TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

**---------🙞🙜🕮🙞🙜---------**



ĐỒ ÁN TỐT NGHIỆP

**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

# MỤC LỤC

[MỤC LỤC 2](#_Toc155868167)

[DANH MỤC HÌNH ẢNH 5](#_Toc155868168)

[DANH MỤC BẢNG BIỂU 5](#_Toc155868169)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ 5](#_Toc155868170)

[I. TỔNG QUAN ĐỒ ÁN 6](#_Toc155868171)

[**1.1** **Đặt vấn đề** 6](#_Toc155868172)

[**1.2** **Mục tiêu đồ án** 6](#_Toc155868173)

[II. CƠ SỞ LÝ THUYẾT 7](#_Toc155868174)

[**2.1** **Ngôn ngữ lập trình python** 7](#_Toc155868175)

[**2.1.1** **Python là gì?** 7](#_Toc155868176)

[**2.1.2** **Một số ứng dụng của python** 7](#_Toc155868177)

[**2.1.2.1** **Phát triển web** 7](#_Toc155868178)

[Python thường được sử dụng để phát triển back-end của trang web hoặc ứng dụng thông qua một số framework như Flask, Django. 7](#_Toc155868179)

[**2.1.2.2** **Khoa học dữ liệu và học máy** 7](#_Toc155868180)

[**2.1.2.3** **Trí tuệ nhân tạo** 7](#_Toc155868181)

[**2.1.2.4** **Trích xuất dữ liệu từ website** 7](#_Toc155868182)

[**2.2** **Tổng quan về học máy** 8](#_Toc155868183)

[**2.2.1** **Khái niệm về học máy** 8](#_Toc155868184)

[**2.2.2** **Một số ứng dụng của học máy** 10](#_Toc155868185)

[**2.2.2.1** **Tự động phân loại** 10](#_Toc155868186)

[**2.2.2.2** **Ứng dụng trong mạng xã hội** 11](#_Toc155868187)

[**2.2.2.3** **Nhận diện hình ảnh** 11](#_Toc155868188)

[**2.2.2.4** **Ứng dụng cho xe tự lái** 11](#_Toc155868189)

[**2.2.2.5** **Phân tích và dự báo một số lĩnh vực** 11](#_Toc155868190)

[**2.3** **Một số dạng học máy** 12](#_Toc155868191)

[**2.3.1** **Học có giám sát (supervised learning)** 12](#_Toc155868192)

[**2.3.2** **Học không giám sát (unsupervised learning)** 13](#_Toc155868193)

[**2.3.3** **Học bán giám sát (semi-supervised learning)** 13](#_Toc155868194)

[**2.3.4** **Học tăng cường (reinforcement learning)** 14](#_Toc155868195)

[**2.4** **Một số thuật toán, mô hình trong học máy** 14](#_Toc155868196)

[**2.4.1** **Linear Regression( hồi quy tính tuyến)** 14](#_Toc155868197)

[**2.4.2** **Logictic Regression ( hồi quy logistic)** 15](#_Toc155868198)

[**2.4.3** **Support Vector machine( SVM)** 16](#_Toc155868199)

[**2.4.4** **Recurrent Neural Network (RNN)** 17](#_Toc155868200)

[**2.4.5** **Long short term memory (LSTM)** 18](#_Toc155868201)

[**2.5** **Tổng quan về tầm quan trọng của dự báo kinh tế** 19](#_Toc155868202)

[**2.6** **Mục tiêu, Đối tượng và phạm vi nghiên cứu** 20](#_Toc155868203)

[**2.6.1** **Mục tiêu nghiên cứu** 20](#_Toc155868204)

[**2.6.2** **Đối tượng nghiên cứu** 21](#_Toc155868205)

[**2.6.3** **Phạm vi nghiên cứu** 21](#_Toc155868206)

[**2.7** **Các công nghệ sử dụng.** 21](#_Toc155868207)

[**2.7.1** **Một số thư viện sử dụng** 21](#_Toc155868208)

[**2.7.1.1** **Thư viện NumPy (Numeric Python)** 21](#_Toc155868209)

[**2.7.1.2** **Thư viện Pandas** 23](#_Toc155868210)

[**2.7.1.3** **Thư viện Matplotlib** 24](#_Toc155868211)

[**2.7.1.4** **Thư viện Scikit-learn** 25](#_Toc155868212)

[**2.7.1.5** **Thư viện TensorFlow** 26](#_Toc155868213)

[2.8 Công cụ hỗ trợ 27](#_Toc155868214)

[III. XÂY DỰNG MÔ HÌNH 28](#_Toc155868215)

[**3.1** **Tổng quan các bước thực hiện bài toán.** 28](#_Toc155868216)

[**3.2** **Thu thập dữ liệu.** 28](#_Toc155868217)

[**3.3** **Xử lý dữ liệu.** 28](#_Toc155868218)

[**3.3.1** **Xử lý dữ liệu** 30](#_Toc155868219)

[**3.4** **Lựa chọn mô hình học máy.** 32](#_Toc155868220)

[**3.5** **Xây dựng mô hình.** 32](#_Toc155868221)

[**3.6** **Huấn luyện mô hình.** 36](#_Toc155868222)

[**3.7** **Đánh giá mô hình.** 37](#_Toc155868223)

[IV. THỰC NGHIỆM, SO SÁNH VÀ ĐÁNH GIÁ 41](#_Toc155868224)

[**4.1** **Thực nghiệm.** 41](#_Toc155868225)

[**4.2** **Phân tích kết quả thực nghiệm.** 48](#_Toc155868226)

[**4.3** **So sánh với các phương pháp khác.** 49](#_Toc155868227)

[**4.4** **Đánh giá hiệu xuất và phân tích kết quả.** 50](#_Toc155868228)

[**4.5** **Kết luận** 51](#_Toc155868229)

[TÀI LIỆU THAM KHẢO 52](#_Toc155868230)

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG BIỂU

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

# TỔNG QUAN ĐỒ ÁN

* 1. **Đặt vấn đề**

Việt Nam, với tốc độ tăng trưởng kinh tế đáng kể trong những năm qua, đã trở thành một điểm đến hấp dẫn cho đầu tư và phát triển. Tuy nhiên, sự phức tạp và không chắc chắn của thị trường kinh tế nước ta đặt ra nhiều thách thức cũng như cơ hội. Hiện nay, tình hình kinh tế Việt Nam đang phải đối mặt với một loạt các yếu tố ảnh hưởng, từ yếu tố nội địa đến những yếu tố quốc tế, đặt ra những thách thức cần được quản lý một cách chặt chẽ.

Việt Nam đã ghi nhận tăng trưởng GDP đáng kể, nhưng sự ảnh hưởng của đại dịch COVID-19 và các biến động thị trường thế giới đang tạo ra những biến động khó dự đoán.

Cách mạng Công nghiệp 4.0 đang thách thức và tạo ra cơ hội mới cho các ngành công nghiệp truyền thống, đặt ra nhu cầu cần áp dụng các công nghệ mới để nâng cao năng suất và cạnh tranh.

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 đối với nền kinh tế nước nhà. 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. **Mục tiêu đồ án**

Xây dựng được mô hình dự báo kinh tế tại việ nam có độ chính xác cao Điều này giúp tạo ra những dự báo đáng tin cậy hỗ trợ quyết định chính sách và chiến lược kinh doanh. Mô hình cần phản ánh đúng tính biến động của thị trường kinh tế, đặc biệt là trong bối cảnh các yếu tố ảnh hưởng nhanh chóng và có thể thay đổi.

Có thể áp dụng mô hình vào thực tế để hỗ trợ quyết định chính sách và quản lý kinh doanh, cung cấp thông tin hữu ích và khả năng dự báo với độ chính xác cao. Hỗ trợ quyết định chiến lược của các doanh nghiệp và chính phủ thông qua dự báo chính xác, giúp họ thích ứng và phát triển theo hướng bền vững.

Mô hình có thể tạo nền tảng cho nghiên cứu và phát triển tiếp theo trong lĩnh vực này, khuyến khích sự đóng góp và phát triển khoa học kỹ thuật.

# CƠ SỞ LÝ THUYẾT

* 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**

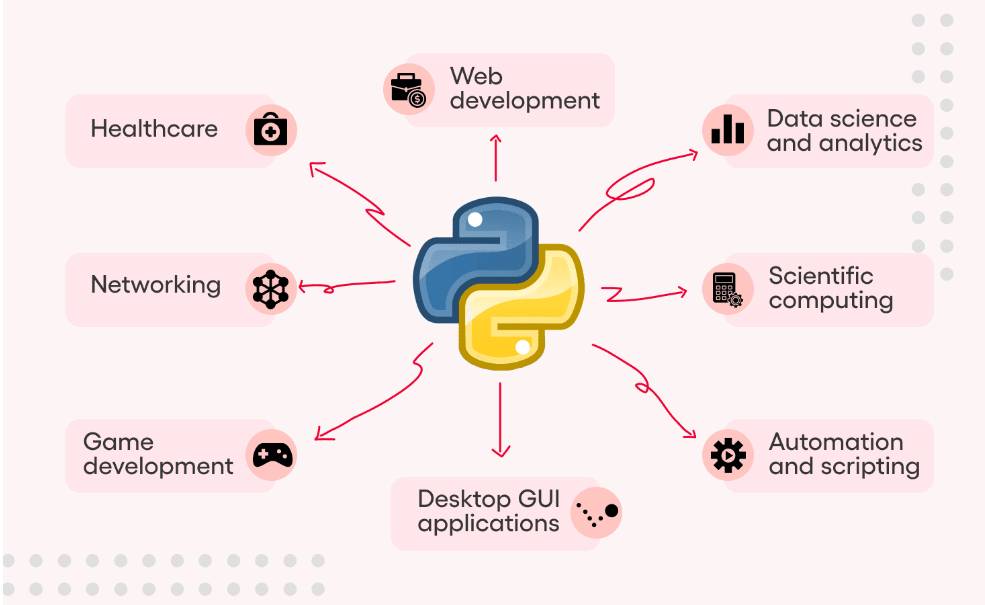
Python cực kỳ phù hợp với việc vận dụng dữ liệu, phân tích triển khai các mô hình thuật toán phức tạp. Có thể thu thập và mô hình hóa dữ liệu được. Xử lý, làm sách dữ liệu, vẽ biểu đồ để trực quan hóa dữ liệu, phân tích thống kê…

* + - 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 2. . 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 2. . Các ứng dụng của 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**

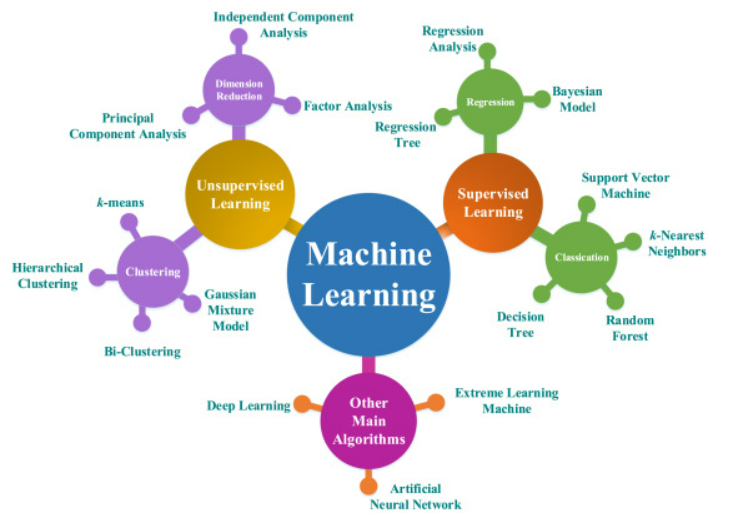
Một trong những ứng dụng thú vị nhất của Machine learning là ô tô tự lái. Machine learning đóng một vai trò quan trọng trong việc vận hành ô tô tự lái. Tesla, công ty sản xuất xe hơi nổi tiếng đang tiến hành các nghiên cứu về xe tự lái bằng phương pháp học không giám sát (unsupervised learning method )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 2. . 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.

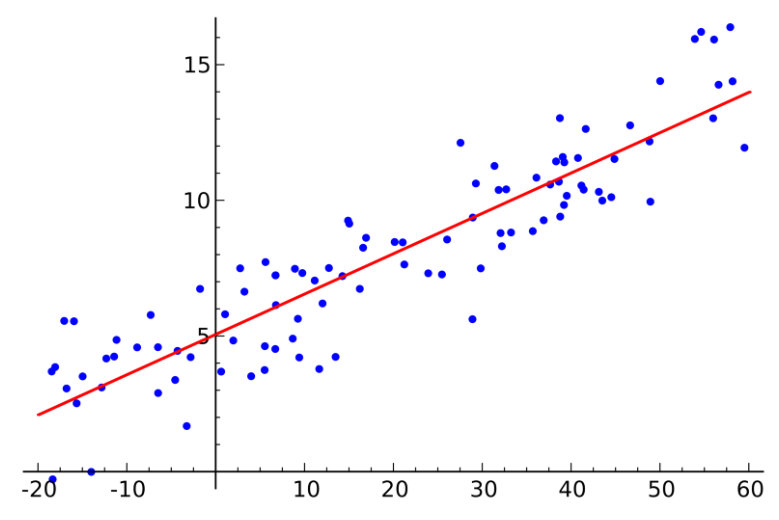
* + 1. **Học tăng cường (reinforcement learning)**

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.

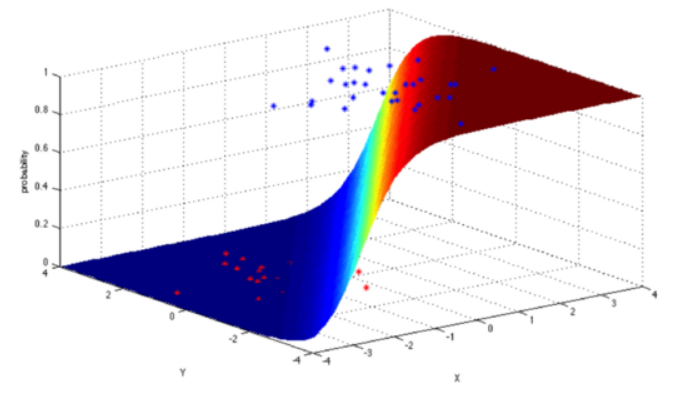


Hình 2. . Giới thiệu hồi quy tuyến tính

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.

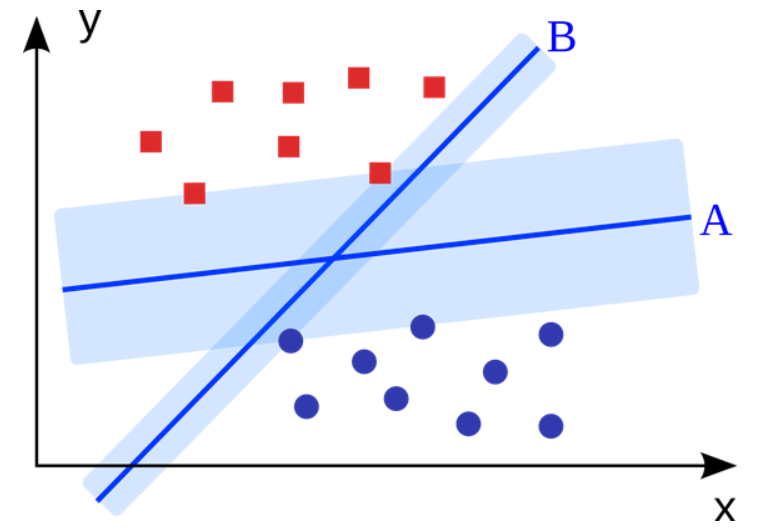


Hình 2. . Giới thiệu Logistic Regression

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. **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 2. . Giới thiệu SVM

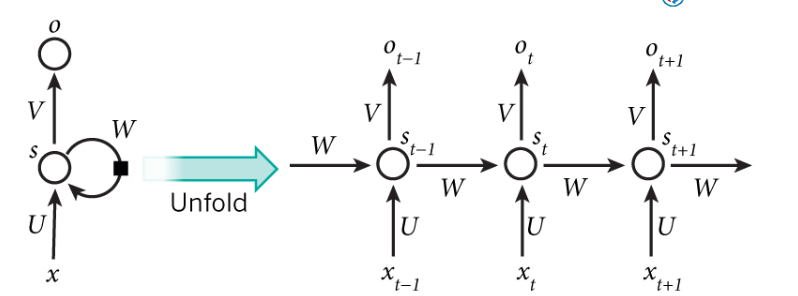
Xét về quy mô, một số vấn đề lớn nhất đã được giải quyết bằng cách sử dụng SVM (với việc thực hiện sửa đổi phù hợp) ví dụ như hiển thị quảng cáo, phát hiện giới tính dựa trên hình ảnh, phân loại hình ảnh có quy mô lớn …

* + 1. **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 2. . 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.

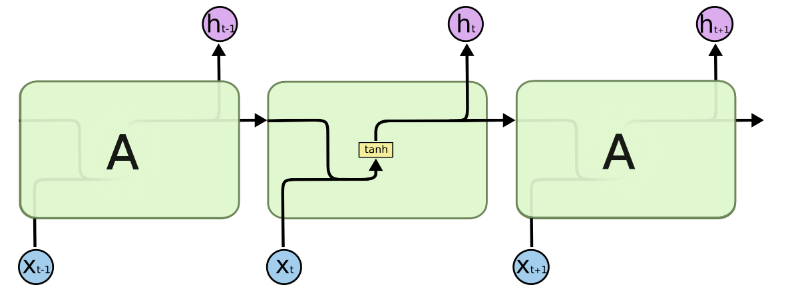
* + 1. **Long short term memory (LSTM)**

Mạng trí nhớ ngắn hạn định hướng dài hạn (Long Short Term Memory), 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 năm 1997 bở Hochreiter & Schmidhuber, và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành.

LSTM đã tỏ ra khắc phục được rất nhiều những hạn chế của RNN trước đây về triệt tiêu đạo hàm.

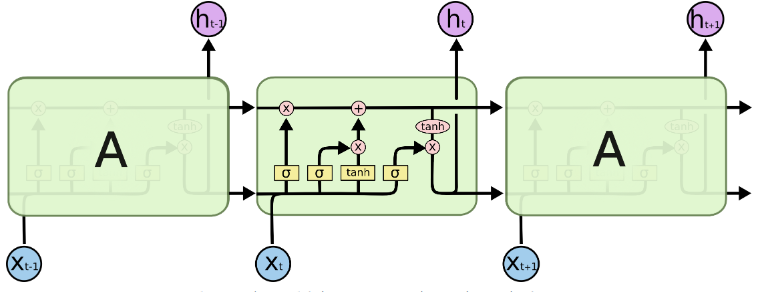
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

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*.



Hình 2. . Giơi thiệu RNN

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 (3 sigmoid và 1 tanh) tương tác với nhau một cách rất đặc biệt.



Hình 2. . Giới thiệu LSTM

Ở sơ đồ trên, mỗi một đường mang một vector 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 vector 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.

* 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 2. . Giới thiệu 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,…



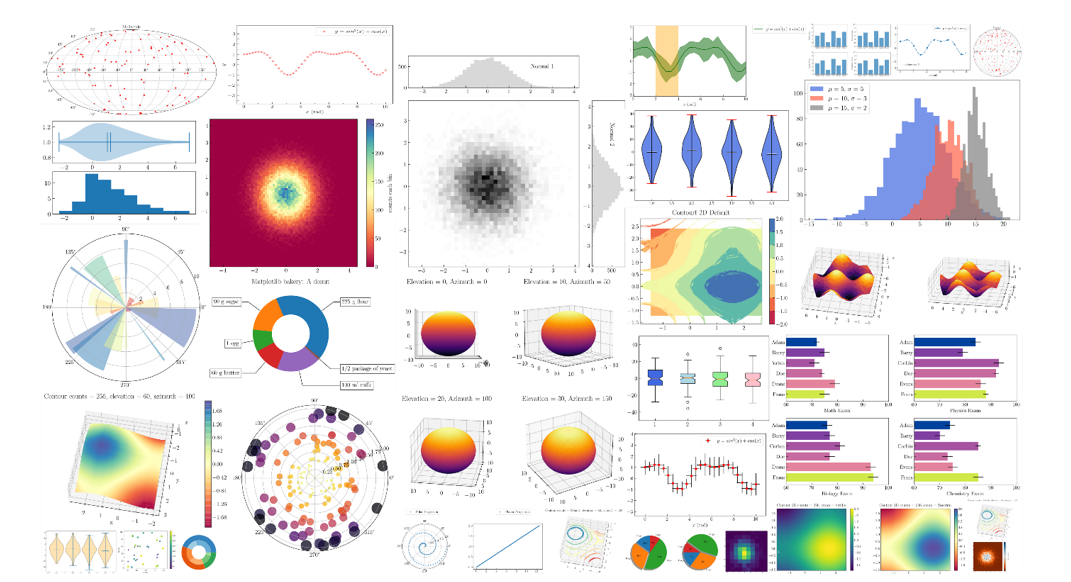
Hình 2. . Một số ứng dụng của pandas

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 2. . Các loại biểu đồ trong thư viện Matplotlib

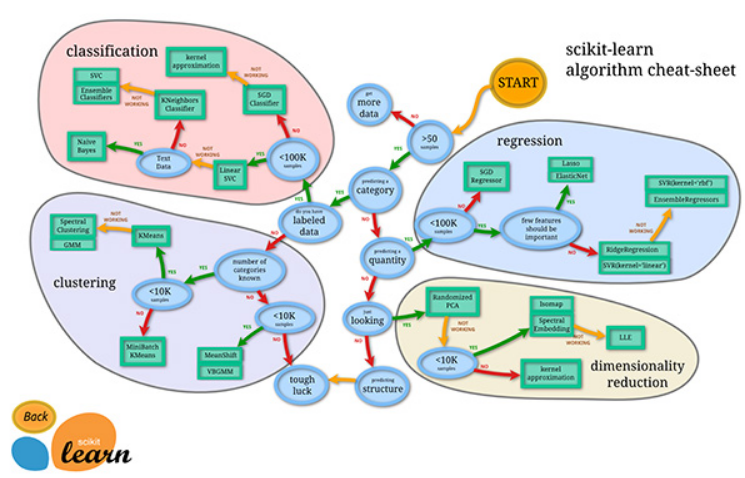
Trong đề tài này, thư viện Matplotlib được sử chủ yếu để biểu diễn kết quả đánh giá mô hình dưới dạng biểu đồ trực quan, dựa vào đó, ta sẽ thực hiện được các bước điều chỉnh các tham số cũng như cấu trúc của mô hình sao cho nó đặt được hiệu năng tối ưu nhất.

* + - 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.

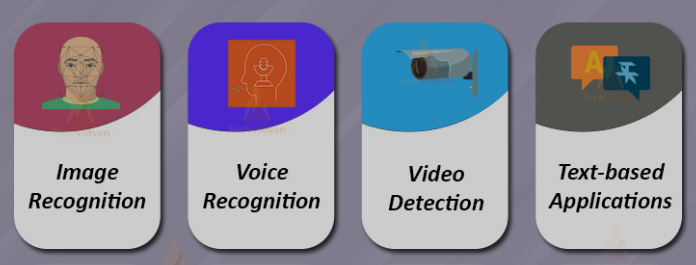
Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.



Hình 2. . 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.



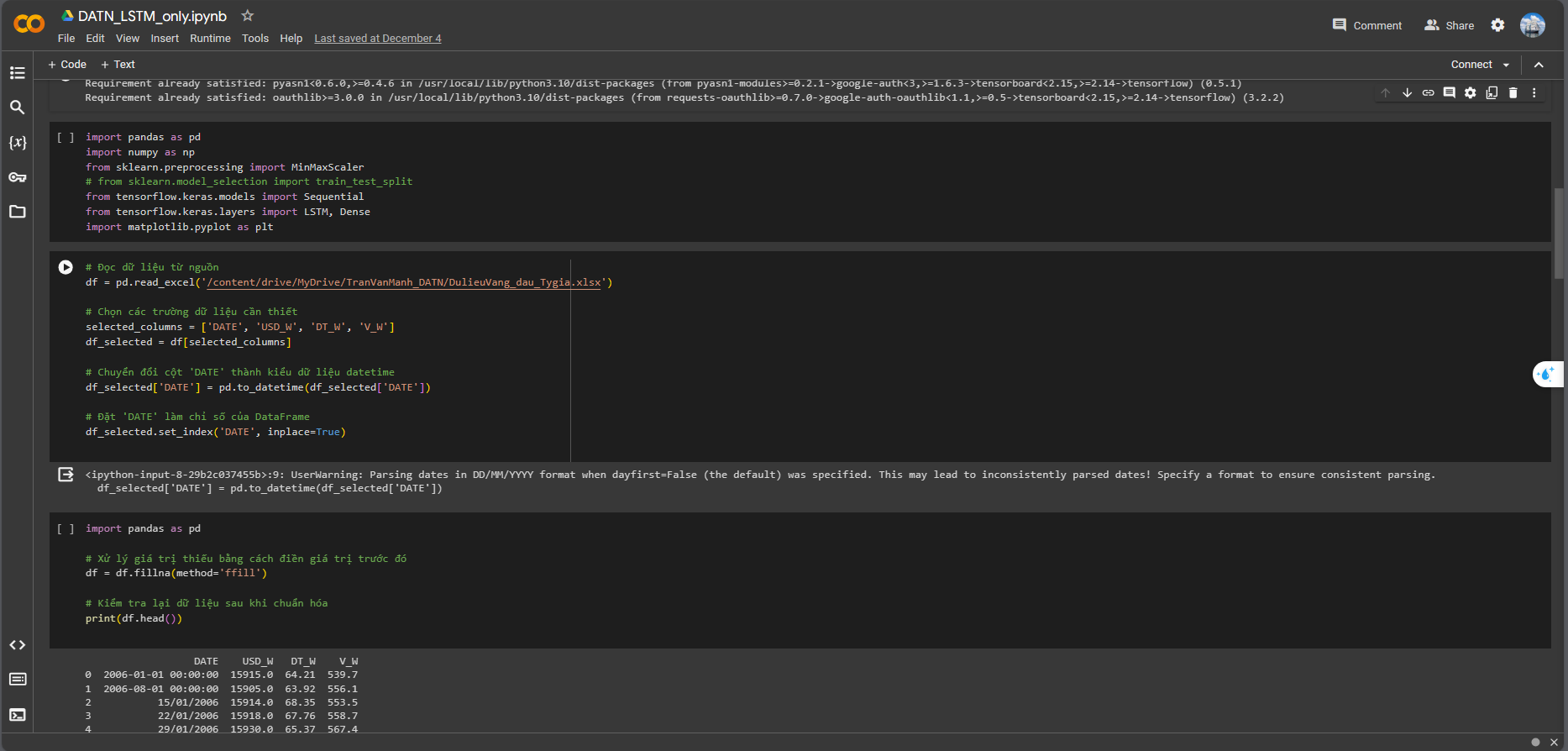
Hình 2. . Một số ứng dụng của thư viện **TensorFlow**

## 2.8 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.

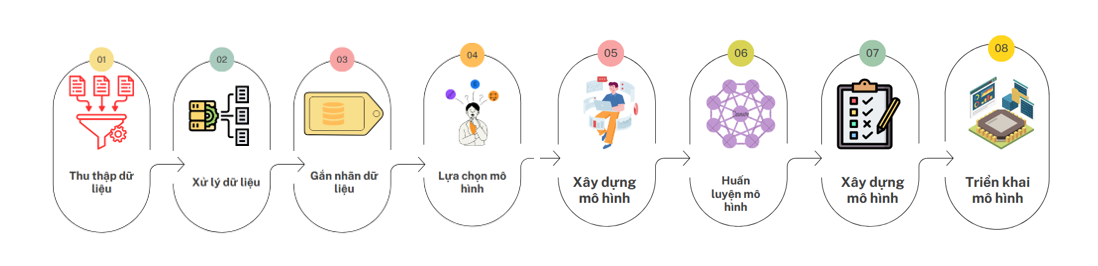


Hình 2. . Ứng dụng google colab trong huấn luyện mô hình học máy

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. **Tổng quan các bước thực hiện bài toán.**



Hình 3. . Tổng quan các bước thực hiện bài toán học máy

* 1. **Thu thập dữ liệu.**

Thu thập dữ liệu là một bước quan trọng trong quá trình phát triển các mô hình học máy. Dữ liệu là nền tảng để các mô hình học máy học hỏi và đưa ra các dự đoán chính xác. Do đó, việc thu thập dữ liệu chất lượng là cần thiết để đảm bảo hiệu quả của các mô hình học máy.

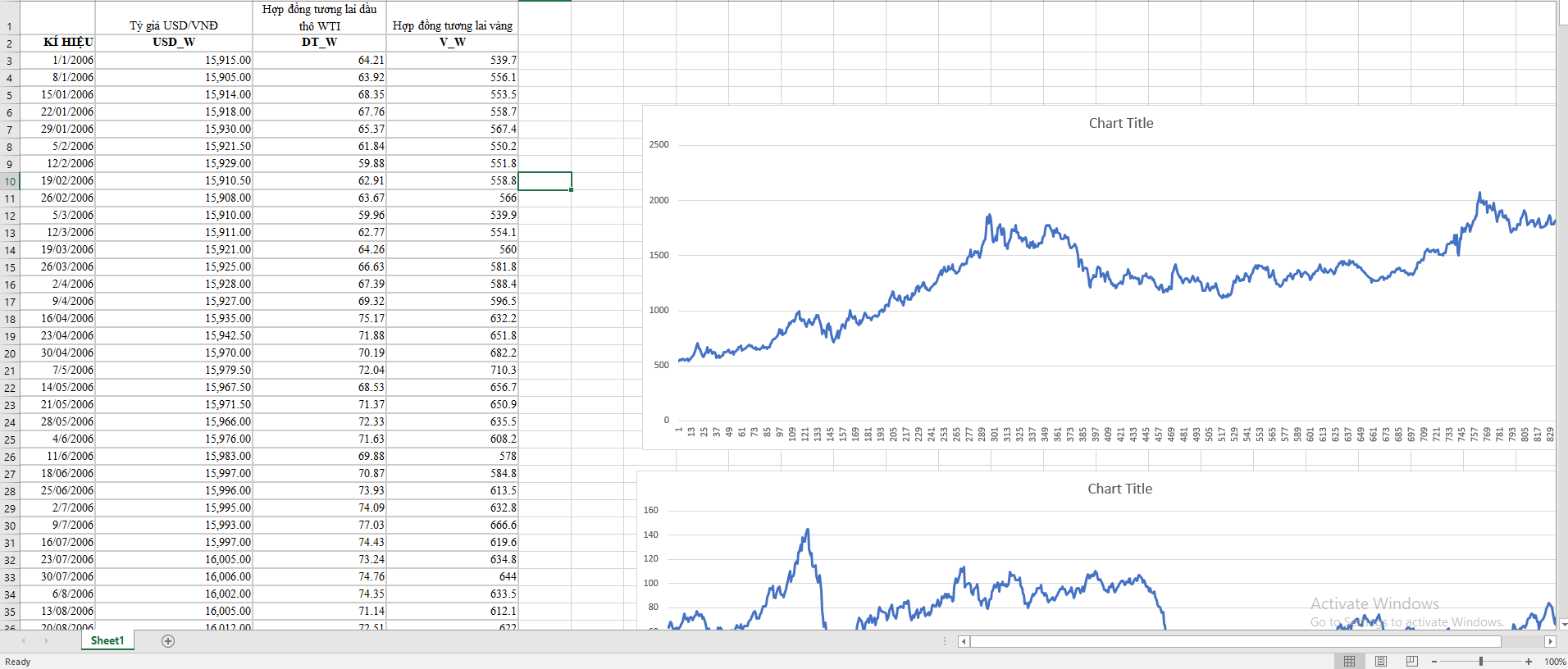
Một tập dữ liệu chất lượng cần phải đáp ứng các tiêu chí sau:

* Đầy đủ: Tập dữ liệu cần bao gồm đầy đủ các thông tin cần thiết để các mô hình học máy có thể học tập và đưa ra các dự đoán chính xác.
* Chính xác: Tập dữ liệu cần chính xác để tránh các sai sót trong quá trình học tập của các mô hình học máy.
* Đa dạng: Tập dữ liệu cần đa dạng để các mô hình học máy có thể học hỏi từ nhiều trường hợp khác nhau và đưa ra các dự đoán chính xác hơn.

Trong bài toán này em lấy dữ liệu từ:

* 1. **Xử lý dữ liệu.**

1. **Phân tích và mô tả dữ liệu**



Hình 3. . Tổng quan bộ 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 dữ liệu**

1. **Tiền xử lý dữ liệu**

Xử lý các giá trị NaN tồn tại trong dữ liệu bằng phương thức 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.
* value: Giá trị sẽ được sử dụng để thay thế NaN.
* inplace: Nếu True, thay đổi trực tiếp trên DataFrame và không trả về DataFrame mới.
* limit: Số lượng giá trị thay thế tối đa.

Xóa dữ liệu trùng lặp:

* Xử dụng phương thức drop\_duplicates() để xóa dữ liệu trùng, các tham số:
* subset: Các cột để kiểm tra sự trùng lặp. Nếu không được chỉ định, tất cả các cột sẽ được xem xét.
* keep: Xác định xem phải giữ lại hàng đầu tiên hay cuối cùng khi có hàng trùng lặp. Giá trị mặc định là 'first'.
* inplace: Nếu True, thay đổi trực tiếp trên DataFrame và không trả về DataFrame mới.
* ignore\_index: Nếu True, tạo lại chỉ số (index) của DataFrame sau khi loại bỏ hàng trùng lặp.

1. **Chuẩn hóa dữ liệu**

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. **Lựa chọn mô hình học máy.**

Bài toán dự báo một số chỉ số kinh tế yêu cầu một mô hình có thể xử lý và lưu trữ thông tin trong thời gian dài và giải quyết vấn đề biến mất gradient thứ khiến thông tin quan trọng sẽ bị mất đi theo thời gian. Để giải quyết những vấn đề trên chúng ta cần một mô hình có khả năng lưu trữ thông tin dài hạn và giảm thiểu sự mất mát thông tin theo thời gian với độ chính xác cao.

Từ kết quả của bộ dữ liệu sau khi thu thập, các chỉ số của dữ liệu có sự biến động theo thời gian đồng thời kết hợp với việc tìm hiểu các mô hình học máy thì thấy mô hình LSTM (Long Sort Term Memory) có khả năng ghi nhớ dữ liệu dài hạn phù hợp để dự đoán các chỉ số kinh tế.

* 1. **Xây dựng mô hình.**

1. Chuẩn bị dữ liệu.

Mục đích của việc chuẩn bị dữ liệu cho LSTM là để giúp mô hình LSTM học được các mối quan hệ giữa các giá trị đầu vào và đầu ra trong một khoảng thời gian nhất định. Việc chuẩn bị dữ liệu theo cách này sẽ giúp mô hình LSTM học được các mối quan hệ này một cách hiệu quả hơn, dẫn đến kết quả dự đoán chính xác hơn.

Xây dựng một hàm chuẩn bị dữ liệu là prepare\_date với hai tham số là

data: Dữ liệu đầu vào

time\_steps: Số lượng bước thời gian quan sát trước đó (khoảng thời gian đầu ra từ noron này trở thành đâu vào phép tính của noron tiếp theo)

Hàm thực hiện các bước như sau:

Bước 1: Kiểm tra xem dữ liệu đầu vào có rỗng hay không. Nếu dữ liệu đầu vào rỗng, hàm sẽ báo lỗi.

Bước 2: Kiểm tra xem time\_steps có nhỏ hơn chiều dài của dữ liệu hay không. Nếu time\_steps lớn hơn hoặc bằng chiều dài của dữ liệu, hàm sẽ báo lỗi.

Bước 3: Khởi tạo hai mảng X và y (X là nơi lưu trữ giá trí đầu vào trước đó time\_steps, y lưu giá trị đầu ra tại bước hiện tại).

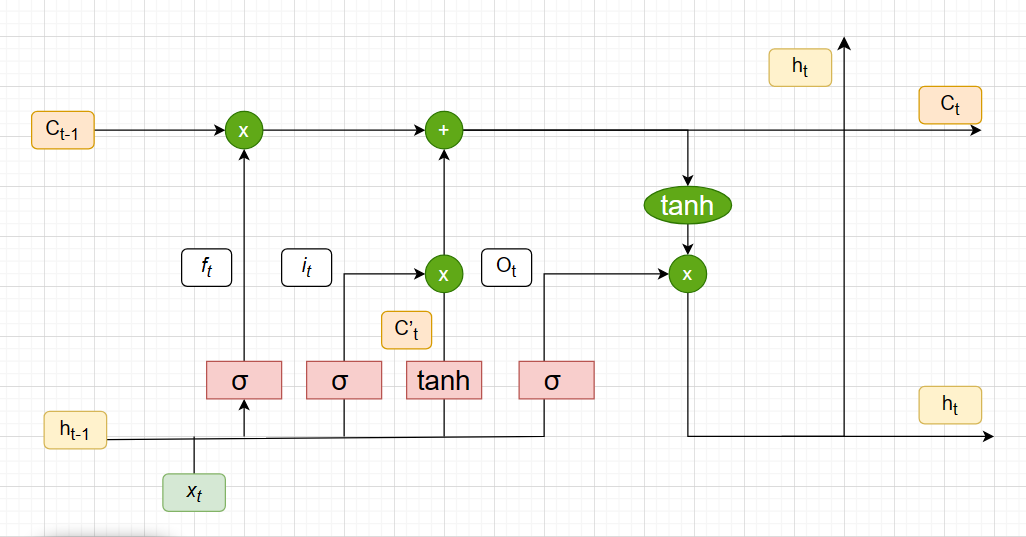
Bước 4: Lặp qua toàn bộ dữ liệu đầu vào.

Bước 5: Tại mỗi bước, lấy time\_steps giá trị đầu vào trước đó và lưu vào mảng X.

Bước 6: Lưu giá trị đầu ra tại bước hiện tại vào mảng y.

Bước 7: Trả về hai mảng X và y.

1. Xây dựng mô hình LSTM(Long short term memory).



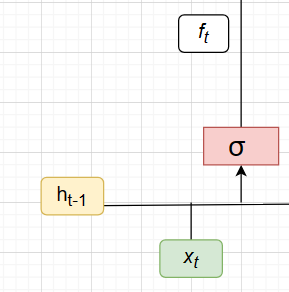
Hình 3. . Cách hoạt động của mô hình LSTM

Ở trạng thái thứ t của mô hình LSTM ta có:

* Output: Ct,ht, gọi c là cell state, h là hidden state.
* Input: Ct-1,ht-1, *xt.*Trong đó *xt* là dữ liệu input ở trạng thái thứ t của mô hình. Ct-1,ht-1 là output của lớp trước.
* Ký hiệu σ , tanh: là ký hiệ u của hàm kích hoạt sigmoid, hàm kích hoạt tanh
* Ký hiệu *ft*, *it*,ot: tương ứng forget gate, input gate, output gate.

Mô hình hoạt động theo các bước sau:

**Bước 1**: Mô hình nhận 2 đầu vào là 2 giá trị *h*t-1 và *x*t trả về một giá trị nằm trong khoảng của hàm kích hoạt tầng quên (forget gate) là sigmoid với khoảng [0,1] cho mỗi giá trị của ô trạng thái Ct-1. Nếu giá trị bằng 1 thì giữ lại thông tin và bằng không thì loại bỏ thông tin.



Hình 3. . cổng quên (forget gate) của mô hình LSTM

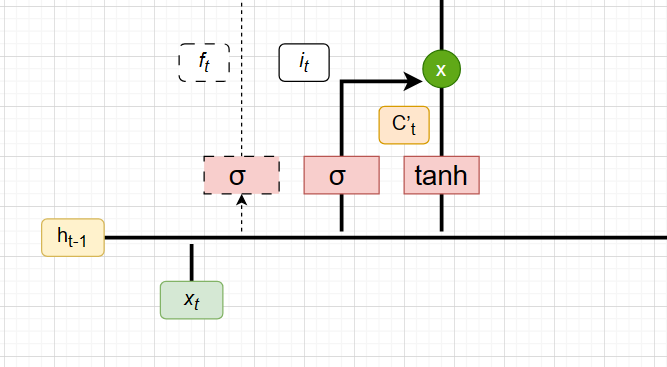
Ta có công thức cổng forget gate: *ft* = σ(*Wf* \* [ht-1, *xt*] + b*f*)

**Bước 2**: quyết định loại thông tin nào sẽ được lưu vào ô trạng thái (cell state). Bước này có 2 phần:

Phần 1: Một tầng ẩn của hàm sigmoid được gọi là cổng vào (input gate layer) sẽ quyết định giá trị được cập nhật.

Phân 2: Tiếp theo, tầng ẩn hàm tanh sẽ được tạo ra một vector của giá trị của trạng thái mới C’t mà có thể được thêm vào trạng thái.

Cuối cùng nhân kết quả 2 tầng để tại thành một cập nhất mới cho ô trạng thái.

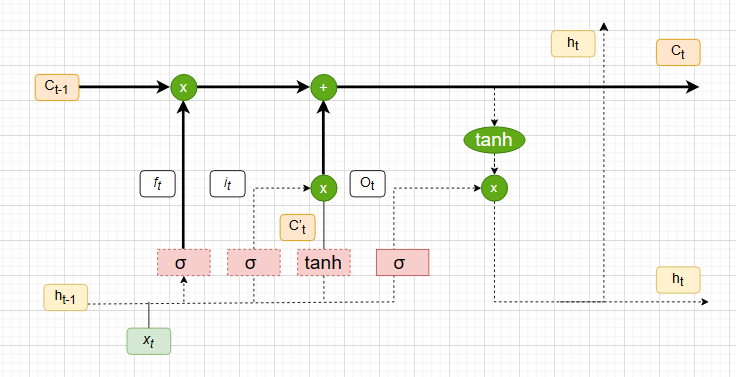


Hình 3. . Cổng vào (input gate) của mô hình LSTM

Ta có công thức cổng tính input gate:  
 a. *i*t = σ(*Wi* \* [ht-1, *xt*] + bi) .

b. C’t = tanh(Wc\*[ht-1, xt] + bC).

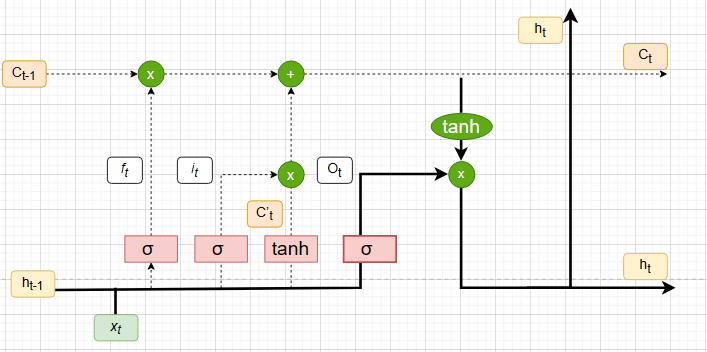
Đây là thời điểm cập nhật một ô trạng thái cũ từ Ct-1 sang ô trạng thái mới là Ct. Chúng ta nhân trạng thái cũ với *ft*  tương ứng với quên những quyết định được phép quên sớm. Phần tử đề cử *it* \* C’t là một giá trị mới được tính toán tương ứng với các giá trị được cập nhật và mỗi giá trị trạng thái.



Hình 3. . Trạng thái mới Ct của mô hình LSTM

Trạng thái mới: Ct = *f*t \* Ct-1 + *i*t \* C’t

**Bước 3**: bước này sẽ quyết định đẩu ra trả về bao nhiêu. Kết quả sẽ dữa vào ô trạng thái, những sẽ được lọc. Ban đầu, các giá trị chạy qua một tầng sigmoid nơi quyết định phần nào của ô trạng thái sẽ ở đầu ra. Tiếp theo, ô trạng thái được đưa qua hàm tanh [-1;1] và nhân nó với đầu ra của một cổng sigmoid, do đó chỉ trả ra phần mà chúng ta quyết định.



Hình 3. . Điều chỉnh thông tin đầu ra mô hình LSTM

Công thức điểu chỉnh đẩu ra:

ot = σ(*Wo* \* [ht-1, *xt*] + bo) .

ht = ot \* tanh(Ct).

* 1. **Huấn luyện mô hình.**

Huấn luyện mô hình trong học máy là quá trình sử dụng dữ liệu để giúp mô hình học cách đưa ra dự đoán hoặc phân loại dữ liệu mới. Quá trình này thường được thực hiện bằng cách sử dụng một thuật toán học máy để điều chỉnh các tham số của mô hình cho đến khi mô hình có thể đưa ra các dự đoán chính xác.

Chúng ta sẽ huấn luyện mô hình với số lần huấn luyện1000 lần(epoches). Với số mẫu cho mỗi lần huấn luyện là 32.

Huấn luyện mô hình LSTM bao gồm các bước sau:

Bước 1: Chuẩn bị dữ liệu: Dữ liệu cần được chuẩn bị trước khi huấn luyện mô hình LSTM. Điều này bao gồm việc loại bỏ các dữ liệu không cần thiết, các dữ liệu trùng lặp.

Đối với dữ liệu chuỗi thời gian trong bài toán này, chúng ta cần chuẩn bị dữ liệu theo các bước sau:

- Cắt chuỗi thời gian thành các đoạn nhỏ, mỗi đoạn có độ dài là số bước thời gian(time\_steps).

- Đặt mỗi đoạn dữ liệu thành một mảng NumPy.

Bước 2: Phân chia dữ liệu: Dữ liệu cần được chia thành hai tập, tập huấn luyện và tập kiểm thử. Tập huấn luyện sẽ được sử dụng để huấn luyện mô hình LSTM, và tập kiểm thử sẽ được sử dụng để đánh giá mô hình.

Tỷ lệ phân chia giữa tập huấn luyện và tập kiểm thử thường là 80:20.

Bước 3: Khởi tạo tham số: Tham số của mô hình LSTM cần được khởi tạo trước khi bắt đầu quá trình huấn luyện. Tham số của mô hình LSTM thường được khởi tạo ngẫu nhiên.

Bước 4: Huấn luyện mô hình: Mô hình LSTM sẽ được huấn luyện bằng dữ liệu tập huấn luyện. Quá trình huấn luyện sẽ được thực hiện lặp đi lặp lại cho đến khi đủ số epochs.

Trong mỗi lần lặp, mô hình LSTM sẽ thực hiện các bước sau:

* Tính toán đầu ra của mô hình.
* Tính toán hàm mất mát.
* Cập nhật tham số của mô hình để giảm thiểu hàm mất mát.
  1. **Đánh giá mô hình.**

Sử dụng chỉ số MSE(Mean Square Error) nó sẽ tính trung bình sai số giữ giá trị thực tế và giá trị dự đoán.

Ta có công thức tính MSE:

2

Trong đó:

* Kí hiệu n: số lượng mẫu trong tập dữ liệu.
* Kiếu hiệu Yi:giá trị thực tế của mẫu thứ i.
* Ký hiệu Y’i:giá trị dự đoán của mô hình cho mẫu thứ i.

Với 1000 lần huấn luyện mô hình và với 32 mẫu mỗi lần ta có kết quả như sau:

* Chỉ số MSE cho tỷ giá USD/VND (USD\_W) là:

MSE: 0.00014461538566912197

* Chỉ số MSE cho tỷ giá USD/VND (USD\_W) là:

MSE: 0.0015033384146075593

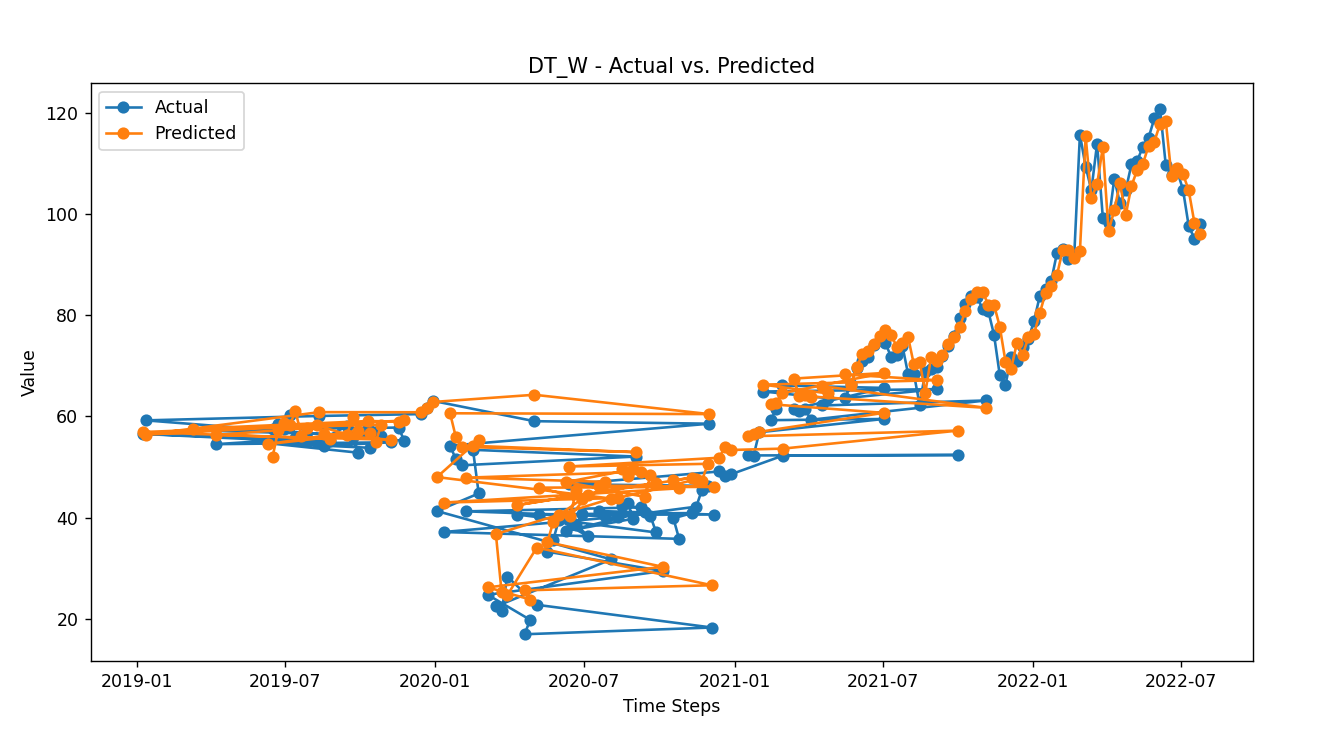
* Chỉ số MSE cho tỷ giá USD/VND (USD\_W) là:

MSE: 0.0007819797512754789

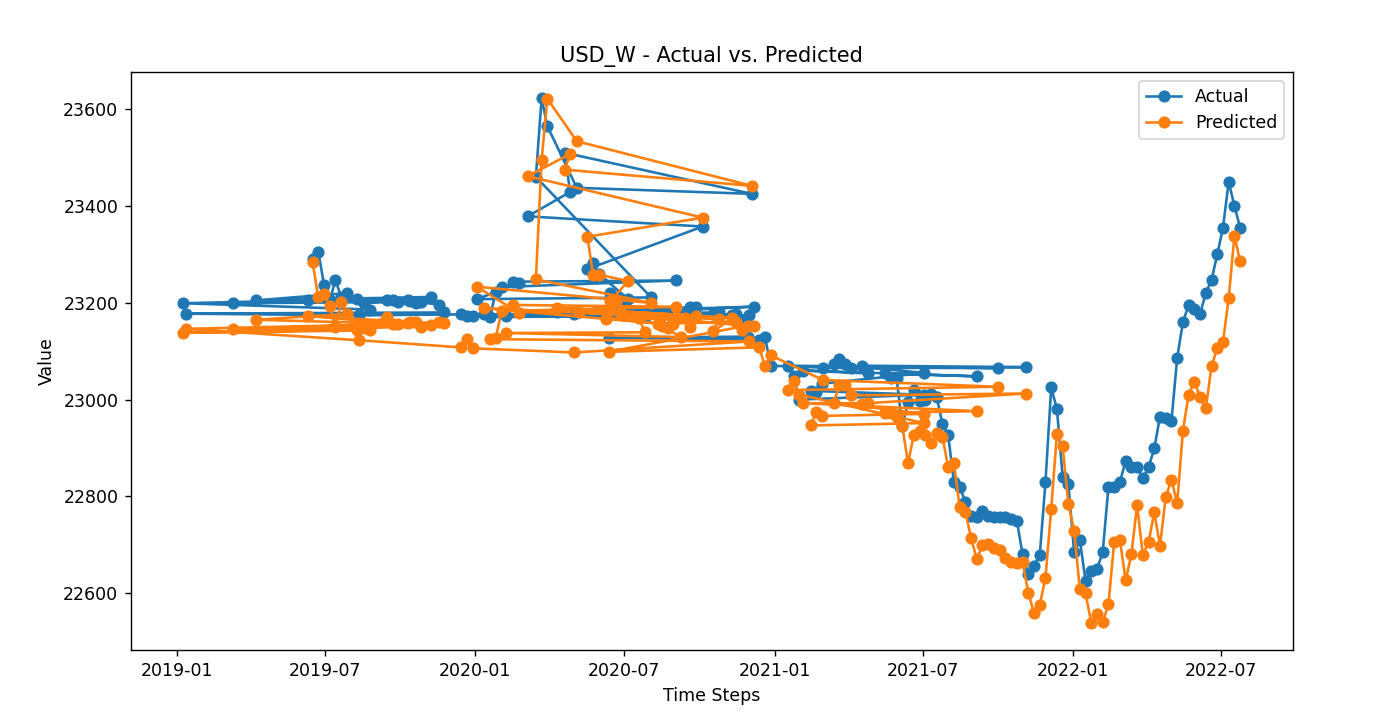
Dựa vào kết quả trên ta thấy:

* MSE của cột USD\_W là rất thấp, chỉ 0.0001446, điều này cho thấy mô hình có khả năng dự đoán giá trị của cột này với độ chính xác cao trên tập kiểm tra. Điều này có thể làm cho mô hình của bạn hữu ích trong việc dự đoán và theo dõi biến động của tỷ giá USD/VNĐ.
* MSE của cột DT\_W là 0.0015033, là một giá trị nhỏ nếu so sánh với giá trị thực tế. Mô hình vẫn cho thấy khả năng dự đoán khá tốt. Điều này có thể làm cho mô hình trở nên hữu ích trong việc đánh giá xu hướng hoặc biến động trong thị trường dầu thô.
* MSE của cột V\_W là 0.0007819, cũng rất thấp, chỉ ra rằng mô hình làm rất tốt trong việc dự đoán giá trị của cột này. Điều này có thể hữu ích trong việc theo dõi thị trường vàng.

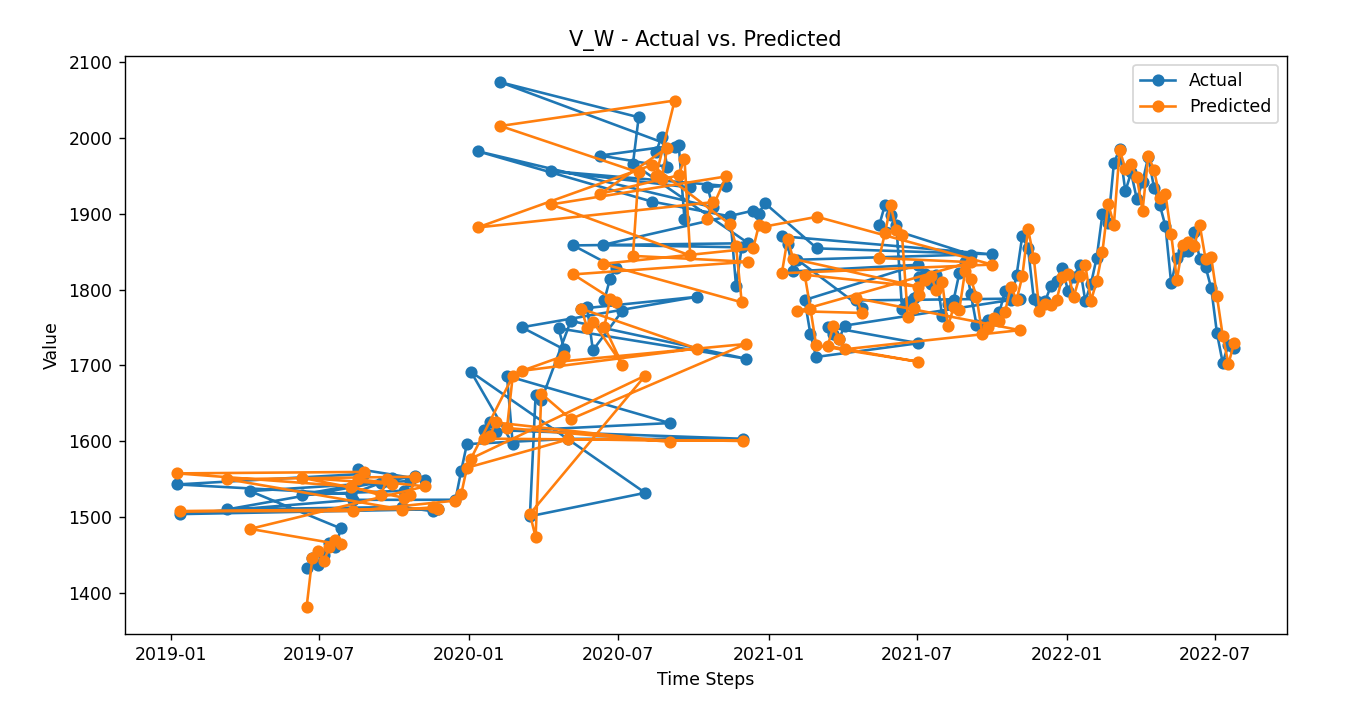
Trực quan hóa mô hinh bằng thư viện matplotlib:



Hình 3. . So sánh dữ liệu cột DT\_W thực tế và dự đoán



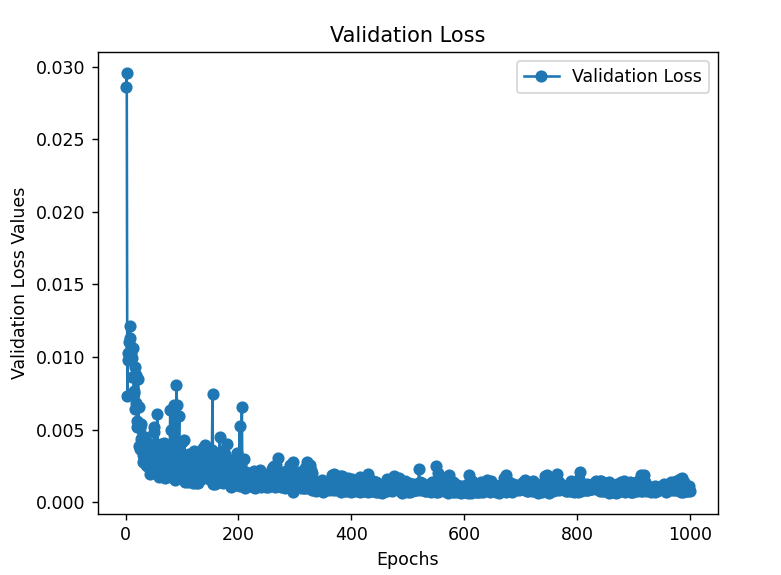
Hình 3. . So sánh dữ liệu cột USD\_W thực tế và dự đoán



Hình 3. . So sánh dữ liệu cột V\_W thực tế và dự đoán

Kết hợp với biểu đồ biểu của các giá trị loss validation:

Loss validation là giá trị hàm mất mát được tính toán trên tập dữ liệu validation. Giá trị loss validation càng nhỏ thì mô hình càng tốt. Giá trị loss validation tăng lên có thể là dấu hiệu cho thấy mô hình đang bị overfitting.



Hình 3. . các giá trị hàm mất mát được tính toán trên tập dữ liệu

# THỰC NGHIỆM, SO SÁNH VÀ ĐÁNH GIÁ

* 1. **Thực nghiệm.**
     1. **Chuẩn bị thiết bị và dữ liệu**
* Sử dụng máy tính cá nhân và huấn luyện mô hình trên Google Colab, hay còn gọi là Colaboratory cung cấp 12.7GB RAM, bộ nhớ 78GB, GPU 15GB.
* Bộ dự liệu gồm các chỉ số về kinh tế gồm: Tỷ giá USD/VNĐ(USD\_W), Hợp đồng tương lai dầu thô WTI (DT\_W), Hợp đồng tương lai vàng (V\_W)
  + 1. **Thực hiện**

1. Chia tập dữ liệu

Chia tập dữ liệu theo tỷ lệ: 80% cho tập huấn luyện và 20% cho tập thử nghiệm. Ở đây chúng ta sẽ sử dụng hàm train\_test\_split trong thư viện Scikit-learn của Python với :

X: là tập dữ liệu đầu vào.

y: là tập dữ liệu đầu ra.

test\_size: Tỷ lệ phần trăm dữ liệu được sử dụng cho tập thử nghiệm. Giá trị mặc định là 0.2, nghĩa là 20% dữ liệu sẽ được sử dụng cho tập thử nghiệm.

random\_state: Số ngẫu nhiên được sử dụng để chia dữ liệu. Giá trị mặc định là None, nghĩa là dữ liệu sẽ được chia ngẫu nhiên.

shuffle: giúp trộn dữ liệu nhưng với mô hình dự đoán chuội thời gian và tập dữ liệu theo thời gian thì điều này là không cần thiết nên trong bài này chúng ta để shuffle là false.

1. Xây dựng hàm chuẩn bị dữ liệu cho mô hình

Mô hình LSTM cần dữ liệu được tổ chức theo dạng các chuỗi thời gian liên tiếp. Bước này giúp tạo ra các cặp dữ liệu đầu vào (X) và đầu ra (y) từ dữ liệu chuỗi thời gian, trong đó X là một phần của chuỗi thời gian và y là giá trị tiếp theo sau một số bước thời gian.

ta có hàm prepare\_data với hai tham số là dữ liệu đầu vào (data) và khoảng thời gian đầu ra từ noron này trở thành đâu vào phép tính của noron tiếp theo (time\_steps).

hàm hoạt động như sau:

Bước 1: Kiểm tra xem dữ liệu đầu vào có rỗng hay không. Nếu dữ liệu đầu vào rỗng, hàm sẽ báo lỗi.

Bước 2: Kiểm tra xem time\_steps có nhỏ hơn chiều dài của dữ liệu hay không. Nếu time\_steps lớn hơn hoặc bằng chiều dài của dữ liệu, hàm sẽ báo lỗi.

Bước 3: Khởi tạo hai mảng X và y (X là nơi lưu trữ giá trí đầu vào trước đó time\_steps, y lưu giá trị đầu ra tại bước hiện tại).

Bước 4: Lặp qua toàn bộ dữ liệu đầu vào.

Bước 5: Tại mỗi bước, lấy time\_steps giá trị đầu vào trước đó và lưu vào mảng X.

Bước 6: Lưu giá trị đầu ra tại bước hiện tại vào mảng y.

Bước 7: Trả về hai mảng X và y.

1. Xây dựng mô hình

Bước 1: tạo một mô hình tuần tự bằng Sequential() của thư viện Keras

Bước 2: thêm lớp LSTM vào Sequential() với các tham số là units(số lượng noron) thử với units = 50, activation(hàm kích hoạt) ở mô hình này ta sử dụng hàm

tanh, input\_shape(xác định hình dạng đầu vào của mô hình)

Bước 3: thêm lớp Dense vào Sequential() là để chuyển các đặc trưng của mô hình thành đầu ra tương ứng với các chỉ số kinh tế trong tập dữ liệu là 3 thì ta có units trong lớp Dense sẽ là 3.

Bước 4: cấu hình mô hình trước khi đưa vào huấn luyện bằng phương thức compile và có các tham số là optimizer(thuật toán tối ưu hóa), loss(hàm mất mát).

ở bài toán này em lựa chọn mô thuật toán tối ưu hóa là Adam và hàm mất mát là Mse(Mean square error) đã được để cập trước đó.

công thức của thuật toán tối ưu là:

mt = β1\*mt-1 + (1- β1)\*gt

vt = β2\*vt-1 + (1- β2)\*gt2

1. Huấn luyện mô hình

Huấn luyện mô hình với số lần huấn luyện là 1000 lần và với số mẫu mỗi lần là 32

* + 1. **Ghi nhận kết quả**

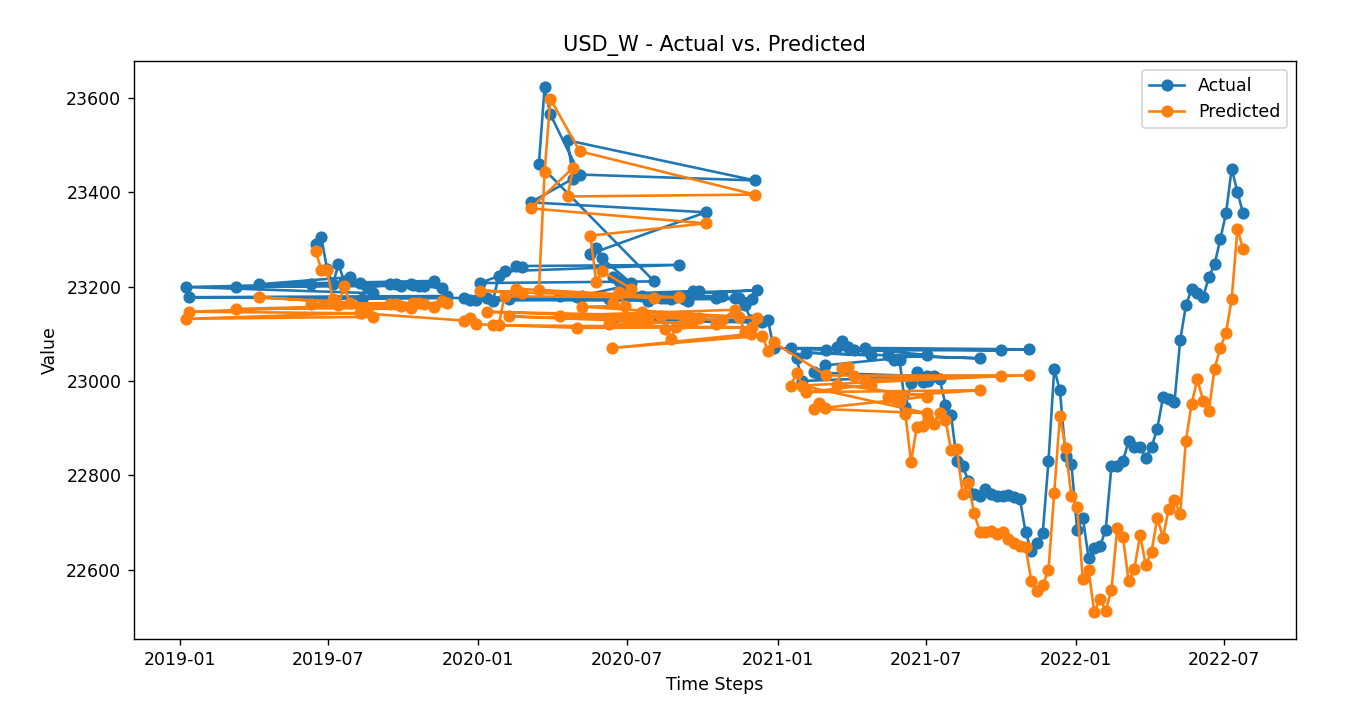
Kết quả chỉ số MSE:

MSE cho chỉ số USD\_W: 0.00021160080511499732

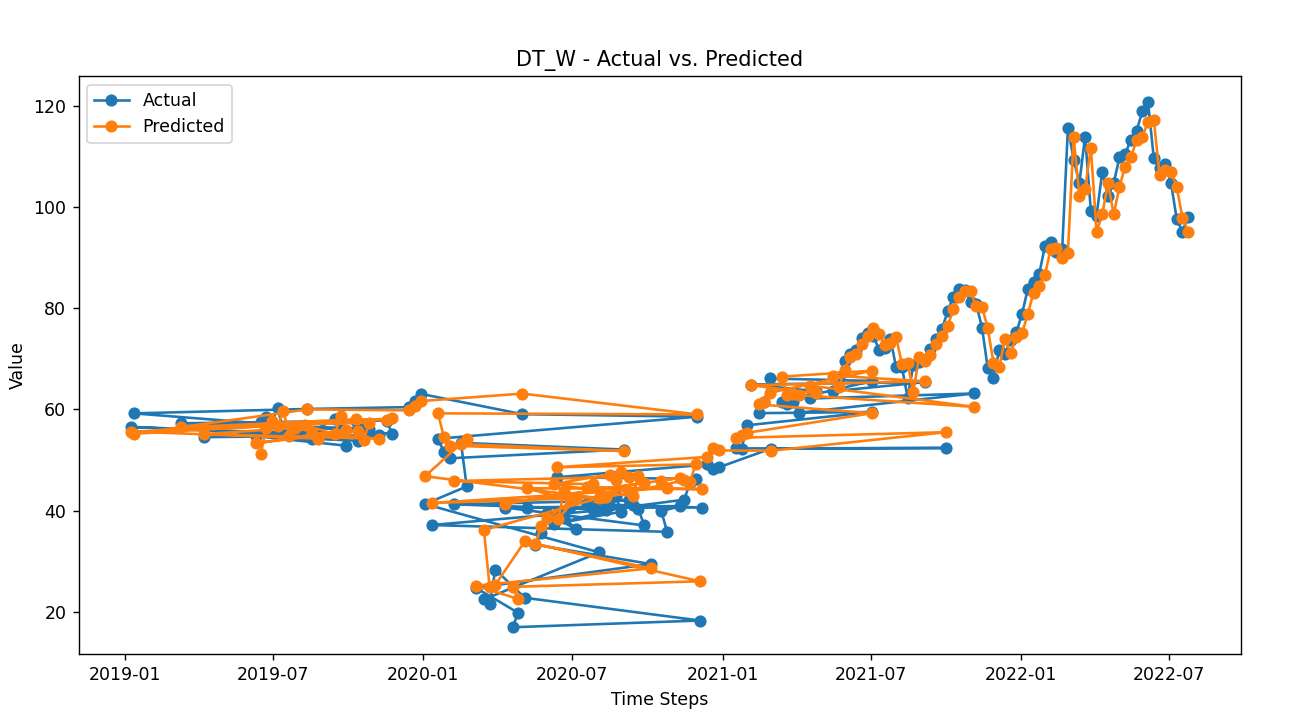
MSE cho chỉ số DT\_W: 0.0012323577986692103

MSE cho chỉ số V\_W: 0.001320481007271706

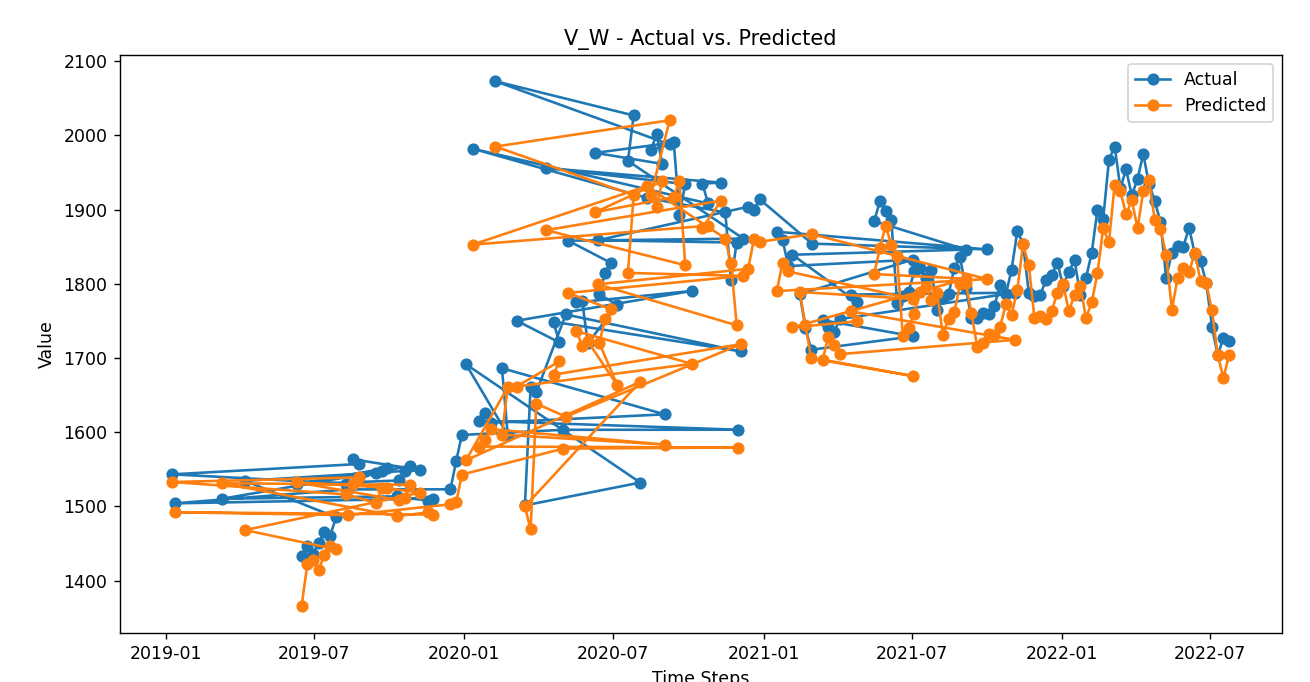
kết quả trực quan hóa:

chỉ số USD\_W

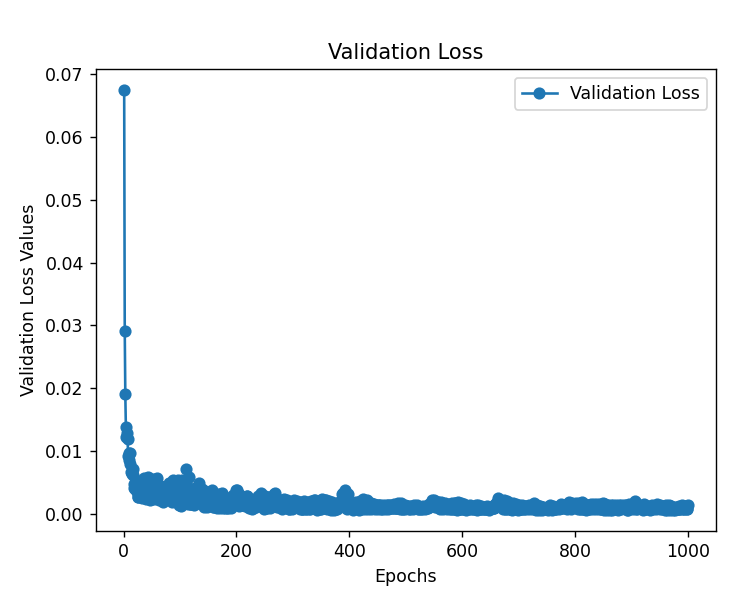
Hình 4. Kết quả trực quan hóa thực nghiệm trên dữ liệu USD\_W



Hình 4. . Kết quả trực quan hóa thực nghiệm trên dữ liệu DT\_W



Hình 4. . Kết quả trực quan hóa thực nghiệm trên dữ liệu V\_W



Hình 4. các giá trị loss validation trong thực nghiệm mô hình

Điều chỉnh mô hình với các siêu tham số

Số đơn vị ẩn LSTM (units): phạm vi [32, 64, 128, 256]

Activation: phạm vi ['sigmoid', 'tanh', 'relu']

Dropout\_rate: phạm vi [0.1, 0.2, 0.25, 0.4, 0.5, 0.6]

Learning\_rate: phạm vi [0.001, 0.005, 0.01]

Các siêu tham số sẽ được tối ưu hóa bằng phương pháp tìm kiếm ngẫu nhiên (random search)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | units | activation | dropout\_rate | learning\_rate | mse USD\_W | mse  DT\_W | mse  VW |
| 1 | 256 | sigmoid | 0.1 | 0.001 | 0.00023959644162248613 | 0.0032285221450377853 | 0.0008137320068528437 |
| 2 | 32 | tanh | 0.2 | 0.005 | 0.0003124462545316641 | 0.0031413996857290183 | 0.0012091686063965186 |
| 3 | 32 | relu | 0.25 | 0.01 | 0.026233877530042325 | 0.010855252680197615 | 0.03981338142085337 |
| 4 | 64 | relu | 0.2 | 0.005 | 0.0249598142600923 | 0.005569586503717791 | 0.015414467408505833 |
| 5 | 128 | sigmoid | 0.1 | 0.01 | 0.01386229011170891 | 0.0029195291349626983 | 0.004110597574732407 |
| 6 | 64 | relu | 0.5 | 0.01 | 0.039247434927172975 | 0.013100905566182383 | 0.042874402775925254 |
| 7 | 32 | relu | 0.2 | 0.005 | 0.027431956866914995 | 0.006728494829258076 | 0.03219558941062228 |
| 8 | 64 | tanh | 0.2 | 0.01 | 0.009485725211171858 | 0.0030950756050748427 | 0.001921846362065696 |
| 9 | 256 | sigmoid | 0.5 | 0.005 | 0.0006903165823452539 | 0.0033198824971980194 | 0.002044020974435606 |
| 10 | 128 | sigmoid | 0.25 | 0.001 | 0.0022373081299222456 | 0.001992975938127149 | 0.0026085330854330696 |
| 11 | 64 | tanh | 0.1 | 0.001 | 9.140344058415682e-05 | 0.0015856517545426861 | 0.0007615532655916349 |
| 12 | 32 | relu | 0.2 | 0.001 | 0.005464975583232948 | 0.001480532083691782 | 0.006496403167220932 |
| 13 | 128 | relu | 0.25 | 0.005 | 0.01936792969744872 | 0.0034132742982846275 | 0.007974984900432146 |
| 14 | 256 | sigmoid | 0.1 | 0.01 | 0.02308563782453565 | 0.004192140974667267 | 0.014247769527718801 |
| 15 | 32 | relu | 0.1 | 0.01 | 0.014980257769157863 | 0.00792069028000999 | 0.015627688885598005 |

* 1. **Phân tích kết quả thực nghiệm.**

Giá trị MSE cho USD\_W là 0.00021160080511499732, tương đương với sai số trung bình khoảng 0.02.

Giá trị MSE cho DT\_W và V\_W cũng tương đối nhỏ, lần lượt là 0.0012323577986692103 và 0.001320481007271706, tương đương với sai số trung bình khoảng 0.12 và 0.13.

Qua nhiều lần thử nghiệm các siêu tham số chúng ta thây được những tham số tối ưu cho mô hình là :

Units: 32

Activation: relu

Dropout rate: 0.2

Learning rate: 0.001

Các siêu tham số này được lựa chọn dựa trên các tiêu chí sau:

* Độ chính xác: Các siêu tham số này cho ra độ chính xác cao nhất cho chỉ số USD\_W.
* Thời gian huấn luyện: Các siêu tham số này giúp giảm thời gian huấn luyện mà không làm giảm đáng kể độ chính xác của mô hình.

Units: Số lượng đơn vị (units) là số lượng nút trong mỗi lớp ẩn của mạng LSTM. Số lượng đơn vị càng lớn thì mạng LSTM càng có thể học được các mối quan hệ phức tạp giữa các biến. Tuy nhiên, số lượng đơn vị quá lớn có thể dẫn đến quá khớp (overfitting). Trong trường hợp này, số lượng đơn vị là 32 là đủ để mạng LSTM học được các mối quan hệ phức tạp cần thiết mà không bị overfitting.

Activation: Hàm kích hoạt (activation) là hàm được sử dụng để tính toán giá trị đầu ra của mỗi nút trong mạng LSTM. Hàm relu là một hàm kích hoạt phổ biến, có thể học được các mối quan hệ phi tuyến phức tạp. Trong trường hợp này, hàm relu cho ra độ chính xác tốt nhất.

Dropout rate: Dropout là một kỹ thuật ngăn chặn overfitting. Dropout sẽ ngẫu nhiên loại bỏ một số nút khỏi mạng LSTM trong quá trình huấn luyện. Điều này giúp ngăn chặn các nút phụ thuộc quá mức vào nhau. Trong trường hợp này, tỷ lệ dropout là 0.2 là đủ để ngăn chặn overfitting mà không làm giảm đáng kể hiệu quả của mô hình.

Learning rate: Learning rate là tốc độ mà mô hình học hỏi. Learning rate quá cao có thể dẫn đến overfitting, trong khi learning rate quá thấp có thể khiến mô hình mất quá nhiều thời gian để học. Trong trường hợp này, learning rate là 0.001 là một giá trị hợp lý, giúp mô hình học hỏi hiệu quả mà không bị overfitting.

* 1. **So sánh với các phương pháp khác.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | LSTM | GRU | RNN |
| MSE USD\_W | 0.00012156719264315488 | 0.0010044152872105286 | 0.0024160506976038513 |
| MSE DT\_W | 0.0024959825379537166 | 0.003642753527877022 | 0.006634826881925219 |
| MSE V\_W | 0.002119580132330438 | 0.002610915782679055 | 0.00792069028000999 |

* 1. **Đánh giá hiệu xuất và phân tích kết quả.**

Dựa trên kết quả MSE, ta có thể rút ra các nhận xét sau:

LSTM:

Dựa trên kết quả MSE, ta có thể rút ra các nhận xét sau:

LSTM:

MSE của LSTM cho các chỉ số USD\_W, DT\_W, và V\_W lần lượt là 0.000121567, 0.00249598 và 0.00211958.

Các giá trị MSE của LSTM đều thấp hơn so với GRU và RNN.

LSTM cho thấy hiệu suất tốt nhất trong việc dự đoán các chỉ số kinh tế, với giá trị MSE nhỏ nhất.

GRU:

MSE của GRU cho các chỉ số USD\_W, DT\_W, và V\_W lần lượt là 0.00100442, 0.00364275 và 0.00261092.

GRU có giá trị MSE lớn hơn so với LSTM, nhưng vẫn nhỏ hơn so với RNN.

GRU cho thấy hiệu suất tốt hơn so với RNN, nhưng vẫn kém hơn so với LSTM.

RNN:

MSE của RNN cho các chỉ số USD\_W, DT\_W, và V\_W lần lượt là 0.00241605, 0.00663483 và 0.00792069.

RNN có giá trị MSE lớn nhất trong cả ba mô hình.

RNN cho thấy hiệu suất kém hơn so với LSTM và GRU trong việc dự đoán các chỉ số kinh tế.

Dựa trên kết quả trên, ta có thể kết luận rằng LSTM cho thấy hiệu suất tốt nhất trong việc dự đoán các chỉ số kinh tế, với MSE thấp nhất. GRU cũng cho thấy hiệu suất tốt hơn so với RNN, nhưng vẫn kém hơn so với LSTM. RNN cho thấy hiệu suất kém nhất trong việc dự đoán các chỉ số kinh tế.

LSTM:

MSE của LSTM cho các chỉ số USD\_W, DT\_W và V\_W lần lượt là 0.000121567, 0.002495982 và 0.002119580.

LSTM cho thấy hiệu suất tốt nhất trong việc dự đoán chỉ số kinh tế, với các giá trị MSE thấp nhất trong cả ba mô hình.

Điều này cho thấy LSTM có khả năng mô hình hóa và dự đoán chuỗi thời gian tốt hơn cả GRU và RNN trong bài toán này.

GRU:

MSE của GRU cho các chỉ số USD\_W, DT\_W và V\_W lần lượt là 0.001004415, 0.003642754 và 0.002610916.

GRU cho thấy hiệu suất tốt hơn so với RNN, nhưng vẫn kém hơn so với LSTM.

Mặc dù MSE của GRU cao hơn so với LSTM, nhưng vẫn ở mức chấp nhận được và có thể được sử dụng trong bài toán dự đoán chỉ số kinh tế.

RNN:

MSE của RNN cho các chỉ số USD\_W, DT\_W và V\_W lần lượt là 0.002416051, 0.006634827 và 0.007920690.

RNN cho thấy hiệu suất kém hơn so với cả LSTM và GRU.

MSE cao của RNN có thể cho thấy khả năng mô hình hóa chuỗi thời gian kém hơn so với LSTM và GRU trong bài toán này.

Dựa trên kết quả trên, ta có thể kết luận rằng LSTM cho thấy hiệu suất tốt nhất trong việc dự đoán các chỉ số kinh tế, với MSE thấp nhất. GRU cũng cho thấy hiệu suất tốt hơn so với RNN, nhưng vẫn kém hơn so với LSTM. RNN cho thấy hiệu suất kém nhất trong việc dự đoán các chỉ số kinh tế.

* 1. **Kết luận**

**Ưu điểm của mô hình LSTM:**

Tích hợp thông tin thời gian**:** LSTM được thiết kế đặc biệt để xử lý dữ liệu chuỗi thời gian, giúp mô hình học được mối quan hệ phức tạp giữa các quan sát theo thời gian.

Khả năng học được các mẫu phức tạp**:** LSTM có khả năng học được mô hình chuỗi dữ liệu có tính chất phức tạp và không tuyến tính.

Tính linh hoạt của kiến trúc**:** Mô hình LSTM có thể được tinh chỉnh và điều chỉnh thông qua việc thay đổi siêu tham số và kiến trúc, giúp nó thích ứng với đặc tính cụ thể của dữ liệu.

**Nhược điểm của mô hình LSTM:**

Đòi hỏi lượng dữ liệu lớn: LSTM thường đòi hỏi một lượng lớn dữ liệu để học hiệu quả, đặc biệt là khi có nhiều tham số hoặc khi mô hình rất phức tạp.

Khó tinh chỉnh siêu tham số: Có nhiều siêu tham số cần tinh chỉnh trong mô hình LSTM, và quá trình này có thể đòi hỏi nhiều thời gian và kỹ năng chuyên sâu.

Nguy cơ overfitting: LSTM có thể dễ dàng bị quá mức học từ dữ liệu huấn luyện và không thể tổng quát hóa tốt trên dữ liệu mới (overfitting), đặc biệt khi lượng dữ liệu không đủ.

Thời gian huấn luyện lâu: Huấn luyện mô hình LSTM có thể mất nhiều thời gian, đặc biệt là trên dữ liệu lớn và khi sử dụng phần cứng có hạn.

**Kết quả của mô hình LSTM trong bài toán:**

Hiệu suất tốt**:** Mô hình LSTM đã cho thấy hiệu suất tốt đối với dữ liệu chỉ số kinh tế tại Việt Nam, đặc biệt là cho cột 'USD\_W'.

Khả năng dự đoán chuỗi thời gian**:** LSTM đặc biệt hiệu quả trong việc dự đoán chuỗi thời gian, giúp bạn có cái nhìn tốt về xu hướng và biến động của chỉ số kinh tế.

**Hướng phát triển tương lai cho mô hình LSTM:**

Tinh chỉnh thêm siêu tham số**:** Có thể tiếp tục điều chỉnh siêu tham số như số lượng units, hàm kích hoạt, và learning rate để cải thiện hiệu suất.

Mở rộng dữ liệu**:** Bổ sung thêm dữ liệu có thể cải thiện khả năng dự đoán của mô hình và giúp nó học được mô hình dữ liệu thị trường thực tế.

Kết hợp với dữ liệu khác**:** Kết hợp với các biến số khác, như dữ liệu thời tiết, giá dầu, hoặc các yếu tố kinh tế quốc tế để cung cấp thông tin bổ sung cho mô hình.

# TÀI LIỆU THAM KHẢO

[1] Chollet, F. (2017). Deep Learning with Python. Manning Publications.

[Deep Learning with Python (wordpress.com)](https://tanthiamhuat.files.wordpress.com/2018/03/deeplearningwithpython.pdf)

[2] Brownlee, J. (2018). Deep Learning for Time Series Forecasting. Machine Learning Mastery.

[Deep Learning for Time Series Forecasting: Predict the Future with MLPs ... - Jason Brownlee - Google Sách](https://books.google.com.vn/books?id=o5qnDwAAQBAJ&printsec=frontcover&hl=vi&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)

[3] How to Develop LSTM Models for Time Series Forecasting.

[How to Develop LSTM Models for Time Series Forecasting - MachineLearningMastery.com](https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/)

[4] Multivariate Time Series Forecasting with LSTMs in Keras.

[Multivariate Time Series Forecasting with LSTMs in Keras - MachineLearningMastery.com](https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras/)

[5] Deep-Learning-for-Time-Series-Forecasting

[Haoran-Zhao/Deep-Learning-for-Time-Series-Forecasting: Deep Learning for Time Series Forecasting (github.com)](https://github.com/Haoran-Zhao/Deep-Learning-for-Time-Series-Forecasting)

[6] Long Short-Term Memory (LSTM) Networks.

[nc.dvi (jku.at)](https://www.bioinf.jku.at/publications/older/2604.pdf)