**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

---------------------------



**BÁO CÁO ĐỀ TÀI MÔN HỌC  
KHAI PHÁ DỮ LIỆU**

**XÂY DỰNG HỆ THỐNG PHÂN LOẠI HÌNH ẢNH SỐ VIẾT TAY BẰNG K- NN**

**GVHD:** Đinh Nguyễn Trọng Nghĩa

**Lớp:** 13DHTH\_TD (thứ 3 tiết 4 – 6)

**NHÓM THỰC HIỆN:**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Bùi Đỗ Phúc Nguyên | 2001223151 |
| 2 | Lê Trương Công Hiếu | 2001221414 |
| 3 | Ngô Thị Thanh Huyền | 2001221741 |

*TP. Hồ Chí Minh, ngày 6 tháng 5 năm 2025*

MỤC LỤC

[MỞ ĐẦU 1](#_Toc197367011)

[CHƯƠNG 1 : LÝ THUYẾT VỀ HỌC CÓ GIÁM SÁT 2](#_Toc197367012)

[**1.1.** **Khái niệm** 2](#_Toc197367013)

[**1.2.** **Các bài toán trong học có giám sát** 2](#_Toc197367014)

[**1.3.** **Một số thuật toán thông dụng** 2](#_Toc197367015)

[CHƯƠNG 2. THUẬT TOÁN K-NN 4](#_Toc197367016)

[**2.1.** **Giới thiệu sơ lược** 4](#_Toc197367017)

[**2.2.** **Chi tiết thuật toán** 4](#_Toc197367018)

[**2.3.** **Ví dụ minh hoạ thuật toán K-NN** 5](#_Toc197367019)

[**2.4.** **Ưu nhược điểm của thuật toán K-NN** 6](#_Toc197367020)

[**3.1.** **Dữ liệu sử dụng** 8](#_Toc197367021)

[**3.1.1.** **Tổng quan về tập dữ liệu MNIST** 8](#_Toc197367022)

[**3.1.2.** **Quá trình xử lý dữ liệu** 9](#_Toc197367023)

[**3.2.** **Xây dựng ứng dụng** 10](#_Toc197367024)

[**3.2.1.** **Các bước thực hiện** 10](#_Toc197367025)

[**3.2.2.** **Các chức năng** 11](#_Toc197367026)

[CHƯƠNG 4: KẾT QUẢ VÀ PHÂN TÍCH 14](#_Toc197367027)

[**4.1.**  **Kết quả thu được** 14](#_Toc197367028)

[**4.1.1. Độ chính xác so với các giá trị k** 14](#_Toc197367029)

[**4.1.2. Hiệu suất của mô hình** 14](#_Toc197367030)

[**4.1.3. Ma trận nhầm lẫn** 15](#_Toc197367031)

[**4.1.4. Mô hình PCA** 17](#_Toc197367032)

[**4.2.**  **Đánh giá kết quả** 17](#_Toc197367033)

[**4.2.1. Ưu điểm** 18](#_Toc197367034)

[**4.2.2. Hạn chế** 18](#_Toc197367035)

[CHƯƠNG 5. NHỮNG THÁCH THỨC VÀ HƯỚNG PHÁT TRIỂN 20](#_Toc197367036)

[KẾT LUẬN 22](#_Toc197367037)

[TÀI LIỆU THAM KHẢO 23](#_Toc197367038)

[PHỤ LỤC 24](#_Toc197367039)

**MỞ ĐẦU**

Trong thời đại bùng nổ công nghệ thông tin và trí tuệ nhân tạo, việc xử lý và phân tích hình ảnh đóng vai trò then chốt trong nhiều lĩnh vực như nhận dạng chữ viết, an ninh, giáo dục và y tế. Một trong những ứng dụng tiêu biểu là nhận dạng chữ viết tay — một bài toán không chỉ mang tính thực tiễn cao mà còn thể hiện rõ những thách thức của lĩnh vực học máy.

Đề tài *“Xây dựng hệ thống phân loại hình ảnh số viết tay bằng KNN”* nhằm mục tiêu nghiên cứu và triển khai thuật toán K-Nearest Neighbors – KNN để phân loại các chữ số viết tay. Đây là một trong những thuật toán đơn giản nhưng hiệu quả trong học máy, đặc biệt phù hợp với các bài toán phân loại dựa trên sự tương đồng giữa các đặc trưng.

Thông qua đề tài này, nhóm thực hiện mong muốn hiểu rõ hơn về quá trình tiền xử lý dữ liệu hình ảnh, trích xuất đặc trưng, cũng như áp dụng và đánh giá hiệu quả của thuật toán KNN trên tập dữ liệu chữ số viết tay phổ biến như MNIST. Bên cạnh đó, hệ thống được xây dựng còn có thể làm nền tảng cho các nghiên cứu sâu hơn trong lĩnh vực nhận dạng và thị giác máy tính.

# **CHƯƠNG 1 : LÝ THUYẾT VỀ HỌC CÓ GIÁM SÁT**

* 1. **Khái niệm**

Học có giám sát (Supervised Learning) là một phương pháp trong Machine Learning, nơi mô hình được huấn luyện dựa trên dữ liệu có nhãn (labeled data). Mỗi mẫu dữ liệu bao gồm:

* + - Đầu vào (Input): Đặc trưng (features) mô tả đối tượng (ví dụ: pixel ảnh, giá trị cảm biến).
    - Đầu ra (Output): Nhãn (label) hoặc giá trị mục tiêu (target) tương ứng (ví dụ: "chó", "mèo" hoặc giá nhà).

Đặc điểm chính:

* Mục tiêu là học một hàm ánh xạ từ đầu vào đến đầu ra.
* Yêu cầu dữ liệu huấn luyện phải được gán nhãn chính xác.
* Ứng dụng khi bài toán có kết quả mong muốn rõ ràng.

Ví dụ:

* Dự đoán giá nhà dựa trên diện tích, số phòng ngủ (hồi quy).
* Phân loại email là "spam" hoặc "không spam" (phân loại).
  1. **Các bài toán trong học có giám sát**

Có hai loại bài toán chính:

a. Bài toán phân loại (Classification)

Mục tiêu: Gán nhãn rời rạc cho dữ liệu.

Đầu ra: Lớp (class) thuộc tập hữu hạn (ví dụ: 0/1, "A/B/C").

Ví dụ:

* Nhận diện chữ số viết tay (MNIST dataset).
* Chẩn đoán bệnh dựa trên triệu chứng (bệnh/có bệnh).

b. Bài toán hồi quy (Regression)

Mục tiêu: Dự đoán giá trị liên tục.

Đầu ra: Số thực (ví dụ: giá cả, nhiệt độ).

Ví dụ:

* Dự báo doanh thu dựa trên chi phí quảng cáo.
* Ước lượng tuổi thọ pin dựa trên số lần sạc.
  1. **Một số thuật toán thông dụng**
* Hồi quy tuyến tính (Linear Regression)

Mục đích: Dự đoán giá trị liên tục bằng cách tìm đường thẳng phù hợp nhất với dữ liệu.

* Cây quyết định (Decision Tree)

Mục đích: Phân chia dữ liệu thành các nhóm dựa trên các câu hỏi if-else.

* SVM (Support Vector Machine)

Mục đích: Tìm siêu phẳng tối ưu để phân tách các lớp dữ liệu.

* Random Forest

Mục đích: Kết hợp nhiều cây quyết định để giảm overfitting.

* K-Nearest Neighbors (KNN)

Mục đích: Phân loại dựa trên khoảng cách giữa các điểm dữ liệu.

**CHƯƠNG 2. THUẬT TOÁN K-NN**

* 1. **Giới thiệu sơ lược**

K-Nearest Neighbors (K-NN) là một thuật toán học máy thuộc nhóm **học có giám sát** (supervised learning). Được phát triển từ những năm 1950, thuật toán này dựa trên ý tưởng đơn giản nhưng hiệu quả**:** phân loại hoặc dự đoán giá trị của một điểm dữ liệu mới dựa trên các điểm lân cận gần nhất trong tập dữ liệu huấn luyện.

Nguyên lý hoạt động:

Khi nhận một điểm dữ liệu mới, K-NN tính toán khoảng cách từ điểm này đến tất cả các điểm trong tập dữ liệu. Sau đó, chọn ra **K điểm gần nhất** (hàng xóm) dựa trên khoảng cách. Trong bài toán phân loại, dự đoán được thực hiện bằng cách chọn nhãn xuất hiện nhiều nhất trong K điểm lân cận (đa số phiếu). Trong bài toán hồi quy, giá trị đầu ra được tính bằng **trung bình giá trị** của K điểm gần nhất.

Đặc trưng của K-NN:

* Không yêu cầu quá trình "học" trước (lazy learning).
* Tính đơn giản và dễ triển khai, nhưng phụ thuộc nhiều vào cách chọn giá trị K và khoảng cách.
  1. **Chi tiết thuật toán**

Quá trình phân cụm được thực hiện thông qua việc xác định các tâm cụm (centroids) và gán các điểm dữ liệu vào các cụm dựa trên khoảng cách đến các tâm cụm này. Thuật toán sẽ lặp đi lặp lại các bước cho đến khi không còn thay đổi đáng kể nào xảy ra trong việc gán nhãn các điểm hoặc vị trí của các tâm cụm.

Các bước chi tiết của thuật toán:

**Bước 1: Chuẩn bị dữ liệu**

Tập dữ liệu huấn luyện cần có các đặc trưng (features) và nhãn (label) do đây là thuật toán học có giám sát. Ví dụ: Một tập dữ liệu huấn luyện về phân loại trái cây có các đặc trưng như màu sắc, kích thước, trọng lượng, và nhãn là loại trái cây (táo, cam,...).

**Bước 2:** **Xác định giá trị K**

K là số lượng hàng xóm gần nhất được sử dụng để dự đoán. Giá trị K ảnh hưởng lớn đến độ chính xác của thuật toán:

* *K quá nhỏ: Dễ bị nhiễu do các ngoại lệ.*
* *K quá lớn: Kết quả bị ảnh hưởng bởi quá nhiều điểm không liên quan.*

**Bước 3: Tính khoảng cách**

Khoảng cách giữa điểm dữ liệu mới và từng điểm trong tập huấn luyện được tính toán

**Bước 4: Chọn K điểm lân cận gần nhất**

Sắp xếp khoảng cách và chọn ra K điểm có khoảng cách nhỏ nhất.

**Bước 5: Dự đoán kết quả**

* **Phân loại**: Nhãn của điểm mới được xác định dựa trên nhãn xuất hiện nhiều nhất trong K điểm.
* **Hồi quy**: Giá trị dự đoán được tính là trung bình các giá trị của K điểm lân cận.
  1. **Ví dụ minh hoạ thuật toán K-NN**

**Bài toán**: Phân loại một khách hàng là **"Người dùng tiềm năng"** hay **"Người dùng không tiềm năng"** dựa trên các đặc trưng như: **độ tuổi, thu nhập (triệu VND/tháng)** và **số lần truy cập trang web/tháng.**

**Tập dữ liệu huấn luyện**

|  |  |  |  |
| --- | --- | --- | --- |
| Độ tuổi | Thu nhập (triệu VND) | Số lần truy cập | Nhãn (Label) |
| 25 | 20 | 10 | Người dùng không tiềm năng |
| 30 | 35 | 20 | Người dùng tiềm năng |
| 22 | 18 | 8 | Người dùng không tiềm năng |
| 35 | 50 | 30 | Người dùng tiềm năng |
| 28 | 25 | 15 | Người dùng tiềm năng |
| 40 | 60 | 25 | Người dùng tiềm năng |
| 20 | 15 | 5 | Người dùng không tiềm năng |
| 26 | 22 | 12 | ? |

**Bước 1:** Xác định số K

Chọn K = 3 (số lượng hàng xóm gần nhất để dự đoán).

**Bước 2:** Tính khoảng cách giữa khách hàng mới với từng khách hàng trong tập dữ liệu huấn luyện thông qua 3 đặc trưng trên.

Sử dụng khoảng cách Euclidean để tính toán khoảng cách giữa khách hàng mới và từng khách hàng trong tập dữ liệu huấn luyện.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Độ tuổi | Thu nhập (triệu VND) | Số lần truy cập | Nhãn (Label) | Khoảng cách |
| 25 | 20 | 10 | Người dùng không tiềm năng | 3 |
| 30 | 35 | 20 | Người dùng tiềm năng | 15.78 |
| 22 | 18 | 8 | Người dùng không tiềm năng | 6.93 |
| 35 | 50 | 30 | Người dùng tiềm năng | 34.5 |
| 28 | 25 | 15 | Người dùng tiềm năng | 4.69 |
| 40 | 60 | 25 | Người dùng tiềm năng | 42.53 |
| 20 | 15 | 5 | Người dùng không tiềm năng | 11.58 |
| 26 | 22 | 12 | ? |

**Bước 3:** Chọn K điểm có khoảng cách gần nhất

Sắp xếp các khách hàng theo khoảng cách:

2.83

5.39

6.93

**Bước 4:** Dự đoán nhãn

Trong 3 khách hàng gần nhất:

* KH1: Không tiềm năng
* KH5: Tiềm năng
* KH3: Không tiềm năng

Nhãn xuất hiện nhiều nhất là **"Không tiềm năng”**

**Kết luận**: Dự đoán khách hàng mới là **"Người dùng không tiềm năng"**.

* 1. **Ưu nhược điểm của thuật toán K-NN**

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Đơn giản, dễ hiểu và dễ triển khai. | Hiệu suất kém với dữ liệu lớn do phải tính khoảng cách cho tất cả các điểm. |
| Thích hợp cho việc phân loại dữ liệu | Nhạy cảm với giá trị K và cách chọn khoảng cách. |
| Phụ thuộc nhiều vào chất lượng dữ liệu, nhạy cảm với ngoại lệ. |

**CHƯƠNG 3. PHÂN TÍCH DỮ LIỆU VÀ XÂY DỰNG ỨNG DỤNG**

* 1. **Dữ liệu sử dụng**
     1. **Tổng quan về tập dữ liệu MNIST**

Tập dữ liệu MNIST (Modified National Institute of Standards and Technology) là một cơ sở dữ liệu nổi tiếng trong lĩnh vực nhận dạng chữ số viết tay. Nó bao gồm tổng cộng 70.000 hình ảnh chữ số viết tay từ 0 đến 9, trong đó có 60.000 mẫu dùng để huấn luyện và 10.000 mẫu dùng để kiểm tra. Mỗi hình ảnh là ảnh xám (grayscale) với kích thước 28x28 pixel, được chuẩn hóa và căn giữa để giảm thiểu sự biến đổi về vị trí và kích thước của chữ số.

MNIST được xây dựng bằng cách kết hợp và xử lý lại từ hai tập dữ liệu gốc của NIST: SD-3 và SD-7. Quá trình này bao gồm việc chọn lọc, căn chỉnh và chuẩn hóa các hình ảnh để tạo ra một tập dữ liệu thống nhất và dễ sử dụng cho các thuật toán học máy. Mỗi hình ảnh trong MNIST được lưu trữ dưới định dạng IDX, một định dạng nhị phân đơn giản được thiết kế để lưu trữ mảng đa chiều, giúp việc đọc và xử lý dữ liệu trở nên hiệu quả hơn.

MNIST thường được sử dụng như một bài toán "Hello World" trong lĩnh vực học máy và thị giác máy tính. Nó là một công cụ tuyệt vời để thử nghiệm và so sánh các thuật toán phân loại, từ các mô hình đơn giản như K-Nearest Neighbors (KNN) đến các mạng nơ-ron sâu (Deep Neural Networks). Độ phổ biến và tính chuẩn hóa của MNIST giúp các nhà nghiên cứu và sinh viên dễ dàng đánh giá hiệu suất của các mô hình và thuật toán mới.

* **Đặc điểm nổi bật**

Một trong những điểm mạnh đáng chú ý của tập dữ liệu MNIST là tính đa dạng về chữ viết. Các chữ số trong MNIST được viết bởi nhiều người khác nhau, bao gồm cả nhân viên của Cục Thống kê Hoa Kỳ và học sinh trung học. Điều này giúp tập dữ liệu phản ánh được sự phong phú trong phong cách viết tay của con người, từ đó nâng cao khả năng khái quát và độ tin cậy khi đánh giá các mô hình nhận dạng chữ số.

Bên cạnh đó, mỗi hình ảnh trong MNIST đều được chuẩn hóa và căn giữa một cách kỹ lưỡng. Cụ thể, các chữ số được đưa về kích thước chuẩn 28x28 pixel, căn giữa trên nền trắng và được điều chỉnh để giảm thiểu sự lệch vị trí hay thay đổi không cần thiết về tỷ lệ. Nhờ vậy, các mô hình học máy có thể dễ dàng tập trung vào các đặc trưng chính của chữ số mà không bị nhiễu bởi các yếu tố hình thức.

Tập dữ liệu MNIST cũng nổi bật bởi tính dễ dàng truy cập và sử dụng. Nó được tích hợp sẵn trong nhiều thư viện học máy phổ biến như TensorFlow, PyTorch và Keras. Điều này cho phép người dùng chỉ cần một vài dòng lệnh đơn giản là có thể tải xuống và sử dụng ngay tập dữ liệu trong các dự án hoặc thử nghiệm mô hình.

Cuối cùng, MNIST sử dụng định dạng dữ liệu hiệu quả là IDX – một định dạng nhị phân đơn giản, tối ưu cho việc lưu trữ và truy xuất các mảng số liệu lớn. Việc sử dụng định dạng này không chỉ giúp giảm dung lượng tập tin mà còn tăng tốc độ đọc và xử lý dữ liệu, đặc biệt có lợi trong các hệ thống yêu cầu hiệu năng cao hoặc khi áp dụng cho các tập dữ liệu lớn hơn

* + 1. **Quá trình xử lý dữ liệu**

Trong hệ thống phân loại chữ số viết tay được xây dựng, dữ liệu đầu vào được xử lý qua nhiều bước nhằm tăng hiệu quả học máy và độ chính xác của mô hình. Đầu tiên, tập dữ liệu MNIST được tải thông qua thư viện torchvision.datasets. Bộ dữ liệu này bao gồm 60.000 ảnh huấn luyện và 10.000 ảnh kiểm tra, với mỗi ảnh là một chữ số từ 0 đến 9, được định dạng dưới dạng grayscale 28x28 pixel.

Sau khi tải về, các ảnh huấn luyện được chuẩn hóa bằng cách chia giá trị pixel cho 255 để đưa dữ liệu về dải giá trị từ 0 đến 1, phù hợp cho việc huấn luyện mô hình. Mỗi ảnh sau đó được làm phẳng từ ma trận 2 chiều (28x28) thành vector 1 chiều gồm 784 phần tử để thuận tiện cho việc đưa vào thuật toán học máy.

Một bước quan trọng trong xử lý dữ liệu là tăng cường dữ liệu (data augmentation) nhằm làm phong phú thêm tập huấn luyện và cải thiện khả năng tổng quát hóa của mô hình. Cụ thể, mỗi ảnh được sao chép và biến đổi nhiều lần thông qua các kỹ thuật như xoay ảnh một góc ngẫu nhiên (±10 độ) và dịch chuyển ảnh theo chiều ngang hoặc dọc

một vài pixel. Việc này giúp mô phỏng các biến thể tự nhiên trong chữ viết tay, từ đó làm cho mô hình trở nên linh hoạt và chính xác hơn khi gặp dữ liệu thực tế.

Sau khi tăng cường, tập dữ liệu huấn luyện được đưa vào quá trình giảm chiều bằng PCA (Principal Component Analysis). Việc giảm chiều từ 784 xuống còn 700 giúp loại bỏ các thành phần nhiễu không quan trọng, làm giảm độ phức tạp tính toán mà vẫn giữ lại phần lớn thông tin cần thiết để phân biệt các chữ số. PCA cũng giúp cải thiện tốc độ huấn luyện và khả năng phân loại của mô hình KNN.

Cuối cùng, dữ liệu sau xử lý được dùng để huấn luyện mô hình K-Nearest Neighbors (KNN) với các giá trị k từ 1 đến 15. Mỗi mô hình được đánh giá độ chính xác trên tập kiểm tra, và mô hình có k tốt nhất (mang lại độ chính xác cao nhất) sẽ được lựa chọn để sử dụng cho việc dự đoán đầu vào thực tế.

* 1. **Xây dựng ứng dụng**
     1. **Các bước thực hiện**

Ngôn ngữ sử dụng: Python

Công cụ: Visual studio code

Đầu tiên, sau khi dữ liệu đã được xử lý và chuẩn hóa, mô hình giảm chiều dữ liệu bằng kỹ thuật PCA (Principal Component Analysis). PCA giúp giảm số lượng đặc trưng đầu vào, từ đó rút ngắn thời gian huấn luyện và tăng độ ổn định cho thuật toán nhận dạng.

Tiếp theo, mô hình K-Nearest Neighbors (KNN) được sử dụng để huấn luyện và dự đoán. Một vòng lặp thử nghiệm nhiều giá trị khác nhau của tham số k (số lượng hàng xóm gần nhất) được thực hiện để tìm ra giá trị tối ưu nhằm đạt độ chính xác cao nhất trên tập kiểm tra. Sau khi xác định được giá trị k phù hợp, mô hình KNN được huấn luyện lại toàn bộ trên tập dữ liệu đã được giảm chiều.

Khi mô hình đã hoàn tất quá trình học, toàn bộ mô hình bao gồm thuật toán PCA, KNN và các thông số tối ưu sẽ được lưu lại bằng thư viện joblib. Việc lưu mô hình cho phép ứng dụng có thể tải mô hình đã học trong những lần chạy tiếp theo mà không cần

phải huấn luyện lại từ đầu, giúp rút ngắn thời gian xử lý và tăng trải nghiệm người dùng.

Trong quá trình sử dụng ứng dụng, khi người dùng vẽ hoặc tải lên ảnh chữ số, ảnh đầu vào sẽ được xử lý tương tự như trong giai đoạn huấn luyện, sau đó đưa vào mô hình PCA để giảm chiều, rồi tiếp tục được đưa vào mô hình KNN đã huấn luyện để dự đoán chữ số. Kết quả dự đoán được hiển thị lên giao diện người dùng dưới dạng số cụ thể và biểu đồ xác suất, giúp người dùng dễ dàng kiểm tra và đánh giá hiệu quả nhận dạng của mô hình.

* + 1. **Các chức năng**

Ứng dụng nhận dạng chữ số viết tay được xây dựng với giao diện trực quan và thân thiện, cung cấp đầy đủ các chức năng cần thiết phục vụ cho quá trình thử nghiệm và đánh giá mô hình học máy. Người dùng có thể trực tiếp vẽ chữ số bằng chuột ngay trên khung vẽ của ứng dụng hoặc tải ảnh chữ số từ máy tính thông qua chức năng "Import Image". Sau khi nhập ảnh, ứng dụng cung cấp chức năng "Predict" để dự đoán chữ số mà người dùng đã vẽ hoặc tải lên. Kết quả nhận dạng được hiển thị ngay trên giao diện dưới dạng chữ số dự đoán, đi kèm với tỷ lệ xác suất dự đoán cho từng lớp chữ số từ 0 đến 9, giúp người dùng hiểu rõ mức độ tự tin của mô hình.

Ngoài ra, ứng dụng còn có chức năng "Clear" để xóa nhanh khung vẽ, giúp người dùng dễ dàng thử lại nhiều lần. Các thông tin mô hình như loại thuật toán sử dụng, giá trị k tối ưu và độ chính xác trên tập kiểm tra cũng được hiển thị rõ ràng, hỗ trợ việc theo dõi và đánh giá chất lượng nhận dạng. Nhờ các chức năng này, ứng dụng vừa đóng vai trò như một công cụ minh họa mô hình học máy, vừa tạo điều kiện thuận lợi cho người dùng trải nghiệm trực tiếp quá trình nhận dạng chữ số viết tay.

**A screenshot of a computer

AI-generated content may be incorrect.**

*Hình 1. Giao diện màn hình chính.*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 2. Giao diện khi đưa ra kết quả dự đoán.*

A screenshot of a computer

AI-generated content may be incorrect., Picture

*Hình 3. Nhận diện hình không phải chữ số.*

**CHƯƠNG 4: KẾT QUẢ VÀ PHÂN TÍCH**

**4.1. Kết quả thu được**

**4.1.1. Độ chính xác so với các giá trị k**

Biểu đồ trên thể hiện mối quan hệ giữa độ chính xác (accuracy) và số lượng hàng xóm (k) trong mô hình K-Nearest Neighbors (KNN). Trục hoành biểu diễn giá trị k – tức là số lượng hàng xóm được xét đến để phân loại một điểm dữ liệu mới. Trục tung biểu diễn độ chính xác đạt được khi mô hình được huấn luyện với từng giá trị k tương ứng.

Từ biểu đồ, có thể thấy rằng khi giá trị k tăng từ 1 đến 9, độ chính xác của mô hình cũng tăng nhẹ, đạt đỉnh tại k = 9 với độ chính xác khoảng 97.54%. Tuy nhiên, sau mốc này, khi k tiếp tục tăng lên, mô hình bắt đầu có dấu hiệu giảm hiệu quả. Cụ thể, tại k = 11, độ chính xác giảm đáng kể, cho thấy sự ảnh hưởng tiêu cực khi lấy quá nhiều hàng xóm, có thể khiến mô hình mất đi tính phân biệt giữa các lớp.

Biểu đồ này giúp xác định giá trị k tối ưu để áp dụng vào mô hình thực tế. Trong trường hợp này, k = 7 là lựa chọn phù hợp nhất vì đạt được độ chính xác cao nhất trên tập kiểm tra.

A graph with green line

AI-generated content may be incorrect., Picture

*Hình 4. Biểu đồ thể hiện độ chính xác với từng giá trị k.*

**4.1.2. Hiệu suất của mô hình**

Biểu đồ trên minh họa bốn chỉ số đánh giá chính của mô hình học máy, bao gồm: Accuracy (Độ chính xác), Precision (Độ chính xác riêng theo lớp), Recall (Khả năng truy hồi) và F1-Score. Các giá trị này đều nằm rất gần ngưỡng tối đa là 1.0, cho thấy mô hình hoạt động với hiệu suất rất cao trên tập dữ liệu kiểm thử.

* Accuracy đạt khoảng 0.975, thể hiện tỷ lệ dự đoán đúng trên toàn bộ tập dữ liệu.
* Precision và Recall có giá trị tương đương nhau, điều này chứng tỏ mô hình không chỉ phân loại đúng mà còn đảm bảo không bỏ sót các mẫu quan trọng.
* F1-Score, là trung bình điều hòa giữa Precision và Recall, cũng ở mức cao, khẳng định sự cân bằng giữa hai yếu tố này trong mô hình.

Tổng thể, biểu đồ này thể hiện rằng mô hình huấn luyện đạt được độ tin cậy cao và có thể áp dụng tốt cho bài toán hiện tại.

A diagram of a model evaluation metrics

AI-generated content may be incorrect., Picture

*Hình 5. Biểu đồ đánh giá hiệu suất mô hình phân loại chữ số.*

**4.1.3. Ma trận nhầm lẫn**

Biểu đồ trên thể hiện ma trận nhầm lẫn của mô hình phân loại chữ số viết tay trên tập kiểm thử. Trục tung biểu diễn nhãn thực tế, trong khi trục hoành là nhãn mà mô hình dự đoán. Mỗi ô (i, j) trong ma trận biểu thị số lần mà mẫu có nhãn thực tế là *i* được dự đoán thành *j*.

Dựa vào ma trận:

1. Các giá trị trên đường chéo chính (từ trái trên đến phải dưới) là số lần mô hình dự đoán đúng. Ta có thể thấy rằng phần lớn các giá trị này đều rất cao, chẳng hạn như:

* Số "1" được dự đoán đúng 1132 lần.
* Số "0" là 975 lần.
* Số "9" là 975 lần, v.v…

1. Các giá trị ngoài đường chéo là các trường hợp dự đoán sai. Tuy xuất hiện rải rác nhưng tần suất tương đối thấp. Ví dụ:

* Số "2" bị nhầm với số "8" trong 15 trường hợp.
* Số "4" bị nhầm thành "9" trong 16 trường hợp.
* Số “1” bị nhầm lẫn với số 7 trong 18 trường hợp.

Nhìn chung, mô hình đạt độ chính xác rất cao với rất ít lỗi nhầm lẫn, chủ yếu xảy ra giữa các chữ số có hình dạng tương đối giống nhau. Biểu đồ này minh họa rõ ràng khả năng tổng quát và độ tin cậy của mô hình khi áp dụng trên dữ liệu thực tế.

A graph with blue squares

AI-generated content may be incorrect., Picture

*Hình 6. Ma trận nhầm lẫn.*

**4.1.4. Mô hình PCA**

Biểu đồ trên thể hiện tỷ lệ phương sai tích lũy của các thành phần chính khi áp dụng kỹ thuật giảm chiều PCA (Principal Component Analysis) lên tập dữ liệu. Trục hoành biểu diễn số lượng thành phần chính (principal components) được giữ lại, trong khi trục tung biểu diễn tổng phương sai tích lũy mà các thành phần này giải thích được.

Quan sát biểu đồ có thể thấy rằng, khi số lượng thành phần chính tăng, tỷ lệ phương sai tích lũy cũng tăng tương ứng. Đặc biệt, chỉ cần khoảng từ 50 đến 100 thành phần chính là đã giữ lại hơn 90% thông tin (phương sai) của toàn bộ dữ liệu gốc. Sau ngưỡng này, việc thêm thành phần chính chỉ mang lại mức cải thiện rất nhỏ, thể hiện qua độ dốc của đường cong giảm dần.

Kết quả này cho thấy việc sử dụng PCA không chỉ giúp giảm đáng kể số chiều của dữ liệu, từ đó tăng hiệu quả tính toán, mà còn vẫn đảm bảo giữ lại hầu hết thông tin quan trọng. Việc lựa chọn số lượng thành phần chính phù hợp sẽ giúp tối ưu mô hình mà không làm mất mát dữ liệu quá nhiều.

A graph with a red line

AI-generated content may be incorrect., Picture

*Hình 7. Biểu đồ tỷ lệ phương sai tích lũy theo số lượng thành phần chính.*

**4.2. Đánh giá kết quả**

**4.2.1. Ưu điểm**

Mô hình phân loại chữ viết tay sử dụng kết hợp giữa PCA và KNN cho thấy hiệu quả rất cao với độ chính xác gần 98% trên tập kiểm thử. Đây là một kết quả đáng ghi nhận, nhất là khi mô hình sử dụng các thuật toán đơn giản và dễ triển khai. Hiệu suất cao này cho thấy khả năng nhận diện chữ số viết tay của mô hình là đáng tin cậy và phù hợp cho các ứng dụng thực tế.

Ngoài ra, các chỉ số như Precision, Recall và F1-Score đều duy trì ở mức ổn định, xấp xỉ 0.98 trên hầu hết các lớp dữ liệu. Điều này chứng minh rằng mô hình hoạt động đồng đều, không bị thiên lệch và có khả năng tổng quát hóa tốt đối với dữ liệu chưa từng thấy. Ma trận nhầm lẫn thu được sau huấn luyện cũng cho thấy số lượng dự đoán sai là rất ít và phân bố đều giữa các lớp.

Việc áp dụng kỹ thuật giảm chiều dữ liệu bằng PCA là một trong những điểm nổi bật của mô hình. Phương pháp này giúp giảm số chiều đặc trưng từ 784 xuống còn khoảng 100–150 mà vẫn giữ được phần lớn thông tin ban đầu. Nhờ đó, mô hình không chỉ giảm được chi phí tính toán mà còn hạn chế hiện tượng quá khớp và tăng tốc độ xử lý mà không làm ảnh hưởng đến hiệu quả phân loại.

Một ưu điểm quan trọng khác là cấu trúc mô hình rất đơn giản và dễ thực hiện. Cả hai thuật toán PCA và KNN đều phổ biến, dễ hiểu, và có thể được triển khai nhanh chóng mà không cần phần cứng mạnh hay các thư viện phức tạp. Điều này đặc biệt phù hợp với các dự án nghiên cứu, giáo dục hoặc các hệ thống nhỏ cần tích hợp chức năng nhận dạng chữ số viết tay một cách hiệu quả.

**4.2.2. Hạn chế**

Mặc dù mô hình cho kết quả chính xác cao, vẫn còn một số hạn chế cần được xem xét. Trước hết, qua biểu đồ ma trận nhầm lẫn có thể thấy rằng một số chữ số như 4, 7 và 9 vẫn bị nhầm lẫn lẫn nhau trong một số trường hợp. Điều này cho thấy mô hình chưa thật sự phân biệt rõ ràng các đặc trưng hình học phức tạp giữa một số chữ số có cấu trúc tương tự.

Bên cạnh đó, biểu đồ thể hiện độ chính xác theo từng giá trị k trong thuật toán KNN cho thấy hiệu quả mô hình thay đổi đáng kể khi tham số này thay đổi. Mặc dù lựa chọn được giá trị k = 9 cho độ chính xác cao nhất, nhưng sự dao động khi k tăng lên cho thấy mô hình khá nhạy với tham số, và không ổn định nếu không được tối ưu kỹ càng. Điều này gây khó khăn khi triển khai ở môi trường thực tế hoặc với tập dữ liệu có phân bố khác biệt.

Một điểm đáng lưu ý khác là thuật toán KNN sử dụng toàn bộ tập huấn luyện khi phân loại, vì vậy chi phí tính toán sẽ tăng lên đáng kể khi triển khai trên quy mô lớn. Mặc dù PCA đã giúp giảm số chiều dữ liệu, nhưng mô hình vẫn không thực sự tối ưu về mặt tốc độ dự đoán, đặc biệt khi số lượng mẫu tăng cao.

Cuối cùng, ứng dụng triển khai nhận diện chữ viết tay bằng canvas vẫn còn hạn chế về mặt giao diện. Qua kiểm tra, vùng người dùng vẽ trên giao diện lớn hơn so với vùng mà mô hình thực sự nhận dạng, do ảnh đầu vào được thu nhỏ xuống 28x28 pixel mà không có cơ chế điều chỉnh tỷ lệ nét vẽ phù hợp. Điều này dẫn đến việc mô hình có thể nhận dạng sai hoặc bỏ sót nét vẽ nếu người dùng vẽ không đủ đậm hoặc không đúng vị trí trung tâm.

**CHƯƠNG 5. NHỮNG THÁCH THỨC VÀ HƯỚNG PHÁT TRIỂN**

Trong suốt quá trình thực hiện đề tài "Phân loại chữ số viết tay sử dụng thuật toán K-NN", nhóm đã đối mặt với nhiều thách thức mang tính kỹ thuật lẫn phương pháp luận, ảnh hưởng trực tiếp đến hiệu quả của hệ thống.

Một trong những thách thức đầu tiên là xử lý và chuẩn hóa dữ liệu đầu vào. Mặc dù tập dữ liệu MNIST đã được cung cấp ở định dạng chuẩn, việc chuyển đổi từng ảnh số (28x28 pixel) sang dạng vector một chiều để đưa vào thuật toán đòi hỏi sự cẩn trọng trong việc bảo toàn các đặc trưng thị giác quan trọng. Nếu quá trình này không được thực hiện chính xác, các đặc điểm nhận dạng đặc trưng của từng chữ số có thể bị làm mờ hoặc mất đi, dẫn đến giảm độ chính xác khi phân loại.

Tiếp theo là vấn đề hiệu năng tính toán của thuật toán K-NN. Do K-NN là một thuật toán học chậm (lazy learning), không xây dựng mô hình học rõ ràng mà lưu lại toàn bộ dữ liệu huấn luyện để sử dụng trong giai đoạn dự đoán, nên thời gian xử lý mỗi mẫu mới phụ thuộc trực tiếp vào kích thước của tập dữ liệu. Khi áp dụng vào tập MNIST – bao gồm 60.000 ảnh huấn luyện và 10.000 ảnh kiểm tra – hệ thống đòi hỏi tài nguyên bộ nhớ lớn và thời gian tính toán lâu, gây ra trở ngại nếu triển khai thực tế trong môi trường thời gian thực.

Một yếu tố khác cũng gây nhiều khó khăn là việc lựa chọn giá trị K tối ưu. Giá trị K đóng vai trò quyết định trong hiệu quả của thuật toán: K quá nhỏ khiến mô hình dễ bị ảnh hưởng bởi nhiễu (overfitting), còn K quá lớn có thể dẫn đến việc mô hình trở nên quá tổng quát, bỏ qua các đặc điểm cục bộ quan trọng. Như được chỉ ra trong phần đánh giá kết quả, sự thay đổi của độ chính xác theo từng giá trị K là đáng kể và không thể dự đoán chính xác nếu chỉ dựa trên cảm tính. Điều này buộc nhóm phải thử nghiệm thủ công nhiều lần, vừa tốn thời gian vừa thiếu hệ thống.

Bên cạnh đó, hạn chế về khả năng học trừu tượng của K-NN cũng là một thách thức lớn. Do không có quá trình huấn luyện, K-NN không học được các đặc trưng phức tạp trong dữ liệu như các mô hình học sâu, nên rất dễ nhầm lẫn khi phân loại các chữ số có nét gần giống nhau như 3 và 5, hoặc 4 và 9. Ngoài ra, mô hình bị phụ thuộc hoàn toàn vào phép đo khoảng cách (thường là Euclidean), vốn không phù hợp khi không gian đặc trưng có chiều quá cao – như trong trường hợp MNIST với 784 chiều – khiến hiệu quả phân loại bị ảnh hưởng bởi "lời nguyền chiều cao" (curse of dimensionality), làm giảm khả năng phân biệt rõ ràng giữa các lớp dữ liệu.Hướng phát triển trong tương lai

Nhằm nâng cao hiệu quả và khả năng ứng dụng của hệ thống, nhóm đề xuất một số định hướng phát triển quan trọng trong tương lai.

Thứ nhất, nhóm dự định tích hợp các kỹ thuật giảm chiều dữ liệu, điển hình là PCA (Principal Component Analysis), nhằm rút gọn số chiều đặc trưng nhưng vẫn giữ lại phần lớn thông tin quan trọng. Việc này không chỉ giảm thiểu gánh nặng tính toán mà còn hạn chế hiện tượng "lời nguyền chiều cao", giúp cải thiện độ chính xác và tốc độ xử lý của mô hình.

Thứ hai, một hướng đi mang tính chiến lược là chuyển đổi sang các mô hình học sâu, tiêu biểu như mạng nơ-ron tích chập (CNN). CNN được thiết kế chuyên biệt cho xử lý ảnh, có khả năng tự học các đặc trưng phân biệt trong dữ liệu đầu vào mà không cần tiền xử lý thủ công. Mô hình này không chỉ cải thiện đáng kể độ chính xác trên tập MNIST mà còn có tính mở rộng cao, phù hợp với các ứng dụng nhận dạng chữ viết phức tạp hơn.

Ngoài ra, để khắc phục hạn chế trong việc lựa chọn giá trị K, nhóm sẽ áp dụng các phương pháp chọn tham số tự động như Grid Search kết hợp với Cross Validation. Cách làm này giúp tìm ra giá trị K tối ưu dựa trên đánh giá hệ thống, thay vì thử nghiệm thủ công tốn thời gian và dễ thiếu khách quan.

Về mặt ứng dụng, nhóm có kế hoạch xây dựng một giao diện tương tác trực quan, cho phép người dùng vẽ chữ số viết tay trực tiếp trên màn hình và nhận kết quả phân loại ngay lập tức. Điều này không chỉ giúp kiểm chứng mô hình trong điều kiện thực tế mà còn mở rộng phạm vi ứng dụng trong các hệ thống giáo dục, số hóa tài liệu viết tay hay chấm điểm tự động.

Cuối cùng, một hướng nghiên cứu đầy tiềm năng là mở rộng mô hình sang các tập dữ liệu khác, chẳng hạn như EMNIST (chứa cả chữ cái viết tay) hoặc các bộ dữ liệu chữ viết tay Tiếng Việt. Việc này vừa kiểm nghiệm tính tổng quát của mô hình, vừa góp phần phát triển các hệ thống nhận dạng ngôn ngữ bản địa – một lĩnh vực còn ít được khai thác tại Việt Nam.

# **KẾT LUẬN**

Trong khuôn khổ đề tài "Xây dựng hệ thống phân loại hình ảnh số viết tay bằng K-NN", nhóm đã triển khai thành công một mô hình nhận dạng chữ số viết tay dựa trên thuật toán K-Nearest Neighbors (K-NN). Đây là một thuật toán học máy đơn giản nhưng hiệu quả, hoạt động dựa trên nguyên lý "láng giềng gần nhất", cho phép phân loại dữ liệu mới dựa trên sự tương đồng với các mẫu đã biết. Với tập dữ liệu MNIST kinh điển, nhóm đã thực hiện các bước từ tiền xử lý ảnh, lựa chọn tham số K phù hợp, tính toán khoảng cách, đến phân loại chính xác các chữ số viết tay.

Kết quả cho thấy thuật toán K-NN có thể đạt độ chính xác cao khi được áp dụng đúng cách, đặc biệt trong những trường hợp dữ liệu không quá lớn và đặc trưng dễ phân tách. Dù còn tồn tại một số hạn chế về tốc độ xử lý và khả năng mở rộng, K-NN vẫn chứng minh được tính khả thi khi ứng dụng vào các bài toán phân loại trong thực tế. Với khả năng triển khai nhanh, dễ hiểu và dễ giải thích, mô hình này có tiềm năng ứng dụng trong nhiều lĩnh vực như giáo dục, kiểm tra trắc nghiệm, nhận dạng mẫu viết tay và các hệ thống hỗ trợ nhập liệu số hóa.

**TÀI LIỆU THAM KHẢO**

|  |  |  |
| --- | --- | --- |
| [1] | Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. | |
| [2] | <https://www.geeksforgeeks.org/k-nearest-neighbours/> |
| [3] | <https://www.researchgate.net/publication/273124795_MNIST_handwritten_digits_Description_and_using> |

# **PHỤ LỤC**

[*Hình 1. Giao diện màn hình chính.* 11](#_Toc197365684)

[*Hình 2. Giao diện khi đưa ra kết quả dự đoán.* 11](#_Toc197365685)

[*Hình 3. Nhận diện hình không phải chữ số.* 12](#_Toc197365686)

[*Hình 4. Biểu đồ thể hiện độ chính xác với từng giá trị k.* 13](#_Toc197365687)

[*Hình 5. Biểu đồ đánh giá hiệu suất mô hình phân loại chữ số.* 14](#_Toc197365688)

[*Hình 6. Ma trận nhầm lẫn.* 15](#_Toc197365689)

[*Hình 7. Biểu đồ tỷ lệ phương sai tích lũy theo số lượng thành phần chính.* 16](#_Toc197365690)

***Mã nguồn****:*[*https://github.com/ltchieu/Handwriting-recognition-DoAnCuoiMon-DataMining.git*](https://github.com/ltchieu/Handwriting-recognition-DoAnCuoiMon-DataMining.git)

***Link video:*** *[https://youtu.be/tfkwMed9mtw?si=hqTW\_3kEZaJy1n-T](https://youtu.be/tfkwMed9mtw?si=hqTW_3kEZaJy1n-T" \t "_blank)*