Contents

[1. Tổng quan về Artificial Intelligence 3](#_Toc146634033)

[1.1. Khái niệm 3](#_Toc146634034)

[AI nghiên cứu cách bộ não con người suy nghĩ, cách con người học hỏi, quyết định, giải quyết 3](#_Toc146634035)

[1.2. Phân loại 3](#_Toc146634036)

[1.2.1. Phân loại theo khả năng ( Based on capacility) 3](#_Toc146634037)

[1.2.2. Phân loại theo chức năng ( Based on functionaility) 4](#_Toc146634038)

[1.3. AI hoạt động như thế nào 4](#_Toc146634039)

[1.4. Điều kiện để AI phát triển 4](#_Toc146634040)

[1.5. Lĩnh vực con của AI 5](#_Toc146634041)

[2. Tổng quan về Machine Learning 6](#_Toc146634042)

[2.1. Khái niệm 6](#_Toc146634043)

[2.2. Ứng dụng 6](#_Toc146634044)

[2.3. Quy trình của Machine Learning 7](#_Toc146634045)

[2.3.1. Thu thập dữ liệu 7](#_Toc146634046)

[2.3.2. Chuẩn bị dữ liệu (Tiền xử lý dữ liệu) 8](#_Toc146634047)

[2.3.3. Huấn luyện mô hình 8](#_Toc146634048)

[2.3.4. Đánh giá mô hình 8](#_Toc146634049)

[2.3.5. Triển khai mô hình 8](#_Toc146634050)

[2.4. Phân loại Machine Learning 8](#_Toc146634051)

[2.4.1. Supervised Learning 9](#_Toc146634052)

[2.4.3. Semi-supervised Learning 11](#_Toc146634053)

[2.4.4. Reinforcement Learning 14](#_Toc146634054)

[2.5. Deep Learning 15](#_Toc146634055)

[3. Bài toán ứng dụng 15](#_Toc146634056)

[3.1. Mô tả bài toán 15](#_Toc146634057)

[3.2. Chuẩn bị dữ liệu ( Tiền xử lý dữ liệu) 16](#_Toc146634058)

[3.2.2. Lựa chọn đặc trung 24](#_Toc146634059)

[3.2.3. Xử lý outlier 25](#_Toc146634060)

[3.2.4. Chuẩn hóa dữ liệu 29](#_Toc146634061)

[3.2.5. Chia dữ liệu huấn luyện 29](#_Toc146634062)

[3.3. Huấn luyện và đánh giá mô hình 30](#_Toc146634063)

[3.3.1. Mô hình Linear Regression 30](#_Toc146634064)

# 1. Tổng quan về Artificial Intelligence

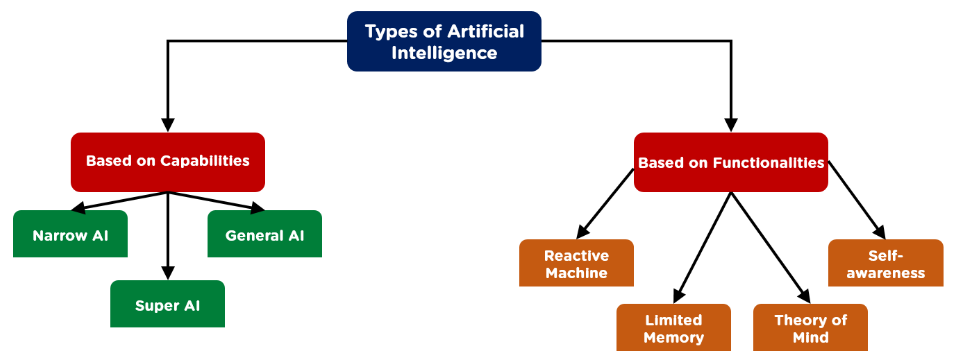
## 1.1. Khái niệm

Artificial Intelligence (AI) là một nhánh của Computer Science theo đuổi việc tạo ra những máy tính hoặc máy móc thông minh như con người.

AI là ngành khoa học và kỹ thuật chế tạo máy móc thông minh, các chương trình máy móc thông minh (McCarthy).

AI nghiên cứu cách bộ não con người suy nghĩ, cách con người học hỏi, quyết định, giải quyết vấn đề, sử dụng kinh nghiệm này làm cơ sở phát triển phần mềm và hệ thống thông minh.

### 1.2. Phân loại

****

## 1.2.1. Phân loại theo khả năng ( Based on capacility)

* **Weak AI / Narrow AI ( Artificial Narrow Intelligence) – Machine Learning**

ANI là một loại AI thực hiện một nhiệm vụ hoặc một nhóm nhiệm vụ cụ thể, không tự nhận thức và sở hữu trí thông minh như con người, giới hạn trong domain hẹp. VD: virtual assistant, recommendation system, facial recognition, fraud detection, Chatbots, ,…

* **Strong AI / General AI ( Artificial General Intelligence) – Machine Intelligence**

AGI là một loại AI có khả năng học hỏi thấu hiểu bất kỳ nhiệm vụ trí tuệ nào mà con người làm được. Hiện nay chúng ta đang nghiên cứu loại AI này. VD: OpenAI’s GPT- 4 (SOTA language model), DeepMind’s AlphaGo and AlphaZero, Self-driving car, Virtual Assistants with Multi-Domain Understanding, Robotics,..

* **Super AI ( Artificial Super Intelligence) – Machine Consciousness**

ASI là một loại AI vượt qua trí thông minh con người, phát triển giống với cảm xúc và trải nghiệm của con người hoặc của riêng nó bao gồm khả năng suy nghĩ, đưa ra phán đoán và quyết định.

### 1.2.2. Phân loại theo chức năng ( Based on functionaility)

* **Reactive Machine ( máy phản ứng)**

Là một loại AI không lưu trữ bộ nhớ hoặc kinh nghiệm trong quá khứ để xác định hành động trong tương lai, hoạt động trên dữ liệu hiện tại và được cung cấp nhiệm vụ cụ thể. Machine Learning model có xu hướng là Reactive Machine.

VD: Siêu máy tính IBM Deep Blue – hệ thống AI chơi cờ vua có thể xác định quân cờ của đối thủ và của nó để đưa ra dự đoán nước đi nhưng không có khả năng ghi nhớ các nước đi sai lầm trong quá khứ để đưa ra quyết định tương lai.

* **Limited Memory (Bộ nhớ giới hạn)**

Là một loại AI bắt chước tế bào thần kinh trong não, thông minh hơn khi nhận được nhiều data để huấn luyện ( Deep Learning, Reinforcement Learning), có thể đào tạo dữ liệu trong quá khứ để đưa ra quyết định, theo dõi các đối tượng hoặc tình huống cụ thể theo thời gian.

VD: Self-driving car quan sát các xe khác trên đường về tốc độ, hướng di chuyển và khoảng cách, nhiều thông tin khác như đèn giao thông, biển báo, khúc cua để giúp xe quyết định tình huống.

* **Theory of mind ( Thuyết tâm trí)**

Là một loại AI tương tác với suy nghĩ và cảm xúc như con người, hiểu được ý định và dự đoán hành vi mô phỏng các mối quan hệ của con người trong xã hội.

* **Self-awareness ( Tự nhận thức)**

Là một loại AI có ý thức về bản thân nó, ý thức được sự tồn tại của chúng.Loại AI này vượt xa lý thuyết về AI, không chỉ hiểu và gợi lên cảm xúc ở người nó tương tác mà còn có cảm xúc, nhu cầu và niềm tin của riêng mình ( chỉ tồn tại trên lý thuyết).

## 1.3. AI hoạt động như thế nào

* AI hoạt động dựa trên kết hợp các tập dữ liệu lớn với các thuật toán xử lý lặp đi lặp lại để học cấu trúc ẩn và tính năng trong dữ liệu.
* Một hệ thống AI xử lý dữ liệu -> kiểm tra đo lường hiệu suất -> phát triển hệ thống.
* AI không bao giờ nghỉ ngơi -> học được nhiều trong thời gian ngắn.
* Hệ thống AI sử dụng các kỹ thuật và quy trình công nghệ khác nhau để giải quyết các.mục tiêu phức tạp khác nhau -> dựa vào đây để hiểu AI thực sự làm gì, cách thức hoạt động của nó.

## 1.4. Điều kiện để AI phát triển

* Dữ liệu khổng lồ, lớn hơn, dễ truy cập hơn nhờ sự phát triển của IoT tạo ra lượng dữ liệu lớn từ các thiết bị được kết nối -> AI có nhiều ứng dụng tiềm năng hơn.
* GPUs cung cấp sức mạnh điện toán để AI xử lý và diễn giải dữ liệu lớn.
* Các thuật toán mới và tiên tiến cho phép các hệ thống AI phân tích dữ liệu nhanh hơn.
* Giao diện lập trình ứng dụng API cho phép thêm các chức năng AI vào các ứng dụng phần mềm và chương trình máy tính truyền thống.

## 1.5. Lĩnh vực con của AI

Một số lĩnh vực của AI:

* **Machine Learning**: cho phép hệ thống máy tính tự động học hỏi và phát triển tốt hơn dựa trên kinh nghiệm, không được lập trình để làm. ML giúp AI tìm các mẫu trong dữ liệu khám phá thông tin và cải thiện kết quả của bất kỳ nhiệm vụ mà hệ thống đặt ra.
* **Deep Learning**: là lĩnh vực con của ML cho phép AI khả năng học và cải thiện bằng cách xử lý dữ liệu. DL sử dụng các mạng thần kinh nhân tạo bắt chước mạng thần kinh trong não người để đưa ra suy luận, tìm mối liên hệ giữa dữ liệu.
* **Natural Language Processing ( NLP):** là một phần quan trọng trong quy trình của AI cho phép computer nhận dạng, phân tích diễn giải và thực sự hiểu ngôn ngữ con người, kể cả viết và nói -> quan trọng trong hệ thống AI tương tác với con người thông qua đầu vào text hoặc speech. VD: email filter, virtual assistant, search engine, translate engine, speech regconition,…
* **Computer Vision**: lĩnh vực con của AI cho phép máy tính giải thích phân tích thế giới trực quan , mô phỏng cách con người nhìn và hiểu môi trường.CV áp dụng các mô hình Deep Learning để phân loại và xác định các đối tượng trong hình ảnh và video kĩ thuật số sau đó cho phép máy tính phản ứng với những gì chúng nhìn thấy.

VD: facial regconition, object detection, self-driving car, robotics, medical diagnosis, production line error detection,…

* **Expert Systems**: là lĩnh vực của AI bao gồm các ứng dụng máy tính được phát triển để giải quyết các vấn đề phức tạp trong 1 lĩnh vực cụ thể, ở mức độ thông minh và chuyên môn phi thường của con người. Expert Systems có khả năng đưa ra lời khuyên, hỗ trợ con người đưa ra quyết định. Các Expert System bao gồm 3 thành phần chính là Knowledge Base, Inference Engine, User Interface.

VD:

Design Domain: ống kính camera, sản xuất ô tô,..

Medical Domain: hệ thống chẩn đoán bệnh,..

Supervise Domain: phát hiện rò rỉ trong đường ống dẫn dầu,..

Knowledge Domain: tìm lỗi ở thiết bị, phương tiện giao thông, máy tính,..

Finance Domain: phát hiện gian lận giao dịch đáng ngờ, lập lịch hàng không, vận chuyển,…

* **Robotics**: lĩnh vực AI tạo ra robot, tác nhân nhân tạo hoạt động trong môi trường thực tế có cấu trúc cơ khí được thiết kế cho hoạt động cụ thể, giúp con người thực hiện các chức năng lặp đi lặp lại, nhiệm vụ khó và làm việc thông minh hiệu quả với input là analog signal ( speech waveform, images,…)

VD:

Industry Domain: robot cắt vật liệu, đào hầm, nâng vật nặng, dây chuyền lắp ráp,..

Military Domain: robot tự hành do thám

Medical Domain: robot thực hiện xét nghiệm lâm sàng, phẫu thuật,…

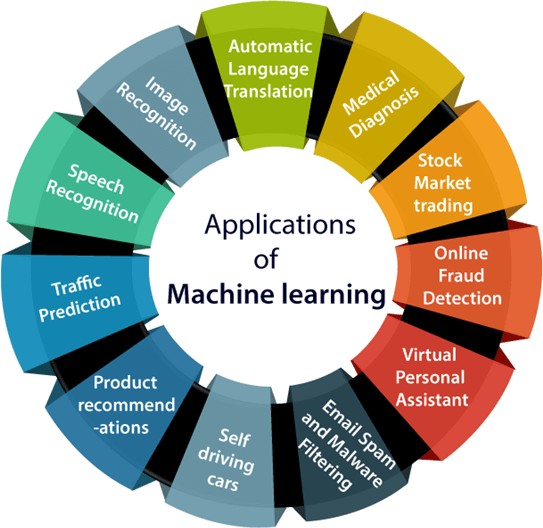
Exploration Domain: robot khám phá trong môi trường nguy hiểm hoặc khắc nghiệt,..

# 2. Tổng quan về Machine Learning

## 2.1. Khái niệm

Machine Learning (ML) là một lĩnh vực con của AI liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép hệ thống học tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Các thuật toán Machine Learning xây dựng mô hình dựa trên dữ liệu mẫu gọi là dữ liệu huấn luyện để đưa ra dự đoán hoặc quyết định mà không cần được lập trình rõ ràng.

## 2.2. Ứng dụng

Machine Learning đang được ứng dụng trong lĩnh vực thị giác máy tính, nhận diện giọng nói, lọc thư điện tử, hệ thống gợi ý, tránh gian lận, dự đoán,…

## 2.3. Quy trình của Machine Learning

Quy trình tiếp cận truyền thống của Machine Learning sẽ áp dụng các thuật toán thuộc nhóm supervised, unsupervised, semi-supervised, reinforcement tùy thuộc vào loại dữ liệu và bài toán cần giải quyết. Bước quan trọng trong cách tiếp cận này ngoài việc thu thập lượng dữ liệu lớn thì còn phải trích chọn các đặc trưng (feature extraction) phù hợp với dữ liệu, sau đó xây dựng một thuật toán tối ưu để tạo được mô hình tốt. Ngoài ra còn có bước tiền xử lý sau giai đoạn thu thập dữ liệu nhằm làm sạch dữ liệu.

### 2.3.1. Thu thập dữ liệu

* Để giải quyết bài toán Machine learning chúng ta cần bao nhiêu dữ liệu là đủ? Không có con số chính xác nhưng số lượng càng nhiều thì sẽ càng tốt. Để có cái nhìn đúng đắn nhất về số lượng dữ liệu khi model đã được huấn luyện (train) và được đánh giá hiệu năng (evaluate performance) trên các dữ liệu mới (unseen data).
* Dữ liệu cần phải liên quan đến bài toán, ít mất mát và trùng lặp, có thể thu thập từ nhiều nguồn và tích hợp với nhau, chất lượng dữ liệu và sự đa dạng của dữ liệu rất quan trọng.
* Phụ thuộc vào độ phức tạp của vấn đề và thuật toán sử dụng. VD: Nhận dạng ảnh và xử lý ngôn ngữ tự nhiên cần tập dữ liệu lớn bởi độ phức tạp vấn đề
* Phụ thuộc vào độ phức tạp của mô hình. VD: Mô hình càng phức tạp càng nhiều tham số -> cần nhiều dữ liệu ( các mô hình ensemble cần nhiều dữ liệu vì sử dụng đa mô hình kết hợp với nhau)
* Phụ thuộc vào chất lượng và tính chính xác của dữ liệu. VD: Giả sử có nhiều nhiễu và thông tin không đúng trong dữ liệu -> cần thiết phải tăng kích thước bộ dữ liệu để tăng độ chính xác cho mô hình.
* Ưóc lượng dữ liệu: Cách tiếp cận rule of thumb cần ít nhất 10 lần observation so với số lượng feature trong dataset.

### 2.3.2. Chuẩn bị dữ liệu (Tiền xử lý dữ liệu)

* Dữ liệu cần phải tiền xử lý trước khi training model
* Khám phá và phân tích dữ liệu ( Exploratory Data Analysis)
* Tìm hiểu thông tin về dataset, số lượng cá thể (instance), số đặc trưng (feature), phân phối (distribution) của đặc trưng, mối quan hệ giữa các independent variable với nhau và với dependent variable.
* Kiểm tra vấn đề missing data, imbalanced data, outlier.
* Để nhìn nhận một cách trực quan các vấn đề ta sẽ sử dụng các biểu đồ.
* Mã hóa categorical variable sang numerical variable
* Chuẩn hóa numerical variable
* Feature engineering, feature extraction, feature selection để chọn ra đặc trưng có giá trị nhất cho mô hình
* Chia dữ liệu thành các tập huấn luyện và kiểm thử

### 2.3.3. Huấn luyện mô hình

* Lựa chọn thuật toán phù hợp cho bài toán để huấn luyện mô hình
* Sau quá trình kiểm thử nếu mô hình không tốt ta có thể cải thiện bằng cách thay đổi các siêu tham số của mô hình (hyperparameter). Mỗi mô hình ML đều có những tham số đặc trưng để thay đổi -> hyperparameter tuning.

### 2.3.4. Đánh giá mô hình

Tính toán, đánh giá kết quả, độ chính xác của mô hình cuối cùng trên tập dữ liệu test, tính quan trọng của thuộc tính trong mô hình, chi phí vận hành để từ đó quyết định xây dựng lại và cải thiện mô hình với các bước trên hay triển khai mô hình.

### 2.3.5. Triển khai mô hình

Đưa mô hình vào thực tế, đánh giá lại mô hình liên tục để xây dựng và cải thiện mô hình.

## 2.4. Phân loại Machine Learning

Machine learning có thể được chia thành 4 phương thức học chính:

### 2.4.1. Supervised Learning

**A diagram of a diagram

Description automatically generated**

Mô hình Supervised learning là một loại máy học mà máy được đào tạo bằng cách sử dụng dữ liệu đào tạo được gán nhãn và trên cơ sở đó máy dự đoán đầu ra. Dữ liệu được gán nhãn là dữ liệu đầu vào đã được gán với đầu ra chính xác đã biết.

Supervised learning được chia làm 2 loại là mô hình Regression và mô hình Classification:

* **Regression:** được sử dụng để mô tả mối quan hệ giữa biến độc lập và phụ thuộc. Mô hình dự đoán giá trị liên tục của biến phụ thuộc dựa vào một hay nhiều biến độc lập.

VD: Dự đoán lượng mưa dựa trên nhiệt độ, áp suất không khí, thời gian, sức gió,…

Các thuật toán chính trong regression là hồi quy tuyến tính ( Linear regression) và hồi quy đa thức ( Polynomial regression).

* Linear regresson là một thuật toán regression sử dụng để mô hình hóa mối quan hệ giữa biến độc lập và phụ thuộc, giả định mối quan hệ tuyến tính giữa biến độc lập và phụ thuộc để tìm một hàm phù hợp mô tả mối quan hệ đó.
* Polynomial regression là một thuật toán regression mô hình hóa mối quan hệ giữa biến độc lập và phụ thuộc dưới dạng hàm đa thức bậc n. Thuật toán giúp diễn giải mối quan hệ phi tuyến tính trong dữ liệu, mô hình chuyển đổi đầu vào sang dạng bậc đa thức -> bậc đa thức càng lớn mô hình sẽ càng khớp với dữ liệu nhưng số lượng tham số tăng lên - > mô hình phức tạp gây ra overfitting.
* **Classification:** được sử dụng để dự đoán lớp của biến phụ thuộc dựa trên các biến độc lập, biến phụ thuộc là giá trị rời rạc.

VD: Phân loại thư rác -> thư được phân loại spam/not spam

Một số thuật toán phân loại cơ bản như Logistic regression, K-nearest-neighbor, Support Vector Machine,..

* Logistic regression là thuật toán phân loại nhị phân dự đoán lớp cho biến phụ thuộc là binary, thuật toán được sử dụng để mô tả dữ liệu và giải thích mối quan hệ giữa biến nhị phân phụ thuộc với một hay nhiều biến độc lập bằng cách ước tính xác suất của kết quả nhị phân. Logistic regression sử dụng logistic function để biến đổi các kết hợp tuyến tính của các biến độc lập, làm cho nó phù hợp để dự đoán xác suất và phân loại dữ liệu thành hai lớp dựa trên threshold. Có 3 loại logistic regression: Binary Logistic Regression, Multinomial Logistic Regression, Oridinal Logistic Regression
* K-nearest-neighbor là một thuật toán phân loại quan sát mới bằng cách tìm điểm tương đồng giữa quan sát mới với dữ liệu sẵn có dựa trên k điểm quan sát gần nhất.
* Support Vector Machine là một thuật toán phân loại dữ liệu bằng cách tìm một siêu phẳng phân chia tốt nhất mà tối đa hóa lề xung quanh siêu phẳng sao cho khoảng cách giữa các lớp là lớn nhất.
* Ngoài ra mô hình Classificaion còn sử dụng các thuật toán tree-based như Decision Tree,các thuật toán ensemble như Gradient Boosting hay Random Forest và sử dụng neural network như thuật toán Neural Network hay Perceptron là cơ sở cho mô hình Deep Learning.

**2.4.2. Unsupervised Learning**

**A diagram of a process

Description automatically generated**

Mô hình Unsupervised learning được huấn luyện với dữ liệu không được gán nhãn, các thuật toán tự mô hình hóa cấu trúc ẩn trong dữ liệu để mô tả tính chất hay đặc tính của dữ liệu.

Các thuật toán Unsupervised learning dựa vào các mục đích khác nhau để phân loại bao gồm:

* **Phân cụm ( Clustering):** là các thuật toán gom dữ liệu vào các nhóm khác nhau sao cho mỗi điểm dữ liệu tương đồng với các điểm dữ liệu khác trong cùng nhóm.

Các thuật toán Clustering được chia ra nhiều loại như:

* **Density-based clustering**: nhóm các thuật toán dựa vào mật độ dữ liệu, dữ liệu được nhóm theo các khu vực có mật độ dữ liệu cao được bao quanh bởi các khu vực có mật độ dữ liệu thấp -> về cơ bản thuật toán tìm những nơi có nhiều điểm dữ liệu gom lại thành cụm. Các thuật toán phân cụm này bỏ qua outlier không gán vào trong cụm.

VD: DBSCAN clustering

* **Distribution-based clustering**: Với cách tiếp cận phân cụm dựa trên phân phối, tất cả các điểm dữ liệu được coi là một phần của cụm dựa trên xác suất chúng thuộc về một cụm nhất định.

VD: Gaussian Mixture Model algorithm

* **Centroid-based clustering**: Các loại thuật toán này phân tách các điểm dữ liệu dựa trên nhiều tâm cụm trong dữ liệu. Mỗi điểm dữ liệu được gán cho một cụm dựa trên bình phương khoảng cách của nó với tâm cụm. Đây là kiểu phân cụm được sử dụng phổ biến nhất.

VD: K-Means clustering

* **Hierarchical-based clustering**: Phân cụm dựa trên phân cấp thường được sử dụng trên dữ liệu phân cấp, giống như từ cơ sở dữ liệu công ty hoặc phân loại. Nó xây dựng một cây gồm các cụm để mọi thứ được sắp xếp từ trên xuống.

VD: Mean-Shift clustering

* **Luật kết hợp ( Association rule)** : luật kết hợp là một loại unsupervised learning nhằm kiểm tra sự phụ thuộc của một mục dữ liệu vào một mục dữ liệu khác, cố gắng tìm mối quan hệ hoặc liên kết giữa các biến của tập dữ liệu, thường được sử dụng trong các bài toán transaction.

VD: Apriori

* **Giảm chiều dữ liệu ( Dimensionality Reduction) :** là phương pháp biến đổi dữ liệu từ không gian chiều cao thành không gian chiều thấp đồng thời giữ lại một số thuộc tính có ý nghĩa của dữ liệu gốc giúp mô hình không bị quá phức tạp.

VD: Singular value decompostion (SVD), principal component analysis ( PCA)

### 2.4.3. Semi-supervised Learning

Là một loại học máy nằm giữa supervised learning và unsupervised learning, sử dụng lượng nhỏ dữ liệu được gán nhãn và lượng lớn dữ liệu không nhãn để huấn luyện mô hình. Mô hình tính toán bằng cách giả định những mẫu không gán nhãn có nhãn bằng nhiều cách và thuật toán như giả định liên tục (Continuity Assumption), giả định cụm (Cluster Assumption), giả định đa điểm (Manifold Asumption).

Một số kĩ thuật semi-supervised learning:

* **Self-training**: kĩ thuật tự đào tạo là quy trình mà có thể sử dụng bất kì phương pháp supervised nào để classification và điều chỉnh để hoạt động trong semi-supervised learning.

VD: Speech regconition của Facebook, việc gán nhãn speech data tốn thời gian công sức nên việc sử dụng semi-supervised ( self- training) bắt đầu với các mô hình cơ bản supervised learning để training 100 giờ các file speech của con người được gán nhãn, sau đó 500 giờ speech data không được gán nhãn để tăng hiệu suất mô hình.

A diagram of a self-training method

Description automatically generated

Cách hoạt động của self-training:

* Lượng nhỏ dữ liệu được gán nhãn đưa vào huấn luyện mô hình supervised learning
* Quy trình pseudo-labeling, sử dụng mô hình đã huấn luyện để predict cho dữ liệu không nhãn
* Chọn ra dự đoán có confidence cao hơn ngưỡng nào đó ( ví dụ bạn muốn confidence > 90% rằng một hình ảnh nhất định hiển thị một con mèo). Nếu bất kì pseudo label nào vượt qua ngưỡng này sẽ được thêm vào tập dữ liệu có nhãn để tiếp tục huấn luyện.
* Quá trình lặp lại nhiều lần với nhiều pseudo label được thêm vào tập dữ liệu huấn luyện để tăng độ chính xác.
* **Co-training**: kĩ thuật được cải tiến từ self-training, hoạt động bằng cách đào tạo đồng thời 2 bộ phân loại dựa trên 2 view của dữ liệu, view là các bộ đặc trưng khác nhau cung cấp thông tin về mỗi thực thể, độc lập với class.

VD: Nhiệm vụ phân loại nội dung trang web thì mô tả của mỗi trang web chia thành 2 view bao gồm các từ xuất hiện trong trang web và các từ anchor được gắn đường dẫn đến trang khác.

A diagram of a training method

Description automatically generated

Cách hoạt động của co-training:

* Sử dụng 2 mô hình phân loại độc lập để huấn luyện dữ liệu gán nhãn.
* Quy trình pseudo-labeling để dự đoán pseudo-label cho dữ liệu không nhãn từ 2 mô hình vừa huấn luyện.
* Chọn ra pseudo-label có confidence cao hơn ngưỡng nào đó, nếu mô hình thứ nhất dự đoán 1 pseudo-label có confidence cao hơn ngưỡng nhưng mô hình 2 dự đoán pseudo-label đó có confidence thấp hơn ngưỡng thì dữ liệu không nhãn có pseudo-label đó ở mô hình thứ nhất sẽ được thêm vào dữ liệu huấn luyện của mô hình 2 và ngược lại.
* Kết hợp kết quả dự đoán của 2 mô hình phân loại để có kết quả cuối cùng.
* Lặp lại nhiều lần và thêm vào tập dữ liệu có nhãn ban đầu để tăng độ chính xác.

Một số thuật toán sử dụng trong semi-supervised learning như FixMatch, MixMatch,…

### 2.4.4. Reinforcement Learning

**A diagram of a diagram of a business

Description automatically generated**

Học tăng cường là một kĩ thuật học máy trong đó 1 tác nhân ( agent) học cách phản hồi trong một môi trường ( enviroment) bằng cách thực hiện các hành động ( action) và xem phần thưởng ( reward) của các hành động đó. Với mỗi good action, tác nhân nhận được phản hồi tích cực và ngược lại.

Trong reinforcement learning agent học tự động từ phản hồi mà không có dữ liệu huấn luyện, học từ kinh nghiệm khi agent tương tác với enviroment xem action nào dẫn đến positive reward và negative reward.

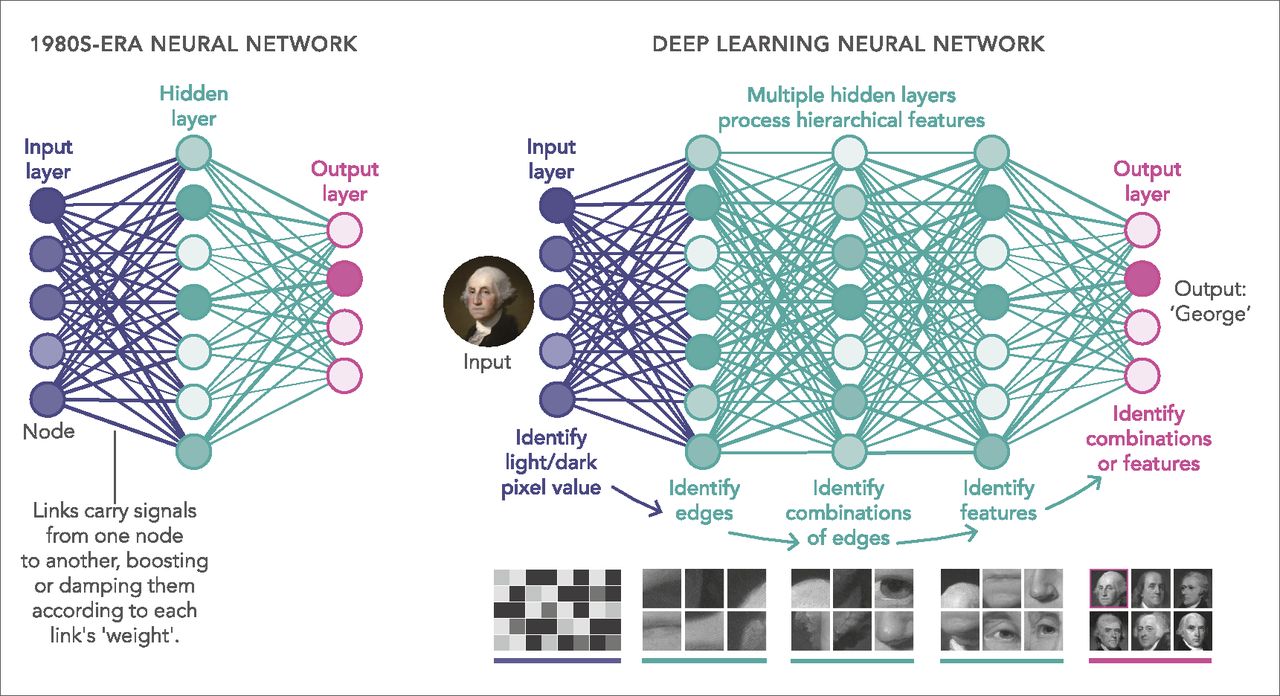
Mục tiêu của agent trong reinforcement learning là cải thiện hiệu suất bằng việc lấy được positive reward lớn nhất bằng cách tối ưu policy.

Reinforcement Learning gồm 4 thành phần chính:

* **Policy**: cách agent hành động tại 1 thời điểm nhất định
* **Reward signal**: Ở mỗi state ( trạng thái), enviroment gửi reward signal đến agent. Agent nhận được reward signal tùy theo good hoặc bad action. Reward signal có thể thay đổi policy vì bad action dẫn đến low reward -> thay đổi policy để chọn action khác trong tương lai.
* **Value function**: hàm giá trị cho biết thông tin về mức độ tốt của tình huống và action cũng như reward mà agent có thể có.
* **Model**: mô hình bắt chước action của enviroment. Model giúp dự đoán về cách enviroment hành động. VD: Nếu agent ở 1 state và đưa ra 1 action thì model sẽ dự đoán state tiếp theo và reward.

Một số thuật toán sử dụng trong Reinforcement learning như Q-learning, Deep Q- Networks, SARSA algorithm, DDPG algorithm,…

## 2.5. Deep Learning



Deep Learning là một lĩnh vực con của Machine Learning. Deep Learning là phương thức học máy phức tạp hơn với mạng nơ ron (Neural Network), mô phỏng giống với cách bộ não con người tư duy và kết luận mà không cần đến nhiều sự can thiệp của con người trong quá trình học như Machine Learning.

Deep Learning hoạt động trên nhiều mạng lưới thần kinh gồm ba lớp trở lên và cố gắng mô phỏng hành vi của bộ não con người. Nó cho phép học từ lượng lớn dữ liệu và giải thích các xu hướng trong dữ liệu.

Deep learning được ứng dụng trong thị giác máy tính, phân tích giọng nói, sinh văn bản, hệ thống lái xe tự động,...

# 3. Bài toán ứng dụng

## 3.1. Mô tả bài toán

Khí thải CO2 từ phương tiện là một trong những nguyên nhân chính gây ô nhiễm môi trường, việc kiểm soát lượng khí thải CO2 từ phương tiện do đó đã gợi lên mối quan tâm toàn cầu trong nghiên cứu ứng dụng trong lĩnh vực phân tích dữ liệu và Machine Learning. Việc ước tính và trực quan mức tiêu thụ nhiên liệu và khí thải là rất quan trọng đối với định lượng chi phí năng lượng và ô nhiễm do giao thông vận tải gây ra. Với mong muốn xây dựng một mô hình có thể dự đoán lượng khí thải CO2 từ phương tiện và biết được những tác nhân quan trọng trong việc gây ra lượng khí thải CO2 để sẵn sàng cho một nền tảng phương tiện tốt hơn qua đó làm giảm khí thải CO2.

Bộ dữ liệu trong bài toán này được tổng hợp ở Canada trong vòng 7 năm từ 2014-2021 ghi lại chi tiết về lượng khí thải CO2 của các phương tiện và các thông tin kĩ thuật về phương tiện đó bao gồm 7385 phương tiện và 12 đặc trưng.

Đây là bài toán regression với đầu ra cần dự đoán là CO2 Emission(g/km) có giá trị liên tục. Các đặc trưng của dữ liệu trong bài toán đề cập ở dưới đây:

|  |  |
| --- | --- |
| **Đặc trưng** | **Ý nghĩa** |
| Make | Hãng xe |
| Model | Mẫu xe |
| Vehicle Class | Phân khúc xe |
| Engine Size(L) | Dung tích động cơ |
| Cylinders | Xilanh động cơ |
| Tranmission | Hộp số truyền động |
| Fuel Type | Loại nhiên liệu |
| Fuel Consumption City (L/100km) | Nhiên liệu tiêu thụ trong đô thị (L/100km) |
| Fuel Consumption Hwy (L/100km) | Nhiên liệu tiêu thụ ngoài đô thị (L/100km) |
| Fuel Consumption Comb (L/100km) | Nhiên liệu tiêu thụ hỗn hợp (L/100km) |
| Fuel Consumption Comb (mpg) | Nhiên liệu tiêu thụ hỗn hợp (dặm trên gallon) |
| CO2 Emission(g/km) | Lượng khí thải CO2 ước tính |

## 3.2. Chuẩn bị dữ liệu ( Tiền xử lý dữ liệu)

**3.2.1. Phân tích khám phá dữ liệu ( Trực quan hóa dữ liệu)**

Trực quan hóa các categorical variables với mỗi giá trị categories duy nhất và tần suất xuất hiện của chúng. Categorical variables bao gồm 5 đặc trưng phân loại bao gồm Make, Model, Vehicle Class, Transmission, Fuel Type.

A graph of a number of people

Description automatically generated with medium confidence

**A graph of different colored bars

Description automatically generated with medium confidence**

**A graph of a bar graph

Description automatically generated with medium confidence**

**A graph of a bar graph

Description automatically generated with medium confidence**

**A graph with different colored bars

Description automatically generated**

Các categorical variables được trực quan với tần số xuất hiện của các giá trị categories duy nhất được sắp xếp theo thứ tự giảm dần. Trong kết quả thống kê này ta thấy được Ford là hãng xe phổ biến nhất cùng với mẫu xe F-150 FFV 4x4 của hãng, SUV-SMALL là phân khúc xe được lựa chọn nhiều nhất, hộp số tự động xuất hiện nhiều nhất ở các phương tiện, xăng thông thường là loại nhiên liệu phổ biến trong bộ dữ liệu. Lý giải cho việc này là Ford là hãng xe nổi tiếng ở các nước Bắc Mỹ như USA và Canada và mẫu F-150 FFV theo khảo sát là mẫu xe yêu thích nhất ở Canada, hộp số tự động xuất hiện ở phần lớn phương tiện có thể thấy được từ sự cải tiến của nó so với hộp số sàn- dễ sử dụng, thuận tiện và tiết kiệm nhiên liệu và nhiều lợi ích khác. Xăng vẫn là loại nhiên liệu phổ biến bởi lượng năng lượng lớn khi đốt cháy, với 1 lượng nhỏ nhiên liệu có thể tạo ra năng lượng lớn. Từ trực quan hóa ta thấy ở đặc trưng Fuel Type với giá trị categories Natural Gas có 1 dữ liệu duy nhất và ngoài thực tế loại nhiên liệu này hầu như không được được sử dụng trên phương tiện bởi vì sự bất lợi của nó so với các loại nhiên liệu khác nên ta sẽ loại bỏ dữ liệu này.

**A graph of a graph

Description automatically generated with medium confidence**

**A graph with a blue line

Description automatically generated**

**A graph of a curve

Description automatically generated with medium confidence**

**A graph with a line going up

Description automatically generated**

**A graph of a graph

Description automatically generated with medium confidence**

**A graph with a line going up

Description automatically generated**

**A graph with a line going up

Description automatically generated**

Các đặc trưng numerical và biến đầu ra được biểu diễn sự phân bố và phân tán của dữ liệu với biểu đồ histogram với mật độ của dữ liệu. Từ biểu đồ cho các đặc trưng numerical ta thấy dữ liệu bị positive skew lệch phải ở hầu hết các đặc trưng qua đó dữ liệu không phải phân phối chuẩn, từ phân phối positive skew đoán được rằng các giá trị bất thường tập trung ở phía giá trị max của đặc trưng và dữ liệu không phân bố đều trên tập dữ liệu. Kiểm tra skewness measure cho các đặc trưng numerical cho thấy tất cả bị positive skew, skewness measure nằm từ 0.52 đến 1.11 với đặc trưng Cylinders bị skew nhiều nhất (1.11) và skew ít nhất là biến target CO2 Emissions. Skewness measure nằm trong khoảng -0.5 đến 0.5 được cho là phân phối đối xứng, từ 0.5 đến 1 và -1 đến -0.5 là positively skew và negatively skew, lớn hơn 1 và bé hơn -1 dữ liệu bị highly skew. Dữ liệu skew ảnh hưởng đến chất lượng của mô hình như Linear Regression giả định dữ liệu phân phối chuẩn từ đó ta xem xét xử lý các bất thường có phải là outlier do lỗi đo lường không hay chỉ là outlier tự nhiên.

**A graph of a graph of cylinders

Description automatically generated with medium confidence**

**A graph of different colored bars

Description automatically generated**

**A graph of a graph

Description automatically generated with medium confidence**

**A graph of a graph

Description automatically generated with medium confidence**

**A graph of a graph

Description automatically generated with medium confidence**

**A graph of a bar graph

Description automatically generated with medium confidence**

Ta sẽ trực quan lượng CO2 trung bình của các đặc trung numerical và xem xét các tác nhân gây ra lượng khí thải nhiều hay ít. Nhìn một cách trực quan, số lượng xilanh càng lớn và kích thước động cơ càng lớn xe càng thải nhiều CO2 bởi vì thực tế những xe này sẽ đốt cháy nhiên liệu nhanh hơn tiêu thụ nhiều nhiên liệu hơn tạo ra nhiều sức mạnh cho xe nhưng sẽ thải ra nhiều khí thải hơn. Lượng tiêu thụ nhiên liệu trong đô thị và ngoài đô thị càng tăng thì càng thải ra nhiều CO2 bởi vậy nhiên liệu tiêu thụ hỗn hợp tăng thì sẽ tăng khí thải. Các đặc trưng này có một mối liên kết chặt chẽ với nhau đã được dự đoán trước.

**A graph with different colored bars

Description automatically generated with medium confidence**

**A chart of different colored rectangles

Description automatically generated**

**A green red and orange squares

Description automatically generated**

**A green red and orange squares

Description automatically generated**

Ta sẽ trực quan lượng CO2 trung bình của các đặc trưng categorical xem xét các tác nhân gây ra lượng khí thải nhiều hay ít. Trong đặc trưng Make ta thấy được các xe phổ thông giá rẻ thải ra CO2 trung bình ít hơn các xe sang và đắt tiền giải thích cho điều này trong bộ dữ liệu vì các xe sang và đắt tiền tiêu thụ nhiên liệu nhiều hơn và kích thước động cơ lớn nhiều xilanh từ đó tiêu thụ nhiều nhiên liệu, trong đặc trưng Vehicle Class các phân khúc xe Van và Truck thải nhiều khí thải nhất cho thấy kích thước xe lớn sẽ cần kích thước động cơ lớn từ đó tiêu thụ nhiều nhiên liệu và thải nhiều CO2 . Trong thực tế, xe sử dụng hộp số tự động sẽ tiêu thụ nhiên liệu nhiều hơn từ đó thải nhiều CO2 , tuy nhiên có sự bất thường trong đặc trưng Fuel Type khi nhiên liệu Ethanol thực tế thải ra ít CO2 trong khi lại thải nhiều CO2 nhất. Lý giải cho điều này là khi tìm hiểu trong bộ dữ liệu thì các xe sử dụng nhiên liệu Ethanol này chiếm phần lớn trong các tác nhân của xe thải nhiều CO2 nhất ví dụ như hộp số tự động, phân khúc xe, hãng xe , số lượng xilanh lớn, kích thước động cơ lớn, tiêu thụ nhiên liệu nhiều,.. bằng cách xem xét tỷ lệ trong các tác nhân gây ra nhiều khí thải nhất ( lớn hơn 80% dữ liệu của đặc trưng đó).

Từ kết quả trực quan này và kiến thức thực tế ta có cơ sở để feature engineering đặc trưng Make, Vehicle Class nhóm các giá trị vào cùng nhóm tương đồng về lượng khí thải CO2 bởi vì các đặc trưng này có số lượng categories duy nhất lớn khi mã hóa sẽ tạo ra số lượng đặc trưng lớn nên cần thiết để giảm số đặc trưng sau mã hóa. Trong bài toán, đặc trưng Make được nhóm lại từ 42 categories về 4 categories dựa trên hãng xe phổ thông, xe sang, xe thể thao hay luxury và căn cứ vào lượng khí thải tương tự với Vehicle Class có 16 categories nhóm về 4 categories. Riêng đặc trưng Transmission có 27 categories nhóm về 5 categories dựa và tiền tố của categories đó ( các loại hộp số M5,M6,M7 nhóm về categories mới là Manual, viết tắt của M là Manual nghĩa là hộp số sàn).

### 3.2.2. Lựa chọn đặc trưng

Các đặc trưng được chọn lọc nhằm đảm bảo các thuật toán thực hiện hiệu quả. Mô hình xét đến tính tương quan của đăc trưng với nhau, đảm bảo không có tương quan quá cao giữa các đặc trưng.

Tính tương quan (correlation) giữa các biến độc lập numerical và với biến phụ thuộc được trực quan hóa với heatmap như sau:

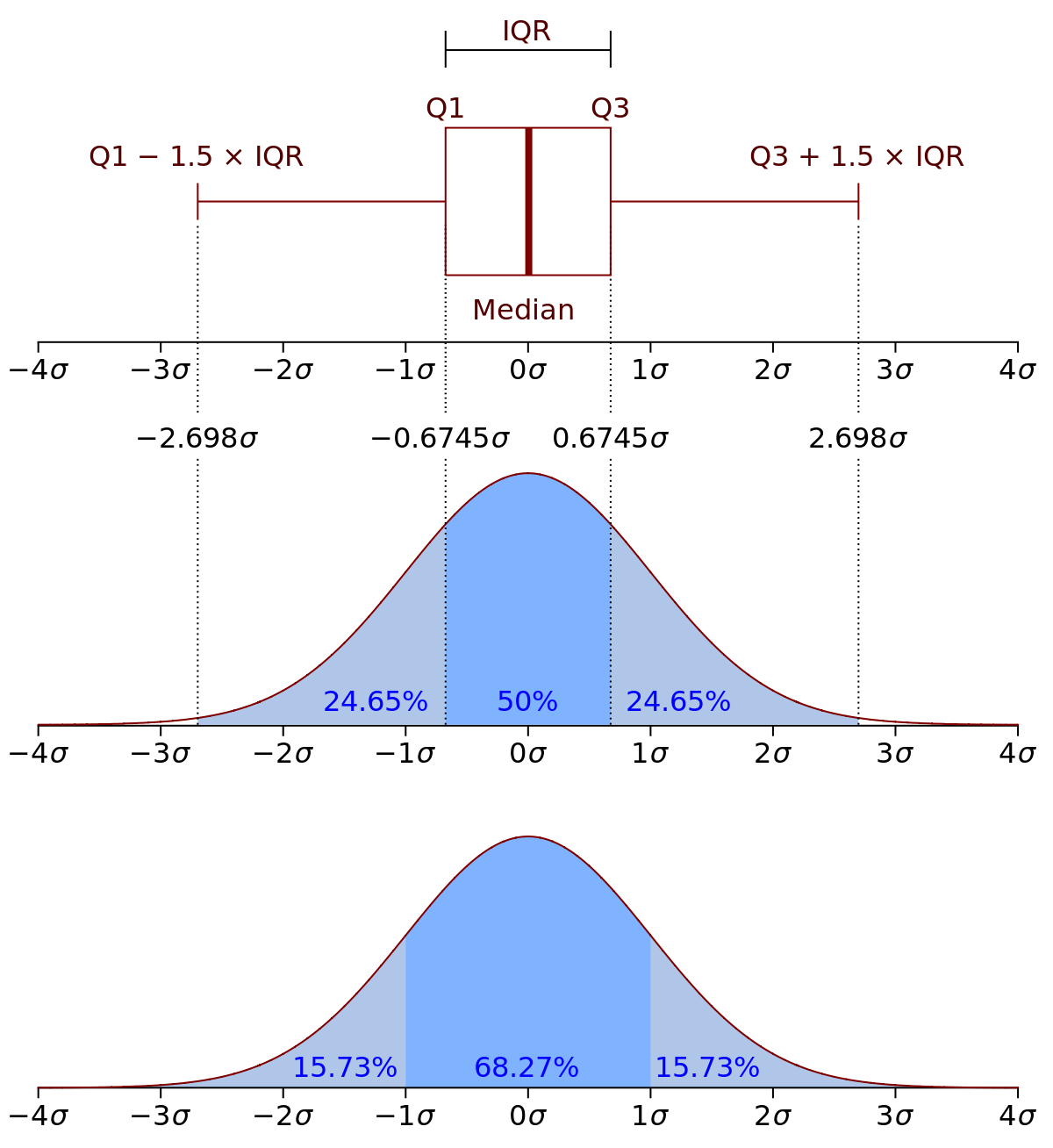
**A grey and blue squares

Description automatically generated**

Dựa vào ý nghĩa và tính tương quan được nêu trên, đặc trưng Fuel Consumption Comb(L/100km) được tính bằng 55% Fuel Consumption City và 45% Fuel Consumption Hwy nên ta sẽ loại bỏ 2 đặc trưng này. Đặc trưng Fuel Consumption Comb (mpg) cũng được loại bỏ bởi cách tính lượng nhiên liệu tiêu thụ theo đơn vị khác sẽ không cần thiết. Đa số các đặc trưng độc lập có mối tương quan cao với biến phụ thuộc cho thấy mối quan hệ tuyến tính của dữ liệu với đầu ra cần dự đoán. Tuy nhiên lại xảy ra hiện tượng đa cộng tuyến khi các biến độc lập tương quan cao với nhau, điều này đã được giải thích ở phần trực quan hóa. Cùng với việc tính hệ số tolerance VIF ( Variance inflation factor) lớn hơn 5 của các biến độc lập cũng cho thấy đa cộng tuyến xảy ra. Phương pháp kiểm định ANOVA ( kiểm tra mối tương quan giữa categorical và numerical) cho biết các biến độc lập categorical cũng có mối tương quan cao với biến phụ thuộc.

### 3.2.3. Xử lý outlier

Outlier là các điểm dữ liệu khác biệt đáng kể so với phần còn lại của dữ liệu, do lỗi đo lường, lỗi nhập dữ liệu và do biến thiên tự nhiên. Phân phối của dữ liệu không phải phân phối chuẩn nên cách phát hiện outlier trong bài toán là sử dụng phương pháp IQR, những điểm nằm ngoài khoảng Q1- 1.5IQR và Q3+1.5IQR được cho là outlier, IQR là khoảng biến thiên tứ phân vị nằm giữa Q1 và Q3 ( quantile(0.25) và quantile( 0.75) ). Tại sao lại là 1.5IQR vì nếu dữ liệu tuân theo phân phối chuẩn thì median chính là mean, theo phân phối chuẩn thì 68% dữ liệu nằm xung quanh 1 lần độ lệch chuẩn từ giá trị mean, 95% nằm trong 2 lần độ lệch chuẩn, 99.7% nằm trong 3 lần độ lệch chuẩn và 0.3% là outlier thì từ đấy người ta tính toán khoảng 1.5IQR là 50% dữ liệu nằm trong 1 lần độ lệch chuẩn từ mean và 99.3% nằm trong 3 lần độ lệch chuẩn. Quy tắc IQR này ước lượng gần đúng quy tắc 3 lần độ lệch chuẩn được khái quát lên với dữ liệu không phải phân phối chuẩn.



**A screenshot of a graph

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

**A graph with a line and a curve

Description automatically generated**

**A graph with a line going up

Description automatically generated**

**A graph of a graph

Description automatically generated**

**A graph with blue bars and a dotted line

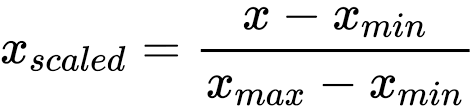
Description automatically generated**

Phương pháp IQR cho thấy có outlier ở các đặc trưng numerical nhưng xem xét các outlier với dữ liệu thực tế thấy được các outlier này không phải là lỗi đo lường mà là do biến thiên tự nhiên, có tồn tại ngoài thực tế với số lượng ít nên ta xem xét không loại bỏ mà thay thế các outlier bằng ngưỡng Q3+1.5IQR. Sau khi xử lý outlier ta thấy phân phối dữ liệu có sự thay đổi, dữ liệu không bị skew nhiều khi chưa xử lý outlier.

### 3.2.4. Chuẩn hóa dữ liệu

Nhằm đảm bảo cho thuật toán tính toán hiệu quả và chính xác, các dữ liệu đầu vào cần được chuẩn hóa dữ liệu.

Đối với những dữ liệu định lượng, phương pháp chuẩn hóa là đặt giá trị lớn nhất và nhỏ nhất là các cận của khoảng và đưa những giá trị trong dữ liệu về cùng một khoảng.

****

Xử lý dữ liệu numerical trong test nằm ngoài khoảng giá trị trong training data

Có 3 trường hợp:

* Scale khoảng [0,1] cho toàn bộ dataset rồi chia train, test
* Chia train, test rồi scale [0,1] cho train và test riêng, tỷ lệ khác nhau do min max khác nhau
* Chia train,test rồi scale cho train [0,1] rồi lấy min max của train data rồi scale cho test data

Khi xuất hiện giá trị nằm ngoài khoảng trong tập train thì ta vẫn sẽ scale cho giá trị mới theo min max của tập train. Vì khi giá trị này nằm ngoài khoảng sẽ thay đổi phân phối dẫn đến kết quả chuẩn hóa thay đổi không nằm trong [0,1]. VD: Các giá trị nằm trong [0,5] trong tập train được chuẩn hóa thành [0,1] thì giá trị mới nằm trong [0,10] trong test sẽ được chuẩn hóa thành [0,2].

Đối với các đặc trưng categorical, bài toán lựa chọn hai cách tiếp cận là dummy encoding và frequency encoding. Các giá trị categorical được đưa về vector chứa 0 và 1 thể hiện sự xuất hiện của lớp của mỗi mẫu dữ liệu, biểu diễn bằng các cột là lớp của thuộc tính trong dữ liệu thông qua get\_dummies, là công cụ trong thư viện scikit-learn. Với đặc trưng Model có 2053 giá trị duy nhất (high-cardinality) thì việc sử dụng biến dummies cho từng categories sẽ tăng số chiều của dữ liệu lên rất nhiều nên phải chọn cách xử lý khác. Có một số cách xử lý high-cardinality như Hash Encoding, Binary Encoding , bài toán sử dụng phương pháp Frequency Encoding để mã hóa đặc trưng này. Frequency Encoding trả về tần suất xuất hiện của từng categories trong toàn bộ tập dữ liệu, phương pháp này có ưu điểm là không tăng số chiều của dữ liệu và dễ thực hiện mà vẫn giữ thông tin của dữ liệu nhưng phương pháp giả định mối tương quan giữa tần suất và biến đầu ra.

Khi triển khai thực tế, những dữ liệu nằm trong lớp chưa xác định so với dữ liệu được huấn luyện, được đưa về lớp tương đồng với lớp có sẵn, hoặc sử dụng lớp “Unknown”, đã xác định trong mô hình. Tuy nhiên, việc thực hiện lại và cập nhật bước chuẩn hóa cũng được xem xét nếu dữ liệu huấn luyện không còn phù hợp với thực tế.

### 3.2.5. Chia dữ liệu

Dữ liệu được chia thành 3 tập khác nhau là tập training, tập validation và tập test.

* Training set: tập dữ liệu lớn nhất để huấn luyện mô hình, training set giúp tìm ra các tham số cho mô hình thông qua việc tìm mối quan hệ giữa các biến trong dữ liệu training, dùng để tạo ra mô hình.
* Validation set: tập dữ liệu sử dụng để đánh giá mô hình trong giai đoạn training, đóng vai trò như unseen data giúp mô hình đánh giá khả năng dự đoán trên dữ liệu mới cùng với quá trình tinh chỉnh tham số cho mô hình nhằm tìm ra mô hình tốt nhất ( không thể lấy tập training hay test để tinh chỉnh được vì dùng tập training sẽ dẫn đến overfitting còn tập test đóng vai trò như đánh giá cuối cùng).
* Test set: đánh giá mô hình hoàn chỉnh cuối cùng.

Khi chia dữ liệu cần chú ý đến vấn đề dữ liệu không cân bằng nhãn dự đoán. Bài toán đã thử nghiệm chia dữ liệu cho 3 tập với biến target nằm ở các khoảng khác nhau rồi huấn luyện cho kết quả rất kém khi R-squared đạt khoảng 50% trên tập train còn validation set cho kết quả âm. Vấn đề này được giải thích rằng mô hình không khái quát được dữ liệu, dự đoán tốt trên tập train nhưng kém trên tập đánh giá, biến độc lập không giải thích sự thay đổi của biến phụ thuộc xung quanh giá trị trung bình và mô hình này không có ý nghĩa trong việc mô hình hóa mối quan hệ giữa các biến độc lập với phụ thuộc trong tập test. Vậy nên dữ liệu đánh giá cần phù hợp và chính xác với mô hình đang sử dụng.

## 3.3. Huấn luyện và đánh giá mô hình

Đối với bài toán dự đoán với dữ liệu tuyến tính, các thuật toán được ứng dụng như Linear Regression, KNN regression, Support Vector Regression tuy nhiên các thuật toán tree-based khác như Random Forest regression, Gradient Boosting regression cũng được sử dụng để đánh giá mô hình một cách khách quan.

Mô hình được đánh giá qua các metric sau:

* R-square : metric thể hiện biến độc lập giải thích bao nhiêu % variance của biến phụ thuộc, là độ chính xác của mô hình càng gần 1 càng tốt, có thể âm.
* Mean squared error: trung bình bình phương giá trị chêch lệch giữa dự đoán và thực tế, càng gần 0 càng tốt.

### 3.3.1. Mô hình Linear Regression

Hồi quy tuyến tính là một phương pháp phân tích mối quan hệ giữa biến phụ thuộc với một hay nhiều biến độc lập, giả định mối quan hệ tuyến tính giữa các biến để tìm ra một hàm phù hợp để mô tả mối quan hệ này.

Có rất nhiều phương pháp để tìm weight của mô hình như phương pháp dùng đại số tuyến tính Least Squared Error nhưng trong bài toán sử dụng Gradient Descent, mục tiêu chính là tìm ra các weight của hàm hồi quy thông qua việc giảm thiểu chênh lệch giữa giá trị dự đoán và thực tế. Mô hình sử dụng thuật toán Gradient Descent tìm ra weight cho mô hình thông qua việc tối ưu loss function. Gradient Descent tìm cực tiểu của loss function bằng việc cập nhật weight của mô hình thông qua đạo hàm riêng của loss function đối với mối weight. Điều kiện sử dụng Gradient Descent là hàm số khả vi có khả năng đạo hàm và là convex function nếu đạo hàm của hàm số là hàm đồng biến tăng trên tập xác định.

Các tham số quan trọng trong mô hình là learning rate (tốc độ học) và iteration (số bước lặp). Learning rate là tốc độ cập nhật weight ở mối bước lặp của gradient, cập nhật weight bằng cách trừ đi learning rate \* gradient là mang dấu ngược với hướng của gradient vì đây là bài toán cực tiểu hóa cần giảm loss function

Learning rate ảnh hưởng đến tốc độ hội tụ của thuật toán, learning rate lớn mô hình hội tụ nhanh hơn nhưng có thể overshooting qua điểm cực tiểu nếu quá lớn, learning rate nhỏ thuật toán hội tụ lâu hơn và có thể chưa đạt đến điểm cực tiểu khi hết số iteration. Điều kiện dừng có thể là hết số iteration hoặc loss function nhỏ hơn một threshhold nào đó.

Kết quả của mô hình Linear Regression với Gradient Descent với tham số đầu vào được xác định sẵn:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **R2 train** | **R2 val** | **MSE train** | **MSE val** |
| LR with GD  learning rate = 0.01, iteration = 500 | 0.83 | 0.82 | 530.66 | 584.18 |

Kết quả trên cho thấy mô hình dự đoán rất tốt trên cả 2 tập không hề xảy ra overfitting hay underfitting nhưng còn có thể tốt hơn nếu ta tuning tham số của mô hình. Ta sẽ tuning tham số learning rate trước cho mô hình và giữ nguyên iteration = 500.

Tuning learning rate nằm trong các giá trị [ 0.001, 0.01, 0.07, 0.1, 0.4, 0.6, 0.8 ] tìm được giá trị learning rate = 0.6 tốt nhất. Kết quả của mô hình sau khi tuning learning rate:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **R2 train** | **R2 val** | **MSE train** | **MSE val** |
| LR with GD learning rate = 0.6, iteration = 500 | 0.98 | 0.98 | 48.61 | 55.35 |

Ta thấy tăng learning rate cải thiện rất nhiều độ chính xác của mô hình, bây giờ thay đổi iteration xem mô hình còn cải thiện được hay không. Giữ nguyên learning rate = 0.6 và tuning iteration nằm trong các giá trị [ 1000, 2000, 6000, 8000 ] tìm được giá trị iteration = 8000 tốt nhất. Kết quả mô hình sau khi tuning iteration:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **R2 train** | **R2 val** | **MSE train** | **MSE val** |
| LR with GD learning rate = 0.6, iteration = 8000 | 0.99 | 0.98 | 29.9 | 42.85 |

Mô hình có thay đổi nhỏ khi tăng số iteration và kết quả chấp nhận được. Kết quả mô hình cuối cùng trên tập test ở bảng dưới đây:

|  |  |  |
| --- | --- | --- |
| **Model** | **R2 test** | **MSE test** |
| LR with GD learning rate = 0.6, iteration = 8000 | 0.98 | 39.59 |

Kết quả mô hình rất tốt do việc mô hình Linear Regression hoạt động tốt trên dữ liệu tuyến tính. Mô hình Linear Regression còn cho biết ảnh hưởng của mỗi đặc trưng đến kết quả dự đoán lượng khí thải CO2 nhờ vào các weight của mô hình, weight càng lớn càng cho thấy tầm ảnh hưởng của đặc trưng lên lượng khí thải CO2 , được biểu diễn dưới đây:

**A screenshot of a computer

Description automatically generated**

Đặc trưng fuel\_cons\_comb là lượng tiêu thụ nhiên liệu hỗn hợp, đặc trưng quan trọng nhất đúng như dự đoán.Ta quan sát thấy rằng loại nhiên liệu Ethanol(E85) có ảnh hưởng thứ hai và có weight âm chứng tỏ sử dụng loại nhiên liệu này giúp giảm lượng khí thải CO2 giúp làm sáng tỏ việc loại nhiên liệu này có lượng khí thải trung bình cao nhất đã được giải thích ở phần trực quan.

**A screen shot of a computer

Description automatically generated**

**A graph with text and numbers

Description automatically generated**

**A screen shot of a graph

Description automatically generated**

1. Xử lý dữ liệu categories ở test data không nằm trong training data

- Kiểm tra số lượng unique value ở categorical feature trong train và test data

- Nếu có categories nằm ngoài train data thì có những cách xử lý sau:

* Trong bài toán đặc trưng Model có 2053 giá trị duy nhất được mã hóa bằng frequency encoding -> lấy số lần xuất hiện của mỗi categories chia cho độ lớn dataset là 7384. Nếu trong test data xuất hiện Model mới thì vẫn sẽ được mã hóa như trên. Tuy nhiên điều này không phù hợp với thực tế nên ta xem xét đến việc phải cập nhật lại train data với các categories mới và thực hiện mã hóa lại.
* Đối với đặc trưng Tranmission có 27 giá trị duy nhất đã được feature engineering đưa về 5 giá trị duy nhất, tương tự với Make Type có 42 đưa về 4 giá trị duy nhất và Vehicle Class có 16 đưa về 4 giá trị duy nhất thì dữ liệu thực tế không nằm trong train data sẽ được đưa về categories tương đồng trong train data( VD: M5,M6,M7 trong Tranmission được nhóm vào categories mới là Manual trong train data thì categories mới ở test set là M8 sẽ được đưa về Manual)
* Với đặc trưng Fuel Type có 4 loại nhiên liệu trong training data mà test data có 4 loại nhiên liệu trên và thêm 1 loại nhiên liệu mới thì ta sẽ thêm 1 đặc trưng vào train data sau khi mã hóa dummies và đặt cột bằng 0 để biểu thị dữ liệu không có sự xuất hiện của categories mới.

1. Xử lý dữ liệu numerical nằm ngoài khoảng giá trị trong training data

Có 3 trường hợp:

* Scale khoảng [0,1] cho toàn bộ dataset rồi chia train, test
* Chia train, test rồi scale [0,1] cho train và test riêng, tỷ lệ khác nhau do min max khác nhau
* Chia train,test rồi scale cho train [0.1] rồi lấy min max của train data rồi scale cho test data
* Khi xuất hiện giá trị nằm ngoài khoảng trong tập train thì ta vẫn sẽ scale cho giá trị mới theo min max của tập train. Vì khi giá trị này nằm ngoài khoảng sẽ thay đổi phân phối dẫn đến kết quả chuẩn hóa thay đổi không nằm trong [0,1]. VD: Các giá trị nằm trong [0,5] trong tập train được chuẩn hóa thành [0,1] thì giá trị mới nằm trong [0,10] trong test sẽ được chuẩn hóa thành [0,2].

1. Dữ liệu đẩy vào các decision tree trong random forest là khác nhau?

* Random forests bootstrap the data and randomly select features. bootstrapping means that it samples a data-set with the same size as the original dataset, but with replacement. So if you have N data points, each tree will use N data points, but some my be duplicated (as it samples them one by one with replacement).
* <https://stats.stackexchange.com/questions/534356/are-random-forests-trained-with-the-whole-dataset>
* <https://stats.stackexchange.com/questions/88980/why-on-average-does-each-bootstrap-sample-contain-roughly-two-thirds-of-observat>
* https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/#:~:text=Boosting%E2%80%93%20It%20combines%20weak%20learners,works%20on%20the%20Bagging%20principle.

1. Tại sao sự kết hợp kết quả của nhiều mô hình kém lại cho 1 kết quả tốt hơn?

<https://stats.stackexchange.com/questions/255230/how-do-ensemble-methods-outperform-all-their-constituents#:~:text=It's%20not%20guaranteed.,somewhat)%20independent%20of%20one%20another>.

1. Support vector ?

<https://www.analyticsvidhya.com/blog/2021/04/insight-into-svm-support-vector-machine-along-with-code/#:~:text=Hard%20and%20Soft%20SVM,-I%20would%20like&text=We%20can%20now%20clearly%20state,allow%20any%20misclassification%20to%20happen>.

<https://viblo.asia/p/support-vector-machine-jvEla304Kkw>

1. Random forest
2. <https://www.quantstart.com/articles/bootstrap-aggregation-random-forests-and-boosted-trees/#:~:text=techniques%20to%20work.-,The%20Bootstrap,with%20a%20machine%20learning%20model>.