

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

VIỆN TRÍ TUỆ NHÂN TẠO



BÁO CÁO

KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN

ĐỀ TÀI:

**THUẬT TOÁN NAÏVE BAYES & LẬP TRÌNH MAP-REDUCE
TRONG PHÂN LỚP DỮ LIỆU SỬ DỤNG MONGODB**

Mã môn học	INT3229 38	
Sinh viên thực hiện	Nguyễn Quang Thao	22022619
	Vũ Thành Đạt	22022620
	Nguyễn Trần Hải Ninh	22022526
Giảng viên hướng dẫn	TS. Trần Hồng Việt	
	CN. Lương Sơn Bá	
	ThS. Ngô Minh Hương	

Hà Nội - 2024

LỜI MỞ ĐẦU

Trong thời gian học tập và hoàn thành bài tập lớn môn Kỹ thuật và công nghệ dữ liệu lớn, chúng em xin được chân thành cảm ơn sự giúp đỡ nhiệt tình của thầy cô đã giúp đỡ, cung cấp nhiều thông tin và kiến thức quý báu tạo điều kiện giúp chúng em hoàn thành tốt bài tập lớn này.

Chúng em cũng xin được gửi lời cảm ơn tới các các giả đã chia sẻ các kiến thức cần có hỗ trợ trong dự án này.

Bản báo cáo này phần lớn đã đạt đủ yêu cầu nội dung cần thiết, tuy nhiên vì vấn đề về thời gian, kiến thức nên báo cáo có thể còn sai sót. Rất mong nhận được sự thông cảm và đóng góp ý kiến từ thầy cô.

Xin chân thành cảm ơn!

MỤC LỤC

1. TỔNG QUAN VỀ DỮ LIỆU LỚN.....	5
1.1 Khái niệm.....	5
1.2 Đặc trưng của dữ liệu lớn	5
1.3 MongoDB	6
1.4 Map-Reduce trong MongoDB	6
2. THUẬT TOÁN NAÏVE BAYES.....	7
2.1 Giới thiệu	7
2.2 Nguyên lý hoạt động.....	7
2.2.1 Công thức xác suất Bayes được biểu diễn như sau:	7
2.2.2 Giả định độc lập:.....	7
2.2.3 Quy trình phân loại	7
2.3 Ứng dụng	8
2.4 Ví dụ minh họa	8
3. ÁP DỤNG MAP-REDUCE TRONG THUẬT TOÁN NAÏVE BAYES	9
3.1 Ý tưởng map-reduce	9
3.2 Lưu đồ thuật toán.....	10
3.3 Triển khai trên MongoDB	10
3.3.1 Chuẩn bị dữ liệu.....	10
3.3.2 Xây dựng bộ dữ liệu.....	11
3.3.3 Thực hiện dự đoán	12
3.3.4 Kết quả	13
4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	15
4.1 Kết luận.....	15
4.2 Hướng phát triển.....	16
TÀI LIỆU THAM KHẢO.....	17

DANH MỤC HÌNH ẢNH

Hình ảnh 1: Dữ liệu có veracity cao và có thể phân tích nhanh có giá trị hơn	5
Hình ảnh 2: Quá trình thực hiện map-reduce trong MongoDB	6
Hình ảnh 3: Lưu đồ thuật toán trên bài toán phân loại email spam	10
Hình ảnh 4: Quá trình insert dữ liệu vào collection train.....	11
Hình ảnh 5: Pipeline đếm số lượng từ trong từng loại email	12
Hình ảnh 6: Pipeline đếm tần suất xuất hiện của mỗi từ trong từng loại email	12
Hình ảnh 7: Collection TotalCounts.....	13
Hình ảnh 8: Collection WordCounts	14
Hình ảnh 9: Confusion matrix đối với tập test	15

1. TỔNG QUAN VỀ DỮ LIỆU LỚN

1.1 Khái niệm

Theo Wikipedia: Dữ liệu lớn là một thuật ngữ cho việc xử lý một tập dữ liệu phức tạp với kích thước vượt xa khả năng quản lý và xử lý dữ liệu trong một thời gian có thể chấp nhận được. Dữ liệu lớn yêu cầu một tập các kỹ thuật và công nghệ được tích hợp theo hình thức mới để khai phá từ tập dữ liệu đa dạng, phức tạp, và có quy mô lớn.

Theo sách Big Data Fundamentals_ Concepts, Drivers & Techniques: Dữ liệu lớn là một lĩnh vực chuyên về phân tích, xử lý và lưu trữ lượng lớn các tập dữ liệu liên tục đến từ nhiều nguồn khác nhau. Các giải pháp và phương pháp xử lý dữ liệu lớn thường được sử dụng khi mà các công cụ xử lý dữ liệu truyền thống trong việc phân tích, xử lý và lưu trữ không thể đáp ứng. Cụ thể, dữ liệu lớn xử lý các yêu cầu đặc thù như kết hợp các tập dữ liệu không liên quan, hay xử lý lượng lớn giá trị không có cấu trúc và khai thác các đặc trưng của dữ liệu nhanh chóng.

1.2 Đặc trưng của dữ liệu lớn

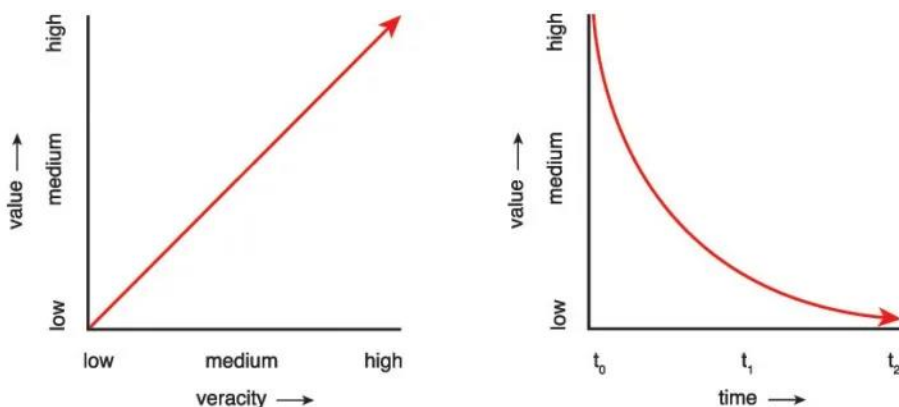
Volume: Khối lượng dữ liệu ngày càng tăng dẫn đến nhu cầu quản lý và lưu trữ cũng ngày càng tăng mạnh. Hầu hết dữ liệu đến từ các nguồn như các giao dịch online, mua bán, các nghiên cứu, dữ liệu cảm biến (nhiệt độ, GPS...) hay mạng xã hội,...

Velocity: Các hệ thống xử lý dữ liệu cần có tốc độ xử lý dữ liệu nhanh để có thể đáp ứng được nhu cầu của doanh nghiệp trong thời đại dữ liệu được sinh ra liên tục. Tuy nhiên ở các lĩnh vực khác nhau lại có tốc độ sinh dữ liệu mới khác nhau.

Variety: Dữ liệu không chỉ có mỗi dạng text mà còn các loại dữ liệu khác như dữ liệu hình ảnh, âm thanh, dữ liệu có cấu trúc hay dữ liệu không có cấu trúc,...

Veracity: Dữ liệu có độ nhiễu càng cao thì càng mang lại ít giá trị.

Value: Giá trị của dữ liệu chính là điều các doanh nghiệp hướng tới. Tuy nhiên dữ liệu càng mất nhiều thời gian để chuyển thành thông tin ý nghĩa thì lại càng mất giá trị với doanh nghiệp.



Hình ảnh 1: Dữ liệu có veracity cao và có thể phân tích nhanh có giá trị hơn

1.3 MongoDB

Theo Wikipedia: MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở đa nền tảng viết bằng C++. Bản ghi trong MongoDB được lưu trữ dạng một dữ liệu văn bản (Document), là một cấu trúc dữ liệu bao gồm các cặp giá trị và trường tương tự như các đối tượng JSON.

Đặc điểm:

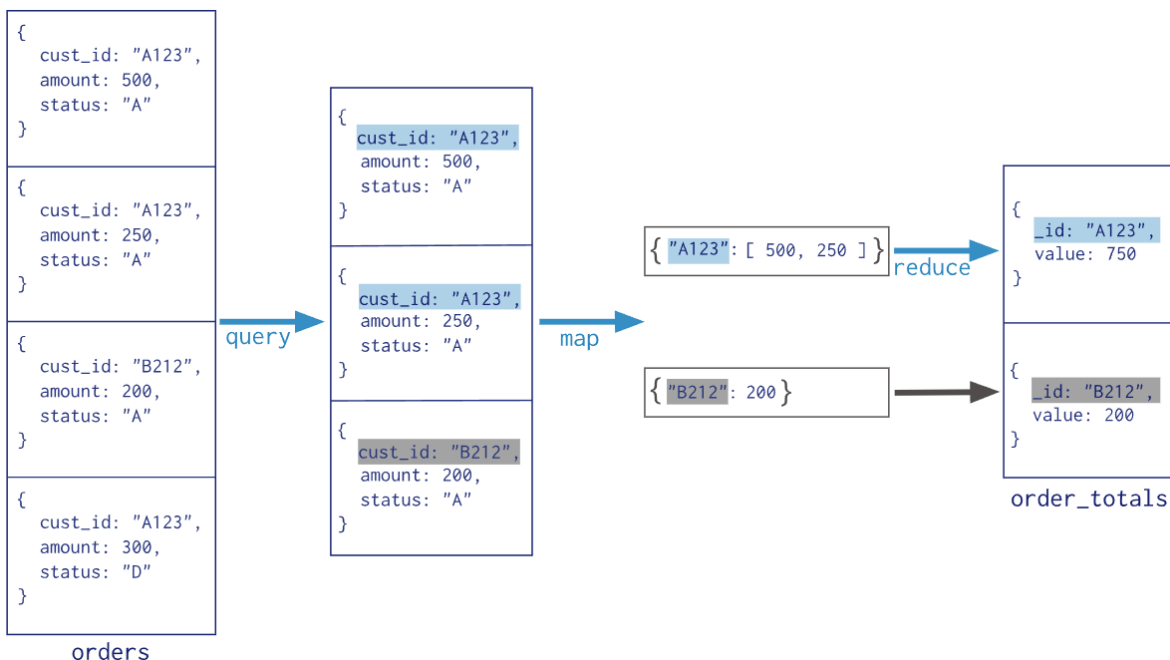
- Lưu trữ định hướng Document, dữ liệu lưu dưới dạng kiểu JSON
- Lập chỉ mục trên bất kỳ thuộc tính nào
- Truy vấn đa dạng
- Cập nhật nhanh

Lợi thế so với CSDL dạng quan hệ (RDBMS):

- Ít Schema hơn: số trường, nội dung và kích cỡ của Document này có thể khác Document khác
- Cấu trúc của một đối tượng không rõ ràng
- Không có các Join phức tạp
- Khả năng truy vấn sâu hơn vì hỗ trợ các truy vấn động trên các Document
- Dễ dàng mở rộng
- Sử dụng bộ nhớ nội tại để lưu trữ phần công việc giúp truy cập dữ liệu nhanh hơn

1.4 Map-Reduce trong MongoDB

Map-reduce là một mô hình xử lý dữ liệu để tính toán lượng lớn dữ liệu thành các kết quả tổng hợp hữu ích. MongoDB đã cung cấp giải pháp thực hiện map-reduce.



Hình ảnh 2: Quá trình thực hiện map-reduce trong MongoDB

Quá trình thực hiện như sau:

- Áp dụng hàm map cho từng document và trả về các cặp key-value
- Thực hiện reduce tổng hợp giá trị lại đối với những document cùng key với nhau

2. THUẬT TOÁN NAÏVE BAYES

2.1 Giới thiệu

Naïve Bayes là một thuật toán học máy thuộc nhóm phân loại (classification). Thuật toán này dựa trên Định lý Bayes trong lý thuyết xác suất và giả định mạnh mẽ rằng các đặc trưng (features) là độc lập với nhau, điều này khiến nó được gọi là "Naïve" (ngây thơ).

2.2 Nguyên lý hoạt động

2.2.1 Công thức xác suất Bayes được biểu diễn như sau:

$$P(C | X) = \frac{P(C) \cdot P(X | C)}{P(X)}$$

Trong đó:

- $P(C|X)$ là xác suất của lớp C khi biết các đặc trưng X (Posterior probability)
- $P(C)$ là xác suất tiên nghiệm của lớp C (Prior probability)
- $P(X|C)$ là xác suất của các đặc trưng X khi biết lớp C (likelihood)
- $P(X)$ là xác suất tiên nghiệm của các đặc trưng X (Evidence)

2.2.2 Giả định độc lập:

- Naive Bayes giả định rằng các đặc trưng X_1, X_2, \dots, X_n là độc lập. Điều này đơn giản hóa bài toán bằng cách:

$$P(X | C) = P(X_1 | C) \cdot P(X_2 | C) \dots P(X_n | C)$$

- Từ đó, công thức tính xác suất của lớp C trở thành:

$$P(C | X) \propto P(C) \cdot \prod_{i=1}^n P(X_i | C)$$

2.2.3 Quy trình phân loại

- Chuẩn bị dữ liệu
 - o Tập dữ liệu huấn luyện được chuẩn bị, trong đó mỗi phần tử dữ liệu được biểu diễn dưới dạng vector chứa nhiều giá trị thuộc tính.
- Tính toán xác suất tiên nghiệm $P(C)$
 - o Xác suất của mỗi lớp C dựa trên tần suất xuất hiện của chúng trong dữ liệu huấn luyện
 - o Xác suất tiên nghiệm $P(C)$ cho mỗi lớp C được tính bằng cách chia số lượng mẫu thuộc lớp đó cho tổng số mẫu trong tập huấn luyện
- Tính toán xác suất Likelihood
 - o Tiếp theo, thuật toán sẽ tính xác suất likelihood $P(X|C)$ cho mỗi đặc trưng trong vector đầu vào X khi biết lớp C . Tùy vào loại dữ liệu, có các cách tính như:

- Đối với biến rời rạc: Sử dụng tần suất xuất hiện của giá trị của đặc trưng trong lớp
- Đối với biến liên tục: Sử dụng phân phối Gaussian để ước lượng xác suất
- Tính toán xác suất hậu nghiệm
 - Sau khi đã có các xác suất tiên nghiệm và likelihood, thuật toán sẽ sử dụng định lý Bayes để tính xác suất hậu nghiệm cho mỗi lớp:

$$P(C | X) \propto P(C) \cdot P(X | C)$$

Trong đó: $P(X | C) = P(X_1 | C) \cdot P(X_2 | C) \dots P(X_n | C)$
 - Sử dụng log để cải thiện tránh vanishing được công thức mới

$$\log P(X | C) = \log[P(X_1 | C) + P(X_2 | C) + \dots + P(X_n | C)]$$
- Gắn nhãn: Gắn nhãn cho dữ liệu đầu vào lớp X với lớp có xác suất hậu nghiệm lớn nhất

2.3 Ứng dụng

- Phân loại văn bản: Phân loại email là thư rác hay không phải thư rác
- Phân tích cảm xúc: Dự đoán cảm xúc trong văn bản dựa trên từ ngữ sử dụng
- Hệ thống gợi ý: Đưa ra gợi ý cho người dùng dựa trên hành vi của họ
- Dự đoán theo thời gian thực: Do tốc độ xử lý nhanh, Naïve Bayes thường được sử dụng trong các ứng dụng yêu cầu phản hồi ngay lập tức như hệ thống cảnh báo

2.4 Ví dụ minh họa

Bảng dữ liệu:

Outlook	Play
Rainy	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Sunny	No
Rainy	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Sunny	No
Sunny	Yes

Rainy	No
Overcast	Yes
Overcast	Yes

Bảng tần suất:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	4

Likelihood table

Weather	Yes	No	
Overcast	5	0	5/14=0.35
Rainy	2	2	4/14=0.3
Sunny	3	2	5/14=0.35
All	10/14=0.7	4/14=0.3	

Ví dụ tính xác suất bằng Bayes: giả sử ta muốn dự đoán liệu một người có thể chơi thể thao vào ngày nắng hay không

$$P(Yes | Sunny) = \frac{P(Sunny | Yes) \cdot P(Yes)}{P(Sunny)} = \frac{0.3 \times 0.7}{0.35} = 0.60$$

$$P(No | Sunny) = \frac{P(Sunny | No) \cdot P(No)}{P(Sunny)} = \frac{0.5 \times 0.3}{0.30} = 0.40$$

⇒ Vì $P(Yes|Sunny) > P(No|Sunny)$ nên đưa ra kết luận là người chơi có thể chơi thể thao vào ngày nắng.

3. ÁP DỤNG MAP-REDUCE TRONG THUẬT TOÁN NAÏVE BAYES

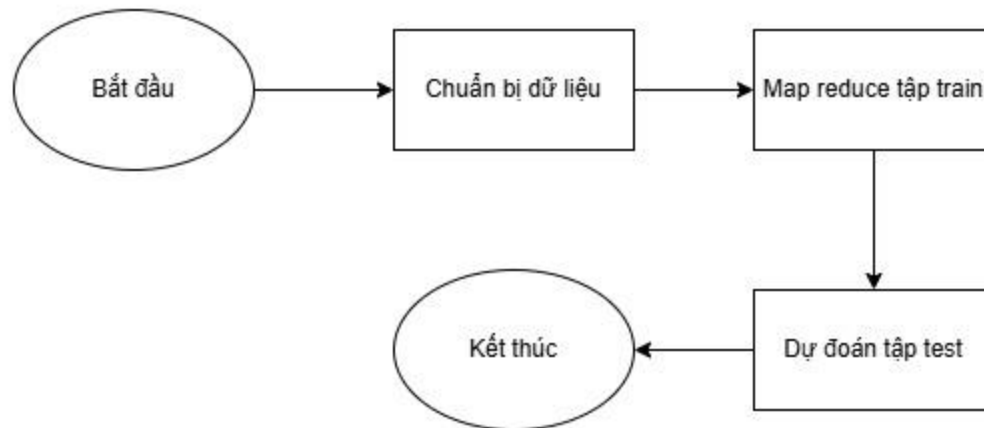
3.1 Ý tưởng map-reduce

Nhiệm vụ: Đếm số lần xuất hiện của một từ trong email thường và email spam.

Ý tưởng:

- Map: Xử lý trên từng document trong collection, chia ra thành các cặp key-value
- Reduce: Tổng hợp các giá trị có cùng key

3.2 Lưu đồ xử lý

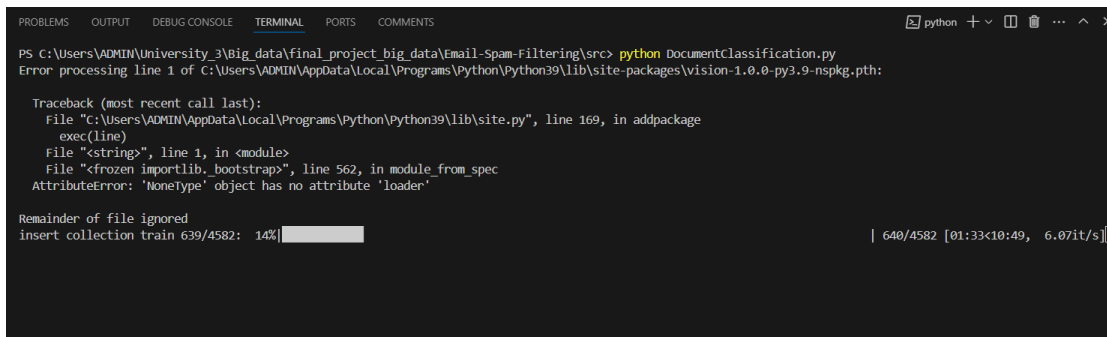


Hình ảnh 3: Lưu đồ thuật toán trên bài toán phân loại email spam

3.3 Triển khai trên MongoDB

3.3.1 Chuẩn bị dữ liệu

- Sử dụng bộ dữ liệu từ [nguồn Kaggle](#) là file csv với 5695 dữ liệu khác nhau. File có 2 cột gồm: “text” – dữ liệu email; “spam” – trạng thái email. Trong đó có 4360 email thường và 1335 email spam
- Chia bộ dữ liệu thành hai tập train và test với tỷ lệ 80:20
- Tạo collection train:
 - Đầu vào: Mảng các dictionary gồm nội dung email và giá trị spam
 - Xử lý:
 - Lần lượt lặp qua các phần tử
 - Thực hiện loại bỏ stop words, non-words và thực hiện lemmatization
 - Thêm dữ liệu vào collection train với mỗi document có cấu trúc: $\{“content”: str[], “classX”: 0|1, “classY”: 0|1\}$
 - Kết quả: Collection train chứa các document có cấu trúc: $\{“content”: str[], “classX”: 0|1, “classY”: 0|1\}$



```
PS C:\Users\ADMIN\University_3\Big_data\final_project_big_data\Email-Spam-Filtering\src> python DocumentClassification.py
Error processing line 1 of C:\Users\ADMIN\AppData\Local\Programs\Python\Python39\lib\site-packages\vision-1.0.0-py3.9-nspkg.pth:

Traceback (most recent call last):
  File "C:\Users\ADMIN\AppData\Local\Programs\Python\Python39\lib\site.py", line 169, in addpackage
    exec(line)
  File "<string>", line 1, in <module>
  File "<frozen importlib._bootstrap>", line 562, in module_from_spec
AttributeError: 'NoneType' object has no attribute 'loader'

Remainder of file ignored
insert collection train 639/4582: 14% | 640/4582 [01:33<10:49, 6.07it/s]
```

Hình ảnh 4: Quá trình insert dữ liệu vào collection train

- Tạo collection test:
 - Đầu vào: Mảng các dictionary gồm nội dung email và giá trị spam
 - Xử lý:
 - Lần lượt lặp qua các phần tử
 - Thực hiện loại bỏ stop words, non-words và thực hiện lemmatization
 - Thêm dữ liệu vào collection train với mỗi document có cấu trúc: `{"content": str[], "classX": 0|1, "classY": 0|1, "predclassX": 0, "predclassY": 0, processed: False}`
 - Kết quả: Collection train chứa các document có cấu trúc: `{"content": str[], "classX": 0|1, "classY": 0|1, "predclassX": 0, "predclassY": 0, processed: False}`

3.3.2 Xây dựng bộ dữ liệu

(Từ phiên bản MongoDB 5.0 trở đi không còn hỗ trợ trực tiếp hàm map-reduce, thay vào đó ta sử dụng phương pháp aggregation pipelines thay thế)

3.3.2.1 Tạo collection TotalCounts

Mục tiêu: Đếm số lượng từ thuộc mỗi loại email

Tạo pipeline để thực hiện thông qua hàm aggregate

```

1 pipeline_total_counts = [
2     {"$addFields": {"wordCount": {"$size": "$content"}}},
3     {
4         "$group": {
5             "_id": None,
6             "TotalclassX": {
7                 "$sum": {"$cond": [{"$eq": ["$classX", 1]}, "$wordCount", 0]}
8             },
9             "TotalclassY": {
10                 "$sum": {"$cond": [{"$eq": ["$classY", 1]}, "$wordCount", 0]}
11             },
12             "TotalWords": {"$sum": "$wordCount"},
13         }
14     },
15     {
16         "$project": {
17             "_id": 0,
18             "clX": "$TotalclassX",
19             "clY": "$TotalclassY",
20             "V": "$TotalWords",
21         }
22     },
23 ]

```

Hình ảnh 5: Pipeline đếm số lượng từ trong từng loại email

Kết quả: Số lượng từ trong từng loại email. Trong đó classX là email spam và classY là email thường.

3.3.2.2 Tạo collection WordCounts

Mục tiêu: Đếm tần suất xuất hiện của mỗi từ trong từng loại email

Tạo pipeline để thực hiện thông qua hàm aggregate

```

1 pipeline_counts_each_word = [
2     {"$unwind": "$content"},
3     {
4         "$project": {
5             "key": "$content",
6             "clX": "$classX",
7             "clY": "$classY",
8         }
9     },
10    {"$group": {"_id": "$key", "clX": {"$sum": "$clX"}, "clY": {"$sum":
11        "$clY"}}},
12 ]

```

Hình ảnh 6: Pipeline đếm tần suất xuất hiện của mỗi từ trong từng loại email

Kết quả: Collection WordCounts chứa các documents trong đó mỗi documents chứa thông tin tần suất xuất hiện của từ trong từng loại email.

3.3.3 Thực hiện dự đoán

- Tính toán xác suất tài liệu thuộc từng lớp dựa trên nội dung (các từ trong document)

- Sử dụng log của xác suất để tránh vấn đề tràn số
- Các bước thực hiện
 - o Truy xuất tần suất xuất hiện của từng từ trong collection WordCounts
 - o Tính xác suất với công thức (sử dụng add-one smoothing)

$$\log P(w | C) = \log \frac{\text{count}(w, C) + 1}{\sum \text{count}(C) + |V|}$$

- o Tổng hợp xác suất log của tất cả các từ trong tài liệu để tính tổng xác suất log cho mỗi lớp
- Kết quả: Trả về dự đoán: 1 (spam) nếu xác suất log của spam lớn hơn, ngược lại là 0 (ham)

3.3.4 Kết quả

email_spam_filtering.TotalCounts

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 47B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes

[Generate queries from natural language in Compass](#) **INSERT DOCUMENT**

[Filter](#) Type a query: { field: **Reset** **Apply** [Options ▶](#)

QUERY RESULTS: 1-1 OF 1

```

_id: ObjectId('674fc59bf561eec55f0ddc92')
c1X : 125960
c1Y : 507462
V : 633422
  
```

Hình ảnh 7: Collection TotalCounts

email_spam_filtering.WordCounts

STORAGE SIZE: 692KB

LOGICAL DATA SIZE:

TOTAL DOCUMENTS:

INDEXES TOTAL SIZE:

1.22MB

30346

972KB

Find

Indexes

Schema Anti-
Patterns 0

Aggregation

Search
Indexes

[Generate queries from natural language in Compass](#)

INSERT DOCUMENT

Filter

Type a query: { field:

Reset

Apply

Options

QUERY RESULTS: 1-20 OF MANY

```
_id: "coneway"  
claX: 0  
claY: 1
```

```
_id: "chartering"  
claX: 0  
claY: 3
```

```
_id: "diagramming"  
claX: 0  
claY: 1
```



PREVIOUS

1-20 of many results

NEXT



Hình ảnh 8: Collection WordCounts

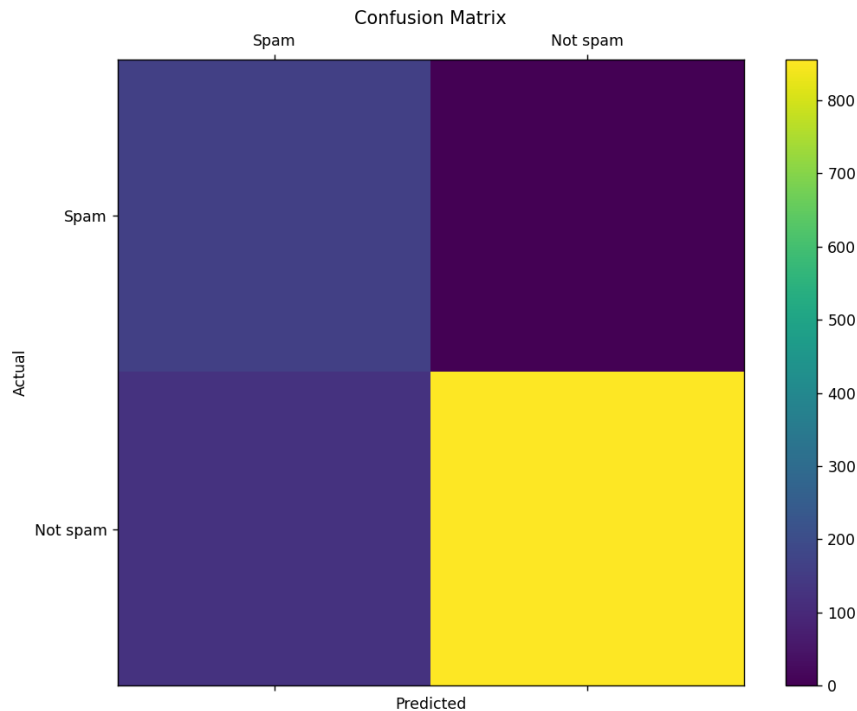
```

PS C:\Users\ADMIN\University_3\Big_data\final_project_big_data\Email-Spam-Filtering\src> python DocumentClassification.py
Error processing line 1 of C:\Users\ADMIN\AppData\Local\Programs\Python\Python39\lib\site-packages\vision-1.0.0-py3.9-nspkg.pth:

Traceback (most recent call last):
  File "C:\Users\ADMIN\AppData\Local\Programs\Python\Python39\lib\site.py", line 169, in addpackage
    exec(line)
  File "<string>", line 1, in <module>
  File "<frozen importlib._bootstrap>", line 562, in module_from_spec
AttributeError: 'NoneType' object has no attribute 'loader'

Remainder of file ignored
Data collection already created
Mapreduce already done
Mapreduce already done
Naive Bayes Classifier for mini test started
This is a test email : Not Spam
money you got from me : Spam
{'TP': 163, 'FP': 0, 'TN': 856, 'FN': 127, 'Precision': 1.0, 'Recall': 0.5620689655172414, 'F1-score': 0.7196467991169978}

```



Hình ảnh 9: Confusion matrix đối với tập test

4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Dữ liệu lớn đã đặt ra nhiều thách thức cho các tổ chức, doanh nghiệp và các cá nhân đồng thời tạo ra nhiều lợi ích và tài sản quý giá. Mô hình Map-reduce phân chia công việc và thực hiện tính toán để đạt được kết quả cuối nhanh hơn là một bước tiến mới. Và cơ sở dữ liệu NoSQL MongoDB cũng đem lại thêm những giải pháp hữu ích cho cộng đồng trong quá trình xử lý và tính toán trong lĩnh vực dữ liệu lớn.

Báo cáo phân tích quá trình thực hiện map-reduce thực hiện thuật toán học máy Naïve Bayes để phân loại email spam. Tuy nhiên nhóm mới chỉ thực hiện trên bộ dữ liệu nhỏ để demo cho đề tài.

4.2 Hướng phát triển

Áp dụng thêm các kiến thức hoặc công nghệ khác như Hadoop, Spark... để có thể cải tiến và xây dựng hệ thống phân loại với dữ liệu lớn hơn hoặc trong lĩnh vực khác.

Cải thiện thuật toán hoặc tăng cường bộ dữ liệu để có thể phân loại đạt độ chính xác cao hơn.

Trong quá trình hoàn thành bài tập lớn, do thời gian có hạn nên việc tìm hiểu và tham khảo các tài liệu bên ngoài vẫn còn hạn chế và không tránh khỏi những thiếu sót. Rất mong nhận được đánh giá, đóng góp của thầy cô để chúng em có thể hoàn thiện kỹ năng của bản thân mình.

TÀI LIỆU THAM KHẢO

- [1] Big Data Fundamentals Concepts, Drivers & Techniques
- [2] [Wikipedia: Dữ liệu lớn](#)
- [3] [Wikipedia: MongoDB](#)
- [4] [Viblo: Tổng quan về MongoDB](#)
- [5] [MongoDB documentation](#)
- [6] [Github: MariaSL/Document-Classification](#)

NHIỆM VỤ CỦA CÁC THÀNH VIÊN

Họ và tên	Công việc
Nguyễn Quang Thao	<ul style="list-style-type: none">+ Làm báo cáo+ Tìm hiểu MongoDB+ Ý tưởng map-reduce với MongoDB+ Cài đặt giao diện demo
Vũ Thành Đạt	<ul style="list-style-type: none">+ Làm slide+ Cài đặt chương trình demo và đánh giá+ Tìm hiểu tổng quan về dữ liệu lớn
Nguyễn Trần Hải Ninh	<ul style="list-style-type: none">+ Tìm hiểu tổng quan về Naïve Bayes+ Ví dụ minh họa Naïve Bayes+ Làm slide+ Cài đặt chương trình demo