laplace smoothing).

**Xây dựng mô hình:** Sử dụng xác suất tính toán được để xây dựng mô hình Naive Bayes. Mô hình này sẽ sử dụng xác suất để phân loại các thông tin đầu vào vào các nhãn hoặc lớp khác nhau.

**Đào tạo mô hình:** Sử dụng tập dữ liệu huấn luyện để đào tạo mô hình Naïve Bayes. Mô hình sẽ học từ các xác suất đã tính toán trước đó.

**Đánh giá mô hình:** Sử dụng tập dữ liệu kiểm thử để đánh giá hiệu suất của mô hình. Các phép đo như độ chính xác (accuracy), độ nhạy (recall), độ chính xác (precision), và F1-score có thể được sử dụng để đánh giá mô hình.

**Tinh chỉnh mô hình:** Nếu mô hình không đạt hiệu suất mong muốn, bạn có thể tinh chỉnh các tham số hoặc thử các biến thể khác của Naive Bayes.

**Triển khai mô hình:** Khi mô hình đã được đào tạo và đánh giá, bạn có thể triển khai nó để thực hiện phân loại cho các dữ liệu mới.

**Theo dõi và duy trì mô hình:** Mô hình cần được duy trì và cập nhật theo thời gian để duy trì hiệu suất.

CHƯƠNG 3./   
XÂY DỰNG BỘ LỌC THƯ RÁC BẰNG NAIVE BAYES

3.1./ Bộ dữ liệu

Bộ dữ liệu "SMSSpamCollection" là một tập dữ liệu văn bản được sử dụng để phát triển và đánh giá các mô hình máy học cho việc phân loại tin nhắn SMS thành hai loại chính: tin rác (spam) và tin thường (ham). Dưới đây là một mô tả chi tiết hơn về bộ dữ liệu này:

* Bộ dữ liệu chứa tổng cộng 5.572 mẫu tin nhắn SMS, được chia thành hai lớp: spam và ham. Số lượng mẫu của mỗi lớp có thể khác nhau.
* Mỗi mẫu dữ liệu trong tập dữ liệu này bao gồm hai phần chính:
  + Văn bản tin nhắn SMS: Đây là nội dung của tin nhắn SMS, được biểu diễn dưới dạng chuỗi văn bản.
  + Nhãn (Label): Mỗi mẫu được gán một nhãn để chỉ ra liệu nó có phải là tin rác (spam) hay không (ham). Nhãn này có thể là "spam" hoặc "ham."
* Bộ dữ liệu có một phân phối không đều giữa hai lớp. Thường thì số lượng mẫu spam ít hơn so với số lượng mẫu ham, vì một phần lớn các tin nhắn SMS là tin thường.
* Bộ dữ liệu "SMSSpamCollection" thường được sử dụng để xây dựng và đào tạo các mô hình máy học để phát hiện tin rác trong hộp thư đến hoặc trên điện thoại di động. Nó có thể được sử dụng trong các ứng dụng gửi SMS tự động hoặc trong các hệ thống chặn tin rác.
* Bộ dữ liệu này thường được sử dụng trong các khóa học và dự án về học máy và xử lý ngôn ngữ tự nhiên để phát triển và đánh giá các mô hình phân loại văn bản. Nó cung cấp một ví dụ thực tế và phù hợp để nghiên cứu và phát triển các mô hình phân loại văn bản dựa trên dữ liệu thực tế.

Bộ dữ liệu "SMSSpamCollection" là một nguồn tài liệu quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên và phân loại văn bản, và nó đã đóng góp đáng kể vào việc phát triển các công cụ phát hiện tin rác trong tin nhắn SMS.

[Link bộ dữ liệu "SMSSpamCollection"](https://drive.google.com/file/d/11QTBX2iCpzBTAprdCi0fnggtBudERs2b/view?usp=sharing)

3.2./ Xây dựng bộ lọc thư rác bằng Naive Bayes

Sử dụng thư viện Pandas để đọc tệp CSV chứa dữ liệu về tin nhắn SMS và nhãn từ một đường dẫn cụ thể. Sau đó, nó tạo một DataFrame có tên "sms\_spam" với hai cột "Label" và "SMS" và in ra kích thước của DataFrame cũng như năm hàng đầu tiên của dữ liệu để kiểm tra.

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

Hình 3. 1./ Ví dụ minh hoạ sử dụng thư viện Pandas để đọc dữ liệu csv

Output:

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 3. 2./ Kết quả của sử dụng thư viện Pandas để đọc dữ liệu csv

Tiếp theo tính và hiển thị tỷ lệ phần trăm của giá trị "spam" và "ham" trong cột "Label" của DataFrame "sms\_spam". Điều này giúp bạn hiểu phân phối của dữ liệu giữa hai loại tin nhắn: tin rác ("spam") và tin thường ("ham").



Hình 3. 3./ Ví dụ hiển thị tỉ lệ phần trăm

Output:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình 3. 4./ Kết quả của hiển thị tỉ lệ phần trăm

Kết quả sau khi thực hiện sms\_spam['Label'].value\_counts(normalize=True) cho thấy tỷ lệ phần trăm giữa các giá trị trong cột "Label" của DataFrame "sms\_spam" như sau:

"ham" (tin thường) chiếm khoảng 86.59% của dữ liệu, được biểu diễn bằng 0.8659 (86.59%).

"spam" (tin rác) chiếm khoảng 13.41% của dữ liệu, được biểu diễn bằng 0.1341 (13.41%).

Điều này cho thấy rằng phân phối của dữ liệu chứa nhiều tin nhắn "ham" hơn "spam," với hơn 86% là tin thường và khoảng 13% là tin rác.

Tiếp theo Tạo một phiên bản ngẫu nhiên của tập dữ liệu "sms\_spam" bằng cách xáo trộn nó.Tính chỉ mục để chia tập dữ liệu thành tập huấn luyện (80%) và tập kiểm tra (20%).Tạo tập dữ liệu huấn luyện và tập dữ liệu kiểm tra từ dữ liệu đã xáo trộn, đồng thời thiết lập lại chỉ mục.In ra kích thước của tập dữ liệu huấn luyện và tập dữ liệu kiểm tra để kiểm tra kích thước của chúng.

Ảnh có chứa văn bản, phần mềm, đa phương tiện, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 3. 5./ Ví dụ tạo một biến ngẫu nhiên

Output:



Hình 3. 6./ Kết quả của tạo biến ngẫu nhiên

Tiếp theo chúng ta kiểm tra phân phối của dữ liệu giữa hai loại tin nhắn, tức là tin rác ("spam") và tin thường ("ham"), trong tập dữ liệu huấn luyện và tập dữ liệu kiểm tra. Điều này giúp xác định mức độ cân bằng giữa các lớp trong dữ liệu, điều quan trọng trong quá trình huấn luyện và đánh giá mô hình máy học.

Nếu phân phối giữa hai lớp quá mất cân bằng (ví dụ: một lớp chiếm hơn 90% của dữ liệu), có thể dẫn đến mô hình bị thiên lệch và không tốt trong việc phân loại các lớp. Bằng cách kiểm tra tỷ lệ phần trăm của mỗi lớp trong tập huấn luyện và tập kiểm tra, bạn có thể đảm bảo rằng dữ liệu được chia đều giữa các lớp, giúp cải thiện hiệu suất của mô hình phân loại.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 3. 7./ Ví dụ kiểm tra trong tập huấn luyện

Kết quả đúng như mong đợi

Thuật toán Naive Bayes được sử dụng để phân loại tin nhắn SMS thành hai lớp: tin rác (Spam) và tin thường (Ham). Để làm điều này, thuật toán sử dụng hai phương trình chính:

1.Phương trình để tính xác suất tin nhắn là tin rác (Spam) dựa trên từng từ trong tin nhắn:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình 3. 8./ Phương trình xác suất là tin nhắn rác

2. Phương trình để tính xác suất tin nhắn là tin thường (Ham) dựa trên từng từ trong tin nhắn:

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ

Mô tả được tạo tự động

Hình 3. 9./ Phương trình xác suất là tin nhắn thường

Ở đây:

* P(Spam) và P(Ham) là xác suất trước (prior probability) của tin rác và tin thường, tức là xác suất một tin nhắn bất kỳ là tin rác hoặc tin thường.
* P(wi|Spam) là xác suất xuất hiện từng từ wi trong tin rác.
* P(wi|Ham) là xác suất xuất hiện từng từ wi trong tin thường.
* N(wi|Spam) là số lần từ wi xuất hiện trong tin rác.
* N(wi|Ham) là số lần từ wi xuất hiện trong tin thường.
* NSpam là tổng số từ trong tất cả các tin rác.
* NHam là tổng số từ trong tất cả các tin thường.
* NVocabulary là tổng số từ vựng trong toàn bộ dữ liệu.

Để tính toán xác suất cho thuật toán Naive Bayes, chúng ta cần biến đổi dữ liệu từ định dạng ban đầu thành một bảng dữ liệu mới. Bảng dữ liệu mới này có các cột đại diện cho từng từ duy nhất trong từ vựng (vocabulary) của tất cả các câu. Mỗi hàng trong bảng này tương ứng với một tin nhắn và chứa thông tin về sự xuất hiện của từng từ trong tin nhắn.

* Mỗi tin nhắn được gán một nhãn, ví dụ "spam" hoặc "ham," để biểu thị liệu đó là tin rác hay tin thường.
* Các từ được biểu diễn ở dạng viết thường để không phân biệt chữ hoa và chữ thường.
* Thứ tự của các từ trong tin nhắn gốc không quan trọng.
* Dấu câu và ký tự đặc biệt không còn được tính đến.

Bằng cách biến đổi dữ liệu này, chúng ta sẽ dễ dàng tính toán xác suất xuất hiện của từng từ trong tin rác và tin thường, giúp xây dựng mô hình Naive Bayes để phân loại tin nhắn mới dựa trên xác suất này.

Bộ dữ liệu lúc đầu:

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động**

Hình 3. 10./ Kết quả bộ dữ liệu lúc đầu

Tiến hành làm sạch dữ liệu đó là đặt tất cả về chữ thường và loại bỏ dấu chấm câu

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, Phần mềm đồ họa

Mô tả được tạo tự động

Hình 3. 11./ Kết quả bộ dữ liệu đã làm sạch

Chúng ta tạo một "vocabulary" (từ vựng) bằng cách thực hiện các bước sau:

1.Chuyển đổi từng tin nhắn trong cột SMS thành danh sách các từ, sử dụng phương thức Series.str.split().

2.Khởi tạo một danh sách trống có tên là "vocabulary".

3.Lặp qua từng tin nhắn trong cột SMS đã chuyển đổi.

4.Lặp qua từng từ trong mỗi tin nhắn và thêm từng từ vào danh sách "vocabulary".

5.Chuyển danh sách "vocabulary" thành một tập hợp để loại bỏ các từ trùng lặp, sau đó chuyển nó trở lại thành một danh sách.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 3. 12./ Ví dụ tạo một "vocabulary"

Kết quả là có tổng cộng 7783 từ duy nhất trong từ vựng của tập dữ liệu huấn luyện. Điều này có nghĩa rằng có 7783 từ khác nhau trong tất cả các tin nhắn huấn luyện

Để chuyển đổi dữ liệu dựa trên từ vựng thành một DataFrame mới, chúng ta xây dựng một từ điển. Mỗi cột trong DataFrame mới tương ứng với một từ trong từ vựng, và mỗi hàng biểu thị một tin nhắn. Giá trị trong mỗi ô là số lần từ đó xuất hiện trong tin nhắn. Điều này giúp biểu diễn dữ liệu theo định dạng phù hợp để sử dụng trong mô hình Naive Bayes.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phần mềm đa phương tiện, phần mềm

Mô tả được tạo tự động

Hình 3. 13./ Ví dụ tạo dữ liệu Dataframe

Để tạo từ điển cho tập dữ liệu huấn luyện của bạn, bạn có thể sử dụng mã như sau:

1.Bắt đầu bằng việc khởi tạo một từ điển có tên là word\_counts\_per\_sms, trong đó mỗi key là một từ duy nhất từ từ vựng và mỗi giá trị là một danh sách có độ dài bằng số tin nhắn trong tập huấn luyện, với mỗi phần tử trong danh sách có giá trị ban đầu là 0.

2.Sử dụng một vòng lặp qua từng tin nhắn trong tập dữ liệu huấn luyện, với sự trợ giúp của hàm enumerate() để có được cả chỉ số và nội dung của tin nhắn (index và sms).

3.Sử dụng một vòng lặp lồng nhau, lặp qua từng từ trong tin nhắn (trong đó tin nhắn được biểu diễn dưới dạng danh sách các chuỗi, mỗi chuỗi tương ứng với một từ).

4.Tăng giá trị tương ứng trong word\_counts\_per\_sms[word][index] lên 1 để đếm số lần xuất hiện của từ đó trong tin nhắn tương ứng.

Kết quả sẽ là một từ điển chứa số lần xuất hiện của mỗi từ trong từng tin nhắn trong tập dữ liệu huấn luyện.

Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, ảnh chụp màn hình

Mô tả được tạo tự động

Hình 3. 14./ Ví dụ code đếm số lần từ xuất hiện

Bây giờ đã có các từ điển chúng ta cần, tiếp theo chúng ta tiến hành chuyển đổi cuối cùng đối với tập huấn luyện.

Ảnh có chứa ảnh chụp màn hình, văn bản, bàn phím, máy tính

Mô tả được tạo tự động

Hình 3. 15./ Chuyển đổi cuối cùng với tập dữ liệu

Chúng ta có thể thấy cột Label bị thiếu, để bổ sung cột "Label" vào DataFrame mà bạn vừa tạo (chứa số lần xuất hiện của từng từ trong từng tin nhắn), bạn có thể sử dụng hàm pd.concat() để nối DataFrame mới này với DataFrame chứa tập huấn luyện. Bằng cách này, bạn sẽ có cả cột "Label" và các cột "SMS" trong DataFrame mới, giúp bạn xây dựng mô hình Naive Bayes dựa trên dữ liệu đã chuẩn bị.

Ảnh có chứa ảnh chụp màn hình, văn bản, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 3. 16./ Ví dụ thêm cột Label

Như vậy là chúng ta đã hoàn thành việc làm sạch tập dữ liệu huấn luyện.

Tiếp theo Để sử dụng thuật toán Naive Bayes để phân loại tin nhắn là rác hoặc không là rác, trước hết chúng ta cần tính toán các giá trị quan trọng sau:

* P(Spam) và P(Ham): Xác suất trước của tin nhắn là rác và không phải là rác.
* NSpam và NHam: Tổng số từ trong tất cả các tin nhắn rác và không phải là rác trong tập huấn luyện.
* NVocabulary: Số lượng từ duy nhất trong từ vựng của chúng ta (7783 từ).
* Laplace Smoothing (α): Thiết lập α=1 để tránh xác suất 0 cho từ không xuất hiện trong lớp tương ứng.

Các giá trị này sẽ được tính toán dựa trên tập dữ liệu huấn luyện và sau đó sử dụng trong thuật toán Naive Bayes để phân loại tin nhắn.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 3. 17./ Ví dụ sử dụng thuật toán naïve bayes

Các tham số P(wi|Spam) và P(wi|Ham) đại diện cho xác suất có điều kiện của một từ wi xuất hiện trong tin nhắn spam và không phải là spam. Các giá trị này được tính bằng cách sử dụng công thức có Laplace smoothing để tránh xác suất 0 khi từ đó không xuất hiện trong một lớp cụ thể. Công thức tính toán tham số này bằng cách thêm một đơn vị vào số lần từ wi xuất hiện trong lớp và thêm α vào tổng số từ trong lớp nhân với số lượng từ duy nhất trong từ vựng. Các tham số này sẽ được sử dụng trong mô hình Naive Bayes để tính toán xác suất của một tin nhắn cụ thể là spam hoặc không phải spam dựa trên từng từ xuất hiện trong tin nhắn.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Hình 3. 18./ ví dụ code sử dụng công thức có Laplace smoothing

Hàm classify() sẽ thực hiện việc phân loại tin nhắn là rác hoặc không phải rác dựa trên các xác suất đã tính toán. Lưu ý các bước sau:

1.Biến message đầu vào cần phải là một chuỗi.

2.Trước khi tính toán xác suất, chúng ta làm sạch dữ liệu trên chuỗi message bằng cách loại bỏ dấu chấm câu, chuyển tất cả chữ cái thành chữ thường và chia chuỗi thành danh sách các từ.

3.Tính toán xác suất P(Spam|message) và P(Ham|message) dựa trên từng từ trong danh sách từ của message.

4.So sánh xác suất P(Ham|message) với P(Spam|message). Nếu xác suất của lớp "Ham" lớn hơn, thì tin nhắn được phân loại là "ham". Nếu xác suất của lớp "Spam" lớn hơn, thì tin nhắn được phân loại là "spam".

5.Nếu xác suất của cả hai lớp bằng nhau, thuật toán có thể yêu cầu sự trợ giúp của con người để phân loại.

Hàm classify() là bước cuối cùng trong quá trình xây dựng bộ lọc spam và sẽ giúp bạn phân loại tin nhắn dựa trên xác suất.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 3. 19./ Ví dụ code hàm classify()

Bây giờ chúng ta sẽ tiến hành kiểm tra bộ lọc thư rác trên 2 thư mới. 1 là spam và 2 là ham

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động

Hình 3. 20./ Kiểm tra bộ lọc thư rác

Kết quả khá tốt, nhưng chúng ta sẽ thử xem bộ lọc hoạt động tốt như thế nào trên tập thử với 1.114 tin nhắn.

Chúng ta sẽ bắt đầu viết 1 hàm trả về các nhãn phân loại thay vì in ra.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Hình 3. 21./ Ví dụ hàm trả về nhãn phân loại

Sau khi đã có 1 hàm trả về các nhãn chúng ta sẽ sử dụng nó để tạo 1 cột mới trong tập kiểm thử.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

Hình 3. 22./ Tạo một cột mới trong tập kiểm thử

Để do lường mức độ hiệu quả của bộ lọc thư rác trong việc phân loại các thư chúng ta có thể so sánh các giá trị dự đoán với các giá trị thực tế bằng cách sử dụng accuracy làm chỉ số. Accuracy (độ chính xác) là một phép đo đánh giá hiệu suất của bộ lọc thư rác. Nó đo lường tỷ lệ phần trăm của các tin nhắn được phân loại đúng trong tổng số tin nhắn được phân loại. Độ chính xác càng cao, bộ lọc càng hiệu quả trong việc phân loại tin nhắn.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Hình 3. 23./ So sáng giá trị dự đoán với giá trị thực tế

Rất tốt khi độ chính xác đạt gần 98.74%! Bộ lọc thư rác đã xử lý 1,114 tin nhắn mới và phân loại chính xác 1,100 tin nhắn trong số đó. Điều này thể hiện hiệu suất ấn tượng của mô hình Naive Bayes.

Như vậy là chúng ta đã hoàn thành việc xây dựng bộ lọc thư rác bằng Naive Bayes và củng cố kiến thức về thuật toán phân loại Naive Bayes. Hy vọng rằng chúng ta sẽ tiếp tục khám phá và ứng dụng kiến thức này trong các dự án và bài viết tiếp theo!

3.3./ Kết luận

Trong quá trình xây dựng bộ lọc thư rác bằng Naive Bayes, chúng ta đã thực hiện nhiều bước quan trọng để huấn luyện và đánh giá hiệu suất của mô hình. Dưới đây là một số điểm quan trọng:

1.Chuẩn bị dữ liệu: Chúng ta đã chuẩn bị dữ liệu bằng cách làm sạch, chia thành từng từ, và xây dựng từ vựng. Điều này giúp mô hình Naive Bayes làm việc với dữ liệu dễ dàng hơn.

2.Huấn luyện mô hình: Chúng ta đã tính toán các tham số quan trọng của mô hình Naive Bayes, bao gồm xác suất tiên nghiệm và xác suất điều kiện của các từ trong tin nhắn. Điều này đã cho phép chúng ta xây dựng một mô hình phân loại tốt.

3.Phân loại tin nhắn: Chúng ta đã viết hàm classify() để phân loại các tin nhắn dựa trên xác suất được tính toán. Mô hình đã phân loại tin nhắn là rác hoặc không phải rác dựa trên xác suất cao nhất.

4.Đánh giá hiệu suất: Chúng ta đã sử dụng độ chính xác để đánh giá hiệu suất của mô hình. Độ chính xác đo lường tỷ lệ phần trăm của các tin nhắn được phân loại đúng trong tổng số tin nhắn được phân loại.

Kết quả của mô hình Naive Bayes cho thấy mức độ hiệu quả cao với độ chính xác gần 98.74%. Mô hình đã xử lý nhiều tin nhắn mới và phân loại chính xác hầu hết trong số chúng. Điều này thể hiện sức mạnh của mô hình Naive Bayes trong việc phân loại thư rác.

Hy vọng rằng bài viết này đã giúp bạn hiểu hơn về cách xây dựng một bộ lọc thư rác bằng Naive Bayes và củng cố kiến thức của bạn về thuật toán này. Hãy tiếp tục áp dụng kiến thức này vào các dự án thực tế và nghiên cứu tiếp theo!

3.4./ Ưu điểm và nhược điểm

Mô hình phân loại Naive Bayes có một số ưu điểm quan trọng, đặc biệt trong các tình huống cụ thể. Dưới đây là một số ưu điểm quan trọng của mô hình Naive Bayes:

* Đơn giản và dễ triển khai: Naive Bayes là một thuật toán đơn giản và dễ triển khai. Nó dựa trên một số giả thiết cơ bản về sự độc lập giữa các biến đầu vào, làm cho quá trình huấn luyện và dự đoán đơn giản.
* Hiệu suất tốt với dữ liệu lớn: Mô hình Naive Bayes có thể hoạt động hiệu quả trên dữ liệu lớn. Nó có thể xử lý hàng trăm nghìn hoặc thậm chí triệu mẫu dữ liệu mà không đòi hỏi nhiều tài nguyên tính toán.
* Tính nhanh và tiết kiệm tài nguyên: Naive Bayes là một thuật toán nhanh và tiết kiệm tài nguyên, thích hợp cho các ứng dụng yêu cầu xử lý thời gian thực hoặc với nguồn dữ liệu lớn.
* Hiệu suất tốt với dữ liệu thưa thớt: Naive Bayes thường hoạt động tốt với dữ liệu thưa thớt, nghĩa là với số lượng biến đầu vào lớn và các giá trị thiếu, nó vẫn có thể tạo ra dự đoán hợp lý.
* Dễ dàng cập nhật mô hình: Naive Bayes cho phép cập nhật mô hình dễ dàng khi có dữ liệu mới, điều này làm cho nó thích hợp cho các ứng dụng theo thời gian thực.
* Hiệu suất tốt trong phân loại văn bản: Mô hình Naive Bayes thường là lựa chọn hàng đầu trong việc phân loại văn bản, chẳng hạn như phân loại thư rác (spam) hoặc phân loại văn bản dựa trên ngữ nghĩa.

Mặc dù mô hình Naive Bayes có nhiều ưu điểm, nhưng cũng tồn tại một số nhược điểm:

* Giả định về độc lập: Naive Bayes giả định rằng các biến đầu vào là độc lập, điều này thường không phản ánh thực tế trong nhiều tình huống thực tế. Trong thực tế, các biến thường tương quan với nhau, điều này có thể dẫn đến các dự đoán không chính xác nếu các biến không độc lập.
* Tính chủ quan của dự đoán: Naive Bayes dựa vào giả định đối với xác suất để tạo dự đoán, điều này có thể tạo ra dự đoán chủ quan và không chính xác nếu các giả định không đúng.
* Xử lý các biến liên tục: Naive Bayes là một thuật toán dựa trên xác suất và thường được thiết kế cho các biến phân loại. Khi sử dụng Naive Bayes với các biến liên tục, chúng ta thường cần chia chúng thành các khoảng (binning) hoặc sử dụng các biểu đồ phân phối để chuyển chúng thành biến phân loại, điều này có thể dẫn đến mất mát thông tin.
* Khả năng xử lý từ vựng lớn: Nếu số lượng từ trong từ vựng (vocabulary) quá lớn, Naive Bayes có thể trở nên không hiệu quả và yêu cầu nhiều tài nguyên tính toán.
* Yêu cầu lượng dữ liệu lớn cho các xác suất đáng tin cậy: Đối với các vấn đề có số lượng biến đầu vào lớn, Naive Bayes yêu cầu một lượng dữ liệu đủ lớn để ước lượng các xác suất đáng tin cậy. Trong các trường hợp dữ liệu hạn chế, điều này có thể dẫn đến việc dự đoán không chính xác.
* Không xử lý được mối quan hệ phi tuyến tính: Naive Bayes giả định rằng mối quan hệ giữa các biến đầu vào là tuyến tính. Điều này có nghĩa là nó không thể xử lý các mối quan hệ phi tuyến tính một cách hiệu quả.

3.5./ Hướng khắc phục

Dưới đây là một số cách để khắc phục hoặc giảm thiểu các nhược điểm của mô hình Naive Bayes:

* Xử lý các biến liên tục: Để xử lý các biến liên tục, bạn có thể sử dụng kỹ thuật như binning (chia thành các khoảng) hoặc sử dụng phân phối xác suất để chuyển chúng thành các biến phân loại.
* Sử dụng phiên bản cải tiến: Có nhiều biến thể của mô hình Naive Bayes, bao gồm Multinomial Naive Bayes, Gaussian Naive Bayes và Complement Naive Bayes, mà có thể phù hợp hơn với dữ liệu cụ thể hoặc giảm một số giả định cơ bản.
* Feature Engineering: Tạo các biến phụ (feature engineering) có thể cải thiện hiệu suất của mô hình Naive Bayes. Ví dụ, tạo các biến dummy, trích xuất thông tin quan trọng từ dữ liệu thô hoặc sử dụng kỹ thuật như TF-IDF trong phân loại văn bản.
* Lọc từ vựng (Vocabulary Reduction): Nếu số từ trong từ vựng quá lớn, hãy xem xét việc lọc bỏ các từ ít quan trọng hoặc sử dụng phương pháp như L1 regularization để chọn ra các biến quan trọng hơn.
* Thu thập thêm dữ liệu: Nếu mô hình Naive Bayes không hoạt động tốt với lượng dữ liệu hiện có, hãy cố gắng thu thập thêm dữ liệu để cải thiện ước lượng xác suất.
* Xử lý các mối quan hệ phi tuyến tính: Nếu có mối quan hệ phi tuyến tính trong dữ liệu, hãy xem xét sử dụng mô hình khác, chẳng hạn như máy vector hỗ trợ (Support Vector Machine) hoặc mạng nơ-ron.
* Kiểm tra giả định độc lập: Kiểm tra cẩn thận các giả định về sự độc lập giữa các biến đầu vào. Nếu cần, sử dụng phương pháp thống kê hoặc phân tích để xác định mức độ tương quan giữa các biến.
* Tinh chỉnh hyperparameters: Nếu bạn sử dụng các biến thể của Naive Bayes, hãy tinh chỉnh hyperparameters để tối ưu hóa hiệu suất của mô hình.
* Kết hợp với các mô hình khác: Trong một số trường hợp, kết hợp mô hình Naive Bayes với các mô hình khác có thể tạo ra hiệu suất tốt hơn.

3.6./ Kết luận thuật toán Naïve Bayes

Naive Bayes là một thuật toán phân loại mạnh mẽ trong nhiều tình huống, đặc biệt là trong phân loại văn bản và ứng dụng liên quan đến xác suất. Tuy nhiên, nó không phải lúc nào cũng là lựa chọn tốt, và cần xem xét cẩn thận giữa ưu điểm và nhược điểm của nó trong từng trường hợp cụ thể. Nếu giả định độc lập không được đáp ứng hoặc nếu bạn có dữ liệu có tính tương quan cao, có thể cần xem xét các mô hình khác như máy vector hỗ trợ (SVM) hoặc mạng nơ-ron.

TÀI LIỆU THAM KHẢO

[1] "Pattern Classification" by Richard O. Duda, Peter E. Hart, and David G. Stork - Cuốn sách này cung cấp kiến thức nền tảng về các thuật toán phân loại, bao gồm Naive Bayes.

[2] "Introduction to Information Retrieval" by Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze - Sách này giới thiệu về phân loại văn bản và cách Naive Bayes được áp dụng trong lĩnh vực này.

[3] "Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy - Cuốn sách này giúp bạn hiểu sâu hơn về các mô hình xác suất và Naive Bayes trong ngữ cảnh học máy.

[4] \*\*"Scikit-learn Documentation" (<https://scikit-learn.org/stable/documentation.html>) - Tài liệu chính thức của thư viện scikit-learn, bao gồm hướng dẫn về cách sử dụng Naive Bayes trong Python.

[5] "A practical explanation of a Naive Bayes classifier" by Jan Van Ryswyck - Bài viết này giúp bạn hiểu cách hoạt động của Naive Bayes và cách áp dụng nó trong thực tế.

[6] "A Gentle Introduction to Naïve Bayes" by Andrew Ng - Khóa học trực tuyến này giúp bạn hiểu về lý thuyết và cách thực hiện Naive Bayes.

[7] "The Unreasonable Effectiveness of Naive Bayes Classifiers" by Jason Rennie - Bài báo này giới thiệu về sự hiệu quả của Naive Bayes trong phân loại văn bản.

[8] Tài liệu, Slide bài giảng ThS. Phạm Đình Tài, giảng viên môn Chuyên đề Tốt nghiệp Khoa học dữ liệu Khoa Công nghệ thông tin, Trường Đại học Nguyễn Tất Thành

[9] Thuật toán phân loại Naive Bayes và ứng dụng, https://whitehat.vn/threads/thuat-toan-phan-loai-naive-bayes-va-ung-dung.13775/

[10] Xây dựng mô hình lọc thư rác bằng Naive Bayes, https://viblo.asia/p/xay-dung-mo-hinh-loc-thu-rac-bang-naive-bayes-Ljy5Vqxklra