**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

A red and black logo

AI-generated content may be incorrect.

**BÁO CÁO KẾT QUẢ**

**HỌC PHẦN: NGHIÊN CỨU TỐT NGHIỆP 1**

**Đề tài: Hệ thống phân loại thư rác dựa trên các giải thuật Học máy**

|  |  |  |
| --- | --- | --- |
| Lớp | : |  |
| Giảng viên hướng dẫn | : | PGS. TS. Huỳnh Quyết Thắng |
| Sinh viên thực hiện | : | Lê Nhật Sơn |

**HÀ NỘI, 5/2025**

**MỤC LỤC**

[**LỜI NÓI ĐẦU 3**](#_3fkukdgbyovh)

**Chương 1: Giới thiệu 4**

**1.1. Mô tả bài toán thực tế được giải quyết 4**

**1.2. Mục tiêu của đồ án 4**

**1.3. Phạm vi của đồ án 4**

**1.4. Cấu trúc báo cáo 4**

**Chương 2: Cơ sở lý thuyết và Tổng quan tài liệu 4**

**2.1. Tổng quan về Học máy trong Phân loại Văn bản 4**

**2.2. Các kỹ thuật Tiền xử lý Dữ liệu Văn bản 5**

**2.3. Các thuật toán Phân loại được sử dụng 6**

**2.4. Các độ đo đánh giá mô hình (Evaluation Metrics) 7**

**Chương 3: Phương pháp và Thiết kế Hệ thống 8**

**3.1. Kiến trúc tổng quan của hệ thống 8**

**3.2. Dữ liệu sử dụng 8**

**3.3. Quy trình thực nghiệm 10**

**3.4. Các chức năng chính của hệ thống (và cách sử dụng) 11**

**Chương 4: Các phương pháp, gói phần mềm, dữ liệu có sẵn được sử dụng 11**

**4.1. Ngôn ngữ lập trình 11**

**4.2. Môi trường phát triển 11**

**4.3. Các thư viện và gói phần mềm chính 11**

**Đồ án sử dụng các thư viện và gói phần mềm chính dưới đây để phục vụ quá trình xây dựng, huấn luyện, đánh giá và so sánh các mô hình Naive Bayes, Decision Tree, Random Forest, Logistic Regression và Support Vector Machine (SVM): 11**

**1. Scikit-learn  
 Là thư viện nền tảng để triển khai cả 5 mô hình phân loại được sử dụng trong đồ án. Cụ thể, scikit-learn cung cấp: 11**

**● Các thuật toán phân loại: 12**

**○ MultinomialNB() cho mô hình Naive Bayes. 12**

**○ DecisionTreeClassifier() cho cây quyết định. 12**

**○ RandomForestClassifier() cho mô hình rừng ngẫu nhiên. 12**

**○ LogisticRegression() cho hồi quy logistic. 12**

**○ SVC() cho mô hình Support Vector Machines (SVM). 12**

**● Công cụ hỗ trợ huấn luyện và đánh giá mô hình: 12**

**○ Hàm chia dữ liệu: train\_test\_split(), sử dụng stratify để giữ cân bằng lớp. 12**

**○ Các phương pháp tối ưu hóa siêu tham số: GridSearchCV và RandomizedSearchCV. 12**

**○ Kỹ thuật đánh giá hiệu năng mô hình: cross\_val\_score, cross\_validate. 12**

**○ Hàm đánh giá độ chính xác, confusion matrix, và báo cáo phân loại: accuracy\_score, confusion\_matrix, classification\_report. 12**

**● Phương pháp tiền xử lý dữ liệu: 12**

**○ TfidfVectorizer để chuyển đổi văn bản sang vector TF-IDF. 12**

**2. Pandas và NumPy 12**

**● Pandas: xử lý dữ liệu dạng bản (DataFrame, Series,..). Thường dùng để đọc, ghi các files, thao tác và phân tích dữ liệu. 12**

**3. Thư viện xử lý ngôn ngữ tự nhiên (NLP) 12**

**● NLTK: Áp dụng cho bước tiền xử lý chung của dữ liệu văn bản, bao gồm tách từ (split), loại bỏ từ dừng (stop words), chuẩn hóa từ ngữ. Quá trình tiền xử lý được thống nhất và sử dụng chung cho cả 5 mô hình phân loại. 13**

**4. Matplotlib và Seaborn 13**

**● Matplotlib và Seaborn: Sử dụng để trực quan hóa dữ liệu đầu vào, phân tích đặc điểm (như độ dài tin nhắn, phân bố lớp spam-ham) và kết quả đánh giá mô hình (như confusion matrix, các độ đo như accuracy, precision, recall, F1-score), giúp hiểu rõ hiệu quả từng mô hình một cách trực quan nhất. 13**

**5. Các thư viện hỗ trợ khác 13**

**● wordcloud: dùng tạo word cloud, trực quan hóa các từ nổi bật trong email spam và ham. 13**

[● SciPy: dùng để cung cấp các hàm phân bố ngẫu nhiên hỗ trợ quá trình tìm kiếm tham số ngẫu nhiên trong RandomizedSearchCV. 13](#_w3ny581uu0q7)

**4.4. Tập dữ liệu có sẵn 13**

[Nguồn dữ liệu 13](#_dqhipommsyx1)

[Đặc điểm dữ liệu 13](#_uu6qknfv7z6k)

[Phân chia dữ liệu 14](#_nj5mms2ffsin)

**Chương 5: Kết quả thực nghiệm và Đánh giá 14**

**5.1. Kết quả của từng thuật toán 14**

**5.2. So sánh hiệu năng giữa các thuật toán (Baseline) 18**

**5.3. Phân tích và thảo luận kết quả 19**

**Chương 6: Các vấn đề/khó khăn gặp phải và giải pháp 21**

**6.1. Các vấn đề/khó khăn gặp phải trong quá trình thực hiện 21**

**6.2. Cách thức được dùng để giải quyết (vượt qua) 22**

**Chương 7: Các tranh luận / khám phá / kết luận, và các đề cử cho việc tiếp tục phát triển và cải tiến trong tương lai 22**

**7.1. Tranh luận / Khám phá / Kết luận chính 22**

**7.2. Đóng góp của đồ án 22**

**7.3. Hạn chế của đồ án 22**

**7.4. Các đề cử cho việc tiếp tục phát triển và cải tiến trong tương lai 22**

**Tài liệu tham khảo 23**

[**Phụ lục (nếu có) 23**](#_h0is5ebrqfs)

# LỜI NÓI ĐẦU

Trong bối cảnh công nghệ thông tin ngày càng thâm nhập sâu rộng vào mọi mặt của đời sống, thư điện tử (email) đã khẳng định vai trò là một phương tiện liên lạc, trao đổi thông tin thiết yếu. Tuy nhiên, sự phát triển này cũng đi kèm với những thách thức không nhỏ, điển hình là vấn nạn thư rác (spam mail) ngày một gia tăng, gây ảnh hưởng tiêu cực đến hiệu suất làm việc, chiếm dụng tài nguyên lưu trữ và tiềm ẩn các nguy cơ về bảo mật. Nhận thức được tầm quan trọng của việc giải quyết bài toán này, nhóm chúng tôi đã chọn đề tài "Xây dựng Hệ thống Phân loại Mail rác sử dụng các Thuật toán Học máy" cho bài tập lớn môn học Nhập môn Trí tuệ Nhân tạo.

Bài báo cáo này là kết quả của quá trình nghiên cứu và ứng dụng các kiến thức nền tảng về Trí tuệ Nhân tạo vào một vấn đề thực tiễn. Chúng tôi đã tập trung vào việc khám phá và triển khai bốn thuật toán học máy tiêu biểu: Naive Bayes, Random Forest, Logistic Regression, và Support Vector Machines (SVM). Mục tiêu cốt lõi là xây dựng các mô hình có khả năng phân biệt hiệu quả giữa thư hợp lệ (ham) và thư rác (spam), đồng thời tiến hành so sánh, đánh giá để tìm ra ưu nhược điểm của từng phương pháp trong bối cảnh cụ thể của bài toán.

Quá trình từ ý tưởng đến hoàn thiện sản phẩm là một hành trình học hỏi đầy giá trị. Chúng tôi không chỉ được củng cố kiến thức lý thuyết mà còn rèn luyện kỹ năng phân tích vấn đề, xử lý dữ liệu, lập trình và làm việc nhóm. Đồ án này không chỉ là một bài tập kỹ thuật đơn thuần mà còn là nơi chúng tôi thể hiện sự tìm tòi, sáng tạo và nỗ lực vượt qua những khó khăn gặp phải.

Với những kết quả đạt được, nhóm chúng tôi hy vọng bài báo cáo sẽ đóng góp một phần nhỏ vào việc hiểu rõ hơn về ứng dụng của các thuật toán học máy trong lĩnh vực phân loại văn bản nói chung và bài toán phát hiện thư rác nói riêng. Đồng thời, chúng tôi mong muốn đây sẽ là nền tảng cho những cải tiến và phát triển sâu hơn trong tương lai, góp phần xây dựng một không gian mạng an toàn và hiệu quả hơn.

## Chương 1: Giới thiệu

### 1.1. Mô tả bài toán thực tế được giải quyết

* Mô tả chi tiết bài toán phân loại mail rác: Bài toán phân loại email rác (spam detection) nhằm tự động phân loại các email thành hai nhóm: "spam" và "ham". Đầu vào là các email chứa nội dung dưới dạng text, đầu ra mong muốn là các email đã được phân loại ("spam" - "ham").
* Các thách thức cụ thể của bài toán :
  + **Sự thay đổi liên tục của nội dung thư rác**: Thư spam thường được cập nhật với các từ khóa mới hoặc kỹ thuật ngụy trang (obfuscation) để tránh bị phát hiện.
  + **Ngôn ngữ tự nhiên phức tạp**: Email có thể sử dụng ngôn ngữ không chuẩn (lỗi chính tả, từ lóng) gây khó khăn trong việc trích xuất đặc trưng.

### 1.2. Mục tiêu của đồ án

* Mục tiêu chính: Xây dựng và đánh giá hiệu quả của các mô hình học máy trong việc phân loại thư rác.
* Các mục tiêu cụ thể:
  + Nghiên cứu và tìm hiểu về các thuật toán Naive Bayes, Decision Tree, Random Forest, Logistic Regression, Support Vector Machines (SVM).
  + Thu thập và tiền xử lý dữ liệu email.
  + Huấn luyện và tối ưu hóa các mô hình.
  + So sánh hiệu năng của các mô hình dựa trên các độ đo phù hợp.

### 1.3. Phạm vi của đồ án

* Giới hạn về tập dữ liệu sử dụng: Đồ án sử dụng một tập dữ liệu email có sẵn được giới hạn bởi ngôn ngữ tiếng Anh và không bao gồm các email đa phương tiện phức tạp (như hình ảnh hoặc tệp đính kèm).
* Giới hạn về các thuật toán được triển khai: Đồ án tập trung vào 5 thuật toán: Naive Bayes, Decision Tree, Random Forest, Logistic Regression, Support Vector Machines (SVM).
* Giới hạn về các tính năng của hệ thống: Hệ thống không bao gồm giao diện người dùng (GUI) hoặc tích hợp thực tế với máy chủ email (như Gmail API).

### 1.4. Cấu trúc báo cáo

* Mô tả ngắn gọn nội dung của từng chương trong báo cáo.

## Chương 2: Cơ sở lý thuyết và Tổng quan tài liệu

### 2.1. Tổng quan về Học máy trong Phân loại Văn bản

* Giới thiệu về học máy (Machine Learning): Học máy là một nhánh của trí tuệ nhân tạo (AI), tập trung vào việc xây dựng các thuật toán cho phép máy tính học hỏi từ dữ liệu mà không cần lập trình tường minh. Thay vì sử dụng các quy tắc cố định, học máy dựa trên việc tìm kiếm các mẫu (patterns) trong dữ liệu để đưa ra dự đoán hoặc quyết định. Học máy đóng vai trò quan trọng trong nhiều lĩnh vực, đặc biệt là xử lý ngôn ngữ tự nhiên (NLP) và phân loại văn bản.
* Các loại bài toán học máy:
  + Học có giám sát (Supervised Learning): Sử dụng dữ liệu đã được gắn nhãn (labeled data) để huấn luyện mô hình. Ví dụ: Phân loại email rác (spam/ham) là bài toán học có giám sát, với đầu vào là nội dung email và đầu ra là nhãn "spam" hoặc "ham".
  + Học không giám sát (Unsupervised Learning): Làm việc với dữ liệu không có nhãn, nhằm tìm kiếm cấu trúc hoặc mẫu tiềm ẩn, chẳng hạn như phân cụm (clustering) các email tương tự.
  + Học tăng cường (Reinforcement Learning): Mô hình học thông qua thử và sai, nhận phản hồi từ môi trường để tối ưu hóa hành động, thường áp dụng trong các hệ thống tự động hóa.
* Ứng dụng của học máy trong xử lý ngôn ngữ tự nhiên và phân loại văn bản: Học máy được ứng dụng rộng rãi trong NLP, bao gồm các bài toán như phân loại văn bản (ví dụ: phân loại email rác), phân tích cảm xúc (sentiment analysis), dịch máy, và nhận dạng thực thể (NER). Trong phân loại văn bản, học máy giúp tự động hóa việc gán nhãn cho các tài liệu, chẳng hạn như xác định email là "spam" hay "ham" dựa trên nội dung, tiêu đề, và các đặc trưng khác.
* Quy trình chung của một bài toán phân loại văn bản:
  + Thu thập dữ liệu: Tập hợp dữ liệu văn bản (ví dụ: tập dữ liệu email) với nhãn tương ứng (spam/ham).
  + Tiền xử lý dữ liệu: Làm sạch và chuẩn hóa văn bản (loại bỏ ký tự đặc biệt, chuyển về chữ thường, tách từ, v.v.).
  + Trích xuất đặc trưng (Feature Extraction): Chuyển văn bản thành dạng số mà mô hình có thể hiểu (ví dụ: TF-IDF, Word Embeddings).
  + Huấn luyện mô hình: Sử dụng thuật toán học máy (như Naive Bayes, SVM) để huấn luyện trên tập dữ liệu.
  + Đánh giá mô hình: Sử dụng các độ đo như accuracy, F1-score để đánh giá hiệu quả.
  + Triển khai và tối ưu: Tinh chỉnh mô hình và áp dụng vào thực tế.

### 2.2. Các kỹ thuật Tiền xử lý Dữ liệu Văn bản

* Loại bỏ ký tự đặc biệt, số: Loại bỏ các ký tự không cần thiết như dấu câu (@, #, $, v.v.) và số (123, 456), vì chúng thường không mang ý nghĩa trong phân loại văn bản. Ví dụ: "Win $1000!!!" sẽ được chuyển thành "Win".
* Chuyển về chữ thường (Lowercase): Chuyển toàn bộ văn bản về chữ thường để đảm bảo tính đồng nhất, tránh phân biệt giữa "WIN" và "win". Ví dụ: "Hello WORLD" thành "hello world".
* Tách từ (Tokenization): Chia văn bản thành các đơn vị từ (tokens). Ví dụ: "I love coding" được tách thành ["I", "love", "coding"]. Điều này giúp mô hình xử lý từng từ riêng lẻ.
* Loại bỏ từ dừng (Stop words removal):Loại bỏ các từ dừng (stop words) như "the", "is", "and" vì chúng không mang ý nghĩa phân loại, giảm kích thước dữ liệu và nhiễu. Ví dụ: "I am the best" sau khi loại stop words còn ["best"].
* Stemming/Lemmatization.
* Vector hóa văn bản:
  + Bag of Words (BoW)
  + TF-IDF (Term Frequency-Inverse Document Frequency)
  + Word Embeddings (Word2Vec, GloVe - nếu có sử dụng)

### 2.3. Các thuật toán Phân loại được sử dụng

* **2.3.1. Naive Bayes**
  + Nguyên lý hoạt động, công thức Bayes:

* + Các biến thể (Multinomial Naive Bayes, Bernoulli Naive Bayes).
  + Ưu điểm: Tính toán nhanh, hiệu quả với dữ liệu văn bản, ít nhạy cảm với dữ liệu không cân bằng.
  + Nhược điểm: Giả định độc lập giữa các từ không chính xác (ví dụ: "free money" có ý nghĩa khác "free").
* **2.3.2. Decision Tree**
  + Nguyên lý hoạt động: Decision Tree (Cây quyết định) chia không gian đặc trưng thành các vùng dựa trên các câu hỏi nhị phân (ví dụ: "Từ 'free' có xuất hiện không?"). Mỗi nút lá (leaf node) đại diện cho một nhãn (spam/ham).
  + Ưu điểm: Dễ hiểu, không yêu cầu chuẩn hóa dữ liệu, có thể xử lý cả đặc trưng phi tuyến.
  + Nhược điểm: Dễ bị overfitting, đặc biệt với dữ liệu văn bản có chiều cao (high-dimensional).
* **2.3.3. Random Forest**
  + Khái niệm về cây quyết định (Decision Tree):Decision Tree là nền tảng của Random Forest.
  + Nguyên lý hoạt động của Random Forest (kết hợp nhiều cây quyết định):Random Forest là một phương pháp tập hợp (ensemble), kết hợp nhiều cây quyết định (Decision Trees) để tăng độ chính xác và giảm overfitting. Mỗi cây được huấn luyện trên một tập con ngẫu nhiên của dữ liệu (bagging) và một tập con ngẫu nhiên của đặc trưng. Kết quả cuối cùng được quyết định bằng đa số phiếu (majority vote).
  + Ưu điểm: Giảm nguy cơ overfitting, hiệu quả cao với dữ liệu phức tạp, ít nhạy cảm với nhiễu.
  + Nhược điểm: Tốn tài nguyên tính toán hơn Decision Tree, khó diễn giải.
* **2.3.4. Logistic Regression**
  + Nguyên lý hoạt động:Logistic Regression dự đoán xác suất một email thuộc lớp "spam" hoặc "ham" bằng cách sử dụng hàm sigmoid.
  + Cách mô hình ước lượng xác suất: Mô hình tối ưu hóa hàm mất mát (log-loss) để tìm các trọng số βi​ sao cho dự đoán gần với nhãn thực tế nhất.
  + Ưu điểm: Hiệu quả với dữ liệu tuyến tính, dễ triển khai, cung cấp xác suất dự đoán.
  + Nhược điểm: Hiệu suất kém với dữ liệu phi tuyến, nhạy cảm với đặc trưng không liên quan.
* **2.3.5. Support Vector Machines (SVM)**
  + Nguyên lý hoạt động: SVM tìm mặt phẳng siêu phân cách tối ưu để tách hai lớp (spam/ham), sao cho khoảng cách giữa mặt phẳng và các điểm gần nhất (support vectors) là lớn nhất.
  + Khái niệm về margin, support vectors:
  + Margin: Khoảng cách từ mặt phẳng siêu phân cách đến các điểm gần nhất của hai lớp.
  + Support Vectors: Các điểm dữ liệu gần mặt phẳng nhất, quyết định vị trí của mặt phẳng.
  + Các loại kernel (linear, polynomial, RBF).
  + Linear Kernel: Dùng khi dữ liệu có thể phân tách tuyến tính.
  + Polynomial Kernel: Dùng cho dữ liệu phi tuyến, ánh xạ dữ liệu lên không gian bậc cao hơn.
  + RBF Kernel (Radial Basis Function): Phổ biến nhất, ánh xạ dữ liệu phi tuyến bằng hàm Gaussian.
  + Ưu điểm: Hiệu quả với dữ liệu chiều cao, ít nhạy cảm với nhiễu, linh hoạt với các kernel.
  + Nhược điểm: Tốn tài nguyên tính toán, khó tối ưu tham số (như C và gamma).

### 2.4. Các độ đo đánh giá mô hình (Evaluation Metrics)

* Accuracy: Tỷ lệ dự đoán đúng trên toàn bộ tập dữ liệu
* Precision: Tỷ lệ dự đoán đúng trong số các dự đoán là "spam"
* Recall (Sensitivity): Tỷ lệ email "spam" được phát hiện đúng
* F1-Score: Trung bình điều hòa giữa Precision và Recall, phù hợp khi cần cân bằng hai độ đo
* Confusion Matrix: Bảng ma trận hiển thị số lượng dự đoán đúng và sai cho từng lớp

## Chương 3: Phương pháp và Thiết kế Hệ thống

### 3.1. Kiến trúc tổng quan của hệ thống

* Sơ đồ khối mô tả các thành phần chính của hệ thống và luồng xử lý dữ liệu.
  + Module thu thập dữ liệu.
  + Module tiền xử lý dữ liệu.
  + Module huấn luyện mô hình (bao gồm các thuật toán).
  + Module đánh giá mô hình.
  + Module phân loại (dự đoán cho email mới).
  + Giao diện người dùng.

### 3.2. Dữ liệu sử dụng

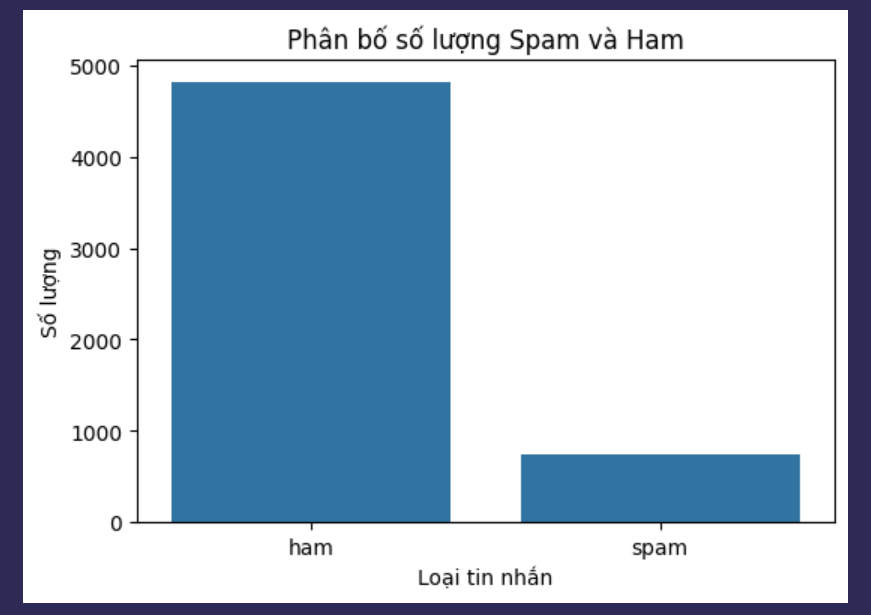
* **3.2.1. Mô tả tập dữ liệu**
  + Nguồn gốc tập dữ liệu: tự thu thập
  + Số lượng mẫu, tỷ lệ giữa mail spam và ham.

Phân bố số mẫu Spam/Ham:

ham 4825

spam 747

Tỷ lệ giữa mail spam và ham:



* + Định dạng dữ liệu (data type)

- Data type: object

+ object là kiểu dữ liệu tổng quát trong pandas, thường dùng cho chuỗi ký tự (string) hoặc dữ liệu không phải số (có thể là text, hỗn hợp kiểu, v.v.).

+ Sau khi xử lý cột của Dataframe, cả hai cột Category và Message đều chứa dữ liệu dạng text (chuỗi), nên data type là object.

* + Mô tả các trường thông tin có trong dữ liệu (nội dung, tiêu đề,...).

Các trường thông tin có trong dữ liệu:

1. Category   
Cột Category có thể chứa 2 giá trị, dùng để phân loại mail: spam hoặc ham

2. Message

Cột Message chứa nội dung của mail ở dạng text

* **3.2.2. Phân chia dữ liệu**
  + Cách chia tập dữ liệu thành tập huấn luyện (training set), tập kiểm định (validation set - nếu có) và tập kiểm thử (test set).

Sử dụng K-Fold Cross Validation (Kiểm định chéo K lần) để phân chia dữ liệu thành tập training set và validation test

Tập kiểm thử:

2 phương pháp Cross Validation:

+ GridSearchCV: thử tất cả các tổ hợp

+ RandomizedSearchCV: thử một số tổ hợp tham số ngẫu nhiên.

* + Tỷ lệ phân chia.

+ Chia dữ liệu: Chia ngẫu nhiên tập train thành K phần, giá trị K thường là 5 hoặc 10

+ Lặp K lần: Quá trình huấn luyện và đánh giá được lặp lại K lần. Trong mỗi lần lặp (ví dụ lần lặp thứ i):

* Chọn Fold kiểm tra: Fold thứ i được giữ lại làm tập dữ liệu kiểm định (validation set) hoặc là "test fold" cho lần lặp này.
* Chọn fold huấn luyện: K-1 phần còn lại được gộp lại để làm tập dữ liệu huấn luyện.
* Huấn luyện: Mô hình học máy được huấn luyện trên tập huấn luyện (k-1 folds)

+ Đánh giá: Mô hình vừa huấn luyện xong được đánh giá hiệu năng (tính f1-score, accuracy,..) trên tập kiểm định (fold thứ i). Kết quả đánh giá được lưu lại.

→ Tổng hợp kết quả: Sau khi thực hiện K lần lặp thì có K kết quả đánh giá. Hiệu năng cuối cùng của mô hình theo phương pháp Cross Validation thường được tính bằng lấy trung bình của K kết quả.

### 3.3. Quy trình thực nghiệm

* **3.3.1. Tiền xử lý dữ liệu chi tiết**
  + Mô tả cụ thể các bước tiền xử lý đã áp dụng cho tập dữ liệu của nhóm.

1. Đầu tiên, chuyển toàn bộ dữ liệu trong cột Message thành chữ thường.

2. Xóa dấu câu, số, ký tự đặc biệt (chỉ giữ chữ cái và khoảng trắng)

3. Tách từ (tokenize đơn giản bằng split)

4. Xóa stop words

5. Nối lại thành chuỗi

* + Các công cụ, thư viện đã sử dụng cho từng bước.

- Các thư viện đã import: sys, os, re, nltk, pandas

- Cách import stopwords: from nltk.corpus import stopwords

* **3.3.2. Huấn luyện mô hình**
  + Mô tả chi tiết quá trình huấn luyện cho từng thuật toán:
    - Naive Bayes: Lựa chọn biến thể, các tham số (nếu có).
    - Decision Tree: Lựa chọn chỉ số phân loại (entropy, gini index).
    - Random Forest: Số lượng cây, độ sâu tối đa của cây, các tham số khác.
    - Logistic Regression: Hàm tối ưu, tham số điều chuẩn (regularization).
    - SVM: Lựa chọn kernel, tham số C, gamma (nếu dùng RBF kernel).
  + Cách lựa chọn tham số (ví dụ: Grid Search, Random Search).
* **3.3.3. Đánh giá và so sánh mô hình**
  + Các độ đo được sử dụng để đánh giá.
  + Cách thức so sánh hiệu năng giữa các mô hình.

### 3.4. Các chức năng chính của hệ thống (và cách sử dụng)

* Mô tả các chức năng mà hệ thống của bạn cung cấp. Ví dụ:
  + Chức năng nhập một email mới để phân loại.
  + Chức năng hiển thị kết quả phân loại (spam/ham) cùng với độ tin cậy (nếu có).
  + Chức năng huấn luyện lại mô hình với dữ liệu mới (nếu có).
* Hướng dẫn sử dụng từng chức năng (có thể kèm theo ảnh chụp màn hình nếu có giao diện).

## Chương 4: Các phương pháp, gói phần mềm, dữ liệu có sẵn được sử dụng

### 4.1. Ngôn ngữ lập trình

Trong đồ án này, ngôn ngữ lập trình chính được sử dụng đó là **Python (phiên bản 3.12.7)**. Do python là ngôn ngữ phổ biến trong lĩnh vực học máy và xử lý ngôn ngữ tự nhiên nhờ sự hỗ trợ từ nhiều thư viện phong phú, có cú pháp rõ ràng và còn được cộng đồng lớn hỗ trợ.

### 4.2. Môi trường phát triển

Quá trình phát triển và thử nghiệm các mô hình trên được thực hiện trong các môi trường sau:

* **Jupyter Notebook**: giúp tổ chức và chạy các đoạn mã Python một cách trực quan, chia các đoạn code thành nhiều phần theo tác vụ, thuận tiện cho việc thử nghiệm, sửa lỗi, trực quan hóa và ghi chú lại kết quả của từng phần.
* **Google Colab**: được sử dụng khi có lượng dữ liệu lớn, cần tài nguyên phần cứng lớn hơn như GPU, giúp đẩy nhanh tốc độ huấn luyện mô hình.
* **Visual Studio Code (VS Code)**: là công cụ soạn thảo mã nguồn, hỗ trợ quản lý dự án và viết các module Python.

### 4.3. Các thư viện và gói phần mềm chính

### Đồ án sử dụng các thư viện và gói phần mềm chính dưới đây để phục vụ quá trình xây dựng, huấn luyện, đánh giá và so sánh các mô hình Naive Bayes, Decision Tree, Random Forest, Logistic Regression và Support Vector Machine (SVM):

### 1. Scikit-learn Là thư viện nền tảng để triển khai cả 5 mô hình phân loại được sử dụng trong đồ án. Cụ thể, scikit-learn cung cấp:

### Các thuật toán phân loại:

### MultinomialNB() cho mô hình Naive Bayes.

### DecisionTreeClassifier() cho cây quyết định.

### RandomForestClassifier() cho mô hình rừng ngẫu nhiên.

### LogisticRegression() cho hồi quy logistic.

### SVC() cho mô hình Support Vector Machines (SVM).

### Công cụ hỗ trợ huấn luyện và đánh giá mô hình:

### Hàm chia dữ liệu: train\_test\_split(), sử dụng stratify để giữ cân bằng lớp.

### Các phương pháp tối ưu hóa siêu tham số: GridSearchCV và RandomizedSearchCV.

### Kỹ thuật đánh giá hiệu năng mô hình: cross\_val\_score, cross\_validate.

### Hàm đánh giá độ chính xác, confusion matrix, và báo cáo phân loại: accuracy\_score, confusion\_matrix, classification\_report.

### Phương pháp tiền xử lý dữ liệu:

### TfidfVectorizer để chuyển đổi văn bản sang vector TF-IDF.

### 2. Pandas và NumPy

### Pandas: xử lý dữ liệu dạng bản (DataFrame, Series,..). Thường dùng để đọc, ghi các files, thao tác và phân tích dữ liệu.

* NumPy: Dùng để hỗ trợ xử lý số liệu và tính toán trên các ma trận và mảng số liệu, cung cấp dữ liệu đầu vào phù hợp cho các thuật toán học máy.

### 3. Thư viện xử lý ngôn ngữ tự nhiên (NLP)

### NLTK: Áp dụng cho bước tiền xử lý chung của dữ liệu văn bản, bao gồm tách từ (split), loại bỏ từ dừng (stop words), chuẩn hóa từ ngữ. Quá trình tiền xử lý được thống nhất và sử dụng chung cho cả 5 mô hình phân loại.

### 4. Matplotlib và Seaborn

### Matplotlib và Seaborn: Sử dụng để trực quan hóa dữ liệu đầu vào, phân tích đặc điểm (như độ dài tin nhắn, phân bố lớp spam-ham) và kết quả đánh giá mô hình (như confusion matrix, các độ đo như accuracy, precision, recall, F1-score), giúp hiểu rõ hiệu quả từng mô hình một cách trực quan nhất.

### 5. Các thư viện hỗ trợ khác

### wordcloud: dùng tạo word cloud, trực quan hóa các từ nổi bật trong email spam và ham.

### SciPy: dùng để cung cấp các hàm phân bố ngẫu nhiên hỗ trợ quá trình tìm kiếm tham số ngẫu nhiên trong RandomizedSearchCV.

### 4.4. Tập dữ liệu có sẵn

### Nguồn dữ liệu

Đồ án sử dụng tập dữ liệu email đã được gán nhãn rõ ràng với hai lớp: "spam" và "ham". Dữ liệu có nguồn gốc từ bộ dữ liệu chuẩn trong lĩnh vực phát hiện thư rác (UCI SpamAssassin), được phép sử dụng cho mục đích nghiên cứu và học thuật.

### Đặc điểm dữ liệu

* Tập dữ liệu gồm hai cột chính:  
  + **Category**: nhãn phân loại ("spam" hoặc "ham").
  + **Message**: nội dung email dưới dạng văn bản thô.
* Dữ liệu có tổng số mẫu là hơn 5,500 email, trong đó có khoảng 747 email spam và 4825 email ham, cho thấy dữ liệu bị mất cân bằng về số lượng giữa hai lớp.
* Cột dữ liệu đều thuộc kiểu object (chuỗi ký tự), phù hợp cho xử lý văn bản.

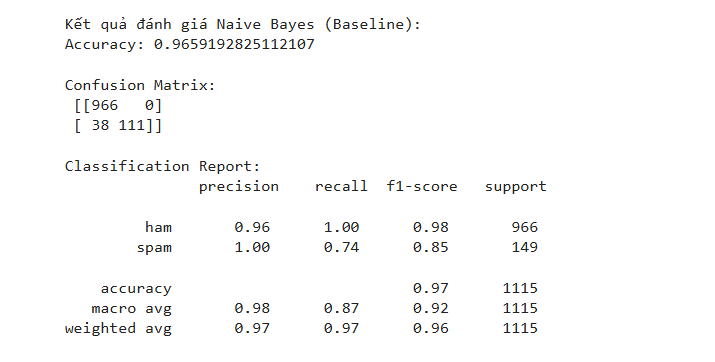
### Phân chia dữ liệu

* Dữ liệu được chia thành tập huấn luyện và kiểm tra theo tỷ lệ 80% và 20%.
* Quá trình chia dữ liệu có sử dụng tham số ‘stratify’ để giữ tỷ lệ lớp spam và ham cân bằng trong các tập con.
* Ngoài ra, áp dụng kỹ thuật **K-Fold Cross Validation** trong quá trình huấn luyện với các phương pháp **GridSearchCV** hoặc **RandomizedSearchCV** để tối ưu siêu tham số mô hình một cách hiệu quả và tránh overfitting.

## Chương 5: Kết quả thực nghiệm và Đánh giá

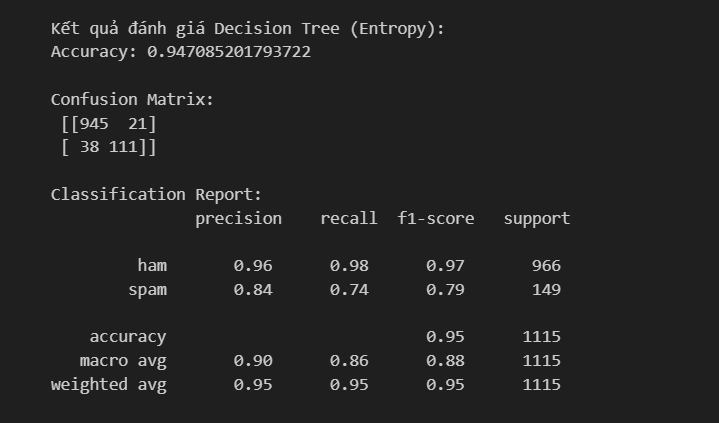
### 5.1. Kết quả của từng thuật toán

* Trình bày kết quả chi tiết cho từng mô hình đã huấn luyện:
  + **5.1.1. Naive Bayes**
    - Bảng Confusion Matrix.
    - Các giá trị Accuracy, Precision, Recall, F1-Score.

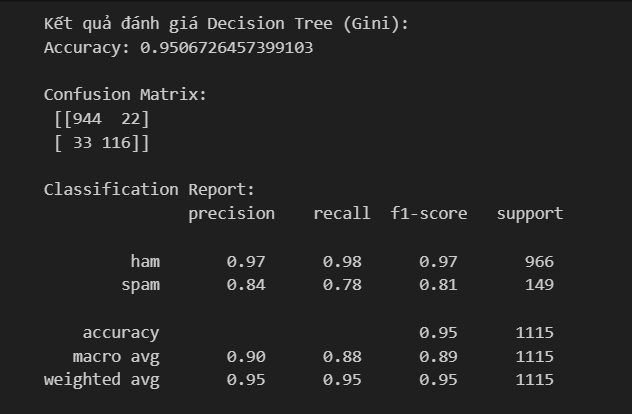


*Kết quả huấn luyện mô hình Naive Bayes (Base-line)*

* + **5.1.2. Decision Tree**
    - Bảng Confusion Matrix.
    - Các giá trị Accuracy, Precision, Recall, F1-Score.

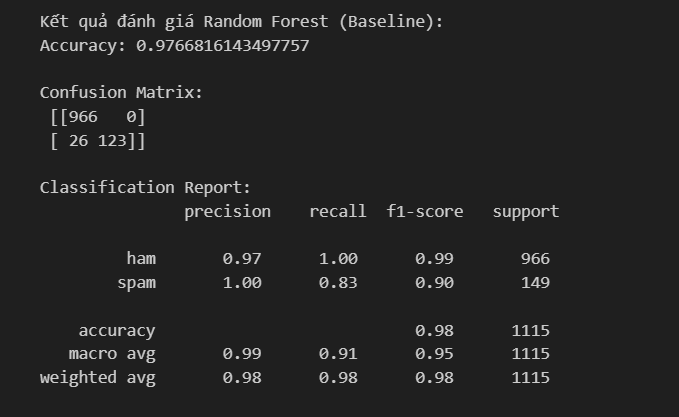


*Kết quả huấn luyện mô hình Decision Tree (Entropy)*

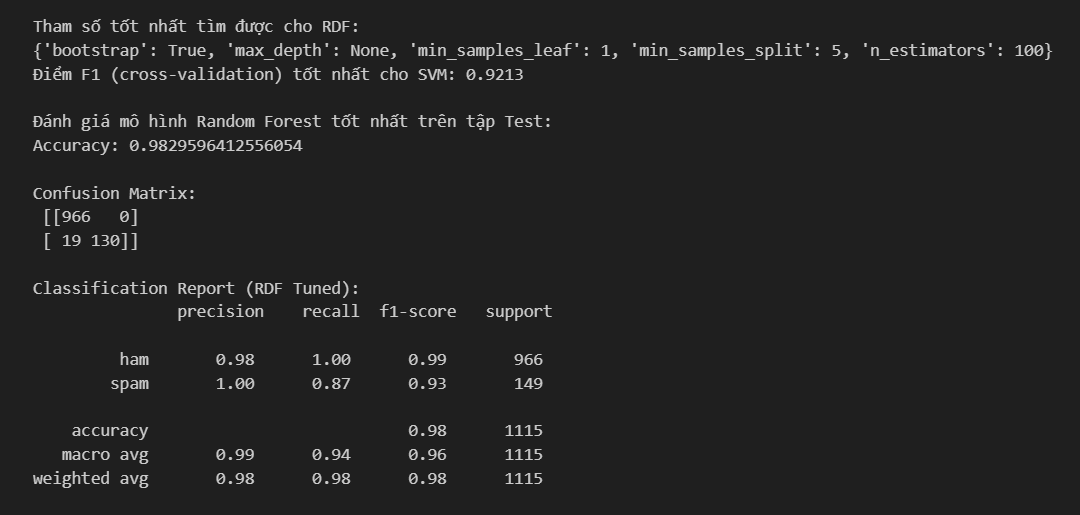


*Kết quả huấn luyện mô hình Decision Tree (Gini Index)*

* + **5.1.3. Random Forest**
    - Tương tự như trên.

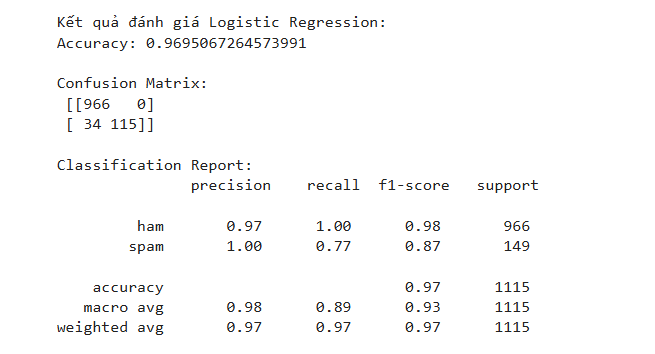


*Kết quả huấn luyện mô hình Ranom Forest (Baseline)*

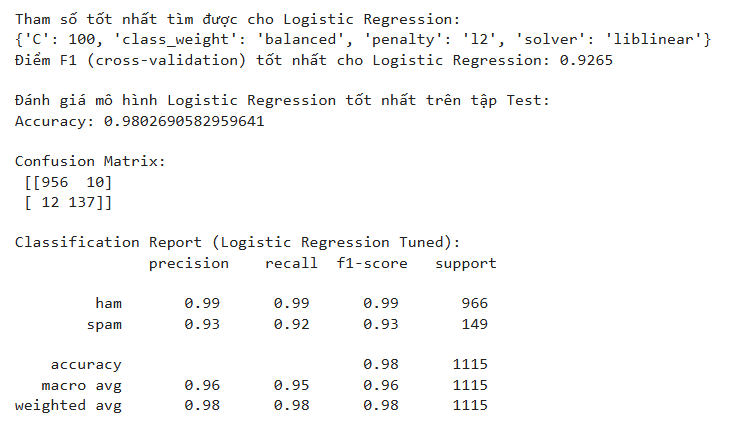
**

*Kết quả huấn luyện mô hình Ranom Forest (Tuning)*

* + **5.1.4. Logistic Regression**
    - Tương tự như trên.

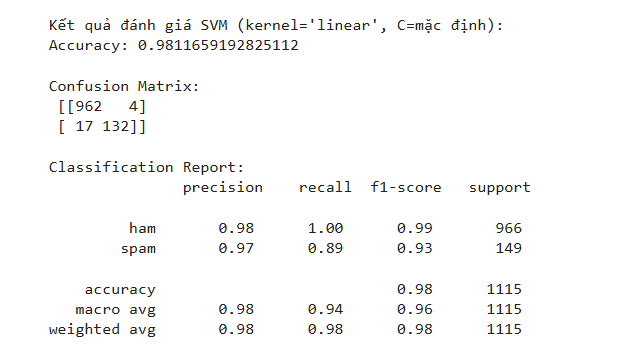


*Kết quả huấn luyện mô hình Logistic Regression (Base-line)*

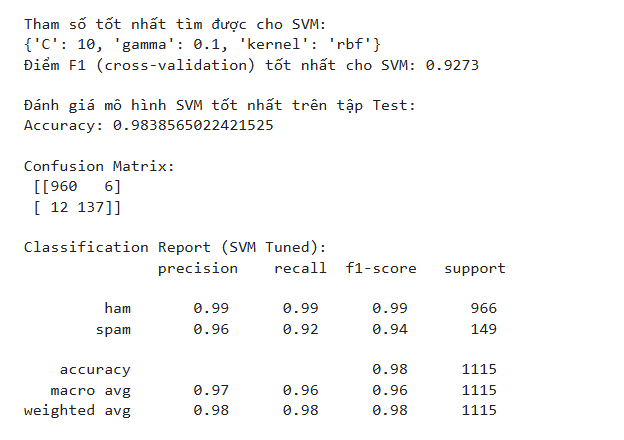


*Kết quả huấn luyện mô hình Logistic Regression (Tuning)*

* + **5.1.5. Support Vector Machines**
    - Tương tự như trên.



*Kết quả huấn luyện mô hình SVM (Base-line)*



*Kết quả huấn luyện mô hình SVM (Tuning)*

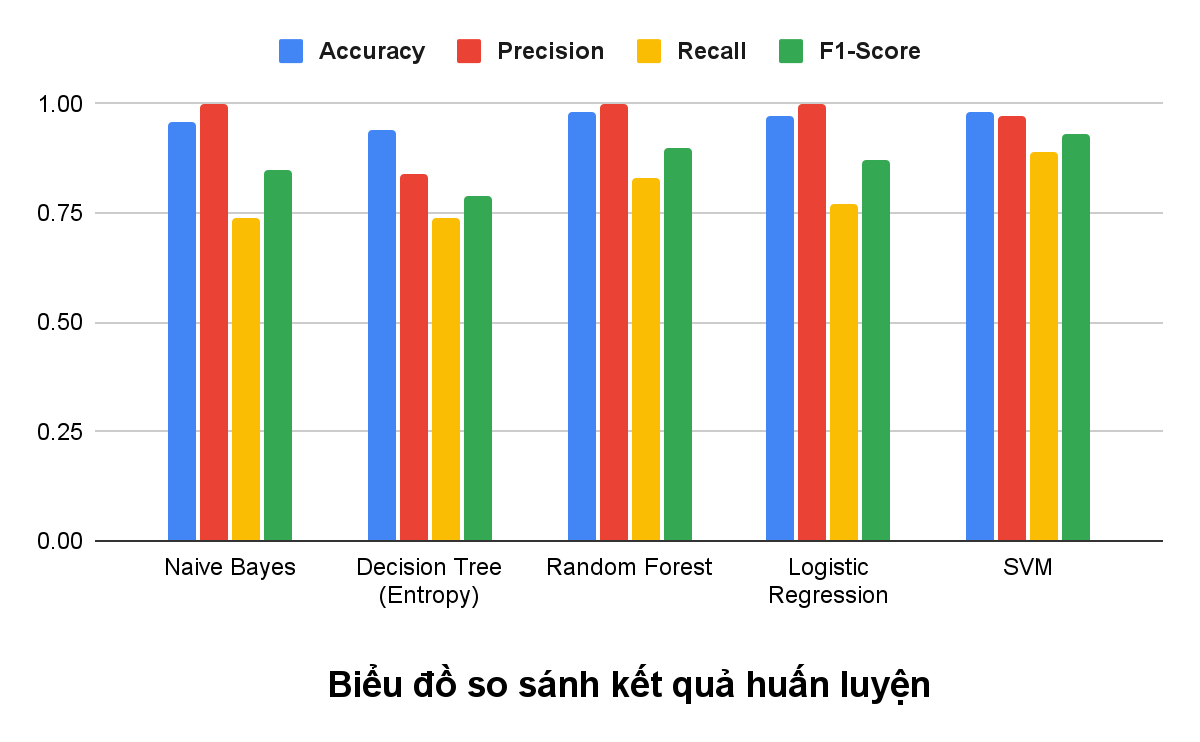
* Có thể trình bày dưới dạng bảng biểu để dễ so sánh.

### 5.2. So sánh hiệu năng giữa các thuật toán (Baseline)

* Lập bảng tổng hợp so sánh các độ đo (Accuracy, Precision, Recall, F1-Score, AUC) của tất cả các thuật toán.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|  |  |  |  |  |
| **Naive Bayes** | 0.96 | 1 | 0.74 | 0.85 |
| **Decision Tree (Entropy)** | 0.94 | 0.84 | 0.74 | 0.79 |
| **Random Forest** | 0.98 | 1 | 0.83 | 0.9 |
| **Logistic Regression** | 0.97 | 1 | 0.77 | 0.87 |
| **SVM** | 0.98 | 0.97 | 0.89 | 0.93 |

* Trực quan hóa sự so sánh bằng biểu đồ cột.



* Nhận xét về thuật toán nào cho kết quả tốt nhất trên tập dữ liệu của nhóm và với các độ đo đã chọn.
  + Từ kết quả, ta có thể rút ra được thứ tự các thuật toán cho kết quả tốt nhất theo chiều giảm dần độ recall là:

SVM > Random Forest > Decision Tree (Gini) > Logistic regression > Naive Bayes ~ Decision Tree (Entropy)

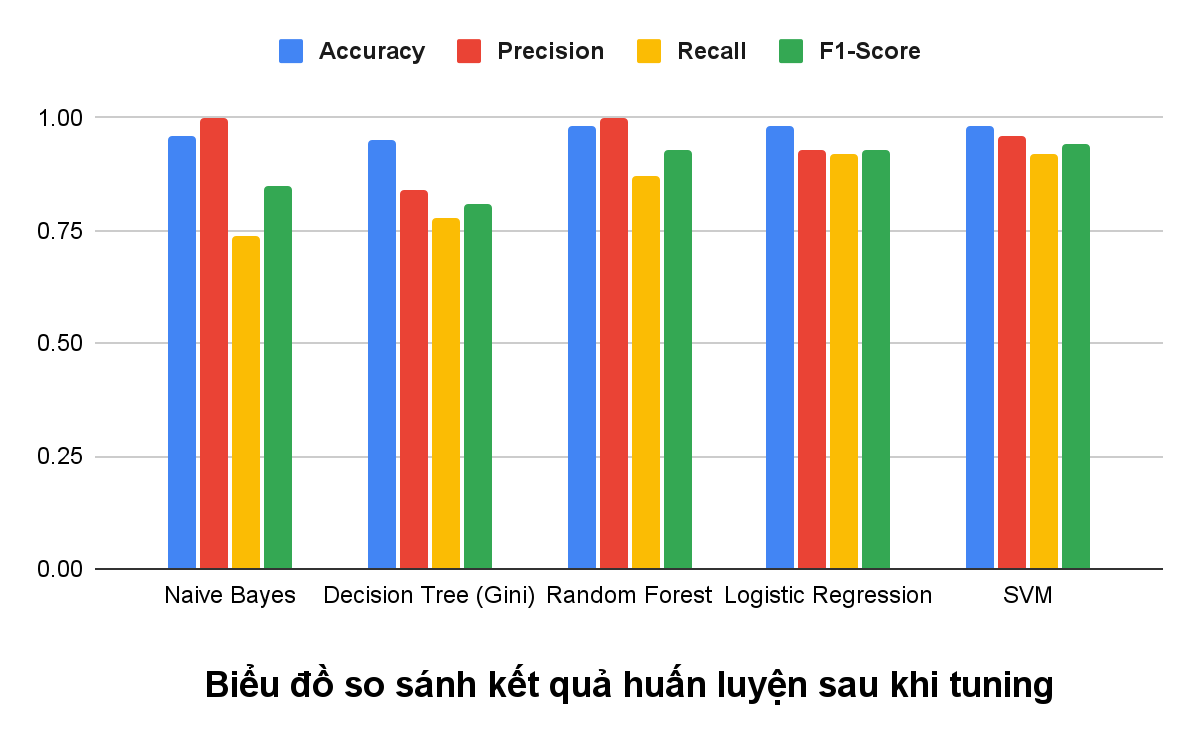
### 5.3. Phân tích và thảo luận kết quả

* Tiêu chí đánh giá: Tỷ lệ thu hồi (recall) hay còn gọi là tỷ lệ dương tính thật (TPR)

Trong một tập dữ liệu mất cân bằng, trong đó số lượng thực tế là dương tính rất thấp, tỷ lệ thu hồi là một chỉ số có ý nghĩa hơn so với độ chính xác vì chỉ số này đo lường khả năng của mô hình xác định chính xác tất cả các thực thể dương tính.

* Phân tích tại sao một số thuật toán hoạt động tốt hơn những thuật toán khác trong trường hợp cụ thể này.
  + SVM thường cho ra kết quả tốt hơn các thuật toán khác và trong trường hợp này cũng thế là do SVM có thể xử lý tốt các loại dữ liệu phi tuyến (trong trường hợp này là dữ liệu văn bản) nhờ kernel trick, đồng thời tập trung vào vecto và biên phân cách nên có khả năng tránh nhiễu tốt hơn các thuật toán khác.
  + Logistic Regression hoạt động tốt với dữ liệu nhiều chiều như văn bản và ít xảy ra overfit do có ít các tham số, nên đầu ra của nó tốt hơn 2 thuật toán cơ bản là Naive Bayes và Decision Tree (entropy). Tuy nhiên, dữ liệu đầu vào không phải tuyến tính nên kết quả đầu ra không thực sự tốt nhất.
  + Random Forest là một phiên bản nâng cao của Decision Tree tuy nhiên khi quyết định đặc trưng của cây, nó có thể bỏ sót một số đặc trưng. Vì vậy mà chỉ số recall cách SVM một khoảng khá lớn.
* Ảnh hưởng của các bước tiền xử lý đến kết quả.
* Ảnh hưởng của việc lựa chọn tham số.
  + Sau khi tinh chỉnh các tham số, ta sẽ được bảng sau:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| **Naive Bayes** | 0.96 | 1 | 0.74 | 0.85 |
| **Decision Tree (Gini)** | 0.95 | 0.84 | 0.78 | 0.81 |
| **Random Forest** | 0.98 | 1 | 0.87 | 0.93 |
| **Logistic Regression** | 0.98 | 0.93 | 0.92 | 0.93 |
| **SVM** | 0.98 | 0.96 | 0.92 | 0.94 |



* + Ta có thể thấy ngoài Naive Bayes không thực hiện tuning thì các thuật toán khác đều cho ra kết quả tốt hơn.
  + Khi sử dụng Gini Index để đánh giá mặc dù accuracy của mô hình không có sự thay đổi đáng kể so với khi sử dụng entropy, nhưng chỉ tỷ lệ thu hồi(recall) lại tăng lên đáng kể, điều này có thể là do sự mất cân bằng trong dữ liệu.
* Các trường hợp mà mô hình dự đoán sai (ví dụ: false positives, false negatives) và phân tích nguyên nhân (nếu có thể).
  + Một số trường hợp tiêu biểu mà mô hình đưa ra dự đoán sai
  + Nguyên nhân có thể là do:
    - Độ lệch hoặc nhiễu trong dữ liệu, bao gồm cả độ sai lệch lấy mẫu cho tập huấn luyện, một số mẫu có chứa rất nhiều từ có tần suất xuất hiện cao trong các mẫu spam nhưng khi đưa ra dự đoán thì mô hình lại kết luận là ham.
    - Quy chuẩn hoá quá mạnh, tức là mô hình bị đơn giản hoá quá mức và bị mất một số độ phức tạp cần thiết, tập dữ liệu được đưa vào chỉ gồm 2 cột, trong đó chỉ có 1 cột chứa nội dung dưới dạng văn bản.
    - Lỗi trong quy trình huấn luyện mô hình

## Chương 6: Các vấn đề/khó khăn gặp phải và giải pháp

### 6.1. Các vấn đề/khó khăn gặp phải trong quá trình thực hiện

* **Về dữ liệu:**
  + Dữ liệu tương đối mất cân bằng nên các thông số như độ chính xác tổng thể (accuracy) bị cao giả tạo, không phản ánh đúng hiệu năng thực của mô hình.
* **Về tiền xử lý:**
  + Tìm phương pháp tiền xử lý tối ưu sao cho không làm giảm ngữ cảnh quan trọng của một văn bản mà vẫn loại bỏ và làm sạch dữ liệu thô được.
* **Về lựa chọn và huấn luyện mô hình:**
  + Hiểu rõ và lựa chọn tham số tối ưu cho từng thuật toán.
* **Về đánh giá mô hình:**
  + Lựa chọn độ đo đánh giá phù hợp với mục tiêu bài toán: f1-score/ precision/ recall

### 6.2. Cách thức được dùng để giải quyết (vượt qua)

* Mô tả chi tiết các giải pháp, kỹ thuật mà nhóm đã áp dụng để giải quyết từng khó khăn đã nêu ở trên.
  + Ví dụ: Sử dụng kỹ thuật SMOTE để xử lý dữ liệu không cân bằng, sử dụng GridSearch CV để tìm tham số tối ưu, áp dụng cross-validation để đánh giá mô hình một cách đáng tin cậy hơn,...

## Chương 7: Các tranh luận / khám phá / kết luận, và các đề cử cho việc tiếp tục phát triển và cải tiến trong tương lai

### 7.1. Tranh luận / Khám phá / Kết luận chính

* Tóm tắt lại những phát hiện quan trọng nhất từ quá trình thực hiện đồ án.
* Những hiểu biết mới mà nhóm đã thu được về bài toán phân loại mail rác và các thuật toán học máy.
* Kết luận về hiệu quả của các phương pháp đã sử dụng.
* Trả lời các câu hỏi nghiên cứu hoặc mục tiêu đã đặt ra ở đầu đồ án.

### 7.2. Đóng góp của đồ án

* Nêu bật những đóng góp chính của đồ án (ví dụ: so sánh một cách hệ thống các thuật toán trên một tập dữ liệu cụ thể, đề xuất một quy trình tiền xử lý hiệu quả,...).

### 7.3. Hạn chế của đồ án

* Thẳng thắn nhìn nhận những hạn chế còn tồn tại của nghiên cứu/hệ thống.
  + Ví dụ: Tập dữ liệu có thể chưa đủ lớn hoặc đa dạng, chỉ tập trung vào nội dung text mà bỏ qua các đặc trưng khác (hình ảnh, metadata), mô hình chưa được triển khai thực tế,...

### 7.4. Các đề cử cho việc tiếp tục phát triển và cải tiến trong tương lai

* Đề xuất các hướng nghiên cứu hoặc cải tiến có thể thực hiện trong tương lai:
  + Sử dụng các thuật toán học sâu (Deep Learning) như CNN, RNN, LSTM, Transformers (BERT).
  + Thử nghiệm với các kỹ thuật trích xuất đặc trưng tiên tiến hơn (ví dụ: word embeddings, sentence embeddings).
  + Mở rộng tập dữ liệu, xử lý dữ liệu đa ngôn ngữ.
  + Xây dựng hệ thống có khả năng học trực tuyến (online learning) để thích ứng với các loại thư rác mới.
  + Phân tích các đặc trưng khác của email (header, sender information, HTML structure).

## Tài liệu tham khảo

*(Liệt kê tất cả các sách, bài báo khoa học, trang web, tài liệu đã tham khảo theo một chuẩn trích dẫn nhất quán, ví dụ: IEEE, APA)*

## Phụ lục (nếu có)

* *(Có thể bao gồm mã nguồn chính, các kết quả chi tiết không tiện trình bày trong thân bài, hoặc các tài liệu bổ sung khác)*