

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
KHOA KHOA HỌC MÁY TÍNH**



**ĐỒ ÁN MÔN HỌC
NGUYÊN LÝ MÁY HỌC**

**ĐỀ TÀI
ĐÁNH GIÁ CÁC MÔ HÌNH PHÂN LỚP
DỰA TRÊN TẬP DỮ LIỆU VĂN BẢN
VÀ HÌNH ẢNH**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
KHOA KHOA HỌC MÁY TÍNH**



**ĐỒ ÁN MÔN HỌC
NGUYÊN LÝ MÁY HỌC**

**ĐỀ TÀI
ĐÁNH GIÁ CÁC MÔ HÌNH PHÂN LỚP
DỰA TRÊN TẬP DỮ LIỆU VĂN BẢN
VÀ HÌNH ẢNH**

Giảng viên hướng dẫn:

TS. Phạm Thế Phi

Sinh viên thực hiện:

Lê Tuấn Đạt B2113328

Cần Thơ, 04/2024

NHẬN XÉT CỦA GIÁO VIÊN

[illegible]

LỜI CẢM ƠN

Để có được bài báo cáo này, em xin được bày tỏ lòng biết ơn chân thành và sâu sắc đến thầy Phạm Thế Phi – người đã trực tiếp tận tình hướng dẫn, giúp đỡ em. Trong suốt quá trình thực hiện bài báo cáo, nhờ những sự chỉ bảo và hướng dẫn quý giá đó mà bài báo cáo được hoàn thành một cách tốt nhất.

Em cũng xin gửi lời cảm ơn chân thành đến các Thầy Cô Giảng viên Đại học Cần Thơ, đặc biệt là các Thầy Cô ở Khoa CNTT & TT, những người đã truyền đạt những kiến thức quý báu trong thời gian qua.

Em cũng xin chân thành cảm ơn bạn bè cùng với gia đình đã luôn động viên, khích lệ và tạo điều kiện giúp đỡ trong suốt quá trình thực hiện để em có thể hoàn thành bài báo cáo một cách tốt nhất.

Tuy có nhiều cố gắng trong quá trình thực hiện bài báo cáo, nhưng không thể tránh khỏi những sai sót. Em rất mong nhận được sự đóng góp ý kiến quý báu của quý Thầy Cô và các bạn để bài báo cáo hoàn thiện hơn.

Cần Thơ, ngày 17 tháng 04 năm 2024
Người viết

Lê Tuấn Đạt

MỤC LỤC

MỤC LỤC	1
DANH MỤC HÌNH	2
DANH MỤC BẢNG	3
PHẦN I: GIỚI THIỆU	4
I. Giới thiệu đề tài:	4
II. Giới thiệu tập dữ liệu:	4
III. Mô tả hệ thống:	5
IV. Mục tiêu đề tài:	5
V. Bố cục báo cáo:	5
PHẦN II: NỘI DUNG	6
I. Khái quát về các mô hình học máy:	6
II. Đánh giá các mô hình học máy dựa trên tập dữ liệu văn bản và hình ảnh:	8
1. Đối với tập dữ liệu văn bản:	8
2. Đối với tập dữ liệu hình ảnh:	9
3. Đối với tập dữ liệu kết hợp văn bản và hình ảnh:	10
PHẦN III. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	12
I. Kết luận:	12
II. Hướng phát triển:	12
TÀI LIỆU THAM KHẢO	13

DANH MỤC HÌNH

Hình 1. Biểu đồ thể hiện độ chính xác và thời gian xử lý các mô hình trên tập dữ liệu văn bản	8
Hình 2. Biểu đồ thể hiện độ chính xác và thời gian xử lý các mô hình trên tập dữ liệu hình ảnh	9

DANH MỤC BẢNG

Bảng 1. Bảng thể hiện số lượng các lớp trong thư mục Train và Test	4
Bảng 2. Kết quả các mô hình phân lớp khi kết hợp văn bản và hình ảnh	10

PHẦN I: GIỚI THIỆU

I. Giới thiệu đề tài:

Bài báo cáo chủ yếu đánh giá hiệu suất các mô hình phân lớp dựa trên tập dữ liệu đa phương tiện (văn bản + hình ảnh) bằng cách sử dụng các mô hình học máy cổ điển như K láng giềng gần nhất, Bayes thơ ngây, cây quyết định, rừng ngẫu nhiên, AdaBoost, SVM và mạng nơ-ron nhân tạo.

II. Giới thiệu tập dữ liệu:

Tập dữ liệu đa phương tiện (văn bản + hình ảnh) mô tả 02 lớp dữ liệu: giày và túi xách.

Tập dữ liệu văn bản được mô tả như sau:

“Zip opening into main compartment, zip pocket inside, 210D Nylon lining with PU backing, PU carry handle with shiny silver metal hardware, PU tape zip puller, Cat Logo on front, embroidered PUMA Formstrips on back side, embroidered Cat Logo side, accent ribbon. Available Colors: Black, Steel Grey, Cordovan.” - Đoạn văn mô tả class 10 của *túi xách*.

“Known for their edgy, modern designs suitable for every fashionable woman, Balenciaga proves that true style is never hard to achieve.” - Đoạn văn mô tả class 10 của *giày*.

Tập dữ liệu hình ảnh gồm các hình ảnh của các class:



Số lượng	Giày		Túi xách	
	Train	Test	Train	Test
Văn bản	18	20	18	20
Hình ảnh	18	20	18	20

Bảng 1. Bảng thể hiện số lượng các lớp trong thư mục Train và Test

III. Mô tả hệ thống:

Gói phần mềm **mlfw** sử dụng ngôn ngữ Python, bao gồm các modules:

- ✧ **Helpers:** Dùng để tiền xử lý dữ liệu văn bản hoặc hình ảnh, rút trích đặc trưng dữ liệu văn bản hoặc hình ảnh. Đặc trưng văn bản gồm tf-idf. Đặc trưng hình ảnh gồm: màu sắc (color), sift, hog, gist, deep (có thể chọn vgg16, vgg19, inception, xception, resnet).
- ✧ **Processors:** Dùng để chuẩn bị, nạp dữ liệu (Data), huấn luyện và thử nghiệm các mô hình máy học trên dữ liệu văn bản (Text) và hình ảnh (Image). Module Processors sử dụng các hàm tiền xử lý dữ liệu và rút trích đặc trưng từ module Helpers.
- ✧ **Ranker:** Dùng để gọi chạy các thành phần của module Processors.
- ✧ **Runner:** Gọi chạy Ranker, truyền 03 tham số: Đường dẫn đến thư mục chứa dữ liệu huấn luyện, thử nghiệm và số lớp.

IV. Mục tiêu đề tài:

1. So sánh các mô hình phân lớp dựa trên tập dữ liệu văn bản.
2. So sánh các mô hình phân lớp dựa trên tập dữ liệu hình ảnh.
3. So sánh các mô hình phân lớp dựa trên tập dữ liệu kết hợp giữa văn bản và hình ảnh.

V. Bố cục báo cáo:

➤ **Phần I. Giới thiệu**

Giới thiệu đề tài, tập dữ liệu, mục tiêu của đề tài và mô tả hệ thống để đánh giá các giải thuật.

➤ **Phần II. Nội dung**

Sơ lược lý thuyết về các mô hình phân lớp được sử dụng trong đề tài.

➤ **Phần III. Kết luận và hướng phát triển**

Kết luận về độ hiệu quả của các mô hình trên các loại dữ liệu và hướng phát triển của đề tài.

PHẦN II: NỘI DUNG

I. Khái quát về các mô hình học máy:

K láng giềng gần nhất (K Nearest Neighbors - KNN): Là thuật toán học máy đơn giản, sử dụng được cho mục đích phân lớp và hồi quy. KNN còn được gọi là thuật toán “lười học”, vì nó không tạo ra mô hình hoặc hàm giả thuyết từ dữ liệu huấn luyện. Thay vào đó, KNN lưu trữ toàn bộ tập dữ liệu huấn luyện và dự đoán dựa trên việc so sánh điểm dữ liệu mới với các điểm dữ liệu trong tập huấn luyện, cụ thể: Khi có phần tử mới đến, thuật toán sẽ tính khoảng cách từ phần tử mới đó đến tất cả các phần tử trong tập huấn luyện, sau đó nếu là bài toán phân lớp, KNN sẽ lấy k phần tử gần nhất và phân loại dựa trên số đông, ngược lại, KNN sẽ tính trung bình k giá trị của k phần tử gần nhất cho phần tử mới. Ưu điểm của KNN là dễ triển khai và dễ hiểu. Tuy vậy, khi tập huấn luyện tăng lên yêu cầu lượng tài nguyên để lưu trữ nhiều hơn, độ phức tạp tính toán cũng tăng lên và việc chọn k thích hợp cũng ảnh hưởng đến hiệu suất của mô hình.

Máy học véc-tơ hỗ trợ (Support Vector Machine - SVM): Là thuật toán học máy được sử dụng cho cả bài toán phân lớp và hồi quy, nhưng thường được sử dụng với mục đích là cho bài toán phân lớp. SVM tìm một siêu phẳng (hyperplane) tối ưu cho phép chia các điểm (phần tử) dữ liệu thành hai phần sao cho các điểm cùng một lớp nằm về một phía của siêu phẳng này (bài toán hai lớp). Bằng việc kết hợp với phương pháp hàm nhân, SVM hỗ trợ cung cấp các mô hình hiệu quả chính xác cho các vấn đề phân lớp và hồi quy tuyến tính và phi tuyến trong thực tế. SVM nhận đầu vào là một hàm nhân (kernel function) sẽ tạo ra một mô hình mới mà không cần đến bất kỳ thay đổi nào từ mã chương trình. Giải thuật học sẽ dẫn đến giải bài toán quy hoạch toàn phương, luôn có kết quả tối ưu toàn cục. Có hai biến thể tiêu biểu của SVM, bao gồm: LinearSVM và RBF SVM. Tuy vậy, SVM đòi hỏi lượng tài nguyên tính toán đủ cho các tập dữ liệu lớn, đặc biệt là khi sử dụng các hạt nhân phức tạp hoặc tập dữ liệu có kích thước lớn và việc chọn siêu tham số cũng khá khó khăn.

Cây quyết định (Decision Tree): Là thuật toán học máy được sử dụng cho cả bài toán phân lớp và hồi quy. Là giải thuật có kết quả sinh ra dễ diễn dịch, khá đơn giản, nhanh, hiệu quả và được sử dụng nhiều. Quá trình xây dựng cây dựa trên việc phân chia tập dữ liệu thành các phần nhỏ hơn dựa trên các thuộc tính của dữ liệu. Quá trình này tiếp tục cho đến khi mỗi phần dữ liệu chỉ chứa các điểm thuộc cùng một lớp (bài toán phân lớp) hoặc không còn cách nào để phân chia (bài toán hồi quy). Tuy nhiên, cây quyết định cũng có thể dễ bị overfitting nếu không kiểm soát được và không hiệu quả khi dữ liệu có sự phụ thuộc phức tạp giữa các đặc trưng.

Rừng ngẫu nhiên (Random Forest): Là thuật toán học máy được sử dụng cho cả bài toán phân lớp và hồi quy. Rừng ngẫu nhiên là một tập hợp của nhiều cây quyết định độc lập. Mỗi cây quyết định được huấn luyện trên một tập dữ liệu con được tạo ra từ việc lấy mẫu ngẫu nhiên từ tập dữ liệu huấn luyện ban đầu. Quá trình này tạo ra sự đa dạng giữa các cây trong rừng và giảm thiểu được tình trạng overfitting. Khi dự đoán, rừng ngẫu nhiên tổng hợp kết quả từ tất cả các cây quyết định trong rừng và chọn ra dự đoán cuối cùng bằng cách sử dụng phương pháp như đa số hoặc trung bình. Tuy nhiên, kết quả sinh ra từ rừng ngẫu nhiên

lại khó diễn giải hơn so với mô hình đơn giản hơn như cây quyết định và khá tốn tài nguyên tính toán.

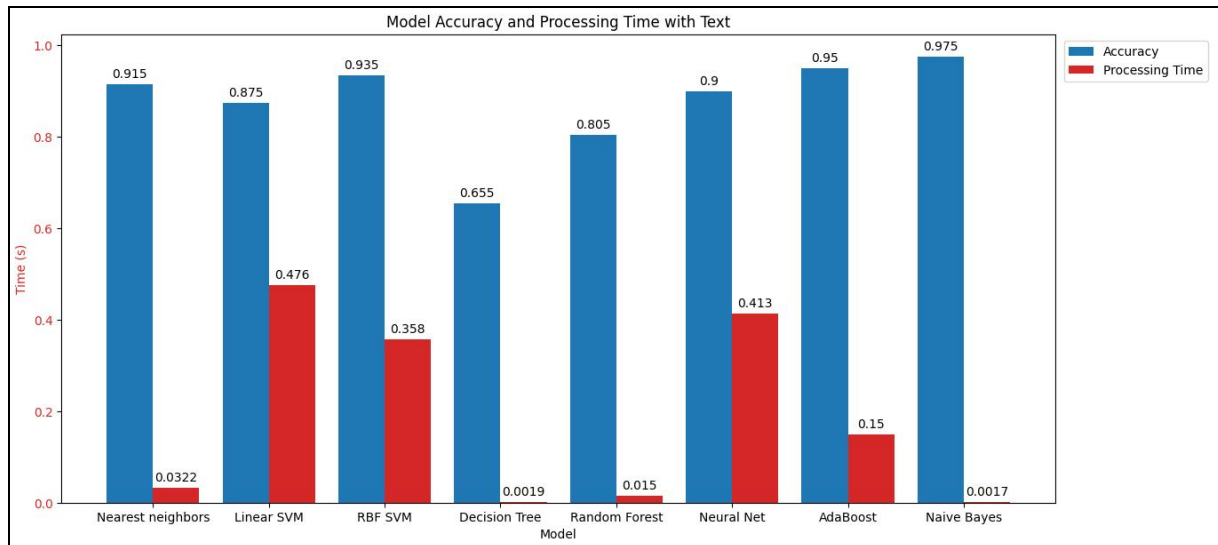
Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN): Mạng nơ-ron nhân tạo là một mô hình tính toán được xây dựng phỏng theo mạng nơ-ron sinh học. Mạng nơ-ron nhân tạo gồm một nhóm các nơ-ron nhân tạo (nút) nối với nhau, và xử lý thông tin bằng cách truyền theo các kết nối và tính giá trị mới tại các nút (cách tiếp cận connectionism đối với tính toán). Có thể hiểu một ANN bao gồm một số lượng lớn các nơ-ron nhân tạo, được tổ chức thành các tầng (layer). Các tầng thông thường bao gồm tầng đầu vào (input layer), một hoặc nhiều tầng ẩn (hidden layer) và một lớp đầu ra (output layer). Quá trình huấn luyện ANN thường dựa trên việc sử dụng các phương pháp như lan truyền ngược (backpropagation) để điều chỉnh các trọng số của mạng dựa trên sự khác biệt giữa dự đoán và kết quả thực tế. Tuy ANN có khả năng học và hiểu dữ liệu phức tạp, mạng nơ-ron nói chung đều đòi hỏi số lượng dữ liệu lớn để đạt được hiệu suất tốt.

AdaBoost (Adaptive Boosting): Là thuật toán học máy dùng trong bài toán phân lớp và là một những thuật toán tập hợp mô hình phổ biến nhất. AdaBoost hoạt động bằng cách xây dựng các mô hình phân loại yếu tuần tự. Mỗi mô hình yếu được huấn luyện trên một phiên bản mới của tập dữ liệu, trong đó có các trọng số được cập nhật sao cho các điểm dữ liệu bị phân loại sai trước đó sẽ nhận được trọng số cao hơn trong quá trình huấn luyện tiếp theo. Kết quả cuối cùng của AdaBoost là sự kết hợp của các dự đoán từ tất cả các mô hình yếu. Trọng số được gán cho mỗi mô hình yếu dựa trên hiệu suất của nó trong quá trình huấn luyện. Tuy vậy, AdaBoost khá nhạy cảm với dữ liệu nhiễu nếu không xử lý đúng cách và yêu cầu nhiều thời gian huấn luyện do quá trình huấn luyện tuần tự.

Bayes ngây thơ (Naive Bayes): Là thuật toán học máy dành cho bài toán phân lớp dựa trên nguyên lý của định lý Bayes với giả định “ngây thơ” (naive), tức là tất cả các đặc trưng đều độc lập với nhau khi đã biết lớp. Naive Bayes dựa trên công thức Bayes để tính xác suất của một lớp dựa trên các đặc trưng của dữ liệu. Cụ thể, nó tính xác suất có điều kiện có mỗi lớp dựa trên các đặc trưng, và sau đó dự đoán lớp với xác suất cao nhất. Naive Bayes có khá nhiều biến thể, bao gồm Naive Bayes Gaussian (cho dữ liệu có phân phối Gaussian), Naive Bayes Multinomial (cho dữ liệu đếm, thích hợp cho văn bản), Naive Bayes Bernoulli (cho dữ liệu nhị phân) và Naive Bayes Categorical (thích hợp cho các tác vụ phân loại văn bản). Nhược điểm của Naive Bayes chính là sự giả định về tính độc lập của các đặc trưng, có thể nó không thực tế trong nhiều tình huống và độ chính xác cũng bị ảnh hưởng nếu có đặc trưng không độc lập với các đặc trưng khác.

II. Đánh giá các mô hình học máy dựa trên tập dữ liệu văn bản và hình ảnh:

1. Đối với tập dữ liệu văn bản:



Hình 1. Biểu đồ thể hiện độ chính xác và thời gian xử lý các mô hình trên tập dữ liệu văn bản

Từ biểu đồ trên, dựa trên tập dữ liệu văn bản, mô hình cho ra độ chính xác cao nhất là Naive Bayes, với 97,5% và thời gian chạy của mô hình là 0.0017s - là mô hình có thời gian xử lý nhanh nhất.

Lí do mô hình Naive Bayes phù hợp với việc phân loại văn bản là:

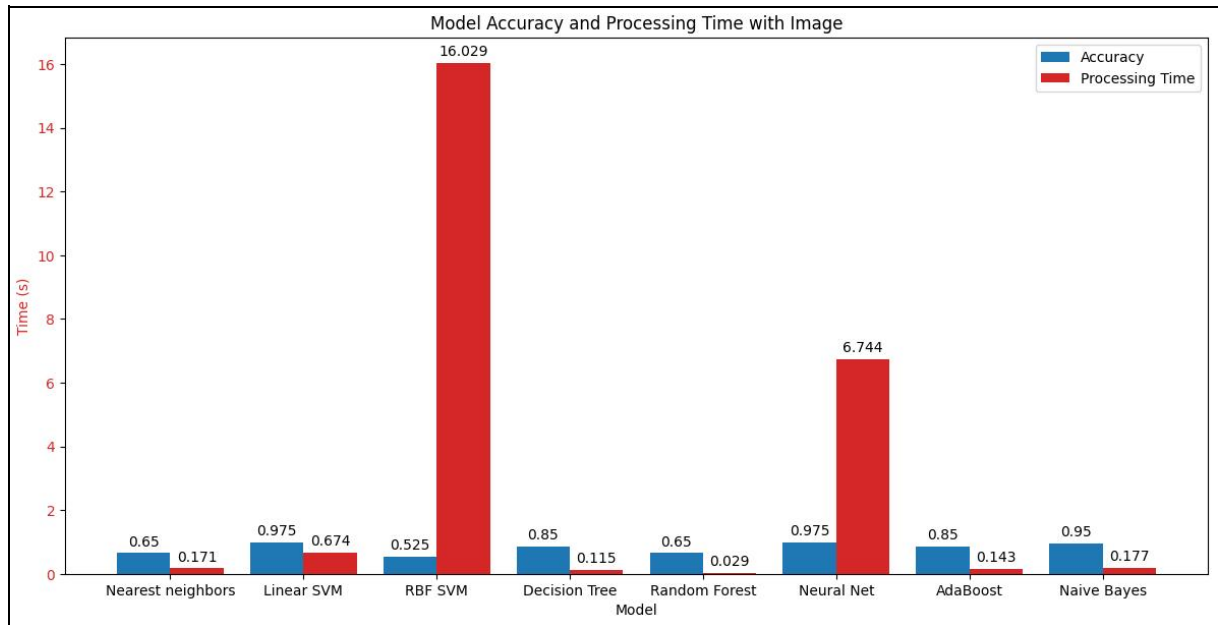
- ✧ Đơn giản và hiệu quả tính toán: Mô hình chỉ tính toán các xác suất đơn giản dựa trên các đặc trưng của văn bản. Chính vì vậy, khi nhắc đến phân loại văn bản, Naive Bayes là một sự lựa chọn lý tưởng.

- ✧ Giả định về độc lập: Mặc dù giả định rằng các đặc trưng độc lập với nhau có thể không hoàn toàn đúng trong ngôn ngữ tự nhiên, nhưng nó thường là một ước lượng hợp lý, đặc biệt khi xem xét các từ hoặc các đặc trưng nhỏ khác trong văn bản.

- ✧ Xử lý các đặc trưng đa chiều: Naive Bayes có thể xử lý các tập dữ liệu văn bản có số chiều cao mà không gặp vấn đề về hiệu suất hoặc sự phức tạp tính toán. Các đặc trưng có thể là các từ, n-gram, hoặc các đặc điểm khác của văn bản.

- ✧ Hiệu suất tốt trong thực tế: Mặc dù Naive Bayes có giả định đơn giản, nhưng nó thường đạt được hiệu suất tốt trong các tác vụ phân loại văn bản thực tế, như phân loại thư rác và phân loại văn bản theo chủ đề.

2. Đối với tập dữ liệu hình ảnh:



Hình 2. Biểu đồ thể hiện độ chính xác và thời gian xử lý các mô hình trên tập dữ liệu hình ảnh

Từ biểu đồ trên, dựa trên tập dữ liệu hình ảnh, có vài mô hình cho ra độ chính xác khá cao, như Linear SVM, Neural Net và Naive Bayes. Nhưng xét thêm khía cạnh thời gian xử lý thì mô hình Linear SVM đạt hiệu suất chấp nhận được do có độ chính xác khá cao (97.5%) và có thời gian xử lý tương đối nhanh (0.674s).

Naive Bayes dù có độ chính xác cao và thời gian xử lý nhanh hơn so với Linear SVM nhưng lại có sự khác nhau về mặt xử lý dữ liệu đầu vào. Naive Bayes giả định các thuộc tính đầu vào là độc lập với nhau và chỉ tính xác suất đơn giản, mà giả định này lại không đúng khi nói về dữ liệu hình ảnh vì các pixel trong hình ảnh tương tác với nhau và có mối quan hệ phức tạp. Neural Net mặc dù cho độ chính xác tương đương như Linear SVM, nhưng thời gian xử lý của Linear SVM lại vượt trội hơn hẳn so với Neural Net khi chỉ tốn chưa đến 1s so với xấp xỉ 7s.

Và khi nói đến xử lý dữ liệu hình ảnh, ngoài các mô hình học sâu kinh điển như CNN, SVM sẽ là một lựa chọn tối ưu, đặc biệt đối với tập dữ liệu này là Linear SVM, là do:

- ✧ Hiệu suất: LinearSVM thường cho hiệu suất tốt trên dữ liệu hình ảnh với các tính chất tuyến tính hoặc gần tuyến tính. Trong nhiều trường hợp, hình ảnh có thể được biểu diễn dưới dạng các vector đặc trưng, và các đặc trưng này có thể có tính tuyến tính.

- ✧ Khả năng xử lý dữ liệu lớn: LinearSVM có thể xử lý tập dữ liệu lớn hiệu quả, bao gồm cả khi số lượng đặc trưng của hình ảnh lớn. Điều này là do quy mô tuyến tính của việc tối ưu hóa trong SVM.

✧ Hiệu quả trong việc xử lý với số chiều cao: Dữ liệu hình ảnh thường có số chiều lớn, đặc biệt là khi xem xét các hình ảnh màu. LinearSVM có thể hoạt động tốt trên các không gian đặc trưng đa chiều mà không gặp vấn đề về sự phức tạp tính toán.

3. Đối với tập dữ liệu kết hợp văn bản và hình ảnh:

Text Image	Nearest Neighbors	Linear SVM	RBF SVM	Decision Tree	Random Forest	Neural Net	AdaBoost	Naive Bayes
Nearest Neighbors	0.775	0.75	0.75	0.7	0.65	0.675	0.725	0.725
Linear SVM	0.975	0.975	0.975	0.95	0.95	0.975	1.0	1.0
RBF SVM	0.925	0.875	0.85	0.65	0.6	0.825	0.95	0.975
Decision Tree	0.9	0.5	0.9	0.5	0.9	0.85	0.9	0.8
Random Forest	0.925	0.925	0.925	0.65	0.85	0.925	0.95	0.825
Neural Net	0.95	0.975	0.95	0.95	0.975	0.975	0.975	0.95
AdaBoost	0.9	0.8	0.8	0.8	0.8	0.85	0.85	0.85
Naive Bayes	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95

Bảng 2. Kết quả các mô hình phân lớp khi kết hợp văn bản và hình ảnh

Từ bảng trên, khi kết hợp hai loại dữ liệu văn bản và hình ảnh, có vài sự kết hợp cho ra độ chính xác 100% là: AdaBoost (Text) - Linear SVM (Image), Naive Bayes (Text) - Linear SVM (Image).

Khi kết hợp AdaBoost cho dữ liệu văn bản và Linear SVM cho dữ liệu hình ảnh cho ra độ chính xác cao là do: AdaBoost là một phương pháp tập trung vào xây dựng một mô hình dự đoán tốt bằng cách kết hợp nhiều mô hình yếu thành một mô hình mạnh. Trong trường hợp dữ liệu văn bản, AdaBoost có thể hiệu quả trong việc xử lý các từ với mức độ phức tạp khác nhau và xây dựng một mô hình tổng quát tốt. Còn đối với dữ liệu hình ảnh, như đã đề cập

trước đó, Linear SVM hoạt động rất tốt. Nên độ chính xác khi kết hợp hai mô hình là khá cao và thời gian xử lý cũng khá nhanh.

Khi kết hợp Naive Bayes cho dữ liệu văn bản và Linear SVM cho dữ liệu hình ảnh cho ra độ chính xác cao là do: Naive Bayes giả định rằng các đặc trưng độc lập với nhau, giúp giảm thiểu về độ phức tạp tính toán và độ chính xác khi phân loại dữ liệu văn bản là khá cao. Linear SVM có khả năng xử lý tốt trên các tập dữ liệu hình ảnh, đặc biệt là khi có mối quan hệ tuyến tính giữa các đặc trưng. SVM có thể tạo ra các ranh giới quyết định tuyến tính hiệu quả, giúp phân loại các hình ảnh vào các lớp khác nhau. Khi kết hợp có thể tạo ra mô hình phân lớp mạnh mẽ với độ chính xác khá cao.

PHẦN III. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

I. Kết luận:

Các mô hình phân lớp khi thực thi với tập dữ liệu văn bản và hình ảnh cho kết quả khá cao và thời gian xử lý có thể coi là chấp nhận được.

Ngoại trừ cây quyết định (Decision Tree) và rừng ngẫu nhiên (Random Forest), các mô hình còn lại đều cho kết quả khá cao và thời gian khá nhanh khi thực thi với dữ liệu văn bản, đặc biệt là Bayes thơ ngây (Naive Bayes).

Còn đối với tập dữ liệu hình ảnh, Linear SVM và mạng nơ-ron nhân tạo (Neural Net) cho kết quả khá cao và thời gian chạy cũng gọi là chấp nhận được.

Khi sử dụng tập dữ liệu văn bản và hình ảnh, các mô hình phân lớp đều cho kết quả khá cao. Sự kết hợp giữa văn bản và hình ảnh bằng thuật toán cây quyết định cho kết quả không được tốt, có thể bị ảnh hưởng từ tập dữ liệu hình ảnh, khi mà các đặc trưng quá phức tạp, dẫn đến cây bị overfitting.

Tóm lại, các mô hình phân lớp sẽ thể hiện được thế mạnh của nó trên những tập dữ liệu khác nhau, như: Bayes thơ ngây sẽ thể hiện được tính độc lập của các đặc trưng khi làm việc với dữ liệu văn bản, còn đối với dữ liệu hình ảnh, mạng nơ-ron nhân tạo hoặc máy học véc-tơ hỗ trợ có thể đạt được độ chính xác cao.

II. Hướng phát triển:

➤ **Tối ưu hóa mô hình hiện tại:** Điều chỉnh các tham số của mô hình hiện tại để tăng độ chính xác và hiệu suất. Các kỹ thuật tối ưu hóa bao gồm điều chỉnh learning rate, tăng số lượng lớp và đơn vị ẩn, sử dụng kỹ thuật regularization như dropout để tránh overfitting.

➤ **Tăng cường dữ liệu huấn luyện:** Một cách phổ biến để cải thiện hiệu suất của mô hình là tăng cường dữ liệu huấn luyện bằng cách áp dụng các kỹ thuật như đảo ngược, cắt, xoay, biến dạng ảnh, và thêm nhiễu vào dữ liệu.

➤ **Sử dụng các mô hình mạng sâu:** Khám phá sự hiệu quả của các mô hình mạng nơ-ron sâu như Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), và các biến thể của chúng trên dữ liệu văn bản và hình ảnh.

➤ **Transfer Learning:** Sử dụng các mô hình đã được huấn luyện trước trên các tập dữ liệu lớn như VGG, ResNet, hoặc BERT và Fine-tuning chúng trên dữ liệu mới. Việc này giúp tận dụng kiến thức đã học được từ dữ liệu lớn và giảm thiểu thời gian huấn luyện.

➤ **Kỹ thuật Ensemble:** Kết hợp nhiều mô hình khác nhau để tạo ra một mô hình tổng hợp có hiệu suất cao hơn. Các phương pháp ensemble bao gồm Majority Voting, Weighted Average, và Stacking.

TÀI LIỆU THAM KHẢO

- [1] Giáo trình Nguyên lý máy học / Đỗ Thanh Nghị, Phạm Nguyên Khang - Cần Thơ: NXB Đại học Cần Thơ, 2012.
- [2] Slides bài giảng Machine Learning từ CMU, School of Computer Science.

--- HẾT ---