E-mail Categorization using KNN, Bernoulli Naive Bayes, and Logistic Regression

Báo cáo môn Machine Learning

Nguyễn Duy Vũ - 22000132

Vũ Đức Duy - 22000078

Đoàn Phạm Ngọc Linh - 22000102

Giảng viên hướng dẫn: Thầy Cao Văn Chung

Trường Đại học Khoa học Tự nhiên – ĐHQGHN
Khoa Toán – Cơ – Tin học
Ngành Toán Tin
Học phần: Machine Learning

Hà Nội, tháng 4 năm 2025

Contents

1 Tóm tắt

Báo cáo này tập trung vào việc xây dựng hệ thống phân loại email tự động dựa trên ba thuật toán học máy: **K-Nearest Neighbors (KNN)**, **Bernoulli Naive Bayes** và **Logistic Regression**. Mục tiêu là phân loại email vào các nhóm chủ đề như công việc, quảng cáo, spam và cá nhân, dựa trên các đặc trưng văn bản như tần suất từ khóa, độ dài và thông tin metadata (người gửi, tiêu đề).

Dữ liệu sử dụng là tập email công khai. Quá trình tiền xử lý bao gồm tách từ, loại bỏ từ dừng và chuẩn hóa. Dữ liệu được chia thành hai phần: huấn luyện (80%) và kiểm thử (20%), đảm bảo cân bằng giữa các lớp.

Bên cạnh đó, hai thuật toán phân cụm không giám sát là **K-Means** và **DBSCAN** cũng được sử dụng để khám phá cấu trúc tiềm ẩn trong tập dữ liệu mà không cần nhãn. Kết quả phân cụm cho thấy K-Means có xu hướng tạo ra các nhóm rõ ràng hơn, trong khi DBSCAN phát hiện được các điểm nhiễu (noise) hiệu quả hơn, hỗ trợ kiểm chứng độ tương quan giữa nhãn gán và phân bố dữ liệu thực tế.

Kết quả thực nghiệm:

Mô hình	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Logistic Regression	0.9507	0.8840	0.9527	0.9171
K-Nearest Neighbors	0.9507	0.9181	0.9088	0.9134
Bernoulli Naive Bayes	0.8841	0.7821	0.8243	0.8026

Dựa trên kết quả thực nghiệm, **Logistic Regression** cho hiệu suất vượt trội với độ chính xác và F1 Score cao nhất, cho thấy mô hình này phù hợp nhất cho bài toán phân loại email. **K-Nearest Neighbor** có khả năng nhận diện (Recall) tốt nhưng độ chính xác tổng thể và F1 Score thấp hơn, trong khi **Bernoulli Naive Bayes** cho kết quả thấp nhất trên mọi chỉ số. Kết hợp thêm phân cụm giúp hiểu rõ hơn về phân bố dữ liệu, nhưng với mục tiêu phân loại cụ thể, Logistic Regression vẫn là lựa chọn tối ưu.

2 Giới thiệu đề tài

2.1 Mục tiêu và bài toán phân loại email

Phân loại email là một bài toán quan trọng trong lĩnh vực học máy và xử lý ngôn ngữ tự nhiên. Nó đóng vai trò then chốt trong việc xây dựng các hệ thống tự động hỗ trợ người dùng quản lý hộp thư một cách hiệu quả. Với mục tiêu phân loại email thành các nhóm chủ đề (chẳng hạn như công việc, quảng cáo, spam, cá nhân), bài toán này không chỉ liên quan đến lý thuyết phân loại mà còn có nhiều ứng dụng thực tế trong môi trường làm việc hiện đại.

Mục tiêu chính của bài toán phân loại email sử dụng mô hình học máy là xác định nhóm chủ đề của một email dựa trên các đặc trưng văn bản, bao gồm nội dung, tiêu đề, người gửi, và các thông tin metadata khác. Với sự phát triển của các kỹ thuật học máy và xử lý ngôn ngữ tự nhiên, việc xây dựng các hệ thống phân loại email tự động trở nên ngày càng khả thi và hiệu quả. Những hệ thống này có thể giúp người dùng tiết kiệm thời gian, phát hiện email quan trọng hoặc lọc bỏ thư rác một cách tự động.

2.2 Ứng dụng thực tế

Phân loại email có thể được ứng dụng trong nhiều lĩnh vực, đặc biệt là trong các hệ thống quản lý thông tin và truyền thông. Dưới đây là một số ứng dụng tiêu biểu:

- **Lọc thư rác (spam filter)**: Hệ thống phân loại có thể phát hiện và đưa các email rác vào thư mục riêng, giúp người dùng không bị làm phiền và tránh nguy cơ lừa đảo.
- **Tự động gắn nhãn và sắp xếp email**: Các email có thể được tự động phân loại thành các nhóm như công việc, cá nhân, quảng cáo,... giúp tối ưu việc tìm kiếm và phản hồi email.
- **Trợ lý ảo và chatbot**: Các trợ lý thông minh có thể sử dụng hệ thống phân loại email để ưu tiên các thư quan trọng hoặc đưa ra phản hồi tự động phù hợp với nội dung email.

- **Phân tích hành vi người dùng và hỗ trợ chăm sóc khách hàng**: Trong các tổ chức lớn, việc phân tích nội dung email đến có thể giúp hiểu rõ hơn về nhu cầu người dùng và cải thiện dịch vụ chăm sóc khách hàng.

2.3 Các phương pháp được lựa chọn

Trong nghiên cứu này, chúng tôi lựa chọn ba phương pháp học máy phổ biến để giải quyết bài toán phân loại email. Mỗi phương pháp đều có những đặc điểm riêng biệt và phù hợp với từng loại dữ liệu hoặc mục tiêu phân tích.

- **K-Nearest Neighbors (KNN)**: KNN là một thuật toán phân loại dựa trên việc so sánh điểm dữ liệu cần phân loại với các điểm lân cận gần nhất trong tập huấn luyện. Trong bài toán phân loại văn bản, KNN hoạt động hiệu quả khi sử dụng các đặc trưng như vector TF-IDF và khoảng cách cosine.

- **Bernoulli Naive Bayes (GNB)**: Naive Bayes là một phương pháp phân loại xác suất đơn giản nhưng hiệu quả, giả định tính độc lập giữa các đặc trưng. Mặc dù giả định này không hoàn toàn đúng với dữ liệu văn bản, Gaussian Naive Bayes vẫn cho kết quả tốt khi dữ liệu có phân phối tương đối chuẩn và số chiều lớn.

- **Logistic Regression (LR)**: Logistic Regression là một mô hình phân loại tuyến tính mạnh mẽ, đặc biệt hiệu quả trong bài toán phân loại văn bản có đặc trưng thưa (sparse features). Nhờ khả năng xử lý tốt dữ liệu nhiều chiều và dễ diễn giải, Logistic Regression thường được sử dụng như một mô hình baseline cho nhiều bài toán phân loại.

Các phương pháp trên được lựa chọn vì tính phổ biến, dễ triển khai và hiệu quả đã được kiểm chứng trong các bài toán phân loại văn bản. Bằng cách áp dụng cả ba phương pháp, chúng tôi hướng đến việc so sánh và đánh giá hiệu suất phân loại dựa trên các tiêu chí như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (specificity) và diện tích dưới đường cong ROC (AUC).

Ngoài ba phương pháp phân loại nêu trên, chúng tôi cũng áp dụng hai thuật toán phân cụm không giám sát là **K-Means** và **DBSCAN** trên dữ liệu đã được giảm chiều bằng **PCA** nhằm trực quan hóa và phân tích cấu trúc tiềm ẩn của tập dữ liệu. Việc sử

dụng PCA giúp làm nổi bật các chiều quan trọng nhất, từ đó cải thiện khả năng phân cụm và trực quan.

Kết quả cho thấy K-Means tạo ra các cụm tương đối rõ ràng, phần lớn trùng khớp với nhãn phân loại thực tế, trong khi DBSCAN phát hiện được một số điểm nhiễu và cụm nhỏ. Điều này giúp củng cố giả định rằng dữ liệu có thể phân tách tốt bằng các mô hình học máy đã chọn, đồng thời cung cấp góc nhìn bổ sung về phân bố của các loại email trong không gian đặc trưng.

Việc lựa chọn ba phương pháp KNN, Bernoulli Naive Bayes và Logistic Regression không chỉ dựa trên tính phổ biến mà còn vì mỗi phương pháp đại diện cho một hướng tiếp cận khác nhau trong học máy:

- **KNN** đại diện cho nhóm thuật toán phi tham số, không xây dựng mô hình huấn luyện rõ ràng mà dựa trên khoảng cách trong không gian đặc trưng để phân loại. Điều này giúp mô hình đơn giản, dễ hiểu và phù hợp với dữ liệu có độ tương đồng cao.

- **Bernoulli Naive Bayes** là một mô hình xác suất dựa trên định lý Bayes và giả định tính độc lập giữa các đặc trưng. Phương pháp này đặc biệt hiệu quả với dữ liệu có phân phối gần chuẩn và mang lại tốc độ huấn luyện rất nhanh, thích hợp với dữ liệu có nhiều chiều như văn bản.

- **Logistic Regression** là một mô hình tuyến tính có khả năng học trọng số của các đặc trưng để tối ưu hóa phân loại. Phương pháp này thường hoạt động tốt trên dữ liệu thưa, có tính tuyến tính, và là một trong những baseline đáng tin cậy trong phân loại văn bản.

Bằng việc kết hợp ba phương pháp thuộc ba hướng tiếp cận khác nhau – dựa trên khoảng cách (KNN), xác suất (Naive Bayes), và mô hình tuyến tính (Logistic Regression) – nghiên cứu nhằm mục tiêu đưa ra đánh giá khách quan, toàn diện hơn về hiệu quả của từng mô hình đối với bài toán phân loại email. Ngoài ra, cả ba phương pháp đều có ưu điểm là dễ cài đặt, thời gian huấn luyện ngắn, và phù hợp với tập dữ liệu có quy mô vừa, giúp đảm bảo tính khả thi trong phạm vi nghiên cứu.

2.4 Phân chia công việc

Để thực hiện nghiên cứu này, nhóm chúng em đã phân công công việc như sau:

- **Đoàn Phạm Ngọc Linh:** Phụ trách tiền xử lý dữ liệu, bao gồm loại bỏ các cột không cần thiết, chuẩn hóa dữ liệu, và giảm chiều dữ liệu bằng **PCA**. Linh cũng chịu trách nhiệm xây dựng và đánh giá mô hình **K-Nearest Neighbors (KNN)**.
- **Nguyễn Duy Vũ:** Thực hiện mô hình **Logistic Regression (LR)**, bao gồm xây dựng, huấn luyện, tối ưu hóa tham số và đánh giá mô hình dựa trên các chỉ số như accuracy, precision, recall và F1 score. Ngoài ra, Vũ cũng đảm nhiệm việc thực hiện phân cụm bằng thuật toán **DBSCAN** trên dữ liệu đã giảm chiều.
- **Vũ Đức Duy:** Phụ trách mô hình **Bernoulli Naive Bayes** và trực quan hóa dữ liệu, bao gồm vẽ biểu đồ phương sai tích lũy (PCA), biểu đồ phân tán và phân tích phân tách các nhóm email spam và không spam. Bên cạnh đó, Duy cũng thực hiện phân cụm dữ liệu bằng thuật toán **K-Means**.

3 Dữ liệu và thực nghiệm

3.1 Dữ liệu

Bộ dữ liệu này bao gồm thông tin về phân loại email tự động, với mục tiêu phân loại email theo các nhóm chủ đề khác nhau như công việc, quảng cáo, spam, và cá nhân. Bộ dữ liệu được thu thập từ các email công khai và chứa nhiều đặc trưng văn bản, bao gồm tần suất từ khóa, độ dài email, và các thông tin metadata như người gửi và tiêu đề. Bộ dữ liệu này rất hữu ích cho các nhà nghiên cứu và các chuyên gia trong lĩnh vực học máy và xử lý ngôn ngữ tự nhiên, đặc biệt là trong việc phát triển các mô hình phân loại văn bản.

3.1.1 Cấu trúc dữ liệu

- **Số lượng bản ghi:** 5172 email.
- **Số lượng cột:** 3002.
- **Loại dữ liệu:**
  - Dữ liệu số thực: Tần suất xuất hiện của các từ phổ biến trong email.
  - Dữ liệu số nguyên: Các chỉ số thống kê khác như độ dài văn bản, số lượng từ.
  - Dữ liệu dạng chuỗi: Tên email, tiêu đề,...

3.1.2 Một số cột đáng chú ý

- **Email No. / Email Name:** Mã định danh và tên email, dùng để tra cứu nhưng không được sử dụng trong quá trình huấn luyện mô hình.
- **Most Common Word:** Các cột thể hiện tần suất xuất hiện của 9 từ phổ biến nhất trong email, bao gồm các từ như “the”, “to”, “ect”, “and”, “for”, “of”, “a”, “you”, “hou”. Đây là những đặc trưng quan trọng phản ánh nội dung và ngữ cảnh email.

- **Tần suất từ:** Giá trị trung bình (mean) và độ lệch chuẩn (std) của mỗi từ cho thấy sự phân tán và phân bố khác nhau giữa các email. Ví dụ: từ ”you” có giá trị trung bình là 55.5 và độ lệch chuẩn 87.6, cho thấy tần suất sử dụng rất khác nhau giữa các email.
- **Label:** Nhãn mục tiêu được gán cho từng email để phục vụ cho bài toán phân loại.

3.1.3 Thông tin bổ sung

Tỷ lệ giữa các lớp trong bộ dữ liệu này khá cân bằng, với tỷ lệ spam và không spam gần như đều nhau. Điều này giúp mô hình phân loại có thể học được các đặc trưng chung mà không bị thiên lệch quá mức về một lớp nào đó.

3.2 Tiền xử lý dữ liệu

Bộ dữ liệu này đã được tiền xử lý sẵn với các đặc trưng được trích xuất từ nội dung email. Tuy nhiên, để cải thiện hiệu suất của các mô hình phân loại, chúng tôi đã thực hiện các bước tiền xử lý bổ sung như sau:

- **Loại bỏ các cột không cần thiết:**
  - Bộ dữ liệu gốc chứa 3002 cột, trong đó có một số cột không phục vụ cho quá trình học máy. Cụ thể, cột `Email No.` chỉ đóng vai trò là chỉ số định danh cho từng email, không mang thông tin đặc trưng về nội dung, nên được loại bỏ khỏi tập dữ liệu đầu vào. Ngoài ra, cột `Prediction` là nhãn mục tiêu cần dự đoán, được lưu tách riêng để phục vụ cho quá trình huấn luyện và đánh giá mô hình.
- **Chuẩn hóa dữ liệu:** Dữ liệu đầu vào có đặc trưng là tần suất xuất hiện từ, nên các giá trị có thể dao động ở nhiều thang đo khác nhau. Để tránh hiện tượng một số đặc trưng có giá trị lớn lấn át các đặc trưng khác và giúp các thuật toán học máy hoạt động hiệu quả hơn, chúng tôi sử dụng phương pháp `StandardScaler`

để chuẩn hóa toàn bộ dữ liệu. Phương pháp này đưa tất cả các đặc trưng về cùng phân phối chuẩn với kỳ vọng 0 và độ lệch chuẩn 1:

$$X' = \frac{X - \mu}{\sigma}$$

Trong đó, μ là giá trị trung bình và σ là độ lệch chuẩn của từng đặc trưng.

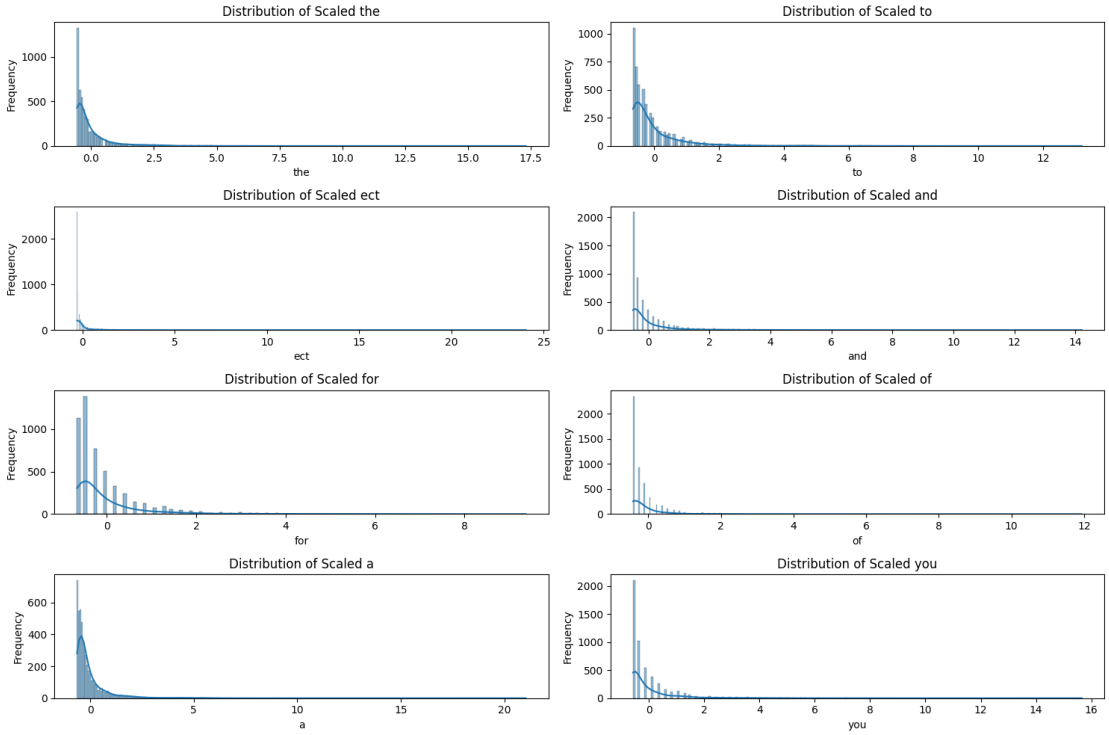


Figure 1: Biểu đồ phân phối của một số trường dữ liệu

- **Giảm chiều dữ liệu:**
  - Áp dụng `PCA (Principal Component Analysis)` để giảm số chiều dữ liệu từ 3000 xuống còn 50 thành phần chính
  - Các thành phần chính này giữ lại hầu hết thông tin quan trọng trong dữ liệu gốc, đồng thời giảm đáng kể độ phức tạp của mô hình.
- **Chia dữ liệu:**

- Chia dữ liệu thành tập huấn luyện (80%) và tập kiểm tra (20%) để đánh giá hiệu suất của mô hình.
- Sử dụng phương pháp phân tầng để đảm bảo tỷ lệ email spam và không phải spam được giữ nguyên trong cả hai tập.

Quá trình tiền xử lý này giúp cải thiện hiệu suất của các mô hình phân loại bằng cách loại bỏ thông tin không cần thiết, chuẩn hóa các đặc trưng và giảm độ phức tạp của dữ liệu.

4 Trực quan hóa dữ liệu

4.1 PCA

Trong phần này, chúng ta tiến hành phân tích hiệu quả của phương pháp PCA trong việc giảm chiều dữ liệu email để phân loại spam. Đầu tiên, chúng ta phân tích lượng thông tin được giữ lại ở từng thành phần chính thông qua phương sai giải thích.

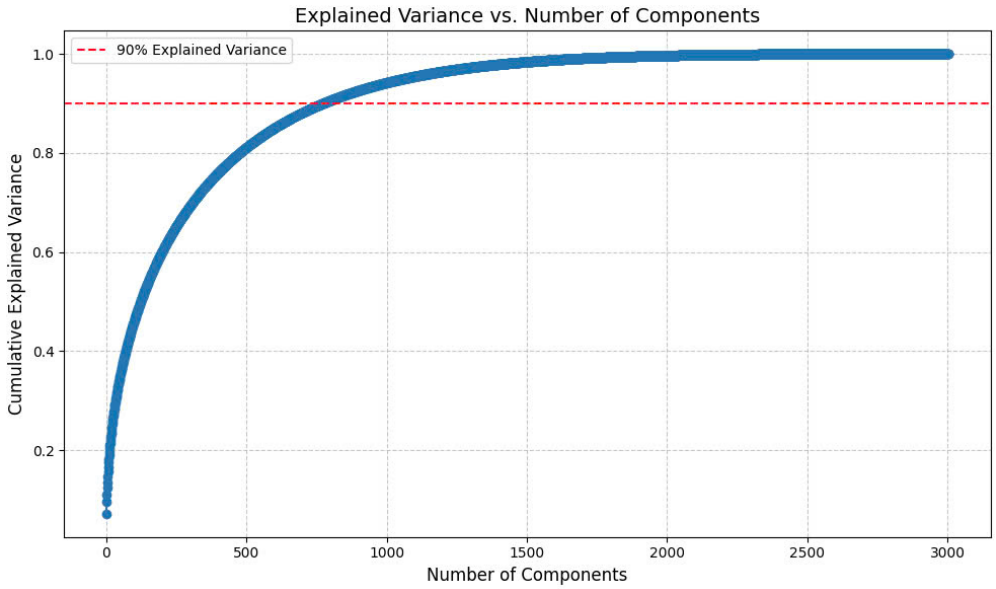


Figure 2: Biểu đồ phương sai tích lũy theo số lượng thành phần chính

Hình trên thể hiện đường cong phương sai tích lũy của dữ liệu email.

- **Phân bố dữ liệu:** Biểu đồ cho thấy cách các email được phân bố trong không gian hai chiều của hai thành phần chính đầu tiên. Điểm màu xanh đại diện cho email không phải spam, điểm màu đỏ đại diện cho email spam.
- **Mức độ phân tách:** Mức độ phân tách giữa các điểm màu xanh và đỏ cho thấy khả năng phân biệt giữa email spam và không spam của hai thành phần chính đầu tiên. Nếu các điểm phân tách rõ ràng, điều này chứng tỏ PCA đã giữ lại được thông tin phân biệt quan trọng.

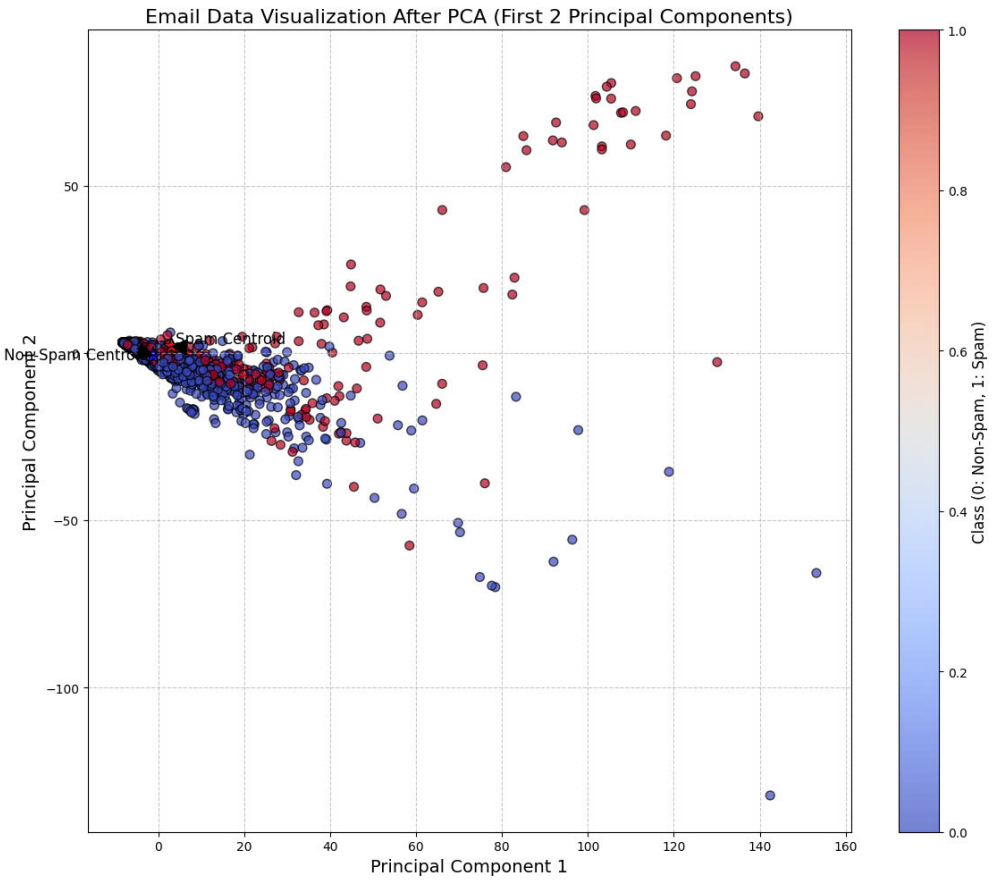


Figure 3: Biểu đồ phân tán của dữ liệu email trên hai thành phần chính đầu tiên

- **Tâm của các nhóm (Centroids):** Các dấu sao đánh dấu tâm của nhóm spam và không spam. Khoảng cách giữa hai tâm này cho thấy mức độ khác biệt trung bình giữa hai loại email.
- **Đường nối giữa các tâm:** Đường kẻ nối giữa hai tâm có thể được xem như một đường phân chia đơn giản giữa hai lớp. Nếu các điểm phân bố rõ ràng về hai phía của đường này, điều đó cho thấy khả năng phân loại tốt ngay cả với mô hình đơn giản.

Hình 3 trình bày phân bố của dữ liệu email trong không gian hai thành phần chính đầu tiên. Các điểm màu xanh đại diện cho email không phải spam, các điểm màu đỏ đại diện cho email spam. Hai điểm đánh dấu tâm của nhóm spam và không spam, với đường thẳng nối giữa hai tâm này.

- **Tốc độ tăng của đường cong:** Đường cong càng dốc ở đầu, càng chứng tỏ một số ít thành phần chính đầu tiên đã nắm giữ phần lớn thông tin trong dữ liệu..
- **Điểm cắt với ngưỡng 90% :** Điểm mà đường cong cắt đường ngang màu đỏ cho biết cần bao nhiêu thành phần chính để giữ lại 90% thông tin trong dữ liệu gốc.
- **Hiệu quả của việc giảm chiều :** Nếu chỉ cần một số nhỏ thành phần chính để đạt được 90% phương sai, điều này chứng tỏ việc giảm chiều dữ liệu rất hiệu quả và không làm mất nhiều thông tin quan trọng.

5 Phân cụm dữ liệu

5.1 KMeans

5.1.1 Giới thiệu

K-means là một thuật toán phân cụm thuộc nhóm học không giám sát (unsupervised learning), được dùng phổ biến để nhóm các điểm dữ liệu thành K cụm sao cho những điểm trong cùng một cụm có tính chất tương đồng. Tiêu chí phân cụm thường dựa trên khoảng cách Euclidean giữa điểm và trung tâm cụm. K-means có nhiều ứng dụng thực tiễn trong phân tích thị trường, nhận dạng ảnh, khai phá văn bản, và nhiều bài toán khai phá dữ liệu khác.

5.1.2 Phân tích toán học

Ký hiệu toán học

Giả sử có N điểm dữ liệu được mô tả bằng ma trận $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in R^{d \times N}$, trong đó $\mathbf{x}_i \in R^{d \times 1}$ là vector đặc trưng cho điểm dữ liệu thứ i trong không gian d-chiều. Với số cụm được chọn là $K < N$, mục tiêu là xác định các trung tâm cụm $\mathbf{m}_1, \ldots, \mathbf{m}_K$ và gán nhãn tương ứng cho từng điểm.

Mỗi nhãn $\mathbf{y}_i$ là một vector one-hot kích thước K, thỏa mãn: - $y_{ik} = 1$ nếu $\mathbf{x}_i$ thuộc cụm k, - $y_{ij} = 0$ với mọi $j \neq k$.

Tức là mỗi điểm dữ liệu chỉ thuộc duy nhất một cụm, được ràng buộc bằng công thức:

$$y_{ik} \in \{0, 1\}, \quad \sum_{k=1}^{K} y_{ik} = 1 \quad \forall i$$

Ta xây dựng ma trận nhãn $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_N]$ và ma trận trung tâm cụm $\mathbf{M} = [\mathbf{m}_1, \ldots, \mathbf{m}_K]$.

Hàm mất mát

Với mỗi điểm $\mathbf{x}_i$, sai số gán vào cụm k được đo bằng khoảng cách bình phương $\|\mathbf{x}_i - \mathbf{m}_k\|_2^2$. Với nhãn one-hot, ta có thể viết sai số:

$$y_{ik}\|\mathbf{x}_i - \mathbf{m}_k\|_2^2 = \sum_{j=1}^{K} y_{ij}\|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Tổng mất mát trên toàn bộ dữ liệu được xác định bởi hàm:

$$L(\mathbf{Y}, \mathbf{M}) = \sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Trong đó, điều kiện $y_{ij} \in \{0, 1\}$ và $\sum_{j=1}^{K} y_{ij} = 1$ vẫn được đảm bảo.

Bài toán tối ưu

Nhiệm vụ của K-means là xác định $\mathbf{Y}$ và $\mathbf{M}$ sao cho hàm mất mát được tối thiểu:

$$\mathbf{Y}, \mathbf{M} = \arg\min_{\mathbf{Y},\mathbf{M}} \sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

với điều kiện:

$$y_{ij} \in \{0, 1\}\ \forall i, j; \quad \sum_{j=1}^{K} y_{ij} = 1\ \forall i$$

Tối ưu hóa luân phiên

Giữ $\mathbf{M}$ *cố định, tìm* $\mathbf{Y}$

Với trung tâm cụm đã biết, ta gán mỗi điểm $\mathbf{x}_i$ vào cụm có trung tâm gần nhất:

$$\mathbf{y}_i = \arg\min_{\mathbf{y}_i} \sum_{j=1}^{K} y_{ij}\|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

với điều kiện:

$$y_{ij} \in \{0, 1\}\ \forall j; \quad \sum_{j=1}^{K} y_{ij} = 1$$

Tương đương với:

$$j = \arg\min_{j} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Giữ **Y** *cố định, cập nhật* **M**

Sau khi xác định nhãn, ta cập nhật trung tâm cụm theo công thức:

$$\mathbf{m}_j = \arg\min_{\mathbf{m}_j} \sum_{i=1}^{N} y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2$$

Lấy đạo hàm và giải phương trình:

$$\frac{\partial}{\partial \mathbf{m}_j} \sum_{i=1}^{N} y_{ij} \|\mathbf{x}_i - \mathbf{m}_j\|_2^2 = 2 \sum_{i=1}^{N} y_{ij} (\mathbf{m}_j - \mathbf{x}_i) = 0$$

Suy ra:

$$\mathbf{m}_j = \frac{\sum_{i=1}^{N} y_{ij} \mathbf{x}_i}{\sum_{i=1}^{N} y_{ij}}$$

Tức là trung tâm $\mathbf{m}_j$ là trung bình cộng của các điểm trong cụm j, làm rõ ý nghĩa tên gọi "K trung bình".

5.1.3 Các bước trong thuật toán K-means

Thuật toán K-means thực hiện quá trình lặp gồm hai bước chính: gán điểm vào cụm và cập nhật trung tâm, cho đến khi các trung tâm hội tụ hoặc không thay đổi đáng kể.

Quy trình tổng quát:

1. Khởi tạo:
    - Xác định số lượng cụm K.
    - Chọn ngẫu nhiên K điểm làm trung tâm ban đầu $\mathbf{m}_1, \ldots, \mathbf{m}_K$. Có thể dùng K-means++ để chọn thông minh hơn:
        - Chọn một điểm bất kỳ làm trung tâm đầu tiên.
        - Tính khoảng cách từ mỗi điểm đến trung tâm gần nhất.
        - Chọn trung tâm tiếp theo với xác suất tỷ lệ với bình phương khoảng cách.
        - Lặp lại cho đến khi đủ K trung tâm.

2. Gán cụm cho từng điểm:

    - Với mỗi điểm $\mathbf{x}_i$:
        - Tính khoảng cách đến tất cả các trung tâm $\mathbf{m}_1, \ldots, \mathbf{m}_K$.
        - Gán điểm vào cụm có trung tâm gần nhất:

$$y_{ij} = \begin{cases} 1 & \text{nếu } j = \arg\min_k \|\mathbf{x}_i - \mathbf{m}_k\|_2^2 \\ 0 & \text{ngược lại} \end{cases}$$

5.2 DBSCAN

5.2.1 Giới thiệu

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là một thuật toán phân cụm dựa trên mật độ, không yêu cầu chỉ định trước số lượng cụm. Khác với K-means vốn giả định các cụm có hình dạng cầu và kích thước tương tự, DBSCAN có thể phát hiện các cụm có hình dạng bất kỳ và xử lý tốt dữ liệu nhiễu. Thuật toán này rất phù hợp cho các bài toán khai phá tri thức trong không gian có mật độ điểm không đồng đều, như phát hiện bất thường, nhận dạng hình dạng phức tạp trong xử lý ảnh hoặc dữ liệu không gian.

5.2.2 Khái niệm và định nghĩa chính

DBSCAN sử dụng hai tham số chính:

- ε: bán kính lân cận để xác định vùng lân cận của một điểm.
- `MinPts`: số lượng điểm tối thiểu để xác định một vùng có mật độ đủ cao.

Các khái niệm quan trọng:

- **Điểm lõi (core point)**: điểm có ít nhất `MinPts` điểm (kể cả bản thân nó) nằm trong vùng lân cận bán kính ε.
- **Điểm biên (border point)**: điểm không phải lõi, nhưng nằm trong vùng lân cận của một điểm lõi.

- **Điểm nhiễu (noise point)**: điểm không thuộc bất kỳ cụm nào.

Quan hệ mật độ:

- **Directly density-reachable**: điểm p trực tiếp đạt được mật độ từ điểm q nếu $p \in N_\varepsilon(q)$ và q là điểm lõi.
- **Density-reachable**: điểm p đạt được mật độ từ q nếu tồn tại chuỗi các điểm $p_1, p_2, \ldots, p_n$, với $p_1 = q$, $p_n = p$, và mỗi p_{i+1} trực tiếp đạt được mật độ từ p_i.
- **Density-connected**: hai điểm p và q được gọi là kết nối theo mật độ nếu tồn tại điểm o sao cho cả p và q đều density-reachable từ o.

5.2.3 Thuật toán DBSCAN

Quy trình của DBSCAN có thể được mô tả như sau:

1. Duyệt qua từng điểm dữ liệu chưa được gán cụm:
    - Nếu điểm đó không phải là điểm lõi (số lượng lân cận nhỏ hơn `MinPts`), đánh dấu là nhiễu.
    - Nếu là điểm lõi, khởi tạo một cụm mới và mở rộng cụm bằng cách kết nối tất cả các điểm đạt được mật độ từ điểm này.
2. Quá trình lặp lại cho đến khi tất cả các điểm đã được xét.

5.2.4 Mở rộng cụm

Việc mở rộng cụm từ một điểm lõi được thực hiện như sau:

- Tìm tất cả các điểm nằm trong vùng ε của điểm lõi hiện tại.
- Nếu số lượng điểm thỏa mãn $\geq$ `MinPts`, thêm tất cả điểm đó vào cụm.
- Với mỗi điểm mới thêm vào cụm, nếu nó là điểm lõi, tiếp tục tìm các điểm trong vùng lân cận của nó và thêm vào cụm theo cách đệ quy.

5.2.5 Ưu và nhược điểm

Ưu điểm:

- Không cần chỉ định số cụm trước.
- Phát hiện được cụm có hình dạng bất kỳ.
- Có khả năng loại bỏ nhiễu hiệu quả.

Nhược điểm:

- Nhạy cảm với tham số ε và `MinPts`.
- Khó hoạt động hiệu quả trên dữ liệu có mật độ cụm không đồng đều.

5.2.6 Kết luận

DBSCAN là một thuật toán mạnh mẽ khi làm việc với dữ liệu có hình dạng cụm không đều và chứa nhiễu. Dù có hạn chế về việc chọn tham số, nhưng nếu lựa chọn phù hợp, nó có thể phân cụm tốt hơn so với các thuật toán như K-means trong nhiều tình huống thực tế.

6 Tổng quan về các phương pháp

6.1 K-Nearest Neighbors (KNN)

6.1.1 Giới thiệu

K-Nearest Neighbors (KNN) là một phương pháp phân loại đơn giản nhưng mạnh mẽ. Phương pháp này không yêu cầu quá trình huấn luyện phức tạp mà thay vào đó, sử dụng toàn bộ dữ liệu huấn luyện để đưa ra quyết định phân loại dựa trên những điểm dữ liệu gần nhất.

6.1.2 Nguyên lý hoạt động

KNN hoạt động bằng cách tìm kiếm k điểm dữ liệu gần nhất với điểm cần phân loại trong không gian đặc trưng. Dự đoán của mô hình sẽ dựa trên đa số nhãn của các điểm gần nhất.

Công thức tính khoảng cách phổ biến nhất là khoảng cách Euclidean:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Trong đó x và y là hai điểm trong không gian đặc trưng với các giá trị thuộc n-chiều.

6.1.3 Ưu điểm và nhược điểm

- **Ưu điểm:**
  - Đơn giản, dễ hiểu và triển khai.
  - Không yêu cầu quá trình huấn luyện.
  - Hoạt động tốt với các dữ liệu không có quá nhiều nhiễu.
- **Nhược điểm:**
  - Chi phí tính toán cao khi dữ liệu lớn.
  - Kém hiệu quả với các dữ liệu có không gian đặc trưng cao.

- Kết quả dự đoán phụ thuộc vào giá trị của k, nếu chọn k không phù hợp sẽ dẫn đến kết quả không chính xác.

6.2 Bernoulli Naive Bayes (BNB)

6.2.1 Giới thiệu

Bernoulli Naive Bayes (BNB) là phương pháp phân loại dựa trên lý thuyết Bayes, thiết kế cho dữ liệu có **đặc trưng nhị phân** (0 hoặc 1). Phương pháp này phù hợp với bài toán phân loại văn bản khi chỉ quan tâm đến sự **xuất hiện** của từ.

6.2.2 Nguyên lý hoạt động

- **Công thức Bayes**:

$$p(k \mid \mathbf{x}) \propto p(\mathbf{x} \mid k) \cdot p(k)$$

Với giả thiết độc lập:

$$p(\mathbf{x} \mid k) = \prod_{i=1}^{d} p(x_i \mid k)$$

- **Tính xác suất điều kiện**:

$$p(x_i \mid k) = (p_{ki})^{x_i} \cdot (1 - p_{ki})^{1-x_i}$$

Trong đó:

$$p_{ki} = \frac{\text{Số mẫu lớp } k \text{ có } x_i = 1 + \alpha}{\text{Tổng số mẫu lớp } k + 2\alpha}$$

($\alpha = 1$ cho làm mịn Laplace)

6.2.3 Ưu điểm và hạn chế

- **Ưu điểm**:
  - Hiệu quả với dữ liệu nhị phân
  - Tốc độ huấn luyện nhanh

- **Hạn chế**:
  - Giả thiết độc lập không thực tế
  - Không xử lý tốt tần suất xuất hiện

6.3 Logistic Regression (LR)

6.3.1 Giới thiệu

Logistic Regression (LR) là một phương pháp phân loại phổ biến, mặc dù tên gọi là ”hồi quy”, nhưng nó thực chất là một thuật toán phân loại. LR được sử dụng để dự đoán xác suất của một lớp dựa trên các đặc trưng đầu vào

6.3.2 Nguyên lý hoạt động

Logistic Regression sử dụng hàm sigmoid để chuyển đổi kết quả của mô hình hồi quy tuyến tính thành xác suất:

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)}$$

Xác suất phân lớp:

$$P(y = 1|x; \theta) = h_\theta(x), \quad P(y = 0|x; \theta) = 1 - h_\theta(x)$$

Tham số θ được ước lượng bằng phương pháp cực đại hóa log-likelihood:

$$\ell(\theta) = \sum_{n=1}^{N} [y_n \log h_\theta(x_n) + (1 - y_n) \log(1 - h_\theta(x_n))]$$

Có thể sử dụng các thuật toán tối ưu như Gradient Descent hoặc Newton-Raphson để tìm

6.3.3 Ưu điểm và nhược điểm

- **Ưu điểm:**
  - Dễ hiểu, dễ triển khai.

- Hiệu quả với dữ liệu có quan hệ tuyến tính giữa đặc trưng và lớp.
- Có thể mở rộng cho bài toán đa lớp.

- **Nhược điểm:**
  - Không phù hợp với dữ liệu có quan hệ phi tuyến tính.
  - Nhạy cảm với đa cộng tuyến và ngoại lệ.

6.4 Thực nghiệm

6.4.1 Phân Cụm

- K-Means Clustering
  Thực hiện phân cụm dữ liệu với số cụm là 2, đánh giá kết quả bằng 3 chỉ số:
  - `Silhouette Score`: Đo độ tách biệt và kết dính của các cụm. Giá trị càng gần 1 thì cụm càng tốt.
  - `Davies-Bouldin Index (DBI)`: Đo mức độ tương đồng giữa các cụm. Giá trị càng nhỏ thì phân cụm càng tốt.
  - `Adjusted Rand Index (ARI)`: So sánh giữa phân cụm của mô hình với nhãn thực tế. Giá trị gần 1 thể hiện kết quả tốt.

  Kết quả các chỉ số đánh giá như sau:

Chỉ số	**Giá trị**
Silhouette Score	0.6924
Davies-Bouldin Index	2.6078
Adjusted Rand Index	0.0400

Table 1: Kết quả đánh giá phân cụm K-Means

  Dựa vào các chỉ số đánh giá thu được:
  - **Silhouette Score = 0.6924**: Đây là một giá trị tương đối cao (gần 1), cho thấy các điểm dữ liệu trong từng cụm khá gần nhau và cách xa với các cụm

khác. Điều này chứng tỏ K-Means đã tạo ra các cụm tương đối rõ ràng và có ranh giới tốt.

- **Davies-Bouldin Index = 2.6078**: Giá trị này tương đối cao (vì DBI càng nhỏ càng tốt), cho thấy giữa các cụm vẫn còn mức độ chồng lấn hoặc cụm chưa đủ phân biệt rõ ràng. Tuy chưa lý tưởng, nhưng vẫn chấp nhận được trong một số ngữ cảnh dữ liệu thực tế.
- **Adjusted Rand Index = 0.0400**: ARI có giá trị gần 0, cho thấy kết quả phân cụm không tương đồng nhiều với nhãn thực tế (nếu có). Điều này có thể do:
  * Dữ liệu thực tế không có cấu trúc cụm rõ ràng.
  * Số lượng cụm chưa phù hợp.
  * K-Means không phải là thuật toán tối ưu cho tập dữ liệu này (do giả định cụm hình cầu).

Biểu đồ tỉ lệ của các nhãn (label) trong các cụm (cluster):

Cụm	Tỷ lệ nhãn 0	Tỷ lệ nhãn 1
0	72.03%	27.97%
1	31.30%	68.70%

Table 2: Tỷ lệ nhãn gốc trong các cụm K-Means

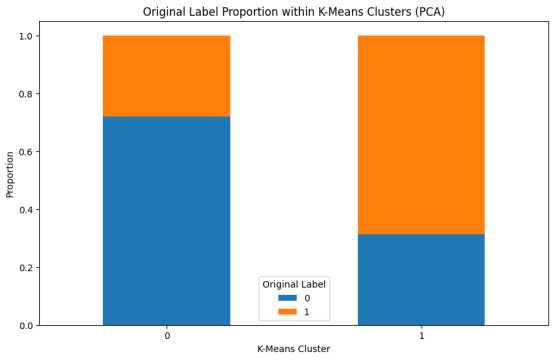


Figure 4: Biểu đồ tỉ lệ phân phối nhãn giữa các cụm với dữ liệu gốc (K-Means Clustering)

Nhận xét:

- **Cụm 0** có phần lớn các điểm dữ liệu thuộc nhãn 0 (72.03%), tuy nhiên vẫn chứa một lượng lớn dữ liệu nhãn 1 (27.97%), cho thấy cụm này không hoàn toàn thuần nhất.
- **Cụm 1** có tỷ lệ nhãn 1 chiếm ưu thế (68.70%), thể hiện khả năng phân biệt tốt hơn giữa hai nhãn trong cụm này.
- Tuy nhiên, số lượng điểm dữ liệu trong cụm 1 khá nhỏ (chỉ 131 điểm), điều này có thể là nguyên nhân dẫn đến giá trị Adjusted Rand Index thấp (0.0400), vì phân cụm chưa phù hợp với nhãn gốc.

Kết luận: Mặc dù có một số khả năng tách cụm nhất định (đặc biệt với nhãn 1 trong cụm 1), nhưng kết quả cho thấy rằng thuật toán K-Means chưa phù hợp để tái hiện cấu trúc phân lớp của dữ liệu gốc trong không gian PCA.

- DBSCAN Clustering
  Kết quả phân cụm được đánh giá bằng ba chỉ số chính:
  - **Silhouette Score**: 0.5995
  - **Davies-Bouldin Index**: 0.3347
  - **Adjusted Rand Index (ARI)**: 0.0311

Phân tích:

- **Silhouette Score = 0.5995**: Đây là một giá trị tương đối tốt (giá trị lý tưởng gần 1). Nó cho thấy rằng các điểm dữ liệu đang được gán vào các cụm khá rõ ràng, tức là các điểm nằm gần cụm của chính nó và xa các cụm khác. Điều này cho thấy DBSCAN đã xác định được các cụm với hình dạng phù hợp cấu trúc dữ liệu.
- **Davies-Bouldin Index = 0.3347**: Chỉ số này càng thấp càng tốt. Giá trị 0.33 là rất thấp, chứng tỏ rằng các cụm cách biệt tốt với nhau và có sự tập trung

cao bên trong cụm. Điều này củng cố nhận định rằng DBSCAN đã tìm được cấu trúc cụm chất lượng.

– **Adjusted Rand Index = 0.0311**: ARI gần bằng 0 cho thấy sự tương quan giữa phân cụm DBSCAN và nhãn gốc là rất thấp. Điều này cho thấy DBSCAN không tái hiện được phân bố nhãn thực tế. Tuy nhiên, cần lưu ý rằng:
  * DBSCAN là thuật toán không giám sát, không nhất thiết phải phù hợp với nhãn gốc.
  * DBSCAN có thể phát hiện các cấu trúc không tuyến tính mà các nhãn ban đầu không biểu hiện rõ.

Biểu đồ tỉ lệ của các nhãn (label) trong các cụm (cluster):

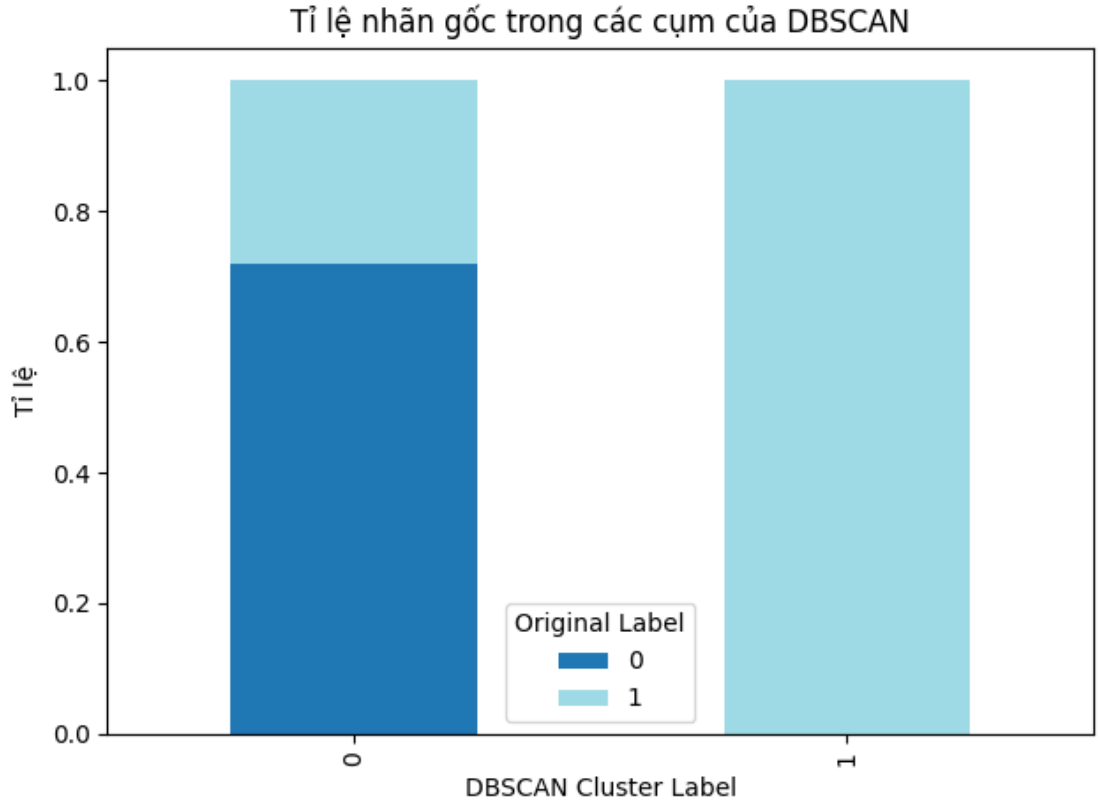


Figure 5: Biểu đồ tỉ lệ phân phối nhãn giữa các cụm với dữ liệu gốc (DBSCAN Clustering)

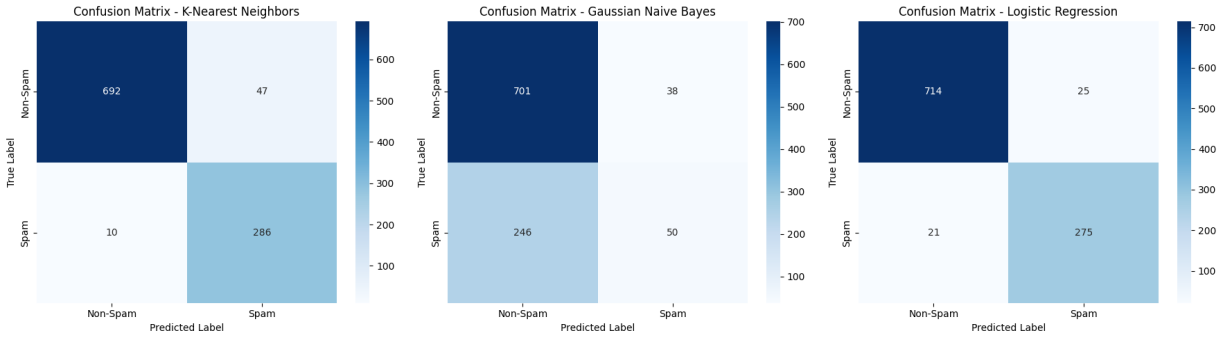
6.4.2 Kết quả các chỉ số của ba mô hình:Bernoulli Naive Bayes, Logistic Regression, KNN

Bảng kết quả trên dữ liệu gốc

Kết quả các mô hình trên dữ liệu gốc

Mô hình	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.9720	0.9435	0.9595	0.9514
K-Nearest Neighbors	0.8628	0.7265	0.8345	0.7767
Bernoulli Naive Bayes	0.8966	0.8412	0.7872	0.8133

Ma trận nhầm lẫn của từng mô hình:



Trên dữ liệu gốc, ba mô hình được so sánh gồm Logistic Regression, K-Nearest Neighbors (KNN) và Bernoulli Naive Bayes. Kết quả như sau:

- **Logistic Regression** đạt độ chính xác (accuracy) 0.9720, F1 Score 0.9514, cho thấy mô hình này hoạt động rất hiệu quả trên dữ liệu gốc. Ma trận nhầm lẫn cho thấy số lượng dự đoán sai (False Positives và False Negatives) thấp, đồng nghĩa với việc mô hình phân biệt tốt giữa email spam và không spam. Điều này phù hợp với đặc điểm của Logistic Regression, vốn xử lý tốt các đặc trưng thưa và nhiều chiều, cũng như tận dụng tốt mối quan hệ tuyến tính giữa các đặc trưng và nhãn phân loại.

- **K-Nearest Neighbors** có accuracy 0.8628 và F1 Score 0.7767, cũng thể hiện hiệu suất tốt. KNN tận dụng được cấu trúc dữ liệu gốc, đặc biệt khi các đặc trưng chưa bị biến đổi nhiều, giúp mô hình xác định được các điểm lân cận chính xác hơn. Tuy nhiên, KNN có thể bị ảnh hưởng bởi các đặc trưng không đồng nhất về thang đo hoặc nhiễu trong dữ liệu.

- **Bernoulli Naive Bayes** Với dữ liệu gốc chưa giảm chiều, cả ba mô hình Logistic Regression, K-Nearest Neighbors (KNN) và Bernoulli Naive Bayes đều cho hiệu suất tốt, trong đó Logistic Regression và KNN vượt trội hơn về độ chính xác và F1-score. Bernoulli Naive Bayes tuy đơn giản, tốc độ nhanh, giả định độc lập giữa các đặc trưng, nhưng do bản chất dữ liệu văn bản thường không thực sự độc lập nên mô hình này cho hiệu suất thấp hơn so với hai mô hình còn lại, đặc biệt là về recall và F1-score. Tuy nhiên, với accuracy đạt khoảng 0.89 và F1-score 0.81, Bernoulli Naive Bayes vẫn có thể là lựa chọn phù hợp khi yêu cầu tốc độ xử lý nhanh và dữ liệu có kích thước lớn.

Nhìn chung, trên dữ liệu gốc, Logistic Regression và KNN đều cho hiệu quả tốt, trong khi Gaussian Naive Bayes bị hạn chế do giả định đơn giản hóa về mối quan hệ giữa các đặc trưng.

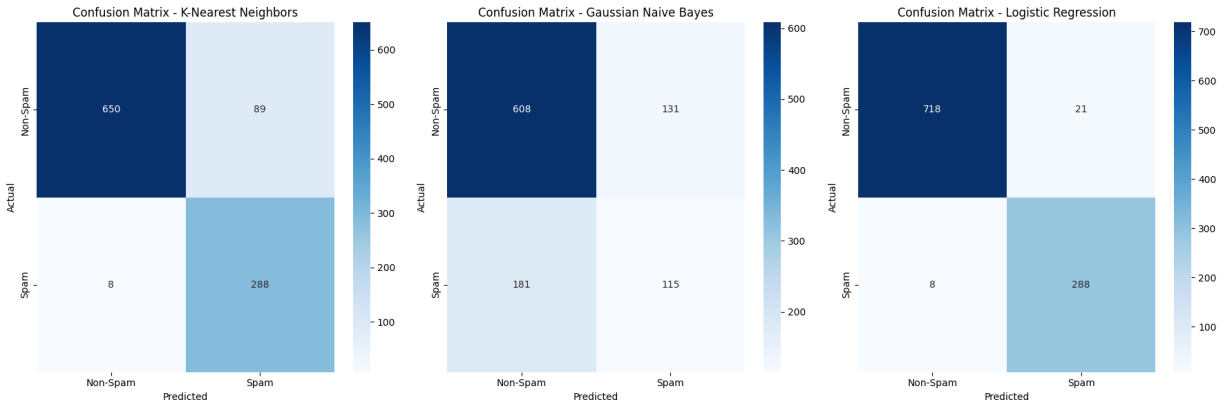
6.4.3 Dữ liệu đã chuẩn hóa và giảm chiều:

Bảng kết quả trên dữ liệu đã chuẩn hóa và giảm chiều

Kết quả các mô hình trên dữ liệu đã chuẩn hóa và giảm chiều

Mô hình	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.9507	0.8840	0.9527	0.9171
K-Nearest Neighbors	0.9507	0.9181	0.9088	0.9134
Bernoulli Naive Bayes	0.8841	0.7821	0.8243	0.8026

Ma trận nhầm lẫn của từng mô hình:



Phân tích kết quả trên dữ liệu đã chuẩn hóa và giảm chiều

Sau khi áp dụng các bước chuẩn hóa (đưa dữ liệu về cùng thang đo) và giảm chiều (PCA), kết quả các mô hình thay đổi rõ rệt:

- **Logistic Regression** đạt kết quả tốt nhất trong ba mô hình với **accuracy 0.9507**, **recall 0.8840**, **precision 0.9527** và **F1-score 0.9171**. Điều này cho thấy Logistic Regression có khả năng học tốt từ các đặc trưng sau tiền xử lý và tổng quát hóa hiệu quả trên dữ liệu kiểm thử. Mô hình hoạt động ổn định và đáng tin cậy trong bài toán phân loại văn bản.

- **K-Nearest Neighbors** cũng đạt độ chính xác cao với **accuracy 0.9507**, nhưng có sự chênh lệch giữa recall (0.9181) và precision (0.9088), dẫn đến F1-score là 0.9134. Kết quả cho thấy KNN hoạt động khá tốt, tuy nhiên độ nhạy cao hơn độ đặc hiệu có thể làm tăng tỷ lệ dự đoán nhầm email không spam thành spam.

- **Bernoulli Naive Bayes** có hiệu suất thấp hơn hai mô hình còn lại với **accuracy 0.8841**, **recall 0.7821**, **precision 0.8243** và **F1-score 0.8026**. Tuy nhiên, đây vẫn là một kết quả khả quan so với bản chất đơn giản của mô hình này. Việc chuẩn hóa và giảm chiều bằng PCA đã giúp mô hình loại bỏ bớt nhiễu và cải thiện hiệu suất, cho thấy rằng với tiền xử lý phù hợp, Naive Bayes vẫn có thể hoạt động hiệu quả ở mức chấp nhận được trong phân loại văn bản.

Kết luận

- Logistic Regression là mô hình hưởng lợi rõ rệt nhất từ việc chuẩn hóa và giảm chiều, cho kết quả tốt nhất ở cả hai giai đoạn.
- KNN phù hợp hơn với dữ liệu gốc, nhưng hiệu suất giảm khi đặc trưng bị biến đổi qua PCA.
- Bernoulli Naive Bayes có thay đổi nhẹ sau chuẩn hóa và giảm chiều, nhưng vẫn kém hiệu quả so với hai mô hình còn lại. Sau khi áp dụng PCA và chuẩn hóa, Bernoulli Naive Bayes đã đạt hiệu suất cao hơn rõ rệt so với kết quả ban đầu, cho thấy tầm quan trọng của việc lựa chọn phương pháp tiền xử lý phù hợp với từng mô hình

Việc lựa chọn phương pháp tiền xử lý phù hợp với từng mô hình là rất quan trọng để đạt hiệu suất tối ưu trong bài toán phân loại email.

7 Kết luận và Phương hướng phát triển

7.1 Kết luận

Trong báo cáo này, chúng tôi đã trình bày quy trình xây dựng hệ thống phân loại email tự động dựa trên các mô hình học máy như **K-Nearest Neighbors (KNN)**, **Bernoulli Naive Bayes** và **Logistic Regression (LR)**. Qua quá trình phân tích, tiền xử lý, huấn luyện và đánh giá mô hình, chúng tôi nhận thấy rằng:

- **Logistic Regression** mang lại hiệu quả cao nhất với độ chính xác 97.2% và F1 Score vượt trội, phù hợp với dữ liệu có đặc trưng thưa.
- **KNN** hoạt động ổn định với khả năng nhận diện (Recall) tốt nhưng hiệu quả tổng thể thấp hơn LR do nhạy cảm với lựa chọn tham số k và chi phí tính toán cao.
- **Bernoulli Naive Bayes** là mô hình đơn giản, tốc độ nhanh nhưng giả định độc lập giữa các đặc trưng không phù hợp với dữ liệu văn bản, dẫn đến độ chính xác thấp hơn.

Ngoài ra, bộ dữ liệu sử dụng trong bài toán có cấu trúc rõ ràng, tỷ lệ các lớp gần cân bằng và không có dữ liệu thiếu. Điều này hỗ trợ tốt cho quá trình huấn luyện và đánh giá mô hình một cách khách quan.

Bên cạnh các mô hình phân loại, chúng tôi cũng đã ứng dụng các thuật toán phân cụm như **K-means** và **DBSCAN** để khám phá cấu trúc tự nhiên của dữ liệu email, giúp phát hiện các nhóm email có nội dung tương tự mà không cần dựa vào nhãn phân loại. Phương pháp này hỗ trợ tốt cho các bước tiền xử lý, giảm chiều dữ liệu và mở rộng khả năng ứng dụng trong các bài toán phân tích dữ liệu lớn.

7.2 Phương hướng phát triển

Trong tương lai, có thể mở rộng và cải tiến nghiên cứu theo các hướng sau:

- **Sử dụng đặc trưng nâng cao:** Áp dụng các kỹ thuật biểu diễn văn bản như TF-IDF, Word2Vec, FastText hoặc BERT thay vì chỉ dùng tần suất từ để mô hình hiểu rõ hơn ngữ cảnh.

- **Thử nghiệm mô hình khác:** Triển khai thêm các mô hình hiện đại như SVM, Random Forest hoặc các mạng nơ-ron như LSTM, BERT để so sánh hiệu suất và khả năng tổng quát hóa.
- **Tối ưu siêu tham số:** Sử dụng các kỹ thuật như Grid Search, Random Search hoặc Bayesian Optimization để tìm tập siêu tham số tốt nhất cho từng mô hình.
- **Mở rộng tập dữ liệu:** Thu thập thêm email thuộc nhiều chủ đề và phong cách khác nhau để tăng tính đa dạng, từ đó giúp mô hình học được nhiều đặc trưng hơn.
- **Tích hợp phân cụm vào quy trình tiền xử lý:** Kết hợp các kết quả phân cụm với các mô hình phân loại để nâng cao chất lượng đặc trưng, phát hiện các nhóm email đặc biệt hoặc tự động gán nhãn cho các nhóm dữ liệu mới.
- **Triển khai ứng dụng thực tế:** Tích hợp mô hình vào hệ thống hỗ trợ quản lý email, thư mục thông minh hoặc trợ lý ảo để tự động lọc, sắp xếp và gợi ý phản hồi cho người dùng.

Những hướng phát triển trên không chỉ nâng cao hiệu suất mô hình mà còn giúp ứng dụng kết quả nghiên cứu vào thực tiễn một cách hiệu quả và bền vững.

8 Tài liệu tham khảo

1. Machine Learning cơ bản - Vũ Hữu Tiệp.
2. Mining of Massive Datasets - Jure Leskovec, Anand Rajaraman, Jeff.
3. Bài giảng Hệ gợi ý - Socit - Đại học Bách khoa Hà Nội.
4. Machine Learning TextBook - Andreas Lindholm, Niklas Wahlstrom, Fredrik Lindsten, Thomas B.Schon.
5. Bhuyan, R., Borah, S. (2023). *A Survey of Some Density Based Clustering Techniques*.
6. Wang, D., Lu, X., Rinaldo, A. (2017). *DBSCAN: Optimal Rates For Density Based Clustering*.
7. Chakraborty, S., Nagwani, N. K., Dey, L. (2014). *Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms*.