

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🟅🙢



**ĐỒ ÁN MÔN HỌC HỌC MÁY**

**ĐỀ TÀI**

**ỨNG DỤNG HỌC MÁY TRONG**

**DỰ ĐOÁN MỨC LƯƠNG NGÀNH CÔNG NGHỆ THÔNG TIN**

**GVDH: TS. Trần Nhật Quang**

**SVTH:**

**Hoàng Hữu Đức 19110349**

**Nguyễn Minh Luân 19110395**

**Trần Kiện Khang 19110375**

**Lê Thị Minh Nguyệt 19110413**

*Thành phố Hồ Chí Minh – Tháng 12/2021*

Mục lục

[CHƯƠNG 1: TÌM HIỂU VẤN ĐỀ ĐẠO VĂN 1](#_Toc89157627)

[1. Khái niệm đạo văn 1](#_Toc89157628)

[2. Các loại đạo văn phổ biến 1](#_Toc89157629)

[3. Cách sử dụng tài liệu và phòng tránh đạo văn hợp lý 1](#_Toc89157630)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 2](#_Toc89157631)

[1. Tổng quan về học máy 2](#_Toc89157632)

[1.1. Khái niệm về học máy 2](#_Toc89157633)

[1.2. Ứng dụng của học máy trong đời sống 2](#_Toc89157634)

[1.3. Quy trình xử lý học máy 3](#_Toc89157635)

[1.4. Cơ hội và thách thức 3](#_Toc89157636)

[1.5. Phương pháp học máy 3](#_Toc89157637)

[1.6. Phân loại học máy 4](#_Toc89157638)

[1.7. Một số khái niệm khác 5](#_Toc89157639)

[2. Quy trình triển khai một dự án Machine Learining 7](#_Toc89157640)

[2.1. Look at the big picture 7](#_Toc89157641)

[2.2. Get data 7](#_Toc89157642)

[2.3. Discover and visualize the data to gain insights 7](#_Toc89157643)

[2.4. Prepare the data for machine learning algorithms 7](#_Toc89157644)

[2.5. Training and evaluate models 7](#_Toc89157645)

[2.6. Fine-tune your models 8](#_Toc89157646)

[2.7. Analyze and test your solution 8](#_Toc89157647)

[2.8. Launch, monitor, and maintain your system 8](#_Toc89157648)

[3. Classification 8](#_Toc89157649)

[3.1. MNIST Dataset 8](#_Toc89157650)

[3.2. Training a binary classifier 8](#_Toc89157651)

[3.3. Multiclass Classification 8](#_Toc89157652)

[4. Training model 9](#_Toc89157653)

[4.1. Linear Regression (Hồi quy tuyến tính) 9](#_Toc89157654)

[4.2. Polynomial regression models 9](#_Toc89157655)

[4.3. Hypothesis functions 9](#_Toc89157656)

[4.4. Cost function 10](#_Toc89157657)

[4.5. Normal Equation 10](#_Toc89157658)

[4.6. Gradient Descent 11](#_Toc89157659)

[5. Thuật toán Support Vector Machines (SVM) 12](#_Toc89157660)

[5.1. Khái niệm thuật toán: 12](#_Toc89157661)

[5.2. SVM Classification 12](#_Toc89157662)

[5.3. Linear SVM Classification 12](#_Toc89157663)

[5.4. Non-Linear SVM Classification 13](#_Toc89157664)

[5.5. SVM Regression 14](#_Toc89157665)

[6. Thuật toán Decision Trees 14](#_Toc89157666)

[6.1. Giới thiệu: 14](#_Toc89157667)

[6.2. Decision trees 15](#_Toc89157668)

[6.3. Impurity measurement 15](#_Toc89157669)

[6.4. Thuật toán CART - Classification and Regression Tree 16](#_Toc89157670)

[6.5. Ưu điểm và khuyết điểm của Decision Trees: 16](#_Toc89157671)

[7. Ensemble learning and Random Forests 17](#_Toc89157672)

[7.1. Khái niệm: 17](#_Toc89157673)

[7.2. Voting method 17](#_Toc89157674)

[7.3. Bagging method 17](#_Toc89157675)

[7.4. Boosting method 18](#_Toc89157676)

[CHƯƠNG 3: GIỚI THIỆU SẢN PHẨM ĐỀ TÀI - ỨNG DỤNG HỌC MÁY TRONG DỰ ĐOÁN MỨC LƯƠNG NGÀNH CÔNG NGHỆ THÔNG TIN 19](#_Toc89157677)

[1. Giới thiệu đề tài 19](#_Toc89157678)

[2. Kết quả thực nghiệm 19](#_Toc89157679)

[3. Nhận xét và đánh giá về đề tài 19](#_Toc89157680)

[3.1. Kết quả đạt được 19](#_Toc89157681)

[3.2. Hạn chế 19](#_Toc89157682)

[3.3. Hướng phát triển và mở rộng 19](#_Toc89157683)

[DANH MỤC TÀI LIỆU THAM KHẢO 20](#_Toc89157684)

[LỜI CẢM ƠN 21](#_Toc89157685)

# CHƯƠNG 1: TÌM HIỂU VẤN ĐỀ ĐẠO VĂN

*Chúng em xin cam đoan đồ án này do các thành viên trong nhóm thực hiện. Chúng em không sao chép, sử dụng bất kỳ tài liệu, mã nguồn… của người khác mà không ghi nguồn. Chúng em xin chịu hoàn toàn trách nhiệm nếu vi phạm đạo văn.*

*Danh sách thành viên nhóm xin cam kết bao gồm:1. Hoàng Hữu Đức, 2. Trần Kiện Khang, 3. Nguyễn Minh Luân, 4. Lê Thị Minh Nguyệt.*

## Khái niệm đạo văn

Đạo văn là chiếm hữu một cách sai trái, ăn cắp, công bố ngôn ngữ, suy nghĩ, ý tưởng, hay cách diễn đạt của người khác và xem chúng như là những gì do mình tự tạo ra. Đạo văn có thể là cố ý hoặc vô ý. Đạo văn được xem là hành vi thiếu trung thực về mặt học thuật và vi phạm đạo đức

## Các loại đạo văn phổ biến

*Đạo văn hoàn toàn* là lấy toàn bộ tác phẩm của người khác và coi nó là của riêng mình

*Đạo văn trực tiếp* là sao chép và dán văn bản vào tác phẩm của riêng mình mà không cần trích dẫn.

*Đạo văn diễn giả* là điễn đạt lại một đoạn văn bản, một ý tưởng bằng từ ngữ của chính mình. Diễn giải mà không trích dẫn là loại đạo văn phổ biến nhất

*Đạo văn chắp vá*, còn được gọi là đạo văn ghép, có nghĩa là sao chép các cụm từ, đoạn văn và ý tưởng từ các nguồn khác nhau và ghép chúng lại với nhau để tạo ra một văn bản mới.

*Tự đạo văn* nghĩa là sử dụng lại tác phẩm mà mình đã gửi hoặc xuất bản trước đó. Hình thức tự đạo văn nghiêm trọng nhất là chuyển một bài đoạn văn, tài liệu mình đã nộp, gửi ở một nơi khác.

## 3. Cách sử dụng tài liệu và phòng tránh đạo văn hợp lý

Để tránh đạo văn, ta phải ghi lại các nguồn tham khảo chính xác bằng cách sử dụng các Chú thích chân văn bản, chú thích cuối câu, chú thích đặt trong dấu ngoặc và phải soạn một trang Tài liệu tham khảo.

Không sử dụng nội dung của người khác trên internet và kết hợp trực tiếp thông tin đó vào sản phẩm của ta mà không diễn giải và thừa nhận nguồn của nó.

Ngoài việc chúng ta hành động một cách thiếu đạo đức, không trung thực và không tiếp thu được bất kỳ kiến thức, kỹ năng gì trong quá trình học tập ở lớp, ở trường. Vì vậy chúng chúng ta cần nhận thức rõ tác hại của việc đạo văn và sử dụng các trích dẫn từ các nguồn tham khảo một cách hợp lý, chính xác.

*.*

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## Tổng quan về học máy

### 1.1. Khái niệm về học máy

Đến ngày nay, chúng ta có rất nhiều định nghĩa về Học máy, một vài định nghĩa phổ biến và kinh điển nhất như sau:

Theo Arthur Samuel (1959): Máy học là ngành học cung cấp cho máy tính khả năng học hỏi mà không cần được lập trình một cách rõ ràng

Theo Giáo sư Tom Mitchell – Carnegie Mellon University: Machine Learning là 1 chương trình máy tính được nói là học hỏi từ kinh nghiệm E từ các tác vụ T và với độ đo hiệu suất P. Nếu hiệu suất của nó áp dụng trên tác vụ T và được đo lường bởi độ đo P tăng từ kinh nghiệm E Machine learning là ngành học cung cấp cho máy tính khả năng học hỏi mà không cần được lập trình một cách rõ ràng.

### 1.2. Ứng dụng của học máy trong đời sống

Học máy có ứng dụng rộng khắp trong các ngành khoa học/sản xuất, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ. Một số ứng dụng thường thấy

*Xử lý ngôn ngữ tự nhiên (Natural Language Processing):* xử lý văn bản, giao tiếp người – máy, …

*Nhận dạng (Pattern Recognition):* nhận dạng tiếng nói, chữ viết tay, vân tay, thị giác máy (Computer Vision) …

*Tìm kiếm (Search Engine) Chẩn đoán trong y tế:* phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động.

*Tin sinh học:* phân loại chuỗi gene, quá trình hình thành gene/protein

*Vật lý:* phân tích ảnh thiên văn, tác động giữa các hạt …

*Phát hiện gian lận tài chính (Financial Fraud):* gian lận thẻ tỉn dụng

*Phân tích thị trường chứng khoán* (Stock Market Analysis)

*Ứng dụng trò chơi giải trí*: tự động chơi cờ, hành động của các nhân vật ảo

*Robot – Người máy*: là tổng hợp của rất nhiều ngành khoa học, trong đó học máy tạo nên hệ thần kinh/bộ não của người máy.

### 1.3. Quy trình xử lý học máy

#### Thu thập dữ liệu (Data collection)

Để máy tính có thể học được, chúng ta cần có một bộ dữ liệu (dataset), dữ liệu này có thể được thu thập từ internet, tự thu thập từ thực tế. Để thuật toán có độ chính xác cao thì dữ liệu cũng cần có sự chỉn chu, chính xác.

#### Tiền xử lý (Preprocessing)

Ở bước này chúng ta cần nhiều thời gian để xử lý dữ liệu, loại bỏ những thuộc tính (features) thừa, bổ sung những thuộc tính còn thiếu, mã hóa một số đặc trưng, rút gọn dữ liệu nhưng vẫn đảm bảo kết quả.

#### Huấn luyện mô hình (Training model)

Ở bước này chúng ta sẽ cho thuật toán học trên những dữ liệu đã thu thập và xử lý ở bước tiền xử lý.

#### Đánh giá mô hình (Evaluating model)

Sau khi đã huấn luyện xong, chúng ta cần dùng thang đo để đánh giá mô hình, tùy vào từng thang đo khác nhau mà mô hình cũng được đánh giá tốt hay không khác nhau. Độ chính xác của mô hình đạt trên 80% được cho là tốt.

#### Cải thiện (Improve)

Sau khi đã xử lý các mô hình, các mô hình có chính xác không cao cần được huấn luyện lại, chúng ta sẽ lặp lại từ bước 3, cho đến khi có được độ chính xác như mong muốn.

### 1.4. Cơ hội và thách thức

Machine Learning đang thúc đẩy công nghệ và đưa ra những cách tiếp cận về công nghệ mới cho doanh nghiệp, nhưng có một số điều về machine learning mà các doanh nghiệp cần lưu ý đó về khả năng và giới hạn của nó.

Máy móc được đào tạo bởi con người và suy nghĩ ​​của con người có thể được đưa vào các thuật toán - nếu thông tin hoặc dữ liệu phản ánh không được đưa vào chương trình Machine Learning, chương trình sẽ học theo cách đó và tạo ra những suy nghĩ tiêu cực.

Trong một số trường hợp, Machine Learning có thể tác hại xấu thêm các vấn đề xã hội. Thực trạng hiện nay cũng cho thấy rõ, Facebook sử dụng thuật toán để thu hút những thứ liên quan mà người dùng quan tâm, từ đó sẽ dẫn đến những điều cực đoan khó tránh phải

### 1.5. Phương pháp học máy

*Phương pháp quy nạp:* Máy học phân biệt các khái niệm (concept) dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.

*Phương pháp suy diễn:* Máy học phân biệt các khái niệm dựa vào các quy luật. Phương pháp này cho phép tận dụng các kiến thức chuyên ngành để hỗ trợ lập trình các giải thuật máy tính.

### 1.6. Phân loại học máy

#### 1.6.1. Supervised Learning - Học có giám sát

Học có giám sát là nơi ta có các biến đầu vào (X) và biến đầu ra (Y) và ta sử dụng thuật toán để tìm hiểu hàm ánh xạ từ đầu vào đến đầu ra.

Y = f (X)

Mục đích là để xây dựng hàm ánh xạ một cách tốt nhất có thể để khi ta có dữ liệu đầu vào mới (X) và ta có thể dự đoán các biến đầu ra (Y) cho dữ liệu đó.

Các thuật toán supervised learning được phân làm 2 loại chính:

**Classification:** một bài toán được xếp vào loại này là một bài toán mà các label và data được xếp vào 1 tập hữu hạn các giá trị.

*Ví dụ:* bài toán phân loại học sinh giỏi, học sinh khá và học sinh trung bình, khi training data ta đưa vào dữ liệu có gán nhãn cho từng data, học sinh này có dữ liệu như này thì sẽ là học sinh giỏi, có dữ liệu dạng khác sẽ là học sinh khá và cứ như thế.

**Regression:** một bài toán được xếp vào loại này là một bài toán có label là một giá trị cụ thể và không có giới hạn về các giá trị.

*Ví dụ:* Về định giá giá trị tiền ảo, các giá trị sẽ luôn thay đổi và không có giới hạn về mặt giá trị. Khi đưa các giá trị data vào chúng ta có cung cấp thông tin về công thức, hay tin tức gán vào label cụ thể cho từng data, từ đó hệ thống có thể tự học được.

#### 1.6.2. Unsupervised learning - Học không giám sát

Trong thuật toán này, chúng ta không biết được nhãn mà chỉ có dữ liệu đầu vào. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Clustering: bài toán thuộc loại này là những bài phân loại toàn bộ dữ liệu thành các nhóm nhỏ hơn dựa trên sự liên quan với nhau trong data. Ví dụ như phân loại đồ vật chúng ta đưa vào các tập dữ liệu đồ vật và hệ thống có thể phân loại các đồ vật này theo tính chất của đồ vật theo chất liệu, hình dáng kích thước.

Association: là dạng bài toán đưa ra quy luật dựa trên data có trước. Ví dụ người hay uống bia sẽ có xu hướng hút thuốc, hoặc những người thích yêu động vật thì sẽ ăn chay.

#### 1.6.3. Semi-Supervised learning - Học bán giám sát

Semi-supervised learning (học bán giám sát): là khi chúng ta có một lượng lớn dữ liệu mà chỉ một phần trong nó được gián nhãn. Ví dụ chỉ có một phần hình ảnh được gắn nhãn và phần còn lại chưa được gán nhãn được thu thập từ internet

#### 1.6.3. Reinforcement learning - Học tăng cường

Loại hình học này được sử dụng để củng cố hoặc tăng cường mạng lưới dựa trên thông tin phê bình. Đó là, một mạng lưới đang được đào tạo theo phương pháp học tăng cường, nhận được một số phản hồi từ môi trường. Tuy nhiên, phản hồi chỉ mang tính đánh giá chứ không mang tính hướng dẫn như trong trường hợp học có giám sát.

*Ví dụ:* việc học cho xa tự hành khi gặp chướng ngại vật sẽ đánh dấu là không được đi tiếp mà phải quay lại, tương tự cho các vật thể, từ đó thuật toán sẽ tốt hơn

### 1.7. Một số khái niệm khác

#### 1.7.1. Dữ liệu xấu và các thuật toán

Dữ liệu xấu là dữ liệu được gắn nhãn không chính xác, đầy lỗi, thiếu giá trị hoặc kém chất lượng. Khi dữ liệu này được sử dụng, kết quả sẽ không phản ánh đúng thực tế. Điều này có nghĩa là các mô hình dự đoán sẽ thất bại. Mô hình có thể dự đoán một điều gì đó sẽ xảy ra ít thường xuyên hơn so với nó nếu nó có bộ dữ liệu đầy đủ và sự cố máy móc tốn kém có thể xảy ra sau đó.

Đôi khi dữ liệu xấu có thể có nghĩa là dữ liệu bị sai lệch hoặc quá phức tạp để thuật toán xử lý. Con người có những thành kiến vô thức mà chúng ta áp dụng cho thế giới xung quanh. Những thành kiến này có thể xâm nhập vào cách chúng tôi thiết kế hệ thống phân tích dữ liệu của mình. Ví dụ, trong những năm gần đây, nhiều thông tin cho rằng hệ thống nhận dạng khuôn mặt nhận dạng khuôn mặt người da đen hoặc người Châu Á kém chính xác hơn nhiều so với khuôn mặt người da trắng. Thuật toán học máy vốn dĩ không thiên về khuôn mặt trắng, nhưng dữ liệu mà nó được cung cấp chủ yếu là khuôn mặt trắng. Điều này có nghĩa là việc xác định những khuôn mặt này trở nên chính xác hơn nhiều.

#### 1.7.2. Khái niệm về test set và generalizaiton error

**Test set:** là tập dữ liệu thứ cấp (hoặc cấp ba) được sử dụng để kiểm tra chương trình học máy sau khi nó đã được đào tạo trên tập dữ liệu đào tạo ban đầu (training set). Tập thử nghiệm có các kích thước gần giống nhau, nhỏ hơn nhiều so với kích thước của tập huấn luyện (training set). Tập thử nghiệm chứa các ví dụ mà thuật toán học tập chưa từng thấy trước đây, vì vậy nếu mô hình hoạt động tốt trong việc dự đoán kết quả của các ví dụ từ bộ thử nghiệm thì mô hình đó được xem là mô hình tốt. A test set được gọi là test data set hoặc test data.

**Generalization error:** Đối với các ứng dụng học có giám sát trong học máy và lý thuyết học thống kê, lỗi tổng quát hóa (còn được gọi là lỗi ngoài mẫu hoặc rủi ro) là thước đo mức độ chính xác của một thuật toán có thể dự đoán các giá trị kết quả cho dữ liệu chưa từng thấy trước đây. Vì thuật toán học được đánh giá trên các mẫu hữu hạn, nên việc đánh giá thuật toán học có thể nhạy cảm với lỗi lấy mẫu. Do đó, các phép đo sai số dự đoán trên dữ liệu hiện tại có thể không cung cấp nhiều thông tin về khả năng dự đoán trên dữ liệu mới. Lỗi tổng quát hóa có thể được giảm thiểu bằng cách tránh trang bị quá nhiều trong thuật toán học tập. Hiệu suất của một thuật toán học máy được trực quan hóa bằng các biểu đồ hiển thị các giá trị ước tính của sai số tổng quát hóa thông qua quá trình học tập, được gọi là đường cong học tập.

#### 1.7.3. Khái niệm Hyperparameter tuning

Có một danh sách các mô hình học máy khác nhau. Tất cả chúng đều khác nhau theo cách này hay cách khác, nhưng điều khiến chúng khác biệt là các tham số đầu vào cho mô hình. Các tham số đầu vào này được đặt tên là Hyperparameters. Những siêu tham số này sẽ xác định kiến trúc của mô hình và phần tốt nhất về những thông số này là ta có thể lựa chọn những thông số này cho mô hình của mình. Tất nhiên, ta phải chọn từ danh sách các siêu tham số cụ thể cho một mô hình nhất định vì nó thay đổi theo từng mô hình.

Các cách tiếp cận để điều chỉnh Hyperparameter

* Manual Search
* Random Search
* Grid Search

#### 1.7.4. Khái niệm Valiation set

Thông thường chúng ta xáo trộn dữ liệu sẽ tách dữ liệu thành 3 tập con: training, validation, test set. Tập training set thường là lớn nhất dùng để đào tạo mô hình. 2 tập còn lại có kích thước gần giống nhau và nhỏ hơn nhiều so với kích thước của tập training. Validation set và test set không được đưa vào learning algorithm để sử dụng. Không có tỷ lệ tối ưu để chia 3 tập con này mà tùy vào từng bài toán. Thông thường người ta có thể chia 70% cho tập training 15% cho tập validation 15% cho tập test set hoặc 95%, 2.5% và 2.5%.

Sử dụng tập validation set để: Chọn thuật toán và tìm giá trị tốt nhất của siêu tham số (hyperparameters)

Sử dụng tập test set để: đánh giá mô hình trước khi sản phẩm được đưa ra sử dụng

Có thể tận dụng các kỹ thuật cao hơn như K-fold Cross Validation để kết hợp dữ liệu traning và validation cho việc học của các mô hình và đánh giá mô hình mà không làm rò rĩ dữ liệu.

## 2. Quy trình triển khai một dự án Machine Learining

Khi triển khai một dự án học máy, ta cần triển khai theo 8 bước phổ biến sau:

### 2.1. Look at the big picture

Bước đầu tiên để thực hiện một dự án về machine learning là nhìn bức tranh 1 cách tổng thể từ những yêu cầu mà khách hàng đặt ra, từ đó chúng ta sẽ đặt ra những vấn đề, câu hỏi, và tiếp tục trả lời các câu hỏi đó.

Các câu hỏi thường có như: Biện pháp hiện có hiện tại là gì, hoạt đông ra sao? Thuật toán sử dụng là gì? Tính toán hiệu suất của thuật toán ra sao? Và quan trọng hơn hết là dữ liệu như thế nào, hiện tại dữ liệu ra sao, cần chuẩn bị thêm những gì?

### 2.2. Get data

Trước tiên ta phải nắm rõ được bài toán cần giải quyết ở đây là gì? Từ đó có thể tìm kiếm dữ liệu cho bài toán một cách cụ thể và chính xác. Trong thực tế thường có 2 cách thu thập dữ liệu cơ bản: thu thập dữ liệu từ các nguồn internet, thu thập dữ liệu từ thực tế ví dụ chụp ảnh các vật thể từ đời sống, 2 cách trên đều có các ưu nhược điểm riêng, thu thập dữ liệu từ internet có ưu điểm là dễ dàng nhanh chóng nhưng khi đó dữ liệu sẽ không được chính xác, và sẽ có nhiều dữ liệu rác, còn thu thập dữ liệu từ thực tế thì độ chính xác sẽ cao, phù hợp với thuật toán đưa ra, nhưng nhược điểm là sẽ tốn rất nhiều tài nguyên, tiền bạc để có thể sử dụng cách trên.

### 2.3. Discover and visualize the data to gain insights

Sau khi đã thu thập được tệp dữ liệu cần thiết, thì chúng ta sẽ khám phá dự liệu trên đã thu thập, khám phá dữ liệu chúng ta sẽ sẽ đồ thị trực quan data lên các hệ tọa độ, từ đó có thể nhìn tổng quan data chúng ta có, chúng ta có thể thấy được các điểm data không phù hợp với bài toán, khi đó chúng ta sẽ có được cái nhìn tốt và đi đến các bước tiếp theo

### 2.4. Prepare the data for machine learning algorithms

* Sau khi đã khám phá dữ liệu và có cái nhìn tổng quát về data, đến bước này chúng ta loại bỏ, bổ sung các data thừa và còn thiếu, ở bước này khá quan trọng, nó chiếm tỉ lệ cao trong việc thuật toán của chúng ta có chạy tốt hay không.

### 2.5. Training and evaluate models

Sau khi lấy được dữ liệu và tiền sử lý xong, chúng ta đi đến bước training, ở bước này chúng ta sẽ chọn ra nhiều thuật toán để train, sau khi train chúng ta sẽ có kết quả của từng thuật toán, khi đó chúng ta sẽ chọn ra thuật toán tốt nhất để tiếp tục xử lý

### 2.6. Fine-tune your models

Sau khi đã chọn được thuật toán tốt, chúng ta sẽ tiếp tục đi đến bước tinh chỉnh, ở bước này khi chúng ta tinh chỉnh, thuật toán ta vừa cho là tốt nhất thì chưa chắc tinh chỉnh xong đã tốt nhât, nên vì thế ở bước này chúng ta nên tinh chỉnh tất cả cái thuật toán đã train được, và chọn ra cái tốt nhất

### 2.7. Analyze and test your solution

Sau khi đã chọn được thuật toán hợp lý, chúng ta sẽ đưa tập dữ liệu test vào và test thử xem thuật toán của chúng ta có hoạt động tốt trên các tập dữ liệu test hay không

### 2.8. Launch, monitor, and maintain your system

Cuối cùng là đưa sản phầm ra thực tế, và tiếp tục phát triển mạnh mẽ hơn.

## 3. Classification

### 3.1. MNIST Dataset

MNIST được giới thiệu năm 1998 bởi Yann Lecun và cộng sự nhằm đánh giá các mô hình phân lớp. MNIST là tập dữ liệu chữ viết từ 0 đến 9.

Trong đó, mỗi hình là một ảnh đen trắng chứa một số được viết tay có kích thước là 28x28. Bộ dataset vô cùng đồ sộ với khoảng 60k data training và 10k data test và được sử dụng phổ biến trong các thuật toán nhận dạng ảnh.

Cơ sở dữ liệu MNIST là một cơ sở dữ liệu lớn chứa các chữ số viết tay thường được dùng trong việc huấn luyện xử lý ảnh khác nhau, Cơ sở dữ liệu được tạo ra bằng cách "trộn lại" các mẫu từ bộ dữ liệu ban đầu của NIST

### 3.2. Training a binary classifier

Thuật toán trên là bài toán phân loại, theo tên gọi của nó thì thuật toán này sẽ chia data thành chỉ 2 classes.

*Ví dụ*: về bài toán nhận diện khuôn mặt và chỉ phân biệt là nam hay là nữ.

### 3.3. Multiclass Classification

Multiclass Classification cũng là một bài toán về phân loại, như nó có số lớp nhiều hơn 2.

*Ví dụ:* như phân biệt các loài hoa khác nhau, khi đó sẽ có nhiều lớp là các loài hoa khác nhau

Phân loại Binary và Multiclass. Ở đây x1 và x2 là các biến của chúng ta mà lớp được dự đoán. Chart, scatter chart

Description automatically generated

## 4. Training model

### 4.1. Linear Regression (Hồi quy tuyến tính)

* Linear Regression là một trong những thuật toán cơ bản và phổ biến nhất của Supervised Learning, trong đó đầu ra dự đoán là liên tục. Thuật toán này thích hợp để dự đoán các giá trị đầu ra là các đại lượng liên tục như doanh số hay giá cả thay vì cố gắng phân loại chúng thành các đại lượng rời rạc như màu sắc và chất liệu của quần áo, hay xác định đối tượng trong một bức ảnh là mèo hay chó, …
* Trong mô hình này chúng ta sử dụng phương trình đường thẳng hay tuyến tính có dạng:

f(x) = w1x1 + w0

### 4.2. Polynomial regression models

Tương tự như linear regression nhưng polunomial regession thường được sử dụng trong mô hình đa thức, chúng ta sử dụng phương trình mũ n – là dạng tổng quát của phương trình tuyến tính

### 4.3. Hypothesis functions

Hypothesis functions là một hàm toán học giúp chương trình machine learning dự đoán và tìm ra các trọng số param của bài toán.

Công thức:

hθ(x)=θ0+θ1x2+θ2x2+...+θnxn

Để thuận tiện cho việc cài đặt và tính toán trên máy tính, người ta đã viết công thức trên lại thành dạng toán sử dụng ma trận và vector. Với θ là vector (n+1, 1) và x là ma trận (m, n+1) với m là số sample và n là số features

### 4.4. Cost function

Cost Function là một hàm dùng để đo độ chính xác của Hypothesis functions từ đó giúp tìm ra các trọng số tối ưu (các θ tốt nhất) cho Hypothesis functions.

Mean Squared Error (MSE) của Cost function là một số trung bình cộng được tính bằng cách sử dụng các cặp dữ liệu (x, y). Với mỗi cặp dữ liệu (x, y) ta sử dụng x để tạo ra dự đoán và tìm chênh lệch với y. Sau đó bình phương lên để mất dấu rồi cộng tất cả các sai số của từng dữ liệu và lấy trung bình cộng.

Ngoài ra ta còn có thể sử dụng MAE (Mean absolute error) thay thế cho MSE. Khác với MSE là lấy bình phương để loại bỏ giá trị âm thì MAE lấy giá trị tuyệt đối của sai số để loại bỏ giá trị sai số âm.

Sau khi đã có MSE thì việc tìm các trọng số tối ưu cũng chính là tìm giá trị θ để hàm MSE Cost function đạt giá trị nhỏ nhất (giá trị cực tiểu).

### 4.5. Normal Equation

Normal equation là một trong 2 cách để tìm giá trị cực tiểu của hàm J (x, θ). Normal equation sử dụng phương pháp giải đạo hàm của J (x, θ) để tìm ra giá trị cực tiểu của nó.

Công thức (dạng ma trận) của normal equation:

θ ̂ = (XT X)-1 XT y

Ưu diểm của normal equation:

- Công thức đơn giản và chỉ cần đổ dữ liệu vào để tính toán

- Không cần các vòng lặp để tính ra kết quả

Khuyết điểm của normal equation:

- Có thể bị non-invertibility

- Tài nguyên để tính toán ma trận nghịch đảo với lượng dữ liệu lớn là rất tốn kém

- Chạy rất chậm nếu số lượng features quá lớn

- Độ phức tạp (O(n3)) rất lớn nếu n lớn

### 4.6. Gradient Descent

Gradient Descent là một trong 2 cách để tìm giá trị cực tiểu của hàm J (x, θ). Gradient Descent sử dụng phương pháp tăng hoặc giảm θ trong một khoảng nhất định để tìm ra giá trị nhỏ nhất (cực tiểu) của hàm J (x, θ).

Công thức (dạng ma trận) của Gradient Descent:

Người ta đã tạo ra 3 biến thể khác nhau của Gradient Descent là BGD, SGD, Mini-Batch GD. 3 biến thể này được tạo ra dựa trên việc sử dụng số sample (m) khác nhau để tính toán.

Ưu điểm của Gradient Descent:

* Số lượng features lớn vẫn chạy tốt
* Vì không sử dụng ma trận nghịch đảo để tính toán nên việc tính toán bằng gradient Descent nhanh, nhẹ nhàng hơn so với Normal equation
* Độ phức tạp (O(kn2)) nhỏ hơn Normal equation

Nhược điểm của Gradient Descent:

* Cần nhiều vòng lặp để tính ra kết quả
* Kết quả thường không chính xác 100%

## 5. Thuật toán Support Vector Machines (SVM)

### 5.1. Khái niệm thuật toán:

Support Vector Machines (SVM) là một thuật toán Machine Learning rất mạnh và linh hoạt. Có thể xử lý nhiều bài toán, như: Classification (cả linear và nonlinear), Regression, hay cả Outlier Detection . Là một trong những thuật toán tốt nhất vào thời điểm trước khi Deep Learning ra đời.

### 5.2. SVM Classification

Thuật toán SVM là Binary Classifier, do đó chỉ dùng trong trường hợp phân loại label có 2 nhãn (0, 1).

Giới thiệu thuật ngữ:

* Margin (hay còn gọi là lề): là khoảng cách từ đường ranh giới giữa 2 class được phân loại cho đến các sample gần nhất từ 2 class trên.
* Support Vectors: là những sample nằm trên đường Margin (Sample X = [x0 x1 x2 x3 …] viết dưới dạng vector). Vì đây là những Vector phân định đường Margin.

SVM Classification được chia làm 2 loại là linear và nonlinear.

### 5.3. Linear SVM Classification

Linear SVM Classification chỉ được sử dụng khi dữ liệu có thể được phân loại bằng một đường thẳng, không thể dùng với những dataset quá phức tạp. Ý tưởng của linear SVM classification là tìm ra một đường ranh giới có margin lớn nhất có thể. Vì khi margin càng lớn, thì khả năng tổng quát hóa dữ liệu sẽ càng tốt hơn. Hay nói cách khác là sẽ mô tả được khuynh hướng của dữ liệu một cách tốt nhất.

SVM classification có thể được phân thành Hard Margin và Soft Margin.

Chart, line chart, scatter chart

Description automatically generatedHard margin chính là phương pháp chọn ra một đường ranh giới sao cho tất cả các sample của một class phải nằm hết về 1 phía của Margin.

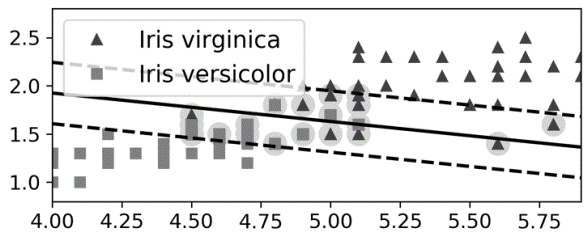
Hình 5.3.1: Hard Margin

Tuy nhiên, Hard margin lại có các vần đề sau: Thứ nhất, nếu đường ranh giới không phải đưởng thẳng thì sẽ không thể tìm được hard margin. Thứ hai, Hard margin quá nhạy cảm với các điểm dữ liệu lạ, có thể làm thay đổi model rất nặng.

Do đó, người ta đã sử dụng Soft margin để khắc phục những điểm yếu trên. Soft margin cũng tương tự như Hard margin nhưng lại cho phép một số sample nằm về phía còn lại của margin. Điều này cho phép model tìm được margin lớn nhất, và cũng khắc phục được điểm yếu của Hard margin đó là quá nhạy cảm với những điểm dữ liệu lạ. Đồng thời, cũng giảm đi hiện tượng overfitting.

### 5.4. Non-Linear SVM Classification

Cũng tương tự như linear SVM classification, nonlinear SVM classification cũng được sử dụng để phân loại 2 class, tuy nhiên đường ranh giới không nhất thiết phải là đưởng thẳng.



Hình 5.4.1: Soft Margin

Một số phương pháp để dùng nonlinear SVM classification:

* Thêm các feature bậc cao (polynomial features) vào linear SVM
* Thêm features vào model linear SVM.
* Sau đó, fit model linear SVM và đường ranh giới sẽ trở thành đường cong.

Về mặt lý thuyết chỉ cần thêm bậc của polynomial thì có thể xử lý mọi dữ liệu dù phức tạp đến bao nhiêu. Tuy nhiên trong thực tế, nếu thêm quá nhiều features sẽ dẫn đến bùng nổ số lượng features dẫn đến việc training rất lâu và không hiệu quả. Tuy nhiên, ta có thể sử dụng một kĩ thuật gọi là kernel trick để giúp đạt được hiệu ứng của polynomial model mà không cần thêm vào các feature bậc cao. Từ đó, sẽ không bị bùng nổ dữ liệu, giúp model chạy nhanh hơn và không bị underfitting.

Thêm các feature tương tự nhau (similarity features):

Similarity function là phương trình để đo độ giống nhau giữa các sample. Một số similarity function thường được dùng:

Euclidean distance (dùng để đo khoảng cách):

Để dùng phương pháp này, đầu tiên ta chọn một mốc để đo khoảng cách từ các sample đến mốc đó. Sau đó, ta thêm một feature là khoảng cách từ mọi điểm đến cột mốc. Từ đó tìm được đường ranh giới là đường thẳng theo model linear SVM classification.

String kernels (dùng để đo dữ liệu dạng chuỗi).

Gaussian Radial Basis Function (RBF): thích hợp mô tả các đường cong.

### 5.5. SVM Regression

Như đã nói ở trên, SVM là một thuật toán rất linh hoạt, nó không chỉ hoạt động với linear và nonlinear classification, mà còn hỗ trợ cho bài toán regression. Để sử dụng SVM cho bài toán regression, ta cần phải thay đổi cách tiếp cận: đó chính là cố gắng fit thật nhiều sample vào trong một margin hẹp nhất có thể, hoàn toàn ngược lại với bài toán classification.

Chart, scatter chart

Description automatically generatedTrong hàm sklearn.svm.LinearSVR, để margin nhỏ lại, ta cần phải giảm epsilon lại, epsilon càng nhỏ, margin càng hẹp.

Hình 5.5.1: SVM Regression

## 6. Thuật toán Decision Trees

### 6.1. Giới thiệu:

Là một thuật toán rất linh hoạt, có thể xử lý cả classification và regression. Đây là một thuật toán rất mạnh, có khả năng fit được cả những bộ dữ liệu rất phức tạp.

Decision trees cũng là một thành phần căn bản của thuật toán Random Forests.

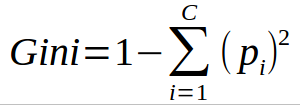
### 6.2. Decision trees

Cách hoạt động của Decision Trees rất dễ hiểu, đầu tiên dữ liệu sẽ đi qua root node, sau đó sẽ kiểm tra điều kiện, nếu đúng sẽ đi qua nút con bên trái, nếu sai sẽ đi qua nút con bên phải, dữ liệu sẽ tiếp tục đi cho đến khi đến được leaf node và được phân loại. Lấy mô hình phân loại hoa Iris ở dưới làm ví dụ, ta bắt đầu ở root node (ở trên cùng), ở nút này điều kiện là độ dài cánh hoa <= 2.45cm. Nếu đúng, sẽ đi qua nút con bên trái, trong trường hợp này, đây chính là leaf node (do không có nút con), vì vậy nó không có điều kiện nữa và nó được phân loại là hoa setosa (class = setosa). Ở trường hợp khác, nếu cánh hoa dài hơn 2.45cm, dữ liệu sẽ đi về nút con bên phải, do đây không phải leaf node, nên ta sẽ tiếp tục xét điều kiện: độ rộng cánh hoa <= 1.75cm. Nếu đáp ứng điều kiện trên, ta sẽ đi về phía nút con bên trái và được phân loại là hoa versicolor. Còn trong trường hợp còn lại, dữ liệu sẽ được phân loại là hoa virginica.

### 6.3. Impurity measurement

Độ tinh khiết của một node: độ tinh khiết chính là số lượng sample không bị lẫn vào trong một class, ở độ tinh khiết cao nhất, nghĩa là tất cả các sample đều chỉ thuộc vào một class duy nhất. Gini càng thấp thì độ tinh khiết của node càng cao (cao nhất khi Gini = 0). Lấy ví dụ ở mô hình trên, ta có thể thấy được gini ở class setosa là 0, điều đó nghĩa là trong class đó chỉ tồn tại các sample thuộc class setosa và không lẫn lộn các sample của class khác, hay nói cách khác, thuật toán đã phân loại được hoa setosa với độ chính xác là 100%.

Cách tính gini:



Với:

* C là số lượng class.
* pi là tỉ lệ của class thứ i (value / samples).

**Entropy:**

* Ở nhiệt động lực học: entropy nghĩa là độ hỗn loạn của các phân tử.
* Ở lý thuyết thông tin: entropy nghĩa là độ khác nhau giữa các thông tin.
* Ở machine learning: dùng để đo impurity của một node. (impurity thấp nhất khi entropy = 0)

Công thức tính Entropy:

Diagram, schematic

Description automatically generated

Với:

* Hi là entropy của node thứ i.
* pi là tỉ lệ của class thứ i (value / samples).
* k là số lượng class.

Vậy thì ta nên dùng Gini hay entropy để tính impurity? Trong thực tế, ở nhiều trường hợp sẽ không có sự khác biệt đáng kể, cả 2 loại đều đưa đến những cái cây như nhau. Gini sẽ tính toán nhanh hơn một tí, trong khi entropy sẽ đem lại một cái cây cân bằng hơn.

### 6.4. Thuật toán CART - Classification and Regression Tree

Để tạo nên Decision trees một cách tối ưu nhất, người ta đã nghĩ ra nhiều thuật toán để làm điều này. Một trong số đó là thuật toán CART. Ý tưởng của thuật toán CART là chọn ra một feature nào đó để chọn ra một threshold, chia feature thành 2 set con sao cho impurity giảm xuống thấp nhất, sau đó lại tiếp tục chọn ra một feature để chia tiếp, cho đến khi đạt đến độ sâu tối đa, hoặc đến khi impurity không thể giảm được nữa.

Text

Description automatically generatedCost function của CART:

Với:

* Gleft/right là impurity ở set con bên trái/phải.
* mleft/right là số sample ở set con bên trái/phải.

Để giảm cost function, thuật toán sẽ tập trung giảm trọng số ở bên có số sample lớn hơn.

### 6.5. Ưu điểm và khuyết điểm của Decision Trees:

Decision trees là một thuật toán đơn giản, dễ hiểu, có độ linh hoạt cao. Nó cũng là một thuật toán rất mạnh, có độ chính xác cao.Tuy nhiên, Decision trees rất nhạy cảm với việc xoay dữ liệu (do các model học ra có dạng bậc thang), và có độ ổn định của đầu ra thấp.

Do decision trees chỉ có thể học ra những đường vuông góc với các trục của feature. Nên khi xoay dữ liệu, sẽ bị ảnh hưởng đến đầu ra. Để khắc phục điểm yếu này, ta có thể sử dụng thuật toán PCA để xoay dữ liệu theo hướng tốt nhất cho decision trees.

Khi training, các feature được decision trees chọn một cách ngẫu nhiên. Do đó, độ ổn định của đầu ra là rất thấp. Điều này đã được khắc phục ở thuật toán random forests.

## 7. Ensemble learning and Random Forests

### 7.1. Khái niệm:

Random Forests chính là thuật toán vượt trội hơn nhiều so với Decision trees. Để tập hợp các thuật toán lại với nhau thì ta có rất nhiều phương pháp. Trong đó, ta có thể chia training set thành nhiều tập con, và cho mỗi thuật toán nhỏ để train, sau đó tổng hợp lại thành kết quả cuối cùng. Và đây cũng là phương pháp mà thuật toán Random Forests sử dụng.

Một số các phương pháp phổ biến để thực hiện Ensemble learning: Voting, Boosting, Stacking.

### 7.2. Voting method

Ý tưởng của voting method:

* Đầu tiên, ta sẽ train nhiều predictor khác nhau một cách độc lập.
* Kế đến, chọn ra một kết quả được nhiều preditor lựa chọn nhất.

Hard voting classifier: chọn theo kết quả được nhiều predictor lựa chọn nhất.

Ngạc nhiên thay, dù cho mỗi classifier có thể tệ, nhưng khi kết hợp lại sẽ cho ra kết quả tốt. Miễn là có thể đáp ứng 2 điều kiện sau: đầu tiên là phải đủ số lượng. Thứ hai, các classifier phải có sự đa dạng, khác biệt với nhau để có thể bù trừ khuyết điểm cho nhau. Ở đây, ta có thể tạo ra sự khác biệt bằng cách dùng training set khác nhau, hoặc có thể dùng cả thuật toán khác nhau.

Soft voting classifier: Mỗi classifier đều cho ra được xác suất của mỗi class, Soft voting sẽ tính toán xác suất dự đoán trung bình của mỗi class rồi mới quyết định. Soft voting có thể sẽ đem lại hiệu năng tốt hơn so với hard voting nhưng nó có tốc độ chậm hơn so với hard voting.

### 7.3. Bagging method

Như đã nói ở trên, để tạo ra sự khác biệt bằng cách có thể dùng cả thuật toán khác nhau, hoặc dùng training set khác nhau. Nếu ta dùng các training set khác nhau sau đó tổng hợp kết quả lại với nhau, ta có thể gọi đây là phương pháp bagging (bootstrap aggregating).

Vậy thì làm sao để lấy ra các training set khác nhau? Ta có thể thực hiện chọn ra các sampling ngẫu nhiên theo 2 cách là Bootstrapping (with replacement) và Non-bootstrapping (without replacement). Với phương pháp Bootstrapping sẽ cho phép ta thực hiện train được nhiều predictor cùng một lúc.

Out-of-bag evaluation: training set được chọn ra một cách ngẫu nhiên, do đó sẽ còn lại một số lượng các sample chưa từng được train. Qua đó, ta có thể chọn ra để làm validation / test set mà không cần phải chọn ra validation / test set từ đầu.

### 7.4. Boosting method

Ở bagging method, khi thực hiện train nhiều model cùng một thời điểm, các model được train cùng lúc có thể sẽ không khắc phục được điểm yếu của nhau. Để khắc phục khuyết điểm trên, người ta dùng boosting method. Ý tưởng của boosting method chính là để chắc chắn model được train sau khắc phục được khuyết điểm của các model trước đó.

Một số thuật toán boosting nổi tiếng: AdaBoost, Gradient boost.

**AdaBoost:**

• Ý tưởng: tập trung vào những sample bị phân loại sai, sau đó train các model sau theo hướng phân loại đúng những sample đó.

• Để thực hiện ý tưởng trên, khi những sample bị phân loại sai thì những sample đó sẽ được tăng trọng số. Khi đó, lần train tiếp theo sẽ tập trung vào phân loại những sample có trọng số cao hơn.

**Gradient Boosting:**

Như AdaBoost, Gradient Boosting sẽ thêm những predictor để sửa lỗi những preditor trước đó. Tuy nhiên, thay vì tăng trọng số của những sample lỗi, Gradient Boosting sử dụng residual error (lỗi của phần dư): là phần sai khác giữa label thật và kết quả dự đoán ra. Gradient Boosting sẽ dùng residual error để làm label cho predictor kế tiếp. Sau đó sẽ lấy tổng của các prediction làm kết quả.

# CHƯƠNG 3: GIỚI THIỆU SẢN PHẨM ĐỀ TÀI - ỨNG DỤNG HỌC MÁY TRONG DỰ ĐOÁN MỨC LƯƠNG NGÀNH CÔNG NGHỆ THÔNG TIN

## 1. Giới thiệu đề tài

## 2. Kết quả thực nghiệm

## 3. Nhận xét và đánh giá về đề tài

### 3.1. Kết quả đạt được

### 3.2. Hạn chế

### 3.3. Hướng phát triển và mở rộng

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Trần Nhật Quang (2021), *Bài giảng học phần Machine Learning*, Trường Đại học Sư phạm Kỹ thuật Tp.HCM
2. Kevin P.Murphy, Machine Learning: A Priobabilistic Perspective, The MIT Press Cambridge, Massachutsetts London, England

# LỜI CẢM ƠN

Để hoàn thành đề tài này, nhóm em xin gửi lời cảm ơn chân thành đến:

Ban giám hiệu trường Đại Học Sư phạm kỹ thuật Thành phố Hồ Chí Minh vì đã tạo mọi điều kiện về cơ sở vật chất, nền tảng trực tuyến, hiện đại, đa dạng các loại sách, tài liệu thuận lợi cho việc tìm kiếm, nghiên cứu thông tin.

Xin cảm ơn giảng viên bộ môn - Thầy Trần Nhật Quang đã giảng dạy tận tình, chi tiết để nhóm có đủ kiến thức và vận dụng chúng vào đề tài này.

Do chưa có nhiều kinh nghiệm làm để tài cũng như còn những hạn chế về kiến thức, nên đề tài chắc chắn sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự nhận xét, ý kiến đóng góp từ phía Thầy để đề tài của nhóm được hoàn thiện hơn.

Lời cuối cùng, chúng em xin kính chúc thầy có nhiều sức khỏe, thành công và hạnh phúc.