**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**- - - 🙞 🕮** **🙜 - - -**

Icon

Description automatically generated

**BÁO CÁO**

MÔN: **NHẬP MÔN KHOA HỌC DỮ LIỆU**

CÂU 5: **Trình bày và xử lý bài toán phân lớp theo Naive Bayes dựa vào học máy bằng ngôn ngữ Pyhon**

**Giảng viên hướng dẫn: TS. Tân Hạnh**

**Sinh viên thực hiện:**

Nguyễn Trần Đức Thuận – N19DCCN203

**TP. HCM, 20/06/2023**

**THUẬT TOÁN NAIVE BAYES**

## **Lý thuyết:**

Thuật toán Naive Bayes là một thuật toán học máy được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên và phân loại văn bản. Được đặt tên theo tên của nhà toán học Thomas Bayes, thuật toán Naive Bayes dựa trên nguyên tắc xác suất Bayes để đưa ra dự đoán.

Ý tưởng cơ bản của thuật toán Naive Bayes là dựa vào các tính năng hoặc thuộc tính của dữ liệu để phân loại các mẫu vào các nhãn khác nhau. Thuật toán giả định rằng các tính năng là độc lập với nhau (tức là "ngây thơ" - "naive"), điều này giúp đơn giản hóa quá trình tính toán.

Cách thức hoạt động của thuật toán Naive Bayes như sau:

Thu thập và chuẩn bị dữ liệu huấn luyện, trong đó mỗi mẫu được gán nhãn.

Tính toán xác suất của từng lớp dựa trên tần suất xuất hiện của các nhãn trong dữ liệu huấn luyện.

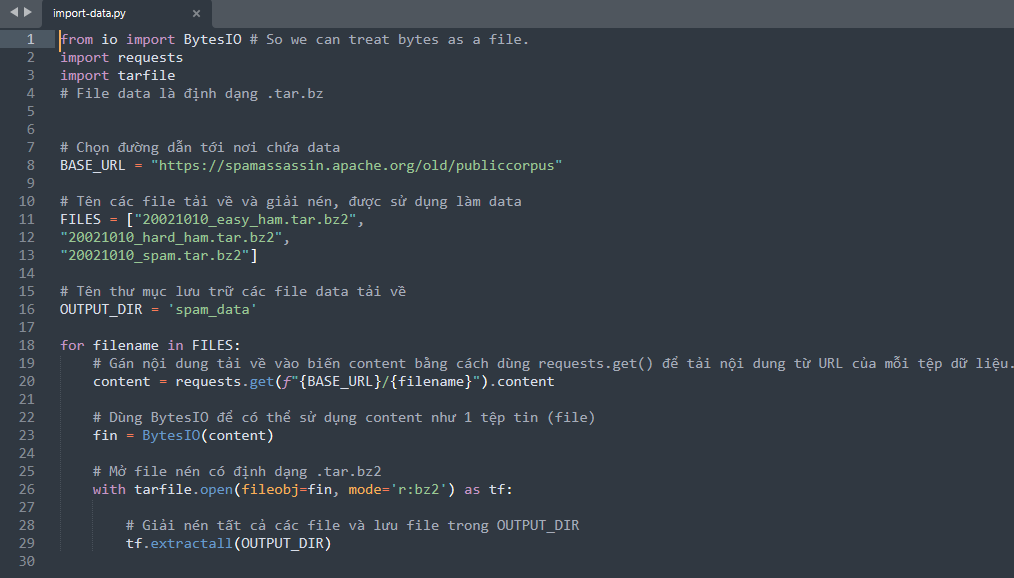
Tính toán xác suất của từng tính năng trong mỗi lớp dựa trên tần suất xuất hiện của tính năng trong từng lớp.

Đưa ra dự đoán bằng cách sử dụng công thức xác suất Bayes và giả định ngây thơ để tính toán xác suất của mỗi lớp cho mẫu dữ liệu mới.

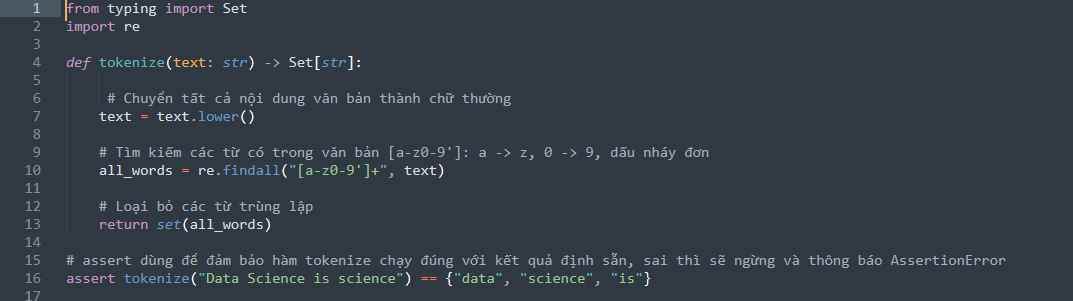
Chọn nhãn có xác suất cao nhất là kết quả phân loại của mẫu dữ liệu mới.

Mặc dù thuật toán Naive Bayes có giả định ngây thơ và một số hạn chế trong xử lý các tính năng phụ thuộc lẫn nhau, nhưng nó vẫn là một thuật toán phân loại mạnh mẽ và được sử dụng rộng rãi trong nhiều ứng dụng thực tế như phân loại email (spam/ham), phân loại tin tức, phân loại tài liệu và nhiều hơn nữa.ữ liệu hoặc các phương pháp đánh giá hiệu suất để chọn giá trị tối ưu cho k.

## **Giải thích code:**

Import-data.py:

naive\_bayes.py:

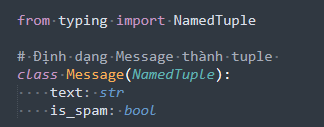


Đây là một hàm tokenize để chuyển đổi một chuỗi văn bản thành tập hợp các từ duy nhất.

Hàm này nhận đầu vào là một chuỗi văn bản (text) và trả về một tập hợp các từ xuất hiện trong văn bản đó. Các bước thực hiện của hàm này như sau:

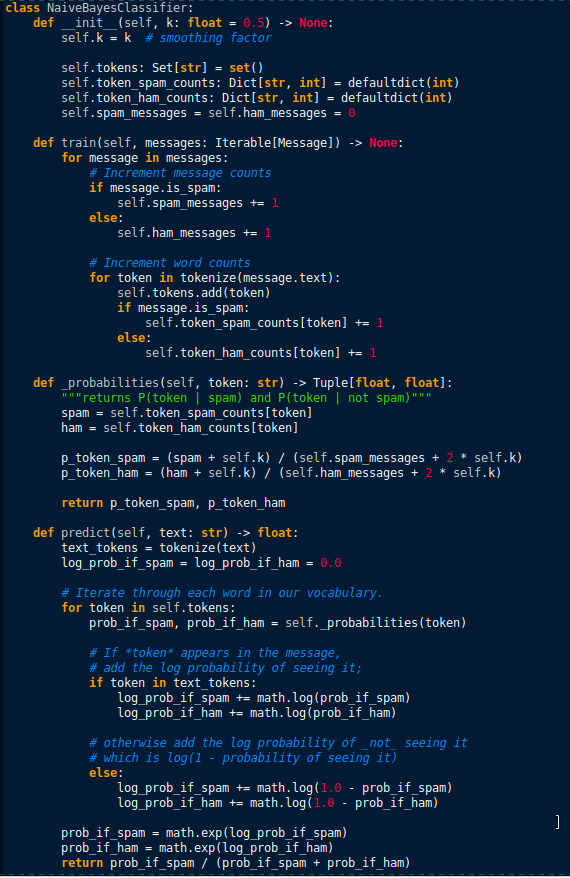
* Chuyển đổi chuỗi văn bản thành chữ thường (lowercase).
* Sử dụng biểu thức chính quy (regular expression) để tìm kiếm các từ trong văn bản. Biểu thức chính quy này sẽ chỉ chấp nhận các từ bao gồm chữ cái (a-z), chữ số (0-9) và dấu nháy đơn (') và loại bỏ các ký tự khác.
* Trả về tập hợp các từ duy nhất xuất hiện trong văn bản.

Ví dụ, khi gọi hàm tokenize("Data Science is science"), kết quả trả về là {"data", "science", "is"}.



Đâylà một lớp Message được định nghĩa sử dụng NamedTuple của Python. Lớp này có hai thuộc tính là text và is\_spam.

Lớp Message được sử dụng để biểu diễn một tin nhắn trong tập dữ liệu huấn luyện hoặc kiểm tra cho mô hình phân loại Naive Bayes. Thuộc tính text chứa nội dung của tin nhắn, còn thuộc tính is\_spam chứa thông tin về loại tin nhắn đó có phải là thư rác hay không. Đây là một cách tiện lợi để lưu trữ và xử lý dữ liệu trong Python, đặc biệt là khi có nhiều thuộc tính với giá trị cố định.



Đây là một đoạn mã Python dùng để xây dựng một mô hình phân loại Naive Bayes có thể được sử dụng để phát hiện thư rác (spam).

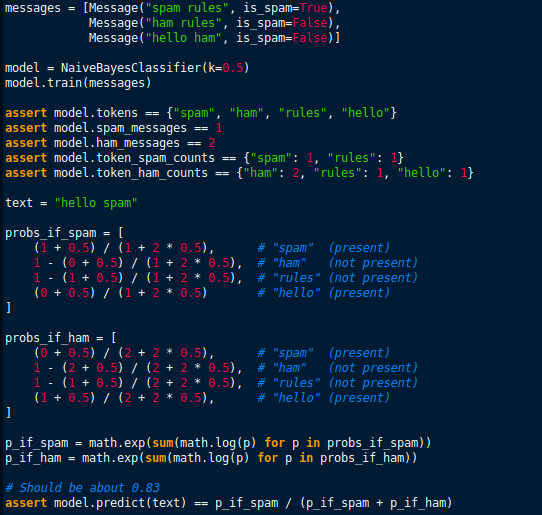
Lớp NaiveBayesClassifier này có các thuộc tính sau:

* k: hệ số smoothing (điều chỉnh độ tin cậy của mô hình)
* tokens: tập hợp các từ xuất hiện trong tập dữ liệu huấn luyện
* token\_spam\_counts: số lần xuất hiện của từ trong các thư rác
* token\_ham\_counts: số lần xuất hiện của từ trong các thư không phải rác
* spam\_messages: số thư rác trong tập dữ liệu huấn luyện
* ham\_messages: số thư không phải rác trong tập dữ liệu huấn luyện

Phương thức train được sử dụng để huấn luyện mô hình, với đầu vào là một tập các tin nhắn (messages). Mỗi tin nhắn có một thuộc tính is\_spam để xác định liệu tin nhắn đó có phải là thư rác hay không. Hàm train sẽ đếm số lần xuất hiện của các từ trong các tin nhắn và cập nhật các thuộc tính của lớp NaiveBayesClassifier.

Phương thức \_probabilities được sử dụng để tính xác suất của một từ trong thư rác và trong thư không phải rác.

Phương thức predict được sử dụng để dự đoán xác suất của một tin nhắn mới là thư rác. Hàm này sử dụng xác suất của từng từ trong tin nhắn để tính xác suất tổng thể của tin nhắn là thư rác.



Đây là một đoạn mã kiểm tra để đảm bảo rằng lớp NaiveBayesClassifier và hàm tokenize được triển khai đúng.

Trong đoạn mã này, tạo ra một danh sách messages gồm 3 tin nhắn, trong đó có 1 tin nhắn là thư rác ("spam rules"). Sau đó, một mô hình Naive Bayes được khởi tạo với k=0.5 và được huấn luyện trên danh sách messages.

Sau khi huấn luyện, đoạn mã kiểm tra các thuộc tính của lớp NaiveBayesClassifier và xác định xem liệu chúng có đúng như mong đợi hay không.

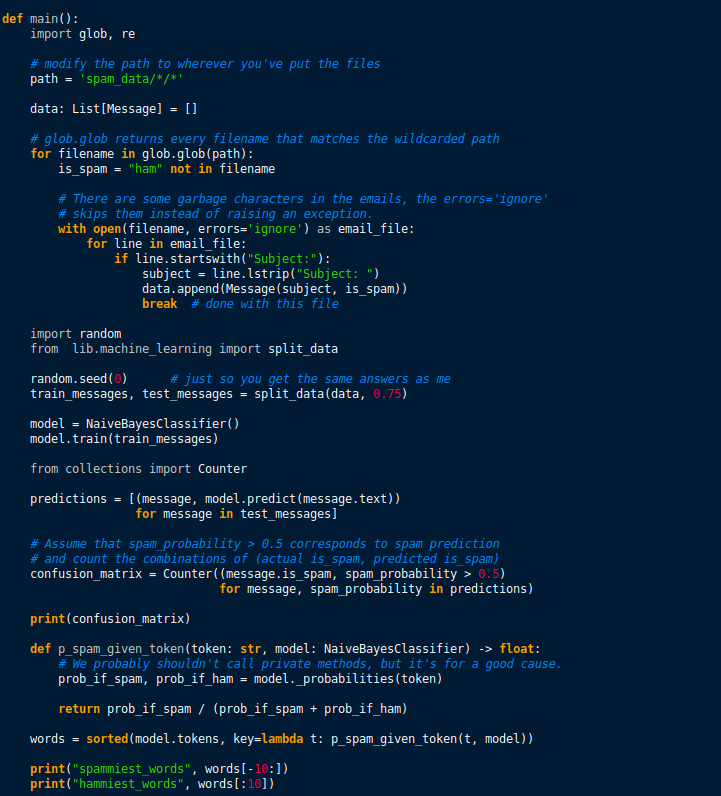
Tiếp theo, đoạn mã tạo ra một tin nhắn mới "hello spam" và tính toán xác suất của tin nhắn đó là thư rác. Kết quả được so sánh với giá trị mong đợi và đảm bảo rằng mô hình phân loại Naive Bayes hoạt động đúng.



Đây là một hàm drop\_final\_s trong Python được sử dụng để loại bỏ ký tự "s" cuối cùng của một từ.

Hàm này sử dụng biểu thức chính quy (regular expression) để tìm kiếm chuỗi "s$" trong từ đầu vào (word). Chuỗi này chỉ định rằng "s" phải xuất hiện ở cuối từ. Nếu đúng như vậy, hàm sẽ thay thế "s" cuối cùng bằng "" (tức là không có gì).

Lý do cho việc loại bỏ ký tự "s" cuối cùng có thể là để chuẩn hóa dữ liệu đầu vào. Ví dụ, trong tiếng Anh, nhiều từ trong số đó có thể có hình thức số ít hoặc số nhiều và việc loại bỏ "s" cuối cùng của từ có thể giúp đưa các từ này về dạng cơ bản hơn và giúp tăng độ chính xác của mô hình phân loại.



Đây là một hàm main được triển khai bằng Python để chạy chương trình phân loại thư rác bằng mô hình Naive Bayes.

Trong hàm này, tất cả các tệp tin trong thư mục "spam\_data" được đọc vào và chuyển đổi thành các đối tượng Message, được lưu trữ trong danh sách data. Mỗi đối tượng Message đại diện cho một tin nhắn trong tệp tin, với thuộc tính is\_spam được xác định dựa trên tên tệp tin.

Dữ liệu trong danh sách data được chia thành hai tập train\_messages và test\_messages, và mô hình Naive Bayes được huấn luyện trên tập train\_messages.

Sau đó, mô hình được sử dụng để dự đoán xác suất của các tin nhắn trong tập test\_messages. Kết quả được tính toán dựa trên ngưỡng 0,5 và được sử dụng để tạo ma trận nhầm lẫn (confusion matrix) đánh giá hiệu suất của mô hình.

Cuối cùng, hàm main hiển thị các từ liên quan đến thư rác nhất và các từ liên quan đến thư thường nhất.