**NHẬN XÉT CỦA GIẢNG VIÊN**

* Họ và tên người nhận xét: Học vị :

**NỘI DUNG NHẬN XÉT**

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

…………………………………………………………………………………....

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

TP. HCM, ngày ….. tháng …. năm 2024

**NGƯỜI NHẬN XÉT**

………………………………..

# LỜI CẢM ƠN

# MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc179805592)

[MỤC LỤC iii](#_Toc179805593)

[DANH MỤC HÌNH ẢNH v](#_Toc179805594)

[DANH MỤC TỪ VIẾT TẮT vi](#_Toc179805595)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI VÀ TIỀN XỬ LÝ DỮ LIỆU 1](#_Toc179805596)

[1.1. Tính cấp thiết/ Lý do chọn đề tài 1](#_Toc179805597)

[1.2. Mục tiêu của dự án 1](#_Toc179805598)

[1.3. Phạm vi và giới hạn của dự án 1](#_Toc179805599)

[1.4. Tổng quan về dữ liệu 2](#_Toc179805600)

[1.5. Thu thập và tiền xử lý dữ liệu 3](#_Toc179805601)

[CHƯƠNG 2: KHÁM PHÁ DỮ LIỆU VÀ LỰA CHỌN MÔ HÌNH 6](#_Toc179805602)

[2.1. Phân tích khám phá dữ liệu 6](#_Toc179805603)

[2.2. Trực quan hóa dữ liệu 8](#_Toc179805604)

[2.3. Lựa chọn mô hình 10](#_Toc179805605)

[*2.3.1. Mô hình và thuật toán được chọn* 10](#_Toc179805606)

[*2.3.2. Kỹ thuật đã sử dụng* 10](#_Toc179805607)

[*2.3.3. Quá trình chia dữ liệu* 11](#_Toc179805608)

[CHƯƠNG 3: HUẤN LUYỆN, TỐI ƯU HÓA VÀ ĐÁNH GIÁ MÔ HÌNH 12](#_Toc179805609)

[3.1. Huấn luyện mô hình 12](#_Toc179805610)

[3.2. Huấn luyện mô hình 12](#_Toc179805611)

[*3.2.1. Mô hình Random Forest* 12](#_Toc179805612)

[*3.2.2. Mô hình Mạng Nơ-ron* 14](#_Toc179805613)

[3.3. So sánh hiệu suất giữa các mô hình 16](#_Toc179805614)

[3.4. Đánh giá hiệu suất mô hình 16](#_Toc179805615)

[*3.4.1. Đánh giá Mô hình Random Forest* 16](#_Toc179805616)

[*3.4.2. Đánh giá Mô hình Mạng Nơ-ron* 17](#_Toc179805617)

[CHƯƠNG 4: TRIỂN KHAI, ỨNG DỤNG VÀ KẾT LUẬN 19](#_Toc179805618)

[4.1. Triển khai và ứng dụng mô hình 19](#_Toc179805619)

[*4.1.1. Quy trình triển khai mô hình* 19](#_Toc179805620)

[*4.1.2. Các bước kiểm tra mô hình sau khi triển khai* 19](#_Toc179805621)

[*4.1.3. Ứng dụng mô hình vào giải quyết vấn đề thực tế* 20](#_Toc179805622)

[4.2. Theo dõi và cập nhật mô hình 20](#_Toc179805623)

[*4.2.1. Theo dõi hiệu suất mô hình sau khi triển khai* 20](#_Toc179805624)

[*4.2.2. Cập nhật mô hình khi cần thiết* 20](#_Toc179805625)

[4.3. Kết luận và đề xuất 21](#_Toc179805626)

[*4.3.1. Tóm tắt những kết quả chính của dự án* 21](#_Toc179805627)

[*4.3.2. Đánh giá lại mục tiêu ban đầu và mức độ đạt được* 21](#_Toc179805628)

[*4.3.3. Đề xuất cải tiến* 21](#_Toc179805629)

[*4.3.4. Hạn chế của dự án và giải pháp tiềm năng* 21](#_Toc179805630)

[TÀI LIỆU THAM KHẢO 23](#_Toc179805631)

# DANH MỤC HÌNH ẢNH

[*Hình 1 : Hình ảnh thông tin của bộ dữ liệu Air Quality 6*](#_Toc179805312)

[*Hình 2 : Hình ảnh tổng giá trị thiếu trong bộ xử lý 7*](#_Toc179805313)

[*Hình 3 : Hình ảnh biểu đồ hộp cac nồng độ ô nhiễm 11*](#_Toc179805314)

[*Hình 4 : Hình ảnh biếu đồ phân phối CO\_level 12*](#_Toc179805315)

[*Hình 5 : Hình ảnh biểu độ phân bố nồng độ CO theo thới gian 12*](#_Toc179805316)

[*Hình 6 : Hình ảnh kết quả đánh giá mô hình Cây quyết định 16*](#_Toc179805317)

[*Hình 7 : Hình ảnh ma trận nhầm lẫn cây quyét định 16*](#_Toc179805318)

[*Hình 8 : Hình ảnh kết quả đánh giá mạng nơ-ron 18*](#_Toc179805319)

[*Hình 9 : Hình ảnh ma trận nhầm lẫn mạng nơ-ron 18*](#_Toc179805320)

[*Hình 10 : Hình ảnh so sánh hiệu suất giữa hai mô hình 19*](#_Toc179805321)

# DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Ý nghĩa đầy đủ** |
| 1 | ANN | *Artificial Neural Network* (Mạng nơ-ron nhân tạo) |
| 2 | API | *Application Programming Interface* (Giao diện lập trình ứng dụng) |
| 3 | CPU | *Central Processing Unit* (Bộ xử lý trung tâm) |
| 4 | CSV | *Comma-Separated Values* (Dữ liệu phân tách bởi dấu phẩy) |
| 5 | FPR | *False Positive Rate* (Tỷ lệ dương tính giả) |
| 6 | GPU | *Graphics Processing Unit* (Bộ xử lý đồ họa) |
| 7 | KNN | *K-Nearest Neighbors* (Phương pháp láng giềng gần nhất K) |
| 8 | ML | *Machine Learning* (Học máy) |
| 9 | RF | *Random Forest* (Rừng ngẫu nhiên) |
| 10 | RMSE | *Root Mean Square Error* (Căn bậc hai của lỗi bình phương trung bình) |

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI VÀ TIỀN XỬ LÝ DỮ LIỆU

## 1.1. Tính cấp thiết/ Lý do chọn đề tài

Trong thời đại xã hội ngày càng phát triển, hướng đến công nghiệp hóa-hiện điẹn đại hóa hiện nay, vấn đề ô nhiễm không khí đang trở thành một trong những thách thức lớn nhất mà các thành phố hiện đại phải đối mặt. Với sự gia tăng dân số và hoạt động mạnh mẽ của các khu công nghiệp, mức độ ô nhiễm không khí ngày càng nghiêm trọng, gây ra nhiều vấn đề về sức khỏe cho cộng đồng và các sinh vật sống. Theo Tổ chức Y tế Thế giới (WHO), ô nhiễm không khí là nguyên nhân gây ra khoảng 7 triệu ca tử vong mỗi năm trên toàn cầu. Nhận thất tầm quan trọng của chất lương không khí em đã quyết định lựa chọn đề tài này để phát triển nguyên cứu, việc nghiên cứu và phát triển các phương pháp để dự đoán và quản lý chất lượng không khí là rất cần thiết.

Đề tài của em nhằm mục tiêu cung cấp một công cụ dự đoán chất lượng không khí sử dụng các kỹ thuật học máy, giúp các nhà quản lý và cộng đồng có cái nhìn tổng quan về tình hình ô nhiễm. Ngoài ra dự án cũng góp phần nâng cao nhận thức về tác động của ô nhiễm không khí đến sức khỏe con người và môi trường

## 1.2. Mục tiêu của dự án

Nhận thấy tầm quan trọng của chất lượng không khí nên ngay từ lúc bắt tay thực hiện đề tài này, em đã đưa ra những mục tiêu cuối và đề tài này phải hoản thành được, cụ thể như sau:

* Em sẽ sử dụng các thuật toán học máy như Random Forest và Mạng nơ-ron để dự đoán các chỉ số ô nhiễm không khí, từ đó cung cấp thông tin hữu ích cho các nhà quản lý.
* Phân tích mối quan hệ giữa các yếu tố môi trường, khám phá các yếu tố có ảnh hưởng đến mức độ ô nhiễm không khí như nhiệt độ, độ ẩm, và các loại khí thải khác.
* Xây dựng một mô hình có khả năng tự động dự đoán. Mô hình sẽ giúp nhận diện các xu hướng ô nhiễm theo thời gian, từ đó đưa ra các giải pháp giảm thiểu ô nhiễm hiệu quả hơn.

## 1.3. Phạm vi và giới hạn của dự án

Phạm vi phân tích dữ liệu

* Đề tài này em tập trung vào việc phân tích các chỉ số như CO, NOx, NMHC, PM10 và O3. Dữ liệu sẽ được lấy từ một khu vực cụ thể, với thời gian ghi nhận từ năm 2004 đến 2005.
* Dữ liệu sẽ được thu thập từ bộ dữ liệu Air Quality UCI, một trong những nguồn dữ liệu phổ biến và đáng tin cậy trong nghiên cứu về chất lượng không khí.

Giới hạn và giả định của dự án

* Dự án chỉ phân tích dữ liệu từ một khu vực địa lý nhất định, do đó không thể đưa ra kết luận tổng quát cho toàn bộ thành phố hay quốc gia.
* Dữ liệu chỉ phản ánh tình hình ô nhiễm trong khoảng thời gian cụ thể và không thể hiện được xu hướng ô nhiễm hiện tại.
* Giả định rằng dữ liệu từ nguồn đã được xử lý và ghi nhận chính xác, mặc dù có thể tồn tại một số giá trị sai lệch hoặc thiếu sót trong quá trình thu thập.

## 1.4. Tổng quan về dữ liệu

**Nguồn dữ liệu sử dụng:**

Dữ liệu được sử dụng trong dự án được lấy từ bộ dữ liệu Air Quality UCI, được công bố bởi UCI Machine Learning Repository. Bộ dữ liệu này bao gồm các thông số đo lường chất lượng không khí từ một trạm quan trắc cụ thể trong một khoảng thời gian dài. Chính ví thế, em rất tin tưởng vào độ chính xác của bộ dữ liệu này

*Nguồn bộ dữ liệu :* [*UCI ML Air Quality Dataset (kaggle.com)*](https://www.kaggle.com/datasets/nishantbhadauria/datasetucimlairquality/data)

**Đặc điểm chính của bộ dữ liệu:**

Số lượng bản ghi: Bộ dữ liệu chứa là 9357 bản ghi

Số lượng hàng và cột : (9357, 16)

Các trường dữ liệu quan trọng trong bộ dữ liệu:

* 1. Date: Ngày ghi nhận (dạng chuỗi)
  2. Time: Thời gian ghi nhận (dạng chuỗi)
  3. CO\_level: Mức độ ô nhiễm khí CO (dạng số nguyên)
  4. NMHC\_GT: Mức độ khí NMHC (dạng số thực)
  5. C6H6\_GT: Mức độ khí C6H6 (dạng số thực)
  6. T: Nhiệt độ (dạng số thực)
  7. RH: Độ ẩm (dạng số thực)
  8. AH: Độ ẩm tuyệt đối (dạng số thực)

Loại dữ liệu :

* 1. Các cột Date, Time là dạng dữ liệu chuỗi.
  2. Các cột số khác là số nguyên hoặc số thực, chứa các giá trị đo lường từ các cảm biến.

Thông tin về bộ dữ liệu :

A screenshot of a computer

Description automatically generated

Hình 1 : Hình ảnh thông tin của bộ dữ liệu Air Quality

## 1.5. Thu thập và tiền xử lý dữ liệu

**Quy trình thu thập dữ liệu**

Dữ liệu được thu thập từ [UCI ML Air Quality Dataset (kaggle.com)](https://www.kaggle.com/datasets/nishantbhadauria/datasetucimlairquality/data) Bộ dữ liệu được tải xuống dưới định dạng CSV, giúp cho việc xử lý và phân tích dữ liệu trở nên thuận tiện hơn với các công cụ như Pandas trong Python.

Các bước tiền xử lý dữ liệu của em như sau:

1. Kiểm tra giá trị thiếu
2. Loại bỏ dữ liệu nhiễu
3. Xử lý lỗi dữ liệu

**Quá trình làm sạch dữ liệu**

\* Kiểm tra giá trị thiếu: Đầu tiên, em tiến hành kiểm tra các giá trị thiếu trong tập dữ liệu. Nếu phát hiện các trường có tỷ lệ giá trị thiếu cao, các trường này sẽ được loại bỏ hoặc xử lý. Cụ thể em xử lý qua câu lệnh sau

|  |
| --- |
| #Kiểm tra các giá trị thiếu  print(data.isnull().sum()) |

Kết quả thu được như sau :

A screenshot of a test

Description automatically generated

Hình 2 : Hình ảnh tổng giá trị thiếu trong bộ xử lý

\* Loại bỏ dữ liệu nhiễu: Các bản ghi có giá trị không hợp lệ hoặc nằm ngoài giới hạn hợp lý (chẳng hạn như giá trị âm trong các trường đo lường ô nhiễm) sẽ bị loại bỏ.

\* Xử lý lỗi dữ liệu: Các lỗi trong dữ liệu, chẳng hạn như sự không nhất quán trong định dạng ngày tháng hoặc các giá trị ghi nhận sai, sẽ được sửa chữa hoặc loại bỏ.

**Bước tiền xử lý dữ liệu**

\* Chuyển đổi định dạng thời gian: Các cột Date và Time được kết hợp thành một cột Datetime và chuyển đổi sang định dạng datetime để thuận tiện cho việc phân tích theo thời gian.

|  |
| --- |
| *# Chuyển đổi cột Date và Time thành định dạng datetime*  data['Datetime'] = pd.to\_datetime(data['Date'] + ' ' + data['Time']) |

\* Chuẩn hóa dữ liệu: Tất cả các giá trị số được chuẩn hóa (scaling) để đưa về cùng một thang đo, giúp mô hình học máy hoạt động hiệu quả hơn.

|  |
| --- |
| *# Chuẩn hóa dữ liệu*  scaler = StandardScaler()  features = data.drop(['CO\_level', 'Datetime'], axis=1)  features\_scaled = scaler.fit\_transform(features) |

\* Mã hóa dữ liệu: Các cột phân loại được mã hóa thành dạng số để phù hợp với định dạng đầu vào của các mô hình học máy.

|  |
| --- |
| *# Mã hóa cột CO\_level thành số nguyên*  le = LabelEncoder()  data['CO\_level'] = le.fit\_transform(data['CO\_level']) |

\* Chia tách dữ liệu: Dữ liệu được chia thành hai phần: tập huấn luyện (80%) và tập kiểm tra (20%) để đánh giá hiệu suất của mô hình.

|  |
| --- |
| *# Chia dữ liệu thành tập huấn luyện và tập kiểm tra*  X\_train, X\_test, y\_train, y\_test = train\_test\_split(features\_scaled, data['CO\_level'], test\_size=0.2, random\_state=42) |

# CHƯƠNG 2: KHÁM PHÁ DỮ LIỆU VÀ LỰA CHỌN MÔ HÌNH

## 2.1. Phân tích khám phá dữ liệu

Phân tích khám phá dữ liệu (EDA) là bước đầu tiên quan trọng trong quá trình phân tích dữ liệu, giúp chúng ta hiểu rõ hơn về cấu trúc và đặc điểm của tập dữ liệu trước khi tiến hành xây dựng mô hình. Trong đề tài này em xin trình bày các thông tin sau khi khai phá bộ dữ liệu Air Quality

\* Danh sách các cột trong dữ liệu :

Bảng 1 : Bảng cột dữ liệu trong bộ dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên cột** | **Giải thích ý nghĩa cột** |
| 1 | Date | Ngày đo |
| 2 | Time | Thời gian đo |
| 3 | CO\_GT | Nồng độ Carbon Monoxide (CO) |
| 4 | PT08\_S1\_CO | Chỉ số cảm biến CO |
| 5 | NMHC\_GT | Nồng độ các hợp chất hữu cơ không bão hòa |
| 6 | C6H6\_GT | Nồng độ Benzene (C6H6) |
| 7 | PT08\_S2\_NMHC | Chỉ số cảm biến NMHC |
| 8 | Nox\_GT | Nồng độ Nitrogen Oxide (NOx) |
| 9 | PT08\_S3\_Nox | Chỉ số cảm biến NOx |
| 10 | NO2\_GT | Nồng độ Nitrogen Dioxide (NO2) |
| 11 | PT08\_S4\_NO2 | Chỉ số cảm biến NO2 |
| 12 | PT08\_S5\_O3 | Chỉ số cảm biến Ozone (O3) |
| 13 | T | Nhiệt độ |
| 14 | RH | Độ ẩm tương đối |
| 15 | AH | Độ ẩm tuyệt đối |
| 16 | CO\_level | Mức độ CO |

\* Thống kê mô tả

Sau khi thực hiện phân tích, em đã thu thập được các thống kê mô tả cho từng biến trong tập dữ liệu. Bảng dưới đây tổng hợp các thông tin này

|  |
| --- |
| *# Code hiển thị các thống kê cơ bản của dữ liệu*  print("\nThống kê mô tả của dữ liệu:")  print(data.describe |

A screenshot of a computer

Description automatically generated

\* Số lượng hàng và cột

Tập dữ liệu có tổng cộng 9357 hàng và 16 cột, cho thấy đây là một tập dữ liệu lớn và phong phú về thông tin, có thể cung cấp cái nhìn sâu sắc về chất lượng không khí trong một khoảng thời gian

\* Thông tin chung về dữ liệu

Dữ liệu được tổ chức dưới dạng DataFrame trong Pandas, giúp em dễ dàng thực hiện các phân tích tiếp theo. Dưới đây là thông tin chi tiết về cấu trúc của DataFrame

|  |
| --- |
| <class 'pandas.core.frame.DataFrame'>  RangeIndex: 9357 entries, 0 to 9356  Data columns (total 16 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 Date 9357 non-null object  1 Time 9357 non-null object  2 CO\_GT 9357 non-null float64  3 PT08\_S1\_CO 9357 non-null int64  4 NMHC\_GT 9357 non-null int64  5 C6H6\_GT 9357 non-null float64  6 PT08\_S2\_NMHC 9357 non-null int64  7 Nox\_GT 9357 non-null int64  8 PT08\_S3\_Nox 9357 non-null int64  9 NO2\_GT 9357 non-null int64  10 PT08\_S4\_NO2 9357 non-null int64  11 PT08\_S5\_O3 9357 non-null int64  12 T 9357 non-null float64  13 RH 9357 non-null float64  14 AH 9357 non-null float64  15 CO\_level 9357 non-null object  dtypes: float64(5), int64(8), object(3)  memory usage: 1.1+ MB |

## 2.2. Trực quan hóa dữ liệu

Trực quan hóa dữ liệu là một công cụ mạnh mẽ giúp em có thể phát hiện các mẫu, xu hướng và ngoại lệ trong dữ liệu. Trong đề tài này em đã sử dụng nhiều loại biểu đồ khác nhau để trình bày kết quả

\* Biểu đồ hộp (Box Plot) :

Biểu đồ hộp được sử dụng để đánh giá sự phân bố của các biến liên quan đến nồng độ chất ô nhiễm. Nó giúp em phát hiện các giá trị ngoại lai và so sánh phân bố giữa các biến

Kết quả thu được :

A diagram of a diagram

Description automatically generated with medium confidence

Hình 3 : Hình ảnh biểu đồ hộp cac nồng độ ô nhiễm

\* Sử dụng biểu đồ phân tán (Scatter Plot)

Biểu đồ phân tán được sử dụng để phân tích mối quan hệ giữa các biến. Em đã sử dụng biểu đồ phân tán để xem xét mối quan hệ giữa nồng độ CO và NO2

A graph of a graph with blue lines

Description automatically generated with medium confidence

Hình 4 : Hình ảnh biếu đồ phân phối CO\_level

\* Biểu đồ đường (Line Plot)

Biểu đồ đường có thể được sử dụng để theo dõi sự thay đổi nồng độ chất ô nhiễm theo thời gian, giúp em hiểu rõ hơn về xu hướng theo thời gian, dưới đây là kếta quả thu được sau khi phân tích

A graph with blue lines

Description automatically generated

Hình 5 : Hình ảnh biểu độ phân bố nồng độ CO theo thới gian

## 2.3. Lựa chọn mô hình

Trong nghiên cứu này, em đã lựa chọn đồng thời cả hai mô hình học máy: Random Forest và Mạng nơ-ron (Neural Network). Cả hai mô hình này được sử dụng để phân tích dữ liệu chất lượng không khí, với mục tiêu chính là dự đoán nồng độ ô nhiễm CO từ các đặc trưng khác trong dữ liệu

### ***2.3.1. Mô hình và thuật toán được chọn***

Random Forest

* Khái niệm : Là một phương pháp học máy thuộc loại ensemble learning, Random Forest kết hợp nhiều cây quyết định để đưa ra dự đoán. Mỗi cây trong rừng được xây dựng từ một mẫu dữ liệu ngẫu nhiên, và kết quả cuối cùng được tính toán bằng cách lấy trung bình hoặc chế độ của các cây
* Lý do lựa chọn : Mô hình này có khả năng xử lý các dữ liệu lớn và đa dạng, đồng thời chống lại hiện tượng overfitting nhờ vào tính ngẫu nhiên trong cách xây dựng các cây. Random Forest cũng dễ dàng trong việc xác định độ quan trọng của các biến, giúp em hiểu rõ hơn về yếu tố ảnh hưởng đến nồng độ ô nhiễm.

Mạng nơ-ron

* Khái niệm cơ bản : Mạng nơ-ron là một mô hình học sâu, mô phỏng cấu trúc và chức năng của não người. Nó bao gồm nhiều lớp nơ-ron, mỗi lớp học các đặc trưng khác nhau từ dữ liệu đầu vào thông qua các hàm kích hoạt.
* Lý do em lựa chọn: Mạng nơ-ron có khả năng phát hiện các mối quan hệ phi tuyến tính trong dữ liệu, điều này rất hữu ích cho các bài toán phức tạp như dự đoán nồng độ ô nhiễm. Thêm vào đó, tính linh hoạt trong việc điều chỉnh cấu trúc mạng cho phép tối ưu hóa mô hình cho từng bài toán cụ thể.

### ***2.3.2. Kỹ thuật đã sử dụng***

Em đã áp dụng các kỹ thuật sau trong quá trình xây dựng mô hình:

\* Tiền xử lý dữ liệu: Bao gồm việc loại bỏ các giá trị không hợp lệ, chuẩn hóa dữ liệu và mã hóa các biến phân loại. Việc này giúp mô hình học máy hoạt động hiệu quả hơn và tăng cường độ chính xác của dự đoán.

\* Chia dữ liệu: Dữ liệu được chia thành ba tập:

1. Tập huấn luyện: Chiếm 80% dữ liệu, dùng để huấn luyện mô hình.
2. Tập kiểm tra: Chiếm 20% dữ liệu còn lại, dùng để đánh giá hiệu suất của mô hình sau khi đã huấn luyện.
3. Tập xác nhận: Mặc dù không được tách riêng trong quá trình nghiên cứu này, nhưng em có thể sử dụng tập kiểm tra để điều chỉnh các tham số mô hình và thực hiện kiểm tra chéo

### ***2.3.3. Quá trình chia dữ liệu***

Quá trình chia dữ liệu được thực hiện như sau:

1. Tập dữ liệu ban đầu: Dữ liệu được đọc từ file CSV và được xử lý để đảm bảo không có giá trị thiếu hay không hợp lệ.
2. Chia dữ liệu: Sử dụng hàm train\_test\_split từ thư viện sklearn, dữ liệu được chia thành tập huấn luyện và tập kiểm tra với tỉ lệ 80/20. Em đã sử dụng tham số random\_state để đảm bảo tính ngẫu nhiên và khả năng tái lập của quá trình chia dữ liệu.

|  |
| --- |
| # Chia dữ liệu thành tập huấn luyện và tập kiểm tra  X\_train, X\_test, y\_train, y\_test = train\_test\_split(features\_scaled, data['CO\_level'], test\_size=0.2, random\_state=42) |

1. Tiền xử lý trước khi chia: Trước khi chia, các giá trị được chuẩn hóa bằng cách sử dụng StandardScaler, giúp đưa tất cả các đặc trưng về cùng một thang đo, làm cho mô hình dễ học hơn.

|  |
| --- |
| # Chuẩn hóa dữ liệu  scaler = StandardScaler()  features = data.drop(['CO\_level', 'Datetime'], axis=1)  features\_scaled = scaler.fit\_transform(features) |

# CHƯƠNG 3: HUẤN LUYỆN, TỐI ƯU HÓA VÀ ĐÁNH GIÁ MÔ HÌNH

## 3.1. Huấn luyện mô hình

Trong chương này, em sẽ trình bày chi tiết quá trình huấn luyện, tối ưu hóa và đánh giá hai mô hình học máy: Random Forest và Mạng Nơ-ron. Mục tiêu là tối ưu hóa độ chính xác và khả năng tổng quát của mô hình trong bài toán phân loại. Chúng tôi sẽ mô tả chi tiết từng bước, bao gồm các tham số mô hình, quá trình huấn luyện, kết quả đánh giá, và sự so sánh giữa hai mô hình.

## 3.2. Huấn luyện mô hình

### ***3.2.1. Mô hình Random Forest***

\* Giới thiệu về Random Forest

Random Forest là một thuật toán học máy thuộc nhóm cây quyết định, hoạt động dựa trên nguyên lý "tập hợp" (ensemble). Thuật toán này xây dựng nhiều cây quyết định trong quá trình huấn luyện và đưa ra dự đoán bằng cách kết hợp các kết quả từ tất cả các cây. Một trong những lợi ích chính của Random Forest là khả năng giảm thiểu hiện tượng quá khớp (overfitting) thường gặp ở các mô hình cây quyết định đơn lẻ

\*Huấn luyện mô hình : em triển khai mô hình Random Forest với các tham số sau:

* + n\_estimators: 100 (số lượng cây quyết định trong rừng).
  + random\_state: 42 (để đảm bảo tính tái lập)

Quá trình huấn luyện mô hình được thực hiện bằng lệnh sau

|  |
| --- |
| from sklearn.ensemble import RandomForestClassifier  from sklearn.metrics import classification\_report, confusion\_matrix  rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)  rf\_model.fit(X\_train, y\_train)  rf\_predictions = rf\_model.predict(X\_test) |

\* Đánh giá Mô hình Random Forest : Mô hình được đánh giá dựa trên các chỉ số như sau

1. Độ chính xác (Accuracy): Tỉ lệ dự đoán đúng trên tổng số mẫu.
2. Độ nhạy (Recall): Tỉ lệ đúng của các mẫu dương tính.
3. Độ chính xác (Precision): Tỉ lệ các mẫu dương tính dự đoán đúng.
4. F1-score: Trung bình điều hòa của độ nhạy và độ chính xác.

Các chỉ số này được tính toán như sau:

|  |
| --- |
| print("Random Forest Classification Report")  rf\_report = classification\_report(y\_test, rf\_predictions, output\_dict=True)  print(classification\_report(y\_test, rf\_predictions))  print("Confusion Matrix")  rf\_confusion = confusion\_matrix(y\_test, rf\_predictions)  print(rf\_confusion) |

Kết quả đánh giá đạt được

A black text on a white background

Description automatically generated

Hình 6 : Hình ảnh kết quả đánh giá mô hình Cây quyết định

Ma trận Nhầm lẫn: Ma trận nhầm lẫn của mô hình Random Forest thu được như sau

A graph with numbers and a number on it

Description automatically generated

Hình 7 : Hình ảnh ma trận nhầm lẫn cây quyét định

### ***3.2.2. Mô hình Mạng Nơ-ron***

\* Giới thiệu về Mạng Nơ-ron

Mạng nơ-ron là một mô hình học sâu (deep learning) có khả năng tự học từ dữ liệu thông qua các kết nối giữa các nơ-ron. Mạng nơ-ron được cấu trúc thành nhiều lớp: lớp đầu vào, lớp ẩn, và lớp đầu ra. Em đã sử dụng hàm kích hoạt ReLU (Rectified Linear Unit) cho các lớp ẩn và hàm sigmoid cho lớp đầu ra

\* Huấn luyện mô hình : Cấu trúc mô hình mạng nơ-ron được thiết kế như sau

* 1. Lớp đầu vào: 32 nơ-ron với hàm kích hoạt ReLU.
  2. Lớp ẩn: 16 nơ-ron với hàm kích hoạt ReLU.
  3. Lớp đầu ra: 1 nơ-ron với hàm kích hoạt sigmoid.

\* Quá trình huấn luyện mô hình được thực hiện bằng câu lệnh sau

|  |
| --- |
| from keras.models import Sequential  from keras.layers import Dense  nn\_model = Sequential()  nn\_model.add(Dense(32, input\_dim=X\_train.shape[1], activation='relu'))  nn\_model.add(Dense(16, activation='relu'))  nn\_model.add(Dense(1, activation='sigmoid'))  nn\_model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])  nn\_model.fit(X\_train, y\_train, epochs=50, batch\_size=10, validation\_data=(X\_test, y\_test)) |

\* Dự đoán và đánh giá mô hình

|  |
| --- |
| nn\_predictions = (nn\_model.predict(X\_test) > 0.5).astype("int32")  print("Neural Network Classification Report")  nn\_report = classification\_report(y\_test, nn\_predictions, output\_dict=True)  print(classification\_report(y\_test, nn\_predictions))  print("Confusion Matrix")  nn\_confusion = confusion\_matrix(y\_test, nn\_predictions)  print(nn\_confusion) |

\* Kết quả đánh giá

A screenshot of a computer

Description automatically generated

Hình 8 : Hình ảnh kết quả đánh giá mạng nơ-ron

\* Ma trận Nhầm lẫn

A diagram of a test

Description automatically generated with medium confidence

Hình 9 : Hình ảnh ma trận nhầm lẫn mạng nơ-ron

## 3.3. So sánh hiệu suất giữa các mô hình

Sau khi hoàn thành quá trình huấn luyện và đánh giá cho cả hai mô hình, em tiến hành so sánh hiệu suất giữa chúng. Để minh họa rõ hơn, biểu đồ dưới đây so sánh độ chính xác của Random Forest và Mạng Nơ-ron

A blue and orange rectangles

Description automatically generated

Hình 10 : Hình ảnh so sánh hiệu suất giữa hai mô hình

Kết quả so sánh :

* 1. Độ chính xác của Random Forest: 97.81%
  2. Độ chính xác của Neural Network: 90.71%

## 3.4. Đánh giá hiệu suất mô hình

Để đánh giá hiệu suất mô hình, em sẽ sử dụng tập dữ liệu kiểm tra để đánh giá các mô hình đã huấn luyện, gồm Random Forest và Mạng Nơ-ron, nhằm kiểm tra khả năng phân loại của chúng trên các mẫu chưa từng được học trước đó. Các chỉ số đo lường hiệu suất sẽ được sử dụng bao gồm Độ chính xác (Accuracy), Độ nhạy (Recall), Độ chính xác (Precision), và F1-score. Nếu có nhiều mô hình được thử nghiệm em sẽ tiến hành so sánh hiệu suất giữa các mô hình và chọn ra mô hình tốt nhất

### ***3.4.1. Đánh giá Mô hình Random Forest***

Các chỉ số đo lường hiệu suất được tính toán dựa trên các dự đoán của mô hình trên tập kiểm tra và nhãn thực tế:

* Độ chính xác (Accuracy): Phản ánh tỷ lệ các dự đoán đúng trên tổng số dự đoán.
* Độ nhạy (Recall): Phản ánh khả năng phát hiện chính xác các mẫu thuộc lớp dương tính.
* Độ chính xác (Precision): Tỷ lệ các dự đoán đúng trong số các mẫu được dự đoán là dương tính.
* F1-score: Trung bình điều hòa của độ nhạy và độ chính xác, là chỉ số tổng hợp để đánh giá hiệu suất mô hình.

Kết quả đánh giá cho Random Forest

Các kết quả đánh giá chi tiết cho mô hình Random Forest như sau:

1. Độ chính xác (Accuracy): 97.81%
2. Độ nhạy (Recall): 100%
3. Độ chính xác (Precision): 95.83%
4. F1-score: 97.87%

Ma trận nhầm lẫn được sử dụng để hiển thị các kết quả phân loại:

|  |  |  |
| --- | --- | --- |
|  | **Dự đoán 0** | **Dự đoán 1** |
| **Thực tế 0** | 140 | 7 |
| **Thực tế 1** | 0 | 36 |

Dựa trên các chỉ số trên, có thể thấy rằng mô hình Random Forest đạt hiệu suất cao với khả năng dự đoán chính xác hầu hết các mẫu trong tập kiểm tra

### ***3.4.2. Đánh giá Mô hình Mạng Nơ-ron***

Mô hình mạng nơ-ron cũng được đánh giá bằng cách tính toán các chỉ số đo lường tương tự. Để cải thiện mô hình em đã thử nghiệm với nhiều cấu trúc khác nhau và điều chỉnh các tham số để tối ưu hóa độ chính xác

Kết quả Đánh giá cho Mạng Nơ-ron : các kết quả đánh giá chi tiết cho mô hình mạng nơ-ron như sau:

* Độ chính xác (Accuracy): 90.71%
* Độ nhạy (Recall): 100%
* Độ chính xác (Precision): 57.50%
* F1-score: 73.02%

Ma trận nhầm lẫn của mô hình mạng nơ-ron:

|  |  |  |
| --- | --- | --- |
|  | **Dự đoán 0** | **Dự đoán 1** |
| **Thực tế 0** | 143 | 4 |
| **Thực tế 1** | 12 | 11 |

Dựa trên các kết quả trên có thể thấy mô hình mạng nơ-ron có độ nhạy cao (khả năng phát hiện các mẫu dương tính tốt), nhưng độ chính xác lại thấp hơn so với Random Forest, cho thấy mô hình có xu hướng phân loại nhầm các mẫu thuộc lớp âm tính

Bảng dưới đây tổng hợp các chỉ số đo lường hiệu suất cho cả hai mô hình để so sánh

|  |  |  |
| --- | --- | --- |
| **Chỉ số** | **Random Forest** | **Mạng Nơ-ron** |
| Độ chính xác | 97.81% | 90.71% |
| Độ nhạy | 100% | 100% |
| Độ chính xác | 95.83% | 57.50% |
| F1-score | 97.87% | 73.02% |

Nhận xét từ bảng số liệu :

* Random Forest đạt được độ chính xác cao hơn (97.81% so với 90.71%) và F1-score tốt hơn (97.87% so với 73.02%). Điều này cho thấy mô hình Random Forest có hiệu suất tốt hơn trong việc dự đoán đúng các mẫu và phân loại chính xác hơn.
* Mạng nơ-ron tuy có độ nhạy cao nhưng độ chính xác lại thấp hơn, cho thấy mô hình có thể cần được tối ưu hóa thêm, chẳng hạn như điều chỉnh các siêu tham số hoặc sử dụng các kỹ thuật cải tiến để giảm thiểu lỗi

Dựa trên các kết quả và so sánh hiệu suất ở trên, em nhận thấy rằng mô hình “Random Forest” được chọn là mô hình tốt nhất cho bài toán này vì đạt được độ chính xác và F1-score cao nhất. Mô hình này không chỉ có khả năng dự đoán chính xác hầu hết các mẫu mà còn hạn chế sai sót trong việc phân loại

# CHƯƠNG 4: TRIỂN KHAI, ỨNG DỤNG VÀ KẾT LUẬN

## 4.1. Triển khai và ứng dụng mô hình

Quá trình triển khai mô hình học máy bao gồm các bước từ việc đưa mô hình vào môi trường thực tế hoặc giả lập, kiểm tra tính chính xác sau khi triển khai, đến việc ứng dụng mô hình vào việc giải quyết các vấn đề thực tế

### ***4.1.1. Quy trình triển khai mô hình***

Bảng 2 : Bảng quy trình triển khai mô hình

|  |  |  |
| --- | --- | --- |
| **STT** | **Quy trình** | **Nội dung chi tiết** |
| 1 | Chuẩn bị môi trường triển khai | Để triển khai mô hình trong môi trường thực tế, cần thiết lập một máy chủ hoặc dịch vụ điện toán đám mây với các công cụ cần thiết như Python, thư viện học máy (scikit-learn, TensorFlow), và các gói xử lý dữ liệu (numpy, pandas) |
| 2 | Triển khai mô hình | Mô hình đã được huấn luyện (Random Forest và Neural Network) sẽ được lưu dưới dạng file .pkl (đối với Random Forest) và .h5 (đối với Neural Network). Sau đó, các mô hình này sẽ được tải lên và sử dụng để dự đoán |
| 3 | Giao diện người dùng | Một ứng dụng web hoặc giao diện lập trình ứng dụng (API) có thể được xây dựng để người dùng có thể nhập dữ liệu mới và nhận kết quả dự đoán trực tiếp, em dự tính đây sẽ là hướng phát triển thêm trong lương lai |

### ***4.1.2. Các bước kiểm tra mô hình sau khi triển khai***

Sau khi mô hình được triển khai, cần thực hiện các bước kiểm tra sau để đảm bảo tính chính xác và khả năng hoạt động ổn định

Kiểm tra chức năng cơ bản : Nhập dữ liệu mẫu và kiểm tra kết quả dự đoán từ cả hai mô hình (Random Forest và Neural Network). Ví dụ, với đoạn mã em đã cung cấp sẵn như sau :

|  |
| --- |
| # Dữ liệu ví dụ để kiểm tra  new\_data\_example = [150, 136, 2.3, 180, 13.4, 9.0, 0.9, 1200, 34.5, 1.2, 0.7, 300, 27]  predict\_new\_data(new\_data\_example) |

Kết quả trả về thu được là:

A close up of a message

Description automatically generated

Kiểm tra độ chính xác: Sử dụng một tập dữ liệu kiểm tra độc lập để đánh giá lại mô hình và so sánh với kết quả trước khi triển khai.

Kiểm tra tính ổn định: Chạy thử các kịch bản khác nhau và kiểm tra xem kết quả có thay đổi lớn hay không khi dữ liệu đầu vào bị nhiễu.

### ***4.1.3. Ứng dụng mô hình vào giải quyết vấn đề thực tế***

Ứng dụng trong dự báo phân loại: Các mô hình Random Forest và Neural Network có thể được ứng dụng để phân loại tình trạng của một sản phẩm hoặc dịch vụ dựa trên các đặc điểm đầu vào. Ví dụ, nếu đầu vào là các chỉ số về chất lượng sản phẩm, mô hình sẽ dự đoán trạng thái chất lượng như "High" (cao) hoặc "Low" (thấp).

Ví dụ cụ thể em đưa ra như sau : Một công ty sản xuất có thể sử dụng mô hình để dự đoán chất lượng của sản phẩm trước khi đưa vào giai đoạn kiểm tra chất lượng cuối cùng, giúp tối ưu hóa quy trình sản xuất và giảm chi phí

## 4.2. Theo dõi và cập nhật mô hình

Để đảm bảo mô hình luôn đạt hiệu suất cao, cần có quy trình theo dõi và cập nhật mô hình định kỳ

### ***4.2.1. Theo dõi hiệu suất mô hình sau khi triển khai***

Sử dụng các chỉ số đo lường (độ chính xác, F1-score) để theo dõi mô hình trong thời gian thực. Nếu hiệu suất giảm dưới mức chấp nhận được, cần xem xét lại mô hình hoặc tập dữ liệu.

Các phản hồi từ người dùng có thể cung cấp thông tin về các trường hợp mô hình dự đoán sai hoặc không chính xác, giúp cải thiện mô hình. Vì thế em gợi ý nên thu thập ý kiến phản hồi của người dùng thường xuyên

### ***4.2.2. Cập nhật mô hình khi cần thiết***

Nếu có thêm dữ liệu mới hoặc mô hình cũ không còn đáp ứng yêu cầu, cần huấn luyện lại mô hình với dữ liệu mới.

Thực hiện tối ưu hóa các siêu tham số (hyperparameters) của mô hình để cải thiện hiệu suất.

Nếu các mô hình hiện tại không đáp ứng được yêu cầu, có thể thử nghiệm các mô hình khác hoặc các phiên bản nâng cấp hơn.

## 4.3. Kết luận và đề xuất

### ***4.3.1. Tóm tắt những kết quả chính của dự án***

Trong dự án này, em đã nguyên cứu phát triển và triển khai thành công hai mô hình học máy, gồm Random Forest và Neural Network, để dự đoán và phân loại dữ liệu chất lượng không khí với độ chính xác cao. Mô hình Random Forest đạt hiệu suất tốt hơn với độ chính xác là 97.81%, trong khi Neural Network đạt 90.71%.

### ***4.3.2. Đánh giá lại mục tiêu ban đầu và mức độ đạt được***

Mục tiêu ban đầu là xây dựng một mô hình học máy có khả năng phân loại chính xác các mẫu dữ liệu, và kết quả đạt được đã vượt qua kỳ vọng với mô hình Random Forest. Các kết quả cho thấy mô hình có thể ứng dụng vào thực tế để dự báo phân loại trong các hệ thống sản xuất hoặc các lĩnh vực liên quan.

### ***4.3.3. Đề xuất cải tiến***

Bảng 3 : Bảng các đề xuất phát triển đề tài

|  |  |  |
| --- | --- | --- |
| **STT** | **Đề xuất cải tiến** | **Nội dung chi tiết** |
| 1 | Cải thiện dữ liệu đầu vào | Tăng cường chất lượng dữ liệu bằng cách loại bỏ nhiễu, tăng số lượng mẫu dữ liệu, và bổ sung thêm các đặc trưng quan trọng |
| 2 | Tối ưu hóa mô hình | Thử nghiệm các kỹ thuật học sâu hơn, như Deep Neural Network, hoặc các phương pháp ensemble khác |
| 3 | Triển khai thêm các kỹ thuật cải tiến | Sử dụng các phương pháp như tăng cường dữ liệu (data augmentation), hoặc học chuyển tiếp (transfer learning) để cải thiện hiệu suất |

.

### ***4.3.4. Hạn chế của dự án và giải pháp tiềm năng***

Hạn chế gặp phải: Chất lượng và số lượng dữ liệu

* Vấn đề em gặp phải:
  1. Chất lượng dữ liệu không đồng nhất, có nhiễu, hoặc thiếu dữ liệu.
  2. Số lượng dữ liệu ít có thể dẫn đến overfitting.
* Giải pháp có thể giải quyết:
  1. Tiền xử lý dữ liệu: làm sạch, xử lý giá trị bị thiếu, chuẩn hóa đặc trưng.
  2. Loại bỏ nhiễu bằng các thuật toán học máy.
  3. Thu thập thêm dữ liệu từ nhiều nguồn.
  4. Tăng cường dữ liệu: sử dụng kỹ thuật Data Augmentation hoặc dữ liệu tổng hợp.

Hạn chế : Độ phức tạp của mô hình

* Vấn đề:

1. Tối ưu hóa mô hình phức tạp khó khăn, mất nhiều thời gian.
2. Nguy cơ overfitting hoặc underfitting nếu chọn sai kiến trúc mạng.

* Giải pháp khắc phục :

1. Tối ưu hóa siêu tham số: sử dụng Random Search, Grid Search.
2. Thử nghiệm các mô hình khác như Gradient Boosting hoặc LightGBM.
3. Áp dụng kỹ thuật Regularization để giảm overfitting.

# TÀI LIỆU THAM KHẢO

[1]. Goodfellow, I., Bengio, Y., Courville, A. - *Deep Learning* – MIT Press, 2016.

[2]. Chollet, F. - *Deep Learning with Python* – Manning Publications, 2018.

[3]. Géron, A. - *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* – O'Reilly Media, 2019.

[4]. Brownlee, J. - *Deep Learning for Time Series Forecasting* – Machine Learning Mastery, 2020.

[5]. Luật Công nghệ Thông tin số 67/2006/QH11 ngày 29 tháng 6 năm 2006 quy định về quản lý và sử dụng công nghệ thông tin tại Việt Nam.

[6]. Luật An ninh mạng số 24/2018/QH14 ngày 12 tháng 6 năm 2018 quy định về bảo mật thông tin và an ninh mạng.

[7]. Đỗ Văn An - *Phân tích và đánh giá mô hình học máy trong phân loại chất lương không khí*– Luận văn thạc sĩ, Trường Đại học Bách Khoa Hà Nội, năm 2020.

[8]. Các trang web học máy và khoa học dữ liệu:

* <http://www.tensorflow.org>
* <http://www.scikit-learn.org>
* [UCI ML Air Quality Dataset (kaggle.com)](https://www.kaggle.com/datasets/nishantbhadauria/datasetucimlairquality/data)

[9]. Tài liệu nội bộ Công ty TNHH DataTech - *Báo cáo nghiên cứu và triển khai mô hình học máy* năm 2023.