TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

**---------🙞🙜🕮🙞🙜---------**



ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**XÂY DỰNG MÔ HÌNH DỰ BÁO MỘT SỐ CHỈ SỐ KINH TẾ TẠI VIỆT NAM SỬ DỤNG HỌC MÁY.**

Sinh viên thực hiện: Trần Văn Mạnh

Giảng viên hướng dẫn: TS. Nguyễn Thọ Thông

HÀ NỘI, NĂM 2023

# MỤC LỤC

[MỤC LỤC 2](#_Toc150253869)

[DANH MỤC HÌNH ẢNH 3](#_Toc150253870)

[DANH MỤC BẢNG BIỂU 3](#_Toc150253871)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ 3](#_Toc150253872)

[I. TỔNG QUAN CƠ SỞ LÝ THUYẾT 3](#_Toc150253873)

[**1.1** **Ngôn ngữ lập trình python** 3](#_Toc150253874)

[**1.2** **Tổng quan về học máy** 3](#_Toc150253875)

[**1.3** **Tổng quan về một số thư viện python hỗ trợ học máy** 3](#_Toc150253876)

[**1.4** **Tổng quan về tầm quan trọng của dự báo kinh tế** 3](#_Toc150253877)

[**1.5** **Đối tượng và phạm vi nghiên cứu** 3](#_Toc150253878)

[**1.6** **Tổng kết công nghệ sử dụng** 3](#_Toc150253879)

[II. XÂY DỰNG MÔ HÌNH 3](#_Toc150253880)

[**2.1** **Thu thập dữ liệu** 3](#_Toc150253881)

[**2.2** **Tiền xử lý dữ liệu** 3](#_Toc150253882)

[**2.3** **Lựa chọn mô hình học máy** 3](#_Toc150253883)

[**2.4** **Xây dựng và tối ưu hóa mô hình** 3](#_Toc150253884)

[**2.5** **Huấn luyện mô hình** 3](#_Toc150253885)

[**2.6** **Đánh giá mô hình** 3](#_Toc150253886)

[III. THỰC NGHIỆM, SO SÁNH VÀ ĐÁNH GIÁ 3](#_Toc150253887)

[**3.1** **Kết quả đạt được** 3](#_Toc150253888)

[**3.2** **Hạn chế** 3](#_Toc150253889)

[**3.3** **Hướng phát triển** 3](#_Toc150253890)

[Kết Luận 3](#_Toc150253891)

[**3.4** **Tóm tắt kết quả nghiên cứu** 3](#_Toc150253892)

[**3.5** **Đánh giá tổng quan về hiệu suất mô hình** 3](#_Toc150253893)

[**3.6** **Đề xuất hướng phát triển tương lai** 3](#_Toc150253894)

[TÀI LIỆU THAM KHẢO 3](#_Toc150253895)

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG BIỂU

# DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

# TỔNG QUAN CƠ SỞ LÝ THUYẾT

* 1. **Lý do chọn đề tài**

Trong bối cảnh nền kinh tế đang phát triển đối diện với nhiều biến đổi phức tạp tại Việt Nam và cũng vì mối quan tâm đặc biệt đối với lĩnh vực kinh tế và sự hứng thú với ứng dụng của học máy trong việc dự đoán một số chỉ số kinh tế quan trọng. Do vậy việc phát triển một mô hình dự báo có thể giúp em hiểu sâu hơn về tương quan giữa các yếu tố kinh tế và dữ liệu. Trong phạm vi đồ án này có thể ứng dụng việc dự báo kinh tế và quản lý tài chính, giúp các quyết định kinh tế và chính trị được dựa trên dữ liệu và dự đoán chính xác hơn. Việc sử dụng các mô hình học máy sẽ giúp cung cấp thông tin quan trọng cho việc đưa ra quyết định chính sách và kế hoạch kinh tế. Vì những lý do trên nên em đã chọn đề tài “xây dựng mô hình dự báo một số chỉ số kinh tế tại Việt Nam sử dụng học máy”.

* 1. **Ngôn ngữ lập trình python**
     1. **Python là gì?**

Python được tạo ra bởi Guido van Rossum và lần đầu tiên được giới thiệu vào năm 1991. Là một ngôn ngữ lập trình bậc cao, đa mục đích, và hướng đối tượng. Thường được sử dụng để xây dựng trang web, phần mềm, phân tích dữ liệu, học máy…

* + 1. **Một số ứng dụng của python**
       1. **Phát triển web**

Python thường được sử dụng để phát triển back-end của trang web hoặc ứng dụng thông qua một số framework như Flask, Django.

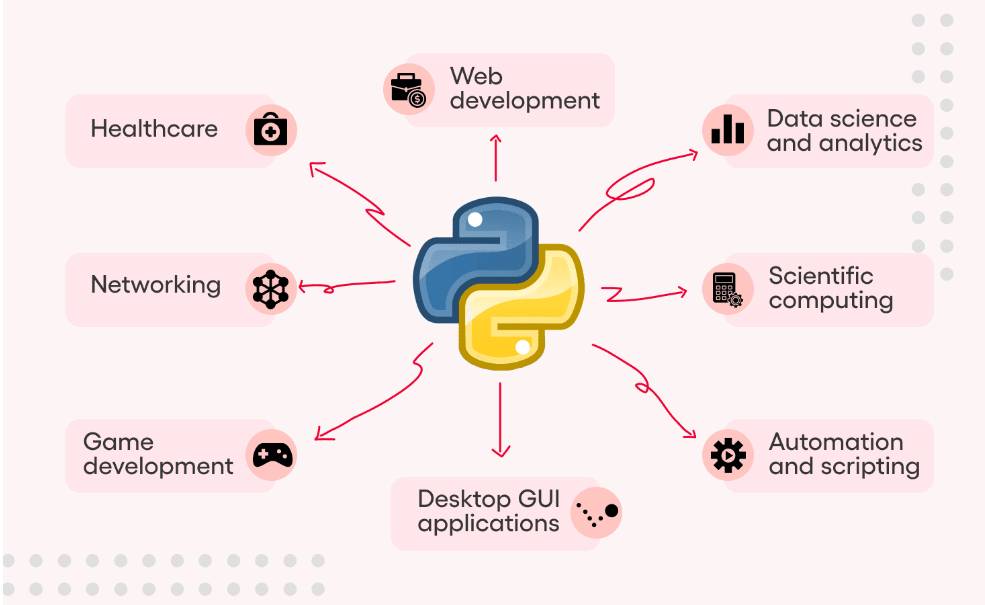
* + - 1. **Khoa học dữ liệu và học máy**

Python cực kỳ phù hợp với việc vận dụng dữ liệu, phân tích triển khai các mô hình thuật toán phức tạp. Có thể thu thập và mô hình hóa dữ liệu được. Xử lý, làm sách dữ liệu, vẽ biểu đồ để trực quan hóa dữ liệu, phân tích thống kê…

* + - 1. **Trí tuệ nhân tạo**

Python đóng một vai trò quan trọng trong lĩnh vực Trí tuệ Nhân tạo (AI) với sự đa dạng của ngôn ngữ và các thư viện hỗ trợ. Ứng dụng trong Xử lý Ngôn ngữ Tự nhiên (NLP) được sử rộng rãi trong phát triển ứng về chatbot, dịch ngôn ngữ. Ứng dụng trong xử lý ảnh và thị giác máy tính để nhận diện đối tượng, khuôn mặt, hình ảnh. Ứng dụng trong thực tế ảo (VR) và thực tế ảo mở rộng (AR)…

* + - 1. **Trích xuất dữ liệu từ website**

Python là một công cụ mạnh mẽ cho việc trích xuất dữ liệu từ các trang web, và có nhiều thư viện hỗ trợ cho mục đích này như BeautifulSoup, Scrapy,…Có thể sử dụng dữ liệu với nhiều mục đích khác nhau như thu thập dữ liệu thị trường, Phân tích dữ liệu thời gian thực, dự đoán và dự báo trong các lĩnh vực kinh tế, chính trị, y tế,…

Hình 1.1: Các ứng dụng của python

* 1. **Tổng quan về học máy**
     1. **Khái niệm về học máy**

Học máy ( Machine Learning) là một nhánh nhỏ của ngành trí tuệ nhân tạo (AI) và khoa học máy tính. Nó tập trung vào việc xử lý data và thuật toán để bắt chước cách mà con người học. Từ đó cải thiện độ chính xác của các dự đoán mà nó đưa ra.

Học máy là một thành phần vô cùng quan trọng trong sự phát triển của khoa học dữ liệu. Thông qua việc sử dụng các phương pháp thống kê và thuật toán, chúng được huấn luyện để đưa ra các phân loại, dự đoán và khám phá những tri thức mới trong các dự án khai phá dữ liệu. Những tri thức này sẽ tác động đến những quyết định trong các ứng dụng và các doanh nghiệp.

Khi internet ngày càng phổ biến, mọi người trong số vẫn thường xuyên lên mạng để cập nhập tin tức, giải trí, mua sắm,... Chính những hoạt động này đã sản sinh ra một lượng lớn dữ liệu. Khi dữ liệu lớn tiếp tục mở rộng và phát triển, nhu cầu của thị trường với các nhà khoa học dữ liệu ngày càng tăng. Nó cũng đòi hỏi nhà khoa học phải hỗ trợ xác định các câu hỏi của doanh nghiệp và đưa ra quyết định phù hợp nhất.

Học máy là một nhánh nghiên cứu rất quan trọng của trí tuệ nhân tạo với khá nhiều ứng dụng thành công trong thực tế. Hiện nay, học máy là một trong những lĩnh vực phát triển mạnh nhất của trí tuệ nhân tạo. Có một số lý do giải thích cho sự cần thiết và phát triển của học máy:

- Thứ nhất, rất khó xây dựng hệ thống thông minh có thể thực hiện các công việc liên quan đến trí tuệ như thị giác máy, xử lý ngôn ngữ tự nhiên mà không sử dụng tới kinh nghiệm và quá trình học. Thông thường, khi viết chương trình, cần có thuật toán rõ ràng để chuyển đổi đầu vào thành đầu ra. Tuy nhiên, trong nhiều bài toán, rất khó để xây dựng được thuật toán như vậy. Như trong ví dụ về nhận dạng chữ ở trên, người bình thường có khả năng nhận dạng các chữ rất tốt nhưng rất khó để giải thích vì sao từ đầu vào là ảnh lại kết luận được đây là ký tự cụ thể nào. Học máy cho phép tìm ra giải pháp cho những trường hợp như vậy dựa trên dữ liệu, chẳng hạn bằng cách tìm ra điểm chung và riêng giữa rất nhiều ảnh chụp các ký tự.

* Thứ hai, nhiều ứng dụng đòi hỏi chương trình máy tính phải có khả năng thích nghi. Ví dụ, hành vi mua sắm của khách hàng có thể thay đổi theo thời điểm cụ thể trong ngày, trong năm, hoặc theo tuổi tác. Việc xây dựng thuật toán cố định cho những ứng dụng cần thích nghi và thay đổi là không phù hợp. Học máy mang lại khả năng thích nghi nhờ phân tích dữ liệu thu thập được.
* Thứ ba, việc tìm được chuyên gia và thu thập được tri thức cần thiết cho việc thiết kế thuật toán để giải quyết các vấn đề tương đối khó, trong khi dữ liệu ngày càng nhiều và có thể thu thập dễ dàng hơn. Khả năng lưu trữ và tính toán của máy tính cũng ngày càng tăng, cho phép thực hiện thuật toán học máy trên dữ liệu có kích thước lớn.
* Cuối cùng, bản thân khả năng học là một hoạt động trí tuệ quan trọng của con người, do vậy học tự động hay học máy luôn thu hút được sự quan tâm khi xây dựng hệ thống thông minh.



Hình 1.2: giới thiệu về Machine learning

* + 1. **Một số ứng dụng của học máy**
       1. **Tự động phân loại**

Phân loại tin tức là một ứng dụng điểm chuẩn khác của phương pháp học máy.Vận dụng như thế nào? Như một vấn đề thực tế là bây giờ khối lượng thông tin đã tăng lên rất nhiều trên web. Tuy nhiên, mỗi người có sở thích hoặc lựa chọn cá nhân của mình. Vì vậy, để chọn hoặc thu thập một phần thông tin phù hợp trở thành một thách thức đối với người dùng từ vô số nội dung trên trang web.

Phân loại các danh mục một cách rõ ràng, dễ điều hướng giúp cho các khách hàng mục tiêu chắc chắn sẽ tăng khả năng truy cập các trang tin tức. Hơn nữa, độc giả hoặc người dùng có thể tìm kiếm tin tức cụ thể một cách hiệu quả và nhanh chóng.

Có một số phương pháp học máy trong mục đích này, tức là, máy vectơ hỗ trợ, naive Bayes, k-nearest neighbor, v.v.

* + - 1. **Ứng dụng trong mạng xã hội**

Học máy đang được sử dụng trong một loạt các ứng dụng ngày nay. Một trong những ví dụ nổi tiếng nhất là Facebook News Feed. Nguồn cấp tin tức sử dụng học máy để cá nhân hóa từng nguồn cấp dữ liệu thành viên.

Nếu một thành viên thường xuyên dừng lại để đọc hoặc thích một bài đăng của một người bạn cụ thể, News Feed sẽ bắt đầu hiển thị nhiều hơn về hoạt động của người bạn đó trước đó trong nguồn cấp dữ liệu.

Đằng sau hệ thống ấy, phần mềm sử dụng phân tích thống kê và phân tích dự đoán để xác định các mẫu trong dữ liệu người dùng và sử dụng các mẫu đó để điền vào News Feed. Nếu thành viên không còn dừng lại để đọc, thích hoặc bình luận trên các bài đăng của bạn bè, dữ liệu mới đó sẽ được bao gồm trong tập dữ liệu và News Feed sẽ điều chỉnh tương ứng.

Không chỉ riêng facebook, ta có thể bắt gặp những tính năng tương tự đó qua các mạng xã hội khác như google, instagram,...

* + - 1. **Nhận diện hình ảnh**

Nhận dạng hình ảnh là một trong những ví dụ về máy học và trí tuệ nhân tạo phổ biến nhất. Về cơ bản, nó là một cách tiếp cận để xác định và phát hiện các đặc trưng của một đối tượng trong hình ảnh kỹ thuật số. Hơn nữa, kỹ thuật này có thể được sử dụng để phân tích sâu hơn, chẳng hạn như nhận dạng mẫu, nhận diện hình khuôn, nhận dạng khuôn mặt, nhận dạng ký tự quang học và nhiều hơn nữa,..

* + - 1. **Ứng dụng cho xe tự lái**

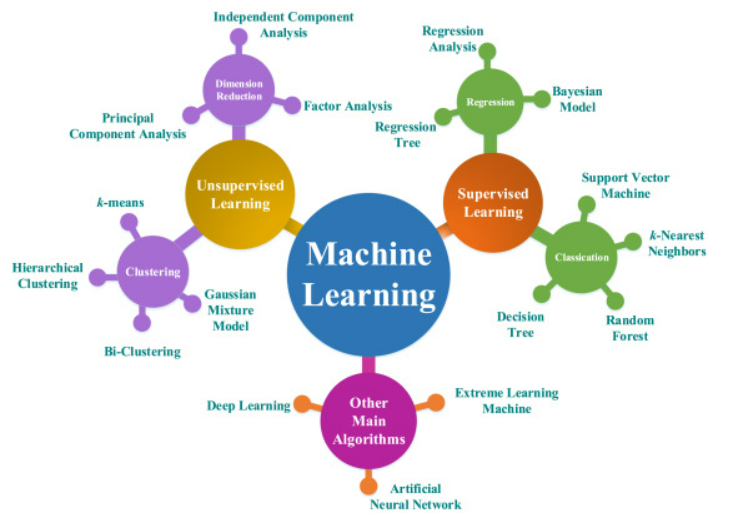
Một trong những ứng dụng thú vị nhất của Machine learning là ô tô tự lái. Machine learning đóng một vai trò quan trọng trong việc vận hành ô tô tự lái. Tesla, công ty sản xuất xe hơi nổi tiếng đang tiến hành các nghiên cứu về xe tự lái bằng phương pháp học không giám sát (unsupervised learning method )để đào tạo các mô hình ô tô tự động phát hiện ra người và đồ vật khi đang lái xe.

* + - 1. **Phân tích và dự báo một số lĩnh vực**

Machine Learning cũng được sử dụng trong các hệ thống dự đoán như dự đoán các chỉ số kinh tế, chuẩn đoán bệnh trong y học bằng cách phân tích các biểu hiện lâm sàng và kết hợp chúng để tiên đoán về sự tiến triển của bệnh tật, phân tích chứng khoán để có được các chiến lược chênh lệch giá,…

* 1. **Một số dạng học máy**

Khi thiết kế và xây dựng hệ thống học máy cần quan tâm tới những yếu tố sau.

* Thứ nhất, kinh nghiệm hoặc dữ liệu cho học máy được cho dưới dạng nào?
* Thứ hai, lựa chọn biểu diễn cho hàm đích ra sao? Hàm đích có thể biểu diễn dưới dạng hàm đại số thông thường nhưng cũng có thể biểu diễn dưới những dạng khác như dạng cây, dạng mạng nơ ron, công thức xác suất .v.v

Hình 1.3: Các dạng học máy

* + 1. **Học có giám sát (supervised learning)**

Là học dựa trên tập dữ liệu có gán nhãn, thường dùng cho bài toán phân lớp. là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (*input, outcome*) đã biết từ trước.

Cặp dữ liệu này còn được gọi là (*data, label*), tức (*dữ liệu, nhãn*) từ đây máy sẽ tạo ra nhiều hiểu biết về vật thể đó, sau đó chúng ta sẽ đưa ra các mẫu kiểm tra và xem máy trả lời kết quả(tìm ra giá trị đầu ra). Nếu giá trị đầu ra là rời rạc thì học có giám sát được gọi là phân loại hay phân lớp (classification). Nếu đầu ra nhận giá trị liên tục, tức đầu ra là số thực, thì học có giám sát được gọi là hồi quy (regression).

Cách học này gần gũi nhất chính là nhận diện khuôn mặt trong Facebook, lúc đầu chúng ta sẽ thực hiện tag các người quen trong ảnh, dựa trên các Tag này Facebook sẽ biết khuôn mặt đó là ai, sau đó các bức ảnh tương lai máy có thể tự tag được khuôn mặt tự động.

* + 1. **Học không giám sát (unsupervised learning)**

Là học dựa trên tập dữ liệu không gán nhãn, thường dùng cho bài toán phân cụm. Tức là  chúng ta không biết được giá trị đầu ra *(outcome)* hay nhãn *(lable)* mà chỉ có dữ liệu đầu vào tuy nhiên thay vì xác định giá trị đầu ra thì phương pháp này dựa trên độ tương tự, tương đồng giữa các dữ liệu để cần phân nhóm điểm tương đồng với nhau.

Loại học này chúng ta có thể trải nghiệm trên Netflix, máy sẽ dựa trên các phim chúng ta đã xem, sau đó tìm các phim tương tự có tính tương đồng với đó để đề xuất, hoặc là tìm kiếm trên google.

* + 1. **Học bán giám sát (semi-supervised learning)**

Là một phương pháp giữa giữa học máy giám sát và học máy không giám sát nghĩa là học dưạ trên cả dữ liệu gán nhãn lẫn không gán nhãn. Các bài toán khi chúng ta có một lượng lớn dữ liệu nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning.

Ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh/văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao.

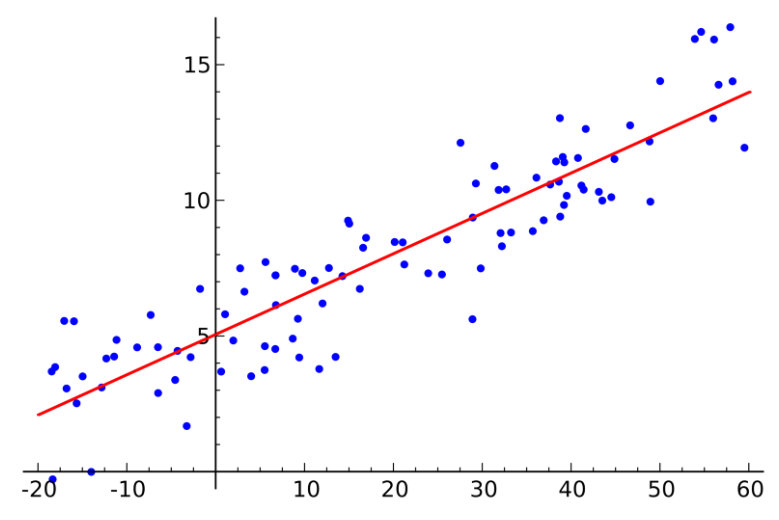
* + 1. **Học tăng cường (reinforcement learning)**

Là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất nghĩa là cho máy các hành động, sau đó tiếp nhận sự thay đổi trạng thái quan sát và kèm theo là phần thưởng để máy biết hành động đó là tốt hay là xấu, theo thời gian máy sẽ đưa ra các hành động tốt dần theo thời gian. Tóm lại học tăng cường chủ yếu là học chuỗi các phản ứng nhằm tối đa hóa lợi ích.

Học tăng cường chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.

* 1. **Một số thuật toán, mô hình trong học máy**
     1. **Linear Regression( hồi quy tính tuyến)**

Nếu bạn biết thống kê, bạn có thể đã nghe nói về hồi quy tuyến tính trước đây. *Bình phương nhỏ nhất* là một phương pháp để thực hiện hồi quy tuyến tính. Bạn có thể suy nghĩ về hồi quy tuyến tính như là nhiệm vụ kẻ một đường thẳng đi qua một tập các điểm. Có rất nhiều chiến lược có thể thực hiện được, và chiến lược “bình phương nhỏ nhất” sẽ như thế này – Bạn có thể vẽ một đường thẳng, và sau đó với mỗi điểm dữ liệu, đo khoảng cách thẳng đứng giữa điểm và đường thẳng. Đường phù hợp nhất sẽ là đường mà các khoảng cách này càng nhỏ càng tốt.

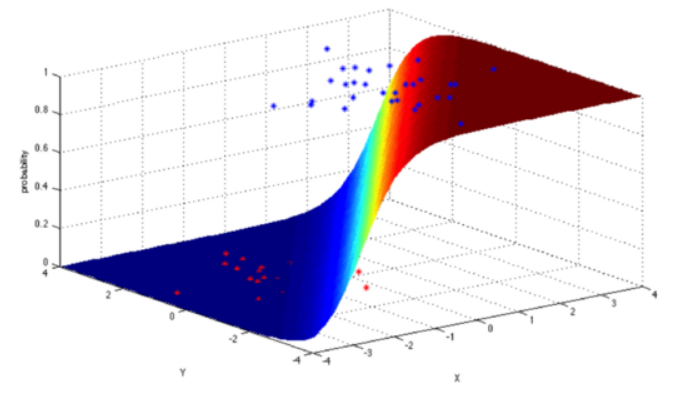


Hình 1.4: Hồi quy tuyến tính

Một số ví dụ là người ta có thể sử dụng mô hình này để dự đoán giá cả (nhà đất, chứng khoán), điểm số,…

* + 1. **Logictic Regression ( hồi quy logistic)**

Hồi quy logistic là một cách thống kê mạnh mẽ để mô hình hóa một kết quả nhị thức với một hoặc nhiều biến giải thích. Nó đo lường mối quan hệ giữa biến phụ thuộc phân loại và một hoặc nhiều biến độc lập bằng cách ước tính xác suất sử dụng một hàm logistic, là sự phân bố tích lũy logistic.

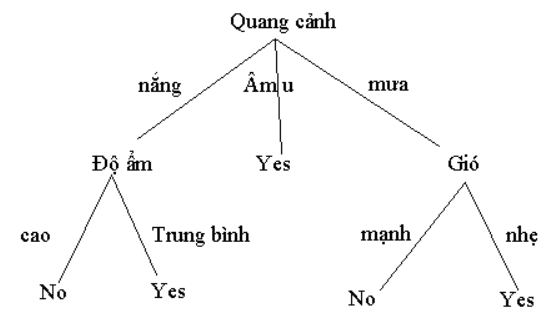


Hình 1.5 Logistic Regression

Thuật toán này được sử dụng trong một số trường hợp:

* Điểm tín dụng ( quyết định có cho khách hàng vay vốn hay không)
* Đo mức độ thành công của chiến dịch marketing
* Dự đoán doanh thu của một sản phẩm nhất định
* Dự đoán động đất …
  + 1. **Decision tree( cây quyết định)**

Cây quyết định là công cụ hỗ trợ quyết định sử dụng biểu đồ dạng cây hoặc mô hình của các quyết định và kết quả có thể xảy ra của chúng, bao gồm kết quả sự kiện ngẫu nhiên, chi phí tài nguyên và lợi ích. Dưới đây là một ví dụ điển hình của cây quyết định:



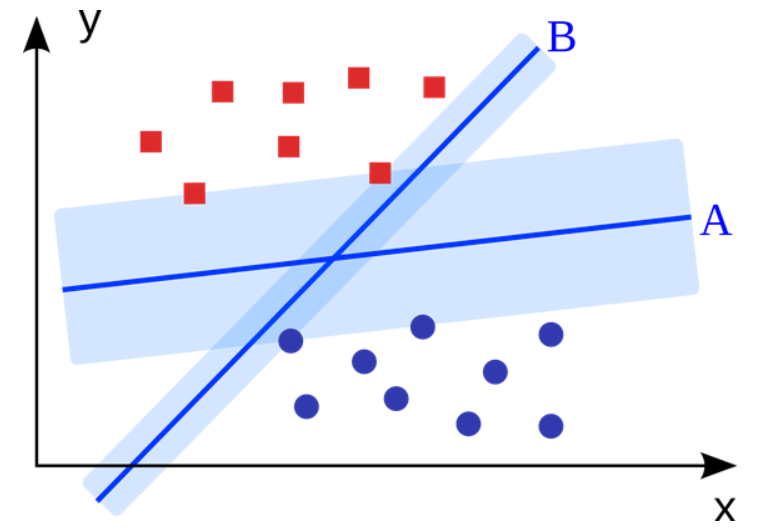
Hình 1.6. Cây quyết định

Cây quyết định này cho ta gợi ý về việc có đi đá bóng hay không. Ví dụ, quang cảnh có nắng, độ ẩm trung bình thì tôi sẽ đi đá bóng. Ngược lại, nếu trời mưa, gió mạnh thì tôi sẽ không đi đá bóng nữa.

Cây quyết định tuy là mô hình khá cũ, khá đơn giản những vẫn còn được ứng dụng khá nhiều và hiệu quả. Đứng dưới góc nhìn thực tế, cây quyết định là một danh sách tối thiểu các câu hỏi dạng yes/no mà người ta phải hỏi, để đánh giá xác suất đưa ra quyết định đúng đắn.

* + 1. **Support Vector machine( SVM)**

SVM là phương pháp phân loại nhị phân. Cho một tập các điểm thuộc 2 loại trong môi trường N chiều, SVM cố gắng tìm ra N-1 mặt phẳng để phân tách các điểm đó thành 2 nhóm. Ví dụ, cho một tập các điểm thuộc 2 loại như hình bên dưới, SVM sẽ tìm ra một đường thẳng nhằm phân cách các điểm đó thành 2 nhóm sao cho khoảng cách giữa đường thẳng và các điểm xa nhất có thể



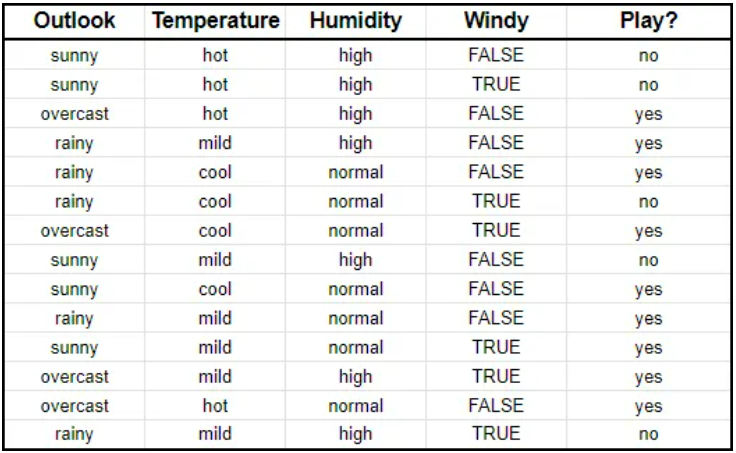
Hình 1.7.SVM

Xét về quy mô, một số vấn đề lớn nhất đã được giải quyết bằng cách sử dụng SVM (với việc thực hiện sửa đổi phù hợp) ví dụ như hiển thị quảng cáo, phát hiện giới tính dựa trên hình ảnh, phân loại hình ảnh có quy mô lớn …

* + 1. **Naive Bayes**

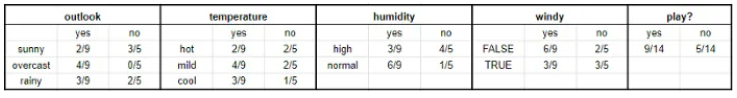
Naive Bayes là một thuật toán phân loại. Điều này có nghĩa là Naive Bayes được sử dụng khi biến đầu ra là rời rạc.

Naive Bayes có vẻ là một thuật toán khó vì nó yêu cầu kiến thức toán học sơ bộ về xác suất có điều kiện và Định lý Bayes, nhưng đó là một khái niệm cực kỳ đơn giản và “ngây thơ”. Hãy xem giải thích về thuật toán này bằng một ví dụ:



Giả sử có dữ liệu đầu vào về các đặc điểm của chỉ số kinh tế (triển vọng – outlook, nhiệt độ – temperature, độ ẩm – Humidity, gió – Windy) và liệu bạn có chơi gôn hay không (tức là cột cuối cùng).

Điều mà Naive Bayes làm về cơ bản là so sánh tỷ lệ giữa mỗi biến đầu vào và các danh mục trong biến đầu ra. Điều này có thể được hiển thị trong bảng dưới đây.



Lấy một ví dụ trong bảng trên để bạn hiểu, trong phần nhiệt độ, trời nóng trong hai ngày trong số chín ngày bạn chơi gôn (tức là có).

Theo thuật ngữ toán học, bạn có thể viết giá trị này là xác suất trời nóng khi bạn chơi gôn . Kí hiệu toán học là P (hot | yes). Đây được gọi là xác suất có điều kiện và là điều cần thiết để hiểu phần còn lại của những gì sắp xem sau đây.

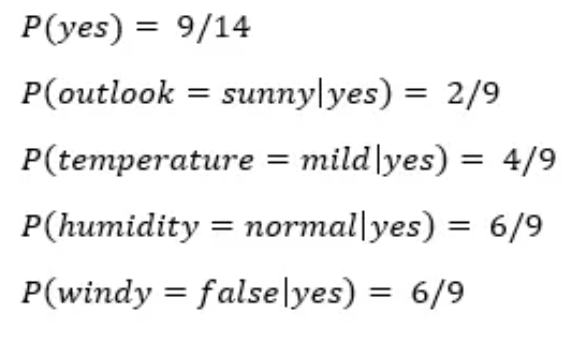
Sau khi có được điều này, bạn có thể dự đoán liệu mình có chơi gôn hay không với bất kỳ sự kết hợp nào của các đặc điểm chỉ số kinh tế.

Hãy tưởng tượng rằng có một ngày mới với những đặc điểm sau:

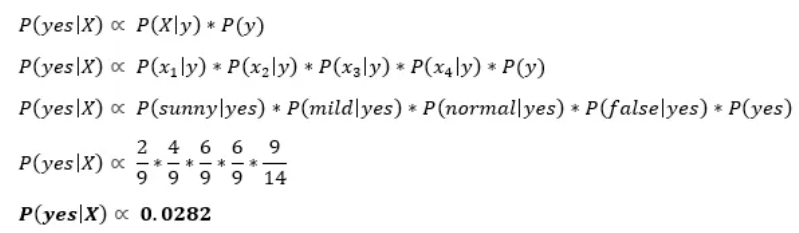
* Triển vọng: nắng
* Nhiệt độ: ôn hòa
* Độ ẩm: bình thường
* Gió: không

Đầu tiên, chúng tôi sẽ tính xác suất bạn sẽ chơi gôn với X, P (yes | X), tiếp theo là xác suất bạn sẽ không chơi gôn với X, P (no | X).

Sử dụng biểu đồ trên, có thể nhận được thông tin sau:



Bây giờ có thể chỉ cần nhập thông tin này vào công thức sau:



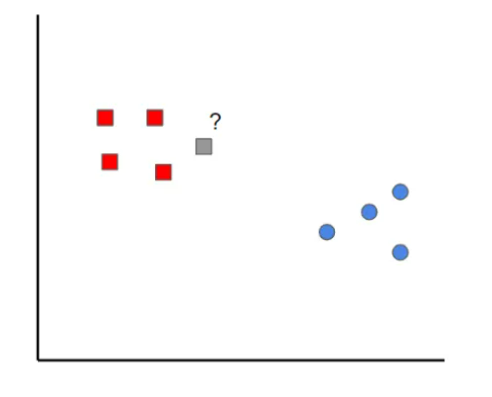
Tương tự, bạn sẽ hoàn thành trình tự các bước tương tự cho P (no | X)



Vì P (yes | X)> P (no | X), nên bạn có thể dự đoán rằng người này sẽ chơi gôn với điều kiện là trời nắng, nhiệt độ ôn hòa, độ ẩm bình thường và không có gió.

Đây là bản chất của Naive Bayes

* + 1. **K-Nearest Neighbors( KNN)**



Hình 1.8 K-Nearest

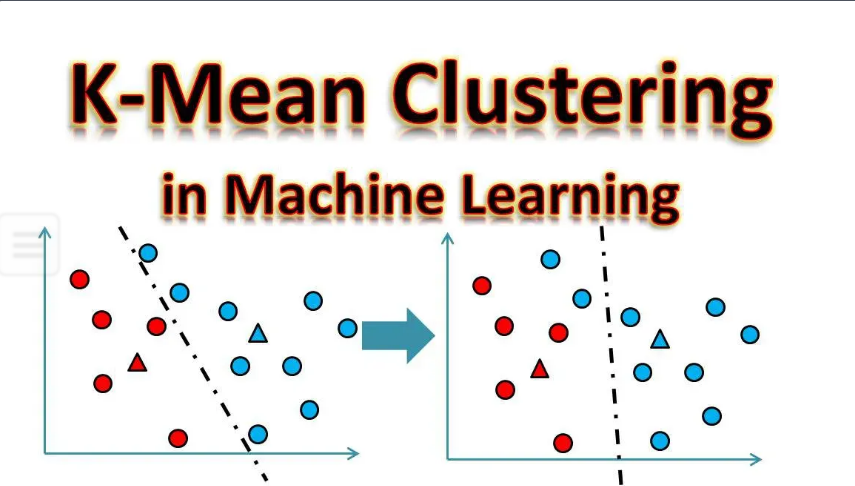
K-nearest neighbors là một ý tưởng đơn giản. Đầu tiên, bạn bắt đầu với dữ liệu đã được phân loại (tức là các điểm dữ liệu màu đỏ và xanh lam). Sau đó, khi bạn thêm một điểm dữ liệu mới, bạn phân loại nó bằng cách xem các điểm được phân loại gần k nhất. Lớp nào nhận được nhiều phiếu bầu nhất sẽ quyết định điểm mới được xếp vào loại nào.

Trong trường hợp này, nếu đặt k = 1, có thể thấy rằng điểm gần nhất đầu tiên với mẫu màu xám là một điểm dữ liệu màu đỏ. Do đó, điểm sẽ được phân loại là màu đỏ.

Một điều cần lưu ý là nếu giá trị của k được đặt quá thấp, nó có thể bị ngoại lệ. Mặt khác, nếu giá trị của k được đặt quá cao thì nó có thể bỏ qua các lớp chỉ có một vài mẫu

* + 1. **K-Means**

K-Means là một thuật toán học tập không giám sát để giải quyết các vấn đề phân cụm. Các tập dữ liệu được phân loại thành một số cụm cụ thể (hãy gọi số đó là K) theo cách mà tất cả các điểm dữ liệu trong một cụm là đồng nhất và không đồng nhất với dữ liệu trong các cụm khác.

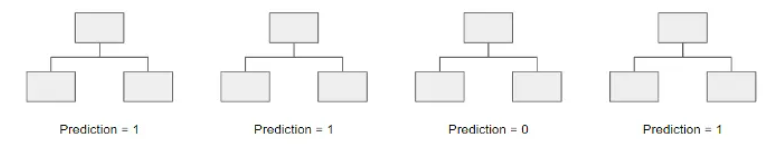


Hình 1.9. K-Mean

Cách K-mean tạo thành các cụm:

* Thuật toán K-mean chọn K số điểm cho mỗi cụm, gọi là centroid.
* Mỗi điểm dữ liệu tạo thành một cụm với các trung tâm gần nhất, tức là cụm K.
* Tạo ra các trung tâm mới dựa trên các cụm thành viên hiện có.
* Với những trung tâm mới này, khoảng cách gần nhất cho mỗi điểm dữ liệu được xác định. Quá trình này được lặp lại cho đến khi các trung tâm không thay đổi
  + 1. **Random Forest**

Random Forest là một kỹ thuật học tập tổng hợp được xây dựng dựa trên các cây quyết định. Random Forest liên quan đến việc tạo nhiều cây quyết định bằng cách sử dụng tập dữ liệu khởi động của dữ liệu gốc và chọn ngẫu nhiên một tập hợp con các biến ở mỗi bước của cây quyết định. Sau đó, mô hình sẽ chọn chế độ (mode) của tất cả các dự đoán của mỗi cây quyết định (bagging). Bằng cách dựa trên mô hình “đa số thắng”, nó làm giảm nguy cơ mắc lỗi từ một cây riêng lẻ



Hình 1.10. Random Forest

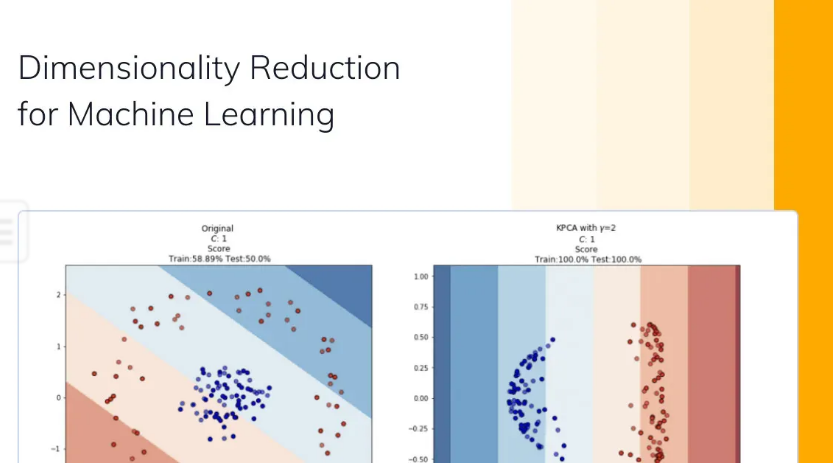
Ví dụ, nếu tạo ra một cây quyết định, cây thứ ba, nó sẽ dự đoán bằng 0. Nhưng nếu dựa vào chế độ (mode) của cả 4 cây quyết định, giá trị dự đoán sẽ là 1. Đây là sức mạnh của các Random Forest

* + 1. **Dimensionality Reduction**

Trong thế giới ngày nay, một lượng lớn dữ liệu đang được các công ty, cơ quan chính phủ và tổ chức nghiên cứu lưu trữ và phân tích. Các dữ liệu thô này chứa rất nhiều thông tin, thách thức ở đây là phải xác định được các mẫu và biến quan trọng.

Nói một cách đơn giản, Dimensionality Reduction là việc chuyển đổi dữ liệu từ không gian chiều cao thành không gian chiều thấp để biểu diễn chiều thấp giữ lại một số thuộc tính có ý nghĩa của dữ liệu ban đầu.

Các thuật toán giảm kích thước, hay giảm thứ nguyên như Decision Tree, Factor Analysis, Missing Value Ratio và Random Forest có thể giúp bạn tìm thấy các chi tiết có liên quan.



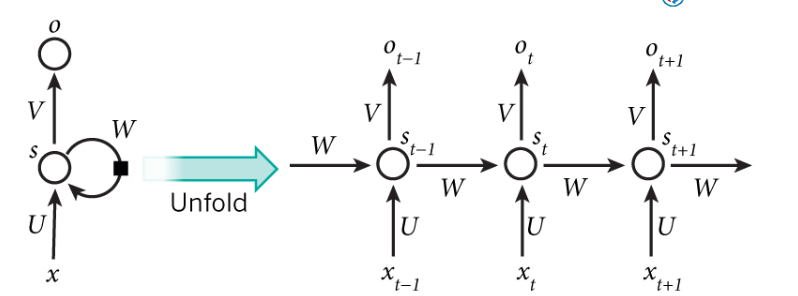
Hình 1.11. Dimensionality Reduction

* + 1. **Recurrent Neural Network (RNN)**

Recurrent Neural Network (RNN) là một loại mạng nơ-ron nhân tạo chủ yếu được sử dụng trong nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên (NLP). RNN được sử dụng trong học tập sâu và trong việc phát triển các mô hình bắt chước hoạt động của các tế bào thần kinh trong não người.

Mạng lặp lại được thiết kế để nhận dạng các mẫu trong chuỗi dữ liệu, chẳng hạn như văn bản, bộ gen, chữ viết tay, lời nói và dữ liệu chuỗi thời gian số phát ra từ cảm biến, thị trường chứng khoán và các cơ quan chính phủ.

Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào chứ nhỉ? RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó. Trên lý thuyết, RNN có thể sử dụng được thông tin của một văn bản rất dài, tuy nhiên thực tế thì nó chỉ có thể nhớ được một vài bước trước đó (ta cùng bàn cụ thể vấn đề này sau) mà thôi. Về cơ bản một mạng RNN có dạng như sau:



*Hình 1.12: Tính toán chuyển tiếp*

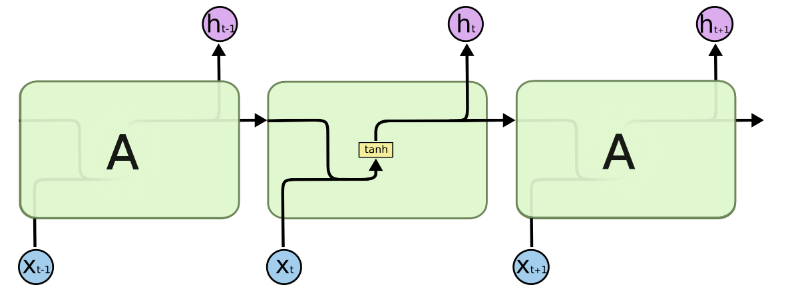
Mô hình trên mô tả phép triển khai nội dung của một RNN. Triển khai ở đây có thể hiểu đơn giản là ta vẽ ra một mạng nơ-ron chuỗi tuần tự.

* + 1. **Long short term memory (LSTM)**

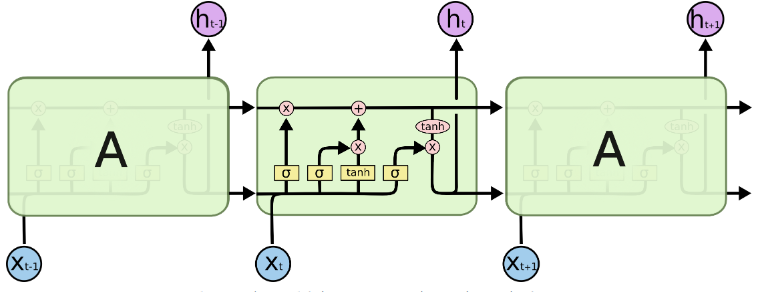
Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi [Hochreiter & Schmidhuber (1997)](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành. Chúng hoạt động cực kì hiệu quả trên nhiều bài toán khác nhau nên dần đã trở nên phổ biến như hiện nay.

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

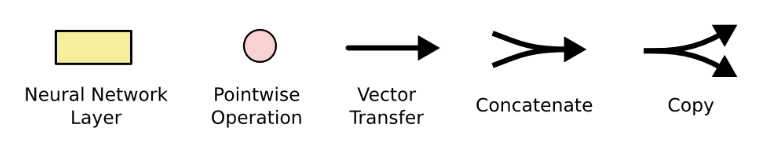
Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng *tanh*.



LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



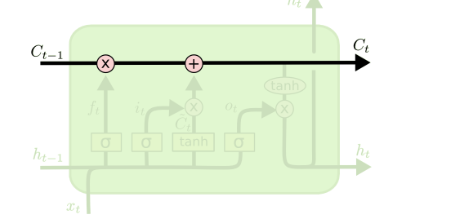
Ở sơ đồ trên, mỗi một đường mang một véc-tơ từ đầu ra của một nút tới đầu vào của một nút khác. Các hình trong màu hồng biểu diễn các phép toán như phép cộng véc-tơ chẳng hạn, còn các ô màu vàng được sử dụng để học trong các từng mạng nơ-ron. Các đường hợp nhau kí hiệu việc kết hợp, còn các đường rẽ nhánh ám chỉ nội dung của nó được sao chép và chuyển tới các nơi khác nhau.



Ý tưởng cốt lõi của LSTM

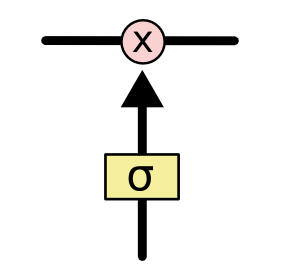
Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ.

Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.



LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.



Tầng sigmoid sẽ cho đầu ra là một số trong khoản [0,1][0,1], mô tả có bao nhiêu thông tin có thể được thông qua. Khi đầu ra là 00 thì có nghĩa là không cho thông tin nào qua cả, còn khi là 11 thì có nghĩa là cho tất cả các thông tin đi qua nó.

Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.

Bên trong LSTM

Bước đầu tiên của LSTM là quyết định xem thông tin nào cần bỏ đi từ trạng thái tế bào. Quyết định này được đưa ra bởi tầng sigmoid - gọi là “tầng cổng quên” (forget gate layer). Nó sẽ lấy đầu vào là ht−1​ và *xt*​ rồi đưa ra kết quả là một số trong khoảng [0,1][0,1] cho mỗi số trong trạng thái tế bào h−1*Ct*−1​. Đẩu ra là 11 thể hiện rằng nó giữ toàn bộ thông tin lại, còn 00 chỉ rằng taonf bộ thông tin sẽ bị bỏ đi.

Quay trở lại với ví dụ mô hình ngôn ngữ dự đoán từ tiếp theo dựa trên tất cả các từ trước đó, với những bài toán như vậy, thì trạng thái tế bào có thể sẽ mang thông tin về giới tính của một nhân vật nào đó giúp ta sử dụng được đại từ nhân xưng chuẩn xác. Tuy nhiên, khi đề cập tới một người khác thì ta sẽ không muốn nhớ tới giới tính của nhân vật nữa, vì nó không còn tác dụng gì với chủ thế mới này.



Bước tiếp theo là quyết định xem thông tin mới nào ta sẽ lưu vào trạng thái tế bào. Việc này gồm 2 phần. Đầu tiên là sử dụng một tầng sigmoid được gọi là “tầng cổng vào” (input gate layer) để quyết định giá trị nào ta sẽ cập nhập. Tiếp theo là một tầng ℎ*tanh* tạo ra một véc-tơ cho giá trị mới ~*Ct*​~​ nhằm thêm vào cho trạng thái. Trong bước tiếp theo, ta sẽ kết hợp 2 giá trị đó lại để tạo ra một cập nhập cho trạng thái.

Chẳng hạn với ví dụ mô hình ngôn ngữ của ta, ta sẽ muốn thêm giới tính của nhân vật mới này vào trạng thái tế bào và thay thế giới tính của nhân vật trước đó.



Giờ là lúc cập nhập trạng thái tế bào cũ h−1*Ct*−1​ thành trạng thái mới *Ct*​. Ở các bước trước đó đã quyết định những việc cần làm, nên giờ ta chỉ cần thực hiện là xong.

Ta sẽ nhân trạng thái cũ với *ft*​ để bỏ đi những thông tin ta quyết định quên lúc trước. Sau đó cộng thêm ∗~*it*​∗*Ct*​~​. Trạng thái mơi thu được này phụ thuộc vào việc ta quyết định cập nhập mỗi giá trị trạng thái ra sao.

Với bài toàn mô hình ngôn ngữ, chính là việc ta bỏ đi thông tin về giới tính của nhân vật cũ, và thêm thông tin về giới tính của nhân vật mới như ta đã quyết định ở các bước trước đó.



Cuối cùng, ta cần quyết định xem ta muốn đầu ra là gì. Giá trị đầu ra sẽ dựa vào trạng thái tế bào, nhưng sẽ được tiếp tục sàng lọc. Đầu tiên, ta chạy một tầng sigmoid để quyết định phần nào của trạng thái tế bào ta muốn xuất ra. Sau đó, ta đưa nó trạng thái tế bảo qua một hàm *tanh* để co giá trị nó về khoảng [−1,1][−1,1], và nhân nó với đầu ra của cổng sigmoid để được giá trị đầu ra ta mong muốn.

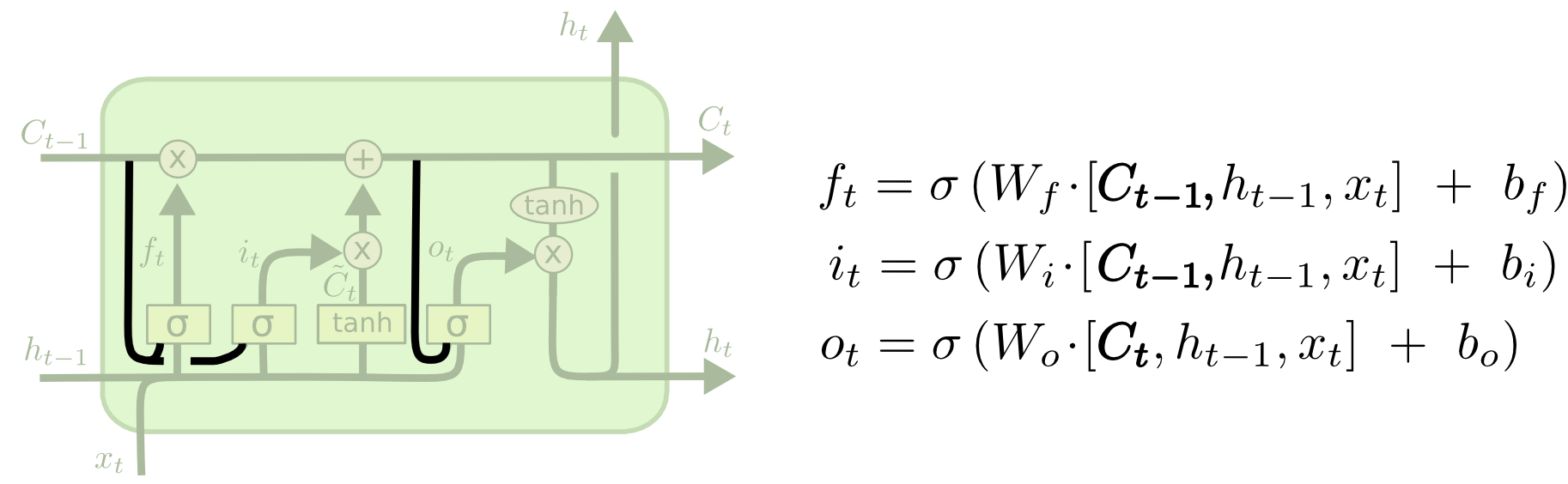
Với ví dụ về mô hình ngôn ngữ, chỉ cần xem chủ thể mà ta có thể đưa ra thông tin về một trạng từ đi sau đó. Ví dụ, nếu đầu ra của chủ thể là số ít hoặc số nhiều thì ta có thể biết được dạng của trạng từ đi theo sau nó phải như thế nào.



Các biến thể của bộ nhớ dài hạn

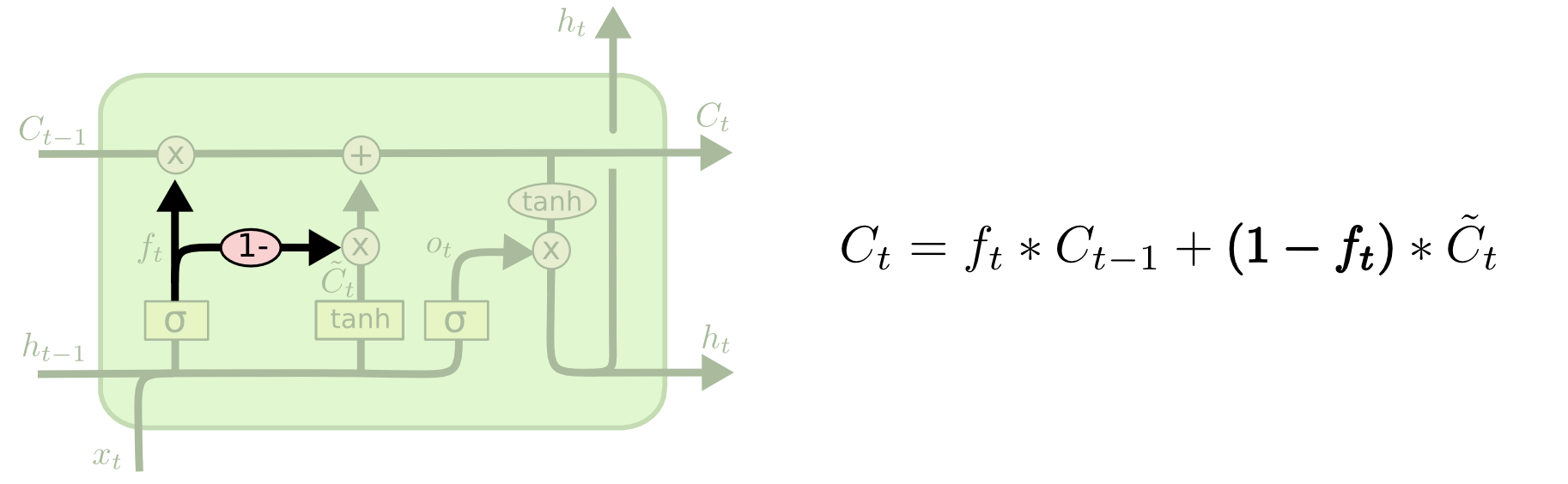
Những thứ ta vừa mô tả ở trên là một LSTM khá bình thường. Nhưng không phải tất cả các LTSM đều giống như vậy. Thực tế, các bài báo về LTSM đều sử dụng một phiên bản hơi khác so với mô hình LTSM chuẩn. Sự khác nhau không lớn, nhưng chúng giúp giải quyết phần nào đó trong cấu trúc của LTSM.

Một dạng LTSM phổ biến được giới thiệu bởi [Gers & Schmidhuber (2000)](ftp://ftp.idsia.ch/pub/juergen/TimeCount-IJCNN2000.pdf) được thêm các đường kết nối “peephole connections”, làm cho các tầng cổng nhận được giá trị đầu vào là trạng thái tế bào.

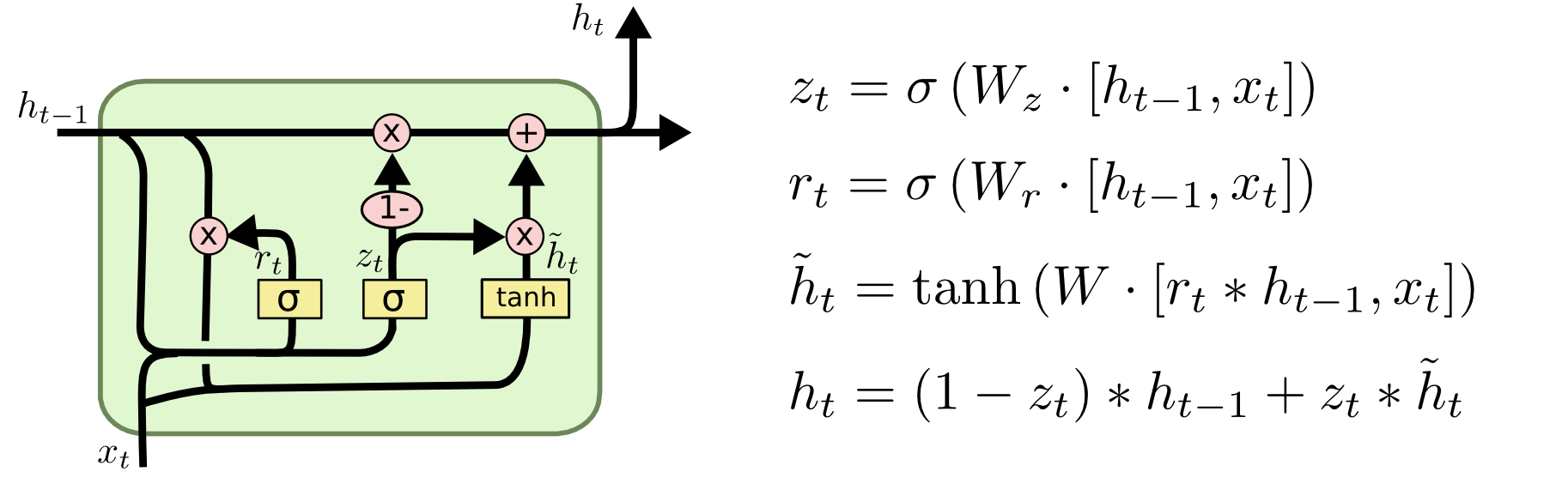


Hình trên mô tả các đường được thêm vào mọi cổng, nhưng cũng có những bài báo chỉ thêm cho một vài cổng mà thôi.

Một biến thể khác là nối 2 cổng loại trừ và đầu vào với nhau. Thay vì phân tách các quyết định thông tin loại trừ và thông tin mới thêm vào, ta sẽ quyết định chúng cùng với nhau luôn. Ta chỉ bỏ đi thông tin khi mà ta thay thế nó bằng thông tin mới đưa vào. Ta chỉ đưa thông tin mới vào khi ta bỏ thông tin cũ nào đó đi.



Một biến thể khá thú vị khác của LSTM là Gated Recurrent Unit, hay GRU được giới thiệu bởi [Cho, et al. (2014)](http://arxiv.org/pdf/1406.1078v3.pdf). Nó kết hợp các cổng loại trừ và đầu vào thành một cổng “cổng cập nhập” (update gate). Nó cũng hợp trạng thái tế bào và trạng thái ẩn với nhau tạo ra một thay đổi khác. Kết quả là mô hình của ta sẽ đơn giản hơn mô hình LSTM chuẩn và ngày càng trở nên phổ biến.



* 1. **Tổng quan về tầm quan trọng của dự báo kinh tế**

Dự báo kinh tế là một trong những công cụ thiết yếu trong điều hành kinh tế vĩ mô để nền kinh tế và các hoạt động xã hội phát triển theo hướng bền vững, tránh những xáo động lớn có thể ảnh hưởng tới mục tiêu ban đầu.

Dự báo các chỉ số kinh tế nhằm phục vụ công tác chỉ đạo, điều hành nền kinh tế của Chính phủ và Bộ Tài chính luôn là trọng tâm của công tác dự báo kinh tế ở các nước xây dựng nền kinh tế thị trường, ở các tổ chức tài chính, tiền tệ như ngân hàng và quản lý các quỹ đầu tư… Với sự phát triển mạnh mẽ của khoa học công nghệ, đặc biệt là công nghệ thông tin, các thông tin dữ liệu được hình thành trong quá trình hoạt động cũng như trong công tác chỉ đạo, điều hành và quản lý nền kinh tế ngày càng có cơ hội được thu thập đầy đủ và toàn diện. Khi đó những yếu tố kinh tế - xã hội có ảnh hưởng đến sự thay đổi của các chỉ số kinh tế sẽ được nhận diện ngày càng nhiều hơn.

Dự báo kịp thời và chính xác sẽ giúp nền kinh tế tận dụng được tối đa cơ hội phát triển và giảm thiểu rủi ro, đồng thời giúp các cơ quan quản lý chủ động sớm đưa ra quyết sách, giải pháp phù hợp nhằm đạt được các mục tiêu quan trọng trong duy trì tăng trưởng kinh tế và giữ ổn định vĩ mô.

* 1. **Mục tiêu, Đối tượng và phạm vi nghiên cứu**
     1. **Mục tiêu nghiên cứu**
* Ứng dụng kiến thức đã học vào thực tế
* Xây dựng được mô hình dự báo các chỉ tiêu kinh tế chủ yếu để ứng dụng trong thực tế nhằm cung cấp thông tin cho cộng đồng dân cư; phục vụ công tác nghiên cứu và điều hành, quản lý của các doanh nghiệp và Bộ Tài chính cũng như một số cơ quan chính phủ khác.
* Xác định tác động của Tỷ giá USD/VNĐ và giá dầu thô WTI (qua Hợp đồng tương lai) đối với xuất nhập khẩu của Việt Nam. Dựa vào các mô hình để dự đoán xu hướng xuất nhập khẩu, từ đó có thể hiểu được ảnh hưởng của các sự kiện quốc tế đến thị trường Việt Nam.
* Khảo sát tác động của giá vàng (qua Hợp đồng tương lai vàng) đối với các chỉ số kinh tế cụ thể của Việt Nam, có thể bao gồm lạm phát, tiêu dùng, và đầu tư. Mục tiêu là xây dựng mô hình dự đoán sự biến động của các chỉ số này dựa trên giá vàng.
* Dựa vào biến động của Tỷ giá USD/VNĐ và giá dầu thô WTI, xây dựng mô hình để dự đoán rủi ro tài chính và quản lý nguồn năng lực tại Việt Nam. Các doanh nghiệp và chính phủ có thể sử dụng thông tin này để thực hiện các biện pháp ngăn chặn và quản lý rủi ro.
  + 1. **Đối tượng nghiên cứu**
* **Nhà Nghiên Cứu:** Sinh viên đang thực hiện đề tài nghiên cứu này, có kiến thức về học máy và quan tâm đến ứng dụng của nó trong dự báo chỉ số kinh tế.
* **Chính Phủ:** Những quyết định và chính sách kinh tế của chính phủ có thể được ảnh hưởng bởi thông tin từ mô hình dự đoán này. Quản lý kinh tế cũng có thể tận dụng thông tin này để đưa ra các chiến lược và biện pháp điều chỉnh.
* **Doanh Nghiệp và Nhà Đầu Tư:** Các doanh nghiệp và nhà đầu tư có thể sử dụng thông tin từ mô hình để dự đoán và quản lý rủi ro tài chính, cũng như đưa ra quyết định đầu tư chiến lược.
* **Người Dân và Tiêu Dùng**: Người dân và tiêu dùng có thể được ảnh hưởng thông qua các yếu tố như lạm phát, thu nhập và giá tiêu dùng. Mô hình có thể cung cấp thông tin hữu ích để dự đoán xu hướng giá cả và tình hình kinh tế chung.
  + 1. **Phạm vi nghiên cứu**
* Thời gian: từ 1/1/2006 đến 24/7/2022
* Chỉ số kinh tế: Tỷ giá USD/VNĐ (USD\_W), Hợp đồng tương lai dầu thô WTI(DT\_W), Hợp đồng tương lai vàng (V\_W).
* Khu vực: tập trung chủ yếu vào Việt Nam, với mục tiêu cung cấp thông tin chi tiết và cụ thể về tình hình kinh tế quốc gia.
  1. **Các công nghệ sử dụng.**
     1. **Một số thư viện sử dụng**
        1. **Thư viện NumPy (Numeric Python)**

Numpy (Numeric Python) là một dự án mã nguồn mở nhằm hỗ trợ tính toán số với Python và là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng “core Python” đơn thuần.

Năm 2005, Travis Oliphant đã tạo NumPy bằng cách đưa các tính năng của mô-đun Numeric vào một mô-đun khác Numarray. Đây là một mô-đun mở rộng của Python, phần lớn được viết bằng C, cung cấp các chức năng khác nhau có khả năng thực hiện các phép tính số với tốc độ cao. Ngoài ra thư viện Numpy còn cung cấp nhiều cấu trúc dữ liệu mạnh mẽ khác nhau, triển khai các mảng và ma trận đa chiều. Các cấu trúc dữ liệu này được sử dụng để tính toán tối ưu thuật toán liên quan đến mảng và ma trận.

Với cuộc cách mạng của khoa học dữ liệu, các thư viện phân tích dữ liệu như NumPy, SciPy, Pandas, … đã có sự phát triển vượt bậc với cú pháp dễ dàng hơn nhiều so với các ngôn ngữ lập trình khác, python là ngôn ngữ lựa chọn hàng đầu cho các nhà khoa học dữ liệu. Thư viện NumPy cung cấp một cách thuận tiện và hiệu quả để xử lý một lượng lớn dữ liệu và cũng rất tiện lợi với phép nhân ma trận và định hình lại dữ liệu, vì thế nó nhanh chóng nên hợp lý khi làm việc với một bộ dữ liệu lớn. Thư viện Numpy có những ưu điểm khi sử dụng để phân tích dữ liệu như sau:

* NumPy thực hiện toán học và logic mảng.
* NumPy thực hiện hiệu quả các mảng đa chiều.
* NumPy thực hiện các phép tính khoa học.
* NumPy có khả năng thực hiện Fourier Transform và định hình lại dữ liệu được lưu trữ trong các mảng đa chiều.
* NumPy cung cấp các hàm tích hợp cho đại số tuyến tính và tạo số ngẫu nhiên.

Sự kết hợp của NumPy với SciPy và Matplotlib được sử dụng thay thế cho MATLAB vì Python là ngôn ngữ lập trình hoàn chỉnh hơn và dễ dàng hơn MATLAB. Các ứng dụng của thư viện NumPy được thể hiện ở Hình 2.2.



Chart, bubble chart

Description automatically generated

Hình 1.13 Các ứng dụng của thư viện Numpy

* + - 1. **Thư viện Pandas**

Thư viện pandas trong python là một thư viện mã nguồn mở, hỗ trợ đắc lực trong thao tác dữ liệu. Đây cũng là bộ công cụ phân tích và xử lý dữ liệu mạnh mẽ của ngôn ngữ lập trình python. Thư viện này được sử dụng rộng rãi trong cả nghiên cứu lẫn phát triển các ứng dụng về khoa học dữ liệu. Thư viện này sử dụng một cấu trúc dữ liệu riêng là Dataframe. Pandas cung cấp rất nhiều chức năng xử lý và làm việc trên cấu trúc dữ liệu này. Chính sự linh hoạt và hiệu quả đã khiến cho pandas được sử dụng rộng rãi.

* DataFrame đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục;
* Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv, text, excel, sql database, hdf5;
* Liên kết dữ liệu thông minh, xử lý được trường hợp dữ liệu bị thiếu. Tự động đưa dữ liệu lộn xộn về dạng có cấu trúc;
* Dễ dàng thay đổi bố cục của dữ liệu;
* Tích hợp cơ chế trượt, lập chỉ mục, lấy ra tập con từ tập dữ liệu lớn.
* Có thể thêm, xóa các cột dữ liệu;
* Tập hợp hoặc thay đổi dữ liệu với group by cho phép bạn thực hiện các toán tử trên tập dữ liệu;
* Hiệu quả cao trong trộn và kết hợp các tập dữ liệu;
* Lập chỉ mục theo các chiều của dữ liệu giúp thao tác giữa dữ liệu cao chiều và dữ liệu thấp chiều;
* Tối ưu về hiệu năng;
* Pandas được sử dụng rộng rãi trong cả học thuật và thương mại. Bao gồm thống kê, thương mại, phân tích, quảng cáo,…

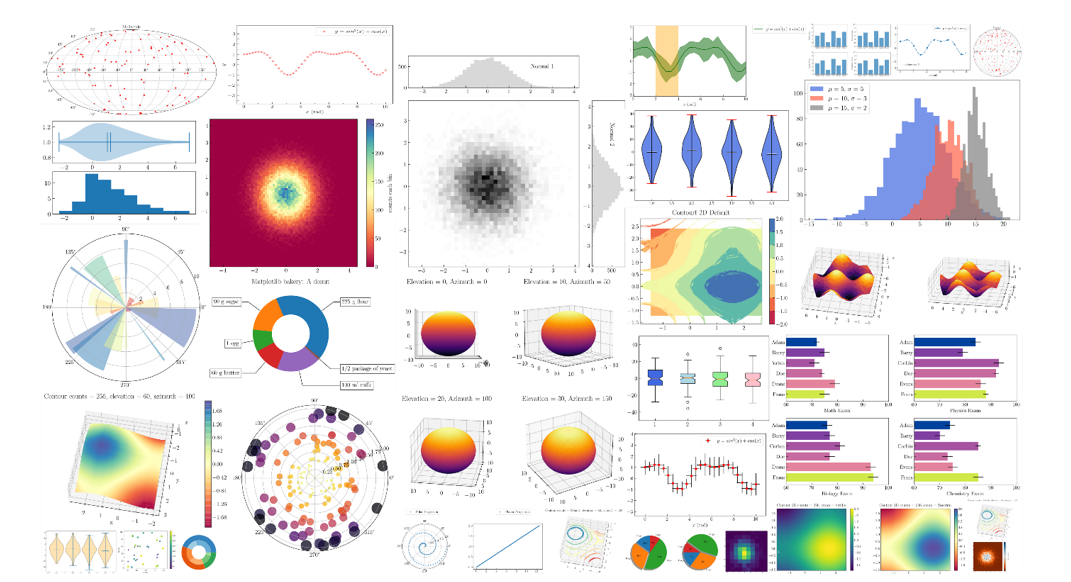


Hình 1.14: một số ứng dụng của thư viện pandas

* + - 1. **Thư viện Matplotlib**

“Matplotlib” là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở. Các kiểu biểu đồ của Matplotlib có thể được tìm thấy như trong Hình dưới đây.

* Mục tiêu đơn giản hóa tối đa công việc vẽ biểu đồ qua câu lệnh được cung cấp sẵn.
* Matplotlib hỗ trợ rất nhiều loại biểu đồ, đặc biệt là các loại được sử dụng trong nghiên cứu hoặc kinh tế như biểu đồ dòng, đường, tần suất (histograms), phổ, tương quan, errorcharts, scatterplots,…
* Cấu trúc matplotlib gồm nhiều phần được phục vụ cho các mục đích sử dụng khác nhau.



Hình 1.13: Các loại biểu đồ trong thư viện Matplotlib

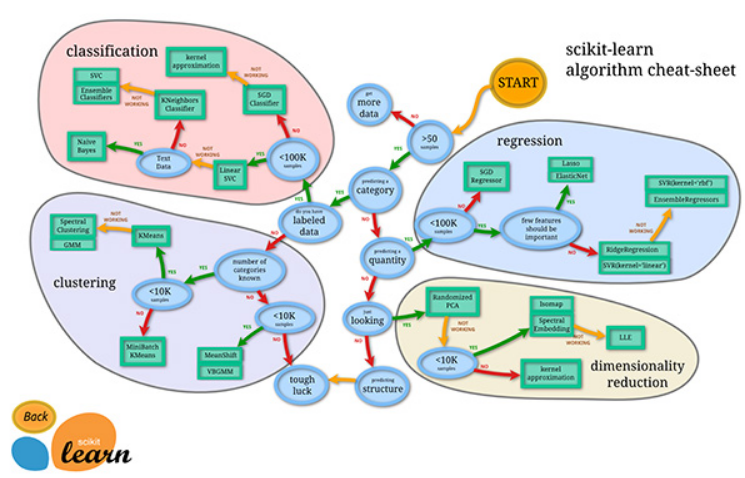
Trong đề tài này, thư viện Matplotlib được sử chủ yếu để biểu diễn kết quả đánh giá mô hình dưới dạng biểu đồ trực quan, dựa vào đó, ta sẽ thực hiện được các bước điều chỉnh các tham số cũng như cấu trúc của mô hình sao cho nó đặt được hiệu năng tối ưu nhất.

* + - 1. **Thư viện Scikit-learn**

Nếu bạn đang sử dụng Python và đang muốn tìm một thư viện mạnh mẽ mà bạn có thể mang các thuật toán học máy (machine learning) vào trong một hệ thống thì không còn thư viện nào thích hợp hơn scikit-learn.

Thư viện này tích hợp rất nhiều thuật toán hiện đại và cố điển giúp bạn vừa học vừa tiến hành đưa ra các giải pháp hữu ích cho bài toán của bạn một cách đơn giản.

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.



Hình 1.15: ứng dụng của thư viện Scikit-learn

* + - 1. **Thư viện TensorFlow**

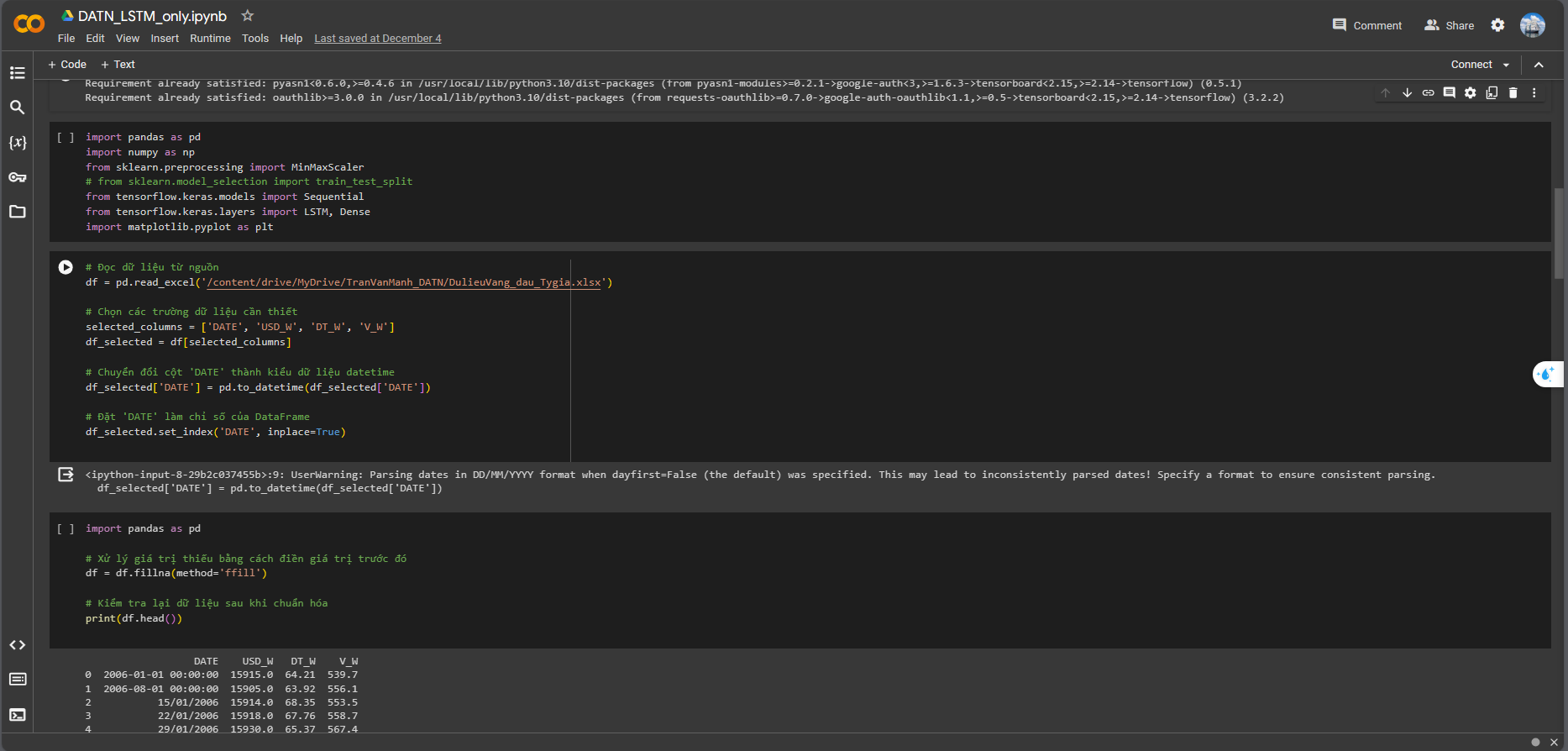
Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lý tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu. Trong đó, các node là các phép tính toán học còn các cạnh biểu thị luồng dữ liệu. Tensor là cấu trúc dữ liệu trong tensorflow đại diện cho tất cả các loại dữ liệu. Nói cách khác, tất cả các kiểu dữ liệu khi đưa vào trong Tensorflow thì đều được gọi là Tensor. Do vậy, có thể hiểu được Tensorflow là một thư viện mô tả, điều chỉnh dòng chảy của các Tensor. Tensor có 03 thuộc tính cơ bản: là rank, shape và type. Rank là số bậc của tensor. Việc phân rank này khá quan trọng vì nó đồng thời cũng giúp phân loại dữ liệu của Tensor. Shape là chiều của Tensor. Kiểu dữ liệu của các phần tử (elements) trong Tensor. Vì 01 Tensor chỉ có duy nhất 01 thuộc tính Type nên từ đó suy ra chỉ có duy nhất một kiểu Type duy nhất cho toàn bộ phần tử có trong Tensor hiện tại.

## 1.8.2 Công cụ hỗ trợ

Colaboratory (Google Colab) là một sản phẩm từ Google Research dựa trên Jupyter Notebook cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, ML và giáo dục. Google Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, người nghiên cứu có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp đầy đủ.

Colab cung cấp nhiều loại GPU và thay đổi theo thời gian. Vì là dịch vụ miễn phí, nên Colab sẽ có những thứ tự ưu tiên trong việc sử dụng tài nguyên hệ thống, cũng như giới hạn thời gian sử dụng, thời gian sử dụng tối đa tới 12 giờ.

Sử dụng Google Colab có những lợi ích ưu việt như: sẵn sàng chạy Python ở bất kỳ thiết bị nào có kết nối Internet mà không cần cài đặt, chia sẻ và làm việc nhóm dễ dàng, sử dụng miễn phí GPU cho các dự án về AI.

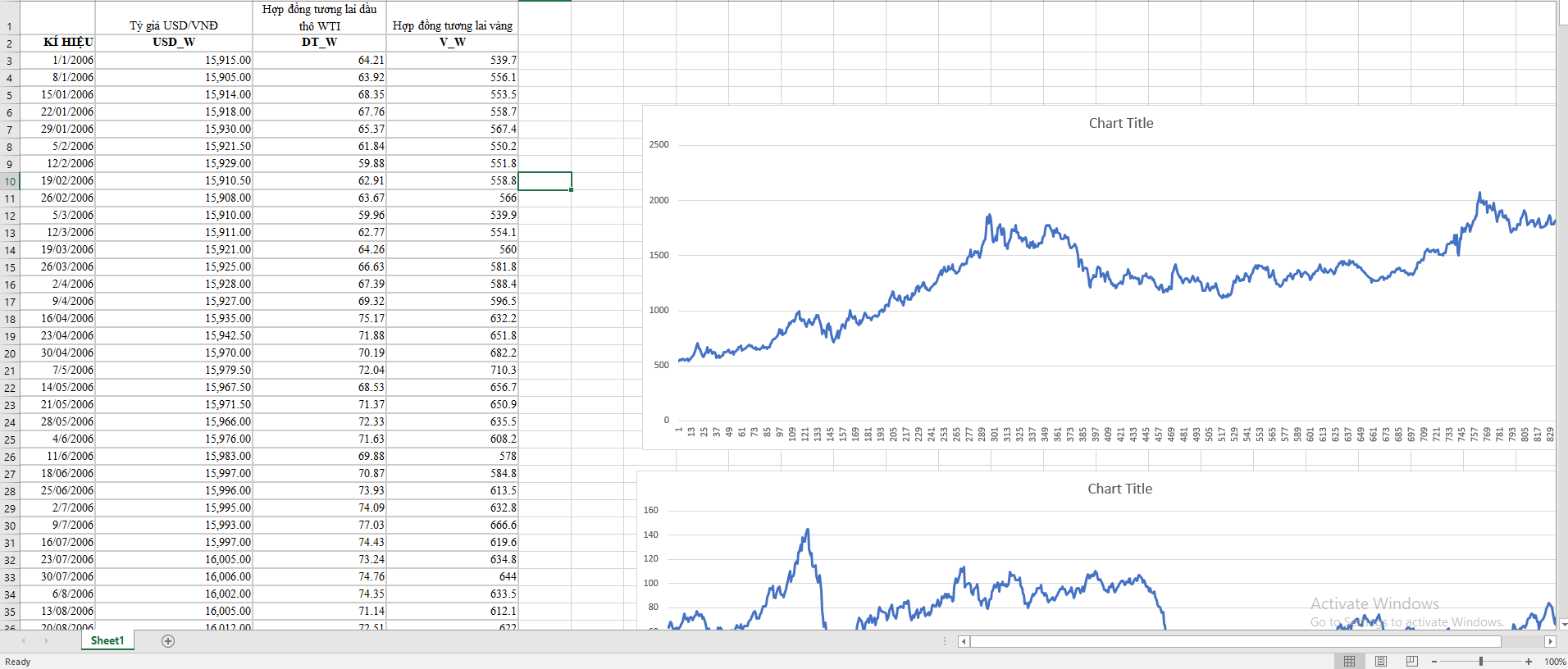


Hình 1.16 Sử dụng Google Colab để đào tạo mô hình

Trong đề tài này, em sử dụng Google Colab làm môi trường lập trình chính để đào tạo mô hình cũng như kiểm thử kết quả huấn luyện bởi sức mạnh đến từ GPU của Google Colab.

# XÂY DỰNG MÔ HÌNH

* 1. **Thu thập dữ liệu.**
  2. **Tiền xử lý dữ liệu.**
     1. **Phân tích và mô tả dữ liệu**



Bộ dữ liệu chứa thông tin về tỷ giá USD/VNĐ, giá hợp đồng tương lai dầu thô WTI, và giá hợp đồng tương lai vàng qua một khoảng thời gian dài. Dữ liệu này có thể làm nền tảng cho nhiều phân tích và nghiên cứu trong lĩnh vực tài chính và thị trường chứng khoán. Dưới đây là một giới thiệu tổng quan:

Bộ dữ liệu tài chính - USD/VNĐ, Dầu Thô WTI, và Vàng

Dữ liệu tỷ giá USD/VNĐ:

* Kí hiệu: USD\_W
* Thời gian bắt đầu: 1/1/2006
* Thời gian kết thúc: 24/07/2007
* Tần suất: Hàng tuần

Dữ liệu hợp đồng tương lai dầu thô wti:

* Kí hiệu: DT\_W
* Thời gian bắt đầu: 1/1/2006
* Thời gian kết thúc: 24/07/2007
* Tần suất: Hàng tuần

Dữ liệu hợp đồng tương lai vàng:

* Kí hiệu: V\_W
* Thời gian bắt đầu: 1/1/2006
* Thời gian kết thúc: 24/07/2007
* Tần suất: Hàng tuần

Mục Tiêu Nghiên Cứu:

1. Phân tích xu hướng:

* Xác định xu hướng của tỷ giá USD/VNĐ, giá dầu thô WTI, và giá vàng qua thời gian.
* Vẽ biểu đồ để minh họa xu hướng và sự biến động của các yếu tố này.

2. Liên kết giữa các biến động:

* Phân tích sự liên kết giữa giá USD/VNĐ, giá dầu thô WTI, và giá vàng.
* Xác định mức độ ảnh hưởng của biến động trong một thị trường đến các thị trường khác.

3. Dự đoán:

* Nếu có, xây dựng mô hình dự đoán cho các biến động trên dựa trên dữ liệu lịch sử.
* Đánh giá hiệu suất dự đoán của mô hình và nắm bắt các yếu tố ảnh hưởng.
  + 1. **Xử lý và chuẩn hóa dữ liệu**
       1. **Xử lý dữ liệu:**

|  |
| --- |
| # Xử lý giá trị NaN  df\_selected = df\_selected.fillna(method="ffill", inplace=False)  # Xóa dữ liệu trùng lặp  df\_selected = df\_selected.drop\_duplicates() |

Xử lý các giá trị NaN tồn tại trong dữ liệu bằng fillna():

* Tham số method là ffill(forward fill): Điền các giá trị NaN bằng giá trị của hàng hoặc cột trước đó. Nó sẽ sao chép giá trị gần nhất phía trước mà không phải là NaN để điền vào giá trị NaN tiếp theo.
* Tham số inplace: quyết định có nên thay thế trực tiếp vào dữ liệu gốc hay là tạo ra một bản sao rồi thay thế vào bản sao đó.
  + - 1. **Chuẩn hóa dữ liệu:**

|  |
| --- |
| scaler = MinMaxScaler()  df\_scaled = scaler.fit\_transform(df\_selected) |

Xử dụng phương pháp Min-Max Scaling:

* Phương pháp này đưa các giá trị về 1 khoảng đặc biệt, thường là [0,1] hoặc [−1,1].
* Phương pháp này giúp biến đổi dữ liệu để chúng có cùng phạm vi hoặc đơn vị đo, giúp mô hình học máy hoạt động hiệu quả hơn và tránh tác động tiêu cực của các biến có giá trị lớn hơn.

Công thức tổng quát:

* Trong đó:

X’ : là giá trị đã chuẩn hóa.

X : là giá trị của dữ liệu ban đầu.

Xmin : là giá trị nhỏ nhất trong dữ liệu ban đầu.

Xmax : là giá trị lớn nhất trong giá trị ban đầu.

Công thức áp dụng trong khoảng [a,b]:

* + 1. **Chia dữ liệu thành tập huấn luyện và tập kiểm tra**

|  |
| --- |
| train\_data, test\_data = train\_test\_split(      df\_scaled, test\_size=0.2, shuffle=False) |

* Chia dữ liệu tỉ lệ 80-20:xử dụng hàm train\_test\_split để chia dữ liệu với 80% là tập huấn luyện và 20% là tập kiểm tra.
  1. **Lựa chọn mô hình học máy.**

**2.3.1 lý do chọn mô hình**

**2.3.2 chuẩn bị dữ liệu cho mô hình**

|  |
| --- |
| def prepare\_data(data, time\_steps):      X, y = [], []      for i in range(len(data) - time\_steps):          X.append(data[i:(i + time\_steps)])          y.append(data[i + time\_steps])      return np.array(X), np.array(y)  time\_steps = 10  X\_train, y\_train = prepare\_data(train\_data, time\_steps)  X\_test, y\_test = prepare\_data(test\_data, time\_steps) |

**2.3.3**

* 1. **Xây dựng mô hình.**

|  |
| --- |
|  |

* 1. **Huấn luyện mô hình.**
  2. **Đánh giá mô hình.**

# THỰC NGHIỆM, SO SÁNH VÀ ĐÁNH GIÁ

* 1. **Kết quả đạt được.**
  2. **So sánh với các phương pháp khác.**
  3. **Đánh giá hiệu xuất và phân tích kết quả.**

# Kết Luận

* 1. **Tóm tắt kết quả nghiên cứu.**
  2. **Đánh giá tổng quan về hiệu suất mô hình.**
  3. **Đề xuất hướng phát triển tương lai.**

# TÀI LIỆU THAM KHẢO